

**PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X
DI SMK NEGERI 1 MAGELANG**

TUGAS AKHIR SKRIPSI

Diajukan Kepada Fakultas Teknik
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan
Guna Memperoleh Gelar Sarjana Pendidikan



Disusun Oleh :

PERMADI AFRIAN

NIM 14520241050

**PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA
JURUSAN PENDIDIKAN TEKNIK ELEKTRONIKA DAN INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2018**

**PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA
PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN
DASAR KELAS X DI SMK NEGERI 1 MAGELANG**

Oleh:

Permadi Afrian
14520241050

ABSTRAK

Selama ini algoritma menjadi tumpuan dalam melakukan pemrograman, tetapi kurang diminati oleh siswa karena dianggap tidak penting. Penelitian ini bertujuan untuk: (1) menghasilkan permainan dengan konsep *puzzle* sebagai media pembelajaran algoritma yang menarik dan menyenangkan, (2) memastikan kualitas permainan yang dikembangkan agar layak digunakan dengan melakukan pengujian aspek fungsionalitas, aspek materi, *compatibility testing*, *balance testing*, *compliance testing*, *playtesting*, dan *usability testing*.

Penelitian ini menggunakan metode penelitian dan pengembangan atau biasa disebut *Research and Development* (R&D). Model pengembangan yang digunakan adalah ADDIE. ADDIE memiliki 5 tahap pengembangan yaitu: (1) Analisis, (2) Desain, (3) Pengembangan, (4) Implementasi, dan (5) Evaluasi. Permainan yang dikembangkan terdiri dari 3 bagian utama: (1) menu utama yang berisi menu main, pengaturan, cara bermain, tentang permainan, profil pengembang, papan skor, dan keluar, (2) menu level yang berisi menu pemilihan level, dan (3) permainan yang berisi karakter, peta, dan blok perintah yang harus disusun.

Hasil dari penelitian ini yaitu: (1) permainan *puzzle* yang digunakan sebagai media pembelajaran konsep dasar algoritma, (2) kualitas permainan layak digunakan dengan hasil pengujian fungsionalitas dan *compatibility* mendapatkan persentase kelayakan 100% (sangat layak), pengujian oleh ahli media mendapatkan persentase kelayakan 85% (sangat layak), pengujian oleh ahli materi mendapatkan persentase kelayakan sebesar 83,93% (layak), pengujian oleh pengguna mendapatkan persentase kelayakan sebesar 82,05% (layak) dengan nilai *Cronbach's Alpha* sebesar 0,92 (*excellent*).

Kata kunci: permainan, *puzzle*, algoritma

**PUZZLE GAME DEVELOPMENT AS ALGORITHM LEARNING MEDIA IN
BASIC PROGRAMMING LESSON OF 10th GRADE IN SMK NEGERI 1**

MAGELANG

Oleh:

Permadi Afrian
14520241050

ABSTRACT

All this time the algorithm has become the foundation in programming, but less desirable by students because considered as not important. This research aim for: (1) develop a game using concept of puzzle as algorithm learning media which is interesting and fun, (2) ensure the quality of the game that has been developed to be feasible to use by doing testing with functionality aspects, material aspects, compatibility testing, balance testing, compliance testing, playtesting, and usability testing.

This research use Research and Development method known as R&D. The development model used in this research is ADDIE. ADDIE has 5 development phase: (1) Analyze, (2) Design, (3) Development, (4) Implementation, and (5) Evaluation. The developed game consist of 3 main section: (1) main menu that contains of play menu, settings, how to play, about the game, developer profile, scoreboard, and exit, (2) level menu that contains of level selection menu, (3) gameplay that contains of character, map, and command block to be arranged.

The result of this research is: (1) puzzle game used as algorithm learning media, (2) the quality of the game is feasible to use with the result of the functionality and compatibility testing get a feasibility percentage of 100% (very feasible), testing by media experts get a feasibility percentage of 85% (very feasible), testing by material experts get a feasibility percentage of 83,93% (feasible), testing by users get a feasibility percentage of 82,05% (feasible) with Cronbach's Alpha of 0,92 (excellent).

Keywords: game, puzzle, algorithm

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA

PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN

DASAR KELAS X DI SMK NEGERI 1 MAGELANG

Disusun oleh :

Permadi Afrian

NIM 14520241050

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan

Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan

Yogyakarta, 3 Juli 2018

Mengetahui,

Ketua Program Studi

Pendidikan Teknik Informatika,

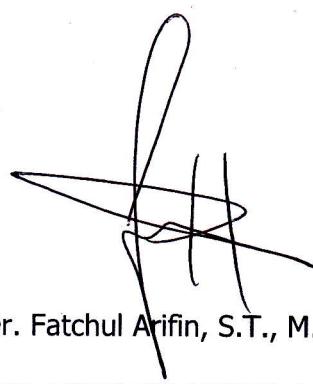


Handaru Jati, Ph.D.

NIP. 19740511 1999031 002

Disetujui,

Dosen Pembimbing,



Dr. Fatchul Arifin, S.T., M.T.

NIP. 19720508 199802 1 002

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Permadi Afrian
NIM : 14520241050
Program Studi : Pendidikan Teknik Informatika
Judul Tas : Pengembangan Permainan *Puzzle* sebagai Media
Pembelajaran Algoritma pada Mata Pelajaran
Pemrograman Dasar Kelas X di SMK Negeri 1
Magelang

menyatakan bahwa skripsi ini benar – benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata cara penulisan karya ilmiah yang telah lazim.

Yogyakarta, 3 Juli 2018

Yang menyatakan,



Permadi Afrian

NIM. 14520241050

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA

PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN

DASAR KELAS X DI SMK NEGERI 1 MAGELANG

Disusun oleh:

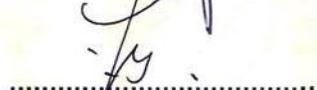
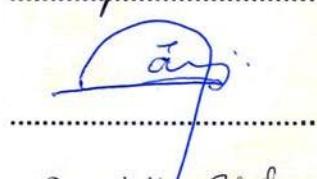
Permadi Afrian

NIM 14520241050

Telah dipertahankan di depan Tim Pengaji Tugas Akhir Skripsi Program Studi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta

Pada tanggal ...18... Juli ...2018...

TIM PENGUJI

Nama/Jabatan	Tanda Tangan	Tanggal
Dr. Fatchul Arifin, S.T., M.T. Ketua Penguji/Pembimbing		24/07 - 2018
Dr. Masduki Zakariyah, M.T. Sekretaris Penguji		24/07/2018
Dr. Putu Sudira, M.P. Penguji Utama		29/07/2018

Yogyakarta, ...25... Juli ...2018...

Fakultas Teknik Universitas Negeri Yogyakarta



Dekan,

Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001

HALAMAN MOTTO

Siapa yang bersabar pasti beruntung

Jangan pergi mengikuti ke mana jalan akan berujung, buat jalanmu sendiri dan
tinggalkan jejak

(Ralph Waldo Emerson)

Do your best and let god do the rest

(Ben Carson)

Just do it

HALAMAN PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

Kedua orang tua yaitu Ibu Budi Tri Puspitasari dan Bapak Widodo Budi Wibowo serta adik kandungku Pusbo Riandini yang telah mendoakan dan memberikan dukungan selama penyelesaian skripsi ini.

Winastri Harlinda yang telah setia menemani, mendoakan, dan menjadi alasan untuk selalu bersabar dan berusaha dalam menyelesaikan skripsi ini.

Bapak Ibu Guru dan Karyawan SMK Negeri 1 Magelang yang telah membantu dalam penyelesaian skripsi ini.

Teman – teman PTI 2014 yang telah menjadi pendorong dan motivasi untuk menyelesaikan skripsi ini.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka untuk memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan judul "**Pengembangan Permainan Puzzle sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang**" dapat disusun sesuai dengan harapan. Tugas Akhir Skripsi ini dapat diselesaikan tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkenaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada yang terhormat:

1. Bapak Dr. Fatchul Arifin, M.T selaku Dosen Pembimbing TAS dan Ketua Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta yang telah berkenan memberikan waktu luang, arahan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini.
2. Bapak Dr. Putu Sudira, M.P selaku Penguji Utama dan Validator instrumen TAS yang telah memberikan masukan dan arahan dalam penyusunan skripsi ini.
3. Bapak Dr. Masduki Zakariyah, M.T selaku Sekretaris yang telah memberikan koreksi perbaikan terhadap TAS ini.
4. Bapak Handaru Jati, Ph.D selaku Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesaiannya TAS ini.
5. Bapak DR. Widarto, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.

6. Bapak Drs. Nisandi, M.T selaku Kepala SMK Negeri 1 Magelang yang telah memberi ijin dan bantuan dalam pelaksanaan penelitian Tugas Akhir Skripsi ini.
7. Para guru dan staf SMK Negeri 1 Magelang yang telah memberikan bantuan memperlancar pengambilan data selama proses penelitian Tugas Akhir Skripsi ini.
8. Semua pihak, secara langsung maupun tidak langsung, yang tidak dapat disebutkan di sini atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini.

Akhirnya, semoga segala bantuan yang telah berikan semua pihak di atas menjadi amalan yang bermanfaat dan mendapatkan balasan dari Allah SWT dan Tugas Akhir Skripsi ini menjadi informasi bermanfaat bagi pembaca atau pihak lain yang membutuhkannya.

Yogyakarta, Juli 2018

Penulis



Permadi Afrian

NIM 14520241050

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
ABSTRAK.....	ii
ABSTRACT.....	iii
LEMBAR PERSETUJUAN	iv
SURAT PERNYATAAN.....	v
HALAMAN PENGESAHAN.....	vi
HALAMAN MOTTO.....	vii
HALAMAN PERSEMBAHAN.....	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvi
BAB I. PENDAHULUAN	1
A. Latar Belakang.....	1
B. Identifikasi Masalah.....	5
C. Batasan Masalah.....	5
D. Rumusan Masalah.....	6
E. Tujuan Penelitian	6
F. Spesifikasi Produk yang Dikembangkan	7
G. Manfaat Penelitian	7
BAB II. KAJIAN PUSTAKA.....	10
A. Kajian Teori	10
1. Pengembangan.....	10
2. Permainan.....	11
3. Media Pembelajaran	20
4. Algoritma	25
5. Materi Algoritma dalam Mata Pelajaran Pemrograman Dasar Kelas X	26
6. Sekolah Menengah Kejuruan	33
B. Kajian Penelitian yang Relevan	36
C. Kerangka Pikir.....	37

D. Pertanyaan Penelitian.....	39
BAB III. METODE PENELITIAN	40
A. Model Pengembangan	40
B. Prosedur Pengembangan	41
1. Analisis	41
2. Desain	42
3. Pengembangan.....	43
4. Implementasi.....	44
5. Evaluasi	45
C. Subjek Penelitian	48
D. Metode Pengumpulan Data.....	48
E. Instrumen Penelitian	49
1. Pengembang	49
2. Ahli Media	51
3. Ahli Materi.....	54
4. Pengguna.....	54
F. Teknik Analisis Data	56
BAB IV. HASIL PENELITIAN DAN PEMBAHASAN.....	59
A. Hasil Penelitian	59
1. Tahap Analisis	59
2. Tahap Desain	63
3. Tahap Pengembangan.....	76
4. Tahap Implementasi.....	87
5. Tahap Evaluasi	95
B. Pembahasan Hasil Penelitian	98
BAB V. SIMPULAN DAN SARAN.....	101
A. Simpulan.....	101
B. Saran.....	102
DAFTAR PUSTAKA.....	104
LAMPIRAN.....	107

DAFTAR TABEL

	Halaman
Tabel 1. Kompetensi Dasar dan Materi Pokok	26
Tabel 2. Simbol – simbol yang Ada pada Flowchart	28
Tabel 3. Kisi – kisi Instrumen <i>Compatibility Testing</i>	50
Tabel 4. Kisi – kisi Instrumen <i>Functionality Testing</i>	51
Tabel 5. Kisi – kisi Instrumen <i>Balance Testing</i>	51
Tabel 6. Kisi – kisi Instrumen <i>Compliance Testing</i>	52
Tabel 7. Kisi – kisi Instrumen <i>Playtesting</i>	53
Tabel 8. Kisi – kisi Instrumen <i>Usability Testing</i>	53
Tabel 9. Kisi – kisi Instrumen Ahli Materi.....	54
Tabel 10. Kisi – kisi Instrumen Pengguna	55
Tabel 11. Interval Skala <i>Likert</i>	56
Tabel 12. Interval Skala <i>Guttman</i>	57
Tabel 13. Kategorisasi Skala.....	57
Tabel 14. Aturan praktis <i>Cronbach's Alpha</i>	58
Tabel 15. Desain <i>Storyboard</i>	64
Tabel 16. <i>Mockup</i>	73
Tabel 17. Hasil <i>Compatibility Testing</i>	86
Tabel 18. Hasil Uji Fungsionalitas	86
Tabel 19. Hasil Pengujian <i>Balance Testing</i>	88
Tabel 20. Hasil Pengujian <i>Compliance Testing</i>	89
Tabel 21. Hasil Pengujian <i>Playtesting</i>	90
Tabel 22. Hasil Pengujian <i>Usability Testing</i>	91
Tabel 23. Hasil Uji Ahli Materi.....	93
Tabel 24. Hasil Uji oleh Pengguna	94

DAFTAR GAMBAR

	Halaman
Gambar 1. Konsep Dasar Algoritma	27
Gambar 2. Struktur Percabangan dalam <i>Flowchart</i>	29
Gambar 3. Struktur Percabangan <i>Flowchart</i> dengan Lebih dari 1 Kondisi....	30
Gambar 4. Struktur Perulangan Kondisi Awal dalam Bentuk <i>Flowchart</i>	31
Gambar 5. Struktur Perulangan Kondisi Akhir dalam Bentuk <i>Flowchart</i>	32
Gambar 6. Perulangan menggunakan Pencacahan dalam <i>Flowchart</i>	32
Gambar 7. Kerangka Pikir Penelitian	38
Gambar 8. Langkah – langkah Model Pengembangan ADDIE	41
Gambar 9. Contoh desain <i>mockup</i> yang dibuat pada <i>software</i> Pencil	43
Gambar 10. Contoh proses pengembangan permainan pada <i>software</i> Unity	44
Gambar 11. Siswa mencoba permainan yang telah dikembangkan	45
Gambar 12. <i>Flowchart</i> Menu Utama	67
Gambar 13. <i>Flowchart</i> Menu Level.....	68
Gambar 14. <i>Flowchart</i> Permainan.....	69
Gambar 15. <i>Flowchart</i> Tambah Blok Perintah	70
Gambar 16. <i>Flowchart</i> Geser Blok Perintah.....	70
Gambar 17. <i>Flowchart</i> Menghubungkan Blok Perintah	71
Gambar 18. <i>Flowchart</i> Menjalankan Blok Perintah	72
Gambar 19. Tampilan 1 Intro	77
Gambar 20. Tampilan 2 Intro	78
Gambar 21. Tampilan 3 Intro	78
Gambar 22. Tampilan Menu Utama.....	78
Gambar 23. Tampilan Panel Input Nama.....	79
Gambar 24. Tampilan Panel Cara Bermain.....	79
Gambar 25. Tampilan Panel Pengaturan.....	79
Gambar 26. Tampilan Panel Tentang Permainan.....	79
Gambar 27. Tampilan Panel Profil Pengembang	80
Gambar 28. Tampilan Panel Papan Skor	80
Gambar 29. Tampilan Pilih <i>Stage</i>	80
Gambar 30. Tampilan Pilih Level.....	81
Gambar 31. Tampilan Halaman Permainan	81

Gambar 32. Tampilan Panel Algoritma	82
Gambar 33. Tampilan Kemenangan	82
Gambar 34. Tampilan Kekalahan	82
Gambar 35. Tampilan proses pengembangan pada Unity 3D.....	85
Gambar 36. Tampilan proses pengembangan pada Visual Studio 2017	85
Gambar 37. Grafik hasil pengujian oleh ahli pengembangan permainan	86
Gambar 38. Grafik rata – rata hasil <i>balance testing</i>	89
Gambar 39. Grafik rata – rata hasil <i>compliance testing</i>	90
Gambar 40. Grafik rata – rata hasil <i>playtesting</i>	91
Gambar 41. Grafik rata – rata hasil <i>usability testing</i>	92
Gambar 42. Grafik hasil pengujian oleh ahli media	92
Gambar 43. Grafik hasil pengujian oleh ahli materi.....	93
Gambar 44. Grafik hasil pengujian oleh pengguna	95

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Keputusan Pembimbing Skripsi	108
Lampiran 2. Surat Ijin Penelitian.....	110
Lampiran 3. Surat Keterangan Telah Melaksanakan Penelitian.....	114
Lampiran 4. Validasi Instrumen	115
Lampiran 5. Hasil Pengujian Ahli Pengembangan Permainan.....	118
Lampiran 6. Hasil Pengujian Ahli Media	124
Lampiran 7. Hasil Pengujian Ahli Materi.....	132
Lampiran 8. Instrumen Pengujian Pengguna	138
Lampiran 9. Hasil Desain Level.....	141
Lampiran 10. Hasil Kode Program.....	144
Lampiran 11. Perhitungan Hasil Pengujian oleh Pengembang Permainan .	203
Lampiran 12. Perhitungan Hasil Pengujian oleh Ahli Media.....	205
Lampiran 13. Perhitungan Hasil Pengujian oleh Ahli Materi	209
Lampiran 14. Perhitungan Hasil Pengujian oleh Pengguna	211
Lampiran 15. Dokumentasi Pengujian oleh Pengguna.....	217

BAB I

PENDAHULUAN

A. Latar Belakang

Penerapan media pembelajaran di kelas mampu meningkatkan motivasi belajar siswa sehingga media tersebut dapat menunjang proses belajar mengajar. Menurut Kurnia (2015) bahwa penggunaan media pembelajaran yang berupa media visual dapat berpengaruh positif dimana hal tersebut dibuktikan dengan hasil belajar siswa yang meningkat setelah diberikan perlakuan menggunakan media pembelajaran tersebut. Penggunaan media pembelajaran berupa *technology-based* menurut Ghavifekr dan Rosdy (2015) mampu meningkatkan efektifitas selama proses belajar mengajar sehingga penyampaian materi oleh pendidik menjadi terbantu dengan diterapkannya suatu media pembelajaran berupa *technology-based*. Media pembelajaran juga membuat lingkungan belajar mengajar menjadi lebih menarik dan meningkatkan pemahaman siswa dalam belajar.

Berdasarkan hasil observasi selama Praktik Lapangan Terbimbing (PLT) di SMK Negeri 1 Magelang pada bulan September 2017 hingga November 2017 ditemukan bahwa sistem blok di SMK Negeri 1 Magelang menggunakan pola suatu mata pelajaran hanya ditempuh dalam jangka waktu 2 minggu sekali. Setiap pertemuan mata pelajaran terdapat jam pertemuan yang panjang yaitu 8 jam. Penyampaian materi menggunakan metode pembelajaran yang salah pada jam pelajaran yang panjang tersebut membuat siswa merasa bosan pada jam – jam akhir sehingga menurunkan pemahaman siswa terhadap materi yang disampaikan. Sehingga hal tersebut membuat materi yang disampaikan oleh pendidik kurang

dapat dipahami oleh siswa. Penerapan media pembelajaran yang interaktif diharapkan mampu menjadikan materi yang banyak dapat tetap membuat siswa tertarik, tidak merasa bosan ketika belajar, dan mudah dalam memahami pelajaran.

Pemrograman dasar adalah salah satu mata pelajaran yang terdapat di SMK dimana mata pelajaran tersebut ditempuh oleh siswa dengan paket keahlian Rekayasa Perangkat Lunak, Teknik Komputer dan Jaringan, dan Multimedia. Setiap minggu mata pelajaran tersebut ditempuh selama 8 jam. Berdasarkan hasil observasi selama PLT di SMK Negeri 1 Magelang, ketika pembelajaran pemrograman dasar siswa banyak tidak memperhatikan guru di kelas. Salah satu penyebab dari kurangnya minat siswa terhadap mata pelajaran ini adalah tidak diterapkannya suatu media pembelajaran penunjang penyampaian materi pada mata pelajaran terkait. Penyebab lain dari kurang minatnya siswa terhadap mata pelajaran pada kasus tertentu adalah karena siswa kurang termotivasi untuk belajar pemrograman karena berbagai alasan salah satunya adalah paksaan dari orang tua untuk masuk ke jurusan atau ke sekolah tersebut.

Pada mata pelajaran pemrograman dasar kelas X terdapat salah satu materi yang membahas mengenai konsep – konsep dasar dari algoritma dimana algoritma menjadi tumpuan dalam melakukan pemrograman. Algoritma secara singkat adalah langkah sistematis yang digunakan dalam pemecahan masalah. Menurut Munir (2011: 12 – 13) algoritma dalam pemrograman itu penting untuk memaksimalkan kinerja dari pemroses dalam menjalankan program sehingga algoritma yang dibuat harus efektif. Pada kenyataanya masih kurang adanya minat belajar siswa dalam mempelajari algoritma karena siswa merasa materi algoritma

tidak terlalu penting. Siswa merasa bahwa mereka dapat langsung melakukan penulisan kode program tanpa harus membuat algoritma terlebih dahulu sehingga anggapan tersebut membuat ketertarikan siswa terhadap belajar algoritma berkurang. Penerapan suatu media pembelajaran diharapkan menjadi alternatif untuk meningkatkan minat belajar siswa dan menyadarkan bahwa algoritma itu sangat penting dalam pemrograman.

Bermain adalah aktivitas yang disukai oleh banyak kalangan terutama anak – anak karena merupakan kegiatan yang menyenangkan. Menurut Ismail (2009: 20) bermain merupakan kebutuhan setiap orang mulai dari anak – anak hingga dewasa karena bermain menumbuhkan rasa gembira dan kepuasan secara emosional. Bermain juga bermanfaat secara sosial karena ketika seseorang bermain dengan temannya atau siapapun itu mereka akan berkomunikasi, belajar menghargai satu sama lain, dan menumbuhkan rasa kebersamaan. Bermain juga dapat menambah pengetahuan dan pengalaman dimana seseorang yang bermain dengan hal – hal yang baru akan mengetahui yang sebelumnya belum mereka ketahui. Menurut Gray (2011) anak – anak didesain secara alami untuk bermain karena kegiatan bermain menyenangkan seperti pendapat Vanderschuren (2010) yang menyatakan bahwa bermain menimbulkan perasaan senang seperti merasakan makanan yang sangat enak. Namun masih sedikit orang yang memandang bermain sebagai sarana untuk belajar terutama belajar dengan suatu media permainan karena bermain hanya dilihat sebagai sesuatu yang membuang – buang waktu dan hanya untuk bersenang – senang.

Bersamaan dengan observasi yang telah dilakukan di SMK Negeri 1 Magelang menemukan bahwa siswa disana banyak bermain permainan pada

komputer atau ponsel. Siswa sering bermain permainan tersebut di kelas ketika pelajaran berlangsung dimana guru terkadang tidak mengetahui hal tersebut karena siswa bermain secara sembunyi – sembunyi. Hal tersebut terjadi karena siswa merasa bahwa permainan lebih terasa menyenangkan daripada belajar atau memperhatikan penjelasan materi oleh guru di kelas. Tindakan tersebut dapat menganggu kegiatan belajar mengajar karena siswa lebih berkonsentrasi pada bermain permainan dibandingkan memperhatikan pelajaran.

Juanda, Gunawan, dan Mujiburrohman (2012) menyatakan bahwa permainan edukasi masih jarang dimanfaatkan oleh guru ataupun orang tua. Hal tersebut dibuktikan dengan permainan edukasi tidak dimanfaatkan sebagai media pembelajaran oleh berbagai pihak di SMK Negeri 1 Magelang. Potensi dari penggunaan permainan sebagai media pembelajaran sangat besar di sekolah mengingat kondisi kelas yang memiliki fasilitas memadai dimana setiap siswa menghadap pada komputer masing – masing di kelas. Berdasarkan fasilitas yang memadai tersebut, maka lebih baik mengarahkan aktifitas bermain permainan untuk hal yang lebih bermanfaat seperti sebagai alat untuk menyampaikan materi yaitu berupa suatu media pembelajaran.

Pemilihan konsep permainan *puzzle* sebagai media pembelajaran yaitu karena menurut Levitin dan Papalaskari (2002) konsep dari algoritma sendiri berbentuk logika sehingga permainan *puzzle* cocok digunakan sebagai media yang akan digunakan pada pembelajaran algoritma. *Puzzle* sendiri memiliki berbagai konsep yang berbeda mulai dari potongan – potongan yang disatukan atau biasa disebut dengan *jigsaw* hingga *puzzle* dengan model labirin. Bentuk *puzzle* secara umum seperti suatu teka – teki yang dimaksudkan agar pemain memecahkan

masalah dimana sesuatu yang diketahui digunakan untuk menyelesaikan masalah yang belum diketahui. Hal tersebut sama dengan konsep dari algoritma yaitu suatu penyelesaian masalah, maka dari itu permainan *puzzle* dapat digunakan sebagai media pembelajaran pada materi algoritma.

Berdasarkan masalah yang telah dipaparkan maka dapat diketahui bahwa media pembelajaran dalam materi algoritma yang ada di SMK Negeri 1 Magelang masih belum diterapkan sehingga memerlukan suatu pengembangan dan penelitian untuk mengembangkan media tersebut. Adanya aktifitas siswa yang sering bermain permainan di kelas maupun di luar kelas, hal tersebut dapat diarahkan pada hal yang lebih baik seperti untuk menyampaikan materi. Media yang dikembangkan berupa permainan *puzzle* dimana konsep dari algoritma yang logis dan permainan *puzzle* yang memerlukan pemikiran logis cocok digunakan untuk menyampaikan materi algoritma.

B. Identifikasi Masalah

Dari pemaparan latar belakang masalah maka dapat diidentifikasi beberapa permasalahan sebagai berikut.

1. Penyampaian pembelajaran yang tidak tepat pada jam pelajaran yang panjang menyebabkan siswa merasa bosan pada jam – jam akhir pembelajaran.
2. Terdapat beberapa kasus dimana siswa merasa terpaksa untuk belajar pada jurusan atau sekolah tersebut karena tuntutan orang tua sehingga motivasi belajar siswa kurang.
3. Minat belajar siswa terhadap materi algoritma kecil karena merasa materi tersebut tidak terlalu penting.

4. Siswa sering bermain permainan komputer dan ponsel di kelas karena siswa merasa bermain permainan lebih menyenangkan daripada memperhatikan guru mengajar.
5. Masih sedikit orang yang melihat permainan sebagai alat edukasi karena dirasa hanya sebagai alat untuk bersenang – senang dan membuang waktu.

C. Batasan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah disebutkan maka permasalahan akan dibatasi pada pemecahan masalah yaitu siswa yang merasa materi algoritma kurang penting dan pemanfaatan kondisi siswa yang sering bermain permainan di kelas, maka dikembangkan sebuah media pembelajaran interaktif sebagai bahan pembelajaran berupa permainan *puzzle* di SMK Negeri 1 Magelang.

D. Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan di atas, maka rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Permainan *puzzle* seperti apa yang harus dikembangkan agar dapat digunakan sebagai media pembelajaran materi algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
2. Bagaimana kelayakan permainan *puzzle* sebagai media pembelajaran materi algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?

E. Tujuan Penelitian

Berdasarkan pada rumusan masalah di atas, maka tujuan dari penelitian ini adalah sebagai berikut.

1. Menghasilkan permainan *puzzle* yang dapat digunakan sebagai media pembelajaran materi algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang.
2. Mengetahui kelayakan permainan *puzzle* sebagai media pembelajaran materi algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang.

F. Spesifikasi Produk yang Dikembangkan

Spesifikasi produk yang dikembangkan dalam penelitian ini adalah sebuah produk media pembelajaran berupa permainan dengan deskripsi sebagai berikut.

1. Permainan dengan konsep permainan *puzzle* dimana pemain memecahkan teka – teki yang diberikan pada setiap *level* permainan.
2. Permainan memiliki 3 halaman utama yaitu halaman menu utama, halaman pemilihan level, dan halaman permainan.
3. Halaman menu utama berisi tombol – tombol menu seperti tombol main, tombol pengaturan, tombol cara bermain, tombol keluar, tombol profil pengembang, tombol tentang permainan, dan tombol papan skor.
4. Halaman pemilihan level memiliki beberapa unsur terkait pemilihan level yaitu pada tahap pertama pengguna memilih *stage*, pada *stage* yang dipilih pengguna memilih level.
5. Halaman permainan berisi level permainan, tombol main, tombol ulangi, tombol jeda, dan tombol buka halaman algoritma dimana pada halaman

algoritma terdapat berbagai blok perintah yang harus disusun oleh pengguna untuk menjalankan karakter permainan.

6. Sistem *scoring* pada permainan ini menggunakan sistem yang memiliki skor paling kecil adalah yang lebih baik dimana skor ditentukan dari penggunaan blok perintah pada susunan dengan asumsi bahwa semakin sedikit pengguna menyusun blok perintah maka algoritma yang dijalankan menjadi semakin efektif.
7. Permainan dengan ekstensi *.exe* dimainkan pada komputer menggunakan *mouse* dan *keyboard*.
8. Teka – teki yang diberikan disesuaikan dengan materi algoritma yang ada pada mata pelajaran pemrograman dasar kelas X.
9. Permainan menggunakan sistem belajar sambil bermain sehingga siswa secara sadar ataupun tidak sadar akan mempelajari materi yang diberikan melalui permainan.

G. Manfaat Penelitian

Berdasarkan tujuan yang telah dipaparkan, maka penelitian ini diharapkan dapat bermanfaat. Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut.

1. Bagi Siswa

Penelitian ini dapat menghasilkan permainan yang layak digunakan sebagai sarana belajar siswa mengenai materi algoritma sehingga mereka dapat belajar secara mandiri di luar jam pelajaran.

2. Bagi Guru

Penelitian ini dapat menghasilkan permainan sebagai alat bantu guru dalam menyampaikan materi – materi algoritma sehingga mempermudah guru dalam penggambaran mengenai konsep algoritma yang dipelajari.

3. Bagi Sekolah

Penelitian ini dapat bermanfaat dan mampu dijadikan referensi sebagai bahan untuk mengembangkan media pembelajaran pada mata pelajaran yang lainnya.

4. Bagi Universitas

Penelitian ini dapat dijadikan sebagai kajian dan bahan referensi pada penelitian – penelitian yang akan datang.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Pengembangan
 - a. Pengertian Pengembangan

Pengembangan berasal dari kata dasar “kembang” yang berarti “buka lebar” atau “bentang”. Pengembangan dalam Kamus Besar Bahasa Indonesia (2016) berarti proses, cara, atau perbuatan mengembangkan. Pengembangan yang dimaksud dalam penelitian ini adalah pengembangan perangkat lunak komputer. Pressman dan Maxim (2015: 27) menyatakan bahwa dalam pengembangan perangkat lunak komputer meliputi proses – proses, metode – metode, dan alat – alat yang membuat sistem berbasis komputer yang kompleks memungkinkan untuk dibangun dengan kualitas yang terjamin. Rosa dan Shalahuddin (2015: 4) menyatakan bahwa rekayasa perangkat lunak merupakan suatu proses pembangunan dengan menggunakan prinsip rekayasa untuk menghasilkan suatu perangkat lunak yang dapat dijalankan secara efisien pada mesin. Berdasarkan pernyataan yang telah dijabarkan maka dapat disimpulkan bahwa pengembangan dalam konteks penelitian ini adalah suatu proses atau cara untuk membangun suatu produk perangkat lunak tertentu melalui metode – metode menggunakan alat – alat dengan tujuan menghasilkan suatu perangkat lunak yang dapat dijalankan secara efisien dengan kualitas yang terjamin.

b. Model Pengembangan

Branch (2009: 2) menyatakan bahwa ADDIE (*Analyze, Design, Develop, Implement, Evaluate*) merupakan konsep yang digunakan untuk mengembangkan produk dengan filosofi edukasi bahwa produk atau metode yang dikembangkan berpusat pada pembelajaran oleh siswa. Jeuring, Rooij, dan Pronost (2013) menyatakan bahwa ADDIE secara luas digunakan sebagai model pengembangan untuk pengembangan metode pengajaran, buku, dan permainan edukasi. Berdasarkan pernyataan maka dapat disimpulkan bahwa ADDIE adalah merupakan konsep yang digunakan untuk mengembangkan suatu produk atau metode dengan tujuan edukasi dimana pembelajaran berpusat pada siswa, dengan demikian model pengembangan ADDIE dapat digunakan sebagai kerangka pengembangan suatu permainan edukasi.

2. Permainan

a. Pengertian Permainan

Main menurut Kamus Besar Bahasa Indonesia (2016) berarti melakukan permainan untuk menyenangkan hati. Bermain dalam Kamus Besar Bahasa Indonesia (2016) berarti melakukan aktifitas untuk bersenang – senang. Permainan menurut Kamus Besar Bahasa Indonesia (2016) adalah sesuatu yang digunakan untuk bermain. Permainan dapat berarti aktivitas yang memiliki aturan tertentu untuk menghubungkan antara kenyataan dengan khayalan untuk mencapai suatu tujuan atau capaian dimana biasanya dimaksudkan untuk bersenang – senang (Drysdale, 2011). Menurut Ismail (2009: 27) permainan memiliki arti yaitu suatu aktifitas dengan aturan menang-kalah dimana seseorang memperoleh kesenangan dan kepuasan melalui kemenangan yang didapatkan dari

bermain permainan tersebut. Yunanto dan Chandrawati (2016) menyatakan bahwa permainan adalah sebuah aplikasi yang biasa digunakan secara luas dalam kehidupan sehari – hari dimana biasanya dimainkan oleh anak – anak muda dengan tujuan untuk sarana hiburan dan penyegaran pikiran. Berdasarkan kajian diatas maka dapat disimpulkan bahwa permainan dalam konteks penelitian ini adalah sebuah aplikasi atau perangkat lunak berupa media interaktif yang dimainkan pada perangkat keras elektronik seperti komputer dan ponsel genggam yang digunakan untuk mencari kesenangan dan kepuasan.

b. Permainan Edukasi

Permainan edukasi yaitu suatu alat untuk dimainkan dengan tujuan sebagai alat untuk mencari kesenangan dan kepuasan sekaligus dapat mendidik dan meningkatkan kemampuan berbahasa, berpikir serta bergaul anak dengan lingkungan (Ismail, 2009: 112). Menurut Yunanto dan Chandrawati (2016) permainan edukasi adalah permainan digital yang merujuk pada *game* yang dimainkan pada komputer yang dibuat untuk tujuan pendidikan. Clark Aldrich menyatakan (Yunanto dan Chandrawati, 2016) bahwa terdapat 5 *genre* yang dapat digunakan dalam *game* edukasi yaitu *Serious Game*, *Educational Simulation*, *Frame Game*, *Class Game*, dan *Virtual World*. Penggunaan permainan edukasi pada anak memiliki manfaat – manfaat yang positif (Ismail, 2009: 113). Manfaat – manfaat tersebut adalah sebagai berikut.

1) Melatih konsentrasi anak

Dengan penggunaan permainan edukasi yang diterapkan dalam mengajar dapat membantu anak dalam mempertahankan daya tangkap karena bahan atau media pengajaran yang mempunyai daya tarik.

2) Mengajar dengan lebih cepat

Proses pemberian materi kepada anak sering kali terhambat karena waktu yang tersedia sangat terbatas. Dengan bantuan media berupa permainan edukasi, penyampaian materi dapat dilakukan dalam waktu yang lebih singkat namun juga mencapai hasil mengajar yang lebih banyak.

3) Mengatasi keterbatasan waktu

Untuk belajar sesuatu yang tidak bisa diulang kembali seperti sejarah, dengan permainan edukasi maka dapat disimulasikan bagaimana peristiwa – peristiwa yang pernah terjadi sehingga masalah keterbatasan waktu dapat teratasi dengan mudah.

4) Mengatasi keterbatasan tempat

Sering kali karena fasilitas yang disediakan terbatas, maka proses belajar mengajar terhambat. Dengan adanya media seperti permainan edukasi dapat mensimulasikan berbagai fasilitas dengan mudah.

5) Mengatasi Keterbatasan bahasa

Kemampuan anak dalam menyerap pengertian suatu kata masih terbatas. Kebanyakan anak – anak tidak dapat memahami istilah – istilah tertentu. Dengan adanya permainan edukasi akan membuat anak mengerti istilah tertentu melalui simulasi yang diberikan.

6) Membangkitkan emosi manusia

Penyampaian materi yang dilakukan dengan gambar – gambar interaktif dan suara akan lebih berhasil, hal tersebut akan membangkitkan emosi manusia dalam konteks keingintahuan.

7) Menambah daya pengertian

Penggunaan permainan edukasi akan membuat anak mengerti mengenai materi yang disampaikan lebih baik melalui penyerapan materi dengan indra pengelihatan dan pendengaran.

8) Menambah ingatan anak

Dalam hal tertentu, menggunakan permainan edukasi untuk menyampaikan materi akan memberikan pengalaman berbeda sehingga ingatan anak akan tertanam lebih baik ketika menyerap materi yang diberikan.

9) Menambah kesegaran dalam mengajar

Penyampaian materi yang biasanya hanya melalui ceramah membuat anak merasa bosan, penggunaan permainan edukasi akan menyegarkan suasana belajar dan mampu membangkitkan motivasi belajar anak.

Setiap elemen yang ada pada permainan akan mempengaruhi kondisi pemain untuk bermain. Dengan adanya elemen – elemen yang tepat pada sebuah permainan akan menarik perhatian dan memotivasi seorang pemain untuk memainkannya tanpa paksaan. Berikut ini 8 elemen yang harus ada pada permainan edukasi menurut Kapp (2014) agar proses belajar melalui permainan edukasi membuat pemain semakin tertarik dan termotivasi.

1) Misteri

Misteri memberikan kesan bahwa ada jarak antara yang diketahui dengan yang tidak diketahui dimana pemain yang merasakan jarak tersebut akan mencoba mencari informasi mengenai sesuatu hal yang tidak diketahui untuk menutup jarak tersebut. Jadi misteri secara singkat adalah apa dan kenapa hal tersebut bisa terjadi. Elemen ini dimunculkan untuk memancing rasa penasaran pada pemain sehingga mereka termotivasi untuk mencari informasi – informasi yang belum diketahui. Permainan dengan elemen misteri memberikan informasi secara terbatas, tidak semua informasi diberikan kepada pemain agar ada jarak yang harus pemain tutupi. Informasi yang diberikan kepada pemain cukup hanya informasi bahwa jarak itu ada sehingga memberikan dorongan kepada pemain untuk menelusuri informasi lebih dalam.

2) Tindakan

Permainan yang bagus selalu dimulai dengan tindakan. Pemain harus melakukan sesuatu untuk menyelesaikan tugas sesuai aturan permainan. Sedikit permainan yang memaksa pemain untuk membaca sesuatu yang menghabiskan tenaga. Sebaliknya, banyak permainan dimulai dengan *tutorial* tentang bagaimana cara melakukan ini dan itu dengan langsung mengarahkan pemain untuk melakukannya.

3) Tantangan

Setiap permainan yang menantang membuat pemain menikmatinya. Pemilihan tingkat kesulitan tantangan juga mempengaruhi kondisi pemain, untuk itu permainan harus mengetahui tingkat kesulitan

seperti apa yang cocok bagi pengguna. Tingkat kesulitan yang terlalu mudah ditebak oleh pemain juga kurang bagus, jadi tingkat kesulitan yang cukup baik adalah pemain masih harus berpikir secara mendalam namun masih dapat dilakukan.

4) Beresiko

Dalam suatu permainan seorang pemain mungkin saja kehabisan poin nyawa sehingga harus mengulang dari awal atau kehilangan semua koin yang telah dikumpulkan karena suatu tindakan yang salah. Pemain harus mengambil risiko untuk kehilangan sesuatu dalam pengambilan keputusan untuk melakukan sebuah tindakan. Ketika seseorang dalam bahaya, mereka menjadi lebih fokus dalam bermain sehingga perhatian pemain akan tertuju pada permainan.

5) Tak terduga

Ketika permainan berlangsung, pemain tidak akan tahu bagaimana hasil yang akan muncul. Pemain seharusnya tidak dapat memprediksi bagaimana alur cerita permainan yang dimainkan sehingga membuat permainan menjadi lebih menarik. Hal ini akan membuat pemain bertanya – tanya apakah mereka akan mampu menyelesaikan tugas yang diberikan atau tidak.

6) Kesempatan untuk menguasai

Kesempatan diberikan agar dalam prosesnya pemain akan mencoba menggapai titik dimana mereka telah menguasai permainan tersebut. Pemain akan merasa senang dan bangga ketika mereka dapat menaklukan sesuatu seperti memecahkan sebuah teka – teki,

mengalahkan bos musuh, atau mencapai garis finish pada posisi pertama.

Pencapaian yang diraih oleh pemain dapat diberikan kepada pemain melalui penghargaan dalam permainan seperti medali.

7) Kemajuan yang terlihat

Permainan memperlihatkan bagaimana capaian pemain. Melalui permainan perlihatkan tingkat penyelesaian permainan kepada pemain melalui *progress bar*, atau medali, atau *achievement*, atau juga perubahan tampilan karakter yang dimainkan.

8) Emosional

Permainan memberikan pemain perasaan emosional mulai dari frustasi hingga kegembiraan, dari kesedihan hingga marah, ataupun kebahagiaan. Permainan membawa keluar seluruh komponen kemanusiaan dari dalam diri. Permainan tidak takut akan emosional pemain, melainkan mendorong dan membawa perasaan emosional itu keluar.

c. Permainan *Puzzle*

Kata *puzzle* berasal dari bahasa Inggris yang dalam bahasa Indonesia berarti membingungkan. Menurut *English Oxford Living Dictionaries* (2017) *puzzle* dapat berarti suatu *game*, permainan, atau mainan yang memiliki suatu masalah untuk diselesaikan dengan tujuan mengasah kecerdasan pemain. Menurut Kim (2006) *puzzle* berarti suatu permainan yang memiliki aturan – aturan tertentu dengan memperlihatkan sisi permasalahan untuk dipecahkan melalui kecerdikan dan kesabaran. *Puzzle* itu menyenangkan, berikut ini alasan menurut Kim (2006) mengapa puzzle terasa menyenangkan.

1) Baru

Puzzle adalah bentuk dari suatu permainan yang berdasarkan batasan aturan dari kehidupan sehari hari dimana dalam menyelesaikan suatu permasalahan pada *puzzle* akan membawa pemain kepada sesuatu hal yang baru namun familiar sehingga membuat permainan menjadi menarik.

2) Tidak terlalu susah, tidak terlalu mudah

Puzzle yang terlalu sulit akan membuat pemain kecewa dan mudah bosan ketika bermain, sebaliknya *puzzle* yang terlalu sulit akan membuat pemain merasa berkecil hati sehingga enggan untuk memainkannya. Yang membuat *puzzle* begitu menarik adalah tingkat kesulitan yang masih dapat diselesaikan namun dengan cara – cara unik sehingga membuat perasaan bahagia ketika menyelesaikannya.

3) Menjebak

Penyajian teka – teki permasalahan pada *puzzle* menjadi poin utama dalam aktivitas permainannya. *Puzzle* yang memiliki teka – teki yang menjebak mampu membuat pemain untuk berpikir lebih dalam peralihan persepsi penyelesaian.

d. Jenis – jenis permainan *puzzle*

Menurut Clontz (2018) dalam artikel pada *website*-nya yang berjudul *Puzzle Types* menyatakan bahwa terdapat 8 jenis permainan *puzzle*, yaitu sebagai berikut ini.

1) *Mechanical Puzzles*

Puzzle Mekanikal adalah jenis yang sangat umum dikenal oleh masyarakat. *Puzzle* jenis ini bisa diselesaikan melalui *trial-and-error*, namun beberapa model *puzzle* dengan jenis ini seperti *Rubik's Cube* terdapat diselesaikan menggunakan rumus – rumus atau metode – metode logika tertentu. Contoh dari *puzzle* jenis ini adalah *Jigsaw*, *Nail Puzzle*, *Rubik's Cube*, dan lainnya.

2) *Logic Puzzles*

Logic puzzle biasanya berupa *puzzle* dengan karakteristik memiliki garis kisi yang harus diselesaikan melalui aturan – aturan yang telah ditetapkan. Contoh dari *puzzle* jenis ini adalah *Sudoku*, *Picross*, *Nonograms*, dan lainnya.

3) *Math Puzzles*

Puzzle matematika adalah *puzzle* yang secara umum berbentuk aljabar dengan permasalahan matematika seperti "carilah nilai x" atau dengan persoalan narasi seperti "terdapat 100 orang dalam suatu ruangan, carilah jumlah interaksi salaman yang terjadi jika setiap orang bersalaman satu sama lain".

4) *Cryptic Puzzles*

Cryptic puzzle selalu memiliki tujuan atau jawaban yang sudah jelas ditentukan. Terkadang terdapat petunjuk secara tidak langsung dalam suatu kalimat atau kata yang disediakan untuk menyelesaikan *puzzle*. Contoh dari *puzzle* jenis ini adalah permainan ruang tertutup untuk mencari jalan keluar dari suatu ruangan.

5) *Word Puzzles*

Puzzle kata adalah jenis *puzzle* yang memerlukan pengetahuan dalam hal bahasa. Pemain biasanya dituntut untuk menebak kata yang memungkinkan dari susunan – susunan huruf yang disediakan. Contoh dari *puzzle* ini adalah Teka – teki silang, *Boggle*, *Scramble*, dan lainnya.

6) *Trivia Puzzles*

Hampir sama dengan *cryptic puzzle*, *puzzle* jenis ini biasanya memiliki desain khusus. *Trivia puzzle* seringkali terdapat unsur *cryptic puzzle* di dalamnya yang begitu jelas dalam kasus tertentu. Pada kasus yang lain, pemain diharuskan memiliki pengetahuan lebih untuk menyelesaikan permasalahan yang disediakan atau kemampuan untuk memahami topiknya.

7) *Riddles*

Riddle adalah jenis *puzzle* berupa pernyataan – pernyataan logika yang saling berhubungan membentuk sebuah misteri. Pemain harus memahami setiap pernyataan untuk menemukan jawabannya.

8) *Pattern Guessing*

Pattern guessing atau tebak pola adalah jenis *puzzle* yang biasanya berupa teka – teki seperti *math puzzle* dimana pemain harus menebak pola yang terjadi pada *puzzle*. Contohnya adalah "1, 4, 8, ...".

3. Media Pembelajaran

a. Pengertian Media Pembelajaran

Kata *media* berasal dari bahasa Latin yaitu *medius* dimana *medius* berarti tengah, perantara atau pengantar (Arsyad, 2017: 3). Gerlach dan Ely (Arsyad,

2017: 3) menyatakan bahwa media secara garis besar yaitu manusia, materi, atau kejadian yang membuat siswa mendapatkan pengetahuan, ilmu, ketrampilan, atau sikap. Wati (2016: 2) mengartikan media sebagai alat bantu yang digunakan untuk menyampaikan pesan dengan maksud agar tujuan pembelajaran dapat tersampaikan. Dalam Kamus Besar Bahasa Indonesia (2016) media adalah alat, sarana komunikasi (koran, majalah, radio, televisi, film, poster, dan spanduk), perantara atau penghubung, zat hara, atau bahan yang digunakan dalam pemasukan dan penyimpanan data di komputer atau dalam perekaman hasil komputer. Arsyad (2017: 3) menyatakan bahwa pengertian media dalam proses pembelajaran dapat berupa alat – alat grafis, fotografis, atau alat elektronik yang memiliki fungsi untuk menangkap, memproses, dan menyusun kembali informasi visual atau verbal. Dengan begitu dapat diartikan bahwa media yaitu alat yang digunakan sebagai penghubung atau perantara yang digunakan untuk menyampaikan suatu informasi.

Media pembelajaran menurut Arsyad (2017: 4) yaitu media yang membawa pesan – pesan atau informasi yang memuat unsur – unsur pengajaran atau bertujuan secara instruksional untuk membantu penyampaian pengetahuan atau ilmu dalam suatu proses pembelajaran. Wati (2016: 3) menyampaikan bahwa media pembelajaran meliputi alat yang secara fisik digunakan dalam penyampaian materi belajar dimana media pembelajaran adalah suatu sumber belajar yang berisi materi instruksional guna memotivasi siswa untuk belajar. Arsyad (2017: 6) menyatakan bahwa media pembelajaran berupa fisik yaitu *hardware* yaitu benda yang dapat dilihat, didengar, atau diraba dengan pancaindera dan *software* yaitu kandungan pesan yang terdapat dalam *hardware* yaitu isi materi yang ingin

disampaikan kepada siswa. Dari pengertian – pengertian diatas dapat dipahami bahwa media pembelajaran yaitu alat yang digunakan sebagai sarana atau perantara untuk membantu penyampaian materi guna memotivasi siswa untuk belajar sehingga penyampaian materi pembelajaran dapat diterima oleh siswa dengan baik.

Melalui Arsyad (2017: 15-17) Gerlach dan Ely menyampaikan tiga ciri – ciri media pembelajaran, yaitu sebagai berikut.

1) Fiksatif

Ciri – ciri fiksatif berarti kemampuan media dalam merekam, menyimpan, dan merekonstruksi suatu peristiwa atau objek. Dengan ciri ini maka media dapat digunakan untuk merekam kejadian atau objek yang terjadi pada suatu waktu tertentu lalu ditransportasikan tanpa mengenal waktu. Hal ini sangat penting dalam penggunaan media sebagai alat pembelajaran karena kejadian – kejadian atau objek tertentu yang telah direkam dapat digunakan setiap saat untuk disampaikan kepada siswa.

2) Manipulatif

Ciri – ciri manipulatif mampu membuat suatu kejadian atau objek disampaikan dalam waktu yang singkat ataupun waktu yang diperpanjang guna mempermudah penyampaian mekanisme kejadian atau objek yang sedang dipelajari. Hal ini dapat berupa video atau film yang diedit untuk dipercepat atau diperlambat, sebagai contoh adalah proses siklus kehidupan kupu – kupu berupa video yang telah dipercepat atau kinerja mesin berupa video yang diperlambat. Jadi ciri manipulatif berarti

penyajian media yang dapat diubah agar sesuai dengan waktu penyampaian materi sehingga proses pembelajaran menjadi lebih efektif.

3) Distributif

Ciri – ciri distributif pada media memungkinkan suatu objek atau kejadian ditransportasikan melalui ruang, dan secara bersamaan kejadian tersebut disajikan kepada siswa dengan stimulus pengalaman yang sama pada kejadian yang sama. Dengan begitu materi yang disampaikan pada siswa satu dengan yang lainnya akan sama tanpa ada perubahan pengalaman penyampaian.

Wati (2016: 10-11) menyampaikan 4 fungsi media pembelajaran, yaitu sebagai berikut ini.

1) Atensi

Media pembelajaran berfungsi sebagai alat untuk menarik dan mengarahkan perhatian siswa agar berkonsentrasi pada materi pembelajaran yang sedang disampaikan.

2) Afektif

Media pembelajaran berfungsi agar penyampaian materi membuat siswa nyaman ketika mempelajari materi sehingga proses pembelajaran yang sedang dilaksanakan akan membuat siswa menjadi lebih emosional dalam memahami materi pembelajaran.

3) Kognitif

Tampilan pada media pembelajaran mempengaruhi proses pemahaman materi. Fungsi kognitif yaitu media pembelajaran berfungsi

agar materi pembelajaran yang disampaikan melalui tampilan yang ada pada media dapat memperlancar pencapaian tujuan pembelajaran.

4) Kompensatoris

Fungsi ini membantu siswa yang sulit dalam pemahaman materi yang hanya menggunakan tulisan atau hanya secara verbal. Media pembelajaran memberikan konteks untuk memahami teks dan membantu siswa mengorganisasikan informasi.

Melalui Arsyad (2017: 23-25) Kemp dan Dayton menyatakan 3 fungsi utama media pembelajaran, yaitu sebagai berikut ini.

1) Memotivasi

Media pembelajaran dapat memotivasi siswa untuk belajar sehingga dalam proses penyampaian materi siswa secara sukarela tanpa paksaan mencoba memahami materi yang diberikan. Dengan begitu pencapaian tujuan belajar dapat dilakukan dengan baik.

2) Menyajikan Informasi

Media pembelajaran sudah seharusnya menyajikan informasi. Informasi yang diberikan yaitu materi – materi yang sedang dipelajari oleh siswa dimana informasi tersebut bersifat umum, sebagai pengantar, ringkasan laporan, atau pengetahuan latar belakang.

3) Memberi Instruksi

Informasi yang terdapat dalam media pembelajaran harus melibatkan siswa dalam bentuk aktivitas yang nyata sehingga proses pembelajaran dapat terjadi. Media pembelajaran harus dapat

memberikan pengalaman yang menyenangkan dan memenuhi kebutuhan siswa dalam proses belajarnya.

4. Algoritma

a. Pengertian Algoritma

Dalam Kamus Besar Bahasa Indonesia (2016) algoritma berarti prosedur sistematis untuk memecahkan masalah matematis dalam langkah – langkah terbatas atau urutan logis pengambilan keputusan untuk pemecahan masalah. Menurut Maryono dan Pambudhi (2014: 8) memberikan definisi bahwa algoritma adalah urutan langkah – langkah penyelesaian masalah yang disusun secara sistematis dan logis. Menurut Ananda dkk (2009: 3) algoritma adalah metode yang digunakan untuk menyelesaikan kasus tertentu. Menurut Munir (2011: 4) algoritma adalah langkah – langkah yang runtut digunakan untuk menyelesaikan masalah. Menurut Levitin (Munir, 2011: 4) algoritma adalah deretan instruksi yang jelas untuk memecahkan masalah dengan tujuan untuk mendapatkan keluaran yang diinginkan dari suatu masukan tertentu dalam waktu yang terbatas atau efisien. Dengan demikian dapat dikatakan bahwa algoritma adalah langkah – langkah yang sistematis digunakan untuk menyelesaikan masalah dengan tujuan agar masalah dapat diselesaikan secara efisien.

b. Ciri – ciri Algoritma

Algoritma yang dibuat harus benar dan algoritma harus memiliki titik penyelesaian atau titik berhenti. Menurut Knuth (Munir, 2011: 12) menyampaikan lima ciri penting yang harus dimiliki sebuah algoritma, yaitu sebagai berikut ini.

- 1) Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
- 2) Setiap langkah harus didefinisikan dengan tepat dan tidak ambigu.

- 3) Algoritma harus memiliki masukan.
 - 4) Algoritma harus memiliki keluaran.
 - 5) Algoritma harus efektif.
5. Materi Algoritma dalam Mata Pelajaran Pemrograman Dasar Kelas X

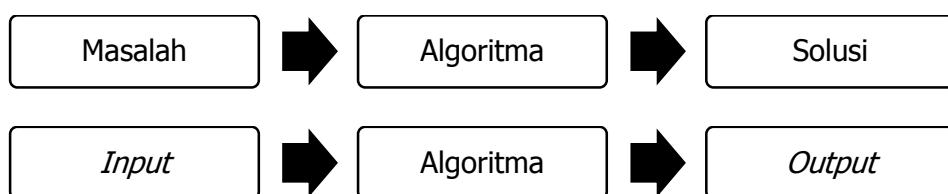
Mata pelajaran pemrograman dasar kelas X adalah mata pelajaran yang ditempuh oleh siswa kelas X dengan jurusan Teknik Komputer dan Informatika pada Sekolah Menengah Kejuruan. Berdasarkan silabus pada mata pelajaran pemrograman dasar kelas X yang berlaku materi algoritma terbagi dalam 3 kompetensi dasar, yaitu sebagai berikut ini.

Tabel 1. Kompetensi Dasar dan Materi Pokok

No	Kompetensi Dasar	Materi
1.	Memahami penggunaan data dalam algoritma dan konsep algoritma pemrograman	Algoritma Pemrograman <ul style="list-style-type: none"> - Konsep algoritma - Struktur algoritma - algoritma menggunakan bahasa natural - Pengenalan Variabel - Pengenalan tipe data - Pengenalan operator - Pseudocode - Flowchart - Penggunaan Tool flowchart
2.	Memahami struktur algoritma serta menganalisis data dalam suatu algoritma percabangan	Algoritma percabangan <ul style="list-style-type: none"> - Percabangan 1 kondisi - Percabangan 2 kondisi - Percabangan lebih dari 2 kondisi - Percabangan bersarang
3.	Memahami struktur algoritma serta menganalisa data dalam suatu algoritma perulangan	Algoritma perulangan <ul style="list-style-type: none"> - Perulangan dengan kondisi diawal - Perulangan dengan kondisi diakhir - Perulangan dengan kondisi akhir diinputkan user - Perulangan sebagai pencacah naik - Perulangan sebagai pencacah turun.

a. Algoritma Pemrograman

Algoritma digunakan dalam pemrograman untuk mempermudah dalam penulisan kode program, sehingga jalannya suatu program bergantung pada algoritma yang dibuat. Konsep dari algoritma adalah pemecahan masalah melalui suatu penjabaran langkah – langkah yang terstruktur, langkah tersebut harus logis dengan tujuan jelas dan dapat ditentukan nilai kebenarannya. Algoritma membutuhkan *input* (masukan) sesuai dengan keperluan permasalahan yang akan diproses menjadi suatu *output* (keluaran) solusi dari permasalahan. Jadi inti dari algoritma pemrograman adalah menemukan solusi dari suatu permasalahan yang ada dalam pemrograman.



Gambar 1. Konsep Dasar Algoritma (Maryono & Pambudhi, 2014: 9)

Struktur algoritma secara umum terdiri dari bagian judul yang memberikan gambaran secara singkat tujuan dari algoritma, bagian deklarasi sebagai tahap persiapan dalam algoritma dimana kebutuhan yang ada dalam algoritma dijabarkan, dan bagian deskripsi yang menjelaskan langkah – langkah atau instruksi yang ada dalam algoritma. Langkah – langkah dalam algoritma harus memiliki kondisi dimana algoritma dimulai dan algoritma diselesaikan.

Salah satu cara mempresentasikan algoritma adalah dengan menggunakan bahasa natural atau bahasa yang digunakan dalam keseharian. Sebagai contoh langkah algoritma dengan bahasa natural yaitu jalan lima langkah ke kanan, masukan air ke dalam panci, atau tambahkan garam secukupnya.

Bentuk lain dalam mempresentasikan algoritma adalah *Pseudocode*. *Pseudocode* adalah bentuk penyajian algoritma dengan menggunakan bahasa yang mirip dengan bahasa pemrograman. Perbedaan *Pseudocode* dengan bahasa pemrograman adalah jika dalam bahasa pemrograman penulisan yang salah sedikit saja dapat menyebabkan *error*, namun dalam *Pseudocode* penulisannya lebih bebas, tidak terikat namun penting, dan mudah dipahami oleh orang yang mengimplementasikan algoritma tersebut ke dalam bahasa pemrograman.

Flowchart atau Diagram Alir adalah penyajian algoritma menggunakan simbol – simbol yang mewakili perintah atau langkah yang dilakukan. Tujuan dari penggunaan *Flowchart* yaitu untuk memberikan gambaran bagaimana data diproses.

Tabel 2. Simbol – simbol yang pada *Flowchart* (Maryono & Pambudhi, 2014: 67)

Simbol	Nama	Fungsi
	<i>Terminator</i>	Permulaan atau akhir algoritma
	<i>Flowline</i>	Arah aliran algoritma
	<i>Preparation</i>	Proses inisialisasi atau pemberian harga awal
	<i>Input/Output</i>	Pemberian masukan atau hasil pengeluaran data
	Proses	Proses pengolahan data
	<i>Decision</i>	Kondisi bersyarat
	<i>On Page Connector</i>	Penghubung bagian yang berada pada satu halaman
	Off Page Connector	Penghubung bagian yang berada pada halaman yang berbeda

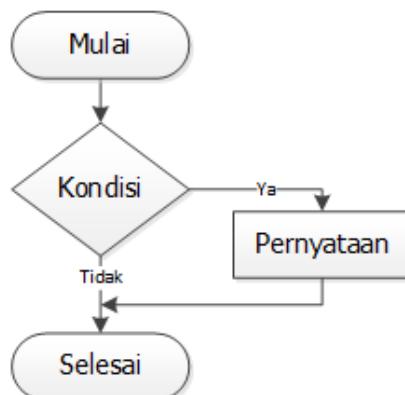
b. Algoritma Percabangan

Algoritma percabangan digunakan untuk membuat suatu pengambilan keputusan dengan suatu kondisi tertentu. Algoritma percabangan ditandai dengan

kata **Jika Maka**. Sedangkan dalam *Flowchart* disimbolkan dengan simbol belah ketupat yang dapat dilihat pada tabel 2. Percabangan ini digunakan untuk operasi – operasi khusus yang memiliki kondisi dimana operasi tersebut akan dilaksanakan jika kondisi operasi dapat terpenuhi. Dibawah ini adalah struktur dari percabangan menggunakan 1 kondisi.

Jika (kondisi) Maka
Pernyataan

Contoh bentuk *Flowchart* dari pernyataan diatas adalah sebagai berikut.



Gambar 2. Struktur Percabangan dalam *Flowchart*

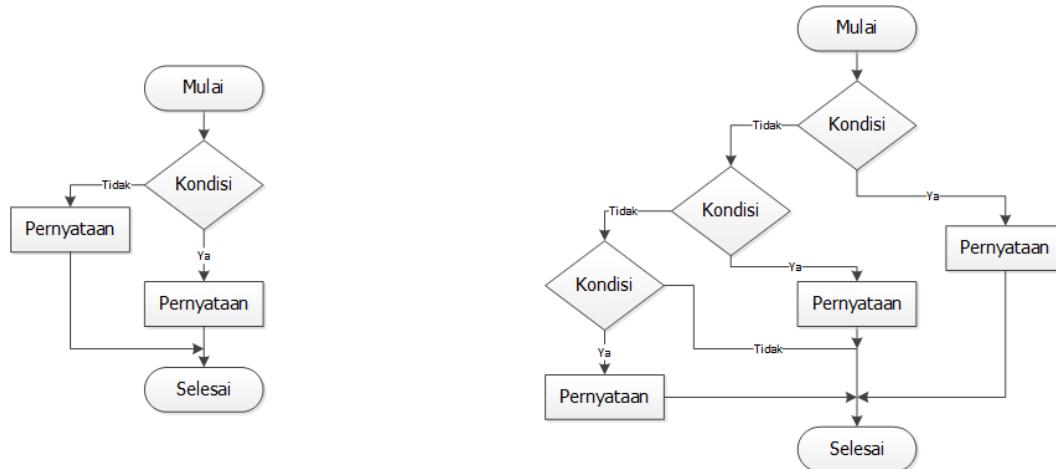
Kondisi yang diberikan tidak terbatas pada satu kondisi saja namun dapat diberikan beberapa kondisi dengan beberapa operasi yang berbeda – beda. Percabangan dengan kondisi lebih dari satu kondisi ditandai dengan kata **Jika Maka** dan **Jika Tidak Maka**. Berikut ini adalah contoh struktur dari percabangan dengan kondisi lebih dari satu.

Jika (kondisi) Maka
Pernyataan
Jika Tidak Maka
Pernyataan

Jika (kondisi) Maka
Pernyataan
Jika (kondisi) Maka
Pernyataan
Jika (kondisi) Maka

Pernyataan Dan seterusnya

Struktur dalam bentuk *Flowchart* dari contoh diatas adalah sebagai berikut.



Gambar 3. Struktur Percabangan *Flowchart* dengan Lebih dari 1 Kondisi

c. Algoritma Perulangan

Perulangan merupakan salah satu struktur penting yang terdapat dalam algoritma. Perulangan memungkinkan algoritma dapat menjalankan suatu rangkaian perintah secara berulang – ulang. Suatu perulangan selalu terdapat titik keluar dari rangkaian perulangan tersebut sehingga tidak terjadi suatu *error* yang biasa disebut dengan *infinite loop*.

Titik keluar atau titik berhenti dari suatu perulangan dapat dilakukan dengan beberapa cara seperti memberikan syarat atau kondisi khusus pada suatu perulangan sehingga ketika kondisi berhenti terpenuhi perulangan akan diselesaikan. Syarat atau kondisi dapat diberikan pada awal perulangan atau pada akhir perulangan.

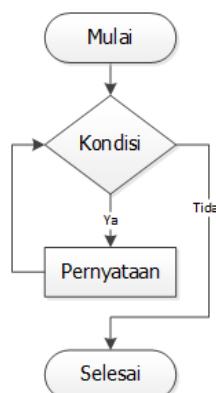
Cara lain dalam menentukan titik keluar dari suatu perulangan adalah menggunakan pencacah. Menggunakan pencacah untuk menentukan titik keluar

perulangan maka berapa kali perulangan akan dilakukan dapat ditentukan. Terdapat pencacah naik dan pencacah turun dimana pencacah naik adalah perhitungannya dari angka kecil menuju ke besar hingga kondisi terpenuhi. Sedangkan pencacah turun adalah perhitungan yang dimulai dari angka besar ke kecil.

Struktur perulangan ditandai dengan kata **Ketika**, **Lakukan**, atau **Ulang Hingga**. Berikut ini adalah contoh dari struktur **Ketika** dengan penentuan kondisi di awal.

Ketika (kondisi)
Pernyataan
Selesai

Struktur diatas dalam bentuk *Flowchart* adalah sebagai berikut ini.



Gambar 4. Struktur Perulangan Kondisi Awal dalam Bentuk *Flowchart*

Ketika kondisi yang ditentukan di akhir perulangan maka digunakan kata **Lakukan** sebagai permulaan dalam perulangan. Berikut ini contoh dari struktur perulangan dengan kondisi diberikan diakhir.

Lakukan
Pernyataan
Ketika (kondisi)

Bentuk dalam *Flowchart* dari struktur diatas adalah sebagai berikut ini.



Gambar 5. Struktur Perulangan Kondisi Akhir dalam Bentuk *Flowchart*

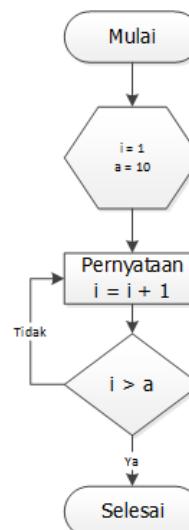
Pencacah dapat menggunakan berbagai macam cara dalam menentukan kondisi untuk titik keluar, salah satunya adalah menggunakan **Ketika** atau **Ulang Hingga**. Berikut ini adalah contoh struktur penggunaan pencacah dalam algoritma perulangan.

```

i = 1
x = 10
Ulang
    Pernyataan
    i = i + 1
Hingga i > x

```

Bentuk *Flowchart* dari struktur diatas yaitu sebagai berikut ini.



Gambar 6. Perulangan menggunakan Pencacah dalam *Flowchart*

6. Sekolah Menengah Kejuruan

Peraturan Pemerintah Nomor 20 Tahun 1990 Pasal 1 menyatakan bahwa Sekolah Menengah Kejuruan adalah pendidikan pada jenjang pendidikan menengah yang mengutamakan pengembangan kemampuan siswa untuk melaksanakan jenis pekerjaan tertentu. Peraturan Pemerintah Nomor 20 Tahun 1990 Pasal 7 menyatakan bahwa Sekolah Menengah Kejuruan menyelenggarakan program – program pendidikan yang disesuaikan dengan jenis – jenis lapangan kerja. Undang – undang Republik Indonesia Nomor 20 Tahun 2003 menyatakan bahwa pendidikan kejuruan merupakan pendidikan menengah yang mempersiapkan peserta didik terutama untuk bekerja pada bidang tertentu. Menurut Ramadhina (2015) menyatakan bahwa Sekolah Menengah Kejuruan dalam pembelajarannya diarahkan untuk menemukan inti materi pembelajaran dengan pendekatan pembelajaran praktis. Sudira (2016: 10) menyatakan bahwa Pendidikan Kejuruan sama dengan Pendidikan Vokasional dimana Sudira (2016: 6) juga menyatakan mengenai Pendidikan Vokasional yang berarti pendidikan untuk dunia kerja dimana pendidikan yang dimaksud berfungsi untuk mengembangkan ke-vokasi-an seseorang. Berdasarkan pernyataan yang telah disebutkan maka dapat disimpulkan bahwa Sekolah Menengah Kejuruan adalah sekolah tingkat menengah dimana dalam proses pembelajarannya menggunakan pendekatan pembelajaran praktis dengan tujuan untuk mengembangkan kemampuan siswa pada jenis pekerjaan tertentu yang dipersiapkan sebagai individu yang siap untuk bekerja pada bidang tertentu.

Prosser dan Quigley (1949) menyatakan mengenai 16 prinsip yang ada pada pendidikan kejuruan. Prinsip – prinsip tersebut adalah sebagai berikut ini.

- 1) Pendidikan kejuruan akan menjadi lebih efisien pada proporsinya dimana siswa belajar pada lingkungan yang menyamai lingkungan kerja.
- 2) Pembelajaran kejuruan yang efektif hanya dapat diberikan dimana pembelajaran dilakukan menggunakan operasi, peralatan, mesin, dan cakupan yang sama dengan yang digunakan pada tempat kerja.
- 3) Pendidikan kejuruan akan menjadi efektif ketika pelatihan individu menggunakan pemikiran dan kebiasaan yang ada pada lingkungan pekerjaan.
- 4) Pendidikan kejuruan akan menjadi efektif ketika pendidikan itu memungkinkan setiap individu untuk mengembangkan minatnya, pengetahuannya, dan keterampilannya pada tingkat tertinggi yang memungkinkan.
- 5) Pendidikan kejuruan yang efektif untuk segala profesi, jabatan, dan pekerjaan hanya bisa diberikan pada orang – orang tertentu yang menginginkannya, membutuhkannya, dan mampu untuk mengambil keuntungan darinya.
- 6) Pelatihan kejuruan akan menjadi efektif ketika pelatihan yang spesifik memberikan pengalaman untuk membangun kebiasaan yang benar dengan melakukan dan berpikir secara berulang – ulang pada titik kebiasaan yang telah ada untuk membiasakan seperti pekerja yang bermanfaat.
- 7) Pendidikan kejuruan akan menjadi lebih efektif ketika instruktur memiliki pengalaman yang sukses pada kemampuan yang diterapkan dan

pengetahuan untuk mengoperasikan dan memproses selama dia melakukan pengajaran.

- 8) Pada setiap pekerjaan terdapat kemampuan produktif minimal dimana seseorang harus mencukupi kebutuhan tersebut untuk menjadi pekerja pada cakupan tersebut.
- 9) Pendidikan kejuruan seharusnya mengenali kondisi pasar dimana mereka melatih seseorang untuk memenuhi kebutuhan pasar sehingga pelatihan menjadi lebih efektif untuk memenuhi kebutuhan pekerja sesuai dengan kondisi yang ada.
- 10) Penyeimbangan yang efektif dari proses kebiasaan dalam diri seorang pebelajar akan memenuhi proporsi kebutuhan sebagaimana mereka mendapatkan pelatihan pada pekerjaan yang nyata, bukan pada latihan atau pekerjaan semu.
- 11) Sumber yang dapat dipercaya untuk mengetahui isi pelatihan pada suatu pekerjaan tertentu adalah dari pengalaman para ahli pada pekerjaan tersebut.
- 12) Setiap pekerjaan memiliki isi yang berbeda – beda satu dengan yang lainnya.
- 13) Pendidikan kejuruan akan membuat pelayanan sosial yang efisien ketika hal itu memenuhi pelatihan yang spesifik pada kebutuhan dari berbagai grup pada waktu yang sama.
- 14) Pendidikan kejuruan akan lebih efisien ketika metode pengajaran yang digunakan berhubungan dengan individu peserta didik berdasarkan pertimbangan sifat – sifat individu tersebut.

- 15) Administrasi dari pendidikan kejuruan akan menjadi efisien ketika hal tersebut lebih fleksibel dan mengikuti arus lingkungan daripada administrasi yang kaku dan terstandar.
- 16) Pendidikan kejuruan seharusnya memerlukan biaya yang masuk akal, jika biaya tersebut tidak terpenuhi maka seharusnya pendidikan kejuruan tidak boleh dipaksakan untuk dilakukan.

B. Kajian Penelitian yang Relevan

1. *Simplifying Algorithm Learning Using Serious Games* (2010)

Penelitian yang dilakukan oleh Sahar Shabana membahas mengenai permainan dengan konsep *serious games* untuk digunakan sebagai media belajar algoritma. Relevansi dengan penelitian ini terletak pada pengembangan media pembelajaran berupa permainan edukasi yang digunakan untuk mengajarkan algoritma. Dalam penelitian tersebut dibahas mengenai permainan yang berbentuk simulasi untuk memberikan pengertian konsep dari beberapa algoritma seperti pencarian dan pengurutan. Perbedaan dengan penelitian ini terletak pada materi yang disampaikan dan konsep dari permainan yang dikembangkan.

2. *Using Puzzles in Teaching Algorithms* (2002)

Penelitian yang dilakukan oleh Anany Levitin dan Mary-Angela Papalaskari meneliti mengenai bagaimana permainan *puzzle* dapat digunakan sebagai media pembelajaran dalam mengajarkan materi algoritma. Relevansi dengan penelitian ini terletak dalam penggunaan *puzzle* sebagai konsep belajar algoritma. Namun terdapat perbedaan dalam penyajian penelitian dimana penelitian tersebut tidak mengembangkan sebuah permainan, namun sebatas pembahasan bagaimana permainan *puzzle* digunakan sebagai media untuk mengajarkan algoritma.

3. *GamEd - Learning Data Structure Algorithm using Computer Game* (2014)

Penelitian yang dilakukan oleh Nazneen Ansari, Noopur Parikh, Anagha Narvekar, Manjiri Phatapekar, dan Anita Yadav membahas mengenai pembelajaran struktur data pada algoritma yang dilakukan dengan *game*. Relevansi dengan penelitian ini adalah bagaimana permainan digunakan sebagai media untuk belajar materi algoritma. Namun konsep materi yang diberikan sedikit berbeda dimana penelitian tersebut hanya membahas tentang struktur data sedangkan dalam penelitian ini materi yang digunakan disesuaikan dengan poin – poin materi yang berlaku sesuai silabus yang ada.

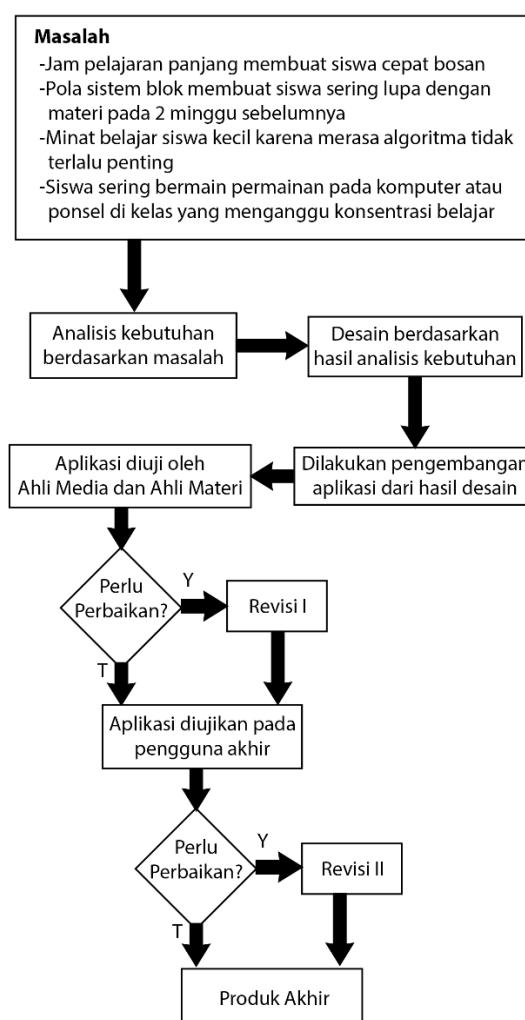
C. Kerangka Pikir

Permainan yang dikembangkan akan digunakan sebagai media penyampaian materi konsep algoritma dengan materi yang disesuaikan pada silabus yang berlaku. Proses pengembangan permainan ini dilakukan melalui beberapa tahap pengembangan yaitu analisis, desain, pengembangan, implementasi, dan pengujian.

Analisis berdasarkan masalah yang dipaparkan yaitu berupa kebutuhan suatu permainan yang dapat digunakan sebagai sarana edukasi atau media untuk belajar konsep – konsep algoritma. Berdasarkan hal tersebut maka dibutuhkan sebuah konsep permainan yang dapat digunakan untuk menggambarkan proses algoritma, digunakanlah konsep permainan *puzzle*. Konsep permainan yang dikembangkan harus sesuai dengan target pengguna. Hasil desain yang telah dilakukan selanjutnya memasuki tahap pengembangan dimana dilakukan pembuatan permainan dengan cara penulisan kode program dengan *software IDE* dan dilanjutkan dengan uji oleh ahli media dan ahli materi, jika terdapat revisi

maka akan didesain kembali berdasarkan masukan dari ahli media dan ahli materi. Selanjutnya diujikan kepada pengguna yaitu siswa dimana masukan – masukan dari pengguna untuk revisi tahap kedua yang nantinya akan menjadi produk akhir yang siap diimplementasikan.

Hasil akhir produk yaitu sebuah permainan *puzzle* yang layak digunakan sebagai sarana belajar algoritma yang disesuaikan dengan materi pada mata pelajaran pemrograman dasar kelas X. Permainan yang telah memenuhi kriteria kelayakan nantinya dapat digunakan untuk sarana belajar konsep dasar algoritma khususnya oleh siswa SMK kelas X.



Gambar 7. Kerangka Pikir Penelitian

D. Pertanyaan Penelitian

1. Bagaimana menganalisa permainan *puzzle* yang mampu digunakan sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
2. Bagaimana mendesain permainan *puzzle* yang mampu digunakan sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
3. Bagaimana mengimplementasikan permainan *puzzle* yang digunakan sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
4. Bagaimana hasil pengujian aspek fungsionalitas dan *compatibility* pada permainan *puzzle* yang dikembangkan sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
5. Bagaimana hasil pengujian aspek *balance*, *compliance*, *playtesting*, *usability*, dan materi pada permainan *puzzle* yang dikembangkan sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang?
6. Apakah hasil pengujian permainan *puzzle* sebagai media pembelajaran algoritma pada mata pelajaran pemrograman dasar kelas X di SMK Negeri 1 Magelang menunjukan bahwa media layak digunakan?

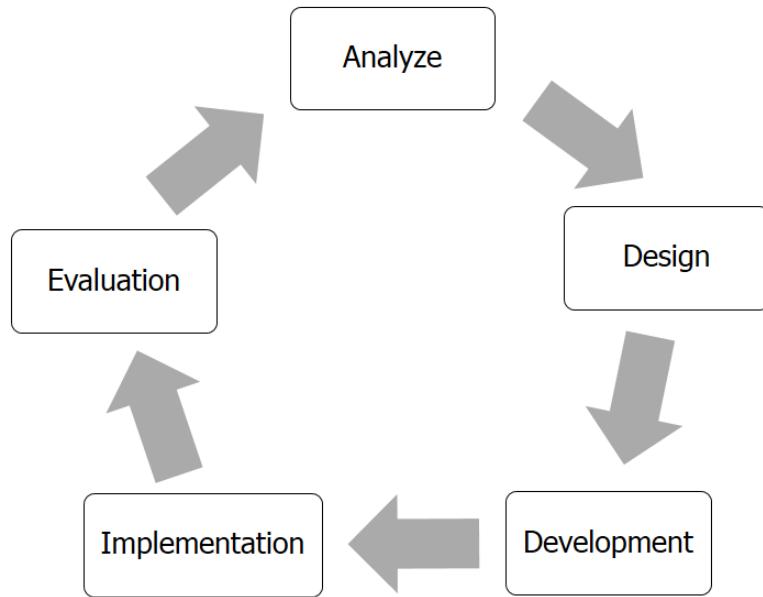
BAB III

METODE PENELITIAN

A. Model Pengembangan

Berdasarkan latar belakang yang telah dipaparkan maka penelitian ini menggunakan metode penelitian *Research and Development (R&D)* atau dalam bahasa Indonesia berarti Penelitian dan Pengembangan. Sugiyono (2017: 297) menyebutkan bahwa metode penelitian *R&D* digunakan untuk menghasilkan produk dan menguji efektivitas dari produk yang dikembangkan tersebut. Putra (2015: 67) menyatakan bahwa *R&D* yaitu metode penelitian yang dilakukan secara sistematis, bertujuan untuk mencaritemukan, merumuskan, memperbaiki, mengembangkan, menghasilkan, menguji keefektifan produk, model, metode, strategi, cara, jasa, prosedur tertentu yang lebih unggul, baru, efektif, efisien, produktif dan bermakna. Metode ini dapat digunakan sebagai metode penelitian dalam berbagai bidang termasuk bidang pendidikan. Pada umumnya metode ini digunakan untuk mengembangkan suatu model baru dalam proses pembelajaran, namun tidak sedikit yang menggunakan metode ini untuk mengembangkan suatu produk media pembelajaran.

Sesuai dengan tujuan penelitian yaitu mengembangkan suatu perangkat lunak yang akan digunakan sebagai media pembelajaran, maka model pengembangan yang digunakan dalam penelitian ini mengacu pada model pengembangan ADDIE. Terdapat 5 tahap pengembangan dalam model tersebut yaitu Analisis, Desain, Pengembangan, Implementasi, dan Evaluasi (*Analyze, Design, Development, Implementation, dan Evaluation*).



Gambar 8. Langkah – langkah Model Pengembangan ADDIE (McGriff, 2000)

B. Prosedur Pengembangan

ADDIE adalah akronim dari 5 langkah dalam proses tahap pengembangan suatu produk yaitu *Analyze, Design, Development, Implementation, and Evaluation* atau dalam bahasa Indonesia yaitu Analisis, Desain, Pengembangan, Implementasi, dan Evaluasi. Proses dalam model tersebut bersifat berkelanjutan atau berulang – ulang dimana setelah tahap terakhir memungkinkan kembali pada proses tahap pertama ketika hal tersebut dibutuhkan.

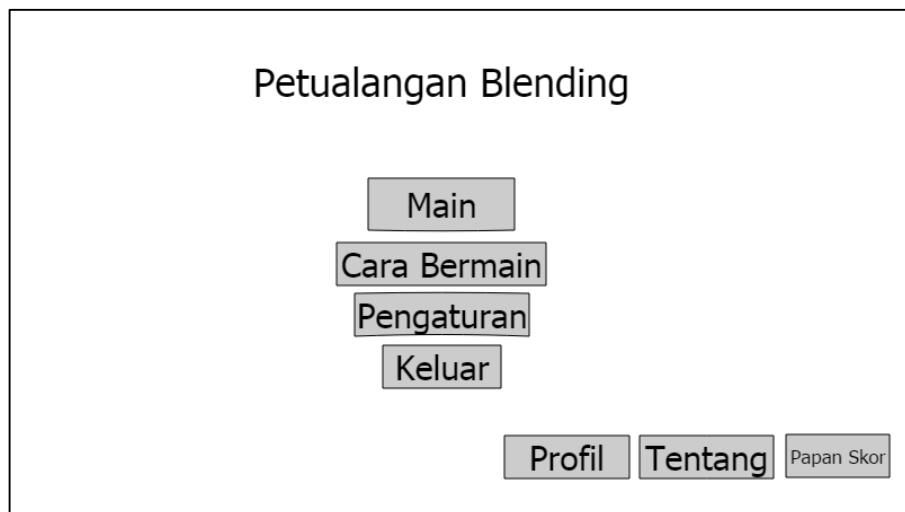
1. Analisis

Tahap yang menjadi awalan dalam penelitian ini adalah tahap analisis dimana tahap ini menjadi fondasi untuk tahap – tahap selanjutnya. Tahap analisis menurut McGriff (2000) adalah suatu tahap dimana masalah dijabarkan dan diidentifikasi lalu dilakukan pemilihan solusi yang memungkinkan. Analisis yang dilakukan dalam penjabaran masalah yaitu observasi atau pengamatan dimana hal tersebut dilakukan guna mencari tahu informasi mengenai masalah dan

kemungkinan penyelesaian masalah yang dapat dilakukan. Setelah kemungkinan penyelesaian masalah ditemukan maka dilakukan analisis kebutuhan apa saja yang menjadi kebutuhan untuk pemecahan masalah tersebut. Dalam hal ini yaitu suatu media aplikasi permainan yang dapat digunakan untuk belajar, maka dari itu dilakukan analisis kebutuhan untuk sistem yang dikembangkan agar media permainan sesuai dengan kebutuhan pengguna atau subjek penelitian sehingga media yang dikembangkan layak digunakan oleh subjek penelitian.

2. Desain

Setelah analisis kebutuhan dilakukan langkah selanjutnya yaitu tahap desain. Tahap desain yang dilakukan yaitu melakukan desain aplikasi yang dikembangkan termasuk desain sistem, struktur data, dan desain antarmuka pengguna. *Storyboard* dibuat untuk memberi gambaran bagaimana aplikasi berinteraksi dengan pengguna, mulai dari awal aplikasi dibuka hingga akhir. Setelah *storyboard* dibuat maka selanjutnya membuat desain sistem dengan *flowchart*. Setiap simbol *flowchart* akan mewakili proses dan fungsi yang berbeda. Selain desain *flowchart* yang dilakukan dalam tahap desain adalah desain antarmuka pengguna yaitu suatu tampilan yang menghubungkan antara sistem dengan pengguna sehingga sistem dapat mengenali perintah yang diberikan oleh pengguna. Rancangan desain antarmuka berupa *mockup*, hal tersebut akan mempermudah pembuatan antarmuka aplikasi yang dikembangkan. *Mockup* menjadi dasar dari pembuatan tampilan permainan yang akan dikembangkan. Hal terakhir pada tahap desain adalah melakukan desain level permainan. Desain level permainan dilakukan untuk membuat tingkat kesulitan untuk setiap level seimbang.

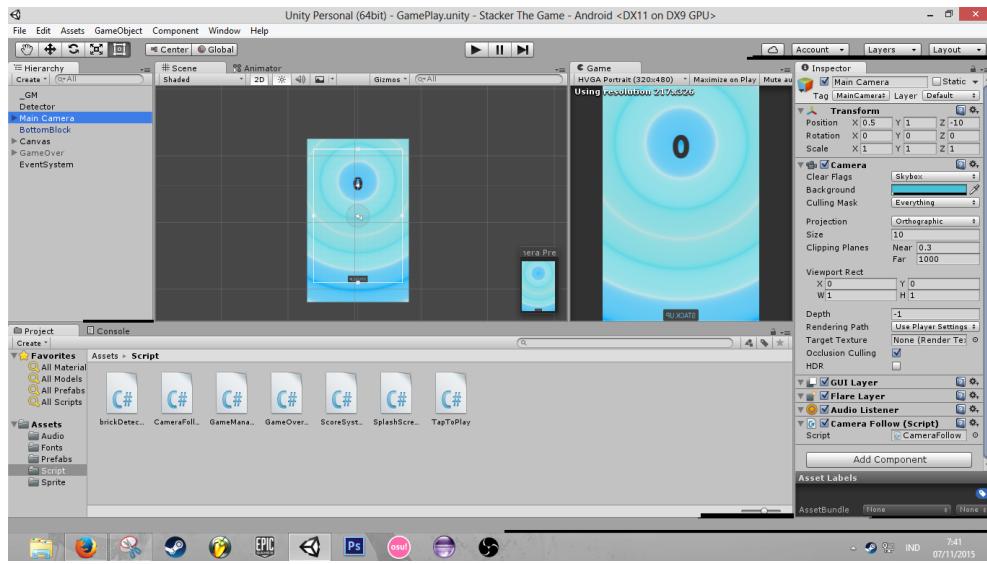


Gambar 9. Contoh desain *mockup* yang dibuat pada *software* Pencil

3. Pengembangan

Tahap pengembangan merupakan tahap dimana aplikasi mulai terbentuk dari desain yang telah dibuat pada tahap sebelumnya. Proses pengembangan terdiri dari pengetikan kode program pada perangkat lunak *IDE (Integrated Development Environment)* dimana nantinya permainan akan dibuat. Perangkat lunak *IDE* yang digunakan adalah *Visual Studio* yang diintegrasikan dengan *game engine Unity3D* dan menggunakan bahasa pemrograman *C# (C Sharp)*. Setelah dilakukan pengetikan kode program maka permainan dapat dijalankan dan diuji fungsionalitasnya sebelum dilakukan pengujian terhadap pengguna. Pengujian tersebut dilakukan pada lingkungan pengembangan. Pengujian ini bertujuan untuk mengetahui kesiapan aplikasi sebelum dilanjutkan pada tahap selanjutnya yaitu implementasi di luar lingkungan pengembang. Aspek yang dinilai dalam uji fungsionalitas ini adalah kesiapan fungsi apakah sesuai dengan hasil yang diharapkan. Pressman dan Maxim (2015: 509) menyatakan uji fungsionalitas menggunakan metode *black-box* yaitu uji yang menentukan perilaku suatu sistem

atau bagian dari suatu sistem. Sistem atau bagian dari suatu sistem akan memroses hasil stimulus menjadi suatu respon.



Gambar 10. Contoh proses pengembangan permainan pada *software* Unity

4. Implementasi

Tahap implementasi merupakan tahap dimana media permainan diterapkan pada lingkungan pengujian. Tahap ini bertujuan untuk menguji mengenai produk yang telah dikembangkan. Pada tahap ini pengguna mencoba menggunakan media yang telah dikembangkan dan melakukan penilaian mandiri terhadap produk yang telah dikembangkan melalui kuesioner yang akan digunakan sebagai bahan evaluasi terhadap produk yang telah dikembangkan. Proses implementasi dapat memakan waktu yang cukup lama berdasarkan bagaimana respon pengguna terhadap produk yang diuji cobakan. Setelah produk diimplementasikan, maka produk akan dilanjutkan untuk memasuki pada tahap selanjutnya yaitu evaluasi.



Gambar 11. Siswa mencoba permainan yang telah dikembangkan

5. Evaluasi

Tahap yang terakhir dari proses model pengembangan ini adalah evaluasi yaitu tahap dimana hasil dari implementasi yang telah dilaksanakan dianalisa guna menentukan tingkat kualitas dan kelayakan dari produk yang dikembangkan. Setelah dievaluasi masih terdapat kemungkinan terdapat kekurangan dan kendala yang terdapat pada produk sehingga proses kembali pada tahap awal untuk dilakukan pembenahan. Teknik yang digunakan untuk mengetahui tingkat kelayakan media permainan yang dikembangkan adalah dengan pengujian oleh ahli media, ahli materi, dan pengguna agar kualitas dari media yang dikembangkan layak digunakan. Uji kualitas media permainan menggunakan *testing disciplines* yang dinyatakan oleh Levy dan Novak (2010: 58-69) yaitu terdapat beberapa pengujian yang dilakukan antara lain adalah *balance testing*, *compatibility testing*, *compliance testing*, *localization testing*, *playtesting*, dan *usability testing*.

a. *Balance testing*

Permainan yang baik adalah permainan yang seimbang yaitu permainan yang konsisten, adil, dan menyenangkan. *Balance testing* atau uji keseimbangan dilakukan agar permainan yang dikembangkan dapat dimainkan secara adil, memiliki isi yang konsisten, dan menyediakan permainan yang menyenangkan. Menurut Rollings dan Adams (2003: 272) terdapat 5 indikator dalam pengujian keseimbangan yaitu *consistent challenge, perceivably fair playing experiences, avoid stagnation, avoid trivialities, dan level difficulties.*

b. *Compatibility testing*

Compatibility testing dilakukan dengan cara menjalankan produk pada sistem dan perangkat keras yang dikonfigurasi berbeda – beda guna menjamin aplikasi yang dikembangkan cocok dengan sistem dan perangkat keras yang ada di pasaran. Pengujian dilakukan pada berbagai perangkat yang berbeda sistem, berbeda perangkat keras, dan berbeda ukuran kapasitas. Levy dan Novak (2010: 60) menyatakan bahwa dalam *compatibility testing* terdapat 4 indikator yang digunakan yaitu *display performance, operating system performance, audio performance, dan user input performance.*

c. *Compliance testing*

Compliance testing merupakan proses pengujian pada produk perangkat lunak yang dikembangkan sesuai dengan standar yang ditetapkan oleh sistem operasi yang menjalankan perangkat lunak tersebut. Pada penelitian ini produk yang dihasilkan menggunakan sistem operasi *Windows* sehingga menggunakan standar kriteria yang telah dijabarkan oleh *Microsoft* pada *website* yang tersedia

dengan alamat [https://msdn.microsoft.com/en-us/library/windows/desktop/mt674710\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/mt674710(v=vs.85).aspx).

d. *Localization testing*

Pengujian ini dilakukan ketika aplikasi yang dikembangkan menggunakan lebih dari satu bahasa. Tujuan dari *localization testing* adalah agar bahasa yang digunakan tidak salah arti ketika menggunakan bahasa yang berbeda. Dalam penelitian ini tidak dilakukan *localization testing* karena hanya menggunakan satu bahasa pokok.

e. *Playtesting*

Playtesting yaitu pengujian yang dilakukan dengan menguji coba aplikasi permainan yang dikembangkan pada pengguna dengan tujuan mengetahui bagaimana permainan berdampak pada pengguna. Kuindersma, Pal, Herik, dan Plaat (2017) menyebutkan terdapat 4 indikator yang dinilai pada *playtesting* yaitu *controls, enjoyment, engagement, dan parallels with reality*.

f. *Usability testing*

Usability testing yaitu pengujian yang dilakukan untuk menilai mengenai bagaimana pengguna menjalankan sistem perangkat lunak yang dikembangkan. Tujuan dari pengujian ini yaitu mengetahui kelemahan dan kelebihan perangkat lunak dalam hal penggunaan oleh pengguna. Bagi produk berupa permainan pengujian ini sangat penting karena berhubungan dengan kendali pengguna dalam memainkan permainan. Jerzak dan Rebelo (2014) menyatakan 4 indikator dalam *usability testing* pada permainan yaitu *documentation and tutorial, feedback, screen layout, dan error prevention*.

C. Subjek Penelitian

Penelitian ini yaitu mengembangkan suatu produk dimana produk tersebut akan diuji, maka subjek pada penelitian ini adalah sumber data yang mampu dijadikan penguji produk yang dikembangkan yaitu pengembang, ahli media, ahli materi, dan pengguna. Pengembang yaitu orang yang ahli pada bidang pengembangan aplikasi permainan, ahli media yaitu dosen Universitas Negeri Yogyakarta yang kompeten di bidangnya, ahli materi yaitu guru SMK Negeri 1 Magelang yang mengampu mata pelajaran yang diteliti, dan pengguna adalah siswa kelas X jurusan Teknik Komputer dan Informatika SMK Negeri 1 Magelang yang menjadi pengguna akhir produk yang dikembangkan.

D. Metode Pengumpul Data

Metode yang digunakan dalam mengumpulkan data pada penelitian ini yaitu sebagai berikut ini.

1. Observasi

Observasi adalah teknik pengumpulan data yang dilakukan dengan cara mengamati objek – objek penelitian secara langsung (Sugiyono, 2017: 145). Kegiatan dapat dilakukan dengan cara ikut berperan serta atau berpartisipasi dalam objek yang diteliti atau hanya menjadi pengamat yang mengamati tanpa berpartisipasi dalam objek. Dalam penelitian ini observasi yang dilakukan adalah peneliti menjadi bagian dari objek yang diteliti sehingga memperoleh data yang lebih lengkap berkaitan dengan kegiatan yang ada pada tempat penelitian yaitu di sekolahannya sehubungan dengan kegiatan belajar mengajar yang ada.

2. Kuesioner

Kuesioner adalah teknik pengumpulan data yang dilakukan dengan cara memberikan pertanyaan – pertanyaan melalui angket yang diberikan kepada responden untuk dijawab (Sugiyono, 2017: 142). Pada penelitian ini jenis pertanyaan yang digunakan yaitu jenis pertanyaan tertutup dengan pertanyaan – pertanyaan berupa *checklist* yang terperinci sehingga memudahkan peneliti dalam melakukan analisis data terhadap seluruh angket yang terkumpul. Pengumpulan data menggunakan angket ini dilakukan pada tahap pengujian produk guna mengetahui kelayakan dari produk yang dikembangkan pada pengguna berdasarkan hasil angket.

E. Instrumen Penelitian

Instrumen penelitian adalah suatu alat yang digunakan untuk mengukur fenomena alam maupun sosial yang diamati (Sugiyono, 2017: 102). Instrumen yang digunakan dalam penelitian ini berupa untuk mengukur kelayakan produk. Terdapat 4 tahap pengujian yaitu uji fungsionalitas dan compatibility, uji oleh ahli media, uji oleh ahli materi, dan uji oleh pengguna.

1. Pengembang

Pengujian ini dilakukan pada lingkungan pengembang guna mengetahui kesiapan produk sebelum dilakukan pengujian di luar lingkungan pengembang. Pada uji oleh pengembang terdapat 2 aspek yang akan dinilai yaitu *compatibility testing* dan *functionality testing*.

a. Instrumen *Compatibility Testing*

Instrumen pada *compatibility testing* ini menggunakan indikator yang digagas oleh Levy dan Novak (2010: 60). Pengujian dilakukan untuk mengetahui

apakah produk dapat dijalankan pada sistem, perangkat, dan sumber daya yang berbeda. Pengujian ini dilakukan oleh pengembang untuk memastikan bahwa produk yang dikembangkan siap dijalankan pada pengguna yang memiliki sistem, perangkat, dan sumber daya yang berbeda - beda. Dalam pengujian ini terdapat 4 indikator yang disajikan dalam kuesioner yaitu *display performance*, *operating system performance*, *audio performance*, dan *user input performance*.

Tabel 3. Kisi – kisi Instrumen *Compatibility Testing*

No	Indikator	Penjelasan	No. Butir
1	<i>Display performance</i>	Media mampu dijalankan pada berbagai ukuran layar	1, 2, 3
2	<i>Operating system performance</i>	Media dapat dijalankan pada sistem operasi yang ditargetkan	4, 5, 6
3	<i>Audio performance</i>	Pada perangkat <i>audio</i> yang berbeda suara masih terdengar jelas	7, 8, 9
4	<i>User input performance</i>	Kendali dengan perangkat keras yang berbeda	10, 11, 12

b. Instrumen *Functionality Testing*

Instrumen ini digunakan untuk mengetahui apakah komponen pada aplikasi sudah berjalan dengan semestinya atau tidak. Indikator yang digunakan pada *functionality testing* berdasarkan dari isi aplikasi yang dikembangkan dimana terdapat 3 halaman bagi pengguna yaitu halaman menu utama, halaman pemilihan level, dan halaman permainan.

Tabel 4. Kisi – kisi Instrumen *Functionality Testing*

No	Indikator	Penjelasan	No. Butir
1	Halaman menu utama	Semua bagian pada halaman menu utama berjalan dengan baik	1 - 22
2	Halaman pemilihan level	Semua bagian pada halaman pemilihan level berjalan dengan baik	23 - 28
3	Halaman permainan	Semua bagian pada halaman permainan berjalan dengan baik	29 - 42

2. Ahli Media

Pengujian oleh ahli media dilakukan untuk menguji kualitas produk dari segi media dan sistem. Instrumen diujikan kepada 2 ahli media menggunakan kuesioner berdasarkan dari aspek *balance testing*, *compliance testing*, *playtesting*, dan *usability testing*.

a. Instrumen *Balance Testing*

Instrumen *balance testing* menggunakan indikator – indikator yang dikemukakan oleh Rollings dan Adams (2003: 272) yaitu 5 indikator dalam pengujian keseimbangan atau *balance testing*.

Tabel 5. Kisi – kisi Instrumen *Balance Testing*

No	Indikator	Penjelasan	No. Butir
1	<i>Consistent challenges</i>	Tantangan yang diberikan selalu konsisten meningkat	2, 10
2	<i>Perceivably fair play experiences</i>	Memberikan pengalaman permainan yang adil	4, 6
3	<i>Avoid stagnation</i>	Selalu terdapat jalan keluar dalam permainan	1, 5, 8

4	<i>Avoid trivialities</i>	Keputusan yang diambil pemain merupakan pilihan yang penting	7, 11
5	<i>Level difficulties</i>	Tingkat kesulitan yang diberikan wajar	3, 9

b. Instrumen *Compliance Testing*

Instrumen *compliance testing* diambil dari standar yang ditentukan oleh sistem operasi *Windows*. Terdapat 7 indikator yang telah ditentukan oleh *Microsoft* untuk *compliance testing* ini pada *website*.

Tabel 6. Kisi – kisi Instrumen *Compliance Testing*

No	Indikator	Penjelasan	No. Butir
1	<i>It must be a standalone app</i>	Aplikasi mampu berjalan secara mandiri pada sistem	1
2	<i>It must run on a local Windows 8.1 computer</i>	Aplikasi dapat berjalan pada sistem operasi <i>Windows 8.1</i>	2
3	<i>It can be a client component of a certified Windows Server app</i>	Aplikasi dapat menjadi <i>client component</i> pada suatu aplikasi <i>Windows Server</i>	3
4	<i>It must have feature complete</i>	Aplikasi memiliki fitur yang lengkap	4
5	<i>It must not communicate with Windows Store apps via local mechanisms</i>	Aplikasi tidak berkomunikasi dengan <i>Windows Store apps</i> melalui mekanisme lokal	5
6	<i>It must not jeopardize or compromise the security or functionality of the Windows system</i>	Aplikasi tidak membahayakan keamanan dan kinerja dari sistem <i>Windows</i>	6
7	<i>It must have a unique name</i>	Aplikasi memiliki nama yang unik	7

c. Instrumen *Playtesting*

Indikator yang digunakan dalam *playtesting* diambil dari pernyataan Kuindersma dkk (2017) yaitu sebagai berikut ini.

Tabel 7. Kisi – kisi Instrumen *Playtesting*

No	Indikator	Penjelasan	No. Butir
1	<i>Controls</i>	Kendali permainan mudah digunakan	1, 6
2	<i>Enjoyment</i>	Permainan terasa menyenangkan	4, 8
3	<i>Engagement</i>	Permainan terasa menarik	3, 5
4	<i>Parallels with reality</i>	Kesejajaran isi permainan dengan kenyataan	2, 7

d. Instrumen *Usability Testing*

Instrumen *usability testing* menggunakan kuesioner yang ditujukan untuk menilai media permainan dari segi kegunaannya. Indikator yang digunakan dalam instrumen *usability testing* menggunakan indikator yang dinyatakan oleh Jerzak dan Rebelo (2014).

Tabel 8. Kisi – kisi Instrumen *Usability Testing*

No	Indikator	Penjelasan	No. Butir
1	<i>Documentation and tutorial</i>	Bantuan dalam permainan	4, 6
2	<i>Feedback</i>	Umpulan balik pada pemain	2, 9
3	<i>Screen layout</i>	Tampilan atau <i>interface</i> pada permainan	3, 5, 7
4	<i>Error prevention</i>	Pencegahan <i>error</i> dalam permainan	1, 8

3. Ahli Materi

Instrumen untuk ahli materi digunakan untuk menilai media dari aspek isi materi dan desain pembelajaran. Pengujian ini dilakukan oleh 2 ahli materi dengan menggunakan kuesioner dengan indikator – indikator berdasarkan materi yang berlaku pada mata pelajaran pemrograman dasar dengan 3 bab materi yaitu konsep algoritma, algoritma percabangan, dan algoritma perulangan.

Tabel 9. Kisi – kisi Instrumen Ahli Materi

No	Indikator	No. Butir
1	Konsep dasar algoritma	1, 2, 6, 11, 14
2	Algoritma percabangan	3, 8, 10, 13
3	Algoritma perulangan	4, 5, 7, 9, 12

4. Pengguna

Instrumen untuk pengguna digunakan untuk menilai media sesuai dengan *testing disciplines* yaitu menilai dari segi keseimbangan permainan, dampak permainan pada siswa, dan kegunaan media. Pengguna dalam hal ini adalah peserta didik kelas X SMK Negeri 1 Magelang yang tengah menempuh mata pelajaran Pemrograman Dasar. Terdapat 4 aspek pengujian dalam instrumen ini sesuai dengan *testing disciplines* yaitu *balance testing*, *compatibility testing*, *playtesting*, dan *usability testing*. Hasil dari pengujian ini nantinya menjadi masukan akhir setelah pengujian oleh ahli media dan menjadi produk yang siap disebarluaskan. Kisi – kisi untuk pengujian oleh pengguna dapat dilihat pada tabel dibawah ini.

Tabel 10. Kisi – kisi Instrumen Pengguna

No	Aspek	Indikator	Penjelasan	No. Butir
1	<i>Balance testing</i>	<i>Consistent challenges</i>	Tantangan setiap <i>level</i> konsisten meningkat	2, 22
		<i>Perceivably fair play experiences</i>	Memberikan pengalaman permainan yang adil	25, 34
		<i>Avoid stagnation</i>	Selalu terdapat jalan keluar dalam permainan	6, 37
		<i>Avoid trivialities</i>	Keputusan yang diambil oleh pemain adalah keputusan penting	26, 31
		<i>Level difficulties</i>	Tingkat kesulitan yang diberikan wajar	4, 10
2	<i>Playtesting</i>	<i>Controls</i>	Kendali permainan mudah dipahami	3, 12, 18
		<i>Enjoyment</i>	Permainan terasa menyenangkan	13, 16, 24
		<i>Engagement</i>	Permainan menarik untuk diselesaikan	8, 14, 38
		<i>Parallels with reality</i>	Kesejajaran isi permainan dengan kenyataan	20, 23
3	<i>Usability testing</i>	<i>Documentation and tutorial</i>	Bantuan dalam permainan	19, 29, 33
		<i>Feedback</i>	Umpatan balik pada pemain	9, 21
		<i>Screen layout</i>	Tampilan atau <i>interface</i> pada permainan	1, 15, 35
		<i>Error prevention</i>	Pencegahan <i>error</i> dalam permainan	32, 39
4	Materi	Konsep dasar algoritma	Media berisi materi konsep dasar algoritma	7, 30, 36
		Algoritma percabangan	Media berisi materi algoritma percabangan	11, 17
		Algoritma perulangan	Media berisi materi algoritma perulangan	5, 27, 28

F. Teknik Analisis Data

Teknik analisis data yang digunakan dalam penelitian ini yaitu analisis data statistik deskriptif. Statistik deskriptif adalah statistik yang digunakan untuk menganalisis data dengan cara mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa bermaksud membuat kesimpulan yang berlaku untuk umum atau generalisasi (Sugiyono, 2017: 147). Penyajian data pada statistik deskriptif dapat dilakukan melalui penyajian tabel, grafik, diagram, skala, perhitungan rata – rata, standar deviasi, dan perhitungan persentase.

Skala yang digunakan dalam data angket yang dibagikan menggunakan skala *Likert* dan skala *Guttman*. Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang gejala atau kejadian sosial (Sugiyono, 2017: 93). Interval pada skala *Likert* dibagi menjadi 2 yaitu untuk pertanyaan positif dan pertanyaan negatif. Gradasi dari skala *Likert* dimulai dari sangat positif hingga sangat negatif. Konversi data kuantitatif pada skala *Likert* dapat dilihat pada tabel dibawah ini.

Tabel 11. Interval Skala *Likert*

Pertanyaan Positif		Pertanyaan Negatif	
Sangat Setuju (SS)	4	Sangat Setuju (SS)	1
Setuju (S)	3	Setuju (S)	2
Tidak Setuju (TS)	2	Tidak Setuju (TS)	3
Sangat Tidak Setuju (STS)	1	Sangat Tidak Setuju (STS)	4

Skala *Guttman* digunakan untuk mengetahui jawaban secara tegas terhadap suatu permasalahan yang ditanyakan seperti “ya-tidak”, “benar-salah”, “setuju-tidak setuju”, dan “positif-negatif” (Sugiyono, 2017: 96).

Tabel 12. Interval Skala *Guttman*

Pertanyaan Positif		Pertanyaan Negatif	
Ya	1	Ya	0
Tidak	0	Tidak	1

Teknik pengategorisasian skor dalam skala yang digunakan menggunakan rumus sebagai berikut ini.

$$Range = (\text{Skor Maksimal} - \text{Skor Minimal})$$

$$\bar{X} = \frac{(\text{Skor Maksimal} + \text{Skor Minimal})}{2}$$

$$\sigma = \frac{\text{Range}}{6}$$

Tabel 13. Kategorisasi Skala

Rentang Skor	Kategori
$\bar{X} - 3\sigma \leq X < \bar{X} - 1.8\sigma$	Sangat Kurang Layak
$\bar{X} - 1.8\sigma \leq X < \bar{X} - 0.6\sigma$	Kurang Layak
$\bar{X} - 0.6\sigma \leq X < \bar{X} + 0.6\sigma$	Cukup Layak
$\bar{X} + 0.6\sigma \leq X < \bar{X} + 1.8\sigma$	Layak
$\bar{X} + 1.8\sigma \leq X \leq \bar{X} + 3\sigma$	Sangat Layak

Keterangan :

- X = Skor Empirik
- \bar{X} = Rerata Hipotetik
- σ = Standar Deviasi
- Skor Maksimal = Jumlah butir instrumen x Nilai skala tertinggi
- Skor Minimal = Jumlah butir instrumen x Nilai skala terendah

Setelah hasil perhitungan data dikategorisasi selanjutnya adalah menghitung konsistensi data atau reabilitas data terhadap instrumen. Perhitungan realibilitas menggunakan *Cronbach's Alpha* yang dihitung menggunakan perangkat lunak Excel. Berdasarkan hasil perhitungan tadi maka ditentukan konsistensi data menggunakan aturan praktis yang dinyatakan oleh Gliem dan Gliem (2003) sebagai berikut.

Tabel 14. Aturan praktis *Cronbach's Alpha*

Cronbach's Alpha	Internal Consistency
$\alpha \geq .9$	<i>Excellent</i>
$.9 > \alpha \geq .8$	<i>Good</i>
$.8 > \alpha \geq .7$	<i>Acceptable</i>
$.7 > \alpha \geq .6$	<i>Questionable</i>
$.5 > \alpha$	<i>Unacceptable</i>

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

1. Tahap Analisis

Tahap pertama dalam suatu pengembangan adalah analisis. Tahap ini menjadi dasar dari proses selanjutnya untuk mendapatkan hasil pengembangan yang sesuai. Pada tahap ini dilakukan analisis kebutuhan pengguna, analisis perangkat keras yang akan digunakan, dan analisis perangkat lunak yang akan digunakan.

a. Analisis kebutuhan pengguna

Proses melakukan analisis kebutuhan pengguna berdasarkan observasi di kelas dimana ditemukan beberapa permasalahan yaitu :

- 1) Jam pelajaran yang panjang akibat penerapan sistem penjadwalan blok membuat siswa mudah bosan ketika di kelas
- 2) Penerapan sistem blok juga berdampak pada proses penyampaian materi dimana terdapat siswa yang meminta untuk mengulangi materi pada pertemuan sebelumnya atau pertemuan pada 2 minggu sebelumnya
- 3) Minat belajar siswa pada materi algoritma kecil karena mereka merasa algoritma adalah materi yang tidak terlalu penting
- 4) Siswa sering bermain permainan pada saat jam pelajaran berlangsung baik pada perangkat komputer ataupun telepon genggam, hal tersebut mengganggu konsentrasi belajar siswa

Berdasarkan permasalahan yang dipaparkan maka dilakukan analisis untuk mendapatkan solusi terkait masalah yang dihadapi. Solusi terkait permasalahan yang telah dipaparkan yaitu suatu media permainan dimana hal tersebut digunakan untuk mengarahkan perilaku siswa yang sering bermain permainan untuk hal yang lebih berguna yaitu belajar. Tujuan dari penggunaan media permainan yang lain adalah untuk membuat belajar mengenai materi konsep algoritma menjadi lebih menyenangkan.

Permainan yang akan dikembangkan merupakan suatu media sebagai sarana menyampaikan materi algoritma, maka dilakukan analisis mengenai materi – materi yang menjadi kebutuhan pengguna. Buku mata pelajaran dan silabus yang berlaku menjadi acuan dalam pengambilan materi yang akan diterapkan pada media permainan. Hasil analisis materi didapatkan 3 kompetensi dasar mengenai materi algoritma yaitu algoritma pemrograman yang membahas mengenai konsep – konsep dasar pada algoritma, algoritma percabangan yang membahas mengenai konsep pengambilan keputusan pada algoritma, dan yang terakhir adalah algoritma perulangan yang membahas mengenai suatu konsep algoritma yang dapat mengulangi satu atau lebih perintah selama kondisi terpenuhi.

Setelah analisis materi dilakukan, langkah selanjutnya adalah menentukan konsep permainan. Mengacu pada jurnal milik Levitin dan Papalaskari (2002) yang membahas mengenai penggunaan *puzzle* untuk mengajarkan algoritma, maka konsep permainan yang akan dikembangkan menggunakan konsep *puzzle*. Penggunaan konsep *puzzle* sesuai dengan materi yang akan disampaikan yaitu mengenai algoritma dimana terdapat kesamaan antara konsep *puzzle* dan algoritma. Kesamaan tersebut adalah keduanya memerlukan pemikiran

logis dan sistematis, sehingga konsep *puzzle* sangat cocok digunakan untuk merepresentasikan konsep dari algoritma.

b. Analisis perangkat keras

Pengembangan suatu aplikasi untuk komputer memerlukan perangkat keras untuk melakukan penulisan kode program. Perangkat keras yang digunakan yaitu komputer. Analisis perangkat keras dilakukan untuk menentukan spesifikasi komputer yang akan digunakan untuk pengembangan. Penentuan spesifikasi komputer yang akan digunakan untuk pengembangan mempertimbangkan kelancaran dalam proses *coding* dan *building* aplikasi. Berdasarkan perangkat lunak yang akan digunakan untuk pengembangan, maka spesifikasi komputer yang akan digunakan yaitu :

- 1) OS : Windows 7, 8, atau 10, versi 64 bit
- 2) RAM : 4 GB atau lebih
- 3) CPU : setara Intel i3 atau lebih
- 4) GPU : *support* DX10

c. Analisis perangkat lunak

Proses pengembangan aplikasi dilakukan pada komputer, maka dibutuhkan suatu perangkat lunak sebagai alat untuk membuat, mengubah, dan mengembangkan suatu permainan pada komputer. Perangkat lunak yang akan digunakan yaitu :

- 1) Unity 2017, sebuah *game engine* yang digunakan untuk membuat aplikasi permainan dengan berbagai dukungan yang memudahkan pengembang dalam pengembangan permainan.

- 2) Adobe Photoshop CC 2017, sebuah perangkat lunak pengolah gambar yang digunakan untuk menggambar karakter permainan, menggambar tampilan permainan, menggambar objek permainan.
- 3) Fruity Loop 12, sebuah perangkat lunak yang digunakan untuk membuat dan mengubah *audio* yang akan digunakan dalam permainan. *Audio* dapat berupa efek suara dan juga musik.
- 4) Pencil, sebuah perangkat lunak pengolah gambar yang khusus digunakan untuk membuat *mockup* tampilan pengguna permainan. Pencil memberikan kemudahan kepada pengembang dalam membuat *mockup* karena terdapat berbagai fitur yang mempercepat proses pembuatan *mockup*.
- 5) Visual Studio 2017, sebuah perangkat lunak yang digunakan untuk menulis kode program permainan. Perangkat lunak ini dapat diintegrasikan dengan Unity 2017 sehingga semakin memudahkan dalam pengembangan permainan.
- 6) Microsoft Visio 2016, sebuah perangkat lunak pengolah gambar untuk membuat *flowchart* yang digunakan sebagai alur sistematis data pada permainan.
- 7) Adobe Illustrator CS6, sebuah perangkat lunak pengolah gambar yang digunakan untuk membuat dan mengedit gambar berupa gambar vektor. Perangkat lunak ini digunakan untuk membuat *storyboard* permainan.

2. Tahap Desain

Tahap desain merupakan tahap dimana aplikasi yang akan dikembangkan dirancang. Tahap desain dimulai dengan pembuatan *storyboard* tentang bagaimana pengguna akan berinteraksi dengan aplikasi. *Storyboard* sederhana menggambarkan tentang alur pengguna berinteraksi dengan permainan mulai dari awal hingga akhir.

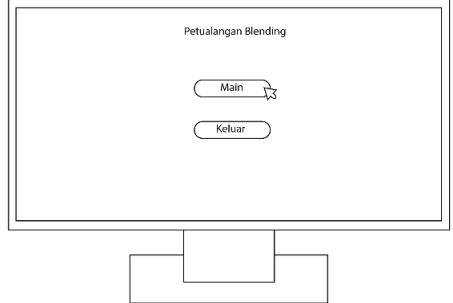
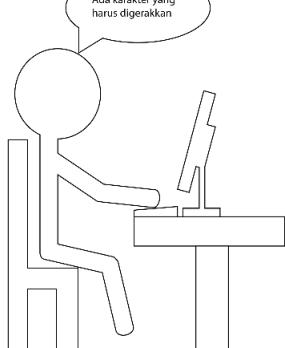
Merujuk pada *storyboard*, selanjutnya dilakukan pembuatan *flowchart* untuk sistem yang akan dikembangkan. Setiap tindakan pada *storyboard* dilakukan penggambaran alur data menggunakan *flowchart* sehingga memudahkan proses penulisan kode program pada saat tahap pengembangan.

Langkah selanjutnya dalam tahap desain adalah pembuatan *mockup* yaitu suatu desain sebagai dasar pembuatan tampilan pengguna. *Mockup* dibuat agar proses pembuatan tampilan pada tahap pengembangan lebih mudah dilakukan. Bentuk dari *mockup* sendiri adalah gambar desain tampilan berupa *wireframe* yaitu gambar berupa jaring – jaring yang mewakili bentuk tertentu.

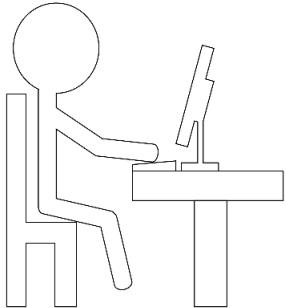
Langkah terakhir dalam tahap desain adalah desain level untuk setiap level yang ada pada permainan. Desain level dilakukan untuk menentukan tingkat kesulitan setiap level dan membuat setiap level tersebut seimbang. Setiap level pada permainan ini berisi teka – teki sesuai dengan konsep permainan yaitu *puzzle*. Teka – teki *puzzle* yang diberikan mewakili materi – materi yang ada pada algoritma sehingga dilakukan desain level agar sesuai.

a. *Storyboard*

Tabel 15. Desain *Storyboard*

No	Gambar	Keterangan
1	 Pengguna melihat intro hingga menu utama aplikasi. Pada menu utama pengguna menentukan selanjutnya akan melakukan apa	Pengguna melihat intro hingga menu utama aplikasi. Pada menu utama pengguna menentukan selanjutnya akan melakukan apa
2	 Pengguna memilih untuk bermain	Pengguna menentukan untuk bermain permainan
3		Dalam permainan, pengguna menemukan bahwa ada karakter yang harus digerakkan untuk menyelesaikan permainan

4		<p>Pengguna mengetahui cara menggerakkan karakter, selanjutnya menentukan perintah yang akan digunakan untuk menggerakkan karakter</p>
5		<p>Pengguna menyusun perintah – perintah yang akan digunakan agar karakter bergerak menuju titik akhir. Pengguna menjalankan perintahnya dengan menekan tombol <i>Play</i></p>
6		<p>Jika perintah yang disusun oleh pengguna benar maka karakter dapat mencapai titik akhir dan memenangkan level permainan. Namun jika perintah yang disusun salah maka pemain akan kalah</p>
7		<p>Setelah memenangkan, maka pengguna akan menentukan apakah akan mengulangi level atau melanjutkan pada level berikutnya</p>

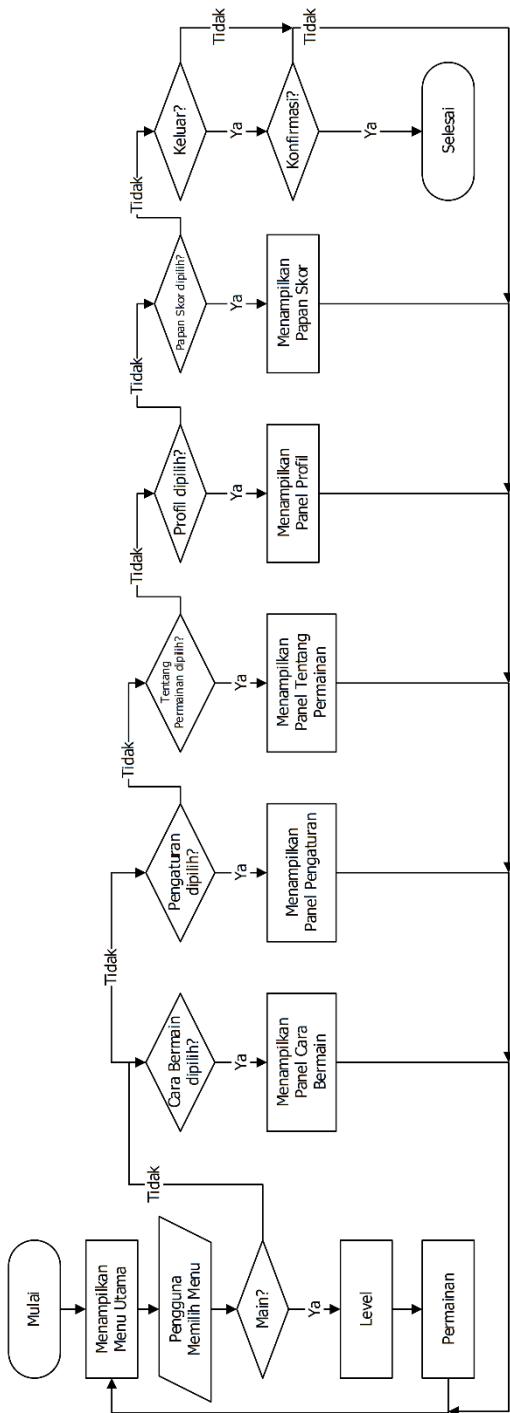
8	 <p>Setelah bermain, harapannya adalah pengguna dapat memahami struktur dan konsep pada algoritma khususnya Flowchart</p>	<p>Setelah bermain permainan ini, pengguna diharapkan dapat memahami bagaimana konsep algoritma bekerja dan dapat memahami struktur – struktur sederhana dari algoritma</p>
---	--	---

Berdasarkan *storyboard* diatas maka ditentukan apa saja yang dapat pengguna lakukan. Maka sebelum membuat *flowchart*, terlebih dahulu menentukan interaksi yang dapat dilakukan oleh pengguna terhadap permainan. Berikut ini interaksi yang dapat dilakukan oleh pengguna terhadap permainan :

- 1) Memulai permainan
- 2) Memilih menu pada menu utama
- 3) Memilih level pada menu level
- 4) Menambahkan blok perintah
- 5) Menggeser blok perintah
- 6) Menghapus blok perintah
- 7) Menghubungkan blok perintah
- 8) Menjalankan blok – blok perintah
- 9) Mengulangi level
- 10) Melanjutkan ke level berikutnya
- 11) Mengakhiri permainan

b. Flowchart

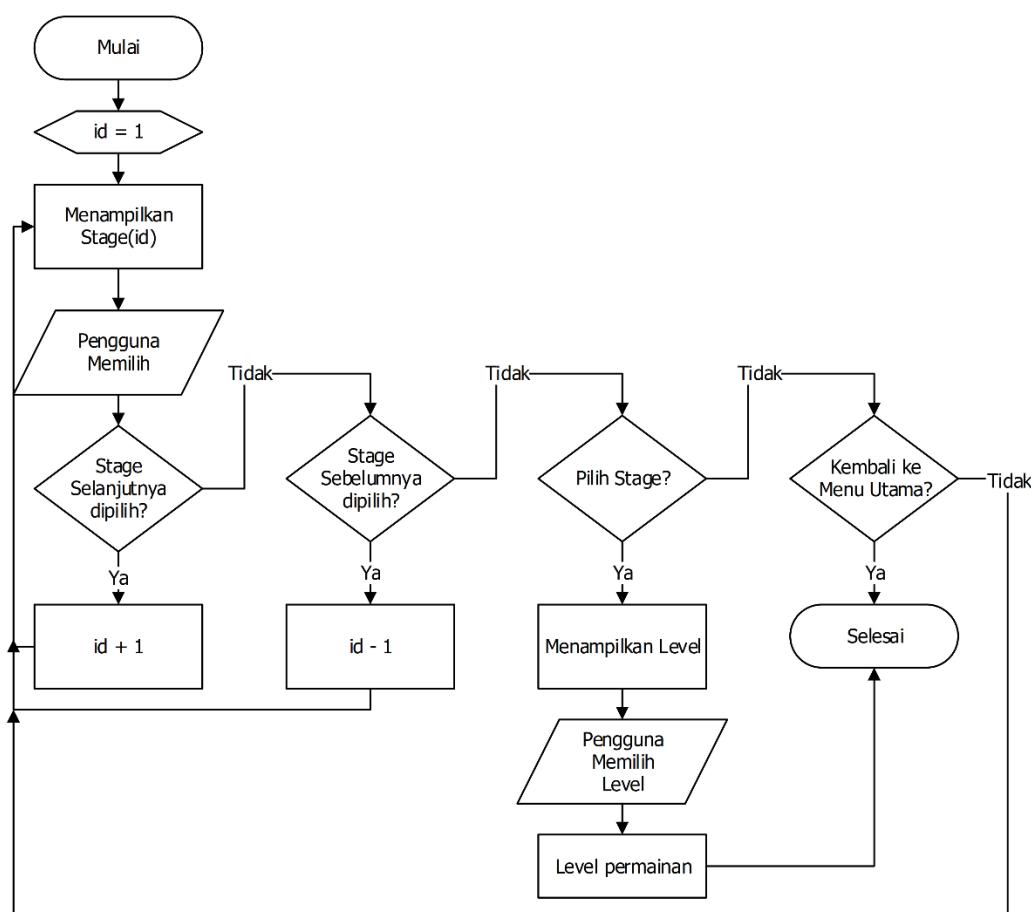
1) Menu utama



Gambar 12. Flowchart Menu Utama

Flowchart menu utama ini digunakan untuk menjelaskan mengenai proses pemilihan menu utama oleh pengguna. Pengguna menentukan untuk melakukan sesuatu pada menu utama. Terdapat beberapa pilihan menu yang ada pada menu utama yaitu "Main", "Cara Bermain", "Pengaturan", "Tentang Permainan", "Profil", "Papan Skor", dan "Keluar". Setiap menu memiliki fungsinya masing – masing yang berhubungan dengan fasilitas pada permainan.

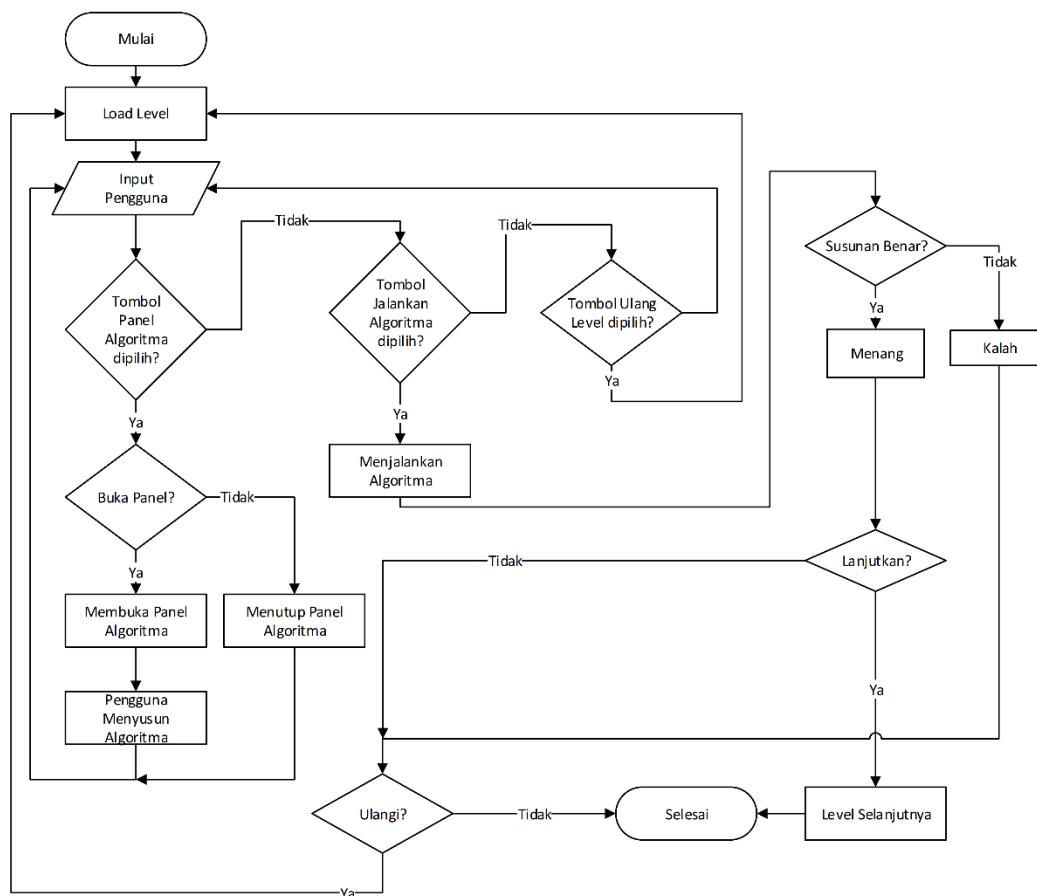
2) Memilih level



Gambar 13. *Flowchart* Menu Level

Flowchart menu level menjelaskan proses pemilihan level oleh pengguna. Terdapat beberapa *stage* dimana setiap *stage* memiliki beberapa level. Setiap *stage* memiliki jumlah level yang berbeda – beda.

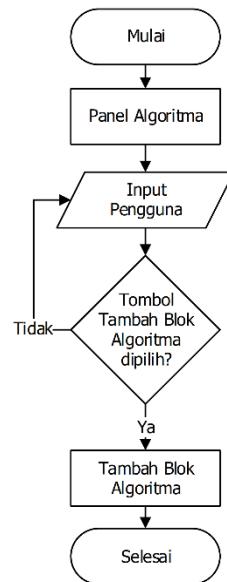
3) Permainan



Gambar 14. *Flowchart* Permainan

Flowchart ini merupakan penjabaran dari proses permainan pada *flowchart* menu utama dan menu level. Pada *flowchart* ini menjelaskan bagaimana permainan berjalan. Proses pengguna menyusun Algoritma atau blok perintah dijabarkan dalam beberapa *flowchart*, mulai dari proses penambahan blok, penggeseran blok, penghubungan blok, dan menjalankan blok perintah.

4) Proses tambah blok perintah

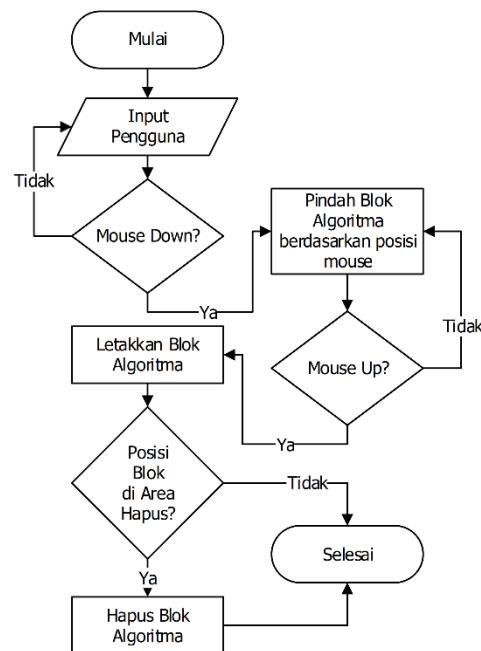


Gambar 15. *Flowchart* Tambah Blok Perintah

Proses penambahan blok perintah dijelaskan pada *flowchart* di atas.

Penambahan blok perintah berdasarkan dari input pengguna. Pengguna menentukan blok perintah mana yang akan ditambahkan.

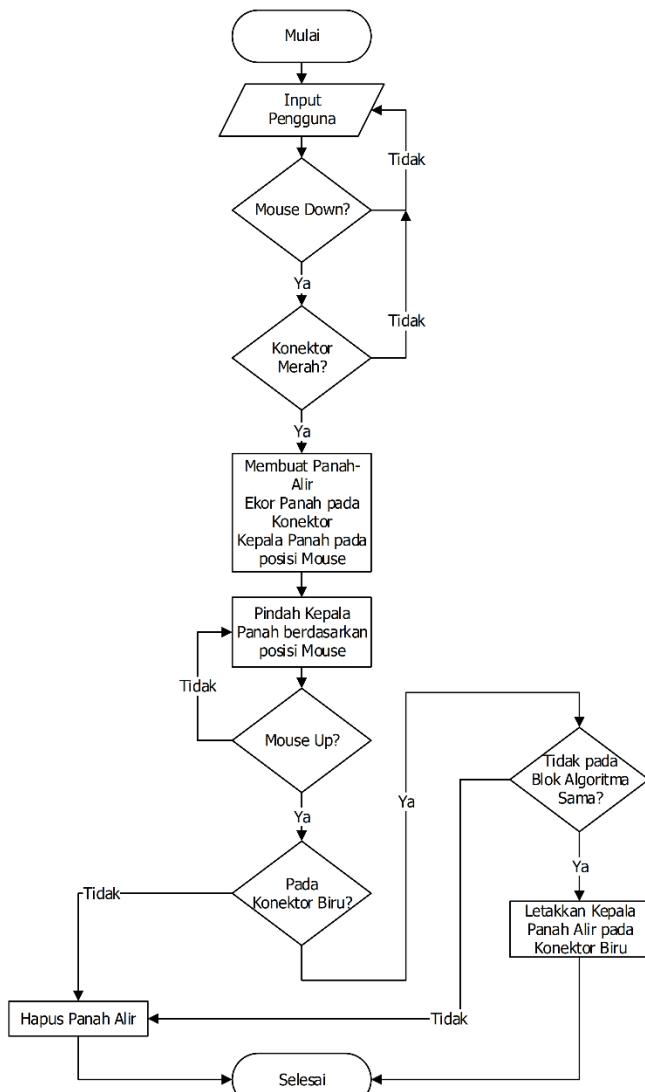
5) Proses geser blok perintah



Gambar 16. *Flowchart* Geser Blok Perintah

Proses penggeseran blok perintah berdasarkan posisi *mouse* pengguna ketika pengguna melakukan *drag* atau penggeseran *mouse* sambil menekan tombol kiri. Ketika tombol diangkat maka blok perintah akan diletakkan pada posisi *mouse* terakhir.

6) Proses menghubungkan blok perintah

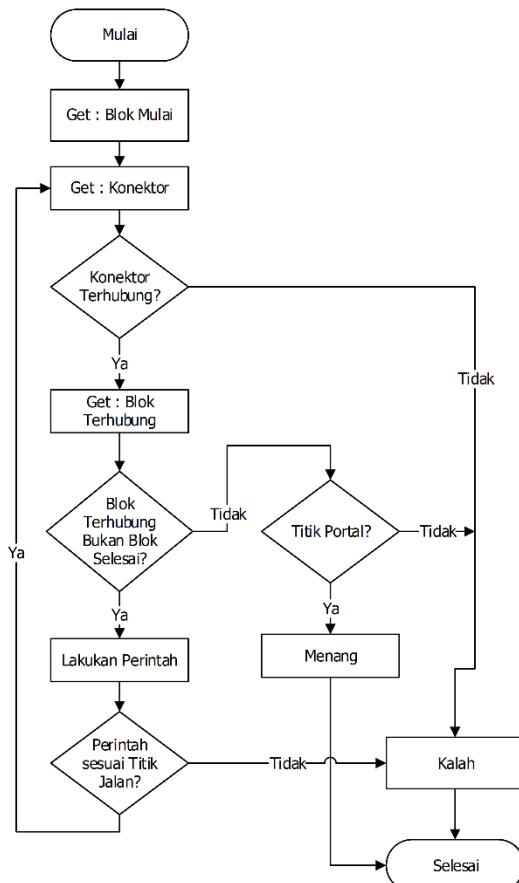


Gambar 17. Flowchart Menghubungkan Blok Perintah

Flowchart ini menjelaskan mengenai proses dimana pengguna menghubungkan blok perintah dengan panah alir yang dibuat dengan cara melakukan *drag and drop* melalui konektor yang ada pada masing – masing blok

perintah. Pada blok perintah terdapat 2 jenis konektor, yaitu konektor biru dan merah. Konektor biru hanya dapat digunakan untuk memulai panah alir dan konektor merah hanya dapat digunakan untuk mengakhiri panah alir.

7) Menjalankan blok perintah yang telah tersusun



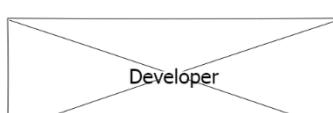
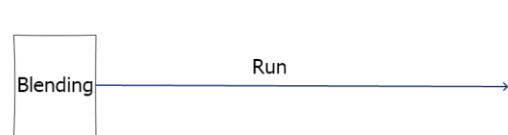
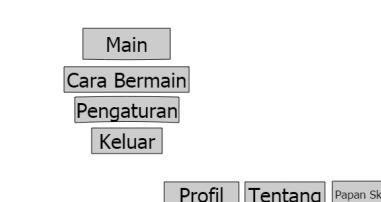
Gambar 18. *Flowchart* Menjalankan Blok Perintah

Flowchart ini menjelaskan mengenai jalannya blok perintah. Karakter akan beraksi sesuai dengan perintah yang sedang dijalankan. Jika perintah sesuai dengan "Titik Jalan" yang ada pada peta permainan maka akan dilanjutkan pada perintah berikutnya. Jika perintah salah maka pemain terhitung kalah. Ketika karakter permainan berhenti pada "Titik Portal" maka pemain terhitung menang.

c. Mockup

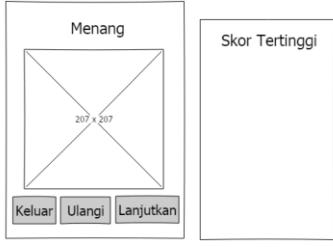
Mockup dibuat untuk memberikan gambaran dan kemudahan dalam proses pengembangan tampilan aplikasi. *Mockup* berbentuk rancangan – rancangan yang nantinya menjadi dasar dalam pembuatan antarmuka aplikasi.

Tabel 16. *Mockup*

No.	Nama	Gambar	Keterangan
1	Opening 1		Tampilan awal dengan animasi yang dilanjutkan transisi menuju Opening 2
2	Opening 2		Tampilan awal kelanjutan yang menampilkan nama pengembang
3	Opening 3		Tampilan awal kelanjutan yang menampilkan judul permainan dan animasi karakter permainan yang sedang berlari
4	Menu Utama		Halaman menu utama dengan berbagai tombol yang memiliki fungsi tertera. PS berarti Papan Skor.

5	Panel Cara Bermain		Halaman ini muncul ketika pengguna memencet tombol cara bermain pada halaman menu utama
6	Panel Pengaturan		Halaman ini muncul ketika pengguna memencet tombol pengaturan pada halaman utama
7	Panel Profil		Halaman ini muncul ketika pengguna memencet tombol profil pada halaman utama
8	Panel Tentang Permainan		Halaman ini muncul ketika pengguna memencet tombol tentang pada halaman utama
9	Papan Skor		Halaman ini muncul ketika pengguna memencet tombol papan skor

10	Menu Pilih Level	<p>Pilih Level</p> <p>Sebelumnya Selanjutnya</p> <p>Kembali</p>	Tampilan utama pada halaman menu pilih level yang menampilkan <i>stage</i>
11	Panel Level	<p>Pilih Level</p> <p>X</p> <p>Kembali</p>	Panel yang muncul setelah pengguna memilih <i>stage</i> , selanjutnya pengguna memilih level
12	Skema Permainan	<p>Judul Level</p> <p>Play Ulangi Pause Papan Algoritma</p>	Tampilan antarmuka pengguna pada halaman permainan
13	Panel Algoritma	<p>Mulai</p> <p>Pernah</p> <p>Selanjutnya</p>	Halaman akan ditampilkan ketika pengguna memencet tombol papan algoritma pada halaman permainan
14	Panel Pause	<p>Pause</p> <p>Lanjutkan Menu Utama Keluar</p>	Panel pause yang terdapat ketika pengguna menekan tombol pause

15	Kondisi Menang		Tampilan ketika pengguna menyelesaikan satu level
16	Kondisi Kalah		Tampilan ketika pengguna kalah pada suatu level

d. Desain level

Tahap terakhir pada tahap ini adalah desain teka – teki untuk setiap level yang akan dimainkan. Desain level disesuaikan dengan materi – materi yang akan disampaikan karena melalui konsep teka – teki pada setiap level materi disampaikan kepada pengguna. Terdapat 27 level berbeda yang dibagi dalam 4 *stage*. Dokumen desain level dapat dilihat pada lampiran 9 desain level.

Sistem scoring untuk setiap level berdasarkan jumlah penggunaan blok perintah. Jumlah blok perintah yang semakin sedikit maka semakin baik karena dapat dikatakan bahwa susunan menjadi lebih efektif sehingga proses yang dilakukan menjadi lebih cepat.

3. Tahap Pengembangan

Tahap pengembangan adalah tahap pembuatan aplikasi yang didasari dari hasil desain. Pada tahap ini desain yang sudah dibuat pada tahap sebelumnya dapat digunakan sebagai tumpuan pembuatan tampilan dan penulisan kode

program untuk aplikasi. Pembuatan aplikasi permainan ini menggunakan perangkat lunak *game engine Unity 3D* dengan bahasa pemrograman C#.

a. Pembuatan tampilan antarmuka pengguna

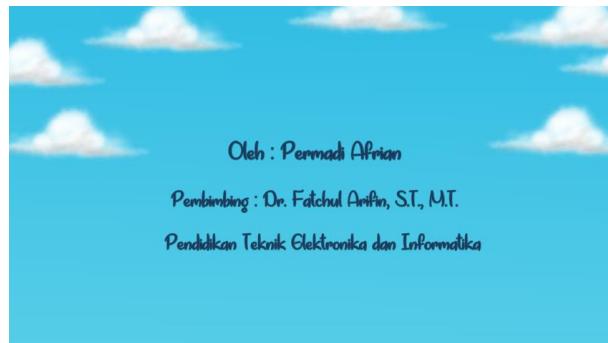
Tampilan antarmuka dibuat dengan menggunakan perangkat lunak pengolah gambar Adobe Photoshop CC 2017. Setiap elemen pada tampilan antarmuka dibuat pada satu halaman namun terpisah untuk mempermudahkan penyusunan antarmuka gambar pada *game engine*. Berikut ini adalah tampilan antarmuka pengguna pada permainan :

1) Halaman intro

Halaman intro adalah halaman yang menjadi pengenalan bagi pengguna untuk mengetahui mengenai aplikasi. Halaman intro berisi gambar dan tulisan dimana pengguna hanya melihat tanpa melakukan suatu apapun. Pada halaman ini terdiri dari beberapa rangkaian tampilan. Berikut ini tampilan yang ada pada halaman intro:



Gambar 19. Tampilan 1 Intro



Gambar 20. Tampilan 2 Intro



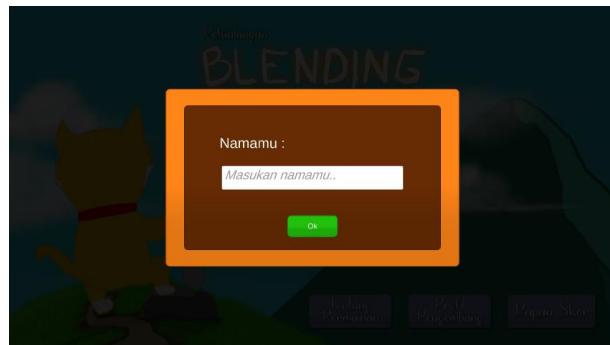
Gambar 21. Tampilan 3 Intro

2) Halaman menu utama

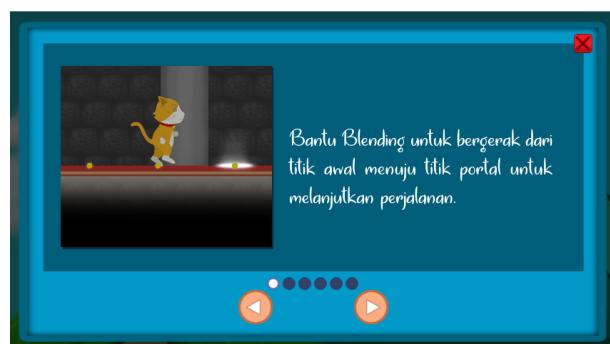
Halaman menu utama adalah tampilan awal bagi pengguna. Saat pengguna pertama kali membuka permainan maka ketika pertama kali masuk pada menu utama akan muncul panel untuk mengisi nama, jika tidak maka akan langsung masuk pada tampilan menu utama. Pada halaman ini terdapat beberapa menu seperti Main, Cara Bermain, Pengaturan, Tentang, Profil, dan Papan Skor.



Gambar 22. Tampilan Menu Utama



Gambar 23. Tampilan Panel Input Nama



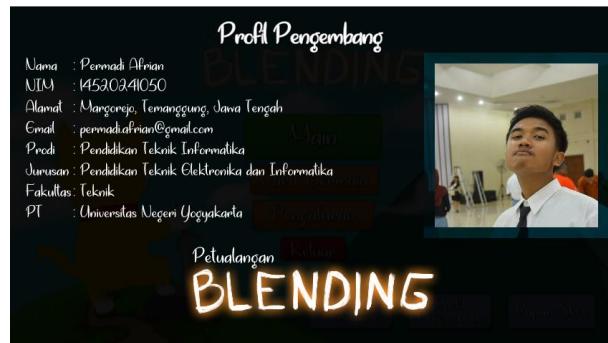
Gambar 24. Tampilan Panel Cara Bermain



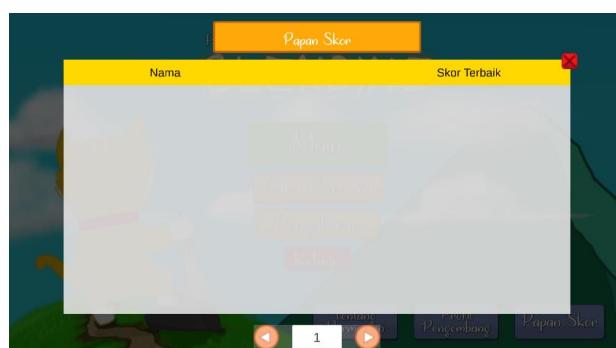
Gambar 25. Tampilan Panel Pengaturan



Gambar 26. Tampilan Panel Tentang Permainan



Gambar 27. Tampilan Panel Profil Pengembang



Gambar 28. Tampilan Panel Papan Skor

3) Halaman pilih level

Halaman pilih level terdiri dari 2 segmen, yaitu segmen untuk memilih *stage* dan segmen untuk memilih level. Pada pemilihan *stage* terdapat indikator *progress* pada *stage* tersebut. Berikut ini adalah tampilan yang ada pada halaman pilih level.



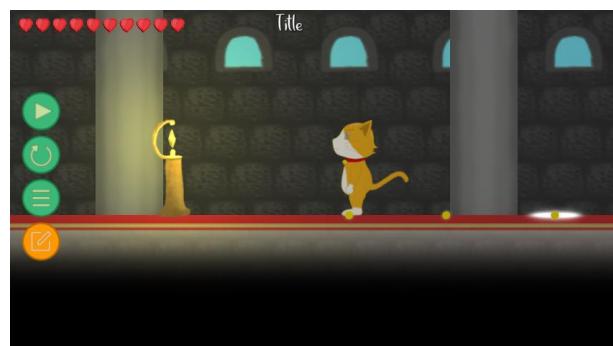
Gambar 29. Tampilan Pilih Stage



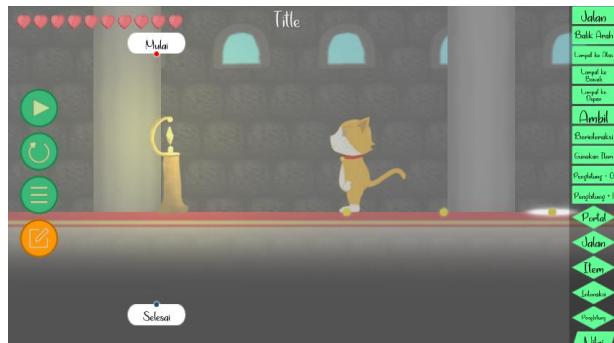
Gambar 30. Tampilan Pilih Level

4) Halaman permainan

Halaman permainan memiliki beberapa elemen yaitu *background*, peta permainan, karakter, dan antarmuka pengguna. Setiap level memiliki antarmuka pengguna dan karakter yang sama, hanya dibedakan pada *background* dan peta permainan. *Background* menyesuaikan dengan *stage* atau tema pada level tersebut. Peta permainan menyesuaikan dengan hasil desain level yang sudah dibuat. Peta permainan berupa jalan yang dilalui oleh karakter dan rintangan – rintangan yang dihadapi.



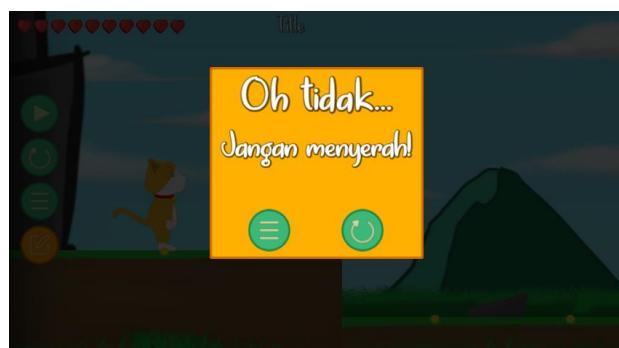
Gambar 31. Tampilan Halaman Permainan



Gambar 32. Tampilan Panel Algoritma



Gambar 33. Tampilan Kemenangan



Gambar 34. Tampilan Kekalahan

b. Pembuatan kode program

Pembuatan kode program atau penulisan kode program dibuat pada perangkat lunak Visual Studio 2017. Penulisan kode program berdasarkan pada data *flowchart* dan kebutuhan tampilan agar dapat berfungsi dengan semestinya. Bahasa pemrograman menyesuaikan dengan bahasa pemrograman yang digunakan oleh *game engine* Unity 3D sehingga menggunakan bahasa C#.

1) Menu utama

Pada menu utama terdapat berbagai menu dengan fungsinya masing – masing. Penulisan kode program menyesuaikan dengan *flowchart* dan fungsi yang dibutuhkan pada menu utama. Terdapat beberapa kode yang berbeda pada menu utama yaitu *MenuManager* untuk mengatur fungsi – fungsi pada menu utama, *AudioContent* untuk mengatur fungsi suara, *SaveLoad* yang digunakan terkait fungsi penyimpanan data pada *disk*, *Highscore* untuk mengatur fungsi – fungsi terkait *download* dan *upload* skor pengguna, *AudioSetting* dan *Quality Dropdown* menjadi fungsi pada pengaturan untuk mengatur suara dan kualitas gambar yang ada pada permainan. Selain kode buatan tersebut juga digunakan kode yang berasal dari *game engine* Unity 3D atau yang disebut sebagai *built-in script*. Seluruh kode yang dibuat untuk menu utama dapat dilihat pada lampiran 10.

2) Menu pilih level

Menu pilih level terdiri dari beberapa fungsi yaitu memilih *stage* dan memilih level. Pada menu pemilihan *stage* menggunakan *built-in script*, sedangkan pada menu pemilihan level menggunakan kode tambahan yaitu *LevelSelectManager*. Kode tersebut mengatur untuk melakukan *loading* terhadap level yang dipilih. Kode pada menu pilih level dapat dilihat pada lampiran 10.

3) Permainan

Permainan memiliki beberapa fungsi terkait jalannya permainan. Salah satu fungsi yang ada yaitu proses menjalankan blok algoritma atau blok perintah yang disusun oleh pengguna. Pembuatan kode fungsi tersebut berdasarkan *flowchart* yang telah dibuat pada tahap desain dan *trial and error* yang terjadi

ketika permainan dijalankan. Kode tersebut dapat dilihat pada lampiran 10 dengan nama *AlgorithmSystem*.

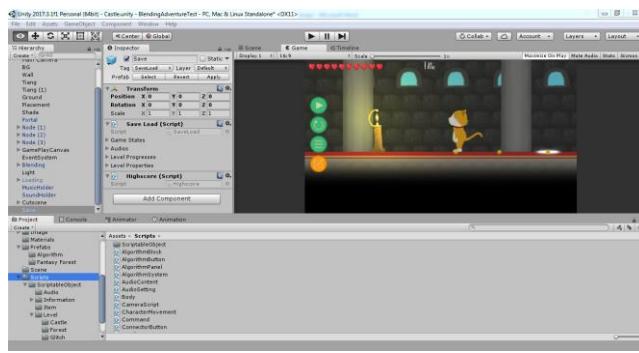
Fungsi lain yang terdapat pada permainan adalah fungsi titik jalan atau *node* agar setiap titik jalan dapat terhubung sesuai dengan arah yang diinginkan. Terdapat beberapa arah gerak yaitu jalan, lompat ke depan, lompat ke atas, dan lompat ke bawah. Setiap gerakan digerakkan menggunakan blok perintah yang sesuai dengan titik yang akan dituju.

LevelManager adalah fungsi yang digunakan untuk mengatur jalannya setiap level. Tugas utama kode tersebut adalah sebagai *loader* untuk level berikutnya atau menu utama. Selain itu kode tersebut berguna sebagai inisiasi halaman permainan. Kode tersebut dapat dilihat pada lampiran 10.

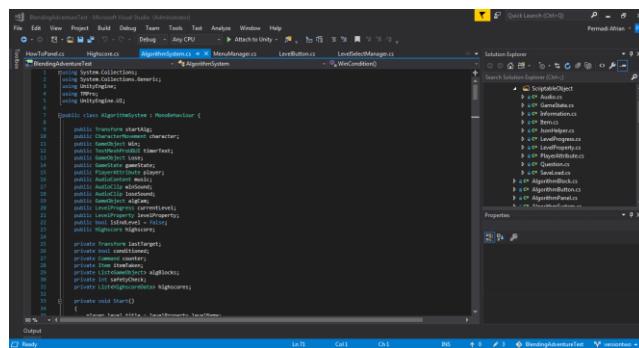
Terdapat 1 kode program yang digunakan untuk mengatur gerakan karakter, yaitu *CharacterMovement*. Kode program ini menentukan gerakan yang sedang dikerjakan dan melakukannya dengan menjalankan animasi yang sesuai dengan gerakan. Kode ini dapat dilihat pada lampiran 10.

Kode program utama yang lain yang sangat penting adalah *AlgorithmBlock*, *AlgorithmButton*, dan *ConnectorButton*. 3 kode tersebut mengatur tentang kinerja fungsi terkait blok perintah seperti pergeseran blok, penambahan blok, penghapusan blok, dan penghubungan blok. Kode – kode tersebut dapat dilihat pada lampiran 10.

Selain kode – kode program utama yang telah disebutkan terdapat beberapa kode tambahan sebagai dukungan program, *optimization* tampilan, dan berbagai fungsi yang ditambahkan ketika proses pengembangan. Semua kode dapat dilihat pada lampiran 10 untuk halaman permainan.



Gambar 35. Tampilan proses pengembangan pada Unity 3D



Gambar 36. Tampilan proses pengembangan pada Visual Studio 2017

4) Uji fungsionalitas dan *compatibility*

Setelah aplikasi selesai dilakukan pengembangan dilakukan 2 pengujian yaitu fungsionalitas dan *compatibility*. Pengujian dilakukan sebelum aplikasi diimplementasikan kepada pengguna. Pada pengujian ini diuji mengenai setiap fungsi yang ada pada permainan terkait fungsi teknis yang ada. Pengujian *compatibility* dilakukan untuk mengetahui kesesuaian aplikasi dengan perangkat yang menjadi target implementasi aplikasi baik perangkat lunak ataupun perangkat keras. Pengujian dilakukan oleh 2 orang ahli pada bidang pengembangan permainan dari perusahaan pengembang permainan yang berbeda. Berikut ini hasil dari uji fungsionalitas dan *compatibility*.

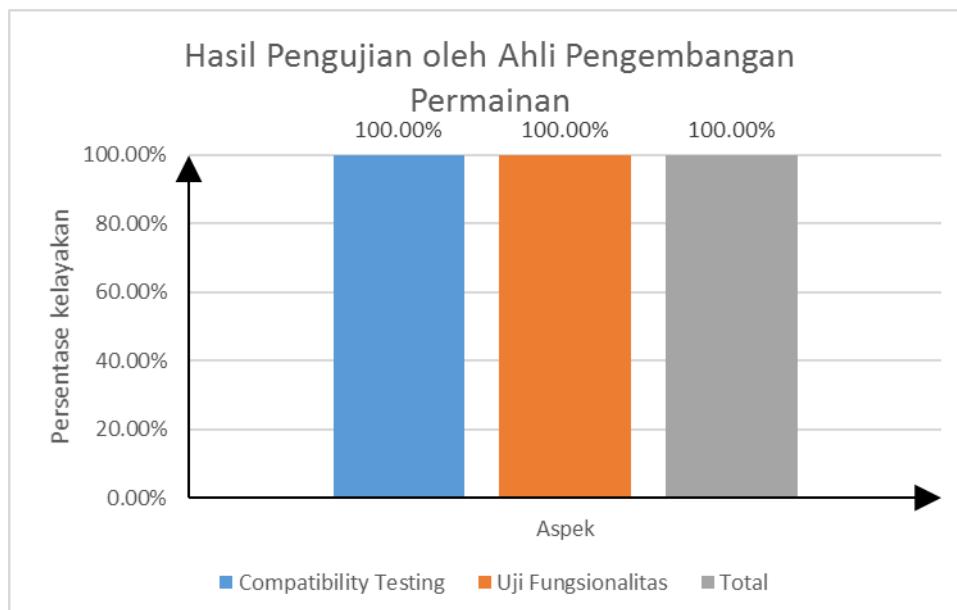
Tabel 17. Hasil *Compatibility Testing*

No	Penguji	Skor Akhir	Skor Ideal	Persentase Kelayakan
1	1	12	12	100%
2	2	12	12	100%

Tabel 18. Hasil Uji Fungsionalitas

No	Penguji	Skor Akhir	Skor Ideal	Persentase Kelayakan
1	1	48	48	100%
2	2	48	48	100%

Dari data diatas dapat digambarkan dalam sebuah diagram batang sebagai berikut ini.



Gambar 37. Grafik hasil pengujian oleh ahli pengembangan permainan Berdasarkan hasil uji oleh ahli pengembangan permainan, uji fungsionalitas mendapatkan persentase kelayakan sebesar 100% sehingga dapat dikatakan bahwa semua fungsi yang ada pada permainan dapat berjalan sesuai dengan harapan. Hasil *compatibility testing* mendapatkan persentase kelayakan

sebesar 100% yang berarti permainan dapat dijalankan pada berbagai perangkat yang ditargetkan. Dengan hasil demikian maka permainan sudah layak diimplementasikan untuk pengujian pada tahap selanjutnya.

4. Tahap Implementasi

Tahap implementasi merupakan tahap dimana aplikasi digunakan oleh pengguna diluar lingkungan pengembang permainan. Pada tahap ini aplikasi diuji kualitasnya dari segi medianya dan juga isi permainannya. Pada tahap pertama pengujian dilakukan pengujian oleh ahli media terhadap aplikasi menggunakan kuesioner. Beberapa aspek yang diuji oleh ahli media adalah aspek *balance*, *compliance*, *playtesting*, dan *usability*. Setelah aplikasi diuji oleh ahli media, selanjutnya dilakukan pengujian terhadap isi materi aplikasi oleh ahli materi. Pengujian isi materi menggunakan kuesioner dengan pernyataan berdasarkan materi yang digunakan pada aplikasi. Tahap terakhir pada tahap implementasi adalah pengujian oleh pengguna. Pengguna dalam pengujian ini adalah siswa SMK kelas X di SMK Negeri 1 Magelang yang berjumlah 30 orang. Pada pengujian ini diujikan aspek – aspek untuk mengetahui kelayakan aplikasi pada pengguna yaitu *balance*, *playtesting*, materi, dan *usability*.

a. Hasil pengujian oleh ahli media

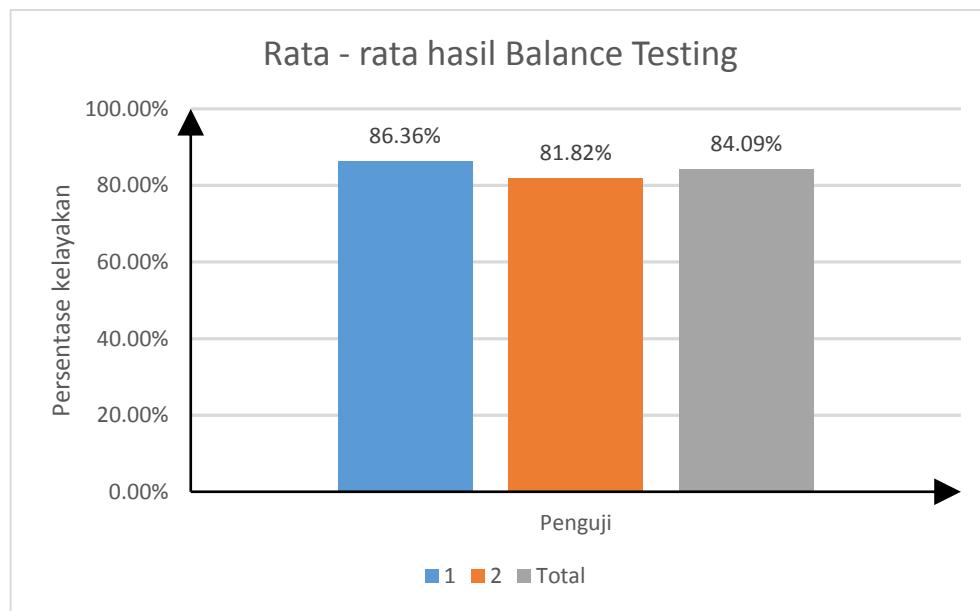
Pengujian aplikasi dilakukan oleh 2 ahli media yang kompeten pada bidangnya. Terdapat 4 aspek yang diujikan pada ahli media yaitu *balance testing*, *compliance testing*, *playtesting*, dan *usability testing*. Aspek *balance testing* untuk menguji keseimbangan isi permainan untuk memastikan tingkat kesulitan agar masih dalam batas wajar dan mudah untuk dipahami tanpa menyulitkan pengguna dalam menggunakan aplikasi. Aspek *compliance testing* adalah pengujian

pemenuhan standar yang didefinisikan oleh *Windows* selaku sistem operasi yang digunakan untuk menjalankan aplikasi. Aspek *playtesting* adalah pengujian untuk mengetahui kemudahan kendali permainan dan mengetahui kondisi permainan. Aspek terakhir adalah *usability testing* yaitu pengujian yang dilakukan untuk mengetahui kegunaan aplikasi, *screen layout*, dan pencegahan kesalahan yang ada pada aplikasi. Berikut ini hasil pengujian oleh ahli media.

Tabel 19. Hasil Pengujian *Balance Testing*

Nomor Butir	Penguji 1	Penguji 2	Total
1	3	4	7
2	4	4	8
3	3	3	6
4	3	3	6
5	4	3	7
6	3	3	6
7	4	3	7
8	3	4	7
9	3	3	6
10	4	3	7
11	4	3	7
Total	38	36	74
Rata - rata	3,45	3,27	3,36
Persentase	86,36%	81,82%	84,09%

Dari data diatas dapat digambarkan dalam bentuk diagram batang dibawah ini.

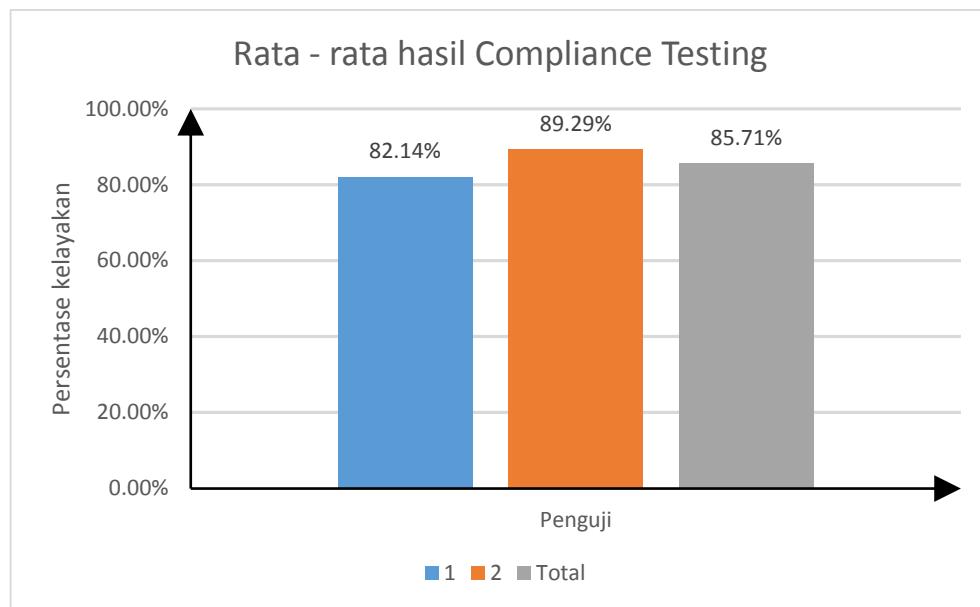


Gambar 38. Grafik rata – rata hasil *balance testing*

Tabel 20. Hasil Pengujian *Compliance Testing*

Nomor Butir	Penguji 1	Penguji 2	Total
1	4	3	7
2	4	4	8
3	3	4	7
4	3	3	6
5	3	3	6
6	3	4	7
7	3	4	7
Total	23	25	48
Rata - rata	3,29	3,57	3,43
Persentase	82,14%	89,29%	85,71%

Dari data diatas dapat digambarkan dalam bentuk diagram batang dibawah ini.

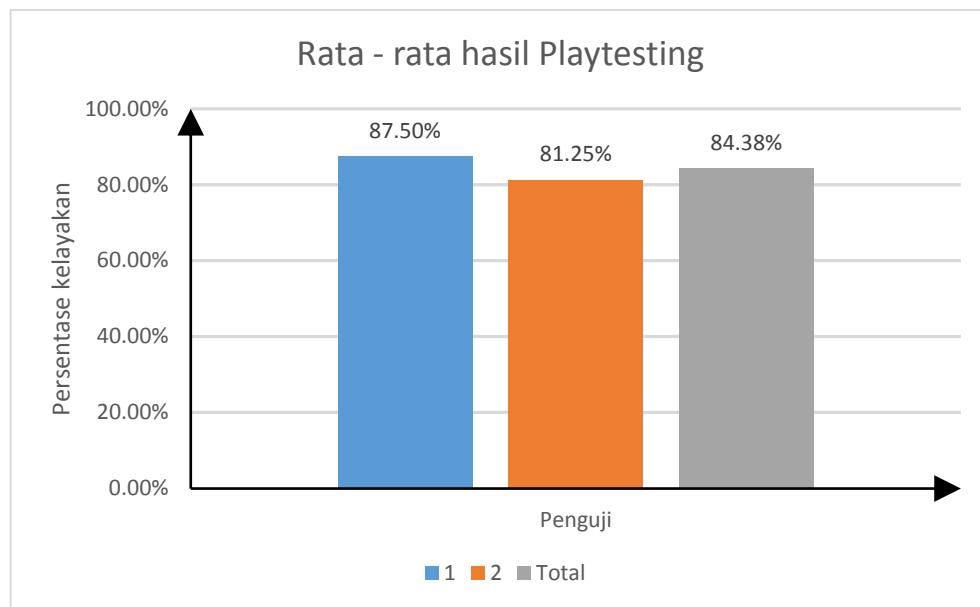


Gambar 39. Grafik rata – rata hasil *compliance testing*

Tabel 21. Hasil Pengujian *Playtesting*

Nomor Butir	Pengugi 1	Pengugi 2	Total
1	3	3	6
2	3	3	6
3	4	3	7
4	4	3	7
5	4	3	7
6	4	4	8
7	3	3	6
8	3	4	7
Total	28	26	54
Rata - rata	3,50	3,25	3,38
Persentase	87,50%	81,25%	84,38%

Dari data diatas dapat digambarkan dalam bentuk diagram batang dibawah ini.

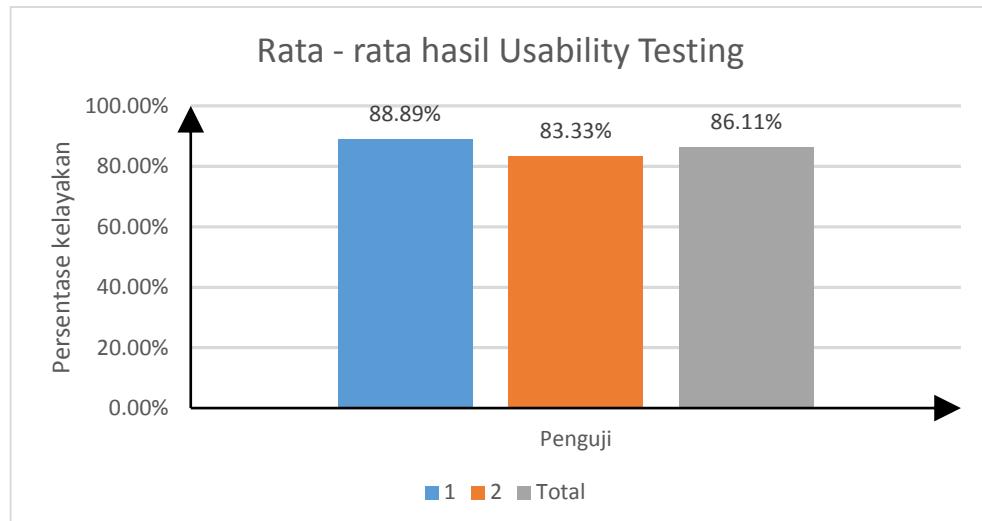


Gambar 40. Grafik rata – rata hasil *playtesting*

Tabel 22. Hasil Pengujian *Usability Testing*

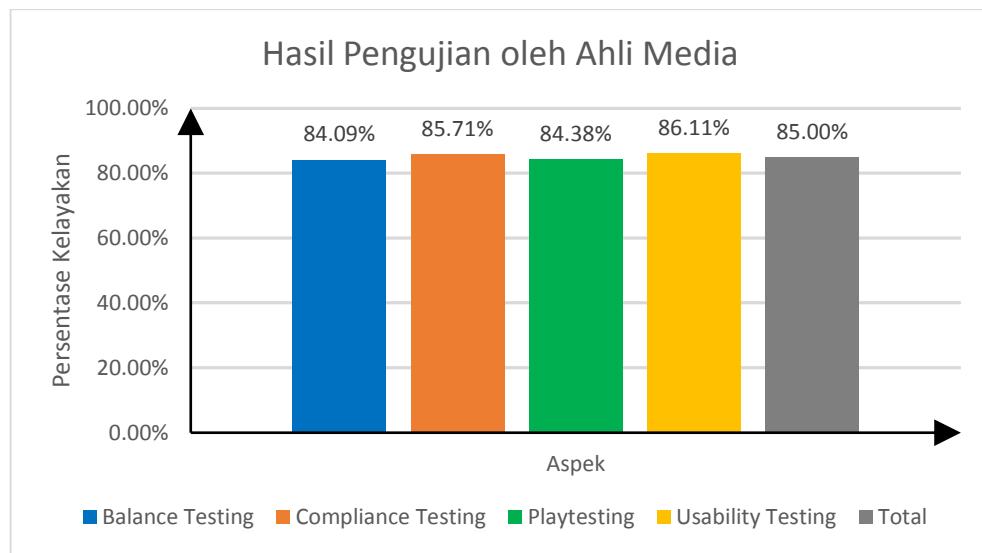
Nomor Butir	Pengugi 1	Pengugi 2	Total
1	4	4	8
2	4	3	7
3	4	3	7
4	4	3	7
5	4	4	8
6	3	4	7
7	3	3	6
8	3	3	6
9	3	3	6
Total	32	30	62
Rata - rata	3,56	3,33	3,44
Persentase	88,89%	83,33%	86,11%

Dari data diatas dapat digambarkan dalam bentuk diagram batang dibawah ini.



Gambar 41. Grafik rata – rata hasil *usability testing*

Data – data yang telah dipaparkan diatas dapat diringkas dalam bentuk diagram batang dalam tiap aspek sebagai berikut ini.



Gambar 42. Grafik hasil pengujian oleh ahli media

Dari data pengujian diatas dapat disimpulkan bahwa hasil pengujian aspek *balance testing* mendapatkan skor 74 dengan rata – rata sebesar 3,36 dan persentase kelayakan 84,09%. Aspek *compliance testing* mendapatkan skor 48 dengan rata – rata sebesar 3,34 dan persentase kelayakan 85,71%. Aspek *playtesting* mendapatkan skor 54 dengan rata – rata sebesar 3,38 dan persentase

kelayakan 84,38%. Aspek *usability testing* mendapatkan skor 62 dengan rata - rata sebesar 3,44 dan persentase kelayakan 86,11%. Secara keseluruhan maka didapatkan persentase kelayakan sebesar 85% dengan kategori **Sangat Layak**. Dengan demikian maka permainan dapat dilakukan pengujian untuk tahap selanjutnya.

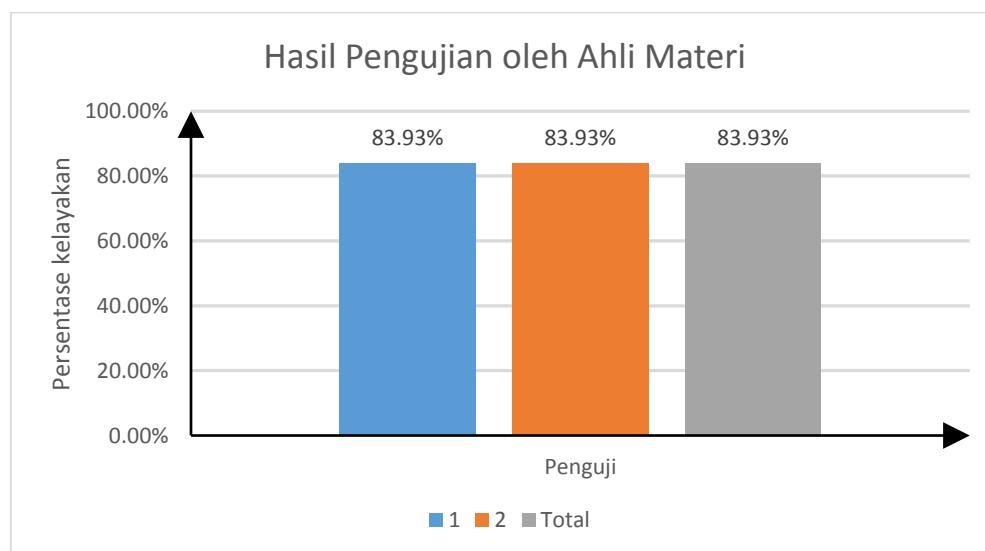
b. Hasil pengujian oleh ahli materi

Pengujian aplikasi dalam aspek materi dilakukan oleh 2 ahli materi yaitu guru di SMK Negeri 1 Magelang. Pada pengujian materi dilakukan uji kualitas materi yang terdapat pada permainan. Berikut ini hasil pengujian oleh ahli materi.

Tabel 23. Hasil Uji Ahli Materi

Penguji	Hasil	Rata - rata	Persentase Kelayakan
1	47	3,36	83,93%
2	47	3,36	83,93%
Total	94	3,36	83,93%

Dari data diatas dapat digambarkan dalam bentuk diagram batang sebagai berikut ini.



Gambar 43. Grafik hasil pengujian oleh ahli materi

Dari data pengujian diatas dapat disimpulkan bahwa hasil pengujian aspek materi mendapatkan kategori **Layak** berdasarkan perhitungan kategorisasi skala pada tabel 13 dengan rata – rata 3,36 dan persentase kelayakan sebesar 83,93%. Dengan demikian maka permainan dapat dilakukan pengujian tahap selanjutnya yaitu pengujian oleh pengguna.

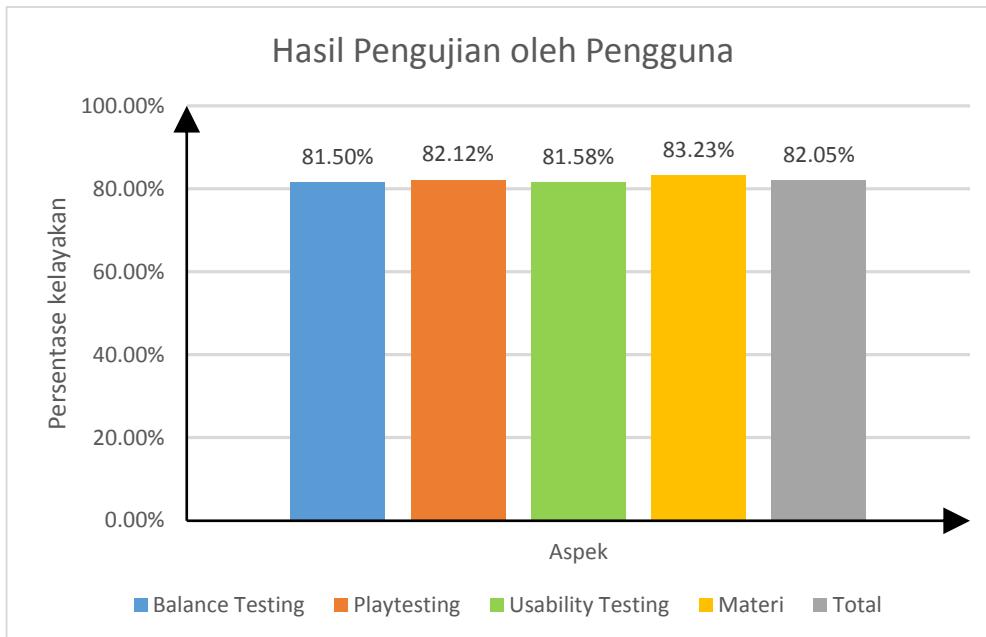
c. Hasil pengujian oleh pengguna

Tahap pengujian yang terakhir dilakukan oleh pengguna. Pada penelitian ini pengguna merupakan siswa kelas X jurusan Teknik Komputer dan Informatika yang menempuh mata pelajaran pemrograman dasar kelas X. Responden berjumlah 30 siswa kelas X di SMK Negeri 1 Magelang jurusan Teknik Komputer dan Informatika. Pengujian menggunakan kuesioner dengan 4 aspek penilaian yaitu *balance testing*, *playtesting*, *usability testing*, dan materi. Pengujian dilakukan guna mengetahui tingkat kelayakan aplikasi ketika digunakan oleh pengguna langsung. Berikut ini hasil pengujian dan perhitungan hasil pengujian yang telah dilakukan.

Tabel 24. Hasil Uji oleh Pengguna

Aspek	Hasil	Rata - rata	Persentase Kelayakan
Balance Testing	978	3,260	81,50%
Playtesting	1084	3,285	82,12%
Usability Testing	979	3,263	81,58%
Materi	799	3,329	83,23%
Total	3840	3,282	82,05%

Dari data diatas dapat digambarkan dengan menggunakan diagram batang sebagai berikut ini.



Gambar 44. Grafik hasil pengujian oleh pengguna

Dari data yang telah disajikan tersebut maka dapat disimpulkan bahwa hasil pengujian permainan oleh pengguna mendapatkan skor sebanyak 3840 dengan persentase kelayakan 82,05% sehingga berdasarkan perhitungan menggunakan kategorisasi skala menggunakan tabel 13, pengujian oleh pengguna mendapatkan kategori **Layak**. Perhitungan *Cronbach's Alpha* untuk menguji nilai konsistensi data menggunakan bantuan perangkat lunak excel. Hasil perhitungan *Cronbach's Alpha* menunjukkan skor 0,92. Berdasarkan skor tersebut maka hasil pengujian konsistensi data masuk dalam kategori ***Excellent***.

5. Tahap Evaluasi

Tahap terakhir dalam pengembangan permainan ini adalah tahap evaluasi. Tahap evaluasi ini berupa proses menganalisa bagaimana hasil dari implementasi dan pengujian yang telah dilakukan pada tahap sebelumnya. Hasil analisa pada tahap evaluasi berupa kualitas dari permainan yang dikembangkan,

kelebihan dan kekurangan yang ada pada permainan, kendala pengembangan permainan, dan rekomendasi untuk pengembangan permainan selanjutnya.

a. Evaluasi Kualitas pada Permainan

Pengujian kualitas permainan dilakukan pada 3 tahap. Tahap pertama dilakukan pengujian terhadap kualitas permainan dari sudut pandang ahli media yang dilakukan oleh 2 ahli media. Berdasarkan pengujian oleh ahli media, hasil uji keseluruhan aspek mendapatkan persentase kelayakan sebesar 85%. Permainan juga mendapatkan tanggapan yang sangat baik dari ahli media. Ahli media menyatakan bahwa konsep dari permainan sangat bagus. Tahap kedua dilakukan pengujian terhadap kualitas isi materi yang ada pada permainan menurut sudut pandang ahli materi. Pengujian dilakukan oleh 2 ahli materi yang telah menguasai materi – materi yang ada pada permainan. Berdasarkan pengujian oleh ahli materi mendapatkan persentase kelayakan sebesar 83,93%. Tanggapan dari ahli materi sangat baik, kedua ahli menyatakan bahwa konsep penggunaan permainan untuk mengajarkan materi tersebut sangat bagus. Tahap terakhir pengujian dilakukan oleh pengguna yaitu siswa kelas X. Hasil pengujian oleh pengguna pada keseluruhan aspek mendapatkan persentase kelayakan sebesar 82,05%.

b. Kelebihan pada Permainan

Berdasarkan tanggapan dari berbagai responden pada tahap implementasi yang telah dilakukan, maka didapatkan beberapa kelebihan – kelebihan pada permainan yaitu sebagai berikut ini.

- 1) Konsep permainan *puzzle* bagus digunakan dalam memberikan materi konsep algoritma.

- 2) Penggunaan konsep *flowchart* pada permainan membuat kegiatan mempelajari konsep *flowchart* pada algoritma menjadi lebih menarik.
- 3) Permainan menjadi alternatif media untuk belajar konsep algoritma.
- 4) Siswa menyukai permainan untuk belajar algoritma karena lebih menarik dan menyenangkan.
- 5) Penggunaan permainan sebagai media pembelajaran algoritma membuat siswa lebih menyukai belajar algoritma.

c. Kekurangan pada Permainan

Berdasarkan tanggapan dari berbagai responden pada tahap implementasi yang telah dilakukan, maka didapatkan beberapa kekurangan pada permainan yaitu sebagai berikut ini.

- 1) Permainan yang dikembangkan masih terbatas pada perangkat komputer.
- 2) Materi yang disampaikan masih terbatas pada konsep dasar algoritma menggunakan *flowchart*.
- 3) Pengguna yang awam dengan jenis permainan akan sedikit kesulitan ketika pertama kali memainkan.

d. Kendala Pengembangan pada Permainan

Selama proses pengembangan permainan, peneliti mengalami beberapa kendala yang menjadi hambatan dalam proses pengembangan permainan tersebut. Hambatan – hambatan tersebut adalah sebagai berikut ini.

- 1) Waktu pengembangan yang terbatas.
- 2) Terbatasnya kemampuan pembuatan model 3 dimensi membuat permainan hanya dibuat dalam bentuk 2 dimensi.

- 3) Terbatasnya referensi penelitian mengenai pengembangan permainan *puzzle* sebagai media pembelajaran algoritma yang ada.

e. Rekomendasi untuk Pengembangan Permainan Selanjutnya

Berdasarkan kekurangan dan kendala yang ada pada proses pengembangan permainan, maka didapatkan rekomendasi untuk pengembangan permainan selanjutnya yaitu sebagai berikut ini.

- 1) Pengembangan untuk perangkat – perangkat lain agar lebih mudah digunakan tanpa terbatas pada perangkat komputer.
- 2) Pengembangan permainan dengan model 3 dimensi agar jenis perintah dan gerak yang ada pada permainan menjadi lebih luas sehingga mencakup materi lebih detail.
- 3) Konsep permainan yang ditambahkan tidak hanya sebatas konsep permainan menggunakan *flowchart* saja tetapi terdapat konsep *pseudocode* dan konsep – konsep algoritma yang lain.
- 4) Pengembangan pengalaman pengguna yang lebih baik agar pengguna yang pertama kali memainkan permainan akan lebih mudah untuk memainkan permainan tersebut.

B. Pembahasan Hasil Penelitian

Terdapat 4 proses pengujian aplikasi permainan yaitu pengujian oleh ahli pengembangan permainan, pengujian oleh ahli media, pengujian oleh ahli materi, dan pengujian oleh pengguna.

a. Pengujian oleh Ahli Pengembangan Permainan

Hasil pengujian oleh ahli pengembangan menunjukkan bahwa seluruh fungsi yang ada pada permainan dapat berjalan dengan baik. Hal tersebut

ditunjukan dengan persentase aspek fungsionalitas pada kedua penguji yaitu sebesar 100%. Hasil pengujian *compatibility* menunjukan bahwa permainan dapat dijalankan pada berbagai perangkat yang berbeda ditunjukan dengan hasil pengujian dengan persentase aspek *compatibility testing* pada kedua penguji sebesar 100%. Hasil pengujian oleh ahli pengembangan permainan tersebut mendapatkan kategori Sangat Layak. Dengan demikian maka dapat disimpulkan bahwa permainan telah memenuhi kelayakan aspek fungsionalitas dan *compatibility*.

b. Pengujian oleh Ahli Media

Pengujian oleh ahli media menilai aspek *balance testing*, aspek *compliance testing*, aspek *playtesting*, dan aspek *usability testing*. Secara keseluruhan hasil pengujian oleh ahli media mendapatkan persentase kelayakan sebesar 85%. Berdasarkan persentase tersebut maka aspek penilaian oleh ahli media mendapatkan kategori Sangat Layak. Dengan demikian dapat disimpulkan bahwa permainan telah memenuhi kelayakan aspek *balance testing*, aspek *compliance testing*, aspek *playtesting*, dan aspek *usability testing*.

c. Pengujian oleh Ahli Materi

Hasil pengujian oleh ahli materi menunjukan bahwa aspek materi pada permainan sangat baik. Hal tersebut berdasarkan hasil pengujian oleh ahli materi dengan persentase kelayakan sebesar 83,93%. Nilai tersebut menunjukan bahwa aspek materi mendapatkan kategori Layak. Dengan demikian dapat disimpulkan bahwa permainan telah memenuhi kelayakan aspek materi.

d. Pengujian oleh Pengguna

Pengujian oleh pengguna menilai beberapa aspek yaitu aspek *balance testing*, aspek *playtesting*, aspek *usability testing*, dan aspek materi. Hasil pengujian oleh pengguna mendapatkan nilai yang baik ditunjukan dengan persentase kelayakan sebesar 82,05%. Berdasarkan nilai tersebut maka pengujian oleh pengguna mendapatkan kategori Layak. Berdasarkan perhitungan *Cronbach's Alpha* diperoleh nilai sebesar 0,92 dengan kategori *Excellent*. Maka dapat disimpulkan bahwa permainan yang diimplementasikan kepada pengguna telah layak digunakan.

BAB V

SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan dari hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa :

1. Penelitian ini menghasilkan sebuah produk berupa permainan dengan konsep *puzzle* sebagai media pembelajaran materi konsep dasar algoritma. Proses pengembangan permainan tersebut menggunakan metode pengembangan ADDIE yang terdiri atas 5 tahap yaitu analisis, desain, pengembangan, implementasi, dan evaluasi. Pada permainan yang dikembangkan terdapat 3 bagian utama yaitu bagian menu utama, bagian menu level, dan bagian permainan. Bagian menu utama berisi menu main, pengaturan, cara bermain, tentang permainan, profil pengembang, papan skor, dan keluar. Bagian menu level berisi mengenai pemilihan level yang terdiri dari 4 stage dengan total terdapat 27 level. Sistem scoring pada setiap level berdasarkan jumlah penggunaan blok perintah dengan ketentuan semakin sedikit semakin baik. Bagian permainan berisi karakter permainan yang harus digerakkan, peta permainan, menu menjalankan blok perintah, menu mengedit blok perintah, dan menu jeda. Konsep utama dari permainan yang dikembangkan adalah menyusun suatu perintah untuk menggerakkan karakter permainan dengan demikian maka pengguna secara sadar ataupun tidak dasar akan mendapatkan pengetahuan mengenai konsep dasar algoritma. Produk akhir permainan berupa file dengan ekstensi .exe yang dijalankan pada perangkat komputer.

2. Hasil pengujian yang dilakukan pada penelitian ini menunjukan bahwa permainan layak digunakan untuk mengajarkan materi algoritma. Hal tersebut ditunjukan dengan hasil pengujian dalam beberapa tahap. Tahap pertama adalah pengujian aspek *compatibility testing* dan uji fungsionalitas yang mendapatkan persentase kelayakan sebesar 100% (Sangat Layak). Pengujian selanjutnya pengujian aspek *balance testing*, *compliance testing*, *playtesting*, dan *usability testing* yang mendapatkan persentase kelayakan pada keseluruhan aspek sebesar 85% (Sangat Layak). Pengujian aspek materi mendapatkan nilai persentase kelayakan sebesar 83,93% (Layak). Tahap pengujian terakhir yang dilakukan adalah pengujian aspek *balance testing*, *playtesting*, *usability testing*, dan materi yang mendapatkan nilai persentase kelayakan keseluruhan sebesar 82,05% (Layak) dengan nilai konsistensi data yang dihitung menggunakan *Cronbach's Alpha* mendapatkan nilai sebesar 0,92 (*Excellent*).

B. Saran

Berdasarkan simpulan yang telah disampaikan, maka disampaikan beberapa hal sebagai berikut :

1. Perlu pengembangan terhadap permainan lebih lanjut agar cakupan materi yang disampaikan pada permainan lebih luas tidak terbatas pada konsep *flowchart*.
2. Pengembangan terhadap konsep permainan agar menggunakan jenis 3 dimensi sehingga memperluas jenis perintah dan lebih banyak teka – teki yang dapat disampaikan.

3. Pengembangan lebih lanjut mengenai perangkat yang digunakan agar tidak terbatas pada perangkat komputer tetapi dapat digunakan pada perangkat *mobile* agar lebih mudah digunakan oleh pengguna tanpa harus menggunakan perangkat komputer.
4. Pengembangan lebih lanjut pada pengalaman pengguna agar permainan lebih mudah digunakan oleh siapapun.

DAFTAR PUSTAKA

- Ananda, D., Suryan, A., Mayadewi, P., Rasiana, L., & Kusmayadi, H. (2009). *Algoritma dan Pemrograman*. Bandung: Politeknik Telkom.
- Ansari, N., Parikh, N., Narvekar, A., Phatapekar, M., & Yadav, A. (2014). GamEd – Learning Data Structure Algorithm Using Computer Game. *IOSR Journal of Computer Science*. Vol. 1 No. 10. Hlm. 62-67.
- Arsyad, A. (2017). *Media Pembelajaran Edisi Revisi*. Jakarta: Rajawali Pers.
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York: Springer.
- Clontz, S. (2018). *Puzzle Types*. Diakses dari <http://clontz.org/puzzles/types/>. pada tanggal 20 April 2018, Jam 20.00 WIB.
- Drysdale, J. (2011). Collaborative Games: An Exploratory View for Instructional Designers. *The Journal of Applied Instructional Design*. Vol. 1 No. 2. Hlm. 47-55.
- Ghavifekr, S., & Rosdy, W. A. W. (2015). Teaching and Learning with Technology: Effectiveness of ICT Integration in Schools. *International Journal of Research in Education and Science*. Vol. 1. No. 2. Hlm. 175-191.
- Gliem, J. A., & Gliem, R. R. (2003). Calculating, Interpreting, and Reporting Cronbach's Alpha Reliability Coefficient for Likert-Type Scales. *Midwest Research to Practice Conference in Adult, Continuing, and Community Education*, Hlm. 82-88..
- Gray, P. (2011). The Decline of Play and the Rise of Psychopathology in Children and Adolescents. *American Journal of Play*. Vol. 3 No. 4. Hlm. 443-463.
- Ismail, A. (2009). *Education Games*. Yogyakarta: Pro-U Media.
- Jerzak, N. & Rebelo, F. (2014). Heuristic Evaluation for Serious Games (HESG): Using HESG for Designing and Evaluating Serious Games. *Advances in Ergonomics In Design, Usability & Special Populations: Part I*. Hlm. 159-168.
- Jeuring, J., Rooij, R. V., & Pronost, N. (2013). *The 5/10 Method: A Method For Designing Educational Games*. Utrecht: Utrecht University.

- Juanda, E. A., Gunawan, T., & Mujiburrohman, D. (2012). Penerapan Pembelajaran Berbasis Games (Tic Tac Toe) dalam Upaya Meningkatkan Pemahaman Konsep pada Mata Diktat Elektronika Dasar. *Jurnal Pendidikan Teknologi dan Kejuruan*. Vol. 21 No. 2. Hlm. 123-129.
- Kapp, K. (2014). *Eight Game Elements to Make Learning More Intriguing*. Diakses dari <https://www.td.org/insights/eight-game-elements-to-make-learning-more-intriguing>. pada tanggal 17 Desember 2017, Jam 20.30 WIB.
- Kim, S. (2006). *What is a Puzzle*. Diakses dari http://cs.wellesley.edu/~cs215/Lectures/L17-IntroGamesJigsawPuzzle/ScottKim-What_is_a_Puzzle.pdf. pada tanggal 17 Desember 2017, Jam 21.00 WIB.
- Kuindersma, E., Pal, J. V. D., Herik, J. V. D., & Plaat, A. (2017). Building a Game to Build Competencies. *Games and Learning Alliance*. Hlm. 14-24.
- Kurnia, A. (2015). Pengaruh Penggunaan Media Pembelajaran Visual Terhadap Hasil Belajar Ekonomi Siswa Kelas X Semester Genap SMA Negeri 1 Pekalongan. *Prosiding Seminar Nasional Pendidikan Ekonomi dan Bisnis, tanggal 07 November 2015*. Surakarta: FKIP UNS.
- Levitin, A. & Papalaskari, M. A. (2002). Using Puzzles in Teaching Algorithms. *SIGCSE, Technical Symposium on Computer Science Education*. Kentucky: ACM.
- Levy, L. & Novak, J. (2010). *Game Development Essentials: Game Testing & QA*. Clifton Park, Delmar: Cengage Learning.
- Maryono, D. & Pambudhi, D. (2014). *Pemrograman Dasar Kelas X Semester 1 Untuk SMK/MAK*. Jakarta: Kementerian Pendidikan dan Kebudayaan Republik Indonesia.
- McGriff, S. J. (2000). *Instructional System Design (ISD): Using the ADDIE Model*. Amerika: Penn State University.
- Microsoft. (2013). *Archive: Certification Requirements for Windows Desktop Apps v3.1*. Diakses dari [https://msdn.microsoft.com/en-us/library/windows/desktop/mt674710\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/mt674710(v=vs.85).aspx). pada tanggal 28 Januari 2018, Jam 10.00 WIB.
- Munir, R. (2011). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.

- Oxford University Press. (2017). *English Oxford Living Dictionaries*. Diakses dari <https://en.oxforddictionaries.com/definition/puzzle>. pada tanggal 17 Desember 2017, Jam 20.00 WIB.
- Pressman, R. S. & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill Education.
- Prosser, C. A. & Quigley, T. H. (1949). *Vocational Education in a Democracy*. Chichago: American Technical Society.
- Putra, N. (2015). *Research and Development Penelitian dan Pengembangan: Suatu Pengantar*. Jakarta: Rajawali Pers.
- Ramadhina, S. (2015). Pembuatan Sistem Informasi Manajemen Bengkel di Sekolah Menengah Kejuruan Negeri 3 Yogyakarta. *Jurnal Pendidikan Teknologi dan Kejuruan*. Vol. 22 No. 3. Hlm. 324-338.
- Rollings, A. & Adams, E. (2003). *Andrew Rollings and Renest Adams on Game Design*. Indiana: New Riders Publishing.
- Rosa, A. S. & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Penerbit Informatika.
- Shabanah, S. S. (2010). *Simplifying Algorithm Learning Using Serious Games. Magister dissertation*. George Mason University. UMI 3407063. Virginia: ProQuest LLC.
- Sudira, P. (2016). *TVET Abad XXI Filosofi, Teori, Konsep, dan Strategi Pembelajaran Vokasional*. Yogyakarta: UNY Press.
- Sugiyono. (2017). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Tim Penyusun KBBI. (2016). *KBBI Online Edisi Kelima*. Diakses dari <https://kbbi.kemdikbud.go.id>. pada tanggal 17 Desember 2017, Jam 20.00 WIB.
- Vanderschuren, L. J. M. J. (2010). How the Brain Makes Play Fun. *American Journal of Play*. Vol 2. No. 3. Hlm. 315-337.
- Wati, E. R. (2016). *Ragam Media Pembelajaran*. Jakarta: Kata Pena.
- Yunanto, S. H., Chandrawati, T. B. (2016). "Smart Puzzle" Game Helping Children Learn to Read. *Sisforma*. Vol. 3 No. 1. Hlm 13-19.

LAMPIRAN

Lampiran 1. Surat Keputusan Pembimbing Skripsi

**KEPUTUSAN DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 70/PINF/PB/IV/2018**

**TENTANG
PENGANGKATAN DOSEN PEMBIMBING TUGAS AKHIR SKRIPSI (TAS) MAHASISWA
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

DEKAN FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

- Menimbang : a. bahwa untuk kelancaran pelaksanaan kegiatan Tugas Akhir Skripsi (TAS) mahasiswa, dipandang perlu mengangkat dosen pembimbingnya;
b. bahwa untuk keperluan sebagaimana dimaksud pada huruf a perlu menetapkan Keputusan Dekan Tentang Pengangkatan Dosen Pembimbing Tugas Akhir Skripsi (TAS) Fakultas Teknik Universitas Negeri Yogyakarta.
- Mengingat : 1. Undang-undang RI Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional (Lembaran Negara Tahun 2003 Nomor 78, Tambahan Lembaran Negara Nomor 4301);
2. Peraturan Pemerintah Republik Indonesia Nomor 4 Tahun 2014 Tentang Penyelenggaraan Pendidikan Tinggi dan Pengelolaan Perguruan Tinggi (Lembaran Negara Tahun 2014 Nomor 16, Tambahan Lembaran Negara Republik Indonesia Nomor 5500);
3. Keputusan Presiden Republik Indonesia Nomor 93 Tahun 1999 Tentang Perubahan Institut Keguruan dan Ilmu Pendidikan menjadi Universitas;
4. Peraturan Mendiknas RI Nomor 23 Tahun 2011 Tentang Organisasi dan Tata Kerja Universitas Negeri Yogyakarta;
5. Peraturan Mendiknas RI Nomor 34 Tahun 2011 Tentang Statuta Universitas Negeri Yogyakarta;
6. Keputusan Menteri Pendidikan dan Kebudayaan RI Nomor 98/MPK.A4/KP/2013 Tentang Pengangkatan Rektor Universitas Negeri Yogyakarta;
7. Peraturan Rektor Nomor 2 Tahun 2014 tentang Peraturan Akademik;
8. Keputusan Rektor Nomor 800/UN.34/KP/2016 tahun 2016 tentang Pengangkatan Dekan Fakultas Teknik Universitas Negeri Yogyakarta.

MEMUTUSKAN

Menetapkan : **KEPUTUSAN DEKAN TENTANG PENGANGKATAN DOSEN PEMBIMBING
TUGAS AKHIR SKRIPSI (TAS) FAKULTAS TEKNIK UNIVERSITAS NEGERI
YOGYAKARTA.**

PERTAMA : Mengangkat Saudara :

Nama	: Dr. Fatchul Arifin, M.T.
NIP	: 19720508 199802 1 002
Pangkat/Golongan	: Penata, III/c
Jabatan Akademik	: Lektor Kepala

sebagai Dosen Pembimbing Untuk mahasiswa penyusun Tugas Akhir Skripsi (TAS) :

Nama	: Permadi Afrian
NIM	: 14520241050
Prodi Studi	: Pend. Teknik Informatika - S1
Judul Skripsi/TA	: PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN DASAR ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR DI SMK NEGERI 1 MAGELANG

- KEDUA : Dosen Pembimbing sebagaimana dimaksud dalam Diktum PERTAMA bertugas merencanakan, mempersiapkan, melaksanakan, dan mempertanggungjawabkan pelaksanaan kegiatan bimbingan terhadap mahasiswa sebagaimana dimaksud dalam Diktum PERTAMA sampai mahasiswa dimaksud dinyatakan lulus.
- KETIGA : Biaya yang diperlukan dengan adanya Keputusan ini dibebankan pada Anggaran DIPA Universitas Negeri Yogyakarta Tahun 2018.
- KEEMPAT : Keputusan ini berlaku sejak tanggal 2 April 2018.

Tembusan Keputusan Dekan ini disampaikan kepada :

1. Para Wakil Dekan Fakultas Teknik;
2. Kepala Bagian Tata Usaha Fakultas Teknik;
3. Kepala Subbagian Keuangan dan Akuntansi Fakultas Teknik;
4. Kepala Subbagian Pendidikan Fakultas Teknik;
5. Mahasiswa yang bersangkutan;

Universitas Negeri Yogyakarta.

Ditetapkan di : Yogyakarta
Pada tanggal : 2 April 2018

DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA,



Lampiran 2. Surat Ijin Penelitian



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI UNIVERSITAS NEGERI YOGYAKARTA FAKULTAS TEKNIK

Alamat : Kampus Karangmalang, Yogyakarta, 55281
Telp. (0274) 586168 psw. 276,289,292 (0274) 586734 Fax. (0274) 586734
Laman: ft.uny.ac.id E-mail: ft@uny.ac.id, teknik@uny.ac.id

Nomor : 362/UN34.15/LT/2018

2 Mei 2018

Lamp. : 1 Bendel Proposal

Hal : Izin Penelitian

- Yth .
1. Gubernur DIY c.q. Kepala Badan Kesatuan Bangsa dan Politik DIY
 2. Kepala Badan Kesatuan Bangsa dan Politik Kota Magelang
 3. Kepala Sekolah SMK Negeri 1 Magelang

Kami sampaikan dengan hormat, bahwa mahasiswa tersebut di bawah ini:

Nama : Permadi Afrian
NIM : 14520241050
Program Studi : Pend. Teknik Informatika - S1
Judul Tugas Akhir : Pengembangan Permainan Puzzle sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang
Tujuan : Memohon izin mencari data untuk penulisan Tugas Akhir Skripsi (TAS)
Waktu Penelitian : 14 Mei - 8 Juni 2018

Untuk dapat terlaksananya maksud tersebut, kami mohon dengan hormat Bapak/Ibu berkenan memberi izin dan bantuan seperlunya.

Demikian atas perhatian dan kerjasamanya kami sampaikan terima kasih.



Tembusan :

1. Sub. Bagian Pendidikan dan Kemahasiswaan ;
2. Mahasiswa yang bersangkutan.



PEMERINTAH DAERAH DAERAH ISTIMEWA YOGYAKARTA
BADAN KESATUAN BANGSA DAN POLITIK
Jl. Jenderal Sudirman No 5 Yogyakarta – 55233
Telepon : (0274) 551136, 551275, Fax (0274) 551137

Yogyakarta, 3 Mei 2018

Kepada Yth. :

Nomor : 074/5710/Kesbangpol/2018
Perihal : Rekomendasi Penelitian

Gubernur Jawa Tengah
Up. Kepala Dinas Peranaman Model dan
Pelayanan Terpadu Satu Pintu Provinsi Jawa
Tengah

di Semarang

Memperhatikan surat :

Dari : Dekan Fakultas Teknik Universitas Negeri Yogyakarta
Nomor : 362/UN34.15/LT/2018
Tanggal : 2 Mei 2018
Perihal : Izin Penelitian

Setelah mempelajari surat permohonan dan proposal yang diajukan, maka dapat diberikan surat rekomendasi tidak keberatan untuk melaksanakan riset/penelitian dalam rangka penyusunan skripsi dengan judul proposal **"PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK NEGERI 1 MAGELANG"** kepada:

Nama : PERMADI AFRIAN
NIM : 14520241050
No.HP/Identitas : 085786517306/3323123004960001
Prodi/Jurusan : Pendidikan Teknik Informatika
Fakultas : Fakultas Teknik Universitas Negeri Yogyakarta
Lokasi Penelitian : SMK Negeri 1 Magelang
Waktu Penelitian : 14 Mei 2018 s.d 8 Juni 2018
Sehubungan dengan maksud tersebut, diharapkan agar pihak yang terkait dapat memberikan bantuan / fasilitas yang dibutuhkan.

Kepada yang bersangkutan diwajibkan:

1. Menghormati dan mentaati peraturan dan tata tertib yang berlaku di wilayah riset/penelitian;
2. Tidak dibenarkan melakukan riset/penelitian yang tidak sesuai atau tidak ada kaitannya dengan judul riset/penelitian dimaksud;
3. Menyerahkan hasil riset/penelitian kepada Badan Kesbangpol DIY selambat-lambatnya 6 bulan setelah penelitian dilaksanakan.
4. Surat rekomendasi ini dapat diperpanjang maksimal 2 (dua) kali dengan menunjukkan surat rekomendasi sebelumnya, paling lambat 7 (tujuh) hari kerja sebelum berakhirnya surat rekomendasi ini.

Rekomendasi Ijin Riset/Penelitian ini dinyatakan tidak berlaku, apabila ternyata pemegang tidak mentaati ketentuan tersebut di atas.

Demikian untuk menjadikan maklum.



Tembusan disampaikan Kepada Yth. :

1. Gubernur DIY (sebagai laporan)
2. Dekan Fakultas Teknik Universitas Negeri Yogyakarta;
3. Yang bersangkutan.



**PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENANAMAN MODAL
DAN PELAYANAN TERPADU SATU PINTU**

Jalan Mgr. Sugiyopranoto Nomor 1 Semarang Kode Pos 50131 Telepon : 024 – 3547091, 3547438,
3541487 Faksimile 024-3549560 Laman <http://dpmpfsp.jatengprov.go.id> Surat Elektronik
dpmpfsp@jatengprov.go.id

REKOMENDASI PENELITIAN

NOMOR : 070/6351/04.5/2018

- Dasar** : 1. Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 07 Tahun 2014 tentang Perubahan atas Peraturan Menteri Dalam Negeri Republik Indonesia Nomor 64 Tahun 2011 tentang Pedoman Penerbitan Rekomendasi Penelitian ;
2. Peraturan Gubernur Jawa Tengah Nomor 72 Tahun 2016 tentang Organisasi dan Tata Kerja Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Provinsi Jawa Tengah ;
3. Peraturan Gubernur Jawa Tengah Nomor 18 Tahun 2017 tentang Penyelenggaraan Pelayanan Terpadu Satu Pintu di Provinsi Jawa Tengah.
- Memperhatikan** : Surat Kepala Badan Kesatuan Bangsa Dan Politik Daerah Istimewa Yogyakarta Nomor : 074/5710/Kesbangpol/2018 Tanggal : 3 Mei 2018 Perihal : Rekomendasi Penelitian

Kepala Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Provinsi Jawa Tengah, memberikan rekomendasi kepada :

1. Nama : PERMADI AFRIAN
2. Alamat : Tuksongo RT. 01 RW. 01 Desa Nglorog, Kecamatan Pringsurat, Kabupaten Temanggung, Provinsi Jawa Tengah
3. Pekerjaan : Mahasiswa

Untuk : Melakukan Penelitian dengan rincian sebagai berikut :

- a. Judul Proposal : PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK NEGERI 1 MAGELANG
b. Tempat / Lokasi : SMK Negeri 1 Magelang
c. Bidang Penelitian : Teknik
d. Waktu Penelitian : 14 Mei 2018 sampai 08 Juni 2018
e. Penanggung Jawab : DR. Fatchul Arifin, S.T., M.T.
f. Status Penelitian : Baru
g. Anggota Peneliti : -
h. Nama Lembaga : Universitas Negeri Yogyakarta

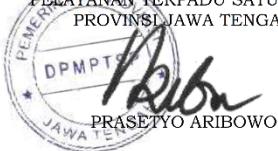
Ketentuan yang harus ditaati adalah :

- a. Sebelum melakukan kegiatan terlebih dahulu melaporkan kepada Pejabat setempat / Lembaga swasta yang akan dijadikan obyek lokasi;
b. Pelaksanaan kegiatan dimaksud tidak disalahgunakan untuk tujuan tertentu yang dapat mengganggu kestabilan pemerintahan;
c. Setelah pelaksanaan kegiatan dimaksud selesai supaya menyerahkan hasilnya kepada Kepala Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu Provinsi Jawa Tengah;
d. Apabila masa berlaku Surat Rekomendasi ini sudah berakhir, sedang pelaksanaan kegiatan belum selesai, perpanjangan waktu harus diajukan kepada instansi pemohon dengan menyertakan hasil penelitian sebelumnya;
e. Surat rekomendasi ini dapat diubah apabila di kemudian hari terdapat kekeliruan dan akan diadakan perbaikan sebagaimana mestinya.

Demikian rekomendasi ini dibuat untuk dipergunakan seperlunya.

Semarang, 08 Mei 2018

KEPALA DINAS PENANAMAN MODAL DAN
PELAYANAN TERPADU SATU PINTU
PROVINSI JAWA TENGAH





**PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENANAMAN MODAL
DAN PELAYANAN TERPADU SATU PINTU**

Jalan Mgr. Sugiyopranoto Nomor 1 Semarang Kode Pos 50131 Telepon : 024 – 3547091, 3547438,
3541487 Faksimile 024-3549560 Laman <http://dpmpfsp.jatengprov.go.id> Surat Elektronik
dpmpfsp@jatengprov.go.id

Nomor : 070/5095/2018
Sifat : Biasa
Lampiran : 1 (Satu) Berkas
Perihal : Rekomendasi Penelitian

Semarang, 08 Mei 2018

Yth. Kepada
Kepala Dinas Pendidikan dan
Kebudayaan Provinsi Jawa Tengah
Di Semarang

Dalam rangka memperlancar pelaksanaan kegiatan penelitian bersama ini terlampir disampaikan Penelitian Nomor 070/6351/04.5/2018 Tanggal 08 Mei 2018 atas nama PERMADI AFRIAN dengan judul proposal PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK NEGERI 1 MAGELANG, untuk dapat ditindaklanjuti.

Demikian untuk menjadi maklum dan terimakasih.

KEPALA DINAS PENANAMAN MODAL DAN
PELAYANAN TERPADU SATU PINTU
PROVINSI JAWA TENGAH



Dr. PRASETYO ARIBOWO, SH, Msoc, SC.
Pembina Utama Madya
NIP.19611115 198603 1 010

Tembusan :

1. Gubernur Jawa Tengah;
2. Sekretaris Daerah Provinsi Jawa Tengah;
3. Kepala Badan Kesbangpol Provinsi Jawa Tengah;
4. Kepala Badan Kesatuan Bangsa Dan Politik Daerah Istimewa Yogyakarta;
5. Sdr. PERMADI AFRIAN

Lampiran 3. Surat Keterangan Telah Melaksanakan Penelitian



**PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENDIDIKAN DAN KEBUDAYAAN
SEKOLAH MENENGAH KEJURUAN NEGERI 1 MAGELANG**
Jalan Cawang Nomor 2 Kota Magelang Kode Pos 56123 Telepon 0293-362172-365543
Faksimile 0293-368821 Surat Elektronik smkn1magelang@yahoo.com

SURAT KETERANGAN
Nomor : 421-S / 511 / 230.SMK.01

Menindak lanjuti surat dari Universitas Negeri Yogyakarta Fakultas Teknik nomor 362/UN34.15/LT/2018 tanggal 2 Mei 2018 perihal Izin Penelitian maka yang bertanda tangan dibawah ini :

Nama	:	Drs. Nisandi, M.T
NIP	:	19600814 198803 1 009
Pangkat/gol. ruang	:	Pembina Tk. I IV/b
Jabatan	:	Kepala Sekolah
Unit Kerja	:	SMK Negeri 1 Magelang

Menerangkan dengan sesungguhnya bahwa :

Nama	:	Permadi Afrian
NIM	:	14520241050
Program Studi	:	Pend. Teknik Informatika - S1

Adalah benar-benar telah melaksanakan penelitian di SMK Negeri 1 Magelang pada tanggal 14 Mei s.d. 8 Juli 2018 dengan judul Penelitian "Pengembangan Permainan Puzzle sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang".

Demikian surat keterangan ini dibuat dengan sesungguhnya untuk dapat dipergunakan sebagaimana mestinya.

Magelang, 28 Mei 2018
Kepala SMK Negeri 1 Magelang

Drs. Nisandi, M.T
Pembina Tk. I
NIP. 19600814 198803 1 009



Lampiran 4. Validasi Instrumen

Hal : Permohonan Validasi Instrumen TAS
Lampiran : 1 Bendel

Kepada Yth,
Dr. Putu Sudira, M.P.
Dosen Jurusan Pendidikan Teknik Elektronika dan Informatika
Di Fakultas Teknik UNY

Sehubungan dengan rencana pelaksanaan Tugas Akhir Skripsi (TAS), dengan ini saya:

Nama : Permadi Afrian
NIM : 14520241050
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Pengembangan Permainan *Puzzle* sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang

dengan hormat mohon Bapak/Ibu berkenan memberikan validasi terhadap instrumen penelitian TAS yang telah saya susun. Sebagai bahan pertimbangan, bersama ini saya lampirkan : (1) proposal TAS, (2) kisi-kisi instrumen penelitian TAS, dan (3) draf instrumen penelitian TAS.

Demikian permohonan saya, atas bantuan dan perhatian Bapak/Ibu diucapkan terimakasih.

Yogyakarta, 21 Mei 2018

Pemohon,



Permadi Afrian

NIM. 14520241050

Kaprodi Pend. Teknik Informatika



Handaru Jati, S.T., M.M., M.T., Ph.D.
NIP. 19740511 199903 1 002

Pembimbing TAS,



Dr. Fatchul Arifin, S.T., M.T.
NIP. 19720508 199802 1 002

SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI

Saya yang bertanda tangan di bawah ini :

Nama : Dr. Putu Sudira, M.P.
NIP : 19641021 198702 1 063
Jurusan : Pendidikan Teknik Elektronika dan Informatika

Menyatakan

Nama : Permadi Afrian
NIM : 14520241050
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Pengembangan Permainan *Puzzle* sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan :

- Layak digunakan untuk penelitian
 Layak digunakan dengan perbaikan
 Tidak layak digunakan untuk penelitian yang bersangkutan

Dengan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta, 23/1/2018
Validator,



Dr. Putu Sudira, M.P.
NIP. 19641021 198702 1 063

Catatan :

- beri tanda √

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Permadi Afrian **NIM.** 14520241050
Judul TAS : Pengembangan Permainan *Puzzle* sebagai Media Pembelajaran Algoritma pada Mata Pelajaran Pemrograman Dasar Kelas X di SMK Negeri 1 Magelang

No.	Instrumen	Saran/Tanggapan
1.		pilihan ya - tidak dilanjutkan
2.		pilihan sih di gunakan
3.		
	Komentar umum/Lain-lain:	

Yogyakarta,
 Validator,



Dr. Putu Sudira, M.P.

NIP. 19641021 198702 1 063

Lampiran 5. Hasil Pengujian Ahli Pengembangan Permainan

LEMBAR EVALUASI OLEH PENGEMBANG
PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG

A. Data Responden

Nama : Fikry Abdullah Ariz
Bidang Keahlian : Game Development

B. Keterangan

1. Lembar evaluasi ini digunakan untuk menguji kesiapan produk sebelum dilakukan pengujian di luar lingkungan pengembang.
2. Lembar evaluasi ini diisi oleh Pengembang.
3. Jawaban diberikan pada skala penilaian **Ya** dan **Tidak**.
4. Berilah tanda (✓) pada kolom yang sesuai dengan hasil pengujian produk terhadap kriteria yang ditujukan.
5. Kuesioner berupa pengujian *compatibility* dan *functionality* pada produk untuk memastikan kinerja produk.

Compatibility Testing

No.	Pernyataan	Jawaban	
		Ya	Tidak
1	Aplikasi berjalan pada layar dengan rasio 4:3	✓	
2	Aplikasi berjalan pada layar dengan rasio 16:10	✓	
3	Aplikasi berjalan pada layar dengan rasio 16:9	✓	
4	Aplikasi berjalan pada sistem operasi Windows 7	✓	
5	Aplikasi berjalan pada sistem operasi Windows 8	✓	
6	Aplikasi berjalan pada sistem operasi Windows 10	✓	
7	Suara aplikasi terdengar jelas pada perangkat audio internal	✓	
8	Suara aplikasi terdengar jelas pada perangkat audio eksternal	✓	
9	Suara aplikasi terdengar jelas dengan headphone atau headset	✓	
10	Touchpad dapat digunakan dengan baik pada aplikasi	✓	
11	Mouse dapat digunakan dengan baik pada aplikasi	✓	
12	Keyboard dapat digunakan dengan baik pada aplikasi	✓	

Functionality Testing

No.	Pernyataan	Berfungsi	
		Ya	Tidak
1	Tombol "Main"	✓	
2	Tombol "Pengaturan"	✓	
3	Tombol "Profil Pengembang"	✓	
4	Tombol "Cara Bermain"	✓	
5	Tombol "Tentang Permainan"	✓	
6	Tombol "Papan Skor"	✓	
7	Tombol "Keluar"	✓	
8	Tombol untuk menutup panel pengaturan	✓	
9	Pengaturan efek suara	✓	
10	Pengaturan musik	✓	
11	Pengaturan kualitas gambar	✓	
12	Pengaturan ganti nama	✓	
13	Tombol untuk menutup panel profil pengembang	✓	
14	Tombol untuk menutup panel tentang permainan	✓	
15	Tombol untuk menutup panel cara bermain	✓	
16	Tombol "Selanjutnya" pada cara bermain	✓	
17	Tombol "Sebelumnya" pada cara bermain	✓	
18	Tombol untuk menutup papan skor	✓	
19	Tombol "Selanjutnya" pada papan skor	✓	
20	Tombol "Sebelumnya" pada papan skor	✓	
21	Tombol "Ya" pada panel dialog tutup aplikasi	✓	
22	Tombol "Tidak" pada panel dialog tutup aplikasi	✓	
23	Tombol "Kembali"	✓	
24	Tombol "Selanjutnya"	✓	
25	Tombol "Sebelumnya"	✓	
26	Tombol pilih level	✓	
27	Tombol untuk menutup panel pilih level	✓	
28	Tombol level	✓	
29	Tombol "Play"	✓	
30	Tombol "Pause"	✓	
31	Tombol "Restart"	✓	

No.	Pernyataan	Berfungsi	
		Ya	Tidak
32	Tombol buka panel algoritma	✓	
33	Tombol tutup panel algoritma	✓	
34	Tombol tambahkan blok algoritma	✓	
35	Geser blok algoritma	✓	
36	Hubungkan blok algoritma	✓	
37	Tombol level selanjutnya pada panel akhir	✓	
38	Tombol menu utama pada panel akhir	✓	
39	Tombol ulangi pada panel akhir	✓	
40	Tombol "Lanjutkan" pada panel jeda	✓	
41	Tombol "Menu utama" pada panel jeda	✓	
42	Tombol "Keluar" pada panel jeda	✓	

Kritik/Saran :

UX UI futur sendela Algoritma bisa diperbaiki

.....

.....

.....

Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
- Layak digunakan untuk penelitian dengan perbaikan
- Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Yogyakarta,

Pengaji,

(...Finay Abdullah Arif.....)

LEMBAR EVALUASI OLEH PENGEMBANG
PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG

A. Data Responden

Nama : Muh. Arifin.....

Bidang Keahlian : Guru Dev......

B. Keterangan

1. Lembar evaluasi ini digunakan untuk menguji kesiapan produk sebelum dilakukan pengujian di luar lingkungan pengembang.
2. Lembar evaluasi ini diisi oleh Pengembang.
3. Jawaban diberikan pada skala penilaian **Ya** dan **Tidak**.
4. Berilah tanda (✓) pada kolom yang sesuai dengan hasil pengujian produk terhadap kriteria yang diujikan.
5. Kuesioner berupa pengujian *compatibility* dan *functionality* pada produk untuk memastikan kinerja produk.

Compatibility Testing

No.	Pernyataan	Jawaban	
		Ya	Tidak
1	Aplikasi berjalan pada layar dengan rasio 4:3	✓	
2	Aplikasi berjalan pada layar dengan rasio 16:10	✓	
3	Aplikasi berjalan pada layar dengan rasio 16:9	✓	
4	Aplikasi berjalan pada sistem operasi Windows 7	✓	
5	Aplikasi berjalan pada sistem operasi Windows 8	✓	
6	Aplikasi berjalan pada sistem operasi Windows 10	✓	
7	Suara aplikasi terdengar jelas pada perangkat audio internal	✓	
8	Suara aplikasi terdengar jelas pada perangkat audio eksternal	✓	
9	Suara aplikasi terdengar jelas dengan headphone atau headset	✓	
10	Touchpad dapat digunakan dengan baik pada aplikasi	✓	
11	Mouse dapat digunakan dengan baik pada aplikasi	✓	
12	Keyboard dapat digunakan dengan baik pada aplikasi	✓	

Functionality Testing

No.	Pernyataan	Berfungsi	
		Ya	Tidak
1	Tombol "Main"	✓	
2	Tombol "Pengaturan"	✓	
3	Tombol "Profil Pengembang"	✓	
4	Tombol "Cara Bermain"	✓	
5	Tombol "Tentang Permainan"	✓	
6	Tombol "Papan Skor"	✓	
7	Tombol "Keluar"	✓	
8	Tombol untuk menutup panel pengaturan	✓	
9	Pengaturan efek suara	✓	
10	Pengaturan musik	✓	
11	Pengaturan kualitas gambar	✓	
12	Pengaturan ganti nama	✓	
13	Tombol untuk menutup panel profil pengembang	✓	
14	Tombol untuk menutup panel tentang permainan	✓	
15	Tombol untuk menutup panel cara bermain	✓	
16	Tombol "Selanjutnya" pada cara bermain	✓	
17	Tombol "Sebelumnya" pada cara bermain	✓	
18	Tombol untuk menutup papan skor	✓	
19	Tombol "Selanjutnya" pada papan skor	✓	
20	Tombol "Sebelumnya" pada papan skor	✓	
21	Tombol "Ya" pada panel dialog tutup aplikasi	✓	
22	Tombol "Tidak" pada panel dialog tutup aplikasi	✓	
23	Tombol "Kembali"	✓	
24	Tombol "Selanjutnya"	✓	
25	Tombol "Sebelumnya"	✓	
26	Tombol pilih level	✓	
27	Tombol untuk menutup panel pilih level	✓	
28	Tombol level	✓	
29	Tombol "Play"	✓	
30	Tombol "Pause"	✓	
31	Tombol "Restart"	✓	

No.	Pernyataan	Berfungsi	
		Ya	Tidak
32	Tombol buka panel algoritma	✓	
33	Tombol tutup panel algoritma	✓	
34	Tombol tambahkan blok algoritma	✓	
35	Geser blok algoritma	✓	
36	Hubungkan blok algoritma	✓	
37	Tombol level selanjutnya pada panel akhir	✓	
38	Tombol menu utama pada panel akhir	✓	
39	Tombol ulangi pada panel akhir	✓	
40	Tombol "Lanjutkan" pada panel jeda	✓	
41	Tombol "Menu utama" pada panel jeda	✓	
42	Tombol "Keluar" pada panel jeda	✓	

Kritik/Saran :

- UI UX perlu diperbaiki, mengingat tata letak tombol masih kurang estetis dan jelas.
 - Gameplay kurang fleksibel untuk orang yg baru mengenal game ini.....

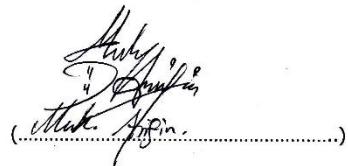
Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
- Layak digunakan untuk penelitian dengan perbaikan
- Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Yogyakarta,

Penguji,



(Mukhlis Hafidz)

Lampiran 6. Hasil Pengujian Ahli Media

**LEMBAR EVALUASI AHLI MEDIA
PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG**

Materi Pokok : Konsep Dasar Algoritma
Peneliti : Permadi Afrian
Evaluator : Sigit Pambudi, S. Pd., M. Eng.
Pekerjaan : Dosen

Petunjuk

1. Lembar validasi ini digunakan untuk menilai kualitas media permainan *puzzle* yang dikembangkan sebagai suatu media pembelajaran untuk materi algoritma.
2. Lembar validasi ini diisi oleh Ahli Media.
3. Jawaban diberikan pada skala penilaian yang sudah disediakan dengan panduan sebagai berikut ini.

SS = Sangat Setuju

S = Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

4. Berilah tanda (✓) pada kolom yang sesuai dengan pendapat ahli media terhadap setiap pernyataan.
5. Terimakasih atas kesediaan Bapak/Ibu untuk mengisi lembar validasi ini.

Balance Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dapat dimainkan dari awal hingga akhir			✓	
2	Tingkat kesulitan permainan meningkat pada setiap level				✓
3	Pemain dapat menalar rintangan yang diberikan permainan			✓	
4	Pemain memiliki kesempatan untuk menyelesaikan setiap level			✓	
5	Permainan tidak menyebabkan pemain terjebak pada fase tertentu				✓
6	Terdapat kesalahan sistem permainan yang menyebabkan pemain selalu dalam kondisi kalah		✓		
7	Pemain menentukan sendiri dalam melakukan sesuatu pada permainan				✓
8	Terdapat fitur untuk mengakhiri permainan			✓	
9	Tingkat kesulitan permainan tidak logis	✓			
10	Permainan terasa semakin sulit pada setiap peningkatan level				✓
11	Pemain menentukan bagaimana permainan akan berjalan				✓

Compliance Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dijalankan secara mandiri				✓
2	Permainan mampu berjalan pada sistem operasi Windows 8.1 dan dibawahnya				✓
3	Permainan dapat menjadi komponen klien pada aplikasi Windows Server			✓	
4	Permainan memiliki fitur yang lengkap			✓	
5	Permainan tidak melakukan kontak langsung dengan Windows Store Apps melalui mekanisme lokal			✓	
6	Sistem permainan tidak membahayakan kinerja sistem Windows			✓	

No	Pernyataan	Jawaban			
		STS	TS	S	SS
7	Permainan memiliki nama yang berbeda dengan aplikasi lain			✓	

Playtesting

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Kendali pada permainan terasa membingungkan			✓	
2	Permainan mengambil konsep sederhana dari kehidupan sehari - hari			✓	
3	Konsep puzzle membuat pemain tertantang untuk dapat memecahkan teka - teki pada permainan				✓
4	Permainan memberikan hiburan yang berbeda sehingga terasa menyenangkan				✓
5	Kesulitan yang dihadapi menarik pemain untuk menyelesaikan permainan				✓
6	Mekanisme kendali permainan dapat dipahami dengan mudah				✓
7	Permainan memiliki hubungan yang sejajar dengan kehidupan sehari - hari			✓	
8	Konsep dan karakter permainan yang ada membuat pemain merasa senang			✓	

Usability Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan tidak mengalami <i>crash</i> sistem				✓
2	Pemain merasakan umpan balik dari aksi yang dilakukan terhadap permainan				✓
3	Warna yang digunakan dapat terlihat dengan jelas				✓
4	Petunjuk cara bermain dapat dipahami dengan mudah				✓
5	Permainan memiliki tampilan yang konsisten				✓
6	Permainan menyediakan petunjuk cara bermain			✓	

No	Pernyataan	Jawaban			
		STS	TS	S	SS
7	Menu permainan tersusun dengan rapi			✓	
8	Tidak terdapat kesalahan sistem pada permainan			✓	
9	Umpan balik yang diberikan permainan sesuai dengan aksi yang dilakukan pemain			✓	

Kritik/Saran :

- Navigasi masih ambigu
 - Reward kurang menarik
 - Petunjuk perlu diperbaiki lagi
-
-
-

Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
 Layak digunakan untuk penelitian dengan perbaikan
 Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Yogyakarta,

Ahli Media,


(Sigit Pamungkas, S.Pd., M.Eng....)

LEMBAR EVALUASI AHLI MEDIA
PENGEMBANGAN PERMAINAN *PUZZLE* SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG

Materi Pokok : Konsep Dasar Algoritma
Peneliti : Permadi Afrian
Evaluator : Ponco Wali Pranoto, M. Pd.
Pekerjaan : Dosen

Petunjuk

1. Lembar validasi ini digunakan untuk menilai kualitas media permainan *puzzle* yang dikembangkan sebagai suatu media pembelajaran untuk materi algoritma.
2. Lembar validasi ini diisi oleh Ahli Media.
3. Jawaban diberikan pada skala penilaian yang sudah disediakan dengan panduan sebagai berikut ini.

SS = Sangat Setuju

S = Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

4. Berilah tanda (✓) pada kolom yang sesuai dengan pendapat ahli media terhadap setiap pernyataan.
5. Terimakasih atas kesediaan Bapak/Ibu untuk mengisi lembar validasi ini.

Balance Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dapat dimainkan dari awal hingga akhir				✓
2	Tingkat kesulitan permainan meningkat pada setiap level				✓
3	Pemain dapat menalar rintangan yang diberikan permainan			✓	
4	Pemain memiliki kesempatan untuk menyelesaikan setiap level			✓	
5	Permainan tidak menyebabkan pemain terjebak pada fase tertentu			✓	
6	Terdapat kesalahan sistem permainan yang menyebabkan pemain selalu dalam kondisi kalah		✓		
7	Pemain menentukan sendiri dalam melakukan sesuatu pada permainan			✓	
8	Terdapat fitur untuk mengakhiri permainan				✓
9	Tingkat kesulitan permainan tidak logis	✓			
10	Permainan terasa semakin sulit pada setiap peningkatan level			✓	
11	Pemain menentukan bagaimana permainan akan berjalan			✓	

Compliance Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dijalankan secara mandiri			✓	
2	Permainan mampu berjalan pada sistem operasi Windows 8.1 dan dibawahnya				✓
3	Permainan dapat menjadi komponen klien pada aplikasi Windows Server				✓
4	Permainan memiliki fitur yang lengkap			✓	
5	Permainan tidak melakukan kontak langsung dengan Windows Store Apps melalui mekanisme lokal			✓	
6	Sistem permainan tidak membahayakan kinerja sistem Windows				✓

No	Pernyataan	Jawaban			
		STS	TS	S	SS
7	Permainan memiliki nama yang berbeda dengan aplikasi lain				✓

Playtesting

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Kendali pada permainan terasa membingungkan		✓		
2	Permainan mengambil konsep sederhana dari kehidupan sehari - hari			✓	
3	Konsep puzzle membuat pemain tertantang untuk dapat memecahkan teka - teki pada permainan			✓	
4	Permainan memberikan hiburan yang berbeda sehingga terasa menyenangkan			✓	
5	Kesulitan yang dihadapi menarik pemain untuk menyelesaikan permainan			✓	
6	Mekanisme kendali permainan dapat dipahami dengan mudah				✓
7	Permainan memiliki hubungan yang sejajar dengan kehidupan sehari - hari			✓	
8	Konsep dan karakter permainan yang ada membuat pemain merasa senang				✓

Usability Testing

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan tidak mengalami <i>crash</i> sistem				✓
2	Pemain merasakan umpan balik dari aksi yang dilakukan terhadap permainan			✓	
3	Warna yang digunakan dapat terlihat dengan jelas			✓	
4	Petunjuk cara bermain dapat dipahami dengan mudah			✓	
5	Permainan memiliki tampilan yang konsisten				✓
6	Permainan menyediakan petunjuk cara bermain				✓

No	Pernyataan	Jawaban			
		STS	TS	S	SS
7	Menu permainan tersusun dengan rapi			✓	
8	Tidak terdapat kesalahan sistem pada permainan			✓	
9	Umpan balik yang diberikan permainan sesuai dengan aksi yang dilakukan pemain			✓	

Kritik/Saran :

Tambahkan umpan balik yg kesalahan spesifik dari setiap level.

.....

.....

.....

.....

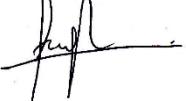
Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
- Layak digunakan untuk penelitian dengan perbaikan
- Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Yogyakarta,

Ahi Media,



(..... Purwo Wp.

Lampiran 7. Hasil Pengujian Ahli Materi

LEMBAR EVALUASI AHLI MATERI
PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG

Materi Pokok : Konsep Dasar Algoritma
Peneliti : Permadi Afrian
Evaluator : *yy Heri Pramudiyanto, S.T.*
Pekerjaan : *PPC Guru*

Petunjuk

1. Lembar validasi ini digunakan untuk menilai kualitas media permainan *puzzle* yang dikembangkan sebagai suatu media pembelajaran untuk materi algoritma.
2. Lembar validasi ini diisi oleh Ahli Materi.
3. Jawaban diberikan pada skala penilaian yang sudah disediakan dengan petunjuk sebagai berikut ini.

SS = Sangat Setuju

S = Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

4. Berilah tanda (✓) pada kolom yang sesuai dengan pendapat ahli materi terhadap setiap pernyataan.
5. Terimakasih atas kesediaan Bapak/Ibu untuk mengisi lembar validasi ini.

KEUSIONER PENELITIAN

Berilah tanda (✓) pada kolom yang sesuai dengan pendapat anda terhadap setiap pernyataan dibawah ini:

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dapat membantu siswa memahami diagram alir			✓	
2	Permainan dapat membantu siswa mengembangkan pemahaman mengenai konsep algoritma			✓	
3	Permainan dapat membantu siswa mengidentifikasi struktur percabangan			✓	
4	Permainan dapat membantu siswa mengidentifikasi bentuk perulangan pada diagram alir			✓	
5	Permainan mengajarkan konsep perulangan dengan kondisi diawal			✓	
6	Permainan dapat membantu siswa memahami konsep dasar algoritma				✓
7	Permainan dapat membantu siswa memahami perulangan dengan pencacahan			✓	
8	Permainan mengajarkan bagaimana percabangan digunakan dalam diagram alir				✓
9	Permainan mengajarkan bagaimana konsep perulangan yang ada pada diagram alir			✓	
10	Permainan dapat membantu siswa memahami konsep percabangan dengan 2 kondisi				✓
11	Permainan dapat membantu siswa memahami struktur algoritma				✓
12	Permainan mengajarkan konsep perulangan dengan kondisi diakhiri			✓	
13	Permainan mengajarkan bagaimana alur algoritma percabangan dalam diagram alir			✓	
14	Permainan mengajarkan siswa bagaimana algoritma bekerja				✓

Kritik/Saran :

Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
 - Layak digunakan untuk penelitian dengan perbaikan
 - Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Magelang,

Ahli Materi,

(Yu Heri Pramudiyanto, S.T.)

LEMBAR EVALUASI AHLI MATERI
PENGEMBANGAN PERMAINAN PUZZLE SEBAGAI MEDIA PEMBELAJARAN
ALGORITMA PADA MATA PELAJARAN PEMROGRAMAN DASAR KELAS X DI SMK
NEGERI 1 MAGELANG

Materi Pokok : Konsep Dasar Algoritma

Peneliti : Permadhi Afrian

Evaluator : *HANANG WIDYANTO*

Pekerjaan : *PNS BURU*

Petunjuk

1. Lembar validasi ini digunakan untuk menilai kualitas media permainan *puzzle* yang dikembangkan sebagai suatu media pembelajaran untuk materi algoritma.
2. Lembar validasi ini diisi oleh Ahli Materi.
3. Jawaban diberikan pada skala penilaian yang sudah disediakan dengan petunjuk sebagai berikut ini.

SS = Sangat Setuju

S = Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

4. Berilah tanda (✓) pada kolom yang sesuai dengan pendapat ahli materi terhadap setiap pernyataan.
5. Terimakasih atas kesediaan Bapak/Ibu untuk mengisi lembar validasi ini.

KEUSIONER PENELITIAN

Berilah tanda (✓) pada kolom yang sesuai dengan pendapat anda terhadap setiap pernyataan dibawah ini:

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Permainan dapat membantu siswa memahami diagram alir				✓
2	Permainan dapat membantu siswa mengembangkan pemahaman mengenai konsep algoritma		✓		
3	Permainan dapat membantu siswa mengidentifikasi struktur percabangan		✓		
4	Permainan dapat membantu siswa mengidentifikasi bentuk perulangan pada diagram alir				✓
5	Permainan mengajarkan konsep perulangan dengan kondisi diawal		✓		
6	Permainan dapat membantu siswa memahami konsep dasar algoritma		✓		
7	Permainan dapat membantu siswa memahami perulangan dengan pencacahan		✓		
8	Permainan mengajarkan bagaimana percabangan digunakan dalam diagram alir		✓		
9	Permainan mengajarkan bagaimana konsep perulangan yang ada pada diagram alir				✓
10	Permainan dapat membantu siswa memahami konsep percabangan dengan 2 kondisi		✓		
11	Permainan dapat membantu siswa memahami struktur algoritma		✓		
12	Permainan mengajarkan konsep perulangan dengan kondisi diakhir				✓
13	Permainan mengajarkan bagaimana alur algoritma percabangan dalam diagram alir		✓		
14	Permainan mengajarkan siswa bagaimana algoritma bekerja				✓

Kritik/Saran :

.....
.....
.....
.....
.....

Media ini dinyatakan *):

- Layak digunakan untuk penelitian tanpa perbaikan
- Layak digunakan untuk penelitian dengan perbaikan
- Tidak layak digunakan untuk penelitian

*) Pilih salah satu dengan memberi tanda ✓

Magelang,

Ahli Materi,

(.....NAHANG WIDYANTO, S.KOM.,.....)

Lampiran 8. Instrumen Pengujian Pengguna

KUESIONER PENGGUNA

"PETUALANGAN BLENDING"

A. Data Responden

1. Nama :
2. Kelas :

B. Petunjuk

1. Mohon ketersediaan Anda untuk menjawab seluruh pernyataan yang ada.
2. Jawaban dijamin kerahasiaanya dan tidak mempengaruhi nilai di sekolah.
3. Jawaban diberikan pada alternatif jawaban yang sudah disediakan. Petunjuk jawaban sebagai berikut:

SS = Sangat Setuju

S = Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

4. Berilah tanda (✓) pada kolom yang sesuai dengan pendapat ahli materi terhadap setiap pernyataan.
5. Terimakasih atas kesediaan Anda karena telah berpartisipasi dalam penelitian ini.

No	Pernyataan	Jawaban			
		STS	TS	S	SS
1	Saya dapat melakukan navigasi dengan mudah pada setiap menu				
2	Saya merasa tingkat kesulitan pada setiap level hampir sama				
3	Saya dapat mengendalikan permainan dengan mudah				
4	Saya merasa dapat menyelesaikan setiap level				
5	Saya merasa terbantu dalam memahami struktur perulangan algoritma				
6	Terdapat suatu kesalahan yang membuat saya tidak dapat melanjutkan permainan				
7	Saya merasa terbantu dalam memahami algoritma menggunakan diagram alir				
8	Saya merasa permainan menyajikan hal yang menarik				

No	Pernyataan	Jawaban			
		STS	TS	S	SS
9	Saya melihat animasi ketika saya melakukan sesuatu terhadap permainan				
10	Saya dapat menyesuaikan diri dengan setiap kesulitan yang ada pada permainan				
11	Saya merasa dapat memahami percabangan pada algoritma dengan lebih baik				
12	Saya merasa kesulitan dalam mengendalikan permainan				
13	Saya merasa tidak senang dengan jenis permainan				
14	Saya merasa bosan ditengah permainan				
15	Saya merasa bingung dengan susunan menu yang ada pada permainan				
16	Saya merasa senang karena permainan memberikan suatu hiburan yang berbeda				
17	Saya merasa terbantu dalam memahami struktur percabangan algoritma dengan diagram alir				
18	Saya dapat memahami kendali permainan dengan baik				
19	Saya merasa petunjuk cara bermain yang diberikan kurang jelas				
20	Saya merasa isi permainan terdapat kesamaan dengan yang ada pada kehidupan nyata				
21	Terdapat efek suara ketika saya melakukan sesuatu terhadap permainan				
22	Saya merasa level yang ada pada permainan semakin lama semakin sulit				
23	Saya merasa isi permainan dapat digunakan pada kehidupan sehari - hari				
24	Saya merasa senang dengan jenis permainan dan karakter pada permainan				
25	Saya merasa permainan telah didesain secara adil				
26	Saya merasa setiap pilihan yang saya ambil ketika bermain tidak terlalu penting				

No	Pernyataan	Jawaban			
		STS	TS	S	SS
27	Saya merasa dapat memahami konsep perulangan pada algoritma dengan lebih baik				
28	Saya memahami konsep perulangan dengan diagram alir dengan lebih baik				
29	Saya merasa dapat memahami petunjuk cara bermain dengan mudah				
30	Saya merasa terbantu dalam memahami konsep dasar algoritma				
31	Saya merasa setiap langkah yang ada permainan mempengaruhi hasil yang akan didapat				
32	Selama bermain permainan saya tidak menemukan kesalahan sistem dalam permainan				
33	Saya merasa memerlukan petunjuk lebih dari yang diberikan oleh sistem				
34	Saya merasa terdapat kecurangan yang membuat kondisi permainan selalu kalah				
35	Saya merasa tombol yang ada tersusun dengan rapi				
36	Saya merasa dapat memahami algoritma lebih baik				
37	Saya selalu menemukan jawaban walaupun setiap level terasa sulit				
38	Saya merasa tertarik untuk menyelesaikan permainan				
39	Saya menemukan beberapa bug selama bermain permainan				

Komentar/Saran : Magelang, 2018

Siswa,

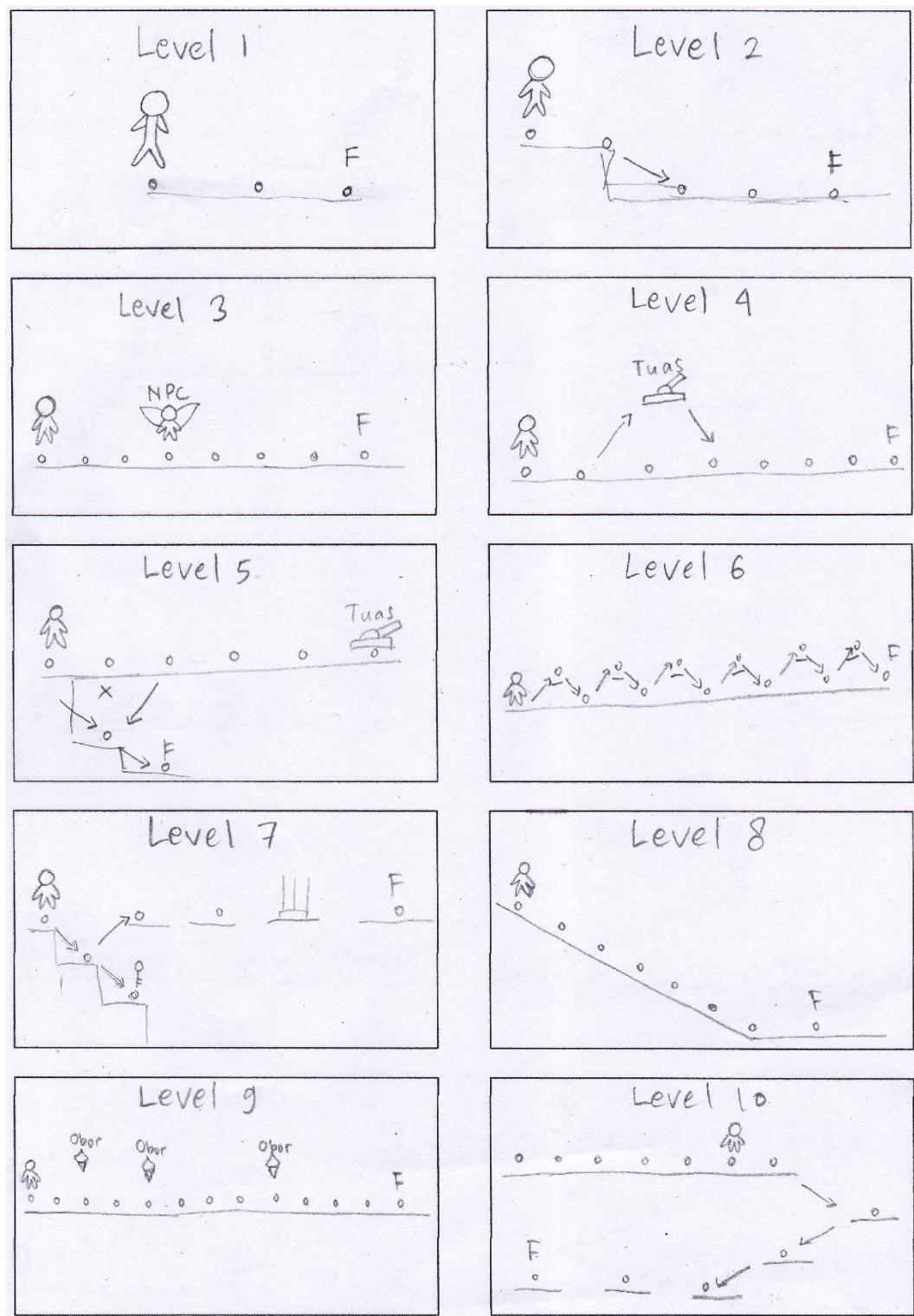
.....

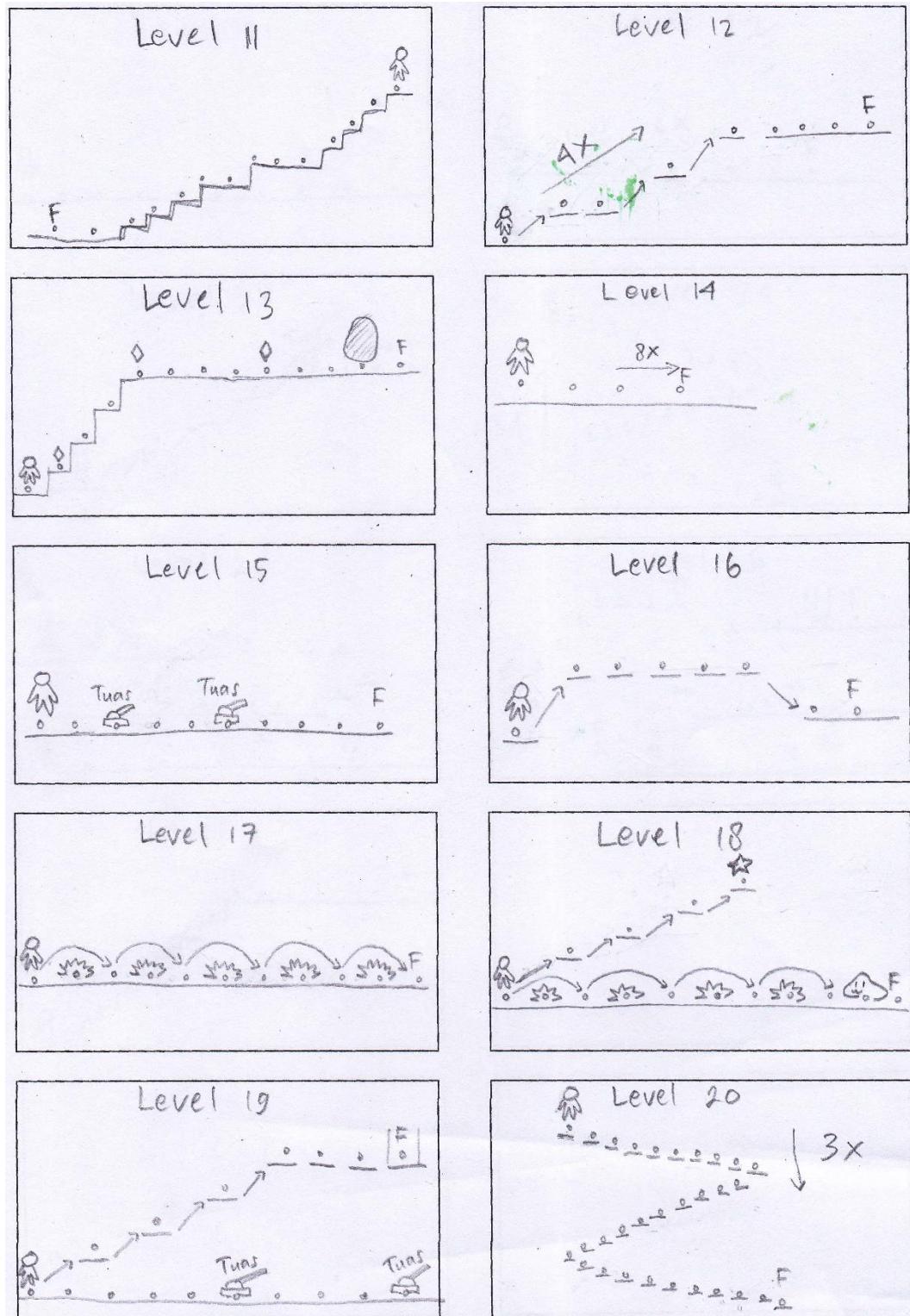
.....

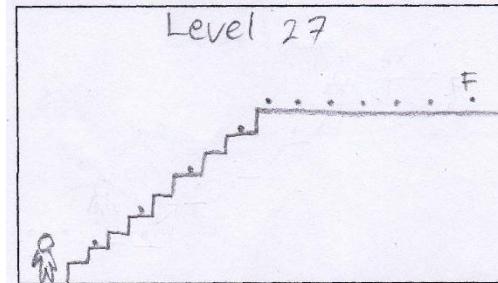
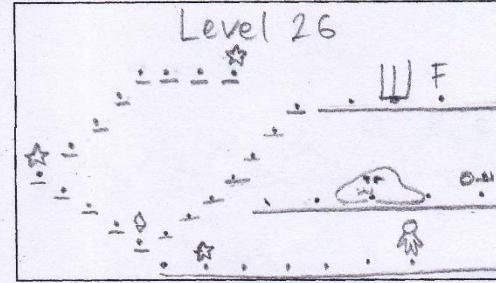
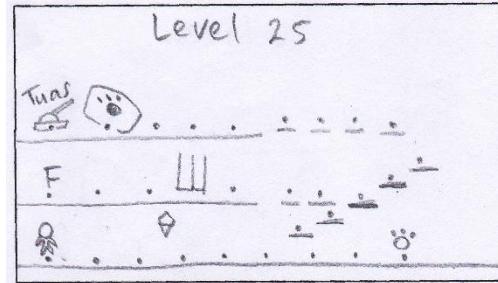
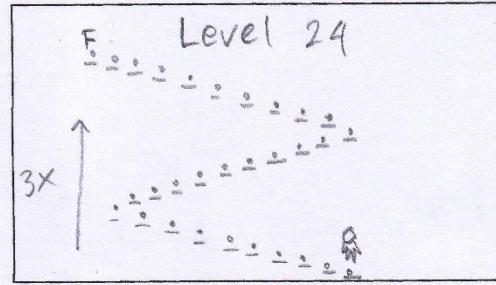
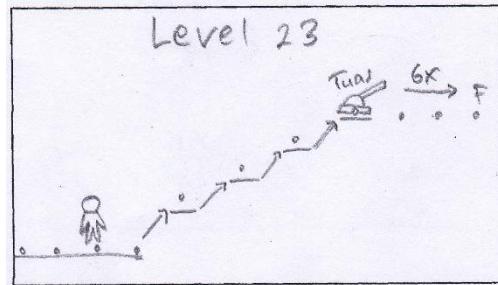
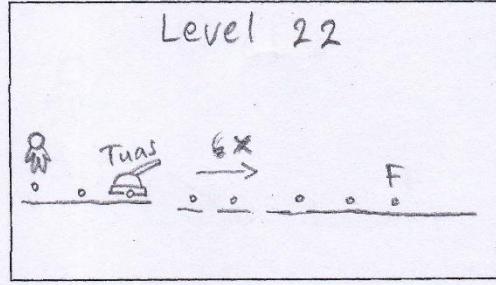
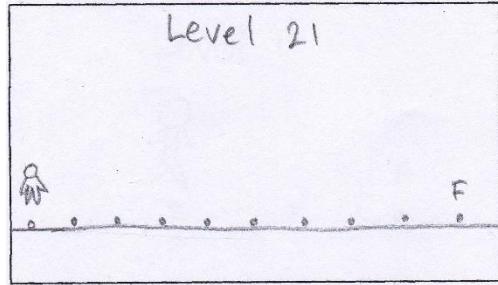
..... (.....)

.....

Lampiran 9. Hasil Desain Level







Lampiran 10. Hasil Kode Program

AlgorithmBlock.cs

```
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.UI;

public class AlgorithmBlock : MonoBehaviour,
    IDragHandler,
    IBeginDragHandler,
    IEndDragHandler,
    IPointerEnterHandler,
    IPointerExitHandler
{
    public ConnectorButton[] connector;
    public bool moveable = true;
    public Color freeColor;
    public Color hoverColor;
    public Command command;

    private Camera cam;
    Vector3 dragOffset;
    Image img;
    bool dragging = false;
    private GameObject m_trash;

    public GameObject Trash
    {
        get
        {
            return m_trash;
        }
        set
        {
            m_trash = value;
        }
    }

    void Start()
    {
        cam = Camera.main;
        img = GetComponent<Image>();
    }

    public void SetCommandName(TMPro.TextMeshProUGUI value)
    {
        command.name = value.text;
    }

    public void OnBeginDrag(PointerEventData eventData)
    {
        if (moveable)
        {
            dragging = true;
            Vector3 pos = cam.ScreenToWorldPoint(new Vector3(eventData.position.x,
                eventData.position.y, 10.73f));
            dragOffset = pos - transform.position;
            if(Trash != null) Trash.SetActive(true);
            var group = gameObject.AddComponent<CanvasGroup>();
            group.blocksRaycasts = false;
        }
    }
}
```

```

public void OnDrag(PointerEventData eventData)
{
    if(moveable)
    {
        Vector3 pos = cam.ScreenToWorldPoint(new Vector3(eventData.position.x,
        eventData.position.y, 10.73f));
        transform.position = new Vector3(pos.x - dragOffset.x, pos.y - dragOffset.y, 0f);
        foreach (ConnectorButton current in connector)
        {
            if(current.dragable)
            {
                if (current.GetTail() != null)
                {
                    current.GetTail().transform.position = current.transform.position;
                    current.SetTailPos(0, current.transform.position);
                    current.SetTailPos(1, current.SnapTo.transform.position);
                    current.MoveAndRotateArrowHead();
                }
            } else
            {
                ConnectorButton s = current.WhoSnapToMe;
                if (s != null)
                {
                    s.SetTailPos(1, current.transform.position);
                    s.MoveAndRotateArrowHead();
                }
            }
        }
    }
}

public void OnPointerEnter(PointerEventData eventData)
{
    if (!dragging)
    {
        img.color = img.color * hoverColor;
    }
}

public void OnPointerExit(PointerEventData eventData)
{
    if (!dragging)
    {
        img.color = freeColor;
    }
}

public void OnEndDrag(PointerEventData eventData)
{
    if (moveable)
    {
        dragging = false;
        img.color = hoverColor;
        Destroy.GetComponent<CanvasGroup>());
        if(Trash != null) Trash.SetActive(false);
    }
}

public Transform rightConnector()
{
    Transform[] childs = new Transform[transform.childCount];
    for (int i = 0; i < transform.childCount; i++)
    {
        childs[i] = transform.GetChild(i);
    }
}

```

```

foreach(Transform child in childs)
{
    if(child.name == "ConnectorRight")
    {
        return child;
    }
}
return null;
}

public Transform bottomConnector()
{
    Transform[] childs = new Transform[transform.childCount];
    for (int i = 0; i < transform.childCount; i++)
    {
        childs[i] = transform.GetChild(i);
    }

    foreach (Transform child in childs)
    {
        if (child.name == "ConnectorBot")
        {
            return child;
        }
    }
    return null;
}

public Transform leftConnector()
{
    Transform[] childs = new Transform[transform.childCount];
    for (int i = 0; i < transform.childCount; i++)
    {
        childs[i] = transform.GetChild(i);
    }

    foreach (Transform child in childs)
    {
        if (child.name == "ConnectorLeft")
        {
            return child;
        }
    }
    return null;
}

public Transform topConnector()
{
    Transform[] childs = new Transform[transform.childCount];
    for (int i = 0; i < transform.childCount; i++)
    {
        childs[i] = transform.GetChild(i);
    }

    foreach (Transform child in childs)
    {
        if (child.name == "ConnectorTop")
        {
            return child;
        }
    }
    return null;
}
}

```

AlgorithmButton.cs

```
using UnityEngine;
using UnityEngine.EventSystems;
using UnityEngine.UI;

public class AlgorithmButton : MonoBehaviour,
    IPointerDownHandler,
    IPointerEnterHandler,
    IPointerExitHandler,
    IPointerUpHandler
{
    public bool interactable = true;
    public GameObject prefabs;
    public AudioClip clip;
    public Color hoverColor;
    public Color downColor;
    public Color disabledColor;
    public Transform parent;
    public GameObject trash;

    private Color oriColor;
    private Image img;
    private AudioContent sfx;

    void Awake()
    {
        GameObject[] a = GameObject.FindGameObjectsWithTag("Audio");

        for (int i = 0; i < a.Length; i++)
        {
            if(a[i].name == "SoundHolder")
            {
                sfx = a[i].GetComponent<AudioContent>();
            }
        }

        img = GetComponent<Image>();
        oriColor = img.color;
        img.color = interactable ? oriColor : disabledColor;
    }

    void IPointerDownHandler.OnPointerDown(PointerEventData eventData)
    {
        if (!interactable) return;
        sfx.PlayOneShot(clip);
        GameObject go = Instantiate(prefabs, Vector3.right * 2f, Quaternion.identity, parent);
        go.transform.SetAsLastSibling();
        go.GetComponent<AlgorithmBlock>().Trash = trash;
        img.color = oriColor * downColor;
    }

    void IPointerEnterHandler.OnPointerEnter(PointerEventData eventData)
    {
        if (!interactable) return;
        img.color = oriColor * hoverColor;
    }

    void IPointerExitHandler.OnPointerExit(PointerEventData eventData)
    {
        if (!interactable) return;
        img.color = oriColor;
    }

    void IPointerUpHandler.OnPointerUp(PointerEventData eventData)
    {
        if (!interactable) return;
        img.color = oriColor;
    }
}
```

AlgorithmPanel.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AlgorithmPanel : MonoBehaviour
{
    public GameState gameState;

    Animator animator;

    private void Awake()
    {
        animator = GetComponent<Animator>();
        gameState.editingMode = !animator.GetBool("Hide");
    }

    public void Click()
    {
        animator.SetBool("Hide", !animator.GetBool("Hide"));
        gameState.editingMode = !animator.GetBool("Hide");
    }

    public void Hide()
    {
        animator.SetBool("Hide", true);
    }
}
```

AlgorithmSystem.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;

public class AlgorithmSystem : MonoBehaviour {

    public Transform startAlg;
    public CharacterMovement character;
    public GameObject Win;
    public TextMeshProUGUI timerText;
    public GameObject Lose;
    public GameState gameState;
    public PlayerAttribute player;
    public AudioContent music;
    public AudioClip winSound;
    public AudioClip loseSound;
    public GameObject algCam;
    public LevelProgress currentLevel;
    public LevelProperty levelProperty;
    public bool isEndLevel = false;
    public Highscore highscore;
```

```

private Transform lastTarget;
private bool conditioned;
private Command counter;
private Item itemTaken;
private List<GameObject> algBlocks;
private int safetyCheck;
private List<HighscoreData> highscores;

private void Start()
{
    player.level_title = levelProperty.levelName;
    player.score = currentLevel.score;
    algBlocks = new List<GameObject>();
    gameState.playingMode = false;
    gameState.finishedPlaying = false;
}

public void StartAlgorithm()
{
    safetyCheck = 1000;
    gameState.playingMode = true;
    conditioned = true;

    ConnectorButton connector = startAlg
        .GetComponent<AlgorithmBlock>()
        .bottomConnector()
        .GetComponent<ConnectorButton>();

    algCam.SetActive(true);

    -----
    Transform next = null;

    if (connector != null)
        next = connector.GetNextCommand();
    if (next != null)
        next = next.parent;

    if (next != null)
        StartCoroutine(StartAction(next));
    else
        StartCoroutine(StartAction(null));
}

void WinCondition()
{
    highscores = highscore.RequestHighscoreList(highscore.m_scoreBoard.entry,
                                                levelProperty.levelId);

    gameState.finishedPlaying = true;

    currentLevel.score = currentLevel.score - levelProperty.algUsed;

    algCam.SetActive(false);

    if (!levelProperty.completed)
    {
        timerText.text = algBlocks.Count - 1 + " Blok Dipakai";
        levelProperty.algUsed = algBlocks.Count - 1;
        currentLevel.score += levelProperty.algUsed;
    }
}

```

```

        currentLevel.progress++;
        levelProperty.completed = true;
    }
    else
    {
        timerText.text = algBlocks.Count - 1 + " Blok Dipakai" +
        "\nBlok Terbaikmu : " + levelProperty.algUsed;

        if (levelProperty.algUsed > algBlocks.Count - 1)
            levelProperty.algUsed = algBlocks.Count - 1;

        currentLevel.score += levelProperty.algUsed;
    }

    HighscoreData highscoreData = new HighscoreData(gameState.username,
                                                    levelProperty.algUsed,
                                                    levelProperty.levelId);

    if(highscores.Exists(x => x.username.Contains(gameState.username)))
    {
        highscores.Remove(highscores.Find(x => x
                                            .username
                                            .Contains(gameState.username)));
    }

    highscores.Add(highscoreData);
    WinPanel wp = Win.GetComponent<WinPanel>();

    AscendingHighscore ah = new AscendingHighscore();
    highscores.Sort(ah);
    HighscoreData[] asc = highscores.ToArray();

    for (int i = 0; i < wp.user.Length; i++)
    {
        if(i < asc.Length)
        {
            wp.user[i].text = asc[i].username;
            wp.score[i].text = asc[i].score.ToString();
        } else
        {
            break;
        }
    }

    Win.SetActive(true);
    highscore.AddNewHighScore(gameState.username,
                             levelProperty.algUsed,
                             levelProperty.levelId);
}

void LoseCondition()
{
    gameState.finishedPlaying = true;
    algCam.SetActive(false);
    Lose.SetActive(true);
}

IEnumerator Interact(Transform t, Interactable interactable)

```

```

{
    Camera m_cam = null;
    foreach (Camera c_cam in Camera.allCameras)
    {
        if(c_cam.name == "Main Camera")
        {
            m_cam = c_cam;
            break;
        }
    }
    CameraScript cs = m_cam.GetComponent<CameraScript>();
    if (cs != null) cs.enabled = false;
    Vector3 oldCamPos = m_cam.transform.position;
    Vector3 tPos = t.position;
    tPos.z = oldCamPos.z;
    float e_time = 0f;
    while(Vector3.Distance(m_cam.transform.position, tPos) > 0.2f)
    {
        m_cam.transform.position = Vector3.Lerp(oldCamPos, tPos, e_time);
        e_time += Time.deltaTime;
        yield return null;
    }
    m_cam.transform.position = tPos;
    interactable.Interact();
    yield return new WaitForSeconds(1f);
    e_time = 0f;
    while(Vector3.Distance(m_cam.transform.position, oldCamPos) > 0.2f)
    {
        m_cam.transform.position = Vector3.Lerp(tPos, oldCamPos, e_time);
        e_time += Time.deltaTime;
        yield return null;
    }
    m_cam.transform.position = oldCamPos;
    if (cs != null) cs.enabled = true;
    yield return null;
}

IEnumerator Take(Transform t, ItemSystem item)
{
    Camera m_cam = null;
    foreach (Camera c_cam in Camera.allCameras)
    {
        if (c_cam.name == "Main Camera")
        {
            m_cam = c_cam;
            break;
        }
    }
    CameraScript cs = m_cam.GetComponent<CameraScript>();
    if (cs != null) cs.enabled = false;
    Vector3 oldCamPos = m_cam.transform.position;
    Vector3 tPos = t.position;
    tPos.z = oldCamPos.z;
    float e_time = 0f;
    while (Vector3.Distance(m_cam.transform.position, tPos) > 0.2f)
    {
        m_cam.transform.position = Vector3.Lerp(oldCamPos, tPos, e_time);
        e_time += Time.deltaTime;
        yield return null;
    }
}

```

```

        m_cam.transform.position = tPos;
        item.TakeItem();
        yield return new WaitForSeconds(1f);
        item.affectedObject.ItemUsed(itemTaken);
        yield return new WaitForSeconds(1f);
        e_time = 0f;
        while (Vector3.Distance(m_cam.transform.position, oldCamPos) > 0.2f)
        {
            m_cam.transform.position = Vector3.Lerp(tPos, oldCamPos, e_time);
            e_time += Time.deltaTime;
            yield return null;
        }
        m_cam.transform.position = oldCamPos;
        if (cs != null) cs.enabled = true;
        yield return null;
    }

    public bool AllConnected()
    {
        ConnectorButton connector = startAlg
            .GetComponent<AlgorithmBlock>()
            .bottomConnector()
            .GetComponent<ConnectorButton>();

        Transform next = null;

        if (connector != null)
            next = connector.GetNextCommand();

        if (next != null)
            next = next.parent;

        while (next != null && next.name != "EndAlg")
        {
            connector = next
                .GetComponent<AlgorithmBlock>()
                .bottomConnector()
                .GetComponent<ConnectorButton>();
            if(connector == null)
            {
                return false;
            } else
            {
                next = connector.GetNextCommand();
            }

            if(next != null)
            {
                next = next.parent;

                if (next.name == "EndAlg") return true;

            } else
            {
                return false;
            }
        }
        return false;
    }
}

```

```

IEnumerator StartAction(Transform n)
{
    if (!conditioned) yield break;

    Transform next = null;

    if (n == null || --safetyCheck <= 0)
    {
        character.animator.SetBool("Failed", true);
        music.audioSource.Stop();
        music.PlayOneShot(loseSound);
        yield return new WaitForSeconds(2f);
        LoseCondition();
        conditioned = false;
        yield break;
    }

    if (!algBlocks.Contains(n.gameObject))
    {
        algBlocks.Add(n.gameObject);
    }

    n.GetComponent<Image>().color = Color.green;

    Command command = n.GetComponent<AlgorithmBlock>().command;

    if (command.name == "End")
    {
        if (character.currentNode.GetComponent<PathNode>().isFinishNode)
        {
            character.animator.SetBool("Win", true);
            music.audioSource.Stop();
            music.PlayOneShot(winSound);
            yield return new WaitForSeconds(2f);
            WinCondition();
            conditioned = false;
            yield break;
        }
        else
        {
            character.animator.SetBool("Failed", true);
            music.audioSource.Stop();
            music.PlayOneShot(loseSound);
            yield return new WaitForSeconds(2f);
            LoseCondition();
            conditioned = false;
            yield break;
        }
    }

    if (command.type == Command.Type.Input)
    {
        int count = int.Parse(command.name);
        count--;
        if (count > 0)
        {
            next = lastTarget;
            command.name = count.ToString();
        } else
        {
            lastTarget = n;
        }
    }
}

```

```

ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
    .bottomConnector()
    .GetComponent<ConnectorButton>();

if (connector != null)
    next = connector.GetNextCommand();

        if (next != null) next = next.parent;
    }
} else
{
    lastTarget = n;
}

if (command.type == Command.Type.Counter)
{
    if (command.name == "Add")
    {
        if (counter != null)
        {
            int count = int.Parse(counter.name);
            count++;
            counter.name = count.ToString();
        }
    }
    else
    {
        counter = command;
    }
}

if (command.type == Command.Type.Process)
{
    if (command.name == "Take")
    {
        ItemSystem item = character
            .currentNode
            .GetComponent<PathNode>()
            .item;
        if (item != null)
        {
            if (item.directEffect)
            {
                itemTaken = item.item;
                yield return StartCoroutine(Take(item
                    .affectedObject
                    .transform,
                    item));
            } else
            {
                itemTaken = item.item;
                item.TakeItem();
                if (item.affectedObject.KeyTaken())
                {
                    yield return StartCoroutine(Take(item
                        .affectedObject
                        .transform,
                        item));
                } else
                {
                    yield return new WaitForSeconds(.3f);
                }
                yield return new WaitForSecondsEndOfFrame();
            }
        }
    }
}

```

```

        }
        else if(command.name == "Interact")
        {
            Interactable interactable = character
                .currentNode
                .GetComponent<PathNode>()
                .interactable;
            if(interactable != null)
            {
                yield return StartCoroutine(Interact(interactable
                    .interactable
                    .transform,
                    interactable));
            }
        }
        else if(command.name == "Turn")
        {
            character.Turning();
            yield return new WaitForSeconds(.5f);
        }
        else
        {
            if (!character.MoveCharacter(command.name))
            {
                character.animator.SetBool("Failed", true);
                music.audioSource.Stop();
                music.PlayOneShot(loseSound);
                yield return new WaitForSeconds(2f);
                LoseCondition();
                conditioned = false;
                yield break;
            }
        }
    }

    while (character.DoingSmth)
    {
        if(player.hp <= 0)
        {
            character.animator.SetBool("Dead", true);
            music.audioSource.Stop();
            music.PlayOneShot(loseSound);
            yield return new WaitForSeconds(2f);
            LoseCondition();
            conditioned = false;
            yield break;
        }
        yield return null;
    }

    if (command.type != Command.Type.Decision &&
        command.type != Command.Type.Input && command.name != "End")
    {
        ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
            .bottomConnector()
            .GetComponent<ConnectorButton>();
        Transform con = null;
        if(connector != null)
            con = connector.GetNextCommand();

        if(con != null) next = con.parent;
    }
}

```

```

} else if(command.type == Command.Type.Decision && command.name != "End")
{

    Decision decision = n.GetComponent<Decision>();
    if(decision.type == Decision.Type.Place)
    {
        if(decision.identifier == "Portal" &&
            character.currentNode.GetComponent<PathNode>().isFinishNode)
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .bottomConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }
        else if(decision.identifier == "Portal")
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .rightConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }
        else if (character.CheckMovement(decision.identifier))
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .bottomConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }
        else if (!character.CheckMovement(decision.identifier))
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .rightConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }

        yield return new WaitForSeconds(.2f);
    }
    else if (decision.type == Decision.Type.Item)
    {
        if (character.
            currentNode.
            GetComponent<PathNode>() .
            item != null &&
            character
            .currentNode

```

```

        .GetComponent<PathNode>()
        .item.name == decision.identifier)
    {
        ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
            .bottomConnector()
            .GetComponent<ConnectorButton>();

        if (connector != null)
            next = connector.GetNextCommand();

        if (next != null) next = next.parent;
    }
    else
    {
        ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
            .rightConnector()
            .GetComponent<ConnectorButton>();

        if (connector != null)
            next = connector.GetNextCommand();

        if (next != null) next = next.parent;
    }

        yield return new WaitForSeconds(.2f);
}
else if (decision.type == Decision.Type.Counter)
{
    if(counter != null)
    {
        if(int.Parse(counter.name) <= int.Parse(decision.identifier))
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .rightConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }
    }

        yield return new WaitForSeconds(.2f);
}
else if (decision.type == Decision.Type.Interactable)
{
    if (character
        .currentNode
        .GetComponent<PathNode>()
        .interactable!= null &&
        character
        .currentNode
        .GetComponent<PathNode>()
        .interactable.name == decision.identifier)
    {
        ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
            .bottomConnector()
            .GetComponent<ConnectorButton>();

        if (connector != null)
            next = connector.GetNextCommand();

        if (next != null) next = next.parent;
    }
}

```

```

        }
        else
        {
            ConnectorButton connector = n.GetComponent<AlgorithmBlock>()
                .rightConnector()
                .GetComponent<ConnectorButton>();

            if (connector != null)
                next = connector.GetNextCommand();

            if (next != null) next = next.parent;
        }

        yield return new WaitForSeconds(.2f);
    }
}

n.GetComponent<Image>().color = Color.white;

if (next != null)
{
    StartCoroutine(StartAction(next));
}
else
{
    character.animator.SetBool("Failed", true);
    music.audioSource.Stop();
    music.PlayOneShot(LoseSound);
    yield return new WaitForSeconds(2f);
    LoseCondition();
    conditioned = false;
    yield break;
}
yield return null;
}
}
}

```

Audio.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu()]
public class Audio : ScriptableObject
{
    public bool enabled;
    [Range(0f, 1f)]
    public float value;
}

```

AudioContent.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AudioContent : MonoBehaviour
{
    public AudioSource audioSource;
    public Audio au;

    private void Start()
    {
        audioSource.mute = !au.enabled;
        audioSource.volume = au.value;
    }

    public void ValueChaged()
    {
        audioSource.mute = !au.enabled;
        audioSource.volume = au.value;
    }

    public void PlayOneShot(AudioClip audio)
    {
        audioSource.PlayOneShot(audio);
    }

    public void ChangeClip(AudioClip audio)
    {
        audioSource.Stop();
        audioSource.clip = audio;
        PlayClip();
    }

    public void PlayClip()
    {
        audioSource.Play();
    }
}
```

AudioSetting.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class AudioSetting : MonoBehaviour
{
    public Audio m_audio;
    public Toggle toggle;
    public Slider slider;

    void Start ()
    {
        toggle.isOn = m_audio.enabled;
        slider.value = m_audio.value;
    }
}
```

```

        public void ValueChanged()
    {
        m_audio.enabled = toggle.isOn;
        m_audio.value = slider.value;
    }
}

```

Body.cs

```

using UnityEngine;
using TMPro;

public class Body : MonoBehaviour
{
    public TextMeshProUGUI username;
    public TextMeshProUGUI score;
}

```

CameraScript.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraScript : MonoBehaviour
{
    public GameState gameState;
    public Transform target;
    public Transform leftFog;
    public Transform rightFog;
    public Transform topFog;
    public Transform botFog;
    [Range(0.1f, 10f)]
    public float smoothness = 0.5f;
    [Range(0.1f, 10f)]
    public float mouseSensitivity = 1f;
    public float xOffset = 2f;
    public float yOffset = 2f;
    private Vector3 vel = Vector3.zero;
    private Vector3 lastPos;
    private Vector3 panOrigin;

    private void LateUpdate()
    {
        if(gameState.playingMode)
        {
            Vector3 targetPos =
                new Vector3(target.position.x,
                target.position.y + 1f,
                target.position.z - 10f);

            targetPos.x =
                Mathf.Clamp(targetPos.x,
                leftFog.position.x + 1.5f,
                rightFog.position.x - 1.5f);
        }
    }
}

```

```

        transform.position =
        Vector3.SmoothDamp(transform.position,
        targetPos,
        ref vel,
        smoothness);

    }
    else if(!gameState.editingMode && !gameState.paused)
    {
        if(Input.GetMouseButtonDown(0))
        {
            lastPos = transform.position;
            panOrigin = Input.mousePosition;
        }

        if(Input.GetMouseButton(0))
        {
            Vector3 pos = GetComponent<Camera>()
            .ScreenToViewportPoint(Input.mousePosition - panOrigin);
            transform.position =
            new Vector3(Mathf.Clamp(lastPos.x - pos.x * mouseSensitivity * 5f,
            leftFog.position.x + xOffset, rightFog.position.x - xOffset),
            Mathf.Clamp(lastPos.y - pos.y * mouseSensitivity * 5f,
            botFog.position.y + yOffset,
            topFog.position.y - yOffset),
            lastPos.z);
        }
    }
}
}

```

CharacterMovement.cs

```

using System.Collections;
using UnityEngine;

public class CharacterMovement : MonoBehaviour {

    public Transform currentNode;
    public PlayerAttribute player;
    public float jumpHeight = 2.5f;

    bool faceRight = true;
    bool doingSmth = false;

    Animator m_animator;

    public bool DoingSmth
    {
        get
        {
            return doingSmth;
        }
        set
        {
            doingSmth = value;
        }
    }

    public Animator animator
    {
        get
        {
            return m_animator;
        }
        set
        {

```

```

        m_animator = value;
    }
}

private void Start()
{
    animator = GetComponent<Animator>();
    faceRight = transform.localScale.x < 0f;
}

public void Turning()
{
    transform.localScale = new Vector3(transform.localScale.x * -1f,
        transform.localScale.y, transform.localScale.z);
    faceRight = transform.localScale.x < 0f;
}

IEnumerator Walking(Transform target)
{
    doingSmth = true;

    Vector3 oldPos = transform.position;

    if(target.GetComponent<PathNode>().npc != null)
    {
        NPC n = target.GetComponent<PathNode>().npc;
        n.InteractWith();
        while(n.talking)
        {
            yield return null;
        }
    }

    if (target.GetComponent<PathNode>().enemy != null)
    {
        Enemy e = target.GetComponent<PathNode>().enemy;
        e.InteractWith();
        while (e.talking)
        {
            yield return null;
        }
    }

    float timetaken = 0f;
    animator.SetBool("Run", true);
    while (Vector3.Distance(transform.position, target.position) > 0.05f)
    {
        transform.position = Vector3.Lerp(oldPos, target.position, timetaken);
        timetaken += Time.deltaTime * .7f;
        yield return null;
    }
    animator.SetBool("Run", false);

    if (target.GetComponent<PathNode>().obstacle != null &&
        target.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
    {
        player.hp--;
        animator.SetBool("TakeDamage", true);
        yield return new WaitForSeconds(1f);
        animator.SetBool("TakeDamage", false);
    }
    currentNode = target;
    doingSmth = false;
}

IEnumerator Jump(Transform target, float heightdesired)
{
    doingSmth = true;
    float timetaken = 0f;
    Vector3 oldPos = transform.position;
    animator.SetBool("Drop", false);
    animator.SetBool("Jump", true);
    while (Vector3.Distance(transform.position, target.position) > 0.2f || timetaken < .5f)
    {
        float height = Mathf.Sin(Mathf.PI * timetaken) * heightdesired;
        transform.position = Vector3.Lerp(oldPos, target.position, timetaken) + Vector3.up * height;
        timetaken += Time.deltaTime * .7f;
        if (Vector3.Distance(transform.position, target.position) < 0.5f && timetaken > .6f)
        {
            animator.SetBool("Jump", false);
            animator.SetBool("Drop", true);
        }
    }
}

```

```

        }
        yield return null;
    }

    transform.position = target.position;

    currentNode = target;

    while (animator.GetCurrentAnimatorStateInfo(0).IsName("Drop") ||
        animator.GetCurrentAnimatorStateInfo(0).IsName("Jump") ||
        animator.GetCurrentAnimatorStateInfo(0).IsName("AirBorne"))
    {
        yield return new WaitForSeconds();
    }

    if (target.GetComponent<PathNode>().obstacle != null &&
        target.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
    {
        player.hp--;
        animator.SetBool("TakeDamage", true);
        yield return new WaitForSeconds(1f);
        animator.SetBool("TakeDamage", false);
    }
    yield return new WaitForSeconds();
    doingSmth = false;
}

public bool CheckMovement(string name)
{
    if (name == "Walk")
    {
        if (faceRight)
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.walkRight != null && node.walkRight.GetComponent<PathNode>().Active)
            {
                if (node.walkRight.GetComponent<PathNode>().obstacle == null)
                {
                    return true;
                }
                else if (node.walkRight.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    return true;
                }
            }
        }
        else
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.walkLeft != null && node.walkLeft.GetComponent<PathNode>().Active)
            {
                if (node.walkLeft.GetComponent<PathNode>().obstacle == null)
                {
                    return true;
                }
                else if (node.walkLeft.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    return true;
                }
            }
        }
    }
    if (name == "Jump_Forward")
    {
        if (faceRight)
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.jumpFR != null && node.jumpFR.GetComponent<PathNode>().Active)
            {
                if (node.jumpFR.GetComponent<PathNode>().obstacle == null)
                {
                    return true;
                }
                else if (node.jumpFR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    return true;
                }
            }
        }
    }
}

```

```

        }
    }
}
else
{
    PathNode node = currentNode.GetComponent<PathNode>();
    if (node.jumpFL != null && node.jumpFL.GetComponent<PathNode>().Active)
    {
        if (node.jumpFL.GetComponent<PathNode>().obstacle == null)
        {
            return true;
        }
        else if (node.jumpFL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
        {
            return true;
        }
    }
}

if (name == "Jump_Up")
{
    if (faceRight)
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpUpR != null && node.jumpUpR.GetComponent<PathNode>().Active)
        {
            if (node.jumpUpR.GetComponent<PathNode>().obstacle == null)
            {
                return true;
            }
            else if (node.jumpUpR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                return true;
            }
        }
    }
    else
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpUpL != null && node.jumpUpL.GetComponent<PathNode>().Active)
        {
            if (node.jumpUpL.GetComponent<PathNode>().obstacle == null)
            {
                return true;
            }
            else if (node.jumpUpL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                return true;
            }
        }
    }
}

if (name == "Jump_Down")
{
    if (faceRight)
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpDropR != null && node.jumpDropR.GetComponent<PathNode>().Active)
        {
            if (node.jumpDropR.GetComponent<PathNode>().obstacle == null)
            {
                return true;
            }
            else if (node.jumpDropR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                return true;
            }
        }
    }
    else
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpDropL != null && node.jumpDropL.GetComponent<PathNode>().Active)
        {
            if (node.jumpDropL.GetComponent<PathNode>().obstacle == null)

```

```

        {
            return true;
        }
        else if (node.jumpDropL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
        {
            return true;
        }
    }
}

return false;
}

public bool MoveCharacter(string name)
{
    if(name == "Start")
    {
        return true;
    }

    if (name == "Walk")
    {
        if (faceRight)
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.walkRight != null && node.walkRight.GetComponent<PathNode>().Active)
            {
                if (node.walkRight.GetComponent<PathNode>().obstacle == null)
                {
                    StartCoroutine(Walking(node.walkRight));
                    return true;
                }
                else if (node.walkRight.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    StartCoroutine(Walking(node.walkRight));
                    return true;
                }
            }
        }
        else
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.walkLeft != null && node.walkLeft.GetComponent<PathNode>().Active)
            {
                if (node.walkLeft.GetComponent<PathNode>().obstacle == null)
                {
                    StartCoroutine(Walking(node.walkLeft));
                    return true;
                }
                else if (node.walkLeft.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    StartCoroutine(Walking(node.walkLeft));
                    return true;
                }
            }
        }
    }

    if (name == "Jump_Foward")
    {
        if(faceRight)
        {
            PathNode node = currentNode.GetComponent<PathNode>();
            if (node.jumpFR != null && node.jumpFR.GetComponent<PathNode>().Active)
            {
                if (node.jumpFR.GetComponent<PathNode>().obstacle == null)
                {
                    StartCoroutine(Jump(node.jumpFR, jumpHeight));
                    return true;
                }
                else if (node.jumpFR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
                {
                    StartCoroutine(Jump(node.jumpFR, jumpHeight));
                    return true;
                }
            }
        }
        else
        {
    }
}

```

```

PathNode node = currentNode.GetComponent<PathNode>();
if (node.jumpFL != null && node.jumpFL.GetComponent<PathNode>().Active)
{
    if (node.jumpFL.GetComponent<PathNode>().obstacle == null)
    {
        StartCoroutine(Jump(node.jumpFL, jumpHeight));
        return true;
    }
    else if (node.jumpFL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
    {
        StartCoroutine(Jump(node.jumpFL, jumpHeight));
        return true;
    }
}
}

if (name == "Jump_Up")
{
    if (faceRight)
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpUpR != null && node.jumpUpR.GetComponent<PathNode>().Active)
        {
            if (node.jumpUpR.GetComponent<PathNode>().obstacle == null)
            {
                StartCoroutine(Jump(node.jumpUpR, jumpHeight));
                return true;
            }
            else if (node.jumpUpR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                StartCoroutine(Jump(node.jumpUpR, jumpHeight));
                return true;
            }
        }
    }
    else
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpUpL != null && node.jumpUpL.GetComponent<PathNode>().Active)
        {
            if (node.jumpUpL.GetComponent<PathNode>().obstacle == null)
            {
                StartCoroutine(Jump(node.jumpUpL, jumpHeight));
                return true;
            }
            else if (node.jumpUpL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                StartCoroutine(Jump(node.jumpUpL, jumpHeight));
                return true;
            }
        }
    }
}

if (name == "Jump_Down")
{
    if (faceRight)
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpDropR != null && node.jumpDropR.GetComponent<PathNode>().Active)
        {
            if (node.jumpDropR.GetComponent<PathNode>().obstacle == null)
            {
                StartCoroutine(Jump(node.jumpDropR, jumpHeight));
                return true;
            }
            else if (node.jumpDropR.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
            {
                StartCoroutine(Jump(node.jumpDropR, jumpHeight));
                return true;
            }
        }
    }
    else
    {
        PathNode node = currentNode.GetComponent<PathNode>();
        if (node.jumpDropL != null && node.jumpDropL.GetComponent<PathNode>().Active)
        {

```

```

        if (node.jumpDropL.GetComponent<PathNode>().obstacle == null)
        {
            StartCoroutine(Jump(node.jumpDropL, jumpHeight));
            return true;
        }
        else if (node.jumpDropL.GetComponent<PathNode>().obstacle.type == Obstacle.Type.Trap)
        {
            StartCoroutine(Jump(node.jumpDropL, jumpHeight));
            return true;
        }
    }
}

return false;
}
}

```

Command.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[Serializable]
public class Command {

    [SerializeField]
    private string m_name;
    public enum Type {Terminator, Decision, Process, Input, Counter};
    public Type type;

    public string name
    {
        get
        {
            return m_name;
        }
        set
        {
            m_name = value;
        }
    }
}

```

ConnectorButton.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.UI;

public class ConnectorButton : MonoBehaviour,
    IPointerEnterHandler,
    IPointerExitHandler,
    IDragHandler,
    IBeginDragHandler,
    IEndDragHandler
{

    public GameObject tailArrow;
    public bool dragable = true;
    public Color snappedColor;
}

```

```

private Camera cam;
LineRenderer tailLR;
GameObject tail;
GameObject head;
bool snapped = false;
Transform nextCommand;
Image img;
Image childImg;
Color freeColor;
ConnectorButton whoSnapToMe;
ConnectorButton snapTo;

public ConnectorButton WhoSnapToMe
{
    get
    {
        return whoSnapToMe;
    }

    set
    {
        whoSnapToMe = value;
    }
}

public ConnectorButton SnapTo
{
    get
    {
        return snapTo;
    }

    set
    {
        snapTo = value;
    }
}

public GameObject GetTail()
{
    return tail;
}

void Start()
{
    cam = Camera.main;
    img = GetComponent<Image>();
    childImg = transform.GetChild(0).GetComponent<Image>();
    freeColor = childImg.color;
}
}

Vector3 arPos;
public void OnBeginDrag(PointerEventData eventData)
{
    if (dragable)
    {
        if(snapped)
        {
            Destroy(tail);
            childImg.color = freeColor;
            nextCommand.GetChild(0).GetComponent<Image>().color =
            nextCommand.GetComponent<ConnectorButton>().freeColor;
            nextCommand.GetComponent<ConnectorButton>().whoSnapToMe = null;
            nextCommand = null;
            snapped = false;
        }
    }
}

```

```

        }
        tail = Instantiate(tailArrow);
        tail.transform.SetParent(transform.parent.transform.parent);
        tail.transform.position = transform.position;
        SetTailPos(0, Vector3.zero);
        head = tail.transform.GetChild(0).gameObject;
        tail.SetActive(true);
        head.SetActive(true);
        head.GetComponent<RectTransform>().sizeDelta = new Vector2(.15f, .15f);
        if (tailArrow != null)
            tailLR = tail.GetComponent<LineRenderer>();
    }
}

Vector3 dir;

public void OnDrag(PointerEventData eventData)
{
    if (dragable)
    {
        MoveArrow(cam.ScreenToWorldPoint(eventData.position));
    }
}

//Move Arrow Based on Target Position
public void MoveArrow(Vector3 target)
{
    SetTailPos(1, target);
    MoveAndRotateArrowHead();
}

public void MoveAndRotateArrowHead()
{
    //Direction of Arrow's Tail
    dir = tailLR.GetPosition(0) - tailLR.GetPosition(1);

    //Positioning Arrow's Head and Rotate it based on Tail Direction
    head.transform.localPosition = tailLR.GetPosition(1);
    head.transform.localPosition = new Vector3(head.transform.localPosition.x,
                                                head.transform.localPosition.y, 0f);
    head.transform.localRotation = Quaternion.Euler(new Vector3(0f, 0f,
                                                               Quaternion.FromToRotation(Vector3.up, dir).eulerAngles.z));
}

public void SetTailPos(int id, Vector3 pos)
{
    arPos = new Vector3(pos.x, pos.y, 0f) - transform.position;
    arPos.z = 0f;
    if (tailLR != null)
        tailLR.SetPosition(id, arPos);
}

//Snapp Arrow to the Target Position
void SnapArrow(Transform target)
{
    nextCommand = target;
    childImg.color = snappedColor;
    target.GetChild(0).GetComponent<Image>().color = snappedColor;
    SnapTo = target.GetComponent<ConnectorButton>();
    SnapTo.whoSnapToMe = this;
    MoveArrow(target.position);
    snapped = true;
}

```

```

public void OnEndDrag(PointerEventData eventData)
{
    if(dragable)
    {
        foreach (GameObject con in eventData.hovered)
        {
            snapped = false;
            if (con.GetComponent<ConnectorButton>() != null && con != gameObject)
            {
                if (con.transform.parent != transform.parent &&
                    !con.GetComponent<ConnectorButton>().dragable &&
                    !con.GetComponent<ConnectorButton>().snapped)
                {
                    SnapArrow(con.transform);
                    break;
                }
            }
        }

        if(!snapped)
        {
            Destroy(tail);
        }
    }
}

public void OnPointerEnter(PointerEventData eventData)
{
    img.color = Color.white;
}

public void OnPointerExit(PointerEventData eventData)
{
    img.color = new Color(1f, 1f, 1f, .4f);
}

public Transform GetNextCommand()
{
    if (dragable)
        return nextCommand;
    else
        return null;
}

private void OnDestroy()
{
    if(whoSnapToMe != null)
    {
        whoSnapToMe.childImg.color = whoSnapToMe.freeColor;
        whoSnapToMe.snapped = false;
        Destroy(whoSnapToMe.tail);
    }
    if(SnapTo != null)
    {
        SnapTo.whoSnapToMe = null;
        SnapTo.childImg.color = SnapTo.freeColor;
        Destroy(tail);
    }
}
}

```

Credit.cs

```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;

public class Credit : MonoBehaviour
{
    public GameObject loadingCanvas;
    public Image loadingPanel;
    public TextMeshProUGUI loadingText;

    public void LoadLevel(string name)
    {
        StartCoroutine(Load(name));
    }

    IEnumerator Load(string name)
    {
        AsyncOperation async = SceneManager.LoadSceneAsync(name);

        async.allowSceneActivation = false;

        loadingCanvas.SetActive(true);

        while (!async.isDone)
        {
            loadingPanel.color = new Color(0f, 0f, 0f, async.progress);
            loadingText.text = "Progres : " + (async.progress * 100) + "%";
            if (async.progress >= .9f)
            {
                loadingPanel.color = new Color(0f, 0f, 0f, 1f);
                loadingText.text = "Bersiap....";
                yield return new WaitForSeconds(3f);
                async.allowSceneActivation = true;
            }
            yield return null;
        }

        yield return null;
    }
}
```

Decision.cs

```
using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;

public class Decision : MonoBehaviour
{

    public enum Type
    {
        Place,
        Item,
        Placement,
    }
```

```

        Counter,
        Interactable
    };
    public Type type;
    public string identifier;

    public void ChangeIdentifier(TextMeshProUGUI label)
    {
        if(type.Equals(Type.Place))
        {
            if(label.text.Equals("Portal"))
            {
                identifier = label.text;
                return;
            }

            if(label.text.Equals("Jalan"))
            {
                identifier = "Walk";
                return;
            }

            if (label.text.Equals("Lompat ke Atas"))
            {
                identifier = "Jump_Up";
                return;
            }

            if (label.text.Equals("Lompat ke Bawah"))
            {
                identifier = "Jump_Down";
                return;
            }

            if (label.text.Equals("Lompat ke Depan"))
            {
                identifier = "Jump_Forward";
                return;
            }
        }

        identifier = label.text;
    }
}

```

DropdownFixer.cs

```

using System.Collections;
using UnityEngine;
using UnityEngine.Events;

public class DropdownFixer : MonoBehaviour,
    IPointerUpHandler
{
    public void OnPointerUp(PointerEventData eventData)
    {
        StartCoroutine(Find("Dropdown List"));
    }

    IEnumerator Find(string name)
    {

```

```

        yield return new WaitForEndOfFrame();

        Transform child = transform.Find(name);
        if(child != null)
        {
            child.GetComponent<Canvas>().sortingLayerID = SortingLayer.NameToID("UI");
        }

        child = GameObject.Find("Blocker").transform;
        if (child != null)
        {
            child.GetComponent<Canvas>().sortingLayerID = SortingLayer.NameToID("UI");
        }

        yield return null;
    }
}

```

GameState.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu()]
public class GameState : ScriptableObject
{

    public bool playingMode;
    public bool editingMode;
    public bool finishedPlaying;
    public bool paused;
    public int quality;
    public bool showHowTo;
    public string username;

}

```

Highscore.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;

public class Highscore : MonoBehaviour
{

    const string privateCode = "xpPtV4_xmEyEThb6UN5Tpg-aoG2zeO-0S0AzXXn9Wjig";
    const string publicCode = "5ae58bfc191a840bccca981dd";
    const string webURL = "http://dreamlo.com/lb/";

    [HideInInspector]
    public LeaderBoard m_scoreBoard;

    private void Awake()
    {
        DownloadHighscores();
    }
}

```

```

public void AddNewHighScore(string username, int score, int levelIndex)
{
    StartCoroutine(UploadNewHighscore(username, score, levelIndex));
}

IEnumerator UploadNewHighscore(string username, int score, int levelIndex)
{
    WWW www = new WWW(webURL + privateCode + "/add/" + WWW.EscapeURL(username)
+ "-" + levelIndex + "/" + score);

    yield return www;

    if (string.IsNullOrEmpty(www.error))
        Debug.Log("Upload Successful");
    else
    {
        Debug.LogError("Error uploading: " + www.error);
    }
}

public void DownloadHighscores()
{
    StartCoroutine("DownloadHighscoresFromDatabase");
}

IEnumerator DownloadHighscoresFromDatabase()
{
    WWW www = new WWW(webURL + publicCode + "/json/");
    yield return www;

    if (string.IsNullOrEmpty(www.error))
    {
        string json = RemoveNJsonFields(www.text, 2);
        m_scoreBoard = JsonUtility.FromJson<LeaderBoard>(json);
        Debug.Log("Download Success");
    }
    else
    {
        Debug.LogError("Error Downloading: " + www.error);
    }
}

public List<HighscoreData> RequestHighscoreList(Entry[] entries, int index)
{
    List<HighscoreData> highscores = new List<HighscoreData>();

    if (entries != null)
        foreach (Entry entry in entries)
        {
            string[] split = entry.name.Split('-');
            int id = int.Parse(split[1]);
            if(id.Equals(index))
            {
                HighscoreData hd = new HighscoreData(split[0], entry.score, id);
                highscores.Add(hd);
            }
        }

    return highscores;
}

private string RemoveNJsonFields(string source, int n)

```

```

        {
            n++;

            int index = source.TakeWhile(c => (n == (c == '{' ? 1 : 0)) > 0).Count();

            return source.Substring(index, source.Length - (index + 2 - n));
        }
    }

[Serializable]
public class LeaderBoard
{
    public Entry[] entry;
}

[Serializable]
public class Entry
{
    public string name;
    public int score;
    public int seconds;
    public string text;
    public DateTime date;
}

public class HighscoreData
{
    public string username;
    public int score;
    public int levelIndex;

    public HighscoreData(string username, int score, int levelIndex)
    {
        this.username = username;
        this.score = score;
        this.levelIndex = levelIndex;
    }
}

public class AscendingHighscore : IComparer<HighscoreData>
{
    public int Compare(HighscoreData x, HighscoreData y)
    {
        if(x == null)
        {
            if(y == null)
            {
                return 0;
            }
            else
            {
                return -1;
            }
        }
        else
        {
            if(y == null)
            {
                return 1;
            }
            else
        }
    }
}

```

```

        {
            if(x.score < y.score)
            {
                return -1;
            }
            else if (x.score == y.score)
            {
                return 0;
            }
            else
            {
                return 1;
            }
        }
    }
}

public class DescendingHighscore : IComparer<HighscoreData>
{
    public int Compare(HighscoreData x, HighscoreData y)
    {
        if (x == null)
        {
            if (y == null)
            {
                return 0;
            }
            else
            {
                return 1;
            }
        }
        else
        {
            if (y == null)
            {
                return -1;
            }
            else
            {
                if (x.score < y.score)
                {
                    return 1;
                }
                else if (x.score == y.score)
                {
                    return 0;
                }
                else
                {
                    return -1;
                }
            }
        }
    }
}

```

HighscorePanel.cs

```
using UnityEngine;
using System.Collections.Generic;
using TMPro;
using UnityEngine.UI;

public class HighscorePanel : MonoBehaviour
{
    public Highscore highscore;
    public GameObject bodyTemplate;
    public GameObject parent;
    public LevelProperty[] levelProperties;
    public TextMeshProUGUI title;
    public Button nextBtn;
    public Button prevBtn;
    public TMP_InputField inputField;

    private List<GameObject> bodies;

    private int currentIndex;

    private void Awake()
    {
        currentIndex = 1;
        bodies = new List<GameObject>();
        OpenHighscorePage(currentIndex.ToString());
    }

    private void OnEnable()
    {
        highscore.DownloadHighscores();
    }

    public void OpenHighscorePage(string index)
    {
        int id = int.Parse(index);

        if (id >= levelProperties.Length)
        {
            id = levelProperties.Length - 1;
            inputField.text = id.ToString();
        } else if(id == 0)
        {
            id = 1;
            inputField.text = id.ToString();
        }

        title.text = levelProperties[id-1].levelName;

        foreach (GameObject b in bodies)
        {
            Destroy(b);
        }

        bodies.Clear();

        List<HighscoreData> datas = highscore.RequestHighscoreList(highscore.m_scoreBoard.entry, id);
        AscendingHighscore asc = new AscendingHighscore();
        datas.Sort(asc);

        foreach (HighscoreData d in datas)
        {
            GameObject go = Instantiate(bodyTemplate, parent.transform);
            Body body = go.GetComponent<Body>();
            body.username.text = d.username;
            body.score.text = d.score.ToString();
            go.SetActive(true);
            bodies.Add(go);
        }

        currentIndex = id;
    }
}
```

```

        checkButton(currentId);
    }

    public void Page(int i)
    {
        if(currentId < levelProperties.Length && currentId > 0)
        {
            currentId = i > 0 ? currentId + 1 : currentId - 1;
            OpenHighscorePage(currentId.ToString());
            title.text = "Level : " + levelProperties[currentId - 1].levelName;
            inputField.text = currentId.ToString();
        }
        checkButton(currentId);
    }

    void checkButton(int i)
    {
        if (i == levelProperties.Length - 1)
            nextBtn.interactable = false;
        else
            nextBtn.interactable = true;

        if (i == 1)
            prevBtn.interactable = false;
        else
            prevBtn.interactable = true;
    }
}

```

Hint.cs

```

using UnityEngine;

public class Hint : MonoBehaviour
{
    public GameState game;

    private void Start()
    {
        game.editingMode = true;
    }

    private void OnEnable()
    {
        game.editingMode = true;
    }

    private void OnDisable()
    {
        game.editingMode = false;
    }
}

```

HowToPanel.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class HowToPanel : MonoBehaviour
{

```

```

public GameState gameState;

public GameObject OKButton;
public Toggle[] toggles;
private int _id;

public int Id
{
    get
    {
        return _id;
    }
    set
    {
        _id = value;
        if (_id >= toggles.Length) _id = 0;
        if (_id < 0) _id = toggles.Length - 1;
        Debug.Log(value);
    }
}

private void Awake()
{
    Id = 0;
    toggles[Id].isOn = true;
}

public void Plus()
{
    Id = _id + 1;
    toggles[Id].isOn = true;
}

public void Minus()
{
    Id = _id - 1;
    toggles[Id].isOn = true;
}

public void ToggleChange(int id)
{
    Id = id;
}

public void ShowHowTo(bool value)
{
    gameState.showHowTo = value ? false : true;
}
}

```

Information.cs

```

[CreateAssetMenu()]
public class Information : ScriptableObject
{
    [TextArea]
    public string text;
}

```

Interactable.cs

```
using UnityEngine;

public class Interactable : MonoBehaviour
{
    public PathNode node;
    public InteractableObject interactable;
    public Animator animator;

    private bool isOn;

    private void Start()
    {
        if (interactable != null)
            if (interactable.Animator != null)
                animator.SetBool("isOn", interactable.Animator.GetBool("On"));
            else
                animator.SetBool("isOn", false);

        isOn = animator.GetBool("isOn");
        node.interactable = this;
    }

    public void Interact()
    {
        if (interactable != null)
        {
            isOn = !isOn;
            interactable.Interacted(isOn);
            animator.SetBool("isOn", isOn);
        }
        else
        {
            isOn = !isOn;
            animator.SetBool("isOn", isOn);
        }
    }
}
```

InteractableObject.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InteractableObject : MonoBehaviour {

    public PathNode[] nodes;
    public Animator animator;
    public bool inverted = false;

    private void Start()
    {
        if (animator != null)
            Interacted(animator.GetBool("On"));

    }
}
```

```

        else
            Interacted(false);
    }

    public void Interacted(bool condition)
    {
        if (animator != null) animator.SetBool("On", condition);
        foreach (PathNode node in nodes)
        {
            node.Active = inverted ? !condition : condition;
        }
    }

    public Animator Animator
    {
        get
        {
            return animator;
        }
    }
}

```

Item.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu()]
public class Item : ScriptableObject
{
    public string itemName;
}

```

ItemSystem.cs

```

using UnityEngine;

public class ItemSystem : MonoBehaviour
{
    public Obstacle affectedObject;
    public bool directEffect;
    public PathNode node;
    public Item item;

    private void Awake()
    {
        node.item = this;
    }

    public void TakeItem()
    {
        gameObject.SetActive(false);
    }
}

```

LevelButton.cs

```
using UnityEngine;
using TMPro;
using UnityEngine.UI;

public class LevelButton : MonoBehaviour
{

    public TextMeshProUGUI text;
    public LevelProgress level;
    public Button nextButton;
    public Image nextImage;
    public Sprite spr;
    public Button[] buttons;

    private void Start()
    {
        text.text = level.progress + " / " + level.maxLevel +
        "\nTotal Blok Digunakan : " + level.score;

        for (int i = 0; i < buttons.Length; i++)
        {
            buttons[i].interactable = true;
        }
    }
}
```

LevelManager.cs

```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;

public class LevelManager : MonoBehaviour
{

    public GameObject pauseMenuPanel;
    public GameObject loadingCanvas;
    public TextMeshProUGUI loadingText;
    public Image loadingPanel;
    public GameState gameState;
    public PlayerAttribute player;
    public SaveLoad saveLoad;

    private bool paused;
    private int m_playerHp;

    private void Start()
    {
        gameState.paused = false;
        saveLoad.Load();
        m_playerHp = player.hp;
        paused = false;
    }

    void Update ()
    {
```

```

        if(Input.GetKeyDown(KeyCode.Escape) && !gameState.finishedPlaying)
    {
        Pause();
    }
}

public void Pause()
{
    if (!paused)
    {
        Time.timeScale = 0f;
        pauseMenuPanel.SetActive(true);
    }
    else
    {
        Time.timeScale = 1f;
        pauseMenuPanel.SetActive(false);
    }
    paused = !paused;
    gameState.paused = paused;
}

public void LoadLevel(string name)
{
    if (paused) Time.timeScale = 1f;
    StartCoroutine(Load(name));
}

IEnumerator Load(string name)
{
    saveLoad.Save();

    AsyncOperation async = SceneManager.LoadSceneAsync(name);
    async.allowSceneActivation = false;

    loadingCanvas.SetActive(true);

    if (SceneManager.GetActiveScene().name == name)
    {
        player.hp = m_playerHp;
    }

    while(!async.isDone)
    {
        loadingPanel.color = new Color(0f, 0f, 0f, async.progress);
        loadingText.text = "Progres : " + (async.progress * 100) + "%";
        if(async.progress >= .9f)
        {
            loadingPanel.color = new Color(0f, 0f, 0f, 1f);
            loadingText.text = "Bersiap....";
            yield return new WaitForSeconds(3f);
            async.allowSceneActivation = true;
        }
        yield return null;
    }

    yield return null;
}

```

```

        public void Restart()
    {
        StartCoroutine(Load(SceneManager.GetActiveScene().name));
    }

        public void Quit()
    {
        saveLoad.Save();
        Application.Quit();
    }
}

```

LevelProgress.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu()]
public class LevelProgress : ScriptableObject
{
    public string levelName;
    public int progress;
    public int maxLevel;
    public int score;
}

```

LevelProperty.cs

```

using UnityEngine;

[CreateAssetMenu()]
public class LevelProperty : ScriptableObject
{
    public string levelName;
    public int algUsed;
    public bool completed;
    public int levelId;
}

```

LevelSelectManager.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
using TMPro;

public class LevelSelectManager : MonoBehaviour
{
    public LevelProgress castle;
    public LevelProgress forest;
    public LevelProgress swamp;
    public LevelProgress glitch;

```

```

public GameObject loadingCanvas;
public Image loadingPanel;
public TextMeshProUGUI loadingText;

public void LoadLevel(LevelProgress level)
{
    if (level == castle)
    {
        if (level.progress < 1)
        {
            StartCoroutine(Loading("Castle"));
        } else if (level.progress >= 1)
        {
            StartCoroutine(Loading("2-2"));
        }
    }
    else if (level == forest)
    {
        if (level.progress < 1)
        {
            StartCoroutine(Loading("3-1"));
        } else if (level.progress <= 1)
        {
            StartCoroutine(Loading("3-2"));
        }
        else if (level.progress <= 2)
        {
            StartCoroutine(Loading("3-3"));
        }
        else if (level.progress <= 3)
        {
            StartCoroutine(Loading("3-4"));
        }
        else if (level.progress <= 4)
        {
            StartCoroutine(Loading("3-5"));
        }
        else if (level.progress <= 5)
        {
            StartCoroutine(Loading("3-6"));
        }
        else if (level.progress <= 6)
        {
            StartCoroutine(Loading("3-7"));
        }
        else if (level.progress <= 7)
        {
            StartCoroutine(Loading("3-8"));
        }
        else if (level.progress <= 8)
        {
            StartCoroutine(Loading("3-9"));
        }
        else if (level.progress <= 9)
        {
            StartCoroutine(Loading("3-10"));
        }
    }
    else if (level.progress >= 10)

```

```

    {
        StartCoroutine(Loading("3-11"));
    }
}
else if (level == swamp)
{
    if (level.progress < 1)
    {
        StartCoroutine(Loading("4-1"));
    }
    else if (level.progress <= 1)
    {
        StartCoroutine(Loading("4-2"));
    }
    else if (level.progress <= 2)
    {
        StartCoroutine(Loading("4-3"));
    }
    else if (level.progress <= 3)
    {
        StartCoroutine(Loading("4-4"));
    }
    else if (level.progress <= 4)
    {
        StartCoroutine(Loading("4-5"));
    }
    else if (level.progress <= 5)
    {
        StartCoroutine(Loading("4-6"));
    }
    else if (level.progress <= 6)
    {
        StartCoroutine(Loading("4-7"));
    }
    else if (level.progress >= 7)
    {
        StartCoroutine(Loading("4-8"));
    }
}
else if (level == glitch)
{
    if (level.progress < 1)
    {
        StartCoroutine(Loading("5-1-1"));
    }
    else if (level.progress <= 1)
    {
        StartCoroutine(Loading("5-1-2"));
    }
    else if (level.progress <= 2)
    {
        StartCoroutine(Loading("5-2"));
    }
    else if (level.progress <= 3)
    {
        StartCoroutine(Loading("5-3"));
    }
    else if (level.progress <= 4)

```

```

        {
            StartCoroutine(Loading("5-4"));
        }
        else if (level.progress >= 5)
        {
            StartCoroutine(Loading("5-5"));
        }
    }
}

public void LoadLevel(string name)
{
    StartCoroutine(Loading(name));
}

IEnumerator Loading(string name)
{
    AsyncOperation async = SceneManager.LoadSceneAsync(name);

    async.allowSceneActivation = false;

    loadingCanvas.SetActive(true);

    while (!async.isDone)
    {
        loadingPanel.color = new Color(0f, 0f, 0f, async.progress);
        loadingText.text =
        "Progres : " + (async.progress * 100).ToString("0.0") + "%";
        if (async.progress >= .9f)
        {
            loadingPanel.color = new Color(0f, 0f, 0f, 1f);
            loadingText.text = "Bersiap.....";
            yield return new WaitForSeconds(2f);
            async.allowSceneActivation = true;
        }
        yield return null;
    }

    yield return null;
}
}

```

MenuManager.cs

```

using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
using UnityEngine.UI;

public class MenuManager : MonoBehaviour
{
    public GameObject loadingCanvas;
    public Image loadingPanel;
    public TextMeshProUGUI loadingText;
    public GameState game;
    public SaveLoad saveLoad;

    public Audio m_audio;

    public GameObject nameInput;

```

```

private void Start()
{
    if (saveLoad != null)
        saveLoad.Load();
    if(game != null)
        QualitySettings.SetQualityLevel(game.quality, true);
    if (game.username == "")
        nameInput.SetActive(true);
}

public void PlayButton()
{
    StartCoroutine(Load("LevelSelect"));
}

IEnumerator Load(string name)
{
    AsyncOperation async = SceneManager.LoadSceneAsync(name);

    async.allowSceneActivation = false;

    loadingCanvas.SetActive(true);

    if(saveLoad != null)
        saveLoad.Save();

    while (!async.isDone)
    {
        loadingPanel.color = new Color(0f, 0f, 0f, async.progress);
        loadingText.text = "Progres : " + (async.progress * 100).ToString("0.0") + "%";
        if (async.progress >= .9f)
        {
            loadingPanel.color = new Color(0f, 0f, 0f, 1f);
            loadingText.text = "Bersiap..";
            yield return new WaitForSeconds(2f);
            async.allowSceneActivation = true;
        }
        yield return null;
    }

    yield return null;
}

public void LoadLevel(string name)
{
    StartCoroutine(Load(name));
}

public void Quit()
{
    Application.Quit();
}
}

```

NameInput.cs

```

using UnityEngine;
using TMPro;
using System.Linq;
using System.Text.RegularExpressions;

public class NameInput : MonoBehaviour
{
    public GameState game;
    public GameObject warning;
    public TMP_InputField textInput;

```

```

private void OnEnable()
{
    textInput.text = game.username;
}

public void SetUsername(TMP_InputField text)
{
    Regex regex = new Regex("^[a-zA-Z]*$");
    if(text.text.Length < 2 || text.text.Length > 9 || !regex.IsMatch(text.text))
    {
        warning.SetActive(true);
        warning.GetComponent<Animator>().Play("warning", -1, 0f);
    }
    else
    {
        game.username = text.text;
        gameObject.SetActive(false);
    }
}
}

```

NameLabel.cs

```

using UnityEngine;
using TMPro;

public class NameLabel : MonoBehaviour
{
    public GameState game;

    private void OnEnable()
    {
        GetComponent<TextMeshProUGUI>().text = game.username;
    }
}

```

NPC.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class NPC : MonoBehaviour
{

    public Information[] informations;
    public Transform dialogPanel;
    public TextMeshProUGUI text;
    public TextMeshProUGUI npcname;
    public PathNode node;
    [HideInInspector] public bool talking;

    private bool clicked;

    private void Start()
    {
        clicked = false;
        talking = false;
        node.npc = this;
    }
}

```

```

        }

    public void InteractWith()
    {
        talking = true;
        StartCoroutine(Dialog());
    }

    IEnumerator Dialog()
    {
        dialogPanel.gameObject.SetActive(true);
        npcname.text = transform.name;
        foreach(Information info in informations)
        {
            text.text = info.text;
            clicked = false;
            while(!clicked)
            {
                yield return null;
            }
        }
        dialogPanel.gameObject.SetActive(false);
        talking = false;
    }

    public void Click()
    {
        clicked = true;
    }
}

```

Obstacle.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Obstacle : MonoBehaviour {

    public int keyNeeded;
    public enum Type { Trap, Block };
    public Type type;
    public PathNode node;
    public Item item;

    private Animator animator;
    private int keyCount = 0;

    private void Awake()
    {
        animator = GetComponent<Animator>();
        if (node != null)
            node.obstacle = this;
        else
            Debug.LogWarning("Node not assigned");
    }
}

```

```

public bool KeyTaken()
{
    keyCount++;

    if (animator != null)
    {
        animator.SetInteger("Key", keyCount);
    }

    if (keyCount >= keyNeeded)
    {
        return true;
    } else
    {
        if(animator != null)
        {
            animator.SetInteger("Key", keyCount);
        }
        return false;
    }
}

public void ItemUsed(Item item)
{
    if(item == this.item)
    {
        Destroy(gameObject);
    }
}
}

```

OpeningLevelLoader.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class OpeningLevelLoader : MonoBehaviour
{
    AsyncOperation async;

    void Start()
    {
        async = SceneManager.LoadSceneAsync("MainMenu");
        async.allowSceneActivation = false;
    }

    void Go()
    {
        async.allowSceneActivation = true;
    }
}

```

OpeningStory.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class OpeningStory : MonoBehaviour
{
    public Information[] information;
    public GameObject dialogPanel;
    public TextMeshProUGUI text;

    private Animator animator;
    private int id;
    private void Start()
    {
        animator = GetComponent<Animator>();
        id = 0;
        dialogPanel.SetActive(true);
        text.text = information[id].text;
    }

    public void Next()
    {
        id++;
        if(id < information.Length)
        {
            text.text = information[id].text;
            if(animator != null && !animator.GetBool("Play"))
            {
                animator.SetBool("Play", true);
            }
            else
            {
                dialogPanel.SetActive(false);
            }
        }
    }
}
```

PathNode.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PathNode : MonoBehaviour
{
    public ItemSystem item;
    public Obstacle obstacle;
    public Interactable interactable;
    public NPC npc;
    public Enemy enemy;
    public Transform jumpUpR;
    public Transform jumpFR;
    public Transform jumpDropR;
```

```

public Transform jumpUpL;
public Transform jumpFL;
public Transform jumpDropL;
public Transform walkRight;
public Transform walkLeft;
public bool isFinishNode;

[SerializeField]
private bool active;

public bool Active
{
    get
    {
        return active;
    }
    set
    {
        if(value)
        {
            GetComponent<SpriteRenderer>().color = Color.yellow;
            active = value;
        }
        else
        {
            GetComponent<SpriteRenderer>().color = Color.gray;
            active = value;
        }
    }
}

Transform child;
Color oldColor;
Color activeColor;

void Awake()
{
    Active = true;
    child = transform.GetChild(0);
    oldColor = child.GetComponent<SpriteRenderer>().color;
}

private void OnMouseEnter()
{
    child.GetComponent<SpriteRenderer>().color = new Color(1f, 1f, 1f, 1f);
}

private void OnMouseExit()
{
    child.GetComponent<SpriteRenderer>().color = oldColor;
}

}

```

PlayerAttribute.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[CreateAssetMenu()]
public class PlayerAttribute : ScriptableObject
{
    [SerializeField]
    [Range(0, 10)]
    private int m_hp;
    public int score;
    public string level_title;

    public int hp
    {
        get
        {
            return m_hp;
        }
        set
        {
            if(value >= 10)
            {
                value = 10;
            }
            m_hp = value;
        }
    }
}
```

PlayerAttributeProperty.cs

```
using UnityEngine;
using TMPro;
using System.Collections;

public class PlayerAttributeProperty : MonoBehaviour
{

    public PlayerAttribute player;
    public GameObject[] hps;
    public TextMeshProUGUI scoreText;
    public TextMeshProUGUI level_title;

    int lastHp;

    private void Start()
    {
        SetHP();
        StartCoroutine(SetLevelTitle());
    }

    private void Update()
    {
        if(lastHp != player.hp)
        {
            SetHP();
        }
    }
}
```

```

        scoreText.text = "Total Blok : " + player.score;
    }

IEnumerator SetLevelTitle()
{
    while (player.level_title == "")
    {
        yield return null;
    }

    level_title.text = player.level_title;
    player.level_title = "";
}

void SetHP()
{
    int x = 1;
    for (int i = 0; i < hps.Length; i++)
    {
        if (x <= player.hp) hps[i].SetActive(true);
        else hps[i].SetActive(false);
        x++;
    }
    lastHp = player.hp;
}
}

```

QualityDropdown.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using System.Linq;

public class QualityDropdown : MonoBehaviour
{
    private TMP_Dropdown dropdown;
    public GameState game;

    // Use this for initialization
    void Start ()
    {
        dropdown = GetComponent<TMP_Dropdown>();
        dropdown.ClearOptions();
        List<string> qList = QualitySettings.names.OfType<string>().ToList();
        dropdown.AddOptions(qList);
        dropdown.value = QualitySettings.GetQualityLevel();
        Debug.Log(QualitySettings.GetQualityLevel());
    }

    public void ChangeQuality()
    {
        QualitySettings.SetQualityLevel(dropdown.value, true);
        game.quality = dropdown.value;
    }
}

```

ResolutionAdaptation.cs

```
using UnityEngine;

public class ResolutionAdaptation : MonoBehaviour
{
    void Awake ()
    {
        int maxH = 0;
        int maxW = 0;

        Resolution[] res = Screen.resolutions;

        foreach(Resolution r in res)
        {
            if (maxH < r.height) maxH = r.height;
            if (maxW < r.width) maxW = r.width;
        }

        int m_w = maxH / 9 * 16;

        if(m_w < maxW)
        {
            Screen.SetResolution(m_w, maxH, true);
        }
        else
        {
            int m_h = maxW / 16 * 9;
            Screen.SetResolution(maxW, m_h, true);
        }
    }
}
```

SaveLoad.cs

```
using System;
using System.Collections.Generic;
using UnityEngine;
using System.IO;
using System.Linq.Expressions;
using System.Linq;

public class SaveLoad : MonoBehaviour
{
    public List<GameState> gameStates;
    public List<Audio> audios;
    public List<LevelProgress> levelProgresses;
    public List<LevelProperty> levelProperties;

    public void Save()
    {
        List<DataHolder> dataHolders = new List<DataHolder>();
        foreach (GameState item in gameStates)
        {
            DataHolder dataHolder = new DataHolder(item.name, item.GetInstanceID());
            Data data = new Data(MemberInfoGetting.GetMemberName(() => item.quality), item.quality);
            Data data1 = new Data(MemberInfoGetting.GetMemberName(() => item.showHowTo), item.showHowTo);
            Data data2 = new Data(MemberInfoGetting.GetMemberName(() => item.username), item.username);
            dataHolder.Add(data);
            dataHolder.Add(data1);
            dataHolder.Add(data2);
            dataHolders.Add(dataHolder);
        }
    }
}
```

```

foreach (Audio a in audios)
{
    DataHolder dataHolder = new DataHolder(a.name, a.GetInstanceID());
    Data data = new Data(MemberInfoGetting.GetMemberName(() -> a.value), a.value);
    Data data1 = new Data(MemberInfoGetting.GetMemberName(() => a.enabled), a.enabled);
    dataHolder.Add(data);
    dataHolder.Add(data1);
    dataHolders.Add(dataHolder);
}

foreach (LevelProgress lp in levelProgresses)
{
    DataHolder dataHolder = new DataHolder(lp.name, lp.GetInstanceID());
    Data data = new Data(MemberInfoGetting.GetMemberName(() -> lp.progress), lp.progress);
    Data data1 = new Data(MemberInfoGetting.GetMemberName(() -> lp.score), lp.score);
    dataHolder.Add(data);
    dataHolder.Add(data1);
    dataHolders.Add(dataHolder);
}

foreach (LevelProperty lp in levelProperties)
{
    DataHolder dataHolder = new DataHolder(lp.name, lp.GetInstanceID());
    Data data = new Data(MemberInfoGetting.GetMemberName(() -> lp.algUsed), lp.algUsed);
    Data data1 = new Data(MemberInfoGetting.GetMemberName(() -> lp.completed), lp.completed);
    dataHolder.Add(data);
    dataHolder.Add(data1);
    dataHolders.Add(dataHolder);
}

string jsonData = JsonHelper.ToString(dataHolders.ToArray());

if(!Directory.Exists(Application.dataPath + "/Resources"))
    Directory.CreateDirectory(Application.dataPath + "/Resources");

File.WriteAllText(Application.dataPath + "/Resources/playerdata.json", jsonData);
}

public void Load()
{
    if (File.Exists(Application.dataPath + "/Resources/playerdata.json"))
    {
        string jsonData = File.ReadAllText(Application.dataPath + "/Resources/playerdata.json");

        List<DataHolder> datas = JsonHelper.FromJson<DataHolder>(jsonData).ToList();

        foreach (DataHolder data in datas)
        {
            foreach (GameState item in gameStates)
            {
                if (data.name == item.name || data.id == item.GetInstanceID())
                {
                    foreach (Data d in data.datas)
                    {
                        if (d.name == MemberInfoGetting.GetMemberName(() -> item.quality))
                        {
                            item.quality = int.Parse(d.value);
                            continue;
                        }
                        if (d.name == MemberInfoGetting.GetMemberName(() -> item.showHowTo))
                        {
                            item.showHowTo = bool.Parse(d.value);
                            continue;
                        }
                        if (d.name == MemberInfoGetting.GetMemberName(() -> item.username))
                        {
                            item.username = d.value;
                            continue;
                        }
                    }
                }
            }
        }
    }
}

```

```

foreach (Audio item in audios)
{
    if (data.name == item.name || data.id == item.GetInstanceID())
    {
        foreach (Data d in data.datas)
        {
            if (d.name == MemberInfoGetting.GetMemberName(() => item.value))
            {
                item.value = float.Parse(d.value);
                continue;
            }
            if (d.name == MemberInfoGetting.GetMemberName(() => item.enabled))
            {
                item.enabled = Boolean.Parse(d.value);
                continue;
            }
        }
    }
}

foreach (LevelProgress item in levelProgresses)
{
    if (data.name == item.name || data.id == item.GetInstanceID())
    {
        foreach (Data d in data.datas)
        {
            if (d.name == MemberInfoGetting.GetMemberName(() => item.progress))
            {
                item.progress = int.Parse(d.value);
                continue;
            }
            if (d.name == MemberInfoGetting.GetMemberName(() => item.score))
            {
                item.score = int.Parse(d.value);
                continue;
            }
        }
    }
}

foreach (LevelProperty item in levelProperties)
{
    if (data.name == item.name || data.id == item.GetInstanceID())
    {
        foreach (Data d in data.datas)
        {
            if (d.name == MemberInfoGetting.GetMemberName(() => item.algUsed))
            {
                item.algUsed = int.Parse(d.value);
                continue;
            }
            if (d.name == MemberInfoGetting.GetMemberName(() => item.completed))
            {
                item.completed = bool.Parse(d.value);
                continue;
            }
        }
    }
}

[Serializable]
public class DataHolder
{
    public string name;
    public int id;
    public List<Data> datas = new List<Data>();

    public DataHolder(string name, int id)
    {
}

```

```

        this.name = name;
        this.id = id;
    }

    public void Add(Data data)
    {
        datas.Add(data);
    }

    public List<Data> GetDatas()
    {
        return datas;
    }
}

[Serializable]
public class Data
{
    public string name;
    public string value;

    public Data (string name, int value)
    {
        this.name = name;
        this.value = value.ToString();
    }

    public Data (string name, bool value)
    {
        this.name = name;
        this.value = value.ToString();
    }

    public Data (string name, float value)
    {
        this.name = name;
        this.value = value.ToString();
    }

    public Data(string name, string value)
    {
        this.name = name;
        this.value = value.ToString();
    }
}

public static class MemberInfoGetting
{
    public static string GetMemberName<T>(Expression<Func<T>> memberExpression)
    {
        MemberExpression expressionBody = (MemberExpression)memberExpression.Body;
        return expressionBody.Member.Name;
    }
}

```

SceneInformation.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;

public class SceneInformation : MonoBehaviour
{
    public TextMeshProUGUI text;

    [TextArea] public string information;

    private Image img;

```

```

private void Start()
{
    text.text = information;
    img = GetComponent<Image>();
    StartCoroutine(infoShow());
}

IEnumerator infoShow()
{
    text.color = Color.clear;
    img.color = Color.clear;

    float time = 0f;
    yield return new WaitForSeconds(1f);
    while(time < 1f)
    {
        Color b = Color.Lerp(new Color(1f,1f,1f,0f), Color.white, time);
        text.color = b;
        Color c = Color.Lerp(new Color(0f,0f,0f,0f), new Color(0f,0f,0f,.5f), time);
        img.color = c;
        time += Time.deltaTime * .5f;
        yield return null;
    }
    text.color = Color.white;
    img.color = new Color(0f,0f,0f,.5f);
    yield return new WaitForSeconds(2f);
    time = 0f;
    while (time < 1f)
    {
        Color b = Color.Lerp(Color.white, new Color(1f,1f,1f,0f), time);
        text.color = b;
        Color c = Color.Lerp(new Color(0f,0f,0f,.5f), new Color(0f,0f,0f,0f), time);
        img.color = c;
        time += Time.deltaTime * .5f;
        yield return null;
    }
    gameObject.SetActive(false);
}
}

```

TrashPanel.cs

```

using System.Reflection;
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.UI;

public class TrashPanel : MonoBehaviour,
    IDropHandler,
    IPointerEnterHandler,
    IPointerExitHandler
{
    public GameObject buttonHolder;
    public Color highlightColor;

    private Image image;
    private Color originalColor;
    private GameObject d;

```

```

void Awake()
{
    image = GetComponent<Image>();
    originalColor = image.color;
}

private void OnEnable()
{
    buttonHolder.SetActive(false);
    image.color = originalColor;
}

private void OnDisable()
{
    buttonHolder.SetActive(true);
}

public void OnDrop(PointerEventData eventData)
{
    if (d == null) return;

    Destroy(d);

    gameObject.SetActive(false);
}

GameObject GetDraggedAlgorithm(PointerEventData data)
{
    GameObject dragged = data.pointerDrag;

    if(dragged == null)
        return null;

    return dragged;
}

public void OnPointerEnter(PointerEventData eventData)
{
    d = GetDraggedAlgorithm(eventData);

    if (d == null) return;

    image.color = highlightColor;
}

public void OnPointerExit(PointerEventData eventData)
{
    if (d == null) return;

    image.color = originalColor;
}
}

```

WinPanel.cs

```
using UnityEngine;
using TMPro;

public class WinPanel : MonoBehaviour
{
    public TextMeshProUGUI[] user;
    public TextMeshProUGUI[] score;

}
```

**Lampiran 11. Perhitungan Hasil Pengujian oleh Pengembang
Permainan**

1. Compatibility testing

Nomor Butir	Penguji 1	Penguji 2	Total Skor
1	1	1	2
2	1	1	2
3	1	1	2
4	1	1	2
5	1	1	2
6	1	1	2
7	1	1	2
8	1	1	2
9	1	1	2
10	1	1	2
11	1	1	2
12	1	1	2
Total	12	12	24
Rata - rata	1	1	1

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\
 &= \frac{24}{24} \times 100\% \\
 &= 100\%
 \end{aligned}$$

2. Uji Fungsionalitas

Nomor Butir	Penguji 1	Penguji 2	Total	Nomor Butir	Penguji 1	Penguji 2	Total
1	1	1	2	22	1	1	2
2	1	1	2	23	1	1	2
3	1	1	2	24	1	1	2
4	1	1	2	25	1	1	2
5	1	1	2	26	1	1	2

6	1	1	2
7	1	1	2
8	1	1	2
9	1	1	2
10	1	1	2
11	1	1	2
12	1	1	2
13	1	1	2
14	1	1	2
15	1	1	2
16	1	1	2
17	1	1	2
18	1	1	2
19	1	1	2
20	1	1	2
21	1	1	2

27	1	1	2
28	1	1	2
29	1	1	2
30	1	1	2
31	1	1	2
32	1	1	2
33	1	1	2
34	1	1	2
35	1	1	2
36	1	1	2
37	1	1	2
38	1	1	2
39	1	1	2
40	1	1	2
41	1	1	2
42	1	1	2

	Total	42	42	84
	Rata - rata	1	1	1

Skor total = 84

$$\text{Persentase} = \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\%$$

$$= \frac{84}{84} \times 100\%$$

$$= 100\%$$

Lampiran 12. Perhitungan Hasil Pengujian oleh Ahli Media

1. Balance testing

Nomor Butir	Pengaji 1	Pengaji 2	Total
1	3	4	7
2	4	4	8
3	3	3	6
4	3	3	6
5	4	3	7
6	3	3	6
7	4	3	7
8	3	4	7
9	3	3	6
10	4	3	7
11	4	3	7
Total	38	36	74
Rata - rata	3,45	3,27	3,36

$$\begin{aligned} \text{Skor ideal} &= \sum \text{Butir} \times \sum \text{Responden} \times 4 \\ &= 88 \end{aligned}$$

$$\text{Skor total} = 74$$

$$\begin{aligned} \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{74}{88} \times 100\% \\ &= 84,09\% \end{aligned}$$

2. Compliance testing

Nomor Butir	Pengaji 1	Pengaji 2	Total
1	4	3	7
2	4	4	8
3	3	4	7
4	3	3	6
5	3	3	6

6	3	4	7
7	3	4	7
Total	23	25	48
Rata - rata	3,29	3,57	3,43

$$\begin{aligned} \text{Skor ideal} &= \sum \text{Butir} \times \sum \text{Responden} \times 4 \\ &= 56 \end{aligned}$$

$$\text{Skor total} = 48$$

$$\begin{aligned} \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{48}{56} \times 100\% \\ &= 85,71\% \end{aligned}$$

3. Playtesting

Nomor Butir	Penguji 1	Penguji 2	Total
1	3	3	6
2	3	3	6
3	4	3	7
4	4	3	7
5	4	3	7
6	4	4	8
7	3	3	6
8	3	4	7
Total	28	26	54
Rata - rata	3,50	3,25	3,38

$$\begin{aligned} \text{Skor ideal} &= \sum \text{Butir} \times \sum \text{Responden} \times 4 \\ &= 64 \end{aligned}$$

$$\text{Skor total} = 54$$

$$\begin{aligned} \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{54}{64} \times 100\% \\ &= 84,38\% \end{aligned}$$

4. Usability testing

Nomor Butir	Penguji 1	Penguji 2	Total
1	4	4	8
2	4	3	7
3	4	3	7
4	4	3	7
5	4	4	8
6	3	4	7
7	3	3	6
8	3	3	6
9	3	3	6
Total	32	30	62
Rata - rata	3,56	3,33	3,44

$$\begin{aligned} \text{Skor ideal} &= \sum \text{Butir} \times \sum \text{Responden} \times 4 \\ &= 72 \end{aligned}$$

$$\text{Skor total} = 62$$

$$\begin{aligned} \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{62}{72} \times 100\% \\ &= 86,11\% \end{aligned}$$

5. Kelayakan (Seluruh Aspek)

$$\text{Skor total (X)} = 238$$

$$\text{Butir pernyataan} = 35$$

$$\begin{aligned} \text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 35 \times 2 \times 4 = 280 \end{aligned}$$

$$\begin{aligned} \text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 35 \times 2 \times 1 = 70 \end{aligned}$$

$$\begin{aligned} \bar{X} &= (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal}) \\ &= (1/2) \times (350) \\ &= 175 \end{aligned}$$

$$\sigma = (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal})$$

$$= (1/6) \times (900)$$

$$= 35$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 35) \leq X < \bar{X} - (1,8 \times 35)$ 70 ≤ X < 112	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 35) \leq X < \bar{X} - (0,6 \times 35)$ 112 ≤ X < 154	Kurang Layak
3	$\bar{X} - (0,6 \times 35) \leq X < \bar{X} + (0,6 \times 35)$ 154 ≤ X < 196	Cukup Layak
4	$\bar{X} + (0,6 \times 35) \leq X < \bar{X} + (1,8 \times 35)$ 196 ≤ X < 238	Layak
5	$\bar{X} + (1,8 \times 35) \leq X \leq \bar{X} + (3 \times 35)$ 238 ≤ X ≤ 280	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Sangat Layak** ($238 \leq X < 280$).

$$\begin{aligned} \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{238}{280} \times 100\% \\ &= 85\% \end{aligned}$$

Lampiran 13. Perhitungan Hasil Pengujian oleh Ahli Materi

No Butir	Pengaji 1	Pengaji 2	Total	No Butir	Pengaji 1	Pengaji 2	Total
1	3	4	7	8	4	3	7
2	3	3	6	9	3	4	7
3	3	3	6	10	4	3	7
4	3	4	7	11	4	3	7
5	3	3	6	12	3	4	7
6	4	3	7	13	3	3	6
7	3	3	6	14	4	4	8

Total	47	47	94
Rata - rata	3,36	3,36	3,36

$$\text{Skor total (X)} = 94$$

$$\text{Butir pernyataan} = 14$$

$$\begin{aligned}\text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 14 \times 2 \times 4 = 112\end{aligned}$$

$$\begin{aligned}\text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 14 \times 2 \times 1 = 28\end{aligned}$$

$$\begin{aligned}\bar{X} &= (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal}) \\ &= (1/2) \times (140) \\ &= 70\end{aligned}$$

$$\begin{aligned}\sigma &= (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal}) \\ &= (1/6) \times (84) \\ &= 14\end{aligned}$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 14) \leq X < \bar{X} - (1,8 \times 14)$ $28 \leq X < 44,8$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 14) \leq X < \bar{X} - (0,6 \times 14)$ $44,8 \leq X < 61,6$	Kurang Layak
3	$\bar{X} - (0,6 \times 14) \leq X < \bar{X} + (0,6 \times 14)$	Cukup Layak

	61,6 ≤ X < 78,4	
4	$\bar{X} + (0,6 \times 14) \leq X < \bar{X} + (1,8 \times 14)$ 78,4 ≤ X < 95,2	Layak
5	$\bar{X} + (1,8 \times 14) \leq X \leq \bar{X} + (3 \times 14)$ 95,2 ≤ X ≤ 112	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($78,4 \leq X < 95,2$).

$$\begin{aligned}\text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{94}{112} \times 100\% \\ &= 83,93\%\end{aligned}$$

Lampiran 14. Perhitungan Hasil Pengujian oleh Pengguna

Siswa	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	Total
1	3	3	3	4	4	4	4	4	3	3	3	4	3	3	4	4	3	3	4	3	4	4	4	4	4	4	3	3	4	4	3	3	133							
2	3	3	4	4	4	3	3	4	4	3	3	4	3	3	4	4	3	3	4	3	3	4	3	3	4	3	3	4	3	3	3	3	130							
3	3	3	3	3	4	4	4	4	4	3	3	4	3	3	3	4	3	3	3	3	4	3	3	4	3	3	4	3	3	4	3	4	131							
4	4	3	3	3	4	4	3	3	4	4	4	3	3	4	4	3	3	4	4	4	4	4	4	4	4	4	3	3	4	4	4	4	4	137						
5	4	3	3	3	3	3	3	4	3	3	3	4	3	3	4	4	3	3	4	3	3	4	4	3	3	3	3	3	3	3	3	3	3	125						
6	3	3	4	4	2	3	3	3	3	4	3	3	4	3	3	3	4	3	3	3	2	3	3	2	3	4	3	3	2	4	2	4	121							
7	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	4	4	4	3	4	4	3	4	4	3	4	2	4	4	4	4	4	4	146						
8	3	3	3	4	3	4	4	4	4	4	3	3	4	4	3	4	4	4	3	4	2	3	4	3	4	3	3	4	3	4	3	4	3	136						
9	3	3	4	4	4	3	3	3	3	4	3	3	3	3	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	125						
10	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	122						
11	3	3	4	4	3	3	4	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	120						
12	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	4	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	124						
13	3	3	3	3	3	3	3	3	2	3	3	3	3	3	4	4	3	3	2	4	4	3	3	3	3	3	3	3	3	3	3	3	3	125						
14	3	4	3	3	3	3	3	3	4	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	119						
15	3	3	3	3	3	3	3	3	3	3	3	4	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	116						
16	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	116						
17	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	120						
18	4	3	3	3	4	3	4	4	3	3	3	3	3	3	3	4	3	3	3	4	3	3	3	4	3	3	3	4	3	3	3	3	3	130						
19	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	121						
20	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	154						
21	4	3	4	3	4	3	3	4	3	2	3	3	4	3	3	4	3	3	4	3	3	4	3	3	4	4	4	4	4	4	4	4	4	132						
22	4	3	4	3	4	3	3	4	3	3	3	4	3	3	4	3	3	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	131						
23	4	4	3	4	4	4	4	4	4	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	147						
24	3	4	3	4	4	4	4	4	4	3	3	4	4	4	4	4	4	4	3	4	3	3	4	3	3	4	3	3	4	3	3	4	3	132						
25	3	3	3	3	3	4	3	3	4	3	3	4	3	3	4	3	3	4	3	3	2	3	3	4	3	3	2	3	3	3	3	3	3	122						
26	3	4	3	4	4	3	3	3	3	3	3	4	3	3	4	3	3	4	3	3	3	3	3	4	3	3	4	3	3	4	3	3	3	126						
27	4	3	3	3	3	4	3	3	3	3	3	3	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	122						
28	3	4	3	3	3	3	3	3	3	3	3	4	3	3	4	3	3	4	3	2	2	3	3	3	2	2	3	3	3	3	3	3	3	116						
29	3	3	3	4	3	3	3	4	3	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	119						
30	4	4	4	4	4	4	4	4	4	3	3	4	4	3	3	4	4	3	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	142						

Total

3840

1. Balance testing

$$\text{Skor total (X)} = 978$$

$$\text{Rata - rata} = 3,260$$

$$\text{Butir pernyataan} = 10$$

$$\begin{aligned}\text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 10 \times 30 \times 4 = 1200\end{aligned}$$

$$\begin{aligned}\text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 10 \times 30 \times 1 = 300\end{aligned}$$

$$\bar{X} = (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal})$$

$$= (1/2) \times (1500)$$

$$= 750$$

$$\begin{aligned}\sigma &= (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal}) \\ &= (1/6) \times (900) \\ &= 150\end{aligned}$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 150) \leq X < \bar{X} - (1,8 \times 150)$ $300 \leq X < 480$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 150) \leq X < \bar{X} - (0,6 \times 150)$ $480 \leq X < 660$	Kurang Layak
3	$\bar{X} - (0,6 \times 150) \leq X < \bar{X} + (0,6 \times 150)$ $660 \leq X < 840$	Cukup Layak
4	$\bar{X} + (0,6 \times 150) \leq X < \bar{X} + (1,8 \times 150)$ $840 \leq X < 1020$	Layak
5	$\bar{X} + (1,8 \times 150) \leq X \leq \bar{X} + (3 \times 150)$ $1020 \leq X \leq 1200$	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($840 \leq X < 1020$).

$$\begin{aligned}\text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{978}{1200} \times 100\% \\ &= 81,50\%\end{aligned}$$

2. Playtesting

$$\text{Skor total (X)} = 1084$$

$$\text{Rata - rata} = 3,285$$

$$\text{Butir pernyataan} = 11$$

$$\begin{aligned}\text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 11 \times 30 \times 4 = 1320\end{aligned}$$

$$\begin{aligned}\text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 11 \times 30 \times 1 = 330\end{aligned}$$

$$\bar{X} = (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal})$$

$$= (1/2) \times (1650)$$

$$= 825$$

$$\sigma = (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal})$$

$$= (1/6) \times (990)$$

$$= 165$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 165) \leq X < \bar{X} - (1,8 \times 165)$ $330 \leq X < 528$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 165) \leq X < \bar{X} - (0,6 \times 165)$ $528 \leq X < 726$	Kurang Layak
3	$\bar{X} - (0,6 \times 165) \leq X < \bar{X} + (0,6 \times 165)$ $726 \leq X < 924$	Cukup Layak
4	$\bar{X} + (0,6 \times 165) \leq X < \bar{X} + (1,8 \times 165)$ $924 \leq X < 1122$	Layak
5	$\bar{X} + (1,8 \times 165) \leq X \leq \bar{X} + (3 \times 165)$ $1122 \leq X \leq 1320$	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($924 \leq X < 1122$).

$$\begin{aligned}\text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{1084}{1320} \times 100\% \\ &= 82,12\%\end{aligned}$$

3. Usability testing

$$\text{Skor total (X)} = 979$$

$$\text{Rata - rata} = 3,263$$

$$\text{Butir pernyataan} = 10$$

$$\begin{aligned}\text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 10 \times 30 \times 4 = 1200\end{aligned}$$

$$\begin{aligned}\text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 10 \times 30 \times 1 = 300\end{aligned}$$

$$\bar{X} = (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal})$$

$$= (1/2) \times (1500)$$

$$= 750$$

$$\sigma = (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal})$$

$$= (1/6) \times (900)$$

$$= 150$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 150) \leq X < \bar{X} - (1,8 \times 150)$ $300 \leq X < 480$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 150) \leq X < \bar{X} - (0,6 \times 150)$ $480 \leq X < 660$	Kurang Layak
3	$\bar{X} - (0,6 \times 150) \leq X < \bar{X} + (0,6 \times 150)$ $660 \leq X < 840$	Cukup Layak
4	$\bar{X} + (0,6 \times 150) \leq X < \bar{X} + (1,8 \times 150)$ $840 \leq X < 1020$	Layak
5	$\bar{X} + (1,8 \times 150) \leq X \leq \bar{X} + (3 \times 150)$ $1020 \leq X \leq 1200$	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($840 \leq X < 1020$).

$$\begin{aligned}\text{Percentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{979}{1200} \times 100\% \\ &= 81,58\%\end{aligned}$$

4. Materi

$$\begin{aligned}
 \text{Skor total (X)} &= 799 \\
 \text{Rata - rata} &= 3,329 \\
 \text{Butir pernyataan} &= 8 \\
 \text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\
 &= 8 \times 30 \times 4 = 960 \\
 \text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\
 &= 8 \times 30 \times 1 = 240 \\
 \bar{X} &= (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal}) \\
 &= (1/2) \times (1200) \\
 &= 750 \\
 \sigma &= (1/2) \times (1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal}) \\
 &= (1/6) \times (720) \\
 &= 150
 \end{aligned}$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 150) \leq X < \bar{X} - (1,8 \times 150)$ $240 \leq X < 384$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 150) \leq X < \bar{X} - (0,6 \times 150)$ $384 \leq X < 528$	Kurang Layak
3	$\bar{X} - (0,6 \times 150) \leq X < \bar{X} + (0,6 \times 150)$ $528 \leq X < 672$	Cukup Layak
4	$\bar{X} + (0,6 \times 150) \leq X < \bar{X} + (1,8 \times 150)$ $672 \leq X < 816$	Layak
5	$\bar{X} + (1,8 \times 150) \leq X \leq \bar{X} + (3 \times 150)$ $816 \leq X \leq 960$	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($672 \leq X < 816$).

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\
 &= \frac{799}{960} \times 100\% \\
 &= 82,23\%
 \end{aligned}$$

5. Kelayakan permainan (Seluruh Aspek)

$$\text{Skor total (X)} = 3840$$

$$\text{Rata - rata} = 3,282$$

$$\text{Butir pernyataan} = 39$$

$$\begin{aligned}\text{Skor tertinggi ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor tertinggi} \\ &= 39 \times 30 \times 4 = 4680\end{aligned}$$

$$\begin{aligned}\text{Skor terendah ideal} &= \text{Butir pernyataan} \times \text{Jumlah responden} \times \text{Skor terendah} \\ &= 39 \times 30 \times 1 = 1170\end{aligned}$$

$$\begin{aligned}\bar{X} &= (1/2) \times (\text{Skor tertinggi ideal} + \text{Skor terendah ideal}) \\ &= (1/2) \times (5850) \\ &= 2925\end{aligned}$$

$$\begin{aligned}\sigma &= (1/2)\times(1/3) \times (\text{Skor tertinggi ideal} - \text{Skor terendah Ideal}) \\ &= (1/6) \times (3510) \\ &= 585\end{aligned}$$

No	Rentang Skor	Kategori
1	$\bar{X} - (3 \times 150) \leq X < \bar{X} - (1,8 \times 150)$ $1170 \leq X < 1872$	Sangat Kurang Layak
2	$\bar{X} - (1,8 \times 150) \leq X < \bar{X} - (0,6 \times 150)$ $1872 \leq X < 2574$	Kurang Layak
3	$\bar{X} - (0,6 \times 150) \leq X < \bar{X} + (0,6 \times 150)$ $2574 \leq X < 3276$	Cukup Layak
4	$\bar{X} + (0,6 \times 150) \leq X < \bar{X} + (1,8 \times 150)$ $3276 \leq X < 3978$	Layak
5	$\bar{X} + (1,8 \times 150) \leq X \leq \bar{X} + (3 \times 150)$ $3978 \leq X \leq 4680$	Sangat Layak

Dari perhitungan menggunakan tabel pengategorian diatas, maka didapatkan hasil bahwa skor total (X) termasuk dalam kategori **Layak** ($3276 \leq X < 3978$).

$$\begin{aligned}\text{Persentase} &= \frac{\text{Skor total}}{\text{Skor tertinggi ideal}} \times 100\% \\ &= \frac{3840}{4680} \times 100\% \\ &= 82,05\%\end{aligned}$$

Lampiran 15. Dokumentasi Pengujian oleh Pengguna



