

ANALISIS DAN PERANCANGAN GAME EDUKASI “*NEED FOR SAFETY*”

SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS

UNTUK ANAK USIA 6-12 TAHUN

TUGAS AKHIR SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta

Untuk Memenuhi Sebagian Persyaratan

Guna memperoleh Gelar Sarjana Pendidikan



Disusun Oleh:

Anung Budianto

NIM. 10520241019

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2014

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

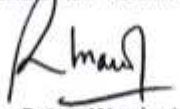
**ANALISIS DAN PERANCANGAN GAME EDUKASI "*NEED FOR SAFETY*"
SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS
UNTUK ANAK USIA 6-12 TAHUN**

Disusun Oleh:

Anung Budianto
NIM. 10520241019

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan
Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan.

Mengetahui,
Ketua Program Studi
Pendidikan Teknik Informatika,


Dr. Ratna Wardani

NIP. 19701218 200501 2 001

Yogyakarta, Juli 2014

Disetujui,
Dosen Pembimbing,


Dr. Eko Marpanaji

NIP. 19670608 199303 1 001

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Anung Budianto

NIM : 10520241019

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun

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 penulisan karya ilmiah yang telah lazim.

Yogyakarta, Juli 2014

Yang menyatakan,



Anung Budianto

NIM. 10520241019

HALAMAN PENGESAHAN

Tugas Akhir Skripsi

**ANALISIS DAN PERANCANGAN GAME EDUKASI "NEED FOR SAFETY"
SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS
UNTUK ANAK USIA 6-12 TAHUN**

Disusun oleh:

Anung Budianto
NIM 10520241019

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

Pada tanggal 28 Agustus 2014

TIM PENGUJI


Nama/Jabatan	Tanda Tangan	Tanggal
Dr. Eko Marpanaji Ketua Penguji/Pembimbing		12-9-2014
Satriyo Agung Dewanto, M.Pd Sekretaris		8-9-2014
Adi Dewanto, M.Kom Penguji		12/9/2014

Yogyakarta, September 2014

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,




Dr. Moch Bruri Triyono

NIP. 19560216 198603 1 0032

MOTTO DAN PERSEMBAHAN

A. MOTTO

"Keberhasilan sesungguhnya adalah mengatasi rasa takut akan ketidakberhasilan", Paul Sweeny.

"Hanya mereka yang berani gagal dapat meraih keberhasilan", Robert F.K.

"Sukses itu dapat terjadi karena persiapan, kerja keras, dan mau belajar dari kegagalan", Colin Powell.

"Dasar dari prestasi besar terletak pada kemauan untuk menjadi yang terbaik", Harold Taylor.

B. PERSEMBAHAN

Syukur kehadiran Allah SWT atas segala nikmat dan karunia-Nya, sehingga saya dapat menyelesaikan tugas akhir skripsi ini untuk kupersembahkan kepada:

Bapak dan Ibu tercinta yang tidak pernah lelah berdoa untuk putra putrinya.

Adik-adikku tersayang yang telah memberi dukungan dan semangat dalam keadaan apapun.

Teman-teman kelas E PTI 2010 yang telah bersama menuntut ilmu dan pastinya akan saya rindukan saat kebersamaan yang tidak akan dapat tergantikan.

ANALISIS DAN PERANCANGAN GAME EDUKASI “*NEED FOR SAFETY*” SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS UNTUK ANAK USIA 6-12 TAHUN

Oleh:
Anung Budianto
NIM. 10520241019

ABSTRAK

Penggunaan media atau sarana belajar yang menarik khususnya dalam pengenalan rambu-rambu lalu lintas adalah cara yang tepat untuk meningkatkan pengetahuan siswa. Aplikasi permainan “*Need For Safety*” merupakan media yang dijalankan di *smartphone* android yang dikembangkan sebagai sarana pendidikan siswa dalam mengenal dan mempelajari rambu-rambu-lalu lintas. Tujuan dari penelitian ini adalah untuk menghasilkan aplikasi permainan sebagai sarana belajar siswa untuk mengenal macam-macam rambu lalu lintas yang memenuhi aspek *functionality*, *efficiency*, *usability*, dan *portability* (ISO 9126).

Penelitian ini menggunakan metode *Research and Development* (R&D), dengan langkah-langkah: 1) Analisis Kebutuhan, 2) Desain dan Perancangan Sistem, 3) Pengembangan Perangkat Lunak, 4) Pengujian *White Box*, *Black Box*, *Portability*, dan *Alpha*, 5) Revisi Tahap I, 6) Pengujian *Beta*, 7) Revisi Tahap II, 8) Produk Akhir. Subjek penelitian yang digunakan adalah guru SDN Pujokusuman 1 dan anggota polisi bagian lalu lintas Polsek Bulaksumur. Pengumpulan data pada penelitian ini menggunakan bantuan lembar observasi dan kuesioner.

Hasil yang diperoleh berdasarkan prosedur pengujian kualitas perangkat lunak pada pengujian *alpha* adalah aspek *functionality* dengan nilai sebesar 90,00% (sangat layak), aspek *efficiency* sebesar 80,00% (layak), dan *usability* sebesar 86,25% (sangat layak). Hasil keseluruhan dari tiga aspek yang menjadi fokus pada uji *alpha* adalah 86,88% (sangat layak). Pengujian selanjutnya yaitu pengujian *beta* diperoleh hasil untuk aspek *functionality* sebesar 96,67% (sangat layak), aspek *efficiency* sebesar 93,60% (sangat layak), aspek *usability* sebesar 94,80% (sangat layak), dan aspek *portability* sebesar 94,00% (sangat layak). Hasil keseluruhan dari empat aspek kualitas perangkat lunak yang menjadi fokus pada uji *beta* adalah 95,20% (sangat layak). Hasil pengujian menunjukkan bahwa perangkat lunak “*Need For Safety*” telah memenuhi kaidah *software quality* sesuai standar ISO 9126 karena minimal memperoleh nilai dengan kategori “layak” pada setiap pengujian aspeknya.

Kata Kunci: media belajar, rambu-rambu lalu lintas, android, ISO 9126

KATA PENGANTAR

Puji syukur kahadirat Allah SWT atas berkat rahmat dan karunia-Nya, Tugas Akhir Skripsi dalam rangka memenuhi sebagian persyaratan untuk mendapatkan gelar Sarjana Pendidikan dengan Judul "Analisis dan Perancangan Game Edukasi "*Need For Safety*" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun" 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, Ibu dan seluruh keluarga yang telah memberikan berbagai dukungan dan doa yang sangat berarti.
2. Bapak Dr. Moch. Bruri Triyono, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi.
3. Bapak Dr. Eko Marpanaji selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini.
4. Bapak Muhammad Munir, M.Pd, Bapak Suparman, M.Pd, dan Ibu Dessy Irmawati, M.T selaku validator instrumen penelitian TAS yang memberikan saran/masukan perbaikan sehingga penelitian TAS dapat terlaksana sesuai dengan tujuan.

5. Bapak Muhammad Munir M.Pd dan Ibu Dr. Ratna Wardani selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan fasilitas selama proses penyusunan pra proposal sampai dengan selesainya TAS ini.
6. Tri Lestari yang telah memberikan dukungan dan semangat untuk menyelesaikan tugas akhir skripsi ini.
7. Teman-teman Informatika E 2010 yang telah bersama-sama menjalani studi selama ini.
8. Semua pihak yang tidak dapat disebutkan satu per satu.

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 2014

Penulis,

Anung Budianto

NIM. 10520241019

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
SURAT PERNYATAAN	iii
HALAMAN PENGESAHAN	iv
MOTTO DAN PERSEMBAHAN	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN.....	xviii
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah.....	5
C. Batasan Masalah	5
D. Rumusan Masalah	6
E. Tujuan Penelitian	6
F. Manfaat Penelitian.....	6
BAB II KAJIAN PUSTAKA	8
A. Tinjauan Pustaka	8
1. Pembelajaran Berbasis Komputer	8
2. <i>Game</i> Edukasi.....	11
3. Anak Usia Sekolah Dasar	17
4. Rambu Lalu Lintas	19
5. Android	23
6. Software Quality	28
7. ISO 9126.....	31
8. UML (Unified Modeling Language)	36
9. Construct 2	45

B. Penelitian yang Relevan	48
C. Kerangka Berfikir.....	50
BAB III METODE PENELITIAN.....	51
A. Model Pengembangan	51
B. Objek Penelitian	52
C. Tempat dan Waktu Penelitian.....	52
D. Sampel Penelitian.....	52
E. Variabel Penelitian.....	53
1. Variabel Penelitian	53
2. Definisi Operasional Variabel	53
F. Prosedur Pengembangan	54
1. Analisis Kebutuhan.....	54
2. Desain	56
3. Implementasi	74
4. Pengujian.....	74
G. Teknik Pengumpulan Data	78
1. Wawancara	79
2. Angket.....	80
3. Observasi	82
4. Tes.....	84
5. Survei	85
6. Teknik Dokumenter.....	86
7. Teknik Eksperimental	87
8. Nominal Group Technique (NGT).....	87
9. Delphi Technique	88
H. Skala Pengukuran	89
1. Skala Guttman.....	89
2. Skala Likert	89
I. Instrumen Penelitian	91
1. Lembar Observasi	91
2. Kuesioner (Angket)	92
J. Uji Instrumen Penelitian.....	100
K. Teknik Analisis Data	101

BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	103
A. Hasil Penelitian	103
1. Implementasi	103
2. Pengujian Perangkat Lunak.....	124
B. Pembahasan.....	153
1. <i>Alpha Testing</i>	153
2. <i>Beta Testing</i>	157
C. Hasil Akhir Produk.....	160
BAB V KESIMPULAN DAN SARAN.....	162
A. Kesimpulan.....	162
B. Saran	163
DAFTAR PUSTAKA	164
LAMPIRAN.....	167

DAFTAR TABEL

	Halaman
Tabel 1. Deskripsi Simbol Diagram <i>Use Case</i>	39
Tabel 2. Deskripsi Simbol Diagram <i>Use Case</i> (lanjutan).....	40
Tabel 3. Deskripsi Simbol Diagram <i>Activity</i>	42
Tabel 4. Definisi Aktor	58
Tabel 5. Definisi <i>Use Case</i> Sistem.....	58
Tabel 6. Skenario Memulai Aplikasi.....	59
Tabel 7. Skenario <i>On/Off</i> Musik <i>Game</i>	60
Tabel 8. Skenario Lihat Skor	60
Tabel 9. Skenario <i>Get Score Data</i>	60
Tabel 10. Skenario Mulai Bermain	61
Tabel 11. Skenario Mulai Bermain (lanjutan)	62
Tabel 12. Skenario Ayo Belajar Rambu	62
Tabel 13. Skenario Pengertian Rambu	63
Tabel 14. Skenario Rambu Peringatan	63
Tabel 15. Skenario Rambu Larangan	63
Tabel 16. Skenario Rambu Perintah.....	64
Tabel 17. Skenario Rambu Petunjuk	64
Tabel 18. Skenario Bantuan	64
Tabel 19. Skenario Keluar Aplikasi	65
Tabel 20. Jenjang dalam Skala Likert.....	90
Tabel 21. Lembar Observasi Uji Portabilitas Perangkat Lunak	92
Tabel 22. Spesifikasi Uji Membuka Aplikasi.....	93
Tabel 23. Spesifikasi Uji Menu <i>Game</i>	93
Tabel 24. Spesifikasi Uji Menu Mulai Bermain	94
Tabel 25. Spesifikasi Uji Menu Ayo Belajar	94
Tabel 26. Spesifikasi Uji Menu Ayo Belajar (lanjutan)	95
Tabel 27. Kisi-Kisi Uji Kelayakan Perangkat Lunak Aspek <i>Functionality</i> , <i>Efficiency</i> , dan <i>Usability</i>	96

Tabel 28. Kelayakan Perangkat Lunak Aspek <i>Functionality, Efficiency</i> , dan <i>Usability</i>	97
Tabel 29. Kisi-Kisi Instrumen Pengujian <i>Beta</i>	98
Tabel 30. Kisi-Kisi Instrumen Pengujian <i>Beta</i>	99
Tabel 31. Kuesioner Pengujian <i>Beta</i>	99
Tabel 32. Kuesioner Pengujian <i>Beta</i> (lanjutan)	100
Tabel 33. Interpretasi Persentase Likert.....	102
Tabel 34. Penyesuaian Interpretasi Persentase Likert.....	102
Tabel 35. Pengaturan Fungsi <i>Event</i> pada "Title Screer"	105
Tabel 36. Pengaturan Fungsi <i>Event</i> pada "Menu Utama"	106
Tabel 37. Pengaturan fungsi <i>Event</i> untuk Menampilkan kotak.....	108
Tabel 38. Nilai Variabel untuk Jumlah Kotak Tiap Level	109
Tabel 39. Fungsi <i>Event</i> untuk Menampilkan Gambar Kotak secara Acak.....	111
Tabel 40. Pengaturan Fungsi <i>Event</i> untuk Halaman Informasi Nilai	112
Tabel 41. Pengaturan Fungsi <i>Event</i> pada Menu Skor.....	113
Tabel 42. Pengaturan Fungsi <i>Event</i> pada Menu Ayo Belajar	114
Tabel 43. Pengaturan Fungsi <i>Event</i> Halaman Pengertian Rambu	115
Tabel 44. Pengaturan Fungsi <i>Event</i> Halaman Contoh Rambu	116
Tabel 45. Pengaturan Fungsi <i>Event</i> Halaman Bantuan.....	118
Tabel 46. Pengaturan Fungsi <i>Event On/ Off</i> Musik	119
Tabel 47. <i>Flow Graph</i> Fungsi Pilih_Kotak.....	126
Tabel 48. Uji <i>Test Case</i> Fungsi Pilih_Kotak	127
Tabel 49. Uji <i>Black Box</i> Memulai Aplikasi	128
Tabel 50. Uji <i>Black Box On/ Off</i> Musik.....	129
Tabel 51. Uji <i>Black Box</i> Lihat Skor	129
Tabel 52. Uji <i>Black Box Get Score Data</i>	130
Tabel 53. Uji <i>Black Box</i> Mulai Bermain	131
Tabel 54. Uji <i>Black Box</i> Mulai Bermain (lanjutan).....	132
Tabel 55. Uji <i>Black Box</i> Ayo Belajar Rambu	133
Tabel 56. Uji <i>Black Box</i> Pengertian Rambu	134
Tabel 57. Uji <i>Black Box</i> Rambu Peringatan	134
Tabel 58. Uji <i>Black Box</i> Rambu Larangan	135

Tabel 59. Uji <i>Black Box</i> Rambu Perintah	135
Tabel 60. Uji <i>Black Box</i> Rambu Petunjuk.....	136
Tabel 61. Uji <i>Black Box</i> Bantuan.....	136
Tabel 62. Uji <i>Black Box</i> Keluar Aplikasi.....	137
Tabel 63. Hasil Pengujian Aplikasi menggunakan <i>Android Virtual Device</i>	138
Tabel 64. Hasil Pengujian Aplikasi pada <i>Smartphone</i> Android.....	139
Tabel 65. Hasil Uji Variabel <i>Portability</i> Perangkat Lunak	145
Tabel 66. Spesifikasi Uji Membuka Aplikasi.....	146
Tabel 67. Spesifikasi Uji Menu <i>Game</i>	147
Tabel 68. Spesifikasi Uji Menu Mulai Bermain	147
Tabel 69. Spesifikasi Uji Menu Ayo Belajar	148
Tabel 70. Spesifikasi Uji Menu Ayo Belajar (lanjutan)	149
Tabel 71. Data Hasil Pengujian <i>Alpha</i> Aspek <i>Functionality</i> , <i>Efficiency</i> , dan <i>Usability</i>	150
Tabel 72. Saran Ahli Pengujian <i>Alpha</i>	150
Tabel 73. Data Hasil Pengujian <i>Beta</i>	152
Tabel 74. Perhitungan Data Hasil Uji <i>Alpha</i>	154
Tabel 75. Interpretasi Persentase Likert untuk Kualitas Perangkat Lunak ...	155
Tabel 76. Persentase Kelayakan Tiap Faktor Pada Uji <i>Alpha</i>	155
Tabel 77. Perhitungan Data Hasil Uji <i>Beta</i>	158
Tabel 78. Persentase Kelayakan Tiap Faktor Pada Uji <i>Beta</i>	159
Tabel 79. <i>Flow Graph</i> Fungsi Simpan_Skor.....	181
Tabel 80. Uji <i>Test Case</i> Fungsi Simpan_Kotak	182
Tabel 81. <i>Flow Graph</i> Alur Menu Ayo Belajar Aplikasi <i>Need For Safety</i>	183
Tabel 82. Perhitungan <i>Cyclomatic Complexity</i> dan <i>Independent Path</i> Alur Menu Ayo Belajar Aplikasi <i>Need For Safety</i>	183
Tabel 83. Uji Uji <i>Test Case</i> Alur Menu Ayo Belajar Aplikasi <i>Need For Safety</i>	184
Tabel 84. <i>Flow Graph</i> Alur Aplikasi <i>Need For Safety</i>	185
Tabel 85. Nilai <i>Cyclomatic Complexity</i> alur aplikasi <i>Need For Safety</i>	185
Tabel 86. Uji <i>Test Case</i> Alur Aplikasi <i>Need For Safety</i>	186

DAFTAR GAMBAR

	Halaman
Gambar 1. Contoh Rambu-Rambu Lalu Lintas	21
Gambar 2. Faktor-Faktor Kualitas Perangkat Lunak Menurut McCall.....	28
Gambar 3. Bagan <i>Software Quality</i> untuk Aplikasi <i>Game</i> dengan ISO 9126 (Gregor, 2008)	31
Gambar 4. Tampilan Halaman Awal Construct 2 R130.....	45
Gambar 5. Tampilan <i>Layout</i> Construct 2	46
Gambar 6. Tampilan Menu <i>Properties</i> dalam Construct 2.....	47
Gambar 7. Tampilan Menu <i>Projects</i> and <i>Layers</i> Construct 2.....	47
Gambar 8. Tampilan <i>Library</i> Construct 2	47
Gambar 9. Tampilan <i>Event Sheet</i> Construct 2	48
Gambar 10. Langkah-Langkah Pembuatan Aplikasi " <i>Need For Safety</i> "	51
Gambar 11. <i>Use Case Diagram</i> <i>Need For Safety</i>	59
Gambar 12. <i>Activity Diagram</i> <i>Need For Safety</i>	67
Gambar 13. Rancangan Halaman Judul Perangkat Lunak	68
Gambar 14. Rancangan Halaman Menu Utama.....	69
Gambar 15. Rancangan Halaman Permainan Level 1.....	69
Gambar 16. Rancangan Halaman Permainan Level 2.....	70
Gambar 17. Rancangan Halaman Permainan Level 3.....	70
Gambar 18. Rancangan Halaman Permainan Level 4.....	70
Gambar 19. Rancangan Halaman Informasi Nilai	71
Gambar 20. Rancangan Halaman Skor.....	71
Gambar 21. Rancangan Tampilan Submenu Ayo Belajar.....	72
Gambar 22. Rancangan Halaman Pengertian Rambu.....	72
Gambar 23. Rancangan Halaman Contoh Rambu.....	73
Gambar 24. Rancangan Halaman Definisi Contoh Rambu	73
Gambar 25. Rancangan Halaman Bantuan	74
Gambar 26. Objek Background Halaman <i>Title Screen</i>	104
Gambar 27. Objek Tombol "Mulai"	104
Gambar 28. Tampilan Halaman " <i>Title Screen</i> "	106

Gambar 29. Tampilan Menu Utama	107
Gambar 30. Permainan Level 1	109
Gambar 31. Permainan Level 2	109
Gambar 32. Permainan Level 3	110
Gambar 33. Permainan Level 4	110
Gambar 34. Tampilan Gambar Rambu secara Acak.....	111
Gambar 35. Informasi Nilai Permainan.....	112
Gambar 36. Menu Skor.....	114
Gambar 37. Submenu Ayo Belajar	115
Gambar 38. Halaman Pengertian Rambu.....	116
Gambar 39. Halaman Contoh Rambu.....	117
Gambar 40. Halaman Pengertian Contoh Rambu	117
Gambar 41. Halaman Bantuan	118
Gambar 42. Pilihan <i>Browser</i> untuk Menjalankan <i>Project</i>	120
Gambar 43. Run <i>Project</i> " <i>Need For Safety</i> "	120
Gambar 44. <i>Project</i> Dijalankan Melalui Google Chrome	121
Gambar 45. <i>Exporting Project</i> Menggunakan Phonegap.....	121
Gambar 46. Pilihan Tempat Menyimpan <i>File</i>	122
Gambar 47. Options Dukungan Mode Layar dan <i>Devices</i>	122
Gambar 48. Pilihan <i>Export HTML5 Project</i>	122
Gambar 49. Halaman <i>Home</i> Situs <i>Phonegap</i>	123
Gambar 50. Registrasi <i>User</i> Baru menggunakan <i>AdobeID</i>	123
Gambar 51. Halaman <i>Upload Project</i>	123
Gambar 52. Membuat <i>Installer</i> apk	124
Gambar 53. Pengujian <i>White Box</i> Fungsi Pilih_Kotak.....	125
Gambar 54. Hasil Pengujian Aplikasi pada AVD (OS 2.3.3 dengan Ukuran Layar 240x320)	140
Gambar 55. Hasil Pengujian Aplikasi pada AVD (OS 3.0 dengan Ukuran Layar 240x400)	140
Gambar 56. Hasil Pengujian Aplikasi pada AVD (OS 4.0.3 dengan Ukuran Layar 480x800)	141

Gambar 57. Hasil Pengujian Aplikasi pada AVD (OS 4.2.2 dengan Ukuran Layar 480x854)	141
Gambar 58. Instalasi Aplikasi pada <i>Smartphone</i> Samsung Galaxy Young S5360	142
Gambar 59. Instalasi Berhasil pada <i>Smartphone</i> Samsung Galaxy Young S5360	142
Gambar 60. Instalasi Aplikasi pada <i>Smartphone</i> IMO S89	143
Gambar 61. Instalasi Berhasil pada <i>Smartphone</i> IMO S89	143
Gambar 62. Instalasi Aplikasi pada <i>Smartphone</i> Cross a88	144
Gambar 63. Instalasi Aplikasi pada Tablet Advan T1H	144
Gambar 64. Diagram Persentase Faktor Kualitas Perangkat Lunak pada Uji <i>Alpha</i>	155
Gambar 65. Diagram Persentase Faktor Kualitas Perangkat Lunak pada Uji <i>Beta</i>	159
Gambar 66. <i>Event</i> Fungsi Simpan_Skor	181

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Izin Penelitian.....	168
Lampiran 2. Validasi Instrumen Penelitian.....	172
Lampiran 3. Pengujian <i>White Box</i> Aplikasi <i>Need For Safety</i>	181
Lampiran 4. Pengujian <i>Alpha</i> oleh Ahli	187
Lampiran 5. Kamus <i>Event Construct</i> 2	201

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Indonesia merupakan salah satu Negara dengan jumlah penduduk yang besar, sehingga menjadikan Negara ini sebagai Negara dengan penduduk terpadat keempat di dunia. Padatnya penduduk di Indonesia menyebabkan timbulnya tuntutan akan kebutuhan transportasi. Transportasi adalah sarana penunjang yang memiliki peran penting yang dibutuhkan dalam aktivitas ekonomi dan sosial. Kebutuhan akan sarana transportasi yang semakin tinggi juga mengakibatkan meningkatnya risiko terjadi kecelakaan lalu lintas.

Kasus kecelakaan lalu lintas di Indonesia masih relatif tinggi, dimana kasus kecelakaan tersebut terjadi setiap tahun dan terus meningkat. Badan Pusat Statistik (BPS) menyebutkan bahwa jumlah kecelakaan di Indonesia tahun 2011 adalah sebesar 108.696 kecelakaan yang mengakibatkan kerugian materi senilai 217.435 juta rupiah dan meningkat di tahun 2012 menjadi 117.949 kecelakaan dan kerugian materi sebesar 298.627 juta rupiah. Outlook 2013 Transportasi Indonesia menyebutkan empat faktor penyebab terjadinya kecelakaan lalu lintas, yaitu kondisi sarana dan prasarana transportasi, faktor manusia dan alam. Faktor manusia atau kelalaian manusia saat berlalu lintas adalah yang menjadi penyebab utama tingginya angka kecelakaan lalu lintas. Kecelakaan lalu lintas sering terjadi karena banyaknya pengguna kendaraan yang berperilaku tidak sesuai di jalan. Perilaku tersebut dapat tercermin dari kurangnya kesadaran masyarakat dalam mematuhi peraturan rambu-rambu lalu lintas, serta kurangnya

pengetahuan para pengendara yang masih awam atau pemula dalam mengerti rambu-rambu lalu lintas yang jumlahnya cukup banyak.

Mematuhi peraturan rambu-rambu lalu lintas merupakan salah satu bentuk *safety riding* dalam berkendara. Sosialisasi akan pentingnya rambu-rambu lalu lintas dari pemerintah selama ini masih kurang. Masyarakat berkendara hanya sebatas pengetahuan mereka masing-masing, akibatnya banyak terjadi kecelakaan lalu lintas yang disebabkan karena pelanggaran rambu-rambu lalu lintas.

Kesadaran akan pentingnya rambu-rambu lalu lintas perlu ditanamkan kepada masyarakat sejak dini yaitu kepada anak-anak. Dunia anak-anak merupakan dunia yang ceria dan menyenangkan untuk belajar hal-hal yang baru. Pembelajaran kepada anak tentang dasar keselamatan lalu lintas adalah untuk mempersiapkan anak-anak dalam membangun pengetahuan tentang lalu lintas dan sikap positif yang akan mendatangkan manfaat saat anak-anak tersebut menjadi dewasa nantinya. Perkembangan teknologi sekarang ini membuat anak-anak tidak kesulitan dalam mendapatkan informasi-informasi dalam proses pembelajaran mereka. Informasi-informasi tersebut didapat karena perkembangan alat bantu belajar anak yang biasa disebut dengan media. Media alat bantu belajar anak tidak semua dapat digunakan secara tepat sebagai sarana pendidikan bagi anak-anak. Media yang sesuai untuk pembelajaran anak-anak adalah media yang menarik yang dikenal baik dan disukai oleh anak-anak.

Salah satu media yang dapat digunakan dalam pendidikan anak terutama dalam hal pengenalan rambu-rambu lalu lintas adalah melalui aplikasi permainan atau *game*. *Game* merupakan kegiatan terstruktur atau semi terstruktur yang

biasanya bertujuan untuk hiburan dan juga dapat digunakan sebagai sarana pendidikan. Karakteristik game yang menyenangkan dan dapat memberikan motivasi membuat *game* digemari banyak orang terutama di kalangan anak-anak. Ariyadi Wijaya (2009:2) menyebutkan bahwa permainan merupakan situasi permasalahan yang nyata bagi siswa sekolah dasar. *Game* dapat digunakan untuk penghilang kejenuhan dalam pemberian materi pendidikan, menciptakan lingkungan belajar yang menyenangkan dan memberikan motivasi kepada anak-anak untuk belajar.

Pemanfaatan *game* sebagai sarana pendidikan atau penyampaian informasi dapat diterapkan pada teknologi yang banyak digunakan saat ini, seperti *mobile phone*. *Mobile phone* merupakan alat komunikasi yang dapat digunakan untuk telepon atau mengirim pesan teks. Perkembangan *mobile phone* saat ini sudah memiliki banyak fitur lengkap yang salah satunya adalah untuk memainkan *game* yang biasa disebut dengan *mobile game*. *Mobile game* merupakan aplikasi *game* yang terdapat dalam *mobile phone*. Aplikasi *game* ini banyak sekali jenisnya tergantung dari kebutuhan pengguna. Salah satu jenis dari *mobile game* adalah jenis *game* edukasi yang di dalamnya berisi pengetahuan-pengetahuan yang dapat memperluas wawasan. *Game* edukasi selain menyenangkan untuk dimainkan, juga memiliki keunggulan lain yaitu dapat memberikan pengetahuan untuk memperluas wawasan terutama untuk anak-anak yang suka bermain *game*. *Mobile game* sebagai sarana pendidikan dapat dimanfaatkan dan dikembangkan seiring dengan pesatnya perkembangan *mobile phone*.

Perkembangan *mobile phone* dari waktu ke waktu mengalami kemajuan pesat yang ditandai dengan munculnya berbagai perangkat *mobile phone* yang

sudah mendukung komputasi dengan berbagai fitur yang disediakan. Salah satu dari perkembangan *mobile phone* tersebut adalah *smartphone*. *Smartphone* merupakan sebuah *device* yang memungkinkan untuk melakukan komunikasi dimana *smartphone* ini sudah memiliki fungsi layaknya komputer. *Smartphone* juga memiliki sistem operasi tersendiri di dalamnya. Salah satu sistem operasi yang digunakan pada *smartphone* adalah sistem operasi android. Kelebihan-kelebihan yang dimiliki *smartphone* android menyebabkan pengembangan aplikasi *smartphone* ini berkembang pesat. *Mobile phone* android dapat digunakan untuk berbagai aktivitas karena memuat banyak fitur dan aplikasi sehingga dapat digunakan untuk memainkan *mobile game* yang salah satu contohnya adalah jenis *game* edukasi.

Semakin banyaknya *game* edukasi berbasis android di kalangan masyarakat membuat peneliti berkeinginan untuk mengembangkan aplikasi *game* yang bertujuan untuk membantu proses pendidikan anak mengenal rambu-rambu lalu lintas. Pengembangan aplikasi ini nantinya akan dilakukan tahap pengujian *software quality* terlebih dahulu sebelum aplikasi dapat diluncurkan dan digunakan oleh pengguna akhir.

Penelitian ini bertujuan untuk menganalisis kualitas akhir dari aplikasi "*Need For Safety*" sebagai media pengenalan rambu-rambu lalu lintas kepada anak-anak. Aplikasi ini dikembangkan berdasarkan aspek *software quality* agar dapat diketahui tingkat kelayakanan sehingga dapat digunakan oleh pengguna nantinya. Hasil penelitian ini nantinya diharapkan dapat menjadi acuan dalam pengembangan aplikasi-aplikasi android lainnya yang ditujukan untuk sarana pendidikan.

B. Identifikasi Masalah

Berdasarkan uraian latar belakang di atas, maka ada beberapa permasalahan yang dapat diidentifikasi, yaitu :

1. Pertumbuhan sarana transportasi sebagai dampak dari meningkatnya jumlah penduduk menimbulkan permasalahan baru yaitu permasalahan lalu lintas.
2. Mayoritas pemicu kecelakaan lalu lintas adalah dari faktor manusia.
3. Kurangnya kesadaran dan pengetahuan masyarakat akan pentingnya mematuhi peraturan rambu-rambu lalu lintas.
4. Sosialisasi akan pentingnya mematuhi peraturan rambu-rambu lalu lintas bagi masyarakat masih dirasa kurang.
5. Dibutuhkan media atau sarana yang tepat yang dapat digunakan sebagai sarana pengenalan pentingnya rambu-rambu lalu lintas kepada masyarakat, terutama untuk anak-anak.
6. Perancangan *game* edukasi "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas kepada anak usia 6-12 tahun.

C. Batasan Masalah

Penelitian ini meliputi pengembangan perangkat lunak *mobile* berbasis android. Permasalahan yang diteliti akan dibatasi sebagai berikut :

1. Penelitian ini tidak sampai meneliti pengaruh keefektifan game "*Need For Safety*" sebagai media pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun.
2. Penelitian ini dibatasi pada pengujian *software quality* perangkat lunak "*Need For Safety*" menurut aspek *functionality*, aspek *efficiency*, aspek *usability*, dan aspek *portability*.

3. Aspek *software quality* yang digunakan mengacu pada faktor kualitas perangkat lunak menurut ISO 9126.

D. Rumusan Masalah

1. Bagaimanakah perancangan perangkat lunak "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12.
2. Bagaimanakah unjuk kerja perangkat lunak "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 berdasarkan aspek *functionality*, *efficiency*, *usability*, dan *portability* (ISO 9126).

E. Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini diharapkan dapat menghasilkan aplikasi yang memberikan manfaat berupa informasi mengenai definisi dan macam-macam rambu lalu lintas untuk anak usia 6-12 tahun yang memenuhi aspek *functionality*, *efficiency*, *usability*, dan *portability*.

F. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat antara lain:

1. Secara teoritis

Manfaat secara teoritis yang diharapkan dari penelitian ini adalah:

- a. Perancangan *game* edukasi "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun.
- b. Hasil uji kualitas perangkat lunak "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun yang

mengacu pada faktor kualitas perangkat lunak menurut ISO 9126 yaitu meliputi aspek *functionality*, *efficiency*, *usability*, dan *portability*.

2. Secara praktis

Manfaat secara praktis dari penelitian ini adalah hasil penelitian dapat menjadi bahan kajian perbandingan dan referensi dalam pengembangan dan implementasi *game* edukasi sehingga dikemudian hari dapat semakin berkembang, inovatif, dan beragam.

BAB II

KAJIAN PUSTAKA

A. Tinjauan Pustaka

1. Pembelajaran Berbasis Komputer

Pembelajaran berbasis komputer merupakan program pembelajaran dengan menggunakan *software* komputer berupa program komputer yang berisi tentang muatan pembelajaran (Rusman, 2012:97). Pembelajaran berbasis komputer digunakan untuk menyampaikan pembelajaran secara individual dan langsung kepada para siswa dengan cara berinteraksi langsung dengan perangkat lunak komputer yang berisi materi yang diajarkan. Pembelajaran berbasis komputer memiliki beberapa model yang digunakan dalam pembelajaran. Model-model pembelajaran berbasis komputer antara lain:

a. Model *Drills*

Model *drills* adalah suatu model dalam pembelajaran dengan jalan melatih siswa terhadap bahan pelajaran yang sudah diberikan (Rusman, 2012:112). Model pembelajaran *drills* dilakukan dengan memberikan pelatihan kepada siswa secara terus-menerus sehingga materi ajar akan tertanam dan menjadi kebiasaan. Penggunaan model *drills* dalam pembelajaran berbasis komputer memiliki tujuan untuk memberikan pengalaman belajar yang konkret melalui penciptaan tiruan bentuk pengalaman yang mendekati suasana sebenarnya.

b. Model Tutorial

Model tutorial adalah bimbingan pembelajaran dalam bentuk pemberian arahan, bantuan, petunjuk, dan motivasi agar para siswa belajar secara efisien

dan efektif (Rusman, 2012:116). Model tutorial digunakan dengan tujuan untuk memberikan bantuan kepada para siswa agar dapat mencapai hasil belajar secara optimal. Program tutorial yang diajarkan melalui program pembelajaran komputer merupakan pembelajaran dengan menggunakan *software* atau perangkat lunak komputer yang berisi materi pelajaran dan soal untuk latihan.

c. Model Simulasi

Model simulasi adalah salah satu strategi pembelajaran yang bertujuan untuk memberikan pengalaman belajar yang lebih konkret melalui penciptaan tiruan yang mendekati suasana yang sebenarnya dan berlangsung dalam suasana yang tanpa resiko (Rusman, 2012:120). Pembelajaran menggunakan komputer dengan model simulasi dikemas dengan menggunakan animasi-animasi yang menarik yang sesuai dengan suasana sebenarnya. Simulasi digunakan untuk meminimalkan resiko, misalkan resiko jatuh atau menabrak saat belajar mengendarai kendaraan.

d. Model *Instructional Games*

Model *instrctional games* merupakan salah satu bentuk metode pembelajaran berbasis komputer yang bertujuan menyediakan pengalaman belajar dengan memberikan fasilitas belajar untuk menambah kemampuan siswa melalui bentuk permainan yang mendidik (Rusman, 2012:122). *Instructional games* sebagai model pembelajaran berbasis komputer adalah permainan dalam bentuk *software* atau perangkat lunak yang didalamnya terdapat pembelajaran untuk siswa. Model pembelajaran ini dirancang sedemikian rupa, sehingga pembelajaran lebih menarik dan menantang. *Instructional games* menurut Azhar Arsyad (2006:162) merupakan perogram permainan yang dirancang dengan baik

dapat memotivasi siswa dan meningkatkan pengetahuan dan keterampilan siswa. Komponen yang digunakan dalam model *instructional games* ini merupakan alat pembangkit motivasi belajar bagi siswa.

Karakteristik *instructional games* sebagai model pembelajaran yaitu memiliki tujuan, aturan, kompetisi, dan tantangan. Tujuan dalam *instructional games* adalah tujuan pembelajaran yang telah ditetapkan. Contoh tujuan dalam *instructional games* adalah pencapaian skor tertinggi. Aturan adalah penetapan tindakan yang dapat dilakukan dan yang tidak dapat dilakukan pemain. Kompetisi adalah bagian dalam *instructional games* yang biasanya membuat pemain menjadi lebih termotivasi dalam memainkannya. Kompetisi dalam *instructional games* contohnya adalah menyerang lawan, melawan diri sendiri, melawan sisa waktu yang disediakan, atau kompetisi untuk mendapatkan skor tertinggi. karakteristik selanjutnya dalam *instructional games* adalah tantangan yang akan membuat pemain lebih termotivasi dalam memainkan permainan.

Kegiatan belajar melalui permainan semakin populer digunakan karena guru mulai menyadari potensi yang dimilikinya untuk memotivasi siswa dalam belajar (Rusman, 2012:123). Motivasi belajar harus diberikan kepada siswa agar siswa memiliki dorongan atau kemauan yang kuat untuk belajar. Peneliti memilih mengembangkan model pembelajaran dalam bentuk *instructional games* karena model pembelajaran yang dirancang dengan menarik membuat siswa lebih termotivasi dalam belajar. Peneliti mengembangkan perangkat lunak permainan atau *instructional games* untuk digunakan sebagai sarana belajar yang akan memotivasi siswa belajar materi pelajaran khususnya adalah untuk mengenal rambu-rambu lalu lintas.

2. *Game* Edukasi

a. Pengertian *Game*

Permainan menurut Andang Ismail (2007:17) adalah bagian mutlak dari kehidupan anak maupun dewasa dan permainan merupakan bagian integral dari proses pembentukan kepribadian anak. Usia dini adalah usia dimana anak-anak tidak akan terlepas dari permainan. Permainan dulu dikenal dengan nama permainan tradisional dan dengan seiring perkembangannya *game* yang menarik harus menggunakan *console* atau mesin pemutar *game*. Permainan atau *game* memiliki banyak jenis mulai dari awal perkembangannya hingga saat ini yang sudah banyak memakai teknologi 3D.

1) Sejarah Singkat Perkembangan *Game*

Game dari awal kemunculannya hingga perkembangannya sekarang ini memiliki jenis dan teknologi yang beragam. *Game* modern yang membutuhkan mesin pemutar *game* atau *console* sudah banyak dikembangkan sampai sekarang ini. Dunia *game* diawali dengan *console-console* pendahulu seperti Atari, Nintendo, Super Nintendo (SNES), dan SEGA yang memiliki tampilan 2D yang sederhana, namun pada jaman itu memang banyak diminati masyarakat.

Perkembangan selanjutnya adalah muncul *console game* seperti Sony Playstation, Nintendo 64, dan XBOX yang masing-masing menyediakan *game-game* yang lebih menarik yaitu dengan menampilkan tampilan grafis dan efek yang begitu memukau. Pada tahun 2007, Anggra (2008:1) menyebutkan bahwa kalangan masyarakat, baik masyarakat awam maupun *gamer*, dihadapkan dengan evolusi baru untuk *console* yaitu Sony Playstation 2 dan 3, Nintendo Wii, serta XBOX 360. Kemunculan produk *console game* tersebut membuat

masyarakat banyak menggunakan *game* untuk mendapatkan kepuasan dan kesenangan saat bermain.

Perkembangan selanjutnya adalah munculnya *game* yang dapat dimainkan dengan menggunakan PC atau *Personal Computer*. Seiring berkembangnya PC dan banyaknya masyarakat yang menggunakan PC dalam kehidupannya sehari-hari menjadikan *game* banyak dikembangkan agar dapat dimainkan secara menarik. Salah satu yang membuat *game* menarik dalam perkembangannya saat ini adalah munculnya teknologi 3D yang membuat para *gamer* seperti masuk ke dalam dunia permainan yang dimainkannya secara langsung. Perkembangan *game* jenis 3D sekarang ini sudah mencapai *console handphone* atau sekarang yang sering dinamakan dengan *smartphone*.

Game atau permainan memiliki beberapa jenis, antara lain adalah sebagai berikut (Anggra, 2008:2):

a) Arcade/Side Scrolling

Game dengan jenis ini sering disebut sebagai *game* klasik. *Game* ini memiliki ciri tampilan 2D dan cara menggerakkan karakter-karakter dalam permainan tersebut adalah ke atas, bawah, kiri, dan kanan. Ciri lainnya adalah pergerakan layar background yang selalu berganti dari satu wilayah ke wilayah yang lainnya. *Game* jenis ini contohnya adalah Sonic (SEGA), Mario Bros (Nintendo), dan Metal Slug (Playstation).

b) Racing

Game yang berkembang mulai dari tahun 2005 ini menjadi salah satu jenis *game* yang berkembang pesat sejak kemunculannya. *Game* ini banyak menarik perhatian karena objek-objek dalam *game* ini adalah sesuai dengan

perkembangan dunia otomotif. Objek-objek kendaraan yang digunakan adalah jenis-jenis kendaraan yang ada di pasar otomotif dunia. Cara bermainnya mudah yaitu pemain hanya harus memenangkan balapan dengan cara masuk garis finish pada posisi pertama. *Game* dengan jenis ini contohnya adalah Grand Turismo (Playstation), Need For Speed Series (Playstation dan PC), GrandPrix (Playstation dan PC).

c) Fighting

Game dengan jenis *fighting* merupakan *game* yang memiliki *gameplay* yang mudah, yaitu pemain harus menang ketiga bertarung dengan lawannya. Awal perkembangannya, *game* jenis ini memiliki tampilan 2D, dan sekarang yang sudah menggunakan tampilan grafis 3D. *Game* jenis ini biasanya memiliki tingkatan level yang membuat lawan bertarung untuk pemain memiliki tingkat kesulitan dari mudah hingga sangat sulit untuk dikalahkan. Contoh dari *game* jenis ini adalah Street Fighter series (SNES dan Playstation), Mortal Kombat (Playstation, PC), dan Tekken (Playstation).

d) Shooting

Game shooting merupakan *game* yang cukup sederhana yang cara memainkannya adalah dengan menembak semua musuh-musuh yang menghalangi selama bermain. *Game* ini memiliki perkembangan berupa inovasi yang dimulai dengan performa grafis yang ditingkatkan sehingga menampilkan tampilan yang menarik, pemberian *sound effect* yang memukau yang membuat pemain seperti seolah-olah masuk ke lingkungan *game* yang dimainkannya, dan penyisipan alur cerita *game* yang membuat pemain memiliki rasa penasaran ketika memulai memainkan permainan ini.

Shooting game dapat dibedakan menjadi dua jenis yaitu *first person shooter* (fps) dan *third person shooter* (tps). Perbedaan diantara kedua jenis *game* tersebut adalah terletak pada sudut pandang pemain di dalam permainan tersebut. Fps merupakan jenis *game* shooting dimana pemain ditempatkan pada sudut pandang sebagai orang pertama dan sebagai karakter utama. *Game* jenis ini menampilkan hanya senjata yang digunakan karakter utama atau juga untuk beberapa jenis *game* lain hanya menampilkan penanda sasaran (*crosshair*) saja. Contoh dari *game* jenis ini adalah DOOM, Far Cry, Stalker, dan lain-lain.

Game shooting yang kedua adalah tps atau *third person shooter*. *Game* jenis ini, sudut pandang pemainnya ditempatkan sebagai orang ketiga yang mengendalikan karakter utama yang dimainkan di dalam *game*. Ciri dari *game* tps ini adalah tampilan karakter utama yang sepenuhnya ditampilkan. Contohnya adalah Tomb Rider dan Splinter Cell series.

e) *RTS (Real Time Strategy)*

Pemain di dalam memainkan *game* jenis ini memiliki kontrol terhadap satu atau lebih tokoh *game*. *Game* strategi adalah *game* yang harus dipikirkan cara untuk memenangkannya. *Game* jenis ini banyak digemari oleh *gamer* karena jalan ceritanya yang mengharuskan pemain untuk berfikir keras agar dapat menang. *Game* berjenis strategi contohnya adalah Command and Conquer series, WarCraft, dan Age Of Empire series.

f) *RPG (Role Playing Game)*

Game jenis ini menyediakan cerita yang penuh dengan intrik, pengembangan watak karakter yang mendalam, dan alur cerita yang

panjang yang membuat *game* ini membutuhkan waktu yang lama untuk memenangkannya. *Game* jenis ini awalnya dipelopori oleh negara Jepang dengan rilis *game* pertamanya yaitu *game* Final Fantasy yang sekarang menjadi inspirasi global para pengembang *game* di seluruh dunia untuk membuat inovasi *game* sejenis. *Game* dengan tipe ini antara lain Final Fantasy series, Legend of Dragon, Rogue Galaxy (Playstation2).

g) Simulation

Simulation adalah *game* yang penggambaran konsep permainannya merupakan segala sesuatu hal di dalam kehidupan nyata, sehingga hal realistik akan lebih sering ditemui dalam *game* ini. *Game* dengan jenis ini dapat menggambarkan tentang kehidupan pribadi kita sehari-hari, kehidupan dalam pekerjaan, pengaturan suatu hal, dan pengoperasian alat-alat tertentu. *Game* jenis ini digemari karena pemain bisa belajar menggunakan sesuatu tanpa memakai alat aslinya. Contohnya saja adalah simulasi menyetir kendaraan. Pemain bisa belajar cara mengendarai kendaraan dengan cara bermain *game* yang akan dirasa menyenangkan dan tidak berbahaya. Contoh dari *game* jenis ini adalah Sim City yang merupakan salah satu jenis *game* untuk membangun sebuah kota.

Kesimpulan dari uraian di atas adalah bahwa *game* merupakan suatu alat atau media yang dapat digunakan seseorang sebagai sarana belajar yang menyenangkan.

b. Pengertian Edukasi

Edukasi atau *education* menurut kamus besar bahasa Inggris memiliki arti pendidikan. Pendidikan menurut kamus besar Bahasa Indonesia (1991)

merupakan proses pengubahan sikap dan tata laku seseorang atau sekelompok orang dalam usaha mendewasakan manusia melalui upaya pengajaran dan pelatihan.

Pendidikan merupakan proses pembelajaran yang didapat oleh semua manusia. Tujuan dari pendidikan adalah untuk membuat seseorang mengerti tentang apa yang diajarkan dan mampu berfikir lebih kritis. Pendidikan harus didapatkan oleh setiap orang mulai dari orang tersebut dilahirkan sampai dewasa. Pendidikan adalah proses yang berlangsung terus menerus. Pendidikan yang didapatkan manusia dapat digolongkan menjadi pendidikan formal dan non formal. Pendidikan formal diperoleh dari suatu pembelajaran yang terstruktur yang telah dirancang oleh suatu institusi. Sedangkan pendidikan non formal adalah pendidikan yang didapat manusia dalam kehidupan sehari-hari yang dialami dan dipelajari dari lingkungan atau orang lain.

Kesimpulan dari uraian diatas adalah bahwa edukasi merupakan proses belajar yang terus menerus tanpa henti yang dilakukan manusia untuk mencapai tujuan mengubah individu menjadi lebih baik dalam segala aspek kehidupannya.

c. Pengertian *Game* Edukasi

Pengertian sederhana dari *game* edukasi adalah *game* yang dirancang dengan tujuan untuk pengayaan pendidikan. Permainan jenis ini dikembangkan untuk mendukung proses pengajaran dan pembelajaran. Menurut Handriyanti (2009), *game* edukasi adalah permainan yang dirancang atau dibuat untuk merangsang daya pikir termasuk meningkatkan konsentrasi dan memecahkan masalah.

Hurd dan Jenuings (2009) menyebutkan *educational game* adalah *game* yang khusus dirancang untuk mengajarkan *user* suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya. *Game* edukasi dapat digunakan dalam memberikan pengajaran, menambah pengetahuan penggunanya dengan cara yang menyenangkan dan menarik. *Game* edukasi dirancang dan biasanya ditujukan untuk anak-anak, karena anak-anak tidak akan pernah lepas dengan yang namanya permainan atau *game*.

Kesimpulan dari uraian di atas adalah bahwa *game* edukasi merupakan salah satu bentuk permainan yang memiliki manfaat untuk menunjang proses belajar-mengajar dengan metode yang menyenangkan dan menarik, dan digunakan untuk memberikan pengajaran atau menambah pengetahuan kepada penggunanya.

3. Anak Usia Sekolah Dasar

Anak usia sekolah dasar yaitu usia enam sampai dua belas tahun berada pada tahap operasional konkret, karena pada tahap ini pikiran anak terbatas pada objek-objek yang dijumpai dari pengalaman-pengalaman langsung (Ishak Abdulhak dan Deni Darmawan, 2013:73). Pemikiran operasi konkret merupakan tonggak kognitif yang memungkinkan anak pada awal sekolah dasar berfikir dan bertindak sebagaimana mestinya. Fase pemikiran konkret ini, anak membutuhkan suatu alat peraga dalam bentuk fisik untuk membantunya dalam berfikir. Cara berfikir anak terhadap sesuatu yang sifatnya abstrak dibantu dengan menggunakan alat peraga atau media, seperti contohnya untuk

mempelajari perhitungan matematika, digunakan media seperti lidi, kancing, manik-manik, dan lainnya. Penggunaan alat atau media tersebut adalah untuk memperjelas dan mempercepat pemahaman materi yang sifatnya abstrak.

Perkembangan kognitif anak menurut George (2008:291) menyatakan bahwa anak dalam tahap operasi konkret, sejak usia tujuh sampai dua belas tahun mulai menggunakan citra dan simbol mental selama proses pemikiran dan dapat membalik operasi. Hal tersebut membuktikan bahwa anak pada usia sekolah dasar mampu berfikir operasional apabila dalam pengerjaannya disertai dengan proses mengutak-atik benda-benda konkret.

Penggunaan media atau alat untuk membantu anak berfikir juga tidak terlepas dari peran teknologi. Selain media dalam bentuk konkret, perkembangan teknologi juga sekarang ini telah menyediakan berbagai macam bentuk media untuk membantu anak belajar. Penggunaan teknologi menurut George (2008:24) perlu digunakan karena membantu pengajar memberdayakan anak dengan kemampuan teknologi yang tepat yang akan meningkatkan, memajukan, dan memperkaya pembelajaran. Perkembangan teknologi perlu diajarkan dan juga diawasi oleh pengajar yang menggunakan teknologi kepada anak didiknya. Peran teknologi dalam membantu pembelajaran anak adalah menggunakan komputer, laptop, maupun media *handphone* yang dapat dimanfaatkan.

Media yang digunakan dalam membantu perkembangan anak usia sekolah dasar juga dimanfaatkan untuk menambah motivasi anak dalam belajar. Motivasi pada anak dapat muncul dengan berbagai cara seperti rasa puas pada anak ketika menggunakan media sebagai alat belajar atau pembuatan suasana belajar yang menyenangkan bagi anak. Motivasi belajar harus ditumbuhkan kepada anak

karena masa anak-anak adalah masa yang penting dan masa yang tepat untuk belajar sebanyak mungkin pengetahuan dan wawasan. Trianto (2011:6) menyebutkan bahwa anak usia dini merupakan periode awal yang paling penting dan mendasar di sepanjang rentang pertumbuhannya dan perkembangan kehidupan manusia. Pada masa ini ditandai dengan berbagai periode penting yang fundamen dalam kehidupan anak selanjutnya sampai periode akhir perkembangannya selanjutnya. Oleh karena itu masa anak-anak terutama anak sekolah dasar usia tujuh hingga dua belas tahun perlu diajarkan berbagai macam pengetahuan dan wawasan yang akan membantunya belajar sampai periode akhir perkembangan selanjutnya.

4. Rambu Lalu Lintas

Zulfiar (2010:32) menyebutkan dalam bukunya yang berjudul Transportasi (suatu pengantar), menyatakan bahwa lalu lintas merupakan gerak kendaraan baik bermotor maupun tidak dengan motor (sepeda, delman, dan lainnya), pejalan kaki, dan hewan di jalan yang berkaitan dengan operasi atau penggunaan jalan. Lalu lintas bukan hanya terdiri dari banyaknya kendaraan ataupun orang yang ada di jalan raya, tetapi juga hewan. Kendaraan, pejalan kaki, dan juga hewan yang ada di jalan raya membutuhkan suatu alat untuk mengatur pergerakannya. Aturan tersebut dinamakan peraturan lalu lintas. Tata cara berlalu lintas di jalan diatur dengan peraturan perundang-undangan yang menyangkut arah lalu lintas, prioritas menggunakan jalan, lajur lalu lintas, pengendalian arus di persimpangan, tata cara melewati suatu perlintasan, tata

cara parkir di jalan serta lainnya yang berkaitan dengan operasi di jalan (Zulfiar, 2010:32).

Penyusunan peraturan lalu lintas untuk mengatur ketertiban pengguna lalu lintas dinamakan dengan rekayasa lalu lintas. Zulfiar (2010:38), mendefinisikan rekayasa lalu lintas sebagai rekayasa untuk mengalirkan lalu lintas orang atau barang secara aman dan efisien dengan merencanakan, membangun dan mengoperasikan geometrik jalan, dan dilengkapi dengan rambu lalu lintas, marka jalan serta alat pemberi isyarat lalu lintas. Penyusunan peraturan lalu lintas termasuk di dalam rekayasa lalu lintas. Leksmono (2008:2), mendefinikan rekayasa lalu lintas adalah bidang kajian yang mempelajari metode perancangan ruang lalu lintas jalan yang aman dan nyaman bagi pengguna jalan dan efisien dari sudut pandang pembiayaan atau penggunaan lahan. Salah satu rekayasa lalu lintas adalah perencanaan peraturan lalu lintas.

Salah satu struktur peraturan lalu lintas yang mengatur pengguna jalan adalah peraturan jalan dan lalu lintas. Leksmono (2008:117) dalam bukunya yang berjudul "Rekayasa Lalu Lintas", menyatakan bahwa aturan di jalan meliputi hal-hal sebagai berikut:

- a. *Right of away*,
- b. Batas kecepatan,
- c. Rambu, sinyal dan marka,
- d. Alat pengendali, dan lain-lain.

Pasal 21(1) UU No. 14/1992 dalam buku yang berjudul "Rekayasa Lalu Lintas" karya Leksmono Suryo Putranto (2008:117) tentang Lalu Lintas dan Angkutan Jalan, tata cara berlalu lintas di jalan (*right of away*) adalah dengan

mengambil jalur sebelah kiri. Batas kecepatan diatur dalam Pasal 80 PP No. 43/1993, marka jalan dijelaskan dalam Pasal 19(1) PP No. 43/1993 dan untuk rambu-rambu lalu lintas diatur dalam pasal 17 PP No. 43/1993 yang terdiri atas empat golongan (Lekmono, 2008:117):

- a. Rambu peringatan (sebagian besar berwarna dasar kuning), digunakan untuk menyatakan peringatan bahaya atau tempat berbahaya.
- b. Rambu larangan (sebagian besar berwarna dasar putih dan bergaris tepi merah), digunakan untuk menyatakan perbuatan yang dilarang dilakukan oleh pemakai jalan.
- c. Rambu perintah (sebagian besar berwarna dasar biru), digunakan untuk menyatakan perintah yang wajib dilakukan oleh pemakai jalan.
- d. Rambu petunjuk (sebagian berwarna dasar putih bergaris tepi biru, berwarna dasar hijau atau cokelat), digunakan untuk menyatakan petunjuk mengenai jurusan, jalan, situasi, kota, tempat, pengaturan, fasilitas dan lain-lain.



Gambar 1. Contoh Rambu-Rambu Lalu Lintas

Macam-macam rambu peringatan antara lain adalah rambu lampu lalu lintas, rambu persimpangan jalan, rambu jalan menanjak dan menurun, dan lain-lain. Contoh dari rambu larangan adalah rambu larangan berhenti, rambu larangan parkir, rambu larangan melewati jalan, dan lain-lain. Contoh rambu perintah adalah rambu untuk berbelok kanan/kiri, rambu untuk melewati jalan berputar, dan lain-lain. Sedangkan untuk rambu petunjuk, contohnya adalah rambu petunjuk jalan, rambu petunjuk jalan buntu, rambu awal daerah/wilayah kota atau akhir daerah kota, dan lain-lain.

Rambu-rambu lalu lintas, marka jalan, dan alat pengatur isyarat lalu lintas merupakan peralatan yang dipergunakan agar kendaraan dapat berjalan dengan cepat, aman, selamat, lancar, tertib, dan teratur (Zulfiar, 2010:50). Rambu lalu lintas yang terdiri dari empat jenis diatas digunakan agar pemakai jalan merasa nyaman dan aman saat sedang berkendara atau berjalan di jalan raya. Rambu lalu lintas biasa ditempatkan di tempat-tempat yang sering terjadi kecelakaan atau ketidaktertiban pengguna jalan. Contoh dari pemasangan rambu lalu lintas antar lain adalah lampu lalu lintas yang dipasang di persimpangan jalan untuk mengatur lalu lintas kendaraan bermotor, rambu peringatan kereta api ditempatkan pada perlintasan jalan dengan jalur rel kereta api, dan lain-lain. Zulfiar (2010:51) menyebutkan bahwa penempatan rambu lalu lintas ditentukan setelah dipelajari kondisi jalan dan lalu lintas, umumnya diletakkan agar mudah dilihat oleh pengemudi. Marka jalan biasanya berbentuk garis yang bisa berupa petunjuk atau menunjukkan kondisi suatu tempat atau pembatas.

Rekayasa lalu lintas berupa perancangan aturan-aturan lalu lintas di jalan raya ditujukan tidak lain adalah untuk mengatur tata tertib lalu lintas. Pengguna

jalan harus mentaati dan menjalankan setiap aturan-aturan yang ada sehingga akan tercipta kenyamanan dan keamanan berlalu lintas. Banyaknya pertumbuhan kendaraan bermotor setiap tahun akan membuat aturan-aturan di jalan harus semakin diperketat dan diberikan sanksi yang berat bagi pelanggarnya.

5. Android

Android merupakan sistem operasi berbasis Linux bagi telepon seluler seperti telepon pintar dan komputer tablet (Safaat, 2012:1). Android telah menyediakan *platform* terbuka yang ditujukan terutama kepada para pengembang yang ingin mengembangkan aplikasi dengan *platform* android. Yuniar (2012:3) dalam bukunya yang berjudul Sistem Operasi Andal Android, mendefinisikan android sebagai perangkat lunak (*software*) sistem operasi yang memakai basis kode komputer yang dapat didistribusikan secara terbuka atau *open source* sehingga pengguna bisa membuat aplikasi baru di dalamnya.

Pengembang aplikasi android dipermudah dengan banyaknya *tools* yang disediakan untuk mengembangkan aplikasi dengan *platform* ini. *Tools* yang disediakan mulai dari aplikasi yang digunakan untuk perancangan aplikasi android sampai *tools* untuk menjalankan aplikasi android tanpa menggunakan *handphone* android. Selain itu, Android memiliki beberapa versi dalam perkembangannya hingga sekarang.

a. Versi Android

Versi android merupakan beberapa sistem operasi android yang berkembang terus menerus dari awal kemunculannya hingga sekarang.

Perkembangan versi android menurut Jazi Eko (2013:6-12) adalah sebagai berikut:

1) Android versi 1.1

Android versi 1.1 dirilis oleh Google pada bulan Maret 2009. Android versi ini dilengkapi dengan fitur-fitur seperti jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan gmail, dan pemberitahuan email.

2) Android versi 1.5 (Cupcake)

Perkembangan android selanjutnya adalah pada pertengahan Mei 2009 dimana Google merilis telepon seluler Android dan juga SDK (*Software Development Kit*). Sistem operasi android versi 1.5 ini menggunakan kode nama Cupcake. Pembaharuan pada sistem operasi ini terdapat pada penambahan fitur dalam seluler yaitu kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3) Android versi 1.6 (Donut)

Sistem operasi android versi 1.6 (Donut) dirilis pada bulan September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan indikator baterai dan kontrol applet VPN. Fitur lain yang dikembangkan adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang diintegrasikan, CDMA/EVDO, *Gesture*, dan *Text-to-speech engine*, kemampuan dial kontak,

teknologi *text to change speech* yang belum pernah ada pada ponsel lainnya saat itu, dan juga kemampuan resolusi VWGA.

4) Android versi 2.0/2.1 (Eclair)

Android versi 2.0/2.1 yang diberi kode nama Eclair ini dirilis pada bulan Desember 2009. Perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 32, MP, digital *zoom* dan *Buletooth* 2.1.

Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik untuk memperkenalkan sistem operasi android dan agar mampu bergerak cepat dalam persaingan perangkat lunak selanjutnya. Perkembangan selanjutnya adalah banyak aplikasi-aplikasi terkenal yang diubah ke dalam sistem operasi android. Contoh dari aplikasi tersebut adalah Shazam, *Backgrounds*, dan WeatherBug. Aplikasi-aplikasi tersebut merupakan hasil kerjasama dengan pihak ketiga yang memiliki minat untuk menyalurkan aplikasi mereka dalam sistem operasi android.

5) Android versi 2.2 (Froyo)

Android dengan kode nama Froyo ini dirilis pada bulan Mei 2010. Perubahan yang dilakukan adalah dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

6) Android versi 2.3 (Gingerbread)

Android versi 2.3 dirilis pada bulan Desember 2010 yang dinamakan Gingerbread. Perubahan yang dilakukan adalah peningkatan kemampuan permainan, peningkatan fungsi *copy paste*, layar antarmuka (*user interface*) yang sudah didesain ulang, dukungan format video VP8 dan WebM, bermacam-macam efek audio baru, dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7) Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk digunakan pada tablet yang memiliki ukuran layar yang lebih besar. *User interface* pada Honeycomb dibuat berbeda karena sudah didesain untuk tablet. Android sudah mendukung *multi prosesor* dan juga akselerasi perangkat keras untuk grafis. Peningkatan fitur tersebut membuat kinerja dari sistem operasi android menjadi lebih cepat dan animasi yang lebih baik.

8) Android versi 4.0 (Ice Cream Sandwich)

Android versi 4.0 diperkenalkan pada bulan Oktober 2011. Fitur-fitur yang dipakai adalah fitur yang telah dikembangkan pada Honeycomb. Android versi ICS juga mengenalkan fitur-fitur baru seperti mampu membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, kontak jaringan sosial terpadu, perangkat tambahan fotografi, pencarian email secara *offline*, dan berbagi informasi dengan menggunakan NFC.

9) Android versi 4.1 (Jelly Bean)

Android versi 4.1 diluncurkan pada acara Google I/O yang membawa sejumlah keunggulan dan fitur baru. Adapaun penambahan fitur baru

diantaranya yaitu meningkatkan input keyboard, desain baru fitur pencarian, UI yang didesain baru dan pencarian menggunakan fitur *Voice Search* yang lebih cepat dan mudah.

Pembaharuan yang lain adalah pada fitur Google Now. Fitur ini mampu memberikan informasi yang tepat pada waktu yang tepat. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas, ataupun hasil pertandingan olahraga.

b. Android SDK

Android SDK merupakan aplikasi atau *tools* pembangun untuk sistem operasi android dimana aplikasi android biasanya menggunakan bahasa java dalam pengembangannya. Android SDK atau *Software Development Kit* menyediakan kebutuhan-kebutuhan pengembang dalam melakukan pengembangan perangkat lunak. Dengan menggunakan SDK, pengembang dapat membuat perangkat lunak pada *platform* tertentu. Android SDK terdapat *programming language, API system, debugging tools*, dan *IDE (Integrated Development Environment)*.

Tools yang disediakan oleh SDK yang sangat bermanfaat bagi pengembang aplikasi android salah satunya adalah android *emulator*. *Emulator android* merupakan *tools* yang mengemulasi sistem operasi android SDK layaknya pada *handset* android. Penggunaan *emulator* android, pengembang dapat menjalankan sistem operasi android seperti internet, GPS, layar sentuh, SD card, dan lain-lain.

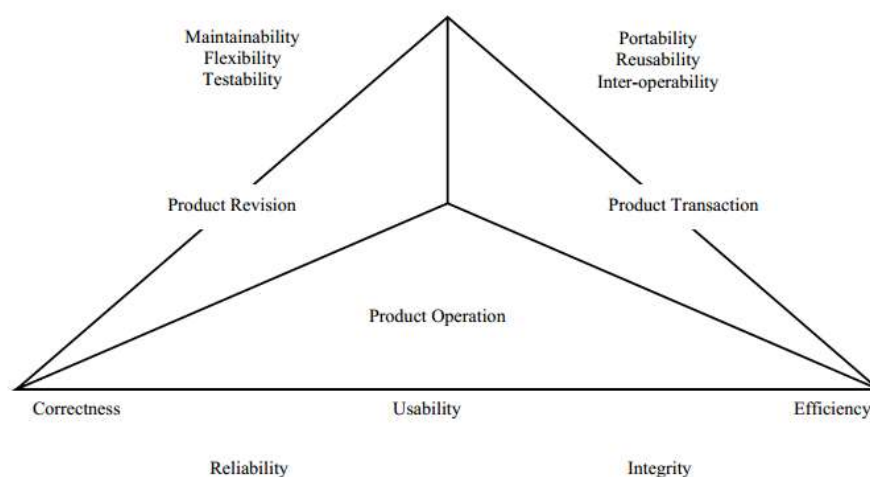
Emulator android dalam penelitian ini dapat digunakan sebagai *tools* untuk melakukan pengujian terhadap aplikasi yang dikembangkan. Penggunaan

emulator ini akan mempermudah pengembang untuk melakukan pengembangan dan pengujian perangkat lunak tanpa harus di *install* ke perangkat android.

6. Software Quality

Software *quality* atau kualitas perangkat lunak dapat didefinisikan sebagai suatu proses perangkat lunak yang efektif diterapkan dalam arti kata proses perangkat lunak yang menyediakan nilai yang dapat diukur untuk mereka yang menghasilkan dan untuk mereka yang menghasilkannya (Pressman, 2010:485). Beberapa tolak ukur yang digunakan untuk mengukur kualitas perangkat lunak, antara lain adalah faktor – faktor kualitas perangkat lunak menurut McCall dan ISO 9126.

Faktor-faktor kualitas perangkat lunak menurut McCall berfokus pada tiga aspek penting dari suatu produk perangkat lunak: karakteristik-karakteristik operasionalnya (*product operation*), kemampuan untuk segera berubah (*product transition*), dan kemampuannya untuk beradaptasi pada lingkungan baru (*product revision*) (Pressman, 2010:487).



Gambar 2. Faktor-Faktor Kualitas Perangkat Lunak Menurut McCall

Faktor-faktor yang digambarkan pada Gambar 2 memiliki deskripsi-deskripsi sebagai berikut:

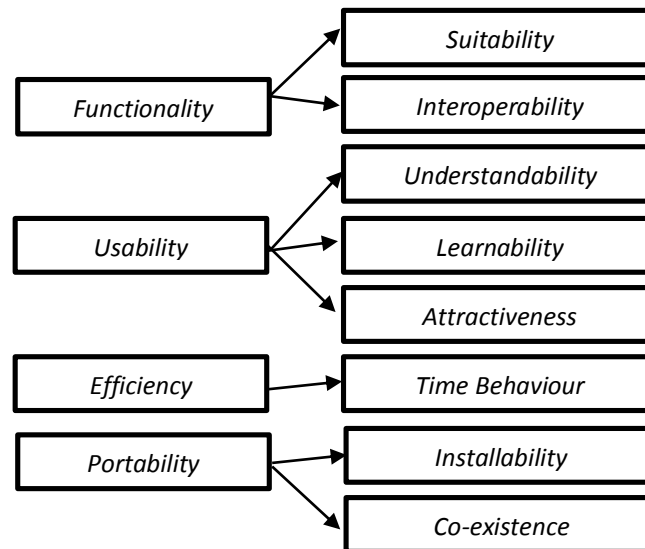
- a. Kebenaran (*correctness*) yang berkaitan dengan bagaimana program akan memberikan hasil sesuai dengan spesifikasi yang telah ditetapkan sebelumnya dan memenuhi sasaran-sasaran pelanggan.
- b. Keandalan (*Reliability*) yang berkaitan dengan bagaimana suatu program diharapkan dapat melakukan fungsi-fungsi tertentu sesuai dengan tingkat ketelitian yang diinginkan.
- c. Efisiensi (*Efficiency*) yang berkaitan dengan jumlah sumber daya komputer yang diperlukan program untuk mampu melaksanakan fungsinya secara baik dan benar.
- d. Integritas (*Integrity*) yang berkaitan dengan bagaimana akses ke perangkat lunak atau ke data oleh orang-orang yang tidak terotorisasi dapat dikendalikan.
- e. Penggunaan (*Usability*) yang berkaitan dengan bagaimana besarnya usaha yang diperlukan untuk mempelajari, mengoperasikan, menyediakan asupan (*input*), dan menafsirkan luaran (*output*) untuk suatu program.
- f. Kemampuan untuk dipelihara (*Maintainability*) yang berkaitan dengan besarnya usaha yang diperlukan untuk melokalisasi dan membetulkan kesalahan-kesalahan yang dapat ditemukan dalam program.
- g. Fleksibilitas (*Flexibility*) yang berkaitan dengan besarnya usaha yang diperlukan untuk memodifikasi suatu program yang bersifat operasional.
- h. Kemampuan untuk menghadapi pengujian (*Testability*) yang berkaitan dengan besarnya usaha yang diperlukan untuk melakukan pengujian atas suatu

program dengan tujuan untuk memastikan bahwa program itu melaksanakan fungsi yang diharapkan.

- i. Portabilitas (*Portability*) yang berkaitan dengan besarnya usaha yang diperlukan untuk mentransfer program dari suatu perangkat keras dan/atau lingkungan perangkat lunak sistem ke perangkat keras dan/atau lingkungan perangkat lunak sistem lainnya.
- j. Penggunaan ulang (*Reusability*) yang berkaitan dengan bagaimana suatu program dapat digunakan ulang di aplikasi/program yang lainnya.
- k. Interoperabilitas (*Interoperability*) yang berkaitan dengan besarnya usaha yang diperlukan untuk menggantikan bagian suatu sistem dengan bagian sistem yang lainnya.

Tolak ukur yang lainnya dalam pengujian kualitas perangkat lunak adalah dengan mengacu pada faktor-faktor kualitas perangkat lunak menurut ISO 9126. Faktor kualitas perangkat lunak menurut ISO 9126 dijelaskan oleh Pressman (2010:488) yaitu terdiri atas enam atribut kualitas kunci, antara lain: *functionality, reliability, usability, efficiency, maintainability*, dan *portability*.

Atribut kualitas perangkat lunak dipilih berdasarkan jenis perangkat lunak yang akan dikembangkan. Gregor (2008:35) memilih empat dari enam faktor kualitas perangkat lunak menurut ISO 9126 pada pengembangan aplikasi entertainment seperti *game* . Empat faktor kualitas perangkat lunak tersebut adalah sebagai berikut:



Gambar 3. Bagan *Software Quality* untuk Aplikasi *Game* dengan ISO 9126 (Gregor, 2008)

Keempat faktor yang dipilih diatas digunakan untuk menguji kualitas perangkat lunak dari aspek fungsionalitas, kemudahan penggunaan, efisiensi, dan portabilitas aplikasi *game* yang dikembangkan.

7. ISO 9126

International Organization of Standardization (ISO) atau organisasi internasional untuk standarisasi telah mendefinisikan satu set standar yang berkaitan dengan perangkat lunak. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kualitas kunci untuk suatu perangkat lunak komputer (Pressman, 2010:488).

International *Standard Organization* melalui ISO 9126 mengidentifikasikan enam karakteristik dalam aspek *software quality* yang meliputi (Pressman, 2010:488):

- a. Fungsionalitas (*functionality*) berkaitan tentang derajat bagaimana perangkat lunak memenuhi kebutuhan yang telah ditetapkan sebelumnya dan memiliki subatribut-subatribut sebagai berikut :
- 1) *Suitability* (kecocokan)
 - 2) *Accuracy* (akurasi)
 - 3) *Interoperability* (interoperabilitas)
 - 4) *Functionality Compliance* (kesesuaian fungsionalitas)
 - 5) *Security* (keamanan)
- b. Keandalan (*reliability*) berkaitan dengan jumlah waktu penggunaan perangkat lunak yang tersedia dan memiliki subatribut-subatribut sebagai berikut :
- 1) *Maturity* (kematangan)
 - 2) *Fault Tolerance* (toleransi kesalahan)
 - 3) *Recoverability* (kemampuan untuk melakukan pemulihan)
 - 4) *Reliability Compliance* (kesesuaian reliabilitas)
- c. Kemudahan pengguna (*usability*) berkaitan dengan derajat tentang bagaimana kemudahan perangkat lunak digunakan, dimana hal ini seringkali diindikasikan menggunakan subatribut-subatribut sebagai berikut :
- 1) *Understandability* (kemudahan untuk dipahami)
 - 2) *Learnability* (kemudahan untuk dipelajari)
 - 3) *Operability* (operabilitas)
 - 4) *Attractiveness* (menarik)
 - 5) *Usability Compliance* (kesesuaian usabilitas)

d. Efisiensi (*efficiency*) berkaitan tentang derajat penggunaan sumber daya sistem secara optimal, dimana hal ini diindikasikan oleh subatribut-subatribut sebagai berikut :

- 1) *Time Behaviour* (perilaku waktu)
- 2) *Resource Utilization* (perilaku sumber daya)
- 3) *Efficiency Compliance* (kesesuaian efisiensi)

e. Kemudahan pemeliharaan (*Maintanability*) berkaitan tentang kemudahan yang menentukan bagaimana perbaikan-perbaikan mungkin dilakukan pada suatu perangkat lunak, dimana hal ini diindikasikan menggunakan subatribut-subatribut sebagai berikut :

- 1) *Analysability* (kemampuan untuk dianalisis)
- 2) *Changeability* (kemampuan untuk dilakukan perubahan)
- 3) *Stability* (hal-hal yang berkaitan dengan stabilitas)
- 4) *Testability* (kemampuan untuk dilakukan pengujian)
- 5) *Maintainability Compliance* (kesesuaian maintainabilitas)

f. Portabilitas (*portability*) berkaitan tentang kemudahan bagaimana perangkat lunak dapat dipindahkan dari suatu lingkungan operasional ke lingkungan operasional yang lainnya, yang hal ini diindikasikan menggunakan subatribut-subatribut sebagai berikut:

- 1) *Adaptability* (kemampuan untuk beradaptasi)
- 2) *Installability* (kemampuan untuk diinstall)
- 3) *Co-existence* (kemampuan berjalan dengan perangkat lain)
- 4) *Replaceability* (kemampuan untuk digantikan)
- 5) *Portability Compliance* (kesesuaian portabilitas)

Peneliti pada penelitian ini menggunakan faktor kualitas perangkat lunak menurut ISO 9126. Variabel yang akan digunakan adalah meliputi aspek *functionality*, *efficiency*, *usability*, dan *portability*. Penggunaan variabel tersebut termasuk subatribut-subatributnya yang dipilih sesuai dengan karakteristik perangkat lunak yang dikembangkan yaitu aplikasi *game* "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun. Empat variabel yang dipilih menurut Gregor (2008) dapat digunakan untuk menilai kelayakan dari perangkat lunak *game* yang dikembangkan. Variabel-variabel tersebut selanjutnya akan dipilih untuk subatributnya yang akan digunakan dalam pembuatan instrumen penelitian berupa kuesioner.

Subatribut-subatribut yang akan digunakan dalam penyusunan instrumen untuk penelitian dan pengembangan perangkat lunak "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun adalah sebagai berikut:

a. *Aspek Functionality*

Aspek *functionality* dipilih dalam pengujian perangkat lunak berupa aplikasi *game* karena *functionality* menurut Gregor (2008:35) dalam bukunya yang berjudul "*Product Software Quality*" menyatakan bahwa "*functionality – has high importance for Entertainment application, because these application should execute their functions that are running the animated gaming application*". Subatribut dari aspek *functionality* yang akan digunakan dalam penyusunan instrumen penelitian adalah:

- 1) *Suitability* merupakan kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
- 2) *Interoperability* merupakan kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.

b. Aspek *Efficiency*

Gregor (2008:36) menyebutkan "*efficiency – has medium importance for Entertainment applications, because hey should not be too slow*". Subatribut yang akan digunakan dalam pembuatan instrumen adalah:

- 1) *Time Behaviour* merupakan kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsi yang ditentukan.

c. Aspek *Usability*

Aspek *usability* adalah aspek yang dapat digunakan untuk melihat kelayakan dari beberapa jenis perangkat lunak yang dikembangkan. Jenis aplikasi *game* adalah salah satu jenis perangkat lunak yang dapat dilihat kelayakannya dari aspek *usability*. Gregor (2008:36) menyebutkan "*usability – has medium importance for Entertaintment applications, because we expect that these application should be easy to understand, easy to use, easy to learn and attractive for the user*". Subatribut yang digunakan dari aspek *usability* adalah:

- 1) *Underdstandability* merupakan kemampuan perangkat lunak dalam kemudahan untuk dipahami.
- 2) *Learnability* merupakan kemampuan perangkat lunak dalam kemudahan untuk dipelajari.

3) *Attractiveness* merupakan kemampuan perangkat lunak dalam menarik pengguna.

d. Aspek *Portability*

Aspek *portability* dapat digunakan untuk menguji kelayakan perangkat lunak karena menurut Gregor (2008:36), "*portability – has medium importance for Entertainment applications, because we expect that these applications can be easily installed and run with other applications*". Subatribut yang digunakan dalam penyusunan instrumen kuesioner antara lain:

- 1) *Installability* merupakan kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
- 2) *Adaptability* merupakan kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.

Subatribut *co-existence* peneliti ganti dengan *adaptability* karena perangkat yang dikembangkan adalah aplikasi *game* android dimana aplikasi ini pada pengujian aspek *portability* nantinya akan dilihat kemampuan aplikasi setelah diinstal pada lingkungan yang berbeda-beda. *Adaptability* dilihat dari apakah aplikasi dapat berjalan dan berfungsi semua fitur-fiturnya dengan baik setelah diinstal pada lingkungan yang berbeda-beda. Aspek *adaptability* akan diuji dengan menggunakan teknik observasi.

8. UML (Unified Modeling Language)

UML merupakan metode pemodelan secara visual sebagai sarana untuk merancang dan membuat *software* berorientasi objek. UML didefinisikan sebagai

bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem (Yuni, 2013:36). UML memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*. UML digunakan sebagai bahasa standar untuk pengembangan sebuah perangkat lunak yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan perangkat lunak.

Elemen pembentuk UML adalah *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum. *Building block* terdiri dari beberapa bagian, yaitu:

a. Benda/ *Things*

Benda merupakan abstraksi yang pertama dalam sebuah model. Benda juga merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen-elemen lainnya dari konsep atau fisik. Bentuk dari beberapa benda antara lain:

- 1) *Classes*, merupakan sekelompok dari objek yang mempunyai atribut, operasi, hubungan yang semantik.
- 2) *Collaboration*, merupakan interaksi dan sebuah kumpulan dari kelas atau elemen yang bekerja secara bersama-sama.
- 3) *Use cases*, adalah rangkaian yang saling terkait dan membentuk sistem secara teratur yang berkomunikasi dengan aktor.

- 4) *Nodes*, merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem.

b. Hubungan/*Relationships*

Merupakan alat komunikasi dari benda-benda. Beberapa macam hubungan yang terdapat dalam UML antara lain adalah:

- 1) *Dependency*, merupakan hubungan antara dua benda yang mana sebuah benda berubah mengakibatkan benda satunya akan berubah pula.
- 2) *Association*, merupakan hubungan antar benda struktural yang terhubung diantara objek.
- 3) *Generalizations*, merupakan penggambaran hubungan secara khusus dalam objek anak yang menggantikan objek induk.
- 4) *Realizations*, merupakan hubungan semantik antara pengelompokan yang menjamin adanya ikatan diantaranya.

c. Bagan/*Diagrams*








UML terdiri dari pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Diagram-diagram UML antara lain adalah:

1) Diagram *Use Case*

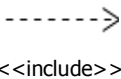
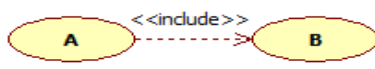
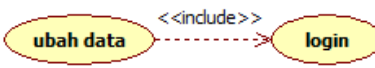
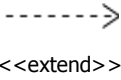
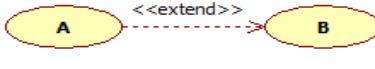
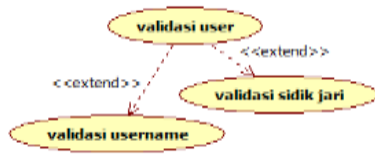
Diagram use case menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. *Use case* sesungguhnya mengatakan "cerita" tentang bagaimana seorang pengguna (yang memainkan satu dari sejumlah peran yang mungkin) yang berinteraksi dengan sistem yang berada di bawah sejumlah situasi dari kondisi yang sifatnya spesifik (Pressman,

2010:160). Diagram *use case* membantu kita dalam menyusun *requirement* sebuah sistem. Diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Yuni, 2013:41). Diagram *use case* digambarkan dalam bentuk simbol yaitu *use case*, aktor, dan relasi. Simbol-simbol diagram *use case* dijelaskan pada tabel berikut:

Tabel 1. Deskripsi Simbol Diagram *Use Case*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem yang dikembangkan. Simbol dari aktor adalah gambar orang tetapi aktor belum tentu orang, jika terdapat sistem lain yang berkomunikasi (memberikan <i>input</i> atau mendapatkan <i>output</i>) dengan sistem yang dikembangkan, maka sistem lain tersebut dapat disebut aktor.
2		<i>Association</i>	Komunikasi atau interaksi antara aktor dengan <i>use case</i> .
3		<i>Use case</i>	Menggambarkan kegiatan yang dilakukan aktor dan sistem untuk mencapai tujuan tertentu. <i>Use case</i> hanya menjelaskan apa yang dilakukan aktor dan sistem bukan bagaimana aktor dan sistem melakukan suatu kegiatan.
4		<i>Dependency</i>	Relasi atau hubungan yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.
5		<i>Generalization</i>	<p>Hubungan khusus antara objek anak/<i>child</i> yang menggantikan objek induk/<i>parent</i>, dimana objek anak berbagi perilaku dan struktur data dari objek induk, misalnya:</p>  <p><i>Use case</i> B merupakan generalisasi dari <i>use case</i> A. <i>Use case</i> B adalah objek anak yang berbagi perilaku dan struktur data dari <i>use case</i> A yang merupakan objek induk.</p> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (<i>parent</i>).</p> <p>Contoh Kasus:</p>  <p><i>Use case</i> tambah data dan <i>use case</i> hapus data adalah generalisasi dari <i>use case</i> kelola data. Arah panah mengarah pada <i>use case</i> kelola data karena <i>use case</i> kelola data adalah <i>parent</i> atau induk.</p>

Tabel 2. Deskripsi Simbol Diagram *Use Case* (lanjutan)

No	Gambar	Nama	Keterangan
6	 <<include>>	Include	<p>Relasi yang menggambarkan bahwa suatu <i>use case</i> memerlukan <i>use case</i> lain dalam menjalankan fungsinya, misalnya:</p>  <p><i>Use case</i> A meng-include <i>use case</i> B, sehingga dalam menjalankan <i>use case</i> A, <i>use case</i> B akan dijalankan terlebih dahulu kemudian baru <i>use case</i> A.</p> <p>Arah panah mengarah pada <i>use case</i> yang harus dijalankan terlebih dahulu.</p> <p>Contoh kasus:</p>  <p><i>Use case</i> ubah data meng-include <i>use case</i> login, sehingga jika ingin mengubah data harus melakukan login terlebih dahulu.</p>
7	 <<extend>>	Extend	<p>Relasi yang menggambarkan bahwa suatu <i>use case</i> merupakan tambahan kegunaan dari <i>use case</i> lain, misalnya:</p>  <p><i>Use case</i> A meng-extend <i>use case</i> B, dimana <i>use case</i> B adalah tambahan kegunaan dari <i>use case</i> A. <i>Use case</i> A dapat dijalankan tanpa harus menjalankan <i>use case</i> B terlebih dahulu.</p> <p>Arah panah mengarah pada <i>use case</i> yang tidak harus dijalankan.</p> <p>Contoh kasus:</p>  <p><i>Use case</i> validasi user meng-extend <i>use case</i> validasi username dan <i>use case</i> validasi sidik jari. <i>Use case</i> validasi username dan validasi sidik jari bersifat pilihan yang tidak harus dijalankan, sehingga <i>use case</i> validasi user dapat dijalankan tanpa harus melewati <i>use case</i> validasi username atau <i>use case</i> validasi sidik jari.</p>

Bagian yang perlu ditekankan dalam pembuatan *use case* diagram adalah “apa” yang diperbuat sistem, bukan “bagaimana” sebuah sistem berjalan. Adi Nugroho (2009:7) menyebutkan bahwa *use case* merupakan deskripsi lengkap tentang interaksi yang terjadi antara para aktor dengan sistem atau perangkat lunak yang sedang kita kembangkan. Aktor digambarkan dengan bentuk manusia dan *use case* digambarkan dengan bentuk elips yang berisi nama *use case* yang bersangkutan.

2) Diagram *Activity*





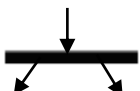

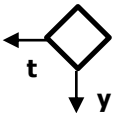
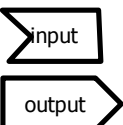
Activity diagram adalah suatu jenis diagram khusus dari *statechart diagram* yang merepresentasikan *state-state* dan transisi-transisi yang terjadi pada akhir operasi-operasi (Adi, 2009:115). Sebuah diagram *activity* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor. *Activity diagram* banyak digunakan untuk mendefinisikan hal-hal berikut:

- a) Rancangan proses bisnis, yaitu digunakan untuk merancang urutan-urutan dari aktivitas sistem yang didefinisikan.
- b) Rancangan pengujian, yaitu bahwa setiap yang dianggap memerlukan sebuah pengujian perlu didefinisikan kasus ujinya.

Diagram *activity* dapat menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Pembuatan diagram *activity* adalah untuk melengkapi *use case* yang telah dibuat sebelumnya dengan memberikan representasi grafis dari aliran interaksi di dalam suatu skenario yang sifatnya spesifik (Pressman, 2010:195). Perbedaan diagram *activity* dengan *use case* adalah jika diagram *activity* menggambarkan proses yang berjalan,

sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Deskripsi dan macam-macam simbol yang terdapat dalam *activity diagram* akan dijelaskan dalam Tabel 3 berikut:

Tabel 3. Deskripsi Simbol Diagram *Activity*.

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Status awal aktivitas sistem. Activity diagram memiliki sebuah status awal yang menandakan awal dari aktivitas sistem.
4		<i>Activity Final Node</i>	Status akhir yang dilakukan sistem. Status akhir menunjukkan akhir aktivitas dari sistem.
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran (aktivitas secara paralel)
6		<i>Join Node</i>	Menunjukkan aktivitas yang digabungkan. Beberapa aktivitas pada tahap tertentu berubah menjadi satu aliran aktivitas.
6		<i>Decisions</i>	Menunjukkan aktivitas yang harus dipilih, apakah pilihan pertama atau kedua. Pilihan alur aktivitas dapat digambarkan dengan inisial "y" atau "ya" dan inisial "t" atau "tidak".
7		<i>Signal</i>	Signal menggambarkan pengirim dan penerima pesan dan aktivitas yang terjadi. Signal terdiri dari dua jenis, yaitu penerima yang digambarkan dengan simbol poligon terbuka, dan sinyal pengirim yang digambarkan dengan simbol convex poligon.

3) Diagram *Class*

Diagram *class* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *class* memiliki sifat statis yaitu menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Diagram *class* memiliki tiga macam hubungan, yaitu *association* yaitu suatu hubungan antara bagian dari dua kelas,

aggregation merupakan salah satu kelas yang merupakan bagian dari suatu kumpulan, dan yang terakhir adalah *generalization* yaitu hubungan turunan dengan mengasumsikan suatu kelas merupakan suatu kelas super dari kelas lain. Pembuatan *class diagram* pada umumnya sangat berkaitan dengan implementasi kode-kode program dengan bahasa pemrograman.

Class diagram menggambarkan struktur dari deskripsi *class*, *package* dan objek beserta hubungan satu sama lain. *Class* memiliki tiga area pokok yaitu nama *class*, atribut, dan metoda. Atribut dan metoda dapat bersifat *private* (tidak dapat dipanggil dari luar *class* yang bersangkutan), *protected* (hanya dapat dipanggil oleh *class* yang bersangkutan), dan *public* (dapat dipanggil oleh siapa saja). Pembuatan diagram *class* ini adalah sebagai bentuk awal proses implementasi kode-kode program.

4) Diagram *Sequence*

Sequence diagram menggambarkan interaksi antar objek dan di sekitar sistem berupa message yang digambarkan terhadap waktu. *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Salah satu bagian dari *sequence diagram* adalah *message* yang digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Bagian lain adalah *activation bar* yang menunjukkan lamanya eksekusi sebuah proses. Penggambaran *sequence diagram* adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau semua *use case* yang telah didefinisikan interaksi

antar *use case*-nya. Semakin banyak *use case* yang dibuat, semakin banyak juga diagram *sequence* yang harus dibuat.

5) Diagram *Collaboration*

Diagram *collaboration* menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi diagram ini lebih fokus pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap pesan pada diagram *collaboration* memiliki angka yang terurut. Pesan dengan tingkatan tertinggi adalah yang memiliki angka 1 dan seterusnya.

6) Diagram *StateChart*

StateChart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lain) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Diagram ini pada umumnya menggambarkan class tertentu dimana satu *class* dapat memiliki lebih dari satu *statechart diagram*. Penggambaran *statechart diagram* berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya. Perpindahan antar *state* memiliki kondisi *guard* yang merupakan syarat terjadinya perpindahan yang bersangkutan. *Action* yang dilakukan sebagai akibat dari *event* tertentu dibuat dengan diawali garis miring.

7) Diagram *Deployment*

Deployment diagram menggambarkan detail bagaimana komponen di-*deploy* dalam bentuk infrastruktur sistem, di mana komponen akan terletak, bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal.

9. Construct 2

Construct 2 merupakan sebuah tool berbasis HTML5 untuk menciptakan sebuah permainan. Construct 2 memiliki fitur-fitur yang mudah untuk digunakan dan dimengerti oleh seorang pemrogram pemula. HTML5 merupakan bahasa markup yang bertujuan untuk penataan dan penyajian konten untuk *World Wide Web*, dan merupakan teknologi inti dari internet yang pada awalnya diusulkan oleh Opera Software. Construct 2 berbeda dengan tools lain yang mengharuskan pemrogram untuk menuliskan baris demi baris agar tercipta sebuah objek, construct 2 sudah berbasis objek sehingga mudah dalam membuat objek-objek dan mengatur atribut-atribut dari objek tersebut.

Construct 2 dikembangkan dengan tujuan untuk memudahkan pada *non-programmer* yang ingin menciptakan sebuah *game* secara *drag and drop* menggunakan editor visual dan berbasis sistem logika perilaku. Editor visual merupakan tempat dimana objek-objek diletakkan atau dibuat, sedangkan pengaturan logika perilaku untuk masing-masing objek yang dibuat dinamakan *event* yang dituliskan dalam *event sheet*. Event dalam construct 2 merupakan kumpulan dari *conditions* dan *actions*. *Conditions* menjelaskan kondisi dari objek dan *actions* adalah aksi yang menggerakkan objek-objek yang dibuat.



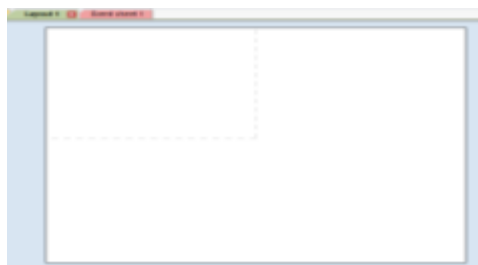
Gambar 4. Tampilan Halaman Awal Construct 2 R130

Construct 2 dirancang untuk pengembangan *game* berbasis 2D. Dengan menggunakan Construct 2, pengembang *game* dapat mem-*publish* aplikasinya ke dalam beberapa *platform*, antara lain:

- a. HTML5 Website,
- b. Google Chrome Webstorage,
- c. Facebook,
- d. Phonegap (Android),
- e. Windows Phone 8.

Construct 2 juga menyediakan bermacam-macam *visual effect* yang menggunakan engine WebGL, dan juga *plugin* dan *behaviour* yang akan membantu pengembang aplikasi dalam menciptakan aplikasi yang menarik dan interaktif. Pemanggilan fungsi-fungsi di Construct 2 hanya menggunakan pengaturan *Event* yang telah disediakan. *Events* merupakan pilihan-pilihan aksi dan kondisi yang akan menjadi penggerak *game*. Construct 2 merupakan *game builder* yang berbasis HTML5, sehingga *game* yang kita buat dapat dijalankan melalui browser seperti Google Chrome sebelum *game* tersebut di-*publish* ke *platform* yang diinginkan. Bagian ruang kerja construct 2 adalah sebagai berikut:

- a. Area kerja construct 2 untuk menggambar berbagai macam objek (objek *sprite*, objek *background*, dan objek-objek lain).



Gambar 5. Tampilan *Layout* Construct 2

- b. Menu *Properties* Construct 2, digunakan untuk mengatur kebutuhan dari objek yang dibuat, seperti mengatur warna *layout*, ukuran objek *sprite*, ukuran *layout*, dan lainnya).



Gambar 6. Tampilan Menu *Properties* dalam Construct 2

- c. *Projects dan Layers*, merupakan menu untuk memilih *project* yang akan dikerjakan dan menu *layer* untuk membuat beberapa *layer* dalam suatu *layout* kerja.



Gambar 7. Tampilan Menu *Projects and Layers* Construct 2

- d. Menu *library* atau kumpulan dari objek-objek yang telah dibuat.



Gambar 8. Tampilan *Library* Construct 2

- e. *Event sheet*, merupakan area kerja construct 2 untuk menuliskan event-event yang akan menggerakkan objek-objek yang telah dibuat.



Gambar 9. Tampilan *Event sheet* Construct 2

B. Penelitian yang Relevan

Beberapa penelitian yang relevan dengan penelitian ini antara lain:

1. Penelitian thesis Universitas Atmajaya Yogyakarta yang dilakukan oleh Lydia Ignacia Setiadi (2011), yang berjudul "Pembangunan Aplikasi Pembelajaran Rambu Lalu Lintas Berbasis Multimedia Interaktif". Penelitian ini relevan dengan penelitian yang dilakukan peneliti pada bagian pengembangan aplikasi tentang pembelajaran rambu lalu lintas. Hasil penelitiannya adalah:
 - a. Pengembang aplikasi berhasil membangun Aplikasi Pembelajaran Rambu Lalu Lintas Berbasis Multimedia Interaktif (RaTas) dengan menggunakan program Adobe Flash CS3.
 - b. Perangkat lunak RaTas dibangun dengan menggunakan teknologi multimedia sehingga dapat memberikan kemudahan dalam mempelajari rambu-rambu lalu lintas.
2. Penelitian skripsi Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta yang dilakukan oleh Arief

Wibowo (2013). Judul penelitiannya adalah Perancangan Iklan Layanan Masyarakat "Tertib Lalu Lintas" Berbasis Animasi 2D sebagai Media Sosialisasi Ditlantas POLDA DIY. Penelitian ini relevan dengan penelitian yang dilakukan peneliti pada bagian pembuatan media pengenalan tertib lalu lintas. Hasil penelitian yang telah dilakukan adalah:

- a. Iklan animasi "Tertib Lalu Lintas" berhasil dikembangkan dengan menggunakan dengan melalui 3 tahapan yaitu proses pra produksi, produksi, dan pasca produksi.
 - b. Pembuatan gambar background yang dilakukan dengan menggambar langsung pada dokumen Adobe Flash dapat menghemat waktu dan biaya.
 - c. Penggunaan gambar dan background dalam ILM "Tertib Lalu Lintas" berbasis vektor.
3. Penelitian yang dilakukan oleh Miftah Yanuar (2013), mahasiswa Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta dengan judul penelitian "Aplikasi Informasi *Safety Riding* Sepeda Motor Berbasis Android". Penelitian ini relevan dengan penelitian yang dilakukan peneliti pada bagian pengembangan aplikasi berbasis android mengenai informasi keselamatan di jalan raya. Hasil penelitian yang telah dilakukan adalah:
- a. Aplikasi *Safety Riding* Sepeda Motor Berbasis Android berhasil dibuat.
 - b. Aplikasi ini dapat menampilkan informasi berkendara secara aman dan pengenalan berbagai rambu lalu lintas.
 - c. Aplikasi ini dapat menampilkan menu penjualan perlengkapan seputar *safety riding*.
 - d. Aplikasi mudah digunakan dan dipelajari *user*.

C. Kerangka Berfikir

Kerangka pikir dari penelitian ini merupakan gambaran logis tentang bagaimana variabel-variabel saling berhubungan. Penyusunan kerangka pikir diawali dengan adanya variabel-variabel yang mewakili masalah penelitian. Pengembangan perangkat lunak "*Need For Safety*" sebagai sarana pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun ini hanya mengacu pada uji kualitas perangkat lunak menurut ISO 9126. Aplikasi "*Need For Safety*" dikaji dengan menggunakan empat aspek *software quality* ISO 9126, yang meliputi aspek *functionality*, *efficiency*, *usability*, dan *portability*.

Penelitian ini diawali dengan adanya permasalahan yang muncul sehingga memerlukan alternatif penyelesaian. Penyelesaian masalah yang dipakai adalah dengan membuat aplikasi "*Need For Safety*". Setelah aplikasi dibuat, dilakukan pengujian dari sisi pengembang yaitu uji *white box* dan uji *black box*. Pengujian selanjutnya adalah pengujian *alpha* yang dilakukan kepada ahli dalam bidang rekayasa perangkat lunak dan juga pengujian *beta* yang menggunakan kuesioner yang berisikan empat aspek pengujian kualitas perangkat lunak menurut ISO 9126 yaitu aspek *functionality*, *efficiency*, *usability*, dan *portability*. Pengujian beta diberikan kepada beberapa responden untuk menilai kelayakan perangkat lunak berdasarkan aspek kualitas perangkat lunak ISO 9126 yang telah ditentukan.

Berdasarkan data yang diperoleh dari proses pengujian diperoleh keterangan bagaimana kualitas perangkat lunak yang dikembangkan menurut kaidah rekayasa perangkat lunak. Keterangan yang dihasilkan nantinya akan dijadikan sebagai acuan apakah perangkat lunak yang dikembangkan layak digunakan oleh pengguna atau tidak.

BAB III

METODE PENELITIAN

A. Model Pengembangan

Jenis penelitian yang digunakan dalam pengembangan perangkat lunak "*Need For Safety*" sebagai media pengenalan rambu-rambu lalu lintas kepada anak-anak ini termasuk dalam jenis penelitian riset dan pengembangan (*research and development*). Jenis penelitian riset dan pengembangan (*research and development*) merupakan metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut (Sugiyono, 2010). Langkah-langkah penelitian yang dilakukan dalam penelitian ini adaah sebagai berikut :



Gambar 10. Langkah-Langkah Pembuatan Aplikasi "*Need For Safety*"

B. Objek Penelitian

Objek penelitian yang akan diteliti adalah fokus pada Aplikasi "*Need For Safety*". Aplikasi pengenalan rambu-rambu lalu lintas ini digunakan sebagai media pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun.

C. Tempat dan Waktu Penelitian

Penelitian perangkat lunak "*Need For Safety*" dilakukan di Polsek Bulaksumur Sleman dan SDN Pujokusuman 1 Yogyakarta. Tempat penelitian dipilih peneliti karena SDN Pujokusuman 1 merupakan SD Percontohan lalu lintas di DIY, sedangkan Polsek Bulaksumur dipilih untuk menilai perangkat lunak yang dikembangkan. Pelaksanaan pengembangan dan penelitian dilakukan pada bulan November 2013 sampai selesai.

D. Sampel Penelitian

Sampel merupakan bagian dari jumlah dan karakteristik yang dimiliki oleh populasi (Sugiyono, 2013:81). Sampel penelitian pada pengujian perangkat lunak "*Need For Safety*" sebagai media pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun menggunakan teknik *purposive sampling*. *Sampling purposive* adalah teknik penentuan sampel dengan pertimbangan tertentu (Sugiyono, 2013:85).

Teknik *purposive sampling* dalam pengujian perangkat lunak "*Need For Safety*" adalah dengan memilih beberapa ahli di bidang tertentu. Pada penelitian ini akan digunakan ukuran sampel sebanyak 27 sampel. Sampel penelitian nantinya akan dibagi dalam dua kelompok. Kelompok yang pertama adalah sampel yang melakukan pengujian *alpha* dan kelompok selanjutnya adalah sampel yang melakukan pengujian *beta* pada perangkat lunak yang

dikembangkan. Pada pengujian *alpha*, peneliti memilih ahli dalam bidang *software engineering* untuk melakukan analisa terhadap kelayakan dan unjuk kerja perangkat lunak. Kelompok sampel pada pengujian *beta* merupakan kelompok sampel yang peneliti pilih untuk menilai kelayakan perangkat lunak berdasarkan aspek kualitas perangkat lunak. Kelompok sampel yang dipilih adalah beberapa responden yang dianggap mengerti terhadap aplikasi yang dikembangkan.

E. Variabel Penelitian

1. Variabel Penelitian

Variabel penelitian yang digunakan dalam penelitian dan pengembangan perangkat lunak "*Need For Safety*" adalah sebagai berikut :

- a. *Functionality*
- b. *Efficiency*
- c. *Usability*
- d. *Portability*

2. Definisi Operasional Variabel

Definisi mengenai variabel di atas menurut ISO 9126 yang dikutip oleh Pressman (2010:488-489) adalah sebagai berikut:

- a. *Functionality* merupakan derajat tentang bagaimana perangkat lunak memenuhi kebutuhan yang telah ditetapkan sebelumnya.
- b. *Efficiency* merupakan derajat penggunaan sumber daya sistem secara optimal.
- c. *Usability* merupakan derajat tentang bagaimana kemudahan perangkat lunak digunakan.

- d. *Portability* merupakan derajat tentang bagaimana perangkat lunak dapat dipindahkan dari suatu lingkungan operasional ke lingkungan operasional yang lainnya.

F. Prosedur Pengembangan

Penelitian aspek *software quality* dalam pengembangan perangkat lunak "*Need For Safety*" ini menggunakan metode pengembangan *Linear Sequential*. Metode pengembangan *Linear Sequential* atau disebut juga model *waterfall* merupakan model pengembangan klasik yang paling banyak digunakan dalam *Software Engineering*. Model *Linear Sequential* mengusulkan sebuah pendekatan kepada pengembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tahap analisis, desain, kode, pengujian, dan pemeliharaan (Pressman, 2012:46). Metode pengembangan *Linear Sequential* pada penelitian ini meliputi tahapan sebagai berikut:

1. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap menganalisis kebutuhan dari perangkat lunak yang dikembangkan. Tahap analisis kebutuhan dilakukan dengan tujuan untuk memberikan kemudahan dalam pengembangan dan pengoperasian perangkat lunak oleh pengguna nantinya. Kebutuhan yang dianalisis antara lain adalah kebutuhan data, kebutuhan *user*, dan kebutuhan sistem. Data utama pada perangkat lunak *game* edukasi ini adalah data-data rambu lalu lintas yang meliputi rambu peringatan, rambu perintah, rambu, rambu larangan, dan rambu petunjuk. Setiap rambu akan dijadikan sebagai materi permainan di setiap level *game*.

Kebutuhan selanjutnya adalah kebutuhan *user*. Kebutuhan *user* terdiri atas *user interface* dan fitur-fitur dalam *game*. Perangkat lunak *game* edukasi pengenalan rambu lalu lintas ini ditujukan untuk anak-anak usia 6-12 tahun, oleh karena itu *user interface game* juga disesuaikan dengan profil anak-anak usia 6-12 tahun. Kebutuhan tersebut dapat diadaptasi dengan pemilihan warna untuk objek *background* dengan warna yang cerah dan menyenangkan untuk dilihat. Pengaturan warna untuk objek *background* dan objek lain yang sesuai membuat anak sedapat mungkin merasakan “dunianya” saat memainkan *game*. Permainan dalam *game* ini juga dikembangkan untuk dimainkan secara santai, yang memberikan kesan rileks dan menyenangkan. Permainan tidak menampilkan batasan waktu untuk menyelesaikan tiap levelnya sehingga membuat *user* lebih santai dalam memainkan permainannya. Permainan yang bersifat santai bukan berarti tidak menantang, *user* pada permainan ini akan ditantang untuk mendapatkan skor atau nilai setinggi-tingginya dalam menyelesaikan level permainannya. Fitur lain dalam aplikasi ini adalah fitur pengenalan rambu lalu lintas yang berisi definisi beserta contoh dari macam-macam rambu lalu lintas.

Kebutuhan sistem dalam pengembangan *game* pengenalan rambu lalu lintas ini dibuat dengan konsep bernuansa animasi. Perangkat lunak yang digunakan dalam pengembangannya adalah *Construct2*. *Construct2* merupakan *software* buatan *scirra*, perusahaan dari kota London, Inggris. *Construct2* dirancang untuk *game* berbasis 2D yang nantinya *game* dapat di *publish* ke beberapa *platform* seperti HTML5, Phonegap (android), dan Windows Phone 8. Pengembangan *game* pengenalan rambu lalu lintas ini membutuhkan aplikasi-aplikasi pendukung lain seperti, *Adobe Photoshop*, *CorelDraw*, dan lain-lain, yang

digunakan untuk menghasilkan sebuah desain yang menarik. Aplikasi ini nantinya akan digunakan untuk *platform* android dengan spesifikasi minimum yaitu android 2.3 (*Gingerbread*). Sistem Operasi android *Gingerbread* menggunakan spesifikasi prosesor 1 GHz atau lebih dan RAM minimal 512 MB. Spesifikasi android dibawah RAM 512 MB, aplikasi tidak akan berjalan secara maksimal.

2. Desain

Desain perangkat lunak adalah proses perencanaan, pembuatan dan penggambaran dari sistem yang akan dikembangkan. Dalam proses perencanaan perangkat lunak, dibuat skenario yang dapat digunakan untuk mengidentifikasi sejumlah penggunaan sistem yang mungkin. Skenario-skenario ini, sering dinamakan sebagai *use case*, menyediakan deskripsi yang rinci tentang bagaimana sistem akan digunakan (Pressman, 2010:159).

Aspek yang dikembangkan meliputi perancangan *Unified Modelling Language* (UML) berupa *use case diagram* dan dilengkapi dengan *activity diagram* serta perancangan antarmuka perangkat lunak atau *user interface*. *Activity diagram* dibuat setelah perancangan *use case* dengan tujuan untuk memberikan representasi grafis dari aliran interaksi yang terdapat dalam *use case*. *Activity diagram* menambahkan rincian-rincian dalam *use case* yang tidak secara langsung dijelaskan (tetapi tersamar) (Pressman, 2010:195).

a. *Use case Diagram*

Use case merepresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Pembuatan *use case* diawali dengan mendefinisikan sejumlah aktor yang akan terlibat di dalam sistem. Aktor merupakan sejumlah orang (atau sarana) yang

berbeda yang menggunakan sistem atau produk di dalam konteks fungsi-fungsi dan perilaku-perilaku yang harus dideskripsikan selanjutnya (Pressman, 2010:160). Sejumlah aktor kemudian di definisikan fungsi-fungsi atau aktivitas-aktivitas dengan cara melihat dari daftar fungsi yang dikehendaki ada dalam sistem yang akan dikembangkan. Aktor dalam sistem perangkat lunak yang akan dikembangkan dinamakan sebagai *user* yaitu pengguna dari sistem, dan aktivitas-aktivitas yang mungkin dapat dilakukan aktor di dalam sistem digambarkan dalam bentuk sejumlah *use case*.

Aplikasi yang dikembangkan memiliki satu aktor yaitu pengguna aplikasi. Komunikasi aktor dengan sistem digambarkan dengan beberapa *use case*, yaitu *use case* memulai aplikasi, *use case* keluar aplikasi, *use case* bantuan, *use case on/off* musik *game*, *use case* lihat skor, *use case* mulai bermain, dan *use case* ayo belajar rambu. *Use case* lihat skor memiliki relasi *include* terhadap *use case get score data*, karena dalam menjalankan fungsi lihat skor, *use case get score data* akan dijalankan terlebih dahulu untuk mendapatkan skor permainan yang tersimpan dalam webstorage. *Use case get score data* memiliki relasi *include* terhadap *use case* mulai bermain, karena untuk mendapatkan skor, terlebih dahulu harus memainkan permainan pada menu mulai bermain.

Fitur ayo belajar rambu memiliki beberapa submenu yaitu belajar materi tentang pengertian rambu, rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk, sehingga digambarkan *use case* ayo belajar rambu memiliki relasi *extend* terhadap *use case* pengertian rambu, *use case* rambu peringatan, *use case* rambu larangan, *use case* rambu perintah, dan *use case* rambu petunjuk. *Use case* ayo belajar rambu dapat dijalankan tanpa harus menjalankan

use case pengertian rambu, *use case* rambu peringatan, *use case* rambu larangan, *use case* rambu perintah, atau *use case* rambu petunjuk. Definisi aktor dan *use case* aplikasi *Need For Safety* adalah sebagai berikut:

1) Definisi Aktor

Tabel 4. Definisi Aktor

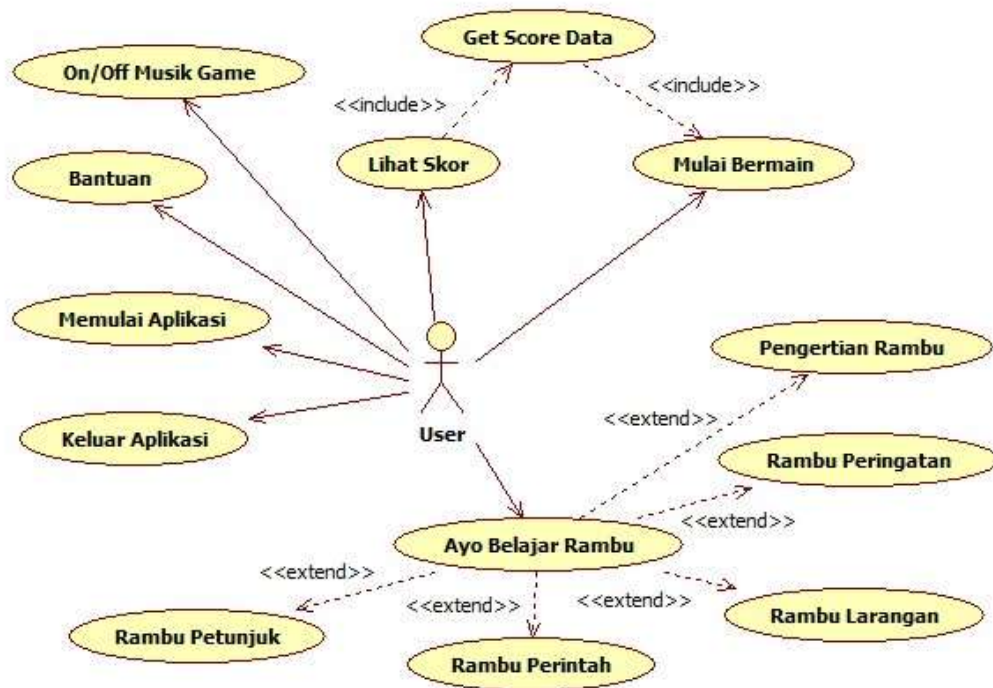
No	Aktor	Deskripsi
1	<i>User</i>	<i>User</i> merupakan aktor dari perangkat lunak, <i>user</i> dapat memulai aplikasi, keluar dari aplikasi, memilih fitur mulai bermain, ayo belajar rambu, bantuan, melihat skor, dan menonaktifkan atau mengaktifkan musik <i>game</i> melalui menu utama.

2) Definisi *Use case*

Tabel 5. Definisi *Use case* Sistem

No	<i>Use case</i>	Deskripsi
1	Memulai Aplikasi	Merupakan proses awal ketika user menjalankan aplikasi. <i>User</i> dapat memulai aplikasi dan akan masuk ke menu utama aplikasi yang berisi fitur-fitur dalam aplikasi.
2	<i>On/Off</i> Musik <i>Game</i>	Merupakan fitur untuk menonaktifkan atau mengaktifkan musik <i>game</i> .
3	Lihat Skor	Merupakan fitur melihat skor yang tersimpan di webstorage.
4	<i>Get Score Data</i>	Proses mendapatkan data berupa nilai atau skor <i>user</i> setelah memainkan permainan pada menu mulai bermain. Nilai atau skor tersebut akan disimpan dalam webstorage.
5	Mulai Bermain	Merupakan fitur untuk mulai bermain <i>game</i> mencocokkan gambar rambu-rambu lalu lintas yang terdiri dari empat level permainan.
6	Ayo Belajar Rambu	Merupakan fitur untuk mempelajari materi rambu lalu lintas. Fitur ini terbagi menjadi beberapa submenu tambahan yaitu pengertian rambu, rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk.
7	Rambu Pengertian	Merupakan fitur tambahan dari ayo belajar rambu untuk melihat definisi rambu lalu lintas.
8	Rambu Peringatan	Merupakan fitur tambahan dari ayo belajar rambu untuk melihat definisi dan contoh rambu peringatan.
9	Rambu Larangan	Merupakan fitur tambahan dari ayo belajar rambu untuk melihat definisi dan contoh rambu larangan.
10	Rambu Perintah	Merupakan fitur tambahan dari ayo belajar rambu untuk melihat definisi dan contoh rambu perintah.
11	Rambu Petunjuk	Merupakan fitur tambahan dari ayo belajar rambu untuk melihat definisi dan contoh rambu petunjuk.
12	Bantuan	Merupakan fitur untuk melihat cara penggunaan aplikasi.
13	Keluar Aplikasi	Merupakan proses ketika <i>user</i> keluar dari aplikasi.

Aktor dan *use case* yang telah didefinisikan pada Tabel 5, kemudian digambarkan menggunakan simbol aktor, *use case*, dan relasi antara aktor dengan *use case*. *Use case* perangkat lunak *Need For Safety* dapat dilihat pada Gambar 11.



Gambar 11. *Use Case Diagram Need For Safety*

3) Skenario *Use case*

Nama *Use case* : Memulai Aplikasi

Tabel 6. Skenario Memulai Aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Menjalankan aplikasi	
	2. Menuju ke halaman judul aplikasi
3. <i>User</i> menekan tombol "mulai"	
	4. Menuju ke halaman menu utama
	5. Mengaktifkan musik <i>game</i>

Nama *Use case* : *On/Off Musik Game*

Tabel 7. Skenario *On/Off Musik Game*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Masuk menu utama kemudian <i>user</i> memilih tombol <i>Off</i> musik	
	2. Menonaktifkan musik <i>game</i>
3. Saat tombol musik off, <i>user</i> memilih tombol <i>On</i> musik	
	4. Mengaktifkan musik <i>game</i>

Nama *Use case* : Lihat Skor

Tabel 8. Skenario Lihat Skor

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu Skor	
	2. Mengecek data skor pada webstorage 3. Terdapat data skor yang disimpan di webstorage, menampilkan 3 skor terbesar
4. <i>User</i> memilih tombol "hapus"	
	5. Mengosongkan data skor pada webstorage
6. <i>User</i> memilih tombol "home"	
	7. Kembali ke menu utama
Skenario Alternatif	
1. <i>User</i> memilih menu Skor	
	2. Mengecek data skor pada webstorage 3. Belum ada data yang tersimpan di webstorage, menampilkan 3 skor terbesar dengan nilai "nol"

Nama *Use case* : *Get Score Data*

Tabel 9. Skenario *Get Score Data*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> meminta data skor dari webstorage	
	2. Webstorage memberikan data skor atau nilai yang telah tersimpan untuk ditampilkan pada menu skor

Nama *Use case* : Mulai Bermain

Tabel 10. Skenario Mulai Bermain

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "mulai bermain"	
	2. Menuju permainan level pertama
3. <i>User</i> memainkan permainan level 1	
	4. Objek kotak habis 5. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 1 6. Menuju level kedua
7. <i>User</i> memilih tombol "home" di level 1	
	8. Menyimpan skor level pertama 9. Kembali ke menu utama
10. <i>User</i> memainkan permainan level 2	
	11. Objek kotak habis 12. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 2 13. Menuju level ketiga
14. <i>User</i> memilih tombol "home" di level 2	
	15. Menyimpan skor level pertama + kedua 16. Kembali ke menu utama
17. <i>User</i> memainkan permainan level 3	
	18. Objek kotak habis 19. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 3 20. Menuju level keempat
21. <i>User</i> memilih tombol "home" di level 3	
	22. Menyimpan skor level pertama + kedua + ketiga 23. Kembali ke menu utama
24. <i>User</i> memainkan permainan level 4	
	25. Objek kotak habis 26. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 4 27. Menuju ke halaman informasi nilai 28. Menampilkan total langkah dan skor 29. Menyimpan skor dalam variabel "gskor" 30. Menyimpan nilai variabel "gskor" dalam webstorage dengan key "dbHiskor"

Tabel 11. Skenario Mulai Bermain (lanjutan)

Aksi Aktor	Reaksi Sistem
Skenario Normal	
31. <i>User</i> memilih tombol "home" di level 4	
	32. Menyimpan skor level pertama + kedua + ketiga + keempat 33. Kembali ke menu utama
34. <i>User</i> memilih tombol "skor" di halaman informasi nilai	
	35. Menuju halaman tampil skor
36. <i>User</i> memilih tombol "back" di halaman informasi nilai	
	37. Menuju permainan level pertama
38. <i>User</i> memilih tombol "home" di halaman informasi nilai	
	39. Menuju menu utama

Nama *Use case* : Ayo Belajar Rambu

Tabel 12. Skenario Ayo Belajar Rambu

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "ayo belajar"	
	2. Menampilkan submenu rambu
3. <i>User</i> memilih submenu "pengertian rambu"	
	4. Menampilkan halaman pengertian rambu
5. <i>User</i> memilih submenu "rambu peringatan"	
	6. Menampilkan halaman pengertian rambu peringatan
7. <i>User</i> memilih submenu "rambu larangan"	
	8. Menampilkan halaman pengertian rambu larangan
9. <i>User</i> memilih submenu "rambu perintah"	
	10. Menampilkan halaman pengertian rambu perintah
11. <i>User</i> memilih submenu "rambu petunjuk"	
	12. Menampilkan halaman pengertian rambu petunjuk
13. <i>User</i> memilih tombol "home"	
	14. Menuju menu utama

Nama *Use case* : Pengertian Rambu

Tabel 13. Skenario Pengertian Rambu

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "pengertian rambu"	
	2. Menampilkan halaman definisi rambu lalu lintas
3. <i>User</i> memilih tombol "down"	
	4. Menampilkan halaman macam rambu lalu lintas
5. <i>User</i> memilih tombol "close"	
	6. Menuju halaman "ayo belajar"

Nama *Use case* : Rambu Peringatan

Tabel 14. Skenario Rambu Peringatan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "rambu peringatan"	
	2. Menampilkan halaman definisi rambu peringatan
3. <i>User</i> memilih tombol "right"	
	4. Menampilkan halaman contoh-contoh rambu peringatan
5. <i>User</i> memilih tombol "close"	
	6. Menuju halaman "ayo belajar"

Nama *Use case* : Rambu Larangan

Tabel 15. Skenario Rambu Larangan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "rambu larangan"	
	2. Menampilkan halaman definisi rambu larangan
3. <i>User</i> memilih tombol "right"	
	4. Menampilkan halaman contoh-contoh rambu larangan
5. <i>User</i> memilih tombol "close"	
	6. Menuju halaman "ayo belajar"

Nama *Use case* : Rambu Perintah

Tabel 16. Skenario Rambu Perintah

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "rambu perintah"	
	2. Menampilkan halaman definisi rambu perintah
3. <i>User</i> memilih tombol "right"	
	4. Menampilkan halaman contoh-contoh rambu perintah
5. <i>User</i> memilih tombol "close"	
	6. Menuju halaman "ayo belajar"

Nama *Use case* : Rambu Petunjuk

Tabel 17. Skenario Rambu Petunjuk

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "rambu petunjuk"	
	2. Menampilkan halaman definisi rambu petunjuk
3. <i>User</i> memilih tombol "right"	
	4. Menampilkan halaman contoh-contoh rambu petunjuk
5. <i>User</i> memilih tombol "close"	
	6. Menuju halaman "ayo belajar"

Nama *Use case* : Bantuan

Tabel 18. Skenario Bantuan

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>User</i> memilih menu "bantuan"	
	2. Menampilkan informasi pengertian aplikasi " <i>Need For safety</i> "
3. <i>User</i> memilih tombol "down"	
	4. Menampilkan informasi cara memainkan permainan
5. <i>User</i> memilih tombol "down" kedua	
	6. Menampilkan informasi petunjuk penggunaan menu dan tombol
7. <i>User</i> memilih tombol "up"	
	8. Menampilkan informasi pengertian aplikasi " <i>Need For safety</i> "
9. <i>User</i> memilih tombol "home"	
	10. Menuju halaman utama

Nama *Use case* : Keluar Aplikasi

Tabel 19. Skenario Keluar Aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. User keluar dari aplikasi	
	2. Menghentikan fungsi yang sedang berjalan

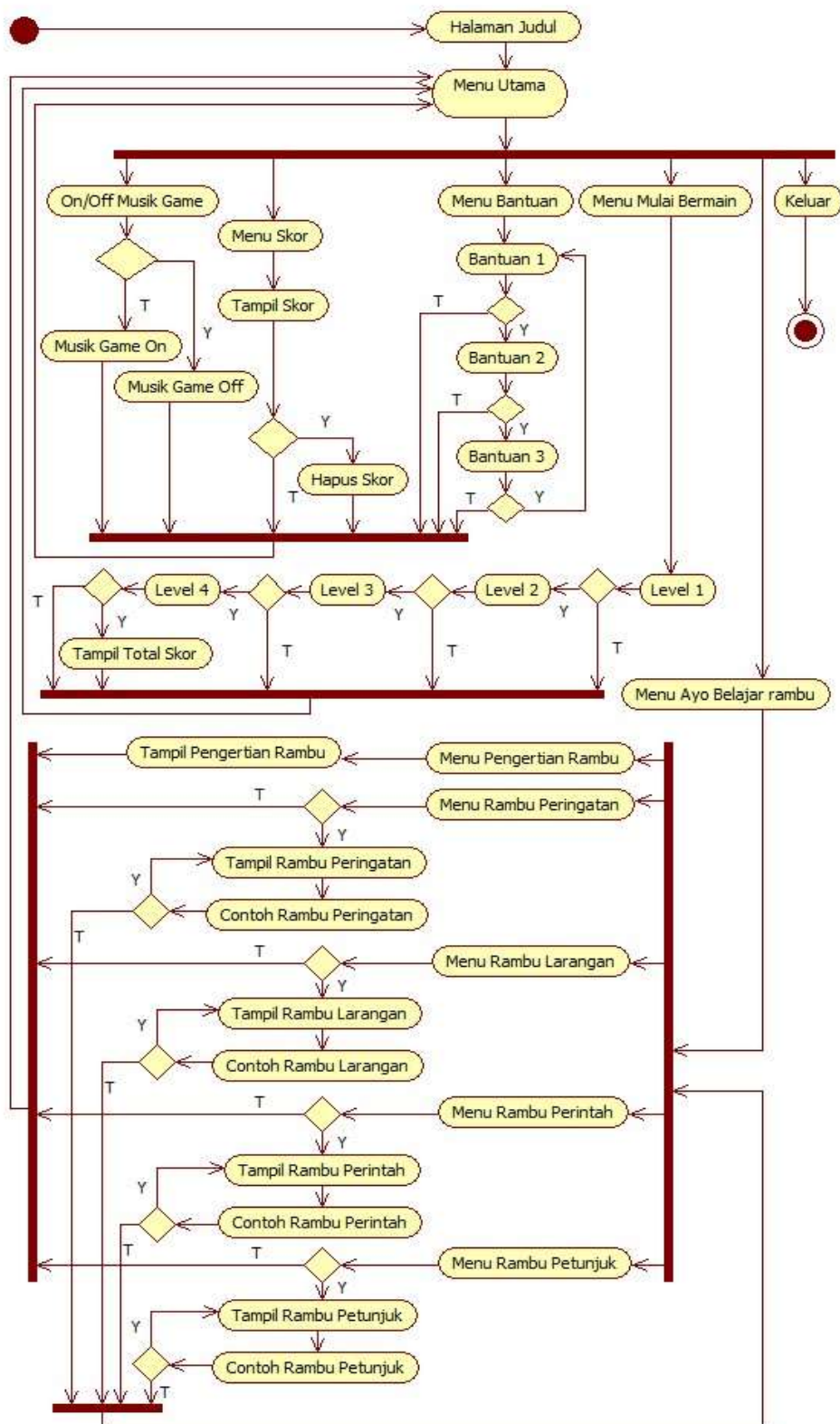
b. *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktivitas secara umum dalam sistem yang dikembangkan. Diagram aktivitas (*activity diagram*) yang disediakan oleh UML melengkapi *use case* yang telah dibuat sebelumnya dengan memberikan representasi grafis dari aliran-aliran interaksi di dalam suatu skenario yang sifatnya spesifik (Pressman, 2010:195). Aktivitas yang terjadi antara lain adalah aktivitas masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana aktivitas ini akan berakhir. Diagram aktivitas aplikasi *Need For Safety* dapat dilihat pada Gambar 12.

Aliran aktivitas berawal dari halaman judul kemudian masuk menu utama yang terdiri dari beberapa menu atau fitur aplikasi seperti menu *mulai bermain*, *ayo belajar*, *bantuan*, *on/off musik game*, dan menu *lihat skor*. Aktivitas sistem pada menu *on/off* musik memiliki *decision* atau percabangan yaitu musik *on* dan musik *off*, kemudian aliran aktivitas sistem kembali ke menu utama. Aktivitas sistem pada menu skor adalah menampilkan skor kemudian terdapat *decision* yaitu aktivitas hapus data dan kembali ke menu utama. Aktivitas sistem pada menu bantuan terdiri dari beberapa *decision* yaitu aktivitas untuk menampilkan bantuan 2 dan kembali ke menu utama, kemudian dilanjutkan *decision* untuk

menampilkan bantuan 3 dan kembali ke menu utama, dan *decision* selanjutnya adalah aktivitas menampilkan bantuan 1 dan kembali ke menu utama.

Aktivitas sistem pada menu mulai bermain adalah menampilkan permainan level 1 kemudian dilanjutkan hingga level 4 dimana pada masing-masing level terdapat percabangan aktivitas yaitu untuk kembali ke menu utama. Aliran aktivitas sistem setelah selesai permainan level 4 adalah menampilkan informasi nilai. Aktivitas sistem pada menu ayo belajar dimulai dengan menampilkan lima submenu ayo belajar yaitu pengertian rambu, rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk. Aliran aktivitas submenu pengertian rambu adalah menampilkan definisi rambu kemudian kembali ke menu utama. Aliran aktivitas submenu rambu peringatan dimulai dengan *decision* menampilkan definisi rambu peringatan dan kembali ke menu utama. Aktivitas sistem setelah tampil definisi rambu peringatan adalah menampilkan contoh rambu peringatan, kemudian dilanjutkan *decision* untuk kembali ke halaman definisi rambu peringatan atau menuju submenu ayo belajar. Rambu larangan, rambu perintah, dan rambu petunjuk memiliki aliran aktivitas sama seperti rambu peringatan. Aliran aktivitas sistem digambarkan berakhir ketika *user* keluar dari aplikasi.



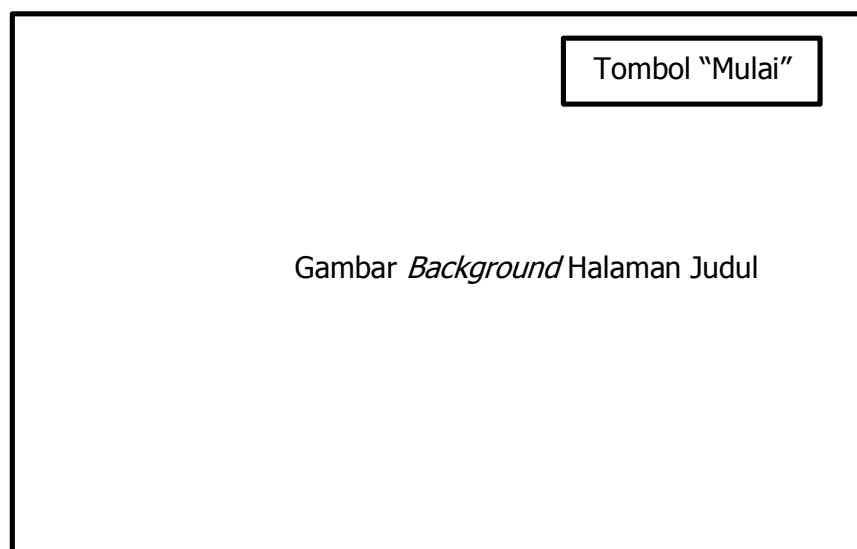
Gambar 12. Activity Diagram *Game Edukasi "Need For Safety"*

c. Perancangan Antarmuka Sistem (*User Interface*)

Perancangan antarmuka sistem merupakan salah satu bagian terpenting dalam pengembangan perangkat lunak. Perancangan antarmuka sistem yang baik akan menentukan apakah perangkat lunak nantinya akan mudah digunakan oleh *user* atau tidak. Perancangan antarmuka atau *user interface* perangkat lunak "*Need For Safety*" sebagai media pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun yang dibuat antara lain sebagai berikut:

1) Halaman Judul Perangkat Lunak

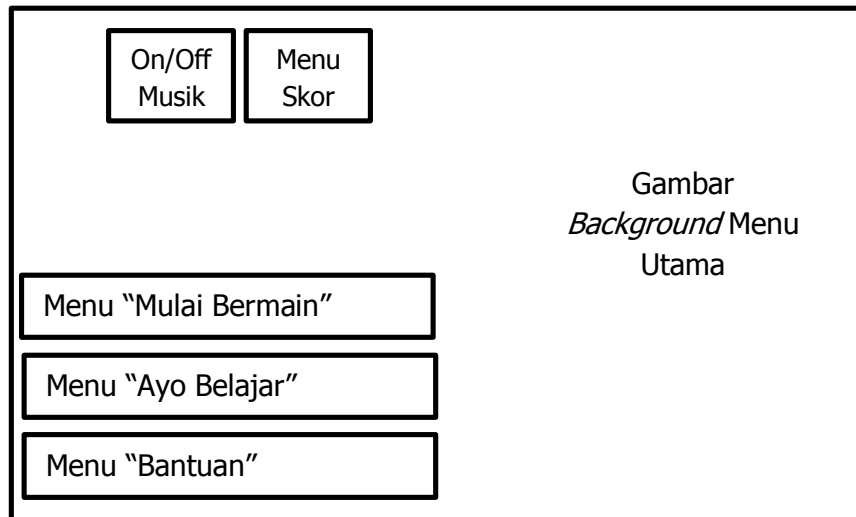
Halaman judul merupakan tampilan awal aplikasi yang berisi gambar *background* dan tombol "mulai" untuk menuju ke halaman menu utama.



Gambar 13. Rancangan Halaman Judul Perangkat Lunak

2) Halaman Menu Utama Aplikasi

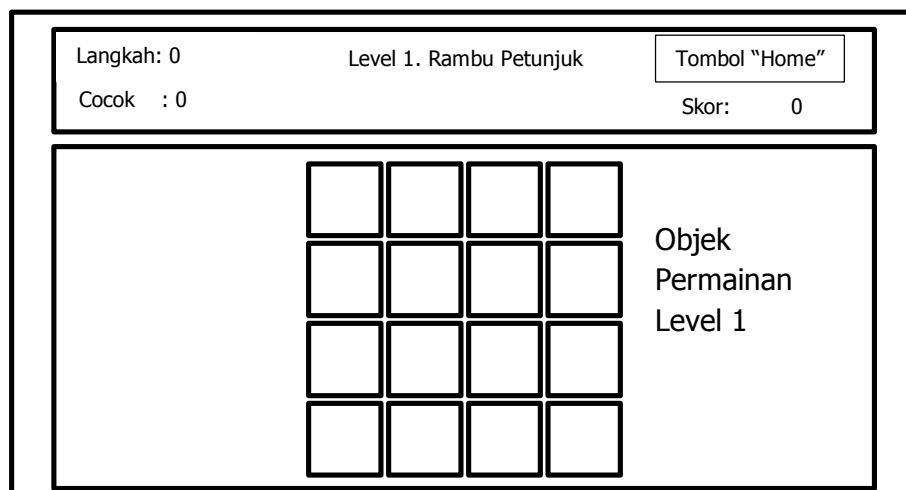
Menu utama akan muncul setelah *user* menekan tombol “masuk” di halaman judul.



Gambar 14. Rancangan Halaman Menu Utama.

3) Halaman Mulai Permainan

Menu permainan akan menampilkan empat level permainan dengan rancangan seperti gambar 15.



Gambar 15. Rancangan Halaman Permainan Level 1

Langkah: 0	Level 2. Rambu Perintah	<div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">Tombol "Home"</div>
Cocok : 0		Skor: 0

Objek Permainan Level 2

Gambar 16. Rancangan Halaman Permainan Level 2

Langkah: 0	Level 3. Rambu Larangan	<div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">Tombol "Home"</div>
Cocok : 0		Skor: 0

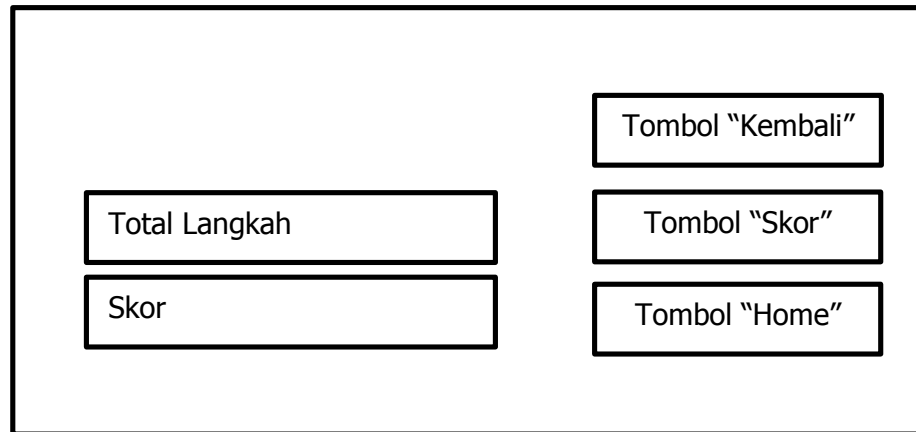
Objek Permainan Level 3

Gambar 17. Rancangan Halaman Permainan Level 3

Langkah: 0	Level 4. Rambu Peringatan	<div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">Tombol "Home"</div>
Cocok : 0		Skor: 0

Objek Permainan Level 4

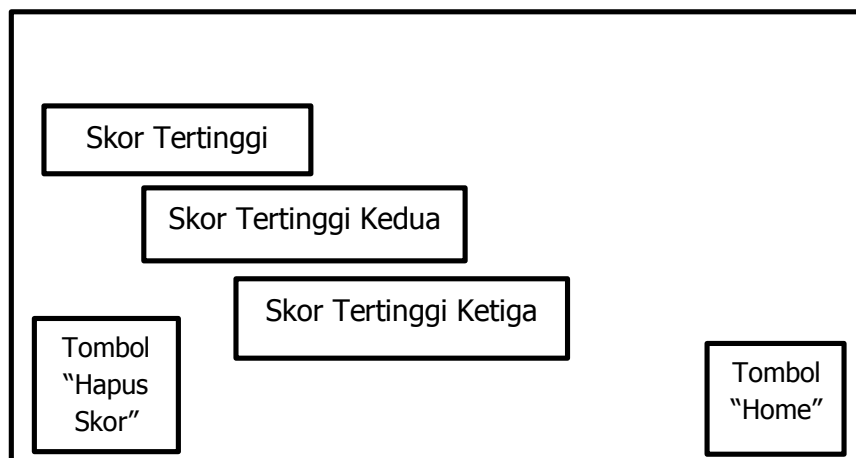
Gambar 18. Rancangan Halaman Permainan Level 4



Gambar 19. Rancangan Halaman Informasi Nilai

4) Halaman Skor

Halaman skor akan menampilkan tiga nilai terbesar yang didapat *user* setelah memainkan permainan. Skor dapat dihapus dengan menggunakan tombol "Hapus".



Gambar 20. Rancangan Halaman Skor.

5) Halaman Submenu Ayo Belajar

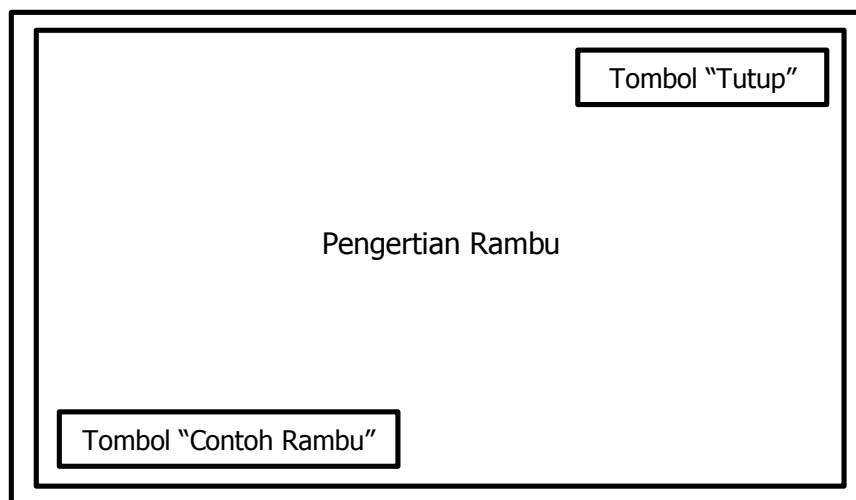
Menu "Ayo Belajar" terdiri dari lima submenu yang masing-masing berisi materi belajar tentang rambu lalu lintas.



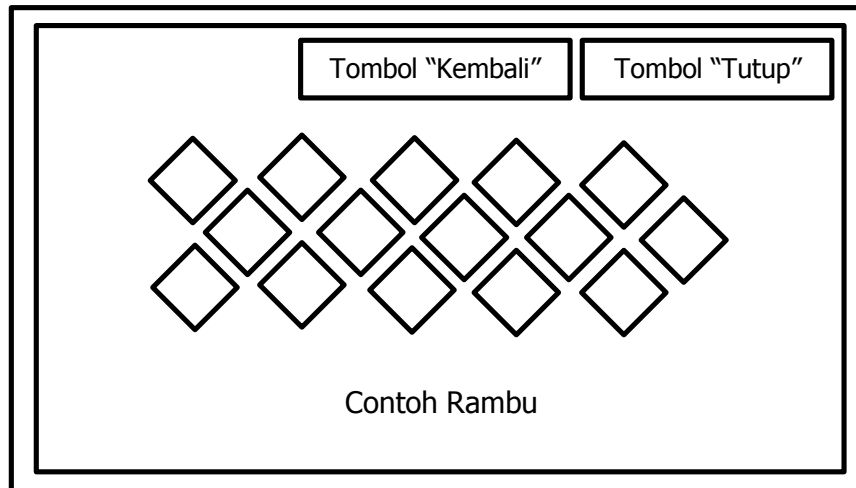
Gambar 21. Rancangan Tampilan Submenu Ayo Belajar.

6) Halaman Pengertian dan Contoh Rambu Lalu Lintas

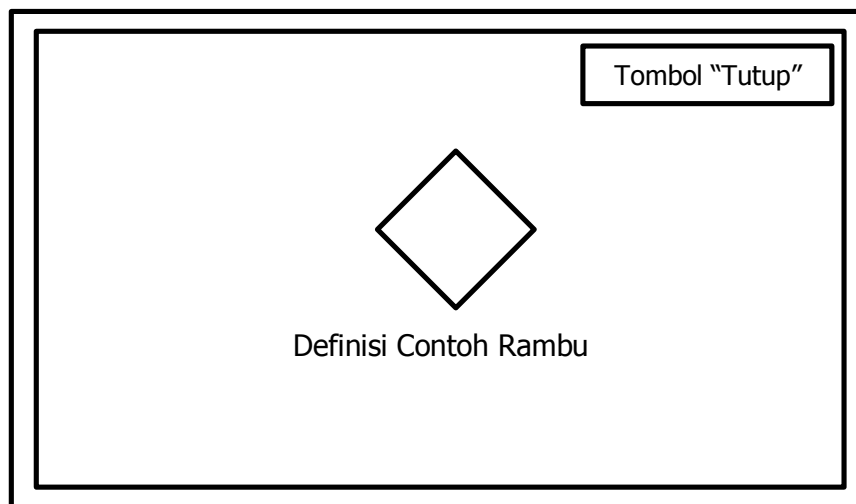
Halaman pengertian rambu berisi definisi dan contoh dari macam-macam rambu. Tombol "Contoh Rambu" adalah untuk menuju ke halaman contoh gambar rambu yang isinya adalah beberapa gambar rambu dan pengertiannya.



Gambar 22. Rancangan Halaman Pengertian Rambu.



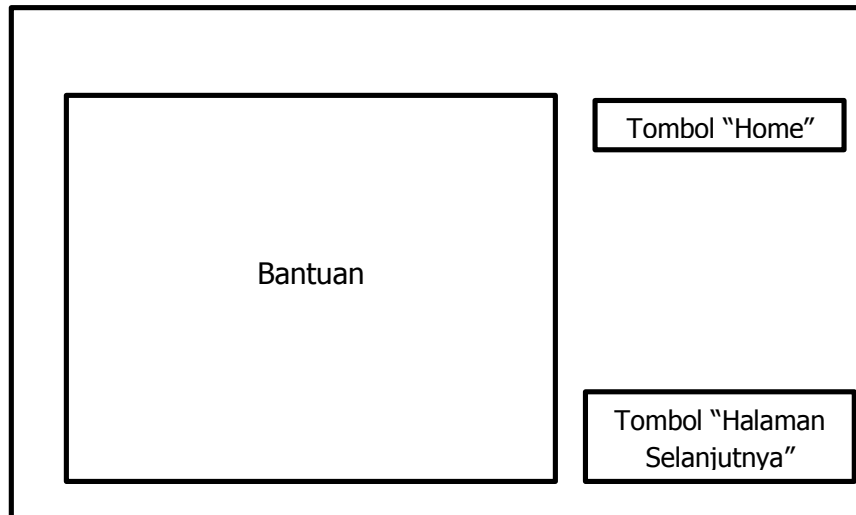
Gambar 23. Rancangan Halaman Contoh Rambu.



Gambar 24. Rancangan Halaman Definisi Contoh Rambu.

7) Halaman Bantuan

Halaman bantuan berisi pengertian aplikasi, cara bermain dan pengertian dari masing-masing menu dan tombol yang ada.



Gambar 25. Rancangan Halaman Bantuan.

3. Implementasi

Rancangan desain antarmuka yang telah dibuat pada tahap sebelumnya diimplementasikan pada tahap ini. Implementasi pengkodean dilakukan berdasarkan rancangan yang telah dibuat dan spesifikasi yang sudah ditetapkan sebelumnya pada tahap analisis kebutuhan. Tahap implementasi ini meliputi implementasi perancangan antarmuka dan pengkodean, dan proses *deployment* perangkat lunak. Tahap awal implementasi, pengembang membuat bagian per bagian sistem sesuai dengan rancangan sebelumnya agar menjadi satu kesatuan sistem yang utuh. Pengembang juga melakukan uji *white box* secara langsung pada tahap ini yang berfokus pada struktur kontrol program.

4. Pengujian

Tahap pengujian ini dilakukan dengan tujuan untuk mengetahui tingkat kelayakan perangkat lunak yang telah dikembangkan. Tingkat kelayakan

perangkat lunak yang dikembangkan ditentukan dengan menguji kualitas perangkat lunak tersebut. Kualitas perancangan merujuk pada karakteristik-karakteristik produk yang dispesifikasi oleh para perancang (Pressman, 2010:484).

Tahap pengujian perangkat lunak yang dikembangkan meliputi pengujian dari sisi pengembang dan juga pengujian dari ahli yang dipilih serta responden untuk menguji kelayakan perangkat lunak nantinya. Pengujian perangkat lunak yang dilakukan antara lain:

a. Pengujian Kotak Putih (*White Box Testing*)

White box testing disebut juga dengan pengujian kotak kaca (*glass box testing*) yaitu sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case* (Pressman, 2010:588). Pengujian *white box* dilakukan menggunakan teknik pengujian jalur dasar (*basic path testing*). *Basic path testing* merupakan teknik pengujian kotak putih, yang pertama kali diajukan oleh Tom McCabe (Pressman, 2010:588). Teknik pengujian ini memungkinkan perancang *test case* untuk menurunkan ukuran kompleksitas logis dari suatu rancangan prosedural dan menggunakan ukuran ini sebagai pedoman untuk menentukan rangkaian dasar jalur eksekusi. Pengujian *white box* dilakukan dengan langkah sebagai berikut:

1) Menggambarkan alur logika ke dalam grafik alir (*flow graph*)

Grafik alir menggambarkan arus kontrol logis dengan menggambarkan notasi yang terdiri dari gambar lingkaran dan panah. Lingkaran atau dinamakan sebagai *node* adalah untuk menyatakan *statement* prosedural pada *source code*.

Sedangkan gambar panah atau disebut juga sebagai *edge* digunakan untuk menyatakan aliran kendali atau alur perjalanan logika.

2) Menentukan *Cyclomatic complexity* dan basis set.

Cyclomatic complexity merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program yang akan diuji. *Cyclomatic complexity* dihitung dengan salah satu dari tiga cara berikut (Pressman, 2010:589):

- a) Jumlah daerah-daerah (*region*) grafik alir yang berhubungan dengan kompleksitas siklomatik.
- b) Kompleksitas siklomatik $V(G)$ untuk grafik alir G didefinisikan sebagai:

$$V(G) = E - N + 2$$

Dimana E adalah jumlah edge grafik alir dan N adalah jumlah node grafik alir.

- c) Kompleksitas siklomatik $V(G)$ untuk grafik aliran G juga didefinisikan sebagai:

$$V(G) = P + 1$$

Dimana P adalah jumlah node predikat yang terdapat dalam grafik alir G .

3) Menentukan basis set dari jalur independen linier.

Penggambaran *flow graph* akan menghasilkan jalur-jalur independen. Jalur independen adalah setiap jalur yang melalui program yang memperkenalkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru (Pressman, 2010:588). Jalur independen minimal melewati sebuah *edge* baru dengan alur yang belum pernah dilalui. Selanjutnya kumpulan dari jalur-jalur independen dari suatu grafik alir akan disebut dengan basis set.

4) Menyiapkan *test case* untuk setiap jalur di basis set.

Langkah terakhir adalah pengujian menggunakan metode *basis path testing*. Pengujian ini menguji semua jalur yang telah didapat setelah penghitungan kompleksitas siklomatik. Pengujian ini dilakukan untuk mengeksekusi semua alur logika yang telah dibuat. Setiap *test case* dieksekusi dan dibandingkan dengan hasil yang diharapkan. Setelah *test case* selesai, penguji dapat yakin bahwa semua pernyataan dalam program ini telah dilaksanakan setidaknya sekali.

b. Pengujian Kotak hitam (*Black Box Testing*)

Pengujian *black box* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak. Pengujian *black box* memungkinkan pengembang untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program (Pressman, 2010:596). Pengujian *black box* dilakukan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran perangkat lunak sudah sesuai dengan spesifikasi yang diperlukan.

Pengujian *black box* dalam penelitian ini dilakukan dengan menguji fungsionalitas perangkat lunak pada desain *use case*. Skenario *use case* yang telah dibuat akan digunakan sebagai acuan untuk melakukan uji *black box*. Masukan dan keluaran dari perangkat lunak pada saat pengujian akan dibandingkan dengan hasil pengujian apakah sudah sesuai dengan fungsionalitas masing-masing atau tidak.

c. Pengujian *Alpha*

Pengujian *alpha* dilakukan disisi pengembang oleh sekelompok perwakilan dari pengguna akhir (Pressman, 2010:570). Pengujian *alpha* dalam penelitian ini

menggunakan kuesioner. Kuesioner diberikan kepada ahli dalam bidang rekayasa perangkat lunak yang dipilih untuk melakukan pengujian *alpha* terhadap perangkat lunak. Data hasil dari pengujian *alpha* nantinya akan dianalisis dan digunakan untuk memperbaiki perangkat lunak sesuai dengan saran yang diberikan oleh ahli. Tindak lanjut dari saran-saran tersebut sebagai revisi awal untuk penyempurnaan perangkat lunak yang dikembangkan.

d. Pengujian *Beta*

Pengujian *beta* adalah aplikasi “hidup” dari perangkat lunak dalam sebuah lingkungan yang tidak dapat dikendalikan oleh pengembang (Pressman, 2010:570). Pengujian *beta* dilakukan diluar lingkungan yang tidak dapat dikendalikan oleh pengembang. Pengujian *beta* dalam penelitian ini dilakukan oleh beberapa responden yang dipilih sebagai *user* untuk mencoba fungsionalitas perangkat lunak “*Need For Safety*”. Aspek yang digunakan untuk mengevaluasi kualitas perangkat lunak yang dikembangkan dalam pengujian *beta* meliputi aspek *functionality*, *efficiency*, *usability*, dan *portability*. Aspek tersebut mengacu pada faktor-faktor kualitas perangkat lunak menurut *quality factors* ISO 9126. Hasil dari penelitian ini akan diketahui kelayakan dari perangkat lunak “*Need For Safety*”.

G. Teknik Pengumpulan Data

Teknik pengumpulan data merupakan teknik yang digunakan peneliti untuk mengumpulkan data-data terkait penelitiannya. Teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data (Sugiyono, 2013:224). Pengumpulan data dapat dilakukan dengan berbagai *setting*, berbagai sumber, dan berbagai

cara. Teknik yang digunakan dalam mengumpulkan data dipilih berdasarkan pada berbagai faktor terutama jenis data dan ciri respondennya. Teknik pengumpulan data antara lain adalah sebagai berikut:

1. Wawancara

Wawancara digunakan dalam teknik pengumpulan data apabila peneliti ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan juga apabila peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah responden yang sedikit atau kecil (Sugiyono, 2013:137). Anggapan yang perlu dipegang oleh peneliti sebagai orang yang melakukan wawancara dalam menggunakan metode wawancara ini adalah sebagai berikut:

- a. Responden adalah orang yang paling tahu tentang dirinya sendiri.
- b. Pernyataan yang dinyatakan oleh responden kepada peneliti adalah benar dan dapat dipercaya.
- c. Interpretasi responden tentang pertanyaan yang diajukan peneliti kepadanya adalah sama dengan apa yang dimaksudkan oleh peneliti.

Wawancara dapat dilakukan secara terstruktur maupun tidak terstruktur, dan juga dapat dilakukan secara langsung kepada subjek dan menanyakan pertanyaan wawancara atau tidak secara langsung.

a. Wawancara Terstruktur

Wawancara terstruktur merupakan jenis wawancara bila peneliti telah mengetahui dengan pasti tentang informasi apa yang akan diperoleh dari hasil wawancara dengan subjeknya. Dalam wawancara ini, peneliti sebelum melakukan wawancara telah menyiapkan daftar pertanyaan dan juga alternatif

jawabannya. Pertanyaan untuk wawancara ini dapat berupa daftar pertanyaan yang dibawahnya diberikan alternatif jawaban seperti pada pertanyaan pilihan ganda.

b. Wawancara Tidak Terstruktur

Wawancara tidak terstruktur merupakan wawancara yang bebas dimana peneliti tidak menggunakan pedoman wawancara yang telah tersusun secara sistematis dan lengkap untuk pengumpulan datanya. Pedoman yang digunakan pewawancara adalah berupa garis besar permasalahannya saja. Wawancara jenis tidak terstruktur digunakan untuk mendapatkan informasi yang lebih mendalam kepada para respondennya.

Pengertian wawancara dari uraian di atas dapat disimpulkan bahwa wawancara merupakan teknik pengumpulan data dengan cara bertatap muka langsung dengan responden dan menanyakan langsung pertanyaan-pertanyaan yang sudah dipersiapkan atau sesuai dengan garis besar permasalahannya yang akan diteliti kepada responden untuk mendapatkan informasi yang lebih mendalam.

2. Angket

Angket merupakan suatu alat pengumpul informasi dengan cara menyampaikan sejumlah pertanyaan tertulis untuk menjawab secara tertulis pula oleh respondennya (Margono, 1997:167). Angket merupakan teknik pengumpulan data yang efisien bila peneliti mengetahui dengan pasti variabel yang diukur dan mengetahui apa yang dapat diharapkan dari responden. Angket dapat digunakan untuk responden yang cukup besar dan tersebar di wilayah

yang luas. Prinsip-prinsip dalam penyusunan angket menurut Sugiyono (2013:142) adalah sebagai berikut:

a. Isi dan Tujuan Pertanyaan

Isi angket pengukuran harus berupa pertanyaan yang dibuat secara teliti, menggunakan skala pengukuran dan jumlah item yang mencukupi untuk mengukur variabel yang diteliti.

b. Bahasa yang Digunakan

Penggunaan bahasa pada angket harus disesuaikan dengan kemampuan bahasa responden. Usia dan kemampuan berbahasa responden menjadi pedoman dalam penyusunan angket.

c. Tipe dan Bentuk Pertanyaan

Tipe pertanyaan dapat berbentuk pertanyaan terbuka atau tertutup. Pertanyaan terbuka merupakan pertanyaan yang mengharapkan responden untuk menuliskan jawabannya dalam bentuk uraian tentang sesuatu hal. Pertanyaan tertutup merupakan pertanyaan yang mengharapkan jawaban singkat atau mengharapkan responden untuk memilih salah satu alternatif jawaban saja.

d. Pertanyaan Tidak Mendua

Pertanyaan yang dibuat harus berbentuk tunggal atau tidak mendua. Pertanyaan mendua menanyakan dua atau lebih pertanyaan dalam suatu pernyataan. Pertanyaan yang mendua akan menyulitkan responden dalam menjawab pertanyaan tersebut.

e. Tidak Menanyakan yang Sudah Lupa

Pertanyaan angket adalah pertanyaan yang sekiranya tidak menanyakan hal-hal yang mungkin responden sudah lupa.

f. Pertanyaan Tidak Menggiring

Pertanyaan dalam angket sebaiknya tidak menggiring ke jawaban yang baik saja atau ke jawaban yang jelek saja.

g. Panjang Pertanyaan

Pertanyaan dalam angket sebaiknya tidak terlalu panjang, sehingga akan membuat jenuh responden dalam mengisi angket tersebut. Bila jumlah variabel yang digunakan banyak, maka angket dibuat variasi dari segi penampilan, skala pengukurannya atau cara menjawab angketnya.

h. Urutan pertanyaan

Urutan pertanyaan pada angket dimulai dari umum ke pertanyaan yang lebih spesifik atau dari pertanyaan yang mudah kemudian pertanyaan yang sulit.

i. Prinsip pengukuran

Angket sebelum digunakan perlu divalidasi dan diuji reliabilitasnya karena angket tujuannya adalah untuk mendapatkan data yang valid dan reliabel.

j. Penampilan fisik angket

Penampilan fisik angket akan mempengaruhi respon atau keseriusan responden dalam mengisi angket.

3. Observasi

Teknik pengumpulan data dengan cara observasi merupakan teknik yang digunakan dengan cara mengamati objek yang diteliti secara langsung dan

kemudian mengambil data yang dibutuhkan. Teknik pengumpulan data dengan observasi digunakan bila penelitian berkenaan dengan perilaku manusia, proses kerja, gejala-gejala alam dan bila responden yang diamati tidak terlalu besar (Sugiyono, 2013:145). Observasi dari segi proses pelaksanaan pengumpulan data dibedakan menjadi observasi berperan serta (*participant observation*) dan observasi tidak berperan serta (*non participant observation*), sedangkan dari segi instrumentasi yang digunakan, observasi dibedakan menjadi observasi terstruktur dan tidak terstruktur.

a. Participant Observation

Observasi ini merupakan jenis observasi dimana peneliti terlibat dengan kegiatan sehari-hari orang yang sedang diamati atau yang digunakan sebagai sumber data penelitian. Data yang didapatkan dari observasi ini akan bersifat lebih lengkap dan tajam.

b. Non Participant Observation

Non Participant memiliki arti bahwa observasi tidak terlibat langsung dengan responden dan hanya sebagai pengamat independen saja. Peran peneliti disini adalah untuk mencatat setiap kegiatan yang diamati dan menyimpulkan hasilnya. Observasi jenis ini tidak sampai mendapatkan data yang mendalam dan tidak sampai pada tingkat makna.

c. Observasi Terstruktur

Observasi terstruktur merupakan observasi yang telah dirancang secara sistematis tentang apa yang akan diamati, kapan dan dimana tempatnya. Observasi ini digunakan jika peneliti telah mengetahui dengan pasti tentang variabel yang akan diteliti.

d. Observasi Tidak Terstruktur

Observasi tidak terstruktur adalah observasi yang tidak dipersiapkan secara sistematis tentang apa yang akan diobservasi. Observasi ini digunakan karena peneliti tidak mengetahui secara pasti apa yang diteliti.

4. Tes

Tes merupakan seperangkat rangsangan yang diberikan kepada seseorang dengan maksud untuk mendapat jawaban yang dapat dijadikan dasar bagi penetapan skor angka (Margono, 1997:170). Penggunaan tes sebagai teknik pengumpulan data harus bersifat valid dan reliabel. Tes yang bersifat valid adalah tes yang benar-benar dapat mengungkap aspek yang diselidiki secara tepat dan bersifat reliabel apabila tes dapat memberikan hasil yang relatif tetap apabila dilakukan secara berulang pada kelompok individu yang sama. Tes yang digunakan dalam pengumpulan data terdiri dari tes lisan dan tes tertulis.

- a. Tes lisan merupakan tes yang terdiri dari sejumlah pertanyaan yang diajukan secara lisan tentang aspek-aspek yang ingin diketahui keadaanya dengan jawaban yang diberikan juga secara lisan.
- b. Tes tertulis merupakan tes yang terdiri dari sejumlah pertanyaan yang diajukan secara tertulis dengan jawaban juga secara tertulis. Tes tertulis terdiri dari *essay test* dan tes objektif. *Essay test* merupakan tes yang menghendaki agar *tester* memberikan jawaban dalam bentuk uraian atau kalimat yang disusun sendiri, sedangkan tes objektif merupakan tes yang disusun dimana setiap pertanyaan tes disediakan alternatif jawaban yang dapat dipilih. Bentuk tes objektif antara lain adalah tes betul-salah, tes pilihan ganda, tes menjodohkan dan tes melengkapi jawaban.

Tes selain bersifat valid dan reliabel juga harus memiliki sifat objektif, diagnostik dan efisien. Tes bersifat objektif apabila dalam memberikan nilai kuantitatif terhadap jawaban, unsur subjektivitas penilai tidak ikut mempengaruhi. Tes bersifat diagnostik memiliki arti bahwa tes harus memiliki daya pembeda dalam arti dapat digunakan untuk individu yang memiliki kemampuan yang tinggi sampai kemampuan yang rendah berdasarkan aspek yang diteliti. Unsur yang harus dipenuhi dalam pembuatan tes yang terakhir adalah tes harus bersifat efisien yaitu mudah cara membuat dan mudah cara penilaiannya.

Pengertian tes dari uraian di atas dapat disimpulkan bahwa tes merupakan teknik pengumpulan data yang harus memiliki sifat valid, reliabel, objektif, diagnostik dan efisien untuk digunakan dalam mendapatkan jawaban dari responden yang dapat dijadikan dasar bagi penetapan skor angka.

5. Survei

Survei merupakan metode pengumpulan data dengan menggunakan instrumen untuk meminta tanggapan dari responden tentang sampel (Gulo, 2002:118). Survei merupakan jenis teknik pengumpulan data yang terdiri atas wawancara dan kuesioner. Wawancara digunakan dalam bentuk tatap muka dengan responden dan menanyakan pertanyaan-pertanyaan dengan menggunakan instrumen yang disebut dengan *schedule*. Kuesioner digunakan dalam bentuk tertulis yang diberikan kepada responden untuk meminta tanggapannya. Teknik pengumpulan data survei memiliki ciri-ciri sebagai berikut:

- a. Survei digunakan pada sampel yang mewakili populasi.
- b. Respon didapatkan secara langsung dari responden.

- c. Penggunaan survei melibatkan banyak responden dan mencakup area yang lebih luas di bandingkan dengan metode lainnya.
- d. Survei dilakukan dalam situasi yang alamiah.

Penggunaan wawancara dan kuesioner dalam teknik pengumpulan survei memiliki keuntungan masing-masing. Keuntungan penggunaan wawancara terletak pada fleksibilitasnya dan tingkat ketergantungan pada responden sedangkan keuntungan penggunaan kuesioner adalah pada keakuratan dan biaya pengeluaran yang rendah.

6. Teknik Dokumenter

Teknik pengumpulan data dokumenter adalah cara mengumpulkan data melalui peninggalan tertulis, seperti arsip-arsip dan termasuk buku-buku tentang pendapat, teori, dalil atau hukum-hukum, dan lain-lain yang berhubungan dengan masalah penelitian (Margono, 1997:181). Pengumpulan data dengan teknik dokumentasi dilakukan peneliti dengan menelusuri berbagai macam dokumen antara lain buku, majalah, koran, notulen rapat, peraturan-peraturan dan sumber informasi lain (Sandjaja dan Albertus, 2006:146).

Teknik dokumenter dapat digunakan dalam penelitian kuantitatif dan kualitatif. Penelitian kuantitatif menggunakan teknik dokumenter berfungsi untuk menghimpun secara efektif bahan-bahan yang dipergunakan di dalam kerangka atau landasan teori, penyusunan hipotesis secara mendalam. Penelitian kualitatif menggunakan teknik dokumenter berfungsi untuk pembuktian hipotesis yang diajukan secara logis dan rasional melalui pendapat, teori atau hukum yang diterima, baik yang mendukung maupun yang menolok hipotesis tersebut.

7. Teknik Eksperimental

Teknik pengumpulan data eksperimental adalah suatu metode yang dipakai untuk mengetahui pengaruh dari suatu media, alat, atau kondisi, yang sengaja diadakan terhadap suatu gejala sosial berupa kegiatan dan tingkah laku seseorang ataupun kelompok individu (Burhan, 2009:146). Pemakaian teknik eksperimental membutuhkan desain eksperimen dengan tujuan untuk mendapatkan bentuk eksperimen yang diinginkan sesuai dengan kondisi lingkungan yang ada sehingga didapatkan bentuk eksperimen yang tepat. Teknik eksperimental merupakan teknik yang praktis untuk mendapatkan data tentang pengaruh-pengaruh atau efektivitas tertentu, tetapi dalam penelitiannya membutuhkan ketelitian dan waktu yang cukup lama.

8. Nominal Group Technique (NGT)

Nominal Group Technique (NGT) adalah metode pengumpulan data yang dilakukan melalui diskusi untuk memperoleh konsensus dari kelompok yang selanjutnya dipergunakan oleh pengambil keputusan sebagai pedoman pengambilan keputusan (Sandjaja dan Albertus, 2006:166). Metode pengumpulan data ini hampir memiliki kesamaan dengan diskusi terpusat untuk mendapatkan suatu hasil diskusi yang sesuai dengan tujuan awal diskusi. Metode ini termasuk dalam metode pengumpulan data yang jarang digunakan oleh para peneliti. Diskusi yang dilakukan dalam teknik ini biasanya beranggotakan delapan hingga sepuluh orang yang mengerti tentang masalah yang dibicarakan atau diteliti. Penataan ruang dalam diskusi teknik ini juga penting untuk dilakukan. Ruangan dibuat dengan pengaturan tempat duduk membentuk huruf "U" dengan tujuan agar setiap peserta dapat bertatap muka satu dengan yang lainnya ketika

mereka saling mengutarakan pendapat. Keuntungan dari metode ini adalah hasil dari diskusi akan menggambarkan masukan dari semua peserta diskusi.

9. Delphi Technique

Teknik *delphi* merupakan teknik pengumpulan data yang hampir sama dengan teknik *Nominal Group Technique* (NGT). Perbedaannya dengan teknik *Nominal Group Technique* adalah bahwa pada teknik *delphi*, tidak ada diskusi melainkan semua kegiatan dilakukan secara tertulis melalui kuesioner yang diedarkan setiap kali kepada peserta (Sandjaja dan Albertus, 2006:168). Perbedaan teknik *delphi* dengan *Nominal Group Technique* selanjutnya adalah pada waktu pelaksanaan diskusi. Teknik *delphi* membutuhkan waktu yang lebih lama dibanding dengan teknik *Nominal Group Technique* karena pemberian kuesioner dan pengisian kuesioner oleh peserta membutuhkan waktu yang lama.

Tujuan dari pengumpulan data menggunakan teknik *delphi* berdasarkan uraian diatas dapat disimpulkan bahwa teknik *delphi* digunakan untuk mengembangkan suatu perkiraan tentang permasalahan yang dihadapi dengan meminta pendapat para ahli, dan pada saat yang sama dapat menyelesaikan permasalahan tersebut. Keuntungan teknik ini adalah dapat memecahkan permasalahan yang sifatnya umum karena pada waktu diskusi menghadirkan ahli-ahli dalam bidang yang sesuai dengan bidang permasalahan yang diungkap.

Penelitian ini menggunakan teknik pengumpulan data observasi dan angket. Teknik observasi digunakan untuk memperoleh data terkait aspek *portability* perangkat lunak "*Need For Safety*", sedangkan teknik pengumpulan data angket digunakan untuk memperoleh data pada pengujian *alpha* dan *beta* terkait aspek *functionality*, *efficiency*, *usability*, dan *portability* perangkat lunak.

H. Skala Pengukuran

Skala pengukuran merupakan kesepakatan yang digunakan sebagai acuan untuk menentukan panjang pendeknya interval yang ada dalam alat ukur, sehingga alat ukur tersebut bila digunakan dalam pengukuran akan menghasilkan data kuantitatif (Sugiyono, 2013:92).

1. Skala Guttman

Skala ini merupakan skala kumulatif yang diperkenalkan pertama kali oleh Louis Guttman. Nilai yang dihasilkan dari skala ini adalah binary skor (0-1), dan digunakan untuk memperoleh jawaban yang tegas dan konsisten seperti misal: Ya-Tidak, Benar-Salah, dan lain-lain. Menurut Sugiyono (2013:96), penelitian menggunakan skala Guttman dilakukan bila ingin mendapatkan jawaban yang tegas terhadap suatu permasalahan yang ditanyakan. Pada penelitian ini, skala Guttman akan digunakan untuk menganalisa data pada proses pengujian *alpha*.

2. Skala Likert

Skala Likert merupakan skala yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono, 2013:93). Skala Likert dapat digunakan untuk mengukur sikap seseorang dengan menyatakan setuju atau tidak setuju terhadap subjek, objek atau kejadian tertentu. Dengan menggunakan skala Likert, variabel dijabarkan menurut urutan variabel, sub variabel, indikator, dan deskriptor. Deskriptor kemudian dijadikan titik tolak untuk membuat butir instrumen berupa pernyataan atau pertanyaan yang perlu dijawab oleh responden. Item-item dalam skala likert menyediakan respon dengan kategori yang berjenjang, dan biasanya memiliki jenjang lima, yaitu: sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat

tidak setuju. Setiap kategori tersebut diberi nilai atau skor. Pernyataan pada skala Likert terdiri dari pernyataan positif dan pernyataan negatif. Contoh lima jenjang dalam skala Likert adalah sebagai berikut:

Tabel 20. Jenjang dalam Skala Likert

Pernyataan Positif		Pernyataan Negatif	
Sangat Baik Sekali (SBS)	5	Sangat Baik Sekali (SBS)	5
Sangat Baik (SB)	4	Sangat Baik (SB)	4
Sedang (S)	3	Sedang (S)	3
Buruk (B)	2	Buruk (B)	2
Sangat Buruk (BS)	1	Sangat Buruk (BS)	1

Total skor tiap responden dalam pengujian menggunakan skala likert dihitung dengan cara menjumlahkan skor-skor item yang diperoleh responden. Prosedur penskalaan likert dikenal dengan nama *Likert's Summated Rating*.

Langkah-langkah dalam menyusun *Likert's Summated Rating* adalah sebagai berikut:

- Menentukan secara tegas sikap terhadap topik apa yang akan diukur.
- Menentukan dengan tegas sub variabel atau dimensi yang menyusun sikap tersebut.
- Menyusun pernyataan yang merupakan alat pengukur dimensi yang menentukan sikap yang akan diukur sesuai dengan indikator.
- Setiap item diberi respon yang bersifat tertutup.
- Untuk setiap respon, jawaban diberi skor berdasarkan kriteria sebagai berikut: apabila item positif maka nilai terbesar diletakkan pada respon "sangat setuju" sedangkan bila item negatif maka nilai terbesar diletakkan respon "sangat tidak setuju".

- f. Menentukan skor maksimal dan minimal yang mungkin dicapai responden untuk mengetahui posisi setiap responden tentang suatu variabel.

Skala Likert dalam pengujian perangkat lunak "*Need For Safety*" digunakan untuk menganalisa data yang dihasilkan dari uji *alpha* dan uji *beta*. Data hasil dari pengujian meliputi aspek *functionality*, *efficiency*, *usability*, dan *portability* akan dianalisa menggunakan skala Likert.

I. Instrumen Penelitian

Instrumen merupakan alat bantu yang digunakan peneliti dalam pengukuran variabel (Zainal Mustafa, 2009:93). Instrumen penelitian yang digunakan untuk memperoleh data pada penelitian ini adalah sebagai berikut:

1. Lembar Observasi

Instrumen penelitian berupa lembar observasi digunakan dalam pengumpulan data dengan teknik observasi yang dilakukan oleh pengamat. Pengamatan dilakukan dengan cara menjalankan perangkat lunak menggunakan aplikasi emulator android dan *handphone* dengan sistem android. Pada lembar observasi ini, pengamat menuliskan hasil pengamatan pada setiap pernyataan dari aspek *portability*. Hasil yang diharapkan dari pengamatan adalah ya atau tidak yang dilengkapi dengan keterangan dari pengamat tentang keberhasilan atau kegagalan saat melakukan pengamatan. Data hasil pengamatan menggunakan lembar observasi kemudian dimasukkan ke dalam tabel keberhasilan uji aspek *portability*.

Tabel 21. Lembar Observasi Uji Portabilitas Perangkat Lunak

Variabel	Indikator	Sub Indikator	Hasil yang Diharapkan	Ketercapaian		Ket.
				Ya	Tidak	
Portability	Installability (Kemampuan untuk Diinstal)	Kemudahan instalasi aplikasi	Aplikasi " <i>Need For Safety</i> " dapat diinstal dengan mudah di <i>platform</i> Android			
		Keberhasilan dalam instalasi	Aplikasi " <i>Need For Safety</i> " berhasil diinstal di <i>platform</i> Android			
	Adaptability (Kemampuan untuk Beradaptasi)	Instalasi di beberapa versi android	Aplikasi " <i>Need For Safety</i> " dapat diinstal di beberapa versi Android yang berbeda			
		Instalasi di beberapa platform android dengan ukuran layar berbeda	Aplikasi " <i>Need For Safety</i> " berhasil dinstal di platform android dengan ukuran layar berbeda			

2. Kuesioner (Angket)

Instrumen lain yang digunakan untuk menguji perangkat lunak yang dikembangkan adalah berupa kuesioner. Instrumen kuesioner digunakan untuk pengujian perangkat lunak dalam uji *alpha* dan uji *beta*.

a. Pengujian *Alpha*

Pengujian *alpha* pada penelitian ini dilakukan dengan menggunakan kuesioner yang diberikan kepada ahli dalam bidang rekayasa perangkat lunak. Kuesioner yang digunakan dalam pengujian *alpha* berupa tabel spesifikasi unjuk kerja aplikasi dan tabel pengujian kelayakan perangkat lunak dari aspek *functionality*, *efficiency*, dan *usability*. Hasil pengujian *alpha* menggunakan tabel spesifikasi akan dianalisa dengan menggunakan skala Guttman sedangkan hasil pengujian menggunakan tabel kelayakan perangkat lunak akan dianalisa dengan

skala Likert. Kuesioner yang digunakan untuk pengujian *alpha* adalah sebagai berikut:

Tabel 22. Spesifikasi Uji Membuka Aplikasi

No	Aktivitas/Menu	Hasil yang diharapkan	Tarf ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Tampil halaman " <i>Title Screen</i> "		
2	<i>User</i> memilih tombol "masuk"	Tampil halaman menu <i>game</i> (menu skor, menu bantuan, menu mulai bermain, menu ayo belajar, dan menu <i>On/Off</i> musik)		

Tabel 23. Spesifikasi Uji Menu *Game*

No	Aktivitas/Menu	Hasil yang diharapkan	Tarf ketercapaian	
			Ya	Tidak
1	Menu <i>Game</i>	Menu <i>game</i> dapat terbuka setelah <i>user</i> menekan tombol "mulai" pada halaman " <i>Title Screen</i> "		
		<i>User</i> dapat masuk ke semua menu dengan memilih tombol menu-menu yang ada		
		<i>User</i> dapat menonaktifkan musik <i>game</i> dengan memilih tombol " <i>off</i> musik" pada menu awal		
		<i>User</i> dapat mengaktifkan musik setelah musik mati dengan memilih tombol " <i>on</i> musik" pada menu awal		
		<i>User</i> dapat kembali ke menu awal dengan memilih tombol "home"		
		<i>User</i> dapat keluar aplikasi dengan memilih tombol "back" pada <i>handphone</i>		

Tabel 24. Spesifikasi Uji Menu Mulai Bermain

No	Aksi/Menu	Hasil yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk Menu Mulai Bermain	<i>User</i> dapat masuk ke menu "mulai bermain"		
		Muncul permainan level 1		
		Setelah <i>user</i> menyelesaikan level 1 langsung menuju level 2		
		Setelah <i>user</i> menyelesaikan level 2 langsung menuju level 3		
		Setelah <i>user</i> menyelesaikan level 3 langsung menuju level 4		
		Setelah <i>user</i> menyelesaikan level 4 langsung muncul informasi nilai		
2	Tampil skor nilai	<i>User</i> dapat menuju halaman skor setelah memilih tombol "skor"		
		<i>User</i> dapat memainkan permainan kembali dari level 1 setelah memilih tombol "left/arrah kiri"		

Tabel 25. Spesifikasi Uji Menu Ayo Belajar

No	Aksi/Menu	Hasil yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Masuk Menu Ayo Belajar	<i>User</i> dapat masuk ke menu "ayo belajar"		
		Tampil submenu ayo belajar		
2	Pengertian Rambu	<i>User</i> dapat masuk ke halaman pengertian rambu setelah memilih submenu pengertian rambu		
3	Rambu Peringatan	<i>User</i> dapat masuk ke halaman pengertian rambu peringatan setelah memilih submenu rambu peringatan		
		Menuju halaman contoh rambu peringatan setelah <i>user</i> memilih tombol "right/arrah kanan"		

Tabel 26. Spesifikasi Uji Menu Ayo Belajar (lanjutan)

No	Aksi/Menu	Hasil yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
		Pada halaman contoh rambu peringatan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut		
5	Rambu Larangan	<i>User</i> dapat masuk ke halaman pengertian rambu larangan setelah memilih submenu rambu larangan		
		Menuju halaman contoh rambu larangan setelah <i>user</i> memilih tombol "right/arrah kanan"		
		Pada halaman contoh rambu larangan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut		
6	Rambu Perintah	<i>User</i> dapat masuk ke halaman pengertian rambu perintah setelah memilih submenu rambu perintah		
		Menuju halaman contoh rambu perintah setelah <i>user</i> memilih tombol "right/arrah kanan"		
		Pada halaman contoh rambu perintah, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut		
7	Rambu Petunjuk	<i>User</i> dapat masuk ke halaman pengertian rambu petunjuk setelah memilih submenu rambu petunjuk		
		Menuju halaman contoh rambu petunjuk setelah <i>user</i> memilih tombol "right/arrah kanan"		
		Pada halaman contoh rambu petunjuk, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut		

Tabel 27. Kisi-Kisi Uji Kelayakan Perangkat Lunak Aspek *Functionality*, *Efficiency*, dan *Usability*

Variabel	Indikator	Sub Indikator	Item Butir
<i>Functionality</i>	<i>Suitability</i> (Kesesuaian)	Menjalankan semua fitur menu dalam <i>game</i>	1,2,3,4
		Semua fitur berjalan sebagai mana mestinya	5
	<i>Interoperability</i> (Interoperabilitas)	Berinteraksi dengan webstorage penyimpanan data	6
<i>Efficiency</i>	<i>Time Behaviour</i> (Perilaku waktu)	Lama waktu operasi tiap aksi	7
		Respon sesuai dengan setiap aksi dari pengguna	8
<i>Usability</i>	<i>Understandability</i> (Kemudahan untuk dipahami)	Fitur-fitur <i>game</i> yang disediakan mudah dipahami	9
		Kejelasan penggunaan setiap fitur	10
		Memberikan informasi yang jelas	11
	<i>Learnability</i> (Kemudahan untuk dipelajari)	Tampilan per-menu mudah dipelajari	12
		Cara memainkan permainan mudah dipelajari	13
	<i>Attractiveness</i> (Menarik)	Tampilan setiap menu <i>game</i>	14
		Kelengkapan menu	15
		Animasi dari setiap objek yang dipilih dan pergantian antar menu	16

Tabel 28. Kelayakan Perangkat Lunak Aspek *Functionality*, *Efficiency*, dan *Usability*

No	Pernyataan	Alternatif Jawaban				
		STS	TS	N	S	SS
Variabel <i>Functionality</i>						
1	Perangkat lunak dapat menampilkan skor/nilai permainan					
2	Perangkat lunak dapat menampilkan permainan mencocokkan gambar					
3	Perangkat lunak dapat menampilkan definisi macam-macam rambu lalu lintas dan contohnya masing-masing					
4	Perangkat lunak dapat menampilkan halaman bantuan berisi pengertian perangkat lunak dan cara memainkan permainannya					
5	Setiap fitur/menu berfungsi sesuai dengan fungsinya masing-masing					
6	Perangkat lunak mampu bekerja memadukan fitur webstorage untuk penyimpanan nilai skor					
Variabel <i>Efficiency</i>						
7	Tiap proses membutuhkan jeda waktu yang singkat					
8	Respon dari setiap proses sesuai dengan fungsinya masing-masing					
Variabel <i>Usability</i>						
9	Kemudahan dalam mempelajari fitur-fitur yang ada					
10	Kejelasan dalam mengoperasikan fitur-fitur yang ada					
11	Memberikan informasi yang mudah dipahami					
12	Penggunaan tampilan per menu/fitur mudah dipelajari					
13	Permainan yang mudah dipelajari/dimainkan					
14	Tampilan per menu/fitur menggunakan objek gambar yang menarik					
15	Kelengkapan menu operasi					
16	Animasi perpindahan menu/fitur yang ada menarik					

b. Pengujian *Beta*

Pengujian *beta* dilakukan dengan menggunakan kuesioner yang dibuat berdasarkan kriteria dari *software quality factors* ISO 9126. Kuesioner pada pengujian *beta* sebelum digunakan terlebih dahulu divalidasi oleh ahli. Pengujian *beta* yang sudah divalidasi digunakan untuk menguji variabel *Functionality*,

Efficiency, Usability, dan Portability. Kuesioner pengujian beta akan diberikan kepada responden yang dipilih untuk menilai kelayakan perangkat lunak. Kuesioner yang akan digunakan pada pengujian *beta* menggunakan kisi-kisi sebagai berikut:

Tabel 29. Kisi-Kisi Instrumen Pengujian *Beta*

Variabel	Indikator	Sub Indikator	Item Butir
<i>Functionality</i>	<i>Suitability</i> (Kesesuaian)	Menjalankan semua fitur menu dalam <i>game</i>	1,2,3,4
		Semua fitur berjalan sebagai mana mestinya	5
	<i>Interoperability</i> (Interoperabilitas)	Berinteraksi dengan webstorage penyimpanan data	6
<i>Efficiency</i>	<i>Time Behaviour</i> (Perilaku waktu)	Lama waktu operasi tiap aksi	7
		Respon sesuai dengan setiap aksi dari pengguna	8
<i>Usability</i>	<i>Understandability</i> (Kemudahan untuk dipahami)	Fitur-fitur <i>game</i> yang disediakan mudah dipahami	9
		Kejelasan penggunaan setiap fitur	10
	<i>Learnability</i> (Kemudahan untuk dipelajari)	Memberikan informasi yang jelas	11
		Tampilan per-menu mudah dipelajari	12
		Cara memainkan permainan mudah dipelajari	13
	<i>Attractiveness</i> (Menarik)	Tampilan setiap menu <i>game</i>	14
		Kelengkapan menu	15

Tabel 30. Kisi-Kisi Instrumen Pengujian *Beta* (lanjutan)

Variabel	Indikator	Sub Indikator	Item Butir
		Animasi dari setiap objek yang dipilih dan pergantian antar menu	16
<i>Portability</i>	<i>Installability</i> (Kemampuan untuk Diinstal)	Kemudahan instalasi aplikasi	17
		Keberhasilan dalam instalasi	18

Kisi-kisi instrumen yang telah dibuat akan digunakan untuk membuat pernyataan-pernyataan yang akan mewakili setiap variabel yang diuji pada pengujian *beta*. Kuesioner yang digunakan pada pengujian *beta* berdasarkan kisi-kisi instrumen yang telah dibuat adalah sebagai berikut:

Tabel 31. Kuesioner Pengujian *Beta*

No	Pernyataan	Alternatif Jawaban				
		STS	TS	N	S	SS
Variabel <i>Functionality</i>						
1	Perangkat lunak dapat menampilkan skor/nilai permainan					
2	Perangkat lunak dapat menampilkan permainan mencocokkan gambar					
3	Perangkat lunak dapat menampilkan definisi macam-macam rambu lalu lintas dan contohnya masing-masing					
4	Perangkat lunak dapat menampilkan halaman bantuan berisi pengertian perangkat lunak dan cara memainkan permainannya					
5	Setiap fitur/menu berfungsi sesuai dengan fungsinya masing-masing					
6	Perangkat lunak mampu bekerja memadukan fitur webstorage untuk penyimpanan data skor					
Variabel <i>Efficiency</i>						
7	Tiap proses membutuhkan waktu yang singkat					

Tabel 32. Kuesioner Pengujian *Beta* (lanjutan)

No	Pernyataan	Alternatif Jawaban				
		STS	TS	N	S	SS
Variabel <i>Efficiency</i>						
8	Respon dari setiap proses sesuai dengan fungsinya masing-masing					
Variabel <i>Usability</i>						
9	Kemudahan dalam mempelajari fitur-fitur yang ada					
10	Kejelasan dalam mengoperasikan fitur-fitur yang ada					
11	Memberikan informasi yang mudah dipahami					
12	Penggunaan tampilan per menu/fitur mudah dipelajari					
13	Permainan yang mudah dipelajari/dimainkan					
14	Tampilan per menu/fitur menggunakan objek gambar yang menarik					
15	Kelengkapan menu operasi					
16	Animasi perpindahan menu/fitur yang ada menarik					
Variabel <i>Portability</i>						
17	Kemudahan dalam melakukan instalasi perangkat lunak					
18	Perangkat lunak dapat di install dengan sukses					

Keterangan : STS = Sangat Tidak Setuju

TS = Tidak Setuju

N = Netral

S = Setuju

SS = Sangat Setuju

J. Uji Instrumen Penelitian

Pengujian instrumen dilakukan untuk membuktikan bahwa instrumen yang akan digunakan untuk pengambilan data bersifat valid dan reliabel. Valid memiliki pengertian bahwa instrumen tersebut dapat digunakan untuk mengukur apa yang seharusnya diukur (Sugiyono, 2013:121). Pengujian validitas instrumen

untuk validitas konstruksi dapat digunakan pendapat dari ahli (*judgment experts*). Instrumen yang digunakan dalam pengambilan data pada penelitian ini akan diuji validitas konstruksinya oleh ahli sehingga instrumen nantinya akan bersifat valid.

K. Teknik Analisis Data

Tahap selanjutnya setelah proses pengumpulan data selesai adalah pengolahan dan analisis data. Analisis data pada penelitian ini dilakukan sesuai dengan pengukuran yang menggunakan instrumen kuesioner yaitu skala pengukuran Guttman dan Likert. Skala Likert digunakan untuk menghitung data variabel yang diujikan yaitu *functionality*, *efficiency*, *usability*, dan *portability*. Data kuantitatif hasil penelitian akan diubah menjadi data kualitatif dengan menggunakan skala Likert. Hasil dari analisis instrumen nantinya akan didapatkan skor tiap instrumen kemudian akan dihitung rata-rata dari instrumen dengan menggunakan rumus:

1. Rumus Perhitungan rata-rata instrumen:

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan : \bar{x} = Skor rata-rata

$\sum x$ = Skor total item

n = Jumlah item

2. Rumus perhitungan persentase skor ditulis dengan rumus berikut:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

Hasil perhitungan kemudian diubah ke dalam bentuk predikat dengan cara menggunakan skala Likert. Konversi persentase ke dalam bentuk pernyataan menurut Riduwan dan Sunarto (2012:23), dinyatakan seperti dalam tabel berikut:

Tabel 33. Interpretasi Persentase Likert

No	Persentase	Interpretasi
1	0% - 20%	Sangat Lemah
2	21% - 40%	Lemah
3	41% - 60%	Cukup
4	61% - 80%	Kuat
5	81% - 100%	Sangat Kuat

Tahap selanjutnya adalah menyesuaikan konversi persentase sesuai dengan penelitian yang dilakukan, maka skala konversi persentase disesuaikan interpretasinya. Penyesuaian tersebut dilakukan karena penelitian yang sedang dilakukan adalah untuk menguji tingkat kelayakan perangkat lunak. Skala konversi persentase disesuaikan menjadi seperti berikut:

Tabel 34. Penyesuaian Interpretasi Persentase Likert

No	Persentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

Hasil perhitungan analisis data penelitian nantinya akan didapat interpretasi kelayakan perangkat lunak. Hasil penelitian akan menentukan kualitas perangkat lunak sesuai dengan tujuan penelitian untuk menentukan kualitas dari perangkat lunak yang dikembangkan.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Hasil Penelitian

1. Implementasi

Implementasi merupakan tahap selanjutnya setelah perancangan desain perangkat lunak. Fungsi yang diperlukan kemudian diubah ke dalam bahasa pemrograman sesuai dengan kebutuhan perangkat lunak. Tahapan implementasi dilakukan secara bertahap sesuai dengan kebutuhan sistem yang sudah ditetapkan sebelumnya. Implementasi dalam proses pengembangan perangkat lunak ini meliputi:

a. Implementasi Desain Antarmuka dan Pengaturan *Event*

Tahap implementasi pada bagian ini dilakukan dengan menggunakan *software* Construct 2. Construct 2 merupakan *software* yang digunakan dalam pengembangan aplikasi jenis permainan. Area kerja Construct 2 terdiri dari dua bagian yang sangat penting pada tahap implementasi, yaitu *layout* dan *event sheet*. Bagian *layout* merupakan tempat untuk membuat atau menambahkan objek (*sprite*), sedangkan *event sheet* merupakan tempat dimana *event* ditambahkan. *Event* merupakan pilihan-pilihan *conditions* dan *actions* yang memuat atribut-atribut objek yang telah dibuat atau ditambahkan pada bagian *layout*. Pemanggilan fungsi-fungsi di Construct 2 hanya menggunakan pengaturan *events* yang telah disediakan pada aplikasi ini.

Tahap implementasi menghasilkan beberapa menu yang dibuat berdasarkan masing-masing objek dan juga fungsi. Menu tersebut adalah "*title*

screen" dan "menu utama" yang di dalamnya terdapat menu "*mulai bermain*", "*ayo belajar*", "*bantuan*", "*skor*", dan "*on/off musik*". Penjelasan di bawah ini akan membahas tentang pengaturan atribut-atribut objek yang digunakan untuk menyusun masing-masing menu yang ada pada perangkat lunak "*Need For Safety*".

1) Halaman Judul (*Title Screen*)

Title screen merupakan halaman awal yang muncul setelah *user* menjalankan aplikasi sebelum masuk menu utama. *Title screen* terdiri dari dua objek yaitu objek gambar *background* (dapat dilihat pada Gambar 26) dan objek tombol "*mulai*" (dapat dilihat pada Gambar 27).



Gambar 26. Objek *Background* Halaman *Title Screen*



Gambar 27. Objek Tombol "Mulai"

Objek gambar *background* dimasukkan ke dalam lembar kerja *layout* Construct 2, kemudian disesuaikan ukurannya sesuai dengan *layout size* pada lembar kerja *layout*. Objek yang kedua dimasukkan juga ke dalam *layout* kemudian diatur atribut objeknya dengan menggunakan pengaturan *event*. Penambahan *event* berupa *conditions* dan *actions* untuk objek-objek pada halaman judul aplikasi adalah sebagai berikut:

Tabel 35. Pengaturan Fungsi *Event* pada "Title Screen"

Conditions		Actions	
System	On start of layout	System	Wait 1.5 seconds
		"mulai"	Set visible
		"mulai"	Set position to (939,66)
Touch	On touch "mulai"	Audio	Play SFX1 not looping at volume 0 db
		System	Go to "menu"

Penjelasan dari penambahan *event* pada Tabel 35 adalah pada saat objek dalam *layout* mulai dijalankan, objek *background* akan tampil setelah jeda waktu 1,5 detik. *Event* untuk menjalankan fungsi jeda waktu adalah "***wait n seconds***", dimana "*n*" adalah lama waktu jeda. Objek *background* akan tampil diikuti objek "*mula*", dimana letak objek "*mula*" sudah diatur yaitu dengan perintah "***set position to (939,66)***". Fungsi "***set position to (x,y)***" adalah fungsi untuk mengatur letak objek dalam *layout*, dimana nilai "*x*" dan "*y*" dalam satuan *pixels*. Kondisi kedua adalah membuat objek "*mula*" ketika ditekan akan memainkan suara "SFX1" yaitu dengan perintah "***Play SFX1 not looping at volume 0 db***". Fungsi "***not looping***" adalah untuk membuat suara dimainkan sekali tanpa perulangan dan "***volume 0 db***" adalah untuk pengaturan volume suara. Nilai diatas 0 db akan membuat suara menjadi lebih keras dan nilai dibawah 0 db akan membuat suara menjadi lebih lemah. Fungsi *event* selanjutnya adalah ***Go to "n"***, dimana "*n*" adalah nama *layout* yang akan dipanggil. Fungsi *go to "menu"* menandakan bahwa setelah objek tombol "mulai" dipilih, maka aksi selanjutnya adalah menuju *layout* "menu".



Gambar 28. Tampilan Halaman "Title Screen"

2) Menu Utama Aplikasi

Menu utama merupakan halaman dimana *user* dapat mengakses beberapa menu yang disediakan dalam aplikasi ini. Menu utama terdiri dari menu "*mulai bermain*", "*ayo belajar*", "*bantuan*", "*skor*", dan "*on/off musik*".

Tabel 36. Pengaturan Fungsi *Event* pada "Menu Utama"

Conditions		Actions	
System	On start of layout	Audio	Play mainfs looping at volume 0 dB
		bgmenu	Fade: Start fade
		System	Wait 0.5 seconds
		bt_mulaimain	LiteTween start from beginning
		bt_ayobelajar	LiteTween start from beginning
		bt_bantuan	LiteTween start from beginning
		bt_vol	Set position to (118,45)
		bt_hs	Set position to (224,45)
		System	Wait 0.5 seconds
Touch	On touch "bt_mulaimain"	bt_mulaimain	Set animation to "onklik"
		Audio	Play SFX1 not looping
		System	Wait 0.3 seconds
		System	Go to mainlevel1
Touch	On touch "bt_ayobelajar"	bt_mulaimain	Set animation to "onklik"
		Audio	Play SFX1 not looping
		System	Wait 0.3 seconds
		System	Go to ayobelajar
Touch	On touch "bt_bantuan"	bt_mulaimain	Set animation to "onklik"
		Audio	Play SFX1 not looping
		System	Wait 0.3 seconds
		System	Go to bantuan
Touch	On touch "bt_vol"	Bt_vol	Set position to (118,-122)
		Bt_vol2	Set position to (118,45)
		Audio	Pause tag "intro main"
Touch	On touch "bt_vol2"	Bt_vol	Set position to (118,45)
		Bt_vol2	Set position to (-135,-2)
		Audio	Resume tag "intro main"
Touch	On touch "bt_hs"	System	Go to highscore

Penjelasan untuk fungsi-fungsi diatas adalah pada kondisi pertama saat objek-objek pada *layout* dijalankan, objek gambar *background* "bgmenu" akan tampil dengan animasi "fade" diikuti dengan objek-objek lain yang berfungsi sebagai tombol atau *button* untuk menuju ke masing-masing menu. *Fade* dan *LiteTween* merupakan animasi yang sudah disediakan oleh Construct 2. **Start Fade** adalah fungsi untuk memulai animasi *fade* pada objek bgmenu, sedangkan animasi *litetween* akan mulai dijalankan dengan perintah "**LiteTween Start From Beginning**". Kondisi selanjutnya adalah membuat fungsi tombol untuk objek dengan nama "bt_mulaimain", "bt_ayobelajar", "bt_bantuan", "bt_vol", "bt_vol2", dan "bt_hs". **Set animation to "Onklik"** adalah fungsi untuk merubah animasi objek dengan nama "Onklik". Fungsi ini membuat tampilan objek yang berfungsi sebagai tombol akan memiliki animasi yang berbeda saat tidak ditekan dan pada saat tombol ditekan.



Gambar 29. Tampilan Menu Utama

3) Menu Mulai Bermain

Menu mulai bermain merupakan menu permainan yang akan tampil setelah *user* memilih tombol "bt_mulaimain" yang adalah di menu utama. Menu mulai bermain terdiri dari terdiri dari empat level permainan mencocokkan gambar.

Tabel 37. Pengaturan Fungsi *Event* untuk Menampilkan Kotak

Conditions		Actions	
System	On start of layout	System	Reset global variables to default
		System	Set gTinggiKotak to $\text{int}((\text{bgc.Height} - ((\text{gJumlahBaris} + 1) * \text{gJarakAntarKotak})) / \text{gJumlahBaris})$
		System	Set gLebarKotak to $\text{int}(\text{gTinggiKotak} * (\text{kotaklevel1.Width} / \text{kotaklevel1.Height}))$
		System	Set gMarginKiri to $\text{int}(((\text{bgc.Width} - (\text{gLebarKotak} + \text{gJarakAntarKotak}) * \text{gJumlahKolom}) / 2) + 130)$
		System	Set gMarginAtas to $\text{int}(((\text{bgc.Height} - (\text{gTinggiKotak} + \text{gJarakAntarKotak}) * \text{gJumlahBaris})) / 2) + 155)$

Reset global variables to default merupakan *event* untuk mengembalikan nilai *global variable* ke nilai awal. ***Set gTinggiKotak to int*** adalah fungsi untuk mengatur nilai dari *global variable* "gTinggiKotak", yang merupakan variabel untuk menentukan ukuran tinggi kotak yang akan dibuat. *Global variable* yang lainnya adalah gLebarKotak untuk menentukan ukuran lebar kotak, gMarginKiri untuk menentukan jarak kotak dari tepi kiri *layout*, dan gMarginAtas untuk menentukan jarak kotak dari tepi atas *layout*.

Menu mulai bermain terdiri dari objek-objek yang sudah dibuat pengembang yaitu "*bgc*" yang merupakan objek gambar *background*, objek kotak gambar rambu, objek teks untuk menampilkan jumlah langkah, cocok, dan skor, serta objek tombol untuk menuju ke menu utama. Level pada menu permainan dibedakan dari jumlah kotak. Jumlah kotak dibuat dengan pengaturan fungsi pada Tabel 37. Jumlah kotak ditentukan dari nilai variabel *gJumlahKolom*, *gJumlahBaris*, dan *gJumlahKotak*. Nilai dari masing-masing variabel tersebut adalah:

Tabel 38. Nilai Variabel untuk Jumlah Kotak Tiap Level

Global Variable	Nilai (Type: number)			
	Level 1	Level 2	Level 3	Level 4
gJumlahKolom	4	5	6	7
gJumlahBaris	4	4	4	4
gJumlahKotak	16	20	24	28

Level pertama pada menu permainan akan menampilkan jumlah kotak sebanyak 16, level 2 sebanyak 20, level 3 sebanyak 24 dan level terakhir akan menampilkan kotak sebanyak 28.



Gambar 30. Permainan Level 1



Gambar 31. Permainan Level 2



Gambar 32. Permainan Level 3



Gambar 33. Permainan Level 4

Kotak pada tiap level menampilkan gambar rambu-rambu lalu lintas. Permainan mencocokkan gambar ini adalah untuk menghabiskan seluruh kotak yang ada dengan cara memilih dua gambar kotak yang sama. Fungsi untuk membuat kotak dapat menampilkan gambar rambu secara acak adalah sebagai berikut:

Tabel 39. Fungsi *Event* untuk Menampilkan Gambar Kotak secara Acak

Conditions		Actions	
System	Repeat gJumlahKotak times	System	Set slot1 to int(random(0,gJumlahKotak))
		System	Set slot2 to int(random(0,gJumlahKotak))
		System	Set Kotak1 to Tumpukan.At(Slot1)
		System	Set Kotak2 to Tumpukan.At(Slot2)
		Tumpukan	Set value at Slot1 to Kotak2
		Tumpukan	Set value at Slot2 to Kotak1

Variabel *Kotak1*, *Kotak2*, *Slot1*, dan *Slot2* diisi dengan nilai 0 (*number*). **Set slot1 to int(random())** merupakan fungsi untuk membuat nilai acak yang akan disimpan dalam variabel *slot1*. Fungsi acak objek gambar rambu kemudian dimasukkan ke dalam array "Tumpukan". Fungsi ini akan mengacak posisi objek gambar sejumlah dengan jumlah kotak pada tiap level.



Gambar 34. Tampilan Gambar Rambu secara Acak

Menu permainan akan menampilkan informasi nilai setelah *user* memainkan semua level. Pengaturan fungsi *event* pada halaman informasi nilai adalah sebagai berikut:

Tabel 40. Pengaturan Fungsi *Event* pada Halaman Informasi Nilai

Conditions		Actions	
System	On start of layout	nilai	LiteTween start from the beginning
		System	Set gTotalLangkah to gLangkah1+gLangkah2+gLangkah3+gLangkah4
		btb	Set position to (1050,235)
		bt_hs	Set position to (1050,368)
		bth	Set position to (1050,502)
Touch	On touch bth	System	Go to menu
Touch	On touch btb	System	Go to mainlevel1
Touch	On touch bt_hs	System	Go to highscore

Set gTotalLangkah to (n) merupakan fungsi untuk mengatur nilai variabel gTotallangkah dengan nilai "n". Menu informasi nilai akan menampilkan total langkah dan total nilai yang didapatkan *user* setelah memainkan semua level. Halaman informasi nilai terdapat tiga *button* yaitu "btb" untuk memulai permainan dari level 1, "bth" untuk menuju menu utama, dan "bt_hs" untuk menuju halaman skor.



Gambar 35. Informasi Nilai Permainan

4) Menu Skor

Menu skor merupakan halaman untuk menyimpan nilai yang didapatkan *user* setelah memainkan permainan. Skor yang ditampilkan pada halaman ini berjumlah tiga yang merupakan skor tertinggi yang didapatkan *user*. Skor

tertinggi akan berada pada urutan paling atas dan diikuti skor dibawahnya.

Halaman menu skor terdapat tombol "*home*" dan "*hapus*".

Tabel 41. Pengaturan Fungsi *Event* pada Menu Skor

Conditions		Actions	
WebStorage	Local key "dbHiskor" exists	System	Set HiSkor to WebStorage.LocalValue("dbHiSkor")
System	For each txtHiSkor	txtHiskor	Set text to arSkor.At(n)
		System	Subtract 1 from 2
		WebStorage	Set local key "dbHiSkor" to arSkor.At(0) & "," & arSkor.At(1) & "," & arSkor.At(2)
Touch	On touched bth	System	Go to menu
		System	Reset global variables to default
Touch	On touched hapus	System	Clear local storage

Penjelasan untuk pengaturan *event* pada Tabel 41 adalah kondisi pertama yaitu **local key "dbHiskor" exists** merupakan kondisi pengecekan ada tidaknya dbHiskor pada webstorage. Fungsi pengecekan akan dilanjutkan jika terdapat dbHiskor maka aksi sistem adalah menyimpan nilai pada variabel lokal dengan nama *HiSkor* ke dalam WebStorage "*dbHiSkor*". Fungsi *event* untuk menyimpan data ke dalam webstorage adalah **set n to webstorage.localvalue(m)**, dimana "*n*" adalah variabel lokal yang berisi nilai yang akan disimpan dan "*m*" adalah variabel yang dibuat di dalam webstorage untuk tempat menyimpan data. Kondisi kedua adalah untuk menjalankan aksi memasukkan nilai yang ada dalam webstorage ke dalam array dengan nama "*arSkor*". Aksi tersebut adalah penjelasan untuk *event set localkey (n) to m*, dimana *n* adalah nilai yang tersimpan dalam webstorage dan *m* adalah array yang telah dibuat. Kondisi ketiga adalah untuk membuat tombol menuju menu utama pada objek "*bth*", dan kondisi terakhir adalah membuat fungsi tombol "*hapus*" untuk mengosongkan nilai yang tersimpan dalam WebStorage dengan perintah "**Clear local storage**".



Gambar 36. Menu Skor

5) Menu Ayo Belajar

Menu ayo belajar merupakan menu untuk menampilkan contoh-contoh gambar rambu lalu lintas. Menu ini terdiri dari lima submenu yaitu submenu pengertian rambu, rambu peringatan, rambu larangan, rambu perintah, dan rambu petunjuk. Menu ini juga terdapat tombol "*home*" untuk menuju menu utama aplikasi.

Tabel 42. Pengaturan Fungsi *Event* pada Menu Ayo Belajar

Conditions		Actions	
System	On start of layout	bg_ayobelajar	Fade: start fade
		ayobelajar_field	LiteTween start from the beginning
		bt_ab	LiteTween start from the beginning
		bt_ab2	LiteTween start from the beginning
		bt_ab3	LiteTween start from the beginning
		bt_ab4	LiteTween start from the beginning
		bt_ab5	LiteTween start from the beginning
		bth	LiteTween start from the beginning
Touch	On touched bth	Audio	Play SFX1 not looping
		System	Go to menu
Touch	On touched bt_ab	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambulalulintas
Touch	On touched bt_ab2	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambuperingatan
Touch	On touched bt_ab3	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambularangan
Touch	On touched bt_ab4	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambuperintah
Touch	On touched bt_ab5	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambupetunjuk

Event pada Tabel 42 berfungsi untuk menampilkan objek berupa submenu-submenu ayo belajar dan juga tombol "*home*". Submenu ayo belajar akan muncul setelah objek gambar *background* "*bg_ayobelajar*" tampil menggunakan animasi *fade*. Submenu akan tampil dengan animasi *litetween* setelah objek gambar *background* tampil.



Gambar 37. Submenu Ayo Belajar

Halaman Ayo Belajar ini terdiri dari lima submenu yang digunakan untuk menampilkan halaman pengertian rambu. Pengertian rambu dibuat dalam bentuk objek gambar. Halaman pengertian rambu ini juga terdapat tombol "x" untuk menutup halaman dan kembali ke halaman sebelumnya, dan juga terdapat tombol "→" untuk menuju ke halaman contoh rambu.

Tabel 43. Pengaturan Fungsi *Event* Halaman Pengertian Rambu

Conditions		Actions	
System	On start of layout	rambuperingatan	Fade: start fade
		btx	Set position (1041,70)
		btn	Set position (380,593)
Touch	On touched btx	Audio	Play SFX1 not looping
		System	Go to ayobelajar
Touch	On touched btn	Audio	Play SFX1 not looping
		System	Go to rambuperingatan



Gambar 38. Halaman Pengertian Rambu

Halaman selanjutnya setelah *user* menekan tombol " → " adalah halaman yang menampilkan contoh rambu-rambu lalu lintas. Objek yang digunakan adalah berupa objek gambar yang sudah dibuat.

Tabel 44. Pengaturan Fungsi *Event* Halaman Contoh Rambu

Conditions		Actions	
System	On start of layout	rambuko song	Fade: start fade
		btx	Set position (1028,90)
		btb	Set position (932,90)
Touch	On touched btx	Audio	Play SFX1 not looping
		System	Go to ayobelajar
Touch	On touched btb	Audio	Play SFX1 not looping
		System	Go to ayobelajar_rambuperingatan
Touch	On touched btx2	Audio	Play SFX1 not looping
		btx2	Set position to (1312,-155)
Touch	On touched r1	g1	Set position to (99,29)
		btx	Set position to (1404,-69)

Halaman contoh rambu akan tampil seperti pada Gambar 39, dan ketika *user* memilih salah satu gambar contoh rambu yang ada, akan tampil objek gambar yang berisi pengertian dari rambu yang dipilih *user* tadi (tampilan pada Gambar 40).



Gambar 39. Halaman Contoh Rambu



Gambar 40. Halaman Pengertian Contoh Rambu

6) Menu Bantuan

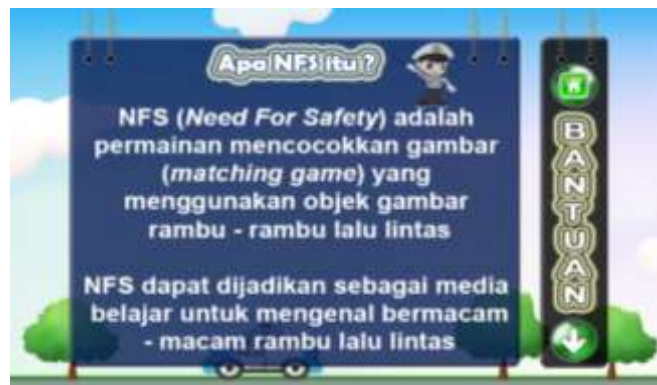
Menu bantuan adalah halaman yang menampilkan beberapa bantuan untuk *user* dalam menggunakan aplikasi "*Need For Safety*". Halaman bantuan terdiri dari teks bantuan dibuat dalam bentuk objek gambar, objek tombol "*home*" untuk menuju menu utama dan objektombol " " untuk menuju halaman bantuan selanjutnya.



Tabel 45. Pengaturan Fungsi *Event* Halaman Bantuan

Conditions		Actions	
System	On start of layout	bgbantuan	Fade: start fade
		bth	Set position (1025,140)
		btg	Set position (1025,645)
		bantuan_field1b	LiteTween start from the beginning
Touch	On touched bth	Audio	Play SFX1 not looping
		System	Go to menu
Touch	On touched btg	Audio	Play SFX1 not looping
		bantuan_field1	Set position to (145,-763)
Touch	On touched btg2	Audio	Play SFX1 not looping
		bantuan_field2	Set position to (145,-763)

Penjelasan untuk fungsi *event* di atas adalah pada saat *layout* dijalankan, objek gambar *background* akan tampil kemudian diikuti dengan objek-objek lain. Kondisi kedua adalah membuat fungsi *button* untuk objek "*bth*" yang berfungsi untuk menuju menu utama, objek "*btg*" dan "*btg2*" untuk menampilkan objek gambar selanjutnya.



Gambar 41. Halaman Bantuan

7) *On/Off* Musik

Fungsi ini adalah untuk pengaturan suara pada aplikasi. Musik yang digunakan (*background*) akan mulai dimainkan setelah *user* masuk pada menu utama aplikasi. *User* dapat menonaktifkan dan juga dapat mengaktifkannya kembali melalui menu utama.

Tabel 46. Pengaturan Fungsi *Event* On/Off Musik

Conditions		Actions	
Touch	On touched bt_vol	Audio	Pause tag "intro main"
		bt_vol2	Set position to (118,45)
Touch	On touched bt_vol2	Audio	Resume tag "intro main"
		bt_vol	Set position to (118,45)

Kondisi pertama pada pengaturan fungsi *event* di atas adalah pada saat button "bt_vol2" yang merupakan button *off* musik ditekan, fungsi ***pause tag "intro main"*** akan menghentikan *background* musik dengan nama "intro main" dan fungsi ***resume tag "intro main"*** adalah untuk memainkan *background* kembali.

b. *Deployment* perangkat lunak

Proses *deployment* merupakan proses menjalankan perangkat lunak untuk mengetahui kesesuaian fungsi yang diimplementasikan dengan kode program sebelumnya. *Deployment* perangkat lunak untuk aplikasi "Need For Safety" dimulai dengan menjalankan *project* melalui browser Google Chrome. Setelah observasi yang langsung dilakukan pada saat uji coba menggunakan aplikasi browser dan semua fungsi-fungsi sudah sesuai dengan yang diharapkan kemudian mengubah atau *export project* menjadi .apk file.

1) Menjalankan *Project* Menggunakan Aplikasi Browser

Langkah-langkah menjalankan *project* dilakukan dengan cara berikut:

- a) Klik Pilihan *Preview browser* pada *Configuration Settings* kemudian pilih Chrome.

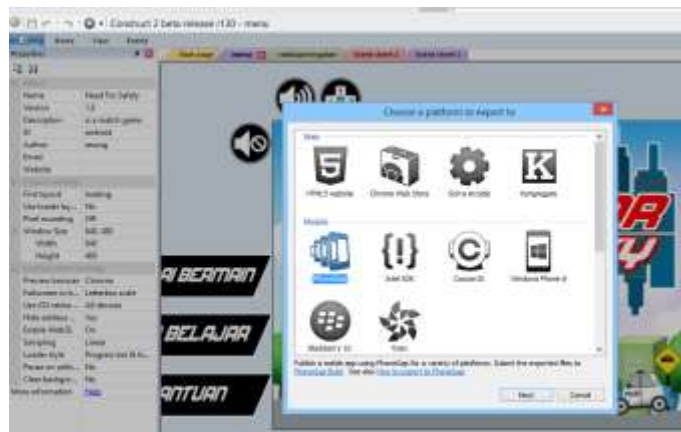


Gambar 44. *Project* Dijalankan Melalui Google Chrome

2) *Export Project* menjadi *.apk file*

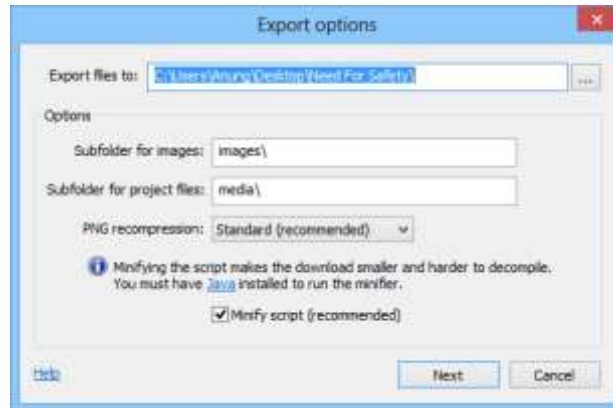
Tahap selanjutnya setelah fungsi-fungsi program yang ada dicek dan sudah sesuai dengan fungsinya masing-masing, *project* akan diubah menjadi *installer .apk file*. Proses *export file project* ke *.apk file* dilakukan secara *online*. Langkah-langkah *exporting project* menjadi *.apk file* adalah sebagai berikut:

- a) Klik *File – export project* kemudian pilih *Phonegap*, dan selanjutnya klik tombol *Next*.



Gambar 45. *Exporting project* menggunakan *Phonegap*

- b) Pilih tempat menyimpan *file project*, kemudian klik tombol *Next*.



Gambar 46. Pilihan tempat menyimpan *file*

- c) Pilih *options Supported orientation* dan *Supported devices*, kemudian klik tombol *export*.



Gambar 47. *Options* dukungan mode layar dan *devices*

- d) Klik *Submit to PhoneGap Build* untuk membuka site <https://build.phonegap.com/>.



Gambar 48. *Options Export HTML5 project*

- e) Tampilan halaman awal site PhoneGap dapat dilihat pada Gambar 49.



Gambar 49. Halaman *Home* Situs *PhoneGap*

- f) *Login* menggunakan AdobeID atau *register* terlebih dahulu jika belum terdaftar dalam AdobeID.



Gambar 50. Registrasi *User* Baru Menggunakan *AdobeID*

- g) *Upload* file hasil *exporting project* sebelumnya dengan menekan tombol *upload a .zip file*.



Gambar 51. Halaman *Upload Project File*

h) Klik tombol *.apk* untuk menjadikan *project* menjadi *installer apk*.



Gambar 52. Membuat *Installer apk*

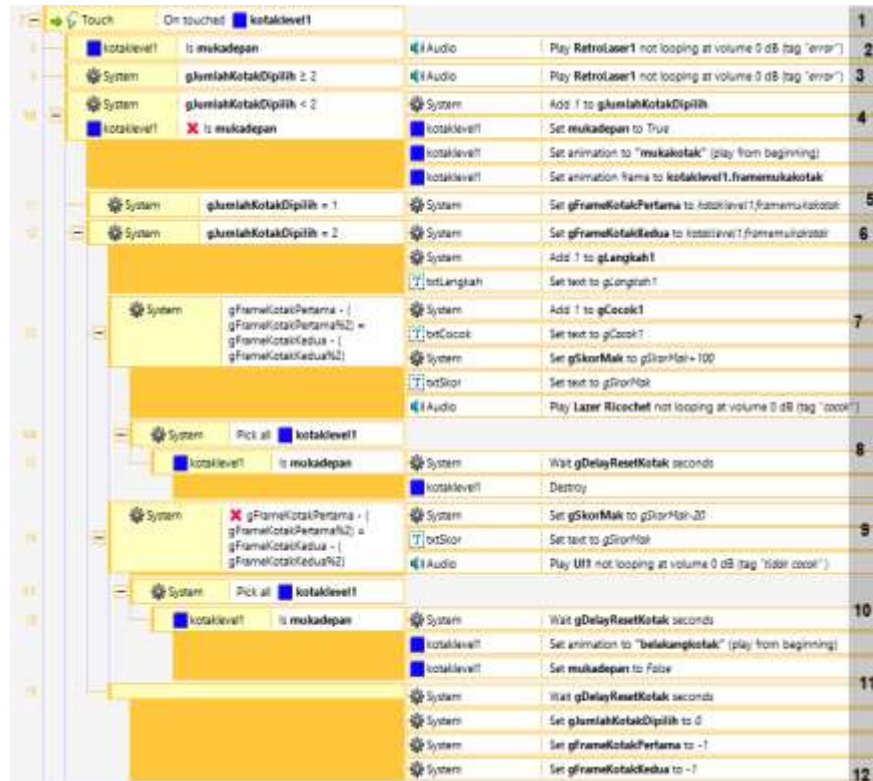
2. Pengujian Perangkat Lunak

a. *White Box Testing*

Pengujian *white box* merupakan pengujian perangkat lunak yang berfokus pada struktur kontrol program. Pengujian ini dilakukan oleh peneliti dengan menggunakan teknik pengujian jalur dasar (*basic path testing*) pada saat proses implementasi perangkat lunak. Langkah-langkah pengujian *white box* pada perangkat lunak "*Need For Safety*" adalah sebagai berikut :

1) Menggambarkan alur logika ke dalam grafik alir (*flow graph*).

Pengujian *white box* peneliti mulai dari bagian fungsi Pilih_Kotak. *Event* pada Pilih_Kotak di menu Mulai Bermain dapat dilihat pada Gambar 53.



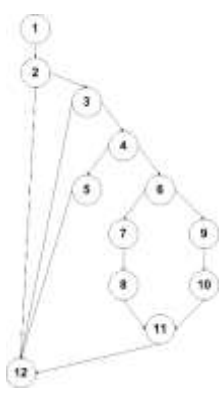
Gambar 53. Pengujian *White Box* Fungsi Pilih_Kotak

Penulisan angka di sebelah kanan *actions* program di atas menunjukkan nomor node *flow graph* pada fungsi Pilih_Kotak. Statemen yang tidak diberi angka menunjukkan bahwa statemen tersebut memiliki nomor yang sama sehingga akan diwakili dengan satu node.

2) Menentukan *cyclomatic complexity* dan basis set

Nomor node *flow graph* yang telah dibuat akan digunakan untuk menggambarkan notasi *flow graph*. *Flow graph* dari fungsi Pilih_Kotak digambarkan di dalam Tabel 47.

Tabel 47. *Flow graph* Fungsi Pilih_Kotak

Flow Graph	Cyclomatic Complexity	Independent Path
	1. Jumlah region = 5 2. $V(G) = 15 - 12 + 2 = 5$ 3. $V(G) = 4 \text{ Node Predikat} + 1 = 5$	1. 1-2-12 2. 1-2-3-12 3. 1-2-3-4-5-12 4. 1-2-3-4-6-7-8-11-12 5. 1-2-3-4-6-9-10-11-12

Perhitungan nilai kompleksitas siklomatik (*cyclomatic complexity*) dapat menggunakan beberapa rumus sebagai berikut:

- Menghitung jumlah *region* atau daerah-daerah yang berhubungan dengan kompleksitas siklomatik. *Flow graph* pada Tabel 47 menggambarkan jumlah *region* sebanyak 5 sehingga nilai kompleksitas siklomatiknya adalah 5.
- Menggunakan rumus $V(G) = E - N + 2$, dimana E adalah jumlah *edge* atau gambar panah yang menunjukkan aliran kendali atau perjalanan logika dan N adalah jumlah node grafik alir. *Flow graph* pada Tabel 47 menggambarkan jumlah *edge* sebanyak 15 dan jumlah node sebanyak 12, sehingga kompleksitas siklomatik atau $V(G) = 15 - 12 + 2 = 5$.
- Menggunakan rumus $V(G) = P + 1$, dimana P adalah jumlah node predikat yang terdapat dalam grafik alir. Node predikat merupakan node yang memiliki percabangan *edge* atau aliran kendali. *Flow graph* pada

Tabel 47 menggambarkan jumlah node predikat sebanyak 4, sehingga kompleksitas siklomatik atau $V(G) = 4+1 = 5$.

Perhitungan nilai kompleksitas siklomatik pada fungsi Pilih_Kotak menggunakan beberapa rumus menghasilkan nilai 5. Nilai yang didapat dari perhitungan kompleksitas siklomatik digunakan dalam menentukan jumlah jalur independen. Jalur independen merupakan jalur pada program yang menghubungkan node awal dengan node akhir. Jumlah jalur independen pada fungsi Pilih_Kotak adalah 5 jalur yang dijelaskan dalam Tabel 47.

3) Membuat *test case* untuk masing-masing basis set.

Langkah terakhir pengujian jalur dasar adalah dengan membuat *test case* untuk masing-masing basis set. *Test case* dari fungsi Pilih_Kotak dijelaskan dalam Tabel 48.

Tabel 48. Uji *Test Case* Fungsi Pilih_Kotak

Path	Input	Output	Hasil Yang Diharapkan
1	Pilih Kotak Gambar saat kotak terbuka	Play audio "RetroLaser1"	Sesuai
2	Pilih Kotak Gambar sebanyak 3 buah	Play audio "RetroLaser1"	Sesuai
3	Pilih Kotak Gambar saat kotak tertutup sebanyak 1 buah	Kotak Gambar terbuka, set gFrameKotakPertama to kotaklevel1.frameemukakotak	Sesuai
4	Pilih 2 Kotak Gambar yang sama gambarnya	Jumlah langkah + 1, Jumlah Cocok + 1, Skor +100, Play audio "Lazer Ricochet", Kotak yang sama hilang	Sesuai
5	Pilih 2 Kotak Gambar yang berbeda gambarnya	Skor -20, Play audio "U11", Kotak yang dipilih kembali menutup, Kotak yang dipilih tidak hilang	Sesuai

Pengujian masing-masing jalur pada fungsi Pilih_Kotak menunjukkan bahwa *output* atau hasil yang diharapkan sudah sesuai harapan. Data pengujian

white box selengkapnya dapat dilihat pada Lampiran 3. Pengujian selanjutnya setelah *white box* adalah pengujian *black box*.

b. Black Box Testing

Pengujian *black box* merupakan pengujian terintegrasi yang dilakukan oleh peneliti untuk memastikan bahwa aplikasi sudah siap untuk diuji tahap selanjutnya yaitu tahapan pengujian *alpha*. Pengujian *black box* dilakukan tanpa menguji desain dan kode program tetapi menguji sistem dari segi spesifikasi fungsionalnya. Peneliti melakukan pengujian *black box* perangkat lunak "*Need For Safety*" dengan membagi menjadi 13 bagian sebagaimana *use case* pada analisis kebutuhan dan perancangan sistem.

Tabel 49. Uji *Black Box* Memulai Aplikasi

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. Menjalankan aplikasi		
	2. Menuju ke halaman judul	Sesuai
3. <i>User</i> menekan tombol "mulai"		
	4. Menuju ke halaman menu utama 5. Mengaktifkan musik <i>game</i>	Sesuai

Pengujian *black box* memulai aplikasi dilakukan dengan menjalankan aplikasi. Reaksi sistem pada awal aplikasi dijalankan adalah menampilkan halaman judul, dan ketika *user* menekan tombol "mulai", sistem menampilkan halaman menu utama dan memainkan musik *game*. Hasil pengujian *black box* memulai aplikasi sudah sesuai dengan harapan yaitu reaksi sistem sesuai dengan *input* yang diberikan oleh aktor.

Tabel 50. Uji *Black Box On/Off Musik Game*

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. Masuk menu utama kemudian <i>user</i> memilih tombol <i>Off</i> musik		
	2. Menonaktifkan musik <i>game</i>	Sesuai
3. Saat tombol musik off, <i>user</i> memilih tombol <i>On</i> musik		
	4. Mengaktifkan musik <i>game</i>	Sesuai

Pengujian *black box On/Off Musik* dilakukan dengan menekan tombol *off* musik pada menu utama, dan reaksi sistem adalah mematikan musik. Aksi aktor selanjutnya adalah menekan tombol *on* musik dan reaksi sistem adalah memainkan musik. Pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 51. Uji *Black Box Lihat Skor*

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu Skor		
	2. Mengecek data skor pada webstorage 3. Terdapat data skor yang disimpan di webstorage, menampilkan 3 skor terbesar	Sesuai
4. <i>User</i> memilih tombol "hapus"		
	5. Mengosongkan data skor pada webstorage	Sesuai
6. <i>User</i> memilih tombol "home"		
	7. Kembali ke menu utama	Sesuai
Skenario Alternatif		
4. <i>User</i> memilih menu Skor		
	5. Mengecek data skor pada webstorage 6. Belum ada data yang tersimpan di webstorage, menampilkan 3 skor terbesar dengan nilai "nol"	Sesuai

Pengujian *black box* menu lihat skor dilakukan dengan menjalankan menu lihat skor. Reaksi sistem adalah mengecek data skor yang tersimpan di webstorage, kemudian menampilkannya. Aksi aktor ketika menekan tombol "*home*" adalah sistem akan menampilkan halaman menu utama, dan ketika aktor menekan tombol "hapus", nilai skor yang tersimpan di webstorage akan dikosongkan. Sistem akan menampilkan nilai skor "nol" jika tidak terdapat nilai yang tersimpan dalam webstorage. Hasil pengujian ini sudah sesuai dengan harapan yaitu reaksi sistem sudah sesuai dengan *input* yang diberikan aktor.

Tabel 52. Uji *Black Box Get Score Data*

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> meminta data skor dari webstorage		
	2. Webstorage memberikan data skor atau nilai yang telah tersimpan untuk ditampilkan pada menu skor	Sesuai

Pengujian *black box get score data* dilakukan dengan menjalankan menu lihat skor. Reaksi sistem ketika *user* ingin melihat skor adalah webstorage akan menampilkan data nilai skor. Pengujian ini sudah sesuai dengan harapan yaitu reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 53. Uji *Black Box* Masuk Menu Mulai Bermain

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "mulai bermain"		
	2. Menuju halaman permainan level 1	Sesuai
3. <i>User</i> memainkan permainan level 1		
	4. Objek kotak habis 5. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 1 6. Menuju level kedua	Sesuai
7. <i>User</i> memilih tombol "home" di level 1		
	8. Menyimpan skor level pertama 9. Kembali ke menu utama	Sesuai
10. <i>User</i> memainkan permainan level 2		
	11. Objek kotak habis 12. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 2 13. Menuju level ketiga	Sesuai
14. <i>User</i> memilih tombol "home" di level 2		
	15. Menyimpan skor level pertama + kedua 16. Kembali ke menu utama	Sesuai
17. <i>User</i> memainkan permainan level 3		
	18. Objek kotak habis 19. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 3 20. Menuju level keempat	Sesuai
21. <i>User</i> memilih tombol "home" di level 3		
	22. Menyimpan skor level pertama + kedua + Ketiga 23. Kembali ke menu utama	Sesuai
24. <i>User</i> memainkan permainan level 4		
	25. Objek kotak habis 26. Menampilkan jumlah langkah, jumlah objek yang dicocokkan, skor level 4 27. Menuju ke halaman tampil skor 28. Menampilkan total langkah dan skor 29. Menyimpan skor dalam variabel "gskor" 30. Menyimpan nilai variabel "gskor" dalam webstorage dengan key "dbHiskor"	Sesuai
31. <i>User</i> memilih tombol "home" di level 4		
	32. Menyimpan skor level pertama + kedua + ketiga + keempat 33. Kembali ke menu utama	Sesuai

Tabel 54. Uji *Black Box* Masuk Menu Mulai Bermain (lanjutan)

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
34. <i>User</i> memilih tombol "skor" di halaman nilai		
	35. Menuju halaman skor	Sesuai
36. <i>User</i> memilih tombol "back" di halaman nilai		
	37. Menuju permainan level 1	Sesuai
38. <i>User</i> memilih tombol "home" di halaman nilai		
	39. Menuju menu utama	Sesuai

Pengujian *black box* mulai bermain dilakukan dengan menjalankan menu mulai bermain. Reaksi sistem setelah *user* memilih menu mulai bermain adalah menampilkan permainan level 1. *User* menyelesaikan permainan level 1 dan sistem akan menampilkan informasi nilai untuk level 1. Reaksi sistem selanjutnya adalah menampilkan permainan level 2. Aksi aktor setelah menyelesaikan permainan level 2 adalah sistem akan menampilkan permainan level 3, dan jika level 3 sudah diselesaikan, sistem akan menampilkan permainan level 4. Reaksi sistem setelah *user* selesai memainkan permainan level 4 adalah menampilkan informasi nilai. Reaksi sistem ketika *user* menekan tombol "skor" adalah menampilkan halaman lihat skor, dan ketika *user* menekan tombol "back", sistem akan menampilkan permainan level 1. Reaksi sistem ketika *user* menekan tombol "home" adalah menampilkan menu utama. Pengujian pada bagian ini sudah sesuai dengan harapan yaitu reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 55. Uji *Black Box* Ayo Belajar Rambu

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "ayo belajar"		
	2. Menampilkan submenu rambu	Sesuai
3. <i>User</i> memilih submenu "pengertian rambu"		
	4. Menampilkan halaman pengertian rambu	Sesuai
5. <i>User</i> memilih submenu "rambu peringatan"		
	6. Menampilkan halaman pengertian rambu peringatan	Sesuai
7. <i>User</i> memilih submenu "rambu larangan"		
	8. Menampilkan halaman pengertian rambu larangan	Sesuai
9. <i>User</i> memilih submenu "rambu perintah"		
	10. Menampilkan halaman pengertian rambu perintah	Sesuai
11. <i>User</i> memilih submenu "rambu petunjuk"		
	12. Menampilkan halaman pengertian rambu petunjuk	Sesuai
13. <i>User</i> memilih tombol "home"		
	14. Menuju menu utama	Sesuai

Pengujian *black box* ayo belajar rambu dilakukan dengan menjalankan beberapa submenu yang ada pada halaman ayo belajar. Reaksi sistem setelah *user* memilih menu ayo belajar adalah menampilkan lima submenu. Reaksi sistem ketika *user* memilih masing-masing submenu adalah sudah sesuai dengan fungsionalitas dari masing-masing submenu tersebut. Halaman ayo belajar juga terdapat tombol "*home*", dan jika *user* memilih tombol ini, maka reaksi sistem adalah menampilkan menu utama. Pengujian pada bagian ini sudah sesuai harapan yaitu reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 56. Uji *Black Box* Pengertian Rambu

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "pengertian rambu"		
	2. Menampilkan halaman definisi rambu lalu lintas	Sesuai
3. <i>User</i> memilih tombol "down"		
	4. Menampilkan halaman macam-macam rambu lalu lintas	Sesuai
5. <i>User</i> memilih tombol "close"		
	6. Menuju halaman "ayo belajar"	Sesuai

Pengujian *black box* pengertian rambu dilakukan dengan menjalankan submenu pengertian rambu. Reaksi sistem adalah menampilkan definisi rambu. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan input dari aktor.

Tabel 57. Uji *Black Box* Rambu Peringatan

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "rambu peringatan"		
	2. Menampilkan halaman definisi rambu peringatan	Sesuai
3. <i>User</i> memilih tombol "right"		
	4. Menampilkan halaman contoh-contoh rambu peringatan	Sesuai
5. <i>User</i> memilih tombol "close"		
	6. Menuju halaman "ayo belajar"	Sesuai

Pengujian *black box* rambu peringatan dilakukan dengan menjalankan submenu rambu peringatan. Reaksi sistem adalah menampilkan definisi rambu peringatan. Reaksi sistem setelah *user* menekan tombol "right" adalah sistem menampilkan contoh rambu peringatan. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* dari aktor.

Tabel 58. Uji *Black Box* Rambu Larangan

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "rambu larangan"		
	2. Menuju halaman rambu larangan 3. Menampilkan halaman definisi rambu larangan	Sesuai
4. <i>User</i> memilih tombol "right"		
	5. Menampilkan halaman contoh-contoh rambu larangan	Sesuai
6. <i>User</i> memilih tombol "close"		
	7. Menuju halaman "ayo belajar"	Sesuai

Pengujian *black box* rambu larangan dilakukan dengan menjalankan submenu rambu larangan. Reaksi sistem adalah menampilkan definisi rambu larangan, dan ketika *user* menekan tombol "right", sistem menampilkan contoh rambu larangan. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 59. Uji *Black Box* Rambu Perintah

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "rambu perintah"		
	2. Menampilkan halaman definisi rambu perintah	Sesuai
3. <i>User</i> memilih tombol "right"		
	4. Menampilkan halaman contoh-contoh rambu perintah	Sesuai
5. <i>User</i> memilih tombol "close"		
	6. Menuju halaman "ayo belajar"	Sesuai

Pengujian *black box* rambu perintah dilakukan dengan menjalankan submenu rambu perintah. Reaksi sistem adalah menampilkan definisi rambu perintah. Reaksi sistem ketika *user* menekan tombol "right" adalah sistem menampilkan contoh rambu perintah. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 60. Uji *Black Box* Rambu Petunjuk

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "rambu petunjuk"		
	2. Menampilkan halaman definisi rambu petunjuk	Sesuai
3. <i>User</i> memilih tombol "right"		
	4. Menampilkan halaman contoh-contoh rambu petunjuk	Sesuai
5. <i>User</i> memilih tombol "close"		
	6. Menuju halaman "ayo belajar"	Sesuai

Pengujian *black box* rambu petunjuk dilakukan dengan menjalankan submenu rambu petunjuk. Reaksi sistem adalah menampilkan definisi rambu petunjuk, dan ketika *user* menekan tombol "right", sistem menampilkan contoh rambu petunjuk. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* yang diberikan oleh aktor.

Tabel 61. Uji *Black Box* Bantuan

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> memilih menu "bantuan"		
	2. Menampilkan informasi pengertian aplikasi "Need For safety"	Sesuai
3. <i>User</i> memilih tombol "down"		
	4. Menampilkan teks bantuan cara memainkan permainan	Sesuai
5. <i>User</i> memilih tombol "down" kedua		
	6. Menampilkan informasi petunjuk penggunaan menu dan tombol	Sesuai
7. <i>User</i> memilih tombol "up"		
	8. Menampilkan informasi pengertian aplikasi "Need For safety"	Sesuai
9. <i>User</i> memilih tombol "home"		
	10. Menuju halaman utama	Sesuai

Pengujian *black box* bantuan dilakukan dengan menjalankan menu bantuan. Reaksi sistem adalah menampilkan informasi pengertian aplikasi. Reaksi sistem ketika *user* memilih tombol "down" adalah menampilkan informasi cara memainkan permainan, dan reaksi sistem ketika *user* memilih tombol "down"

yang kedua adalah sistem akan menampilkan informasi pengertian menu dan tombol dalam aplikasi. Hasil pengujian ini sudah sesuai dengan harapan karena reaksi sistem sudah sesuai dengan *input* yang diberikan aktor.

Tabel 62. Uji *Black Box* Keluar Aplikasi

Aksi Aktor	Reaksi Sistem	Hasil
Skenario Normal		
1. <i>User</i> keluar dari aplikasi		
	2. Menghentikan fungsi yang sedang berjalan	Sesuai

Pengujian *black box* keluar aplikasi dilakukan dengan cara keluar aplikasi menggunakan tombol "*back*" pada *smartphone* android. reaksi sistem adalah menghentikan semua fungsi dan keluar dari aplikasi. Hasil pengujian pada bagian ini sudah sesuai dengan harapan yaitu reaksi sistem sudah sesuai dengan *input* yang diberikan aktor.

Hasil pengujian *black box* secara keseluruhan adalah sistem yang dikembangkan memiliki fungsionalitas yang baik. Fitur-fitur yang ada pada aplikasi sudah dilakukan pengujian dan mendapatkan hasil yaitu sudah sesuai dengan fungsinya masing-masing. Pengujian selanjutnya adalah pengujian *portability* perangkat lunak.

c. Pengujian Perangkat Lunak Aspek *Portability*

Pengujian perangkat lunak aspek *portability* menurut Pressman (2010:489) merupakan pengujian untuk mengetahui kemudahan tentang bagaimana perangkat lunak dapat dipindahkan dari suatu lingkungan operasional ke lingkungan operasional lainnya. Pengujian perangkat lunak aspek *portability* ini dilakukan dengan menjalankan perangkat lunak pada emulator android (*Android*

Virtual Device) dan beberapa *smartphone* dengan sistem operasi android yang berbeda. Hasil pengujian aplikasi dengan menggunakan emulator (*Android Virtual Device*) dapat dilihat pada Tabel 63 berikut:

Tabel 63. Hasil Pengujian Aplikasi Menggunakan *Android Virtual Device*

No	Nama	OS Versi	Ukuran Layar	Hasil
1	PortabilityTest1	Android 2.3.3 (Gingerbread)	240 x 320 pixels	Aplikasi berjalan dengan baik (dokumentasi pada Gambar 54)
2	PortabilityTest2	Android 3.0 (Honeycomb)	240 x 400 pixels	Aplikasi berjalan dengan baik (dokumentasi pada Gambar 55)
3	PortabilityTest3	Android 4.0.3 (Ice Cream)	480 x 800 pixels	Aplikasi berjalan dengan baik (dokumentasi pada Gambar 56)
4	PortabilityTest4	Android 4.2.2 (Jelly Bean)	480 x 854 pixels	Aplikasi berjalan dengan baik (dokumentasi pada Gambar 57)

Pengujian *portability* dilakukan menggunakan *Android Virtual Device* dengan beberapa macam pengaturan yaitu pada jenis OS android dan ukuran layar untuk menampilkan aplikasi. Pengujian ini mendapatkan hasil bahwa aplikasi dapat berjalan dengan baik pada jenis OS yang berbeda mulai dari OS Android Gingerbread hingga Jelly Bean. Aplikasi juga dapat berjalan dengan baik pada beberapa ukuran layar yang berbeda.

Hasil pengujian aplikasi pada beberapa *smartphone* dengan sistem operasi android dapat dilihat pada Tabel 64 berikut:

Tabel 64. Hasil Pengujian Aplikasi pada *Smartphone* Android

No	Jenis	OS Versi	Ukuran Layar	Hasil
1	<i>Smartphone</i> Samsung Galaxy Young S5360	Android 2.3.3 (Gingerbread)	240 x 320 <i>pixels, inches</i> 3.0	Aplikasi berhasil diinstall dan dapat berjalan dengan baik (dokumentasi pada Gambar 58 dan 59)
2	<i>Smartphone</i> IMO s89	Android 4.2.2 (Jelly Bean)	540 x 960 <i>pixels, inches</i> 4.5	Aplikasi berhasil diinstall dan dapat berjalan dengan baik (dokumentasi pada Gambar 60 dan 61)
3	<i>Smartphone</i> Cross Andromeda a88	Android 4.2.2 (Jelly Bean)	450 x 854 <i>pixels, inches</i> 4.5	Aplikasi berhasil diinstall dan dapat berjalan dengan baik (dokumentasi pada Gambar 62)
4	Tablet Advan T1H	Android 4.1.2 (Jelly Bean)	1024 x 600 <i>pixels, inches</i> 7.0	Aplikasi berhasil diinstall dan dapat berjalan dengan baik (dokumentasi pada Gambar 63)

Pengujian *portablity* selanjutnya adalah menggunakan *smartphone* android. *Smartphone* android yang digunakan adalah android Gingerbread dan Jelly Bean. Ukuran layar yang digunakan dalam pengujian ini adalah *smartphone* layar 3 inchi, *smartphone* layar 4,5 inchi, dan tablet layar 7 inchi. Pengujian ini mendapatkan hasil bahwa aplikasi dapat diinstal dengan mudah dan aplikasi dapat berjalan dengan baik.

Dokumentasi pengujian aplikasi pada *Android Virtual Device* untuk Android OS 2.3.3 dan ukuran layar 240x320 *pixels* dapat dilihat pada Gambar 54.



Gambar 54. Hasil Pengujian Aplikasi pada AVD (OS 2.3.3 dengan ukuran layar 240x320)

Dokumentasi pengujian aplikasi pada *Android Virtual Device* untuk Android OS 3.0 dan ukuran layar 240x400 *pixels* dapat dilihat pada Gambar 55 di bawah ini:



Gambar 55. Hasil Pengujian Aplikasi pada AVD (OS 3.0 dengan ukuran layar 240x400)

Dokumentasi pengujian aplikasi pada *Android Virtual Device* untuk Android OS 4.0.3 dan ukuran layar 480x800 *pixels* dapat dilihat pada Gambar 56 di bawah ini:



Gambar 56. Hasil Pengujian Aplikasi pada AVD (OS 4.0.3 dengan ukuran layar 480x800)

Dokumentasi pengujian aplikasi pada *Android Virtual Device* untuk Android OS 4.2.2 dan ukuran layar 480x854 *pixels* dapat dilihat pada Gambar 57 di bawah ini:



Gambar 57. Hasil Pengujian Aplikasi pada AVD (OS 4.2.2 dengan ukuran layar 480x854)

Dokumentasi pengujian aplikasi pada *smartphone* Samsung Galaxy Young S5360 dapat dilihat pada gambar di bawah ini:

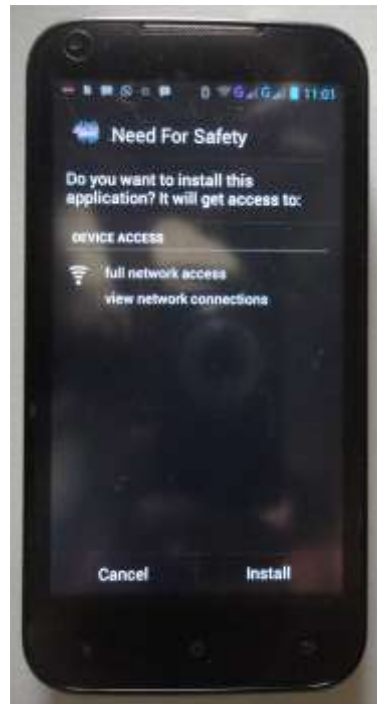


Gambar 58. Instalasi Aplikasi pada Smartphone *Samsung* Galaxy Young S5360



Gambar 59. Instalasi Berhasil pada *Smartphone* Samsung Galaxy Young S5360

Dokumentasi pengujian aplikasi pada *smartphone* IMO S89 dapat dilihat pada gambar di bawah ini:



Gambar 60. Instalasi Aplikasi pada *Smartphone* IMO S89



Gambar 61. Instalasi Berhasil pada *Smartphone* IMO S89

Dokumentasi pengujian aplikasi pada *smartphone* Cross Andromeda a88 dapat dilihat pada gambar di bawah ini:



Gambar 62. Instalasi Aplikasi pada *Smartphone* Cross a88

Dokumentasi pengujian aplikasi pada tablet Advan T1H dapat dilihat pada Gambar 63 di bawah ini:



Gambar 63. Instalasi Aplikasi pada Tablet Advan T1H

Hasil pengujian *portability* aplikasi secara umum ditampilkan dalam Tabel 65 di bawah ini:

Tabel 65. Hasil Uji Variabel *Portability* Perangkat Lunak

Indikator	Sub Indikator	Hasil yang Diharapkan	Ketercapaian		Ket.
			Ya	Tidak	
<i>Installability</i> (Kemampuan untuk Diinstal)	Kemudahan instalasi aplikasi	Aplikasi " <i>Need For Safety</i> " dapat diinstal dengan mudah di <i>platform</i> Android	√		Berjalan baik tanpa ada pesan kesalahan
	Keberhasilan dalam instalasi	Aplikasi " <i>Need For Safety</i> " berhasil diinstal di <i>platform</i> Android	√		Berjalan baik tanpa ada pesan kesalahan
<i>Adaptability</i> (Kemampuan untuk Beradaptasi)	Instalasi di beberapa versi android	Aplikasi " <i>Need For Safety</i> " dapat diinstal di beberapa versi Android yang berbeda	√		Berjalan baik tanpa ada pesan kesalahan
	Instalasi di beberapa platform android dengan ukuran layar berbeda	Aplikasi " <i>Need For Safety</i> " berhasil diinstal di platform android dengan ukuran layar berbeda	√		Berjalan baik tanpa ada pesan kesalahan

Tahapan pengujian *portability* secara keseluruhan yang telah dilakukan peneliti mendapatkan hasil berupa aplikasi dapat diinstall dengan mudah pada *platform* android dan dapat berjalan di beberapa versi sistem operasi android dan ukuran layar yang berbeda. Kesimpulan dari hasil pengujian aspek *portability* adalah perangkat lunak "*Need For Safety*" memiliki portabilitas yang baik. Pengujian selanjutnya adalah pengujian *alpha*.

d. *Alpha Testing*

Pengujian *alpha* merupakan tahapan pengujian untuk memvalidasi produk yang dilakukan oleh para ahli. Pengujian *alpha* yang dilakukan oleh ahli akan mengoreksi kesalahan-kesalahan dan kekurangan yang ada dalam produk. Ahli juga memberikan saran dan komentar serta rekomendasi untuk memperbaiki produk yang sedang dikembangkan agar menjadi lebih baik. Proses pengujian perangkat lunak ini dilakukan dengan melibatkan dua ahli dalam bidang rekayasa perangkat lunak. Hasil dari pengujian *alpha* adalah sebagai berikut:

Tabel 66. Uji Alpha Spesifikasi Uji Membuka Aplikasi

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Tampil halaman " <i>Title Screen</i> "	√	
2	<i>User</i> memilih tombol "mulai"	Tampil halaman menu <i>game</i> (menu skor, menu bantuan, menu mulai bermain, menu ayo belajar, dan menu <i>On/Off</i> musik)	√	

Pengujian *alpha* membuka aplikasi yang dilakukan oleh dua ahli mendapatkan hasil bahwa aplikasi dapat berjalan dengan baik. *Input* yang diberikan oleh penguji menghasilkan *output* sistem yang sesuai yaitu pada saat membuka aplikasi, sistem menampilkan halaman judul atau "*title screen*", kemudian penguji menekan tombol "mulai" dan sistem menampilkan halaman menu utama aplikasi. Hasil uji *alpha* pada bagian membuka aplikasi adalah aplikasi memiliki fungsi yang sudah sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya.

Tabel 67. Spesifikasi Uji Menu *Game*

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Menu <i>Game</i>	Menu <i>game</i> dapat terbuka setelah <i>user</i> menekan tombol "mulai" pada halaman "Title Screen"	√	
		<i>User</i> dapat masuk ke semua menu dengan memilih tombol menu-menu yang ada	√	
		<i>User</i> dapat menonaktifkan musik <i>game</i> dengan memilih tombol "off musik" pada menu awal	√	
		<i>User</i> dapat mengaktifkan musik setelah musik mati dengan memilih tombol "on musik" pada menu awal	√	
		<i>User</i> dapat kembali ke menu awal dengan memilih tombol "home"	√	
		<i>User</i> dapat keluar aplikasi dengan memilih tombol "back" pada <i>handphone</i>	√	

Pengujian *alpha* bagian menu *game* dilakukan oleh ahli dengan membuka menu utama aplikasi dan menjalankan menu-menu yang ada, serta menggunakan tombol-tombol untuk kembali ke menu utama. Hasil dari uji *alpha* bagian ini adalah menu-menu yang ada pada menu utama sudah memiliki fungsi yang sesuai dengan kebutuhan yang telah ditetapkan sebelumnya.

Tabel 68. Spesifikasi Uji Menu Mulai Bermain

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Mulai Bermain	<i>User</i> dapat masuk ke menu "mulai bermain"	√	
		Muncul permainan level 1	√	
		Setelah <i>user</i> menyelesaikan level 1 langsung menuju level 2	√	
		Setelah <i>user</i> menyelesaikan level 2 langsung menuju level 3	√	
		Setelah <i>user</i> menyelesaikan level 3 langsung menuju level 4	√	
		Setelah <i>user</i> menyelesaikan level 4 langsung muncul informasi nilai	√	
2	Tampil skor nilai	<i>User</i> dapat menuju halaman skor setelah memilih tombol "skor"	√	
		<i>User</i> dapat memainkan permainan kembali dari level 1 setelah memilih tombol "left/arah kiri"	√	

Pengujian *alpha* bagian menu mulai bermain dilakukan ahli dengan membuka menu permainan. Sistem akan menampilkan permainan level 1 sampai level 4, dan setelah selesai permainan level 4, sistem akan menampilkan informasi nilai. Halaman informasi nilai terdapat tombol "skor" untuk menampilkan skor yang sudah disimpan dan juga tombol "*left*" untuk memainkan kembali permainan. Hasil pengujian *alpha* bagian ini adalah menu mulai bermain sudah berjalan sesuai dengan fungsinya yaitu sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya.

Tabel 69. Spesifikasi Uji Menu Ayo Belajar

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Ayo Belajar	<i>User</i> dapat masuk ke menu "ayo belajar"	√	
		Tampil submenu ayo belajar	√	
2	Pengertian Rambu	<i>User</i> dapat masuk ke halaman pengertian rambu setelah memilih submenu pengertian rambu	√	
3	Rambu Peringatan	<i>User</i> dapat masuk ke halaman pengertian rambu peringatan setelah memilih submenu rambu peringatan	√	
		Menuju halaman contoh rambu peringatan setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	√	
		Pada halaman contoh rambu peringatan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	√	
5	Rambu Larangan	<i>User</i> dapat masuk ke halaman pengertian rambu larangan setelah memilih submenu rambu larangan	√	
		Menuju halaman contoh rambu larangan setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	√	
		Pada halaman contoh rambu larangan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	√	
6	Rambu Perintah	<i>User</i> dapat masuk ke halaman pengertian rambu perintah setelah memilih submenu rambu perintah	√	
		Menuju halaman contoh rambu perintah setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	√	

Tabel 70. Spesifikasi Uji Menu Ayo Belajar (lanjutan)

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
		Pada halaman contoh rambu perintah, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	√	
7	Rambu Petunjuk	<i>User</i> dapat masuk ke halaman pengertian rambu petunjuk setelah memilih submenu rambu petunjuk	√	
		Menuju halaman contoh rambu petunjuk setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	√	
		Pada halaman contoh rambu petunjuk, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	√	

Pengujian *alpha* bagian menu ayo belajar dilakukan ahli dengan cara menjalankan menu ayo belajar dan menjalankan submenu-submenu yang ada pada menu ayo belajar. Penguji menjalankan masing-masing submenu yaitu menampilkan halaman pengertian rambu dan contoh rambu. Hasil dari uji *alpha* pada bagian ini adalah menu ayo belajar sudah berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya.

Hasil pengujian *alpha* secara keseluruhan berdasarkan Tabel 66-70 adalah penguji menyimpulkan bahwa fungsi-fungsi yang ada pada perangkat lunak *Need For Safety* sudah berjalan sesuai dengan spesifikasi yang telah ditetapkan pada tahap analisis kebutuhan sebelumnya.

Uji *alpha* yang telah dilakukan penguji selain menggunakan tabel spesifikasi juga menggunakan tabel pengujian kelayakan perangkat lunak. Data hasil uji *alpha* menggunakan tabel pengujian kelayakan aspek *functionality*, *efficiency*, dan *usability* perangkat lunak menurut ahli dapat dilihat pada Tabel 71 berikut:

Tabel 71. Data Hasil Uji *Alpha* Aspek *Functionality*, *Efficiency*, dan *Usability*

Ahli	Pertanyaan															
	<i>Functionality</i>						<i>Efficiency</i>		<i>Usability</i>							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	5	5	5	5	5	5	5	5	4	5	5	5	5	5	4	4
2	4	4	4	4	4	4	2	4	4	4	4	4	4	4	4	4

Hasil uji *alpha* pada Tabel 71 berupa data untuk masing-masing aspek kelayakan akan dibahas selanjutnya pada bagian pembahasan. Ahli rekayasa perangkat lunak yang melakukan pengujian *alpha* juga memberikan saran untuk memperbaiki bagian-bagian tertentu aplikasi. Bagian-bagian aplikasi yang perlu diperbaiki dijelaskan pada Tabel 72.

Tabel 72. Saran Ahli Pengujian *Alpha*

No	Bagian yang perlu diperbaiki	Saran
1	Bagian "Title Screen"	Dituliskan identitas pengembang aplikasi (nama, nim, prodi)
2	Halaman Pengertian Rambu	Dibuat tombol "back" untuk menuju ke halaman sebelumnya
3	Bagian Submenu Ayo Belajar	Tampilan menu dibuat lebih mencerminkan masing-masing materi
4	Bagian Menu Utama	Animasi menu utama untuk lebih dipercepat

Saran dari penguji kemudian menjadi bahan untuk melakukan revisi tahap uji *alpha*. Revisi yang dilakukan pada tahap ini sesuai dengan saran-saran dan

rekomendasi penguji ahli untuk memperbaiki produk yang sedang dikembangkan agar menjadi lebih baik. Pengujian selanjutnya setelah selesai uji *alpha* adalah pengujian *beta*.

e. Beta Testing

Tahap uji *beta* dilakukan dengan melibatkan 25 pengguna aplikasi yang telah ditentukan sebelumnya. Pengguna aplikasi tersebut akan menguji perangkat lunak untuk mengetahui kualitas perangkat lunak pada aspek *functionality*, *efficiency*, *usability* dan *portability*. Tahapan pengujian *beta* menggunakan bantuan kuesioner yang berisi butir-butir pertanyaan dari setiap faktor kualitas perangkat lunak yang menjadi fokus penelitian.

Kuesioner pada uji *beta* dibuat berdasarkan sub-karakteristik dari faktor kualitas perangkat lunak menurut ISO 9126 yaitu faktor *functionality*, *efficiency*, *usability* dan *portability*. Kuesioner uji *beta* ini sebelum digunakan telah lolos dalam validasi angket yang dilakukan oleh tiga orang ahli *validator*.

Pengujian *beta* perangkat lunak dilakukan di SDN Pujokusuman 1 dengan pengguna yang telah ditentukan adalah guru sebanyak 20 orang. Uji *beta* juga dilakukan oleh polisi bagian lalu lintas Polsek Bulaksumur sebanyak 5 orang. Data hasil pengujian *beta* dapat dilihat pada Tabel 73.

Tabel 73. Data Hasil Pengujian *Beta*

Pengguna	Pertanyaan																	
	<i>Functionality</i>						<i>Efficiency</i>		<i>Usability</i>								<i>Portability</i>	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	4	4
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
3	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4	5	5	5	5	5	5	4	5	4	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5
6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5
7	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
8	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
9	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
10	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5
11	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4
12	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5
13	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
14	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5
15	5	5	5	5	5	4	4	4	5	5	5	5	5	4	4	4	5	5
16	5	5	5	5	5	5	5	5	5	5	5	4	5	4	5	5	4	5
17	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
18	5	5	5	5	5	4	4	4	5	5	5	5	5	4	4	4	5	5
19	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
20	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
21	5	5	5	4	5	5	4	5	4	4	4	5	5	5	5	5	4	5
22	5	5	5	5	3	3	4	4	4	3	4	3	4	4	3	4	3	4
23	4	4	4	4	4	3	4	4	3	4	4	3	4	3	4	4	3	4
24	5	5	4	4	4	4	4	3	3	4	4	4	5	5	5	4	4	4
25	4	3	5	4	3	5	4	3	5	4	5	4	5	5	3	5	4	5

Hasil uji *alpha* pada Tabel 73 berupa data untuk masing-masing aspek kelayakan yaitu aspek *functionality*, *efficiency*, *usability* dan *portability* akan dibahas selanjutnya pada bagian pembahasan.

Pengguna terpilih yang melakukan pengujian *beta* juga memberikan beberapa saran dalam pengembangan perangkat lunak. Saran-saran pengguna terpilih dalam tahap pengujian *beta* adalah sebagai berikut:

- 1) Penambahan level permainan mencocokkan gambar sehingga permainan lebih menantang.
- 2) Menu permainan lebih diperjelas sehingga pengguna awam lebih mudah dalam menggunakan perangkat lunak.
- 3) Bentuk permainan ditambah tidak hanya mencocokkan gambar, misalnya permainan mengendarai motor dan mengendarai mobil sehingga lebih variatif.

B. Pembahasan

1. *Alpha Testing*

Pengujian *alpha* menurut Pressman (2010:570) merupakan pengujian yang dilakukan oleh sekelompok perwakilan dari pengguna akhir. Pengujian *alpha* dalam penelitian ini dilakukan oleh ahli dalam bidang rekayasa perangkat lunak dengan tujuan untuk mendapatkan unjuk kerja dari perangkat lunak yang dikembangkan.

Penilaian unjuk kerja perangkat lunak pada pengujian *alpha* meliputi 5 bagian, yaitu (1) bagian membuka aplikasi, (2) bagian menu *game*, (3) bagian menu mulai bermain, (4) bagian menu ayo belajar, dan (5) pengujian perangkat lunak menurut aspek *functionality*, *efficiency*, dan *usability*. Hasil pengujian *alpha* menjadi bahan pertimbangan dan masukan yang berharga bagi peneliti untuk

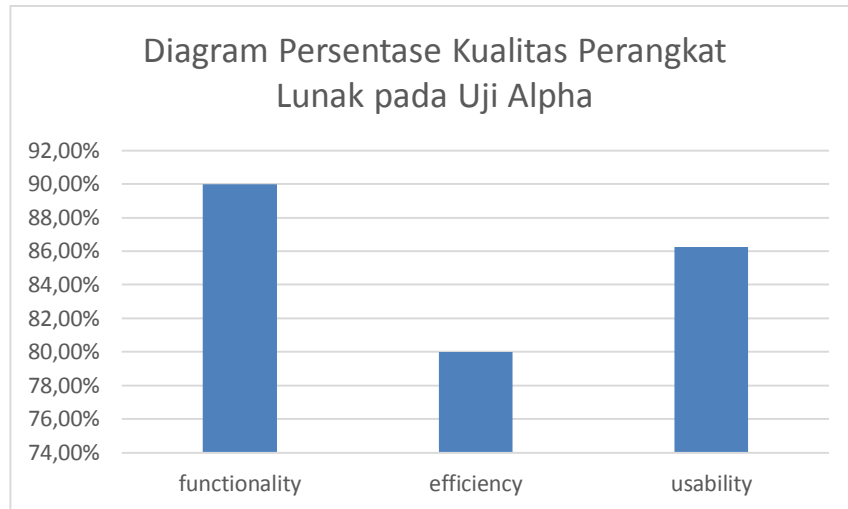
memperbaiki kekurangan-kekurangan yang ada dalam tahap pengembangan perangkat lunak ini.

Hasil pengujian *alpha* menunjukkan bahwa menu-menu yang terdapat dalam aplikasi sudah memiliki fungsi yang sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya. Hasil tersebut dibuktikan dengan tabel pengujian *alpha* yang mendapatkan hasil pengujian yang sudah sesuai dan mendapatkan kesimpulan perangkat lunak dapat bekerja dengan baik. Data hasil pengujian *alpha* untuk aspek *functionality*, *efficiency*, dan *usability* menurut ahli dalam bidang rekayasa perangkat lunak yang ditampilkan pada Tabel 71, kemudian diolah untuk mendapatkan hasil berupa tingkat kelayakan perangkat lunak aspek *functionality*, *efficiency*, dan *usability* menurut ahli. Hasil pengolahan data pada uji *alpha* adalah sebagai berikut:

Tabel 74. Perhitungan Data Hasil Uji *Alpha*

Pengguna	<i>Functionality</i>	<i>Efficiency</i>	<i>Usability</i>
1	30	10	37
2	24	6	32
Total Skor	54	16	69
Skor Maksimum	60	20	80
Rerata Item	4,50	4,00	4,31
Persentase	90,00%	80,00%	86,25%

Perhitungan persentase faktor kualitas perangkat lunak kemudian digambarkan dalam bentuk diagram. Diagram persentase faktor kualitas perangkat lunak "*Need For Safety*" pada uji *alpha* dapat dilihat pada Gambar 64.



Gambar 64. Diagram Persentase Faktor Kualitas Perangkat Lunak Pada Uji *Alpha*

Hasil perhitungan untuk mengetahui kualitas perangkat lunak dari segi *functionality*, *efficiency*, dan *usability* kemudian dibandingkan dengan tabel Interpretasi Persentase Likert.

Tabel 75. Interpretasi Persentase Likert untuk Kualitas Perangkat Lunak

No	Persentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

Tingkat kelayakan masing-masing faktor kualitas perangkat lunak setelah membandingkan dengan tabel interpretasi persentase kelayakan perangkat lunak di atas adalah sebagai berikut:

Tabel 76. Persentase Kelayakan Tiap Faktor Pada Uji *Alpha*

No	Faktor	Persentase	Tingkat Kelayakan
1	<i>Functionality</i>	90,00%	Sangat Layak
2	<i>Efficiency</i>	80,00%	Layak
3	<i>Usability</i>	86,25%	Sangat Layak

Perhitungan persentase kualitas perangkat lunak secara keseluruhan dari data yang sudah didapatkan pada pengujian *alpha* menurut ahli adalah menggunakan rumus:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = \frac{139}{160} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = 86,88\%$$

Persentase kelayakan perangkat lunak secara keseluruhan pada uji *alpha* adalah 86,88%. Hasil persentase tersebut kemudian dibandingkan dengan tabel Interpretasi Persentase Likert. Hasil yang didapatkan setelah membandingkan persentase kelayakan perangkat lunak adalah menyatakan bahwa perangkat lunak masuk dalam kategori "Sangat Layak".

a. Revisi *Uji Alpha*

Revisi pada tahap uji *alpha* terhadap perangkat lunak yang dikembangkan adalah sesuai dengan saran dan rekomendasi ahli yang menjadi penguji *alpha*. Revisi yang pertama adalah untuk menambahkan identitas pengembang pada bagian "*title screen*". Identitas yang ditambahkan adalah nama pengembang, nim pengembang, dan program studi pengembang perangkat lunak. Identitas pengembang ditampilkan pada "*title screen*" yang tampil setelah aplikasi mulai dijalankan.

Revisi kedua adalah menambahkan tombol "*back*" pada halaman materi pertama. Tombol ini berfungsi untuk kembali ke halaman selanjutnya karena pada materi yang pertama terdiri dari dua halaman. Halaman yang pertama

adalah halaman pengertian rambu lalu lintas dan halaman kedua adalah halaman jenis rambu lalu lintas.

Revisi ketiga yang dilakukan peneliti adalah merubah tampilan submenu "ayo belajar" dengan mengubah warna dan menambahkan ikon rambu untuk masing-masing materi jenis rambu. Warna untuk submenu rambu peringatan diubah menjadi warna kuning, rambu larangan diubah menjadi warna merah, rambu perintah diubah menjadi warna biru, dan rambu petunjuk diubah menjadi warna hijau.

Revisi keempat adalah untuk mempercepat animasi pada halaman menu utama aplikasi. Animasi tampil menu aplikasi yang semula 2 detik, dipercepat menjadi 1,5 detik dengan selang waktu jeda antar menu untuk tampil 0,5 detik. Perubahan pada revisi ini adalah untuk membuat pengguna tidak jenuh menunggu waktu tampil menu utama setiap kali pengguna masuk ke menu utama.

2. *Beta Testing*

Pengujian *beta* merupakan tahapan pengujian setelah pengujian *alpha* selesai dilakukan. Pengujian *beta* dilakukan untuk mendapatkan usulan dari pengguna akhir sebelum persiapan rilis produk akhir (Pressman, 2012:570). Pengujian *beta* dilakukan oleh 25 pengguna yang terdiri dari guru-guru di SDN Pujokusuman 1 yang merupakan SD Percontohan Lalu Lintas di DIY, dan beberapa polisi bagian lalu lintas di Polsek Bulaksumur.

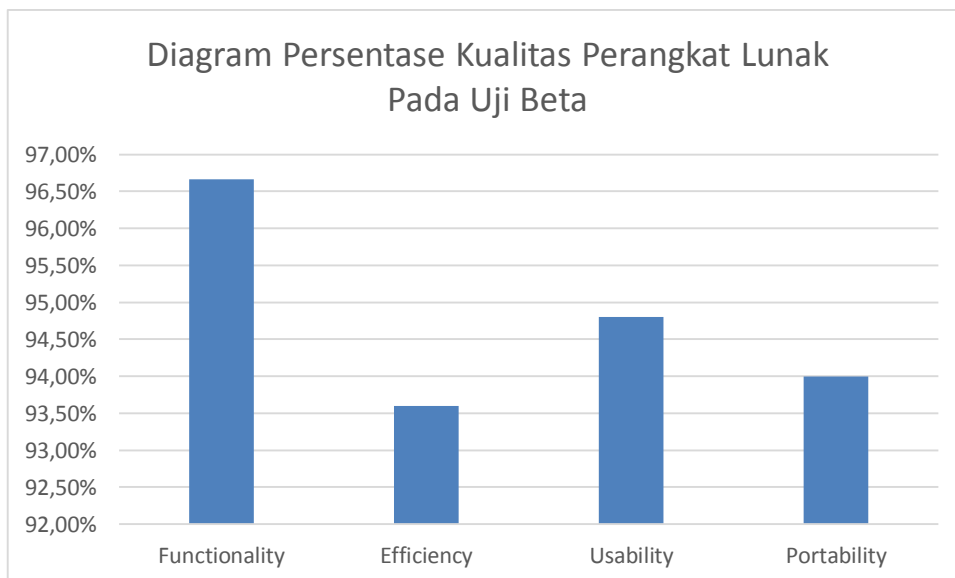
Pengujian *beta* ini dilakukan untuk menguji kualitas perangkat lunak dari aspek *functionality*, *efficiency*, *usability*, dan *portability* berdasarkan software

quality factor ISO 9126. Pengujian ini menggunakan bantuan kuesioner yang berisi pertanyaan yang sesuai dengan aspek-aspek yang menjadi fokus penelitian. Data hasil pengujian *beta* yang telah dikumpulkan kemudian diolah. Hasil pengolahan data pada pengujian *beta* adalah sebagai berikut:

Tabel 77. Perhitungan Data Hasil Uji *Beta*

Pengguna	<i>Functionality</i>	<i>Efficiency</i>	<i>Usability</i>	<i>Portability</i>
1	30	10	39	8
2	30	10	40	10
3	29	10	40	10
4	30	9	38	10
5	30	10	39	10
6	30	10	39	10
7	30	10	40	10
8	30	10	40	10
9	30	10	40	10
10	30	10	39	10
11	30	10	39	8
12	30	10	39	10
13	30	10	40	10
14	30	10	39	10
15	29	8	37	10
16	30	10	38	9
17	30	10	40	10
18	29	8	37	10
19	30	10	40	10
20	30	10	40	10
21	29	9	37	9
22	26	8	29	7
23	23	8	29	7
24	26	7	34	8
25	24	7	36	9
Total Skor	725	234	948	235
Skor Maksimum	750	250	1000	250
Rerata Item	4,83	4,68	4,74	4,70
Persentase	96,67%	93,60%	94,80%	94,00%

Perhitungan persentase faktor kualitas perangkat lunak kemudian digambarkan dalam bentuk diagram. Diagram persentase faktor kualitas perangkat lunak pada uji *beta* dapat dilihat pada Gambar 65.



Gambar 65. Diagram Persentase Faktor Kualitas Perangkat Lunak Pada Uji *Beta*

Hasil perhitungan untuk mengetahui kualitas perangkat lunak dari segi *functionality*, *efficiency*, *usability*, dan *portability* kemudian dibandingkan dengan Interpretasi Persentase Likert pada Tabel 75. Tingkat kelayakan masing-masing faktor kualitas perangkat lunak setelah membandingkan dengan tabel interpretasi persentase kelayakan perangkat lunak adalah sebagai berikut:

Tabel 78. Persentase Kelayakan Tiap Faktor Pada Uji *Beta*

No	Faktor	Persentase	Tingkat Kelayakan
1	<i>Functionality</i>	96,67%	Sangat Layak
2	<i>Efficiency</i>	93,60%	Sangat Layak
3	<i>Usability</i>	94,80%	Sangat Layak
4	<i>Portability</i>	94,00%	Sangat Layak

Perhitungan persentase kualitas perangkat lunak secara keseluruhan dari data hasil pengujian *beta* menggunakan rumus:

$$\text{Persentase Kelayakan (\%)} = \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = \frac{2142}{2250} \times 100\%$$

$$\text{Persentase Kelayakan (\%)} = 95,20\%$$

Hasil pengolahan data pada uji *beta* di atas menyatakan bahwa perangkat lunak secara keseluruhan memiliki persentase kelayakan sebesar 95,20%. Hasil persentase tersebut kemudian dibandingkan dengan tabel Interpretasi Persentase Likert. Hasil yang didapatkan setelah membandingkan persentase kelayakan perangkat lunak adalah dapat disimpulkan bahwa kualitas perangkat lunak *Need For Safety* memiliki kategori "Sangat Layak".

a. Revisi Uji *Beta*

Saran dan masukan dari pengguna pada pengujian *beta* digunakan untuk memperbaiki perangkat lunak. Revisi perangkat lunak yang dilakukan adalah memperbaiki tampilan menu utama agar tidak membingungkan bagi pengguna awam. Revisi ini dilakukan untuk mempermudah pengguna awam untuk menggunakan menu-menu yang ada pada aplikasi ini.

C. Hasil Akhir Produk

Perangkat lunak "*Need For Safety*" sebagai pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun ini telah dikembangkan dan melalui berbagai tahapan yaitu mulai dari analisis kebutuhan, desain, implementasi, dan pengujian sesuai dengan model pengembangan *Linear Sequential*. Implementasi perangkat

lunak ini menggunakan aplikasi Construct 2 yang merupakan aplikasi *game builder* berbasis HTML5. Tahap pengujian perangkat lunak meliputi uji verifikasi *white box testing* dan *black box testing*, kemudian tahapan uji validasi perangkat lunak telah melalui uji validasi *alpha* dan *beta*.

Hasil akhir dari penelitian ini adalah terciptanya perangkat lunak "*Need For Safety*" sebagai pengenalan rambu-rambu lalu lintas untuk anak usia 6-12 tahun. Perangkat lunak ini memiliki menu-menu antara lain menu mulai bermain, menu ayo belajar, menu bantuan, dan menu skor. Perangkat lunak "*Need For Safety*" yang telah dikembangkan ini memiliki kelebihan dan kelemahan. Kelebihan dari perangkat lunak ini antara lain:

1. Perangkat lunak ini menyediakan materi pengenalan jenis rambu lalu lintas.
2. Perangkat lunak ini dapat berjalan pada beberapa versi android mulai dari android v2.3 (Gingerbread) sampai android v4.2.2 (Jelly Bean).
3. Perangkat lunak ini dapat berjalan baik di beberapa *platform* android dengan ukuran layar yang berbeda-beda.
4. Perangkat lunak ini memiliki tingkatan level untuk menu permainannya.
5. Perangkat lunak ini mampu menyimpan dan menampilkan skor yang telah didapatkan pengguna.

Kelemahan perangkat lunak ini antara lain adalah:

1. Perangkat lunak ini hanya memiliki 4 level permainan.
2. Perangkat lunak ini hanya memiliki satu jenis permainan yaitu permainan mencocokkan gambar rambu lalu lintas.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dalam pengembangan perangkat lunak "*Need For Safety*", maka peneliti mengambil beberapa kesimpulan, yaitu:

1. Perancangan perangkat lunak "*Need For Safety*" melalui 8 tahap pengembangan memberi kemudahan bagi pengembang karena keseluruhan tahapan disusun secara berurutan dimulai dari menetapkan kebutuhan sebagai acuan perancangan perangkat lunak hingga menjadi produk akhir.
2. Hasil unjuk kerja perangkat lunak "*Need For Safety*" masuk dalam kategori "Sangat Layak". Hasil pengujian unjuk kerja ini didukung oleh hasil pengujian *alpha* dan *beta* perangkat lunak. Nilai persentase hasil pengujian *alpha* perangkat lunak untuk setiap faktor, yaitu: *functionality* sebesar 90,00%, *efficiency* sebesar 80,00%, dan *usability* sebesar 86,25%. Persentase total dari pengujian *alpha* untuk kualitas perangkat lunak adalah 86,88% (Sangat Layak). Nilai persentase hasil pengujian *beta* perangkat lunak untuk setiap faktor, yaitu *functionality* sebesar 96,67%, *efficiency* sebesar 93,60%, *usability* sebesar 94,80%, dan *portability* sebesar 94,00%. Persentase total dari pengujian *beta* untuk kualitas perangkat lunak adalah 95,20% (Sangat Layak).

B. Saran

Penelitian ini tentunya masih terdapat banyak kekurangan dan hal-hal yang masih perlu dikaji dan dikembangkan kembali. Peneliti memiliki beberapa pemikiran dan saran untuk pengembangan selanjutnya, yaitu:

1. Pengembangan aplikasi untuk level permainan dan jenis permainan yang lebih beragam dan menantang.
2. Penambahan referensi untuk materi yang disajikan sehingga menjadi lebih lengkap.

DAFTAR PUSTAKA

- Adi Nugroho. (2009). *Rekayasa Perangkat Lunak Menggunakan UML dan JAVA*. Yogyakarta : ANDI.
- Andang Ismail. (2007). *Educational Games*. Yogyakarta : Pilar Media.
- Anggra. (2008). *Memahami Teknik Dasar Pembuatan Game Berbasis Flash*. Yogyakarta : Gava Media.
- Ariyadi Wijaya. (2009). *Manfaat Permainan Tradisional untuk PMRI*. Yogyakarta: Universitas Negeri Yogyakarta.
- Azhar Arsyad. (2006). *Media Pembelajaran*. Jakarta : PT RajaGrafindo Persada.
- BIN. (2013). *Kecelakaan Lalu Lintas Menjadi Pembunuh Terbesar Ketiga*. Diakses dari <http://www.bin.go.id/awas/detil/197/4/21/03/2013/kecelakaan-lalu-lintas-menjadi-pembunuh-terbesar-ketiga> pada tanggal 2 Desember 2013.
- Burhan Bungin. (2009). *Metodologi Penelitian Kuantitatif*. Jakarta : Kencana Prenada Media Group.
- Depdikbud. (1991). *Kamus Besar Bahasa Indonesia*. Jakarta : Balai Pustaka.
- Eko Budi Kristanto. (2013). *Kualitas Perangkat Lunak Model ISO 9126*. Diakses dari <http://fxekobudi.net/software-engineering/kualitas-perangkat-lunak-model-iso-9126/> pada tanggal 4 Maret 2014.
- Eva Handriyantini. (2009). *Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar*. Malang : Sekolah Tinggi Informasi & Komputer Indonesia.
- Gulo. (2002). *Metodologi Penelitian*. Jakarta : Grasindo.
- Hurd, Daniel & Jenuings, Erin. (2009). *Standardized Educational Games Rating: Suggested Criteria*. Karya Tulis Ilmiah.
- Ishak Abdulhak & Deni Darmawan. (2013). *Teknologi Pendidikan*. Bandung : PT Remaja Rosdakarya.
- Jazi Eko Istiyanto. (2013). *Pemrograman Smartphone Menggunakan SDK Android dan Hacking Android*. Yogyakarta: Graha Ilmu.
- Jogiyanto. (2008). *Metodologi Penelitian Sistem Informasi*. Yogyakarta : ANDI.
- Leksmono Suryo Putranto. (2008). *Rekayasa Lalu Lintas*. Indonesia : PT Macanan Jaya Cemerlang.

- Margono. (1997). *Metodologi Penelitian Pendidikan*. Jakarta: Rineka Cipta.
- McConnel, Steve. (2004). *Code Complete*. Redmond : Microsoft Press.
- Muhammad Huda. (2013). *Mudahnya Membuat Game HTML5 Dengan Construct2*. Diakses dari <http://mkhuda.com/html5/mudahnya-membuat-game-html5-dengan-construct-2/> pada tanggal 11 Maret 2014.
- Morrison, George. (2008). *Dasar-Dasar Pendidikan Anak Usia Dini (Edisi Kelima)*. (Alih bahasa: Suci Romadhona dan Apri Widiastuti). Jakarta : PT Indeks.
- Naik, Kshirasagar & Tripathy Priyadarshi. (2008). *Software Testing and Quality Assurance*. New Jersey : John Wiley & Sons, Inc.
- Nazrudin Safaat. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : Informatika.
- Panovski, Gregor. (2008). *Product Quality Software*. Master's Thesis. Eindhoven University of Technology.
- Pressman, Roger S. (2010). *Software Engineering : a Practitioner's Approach- Seventh Edition*. New York : McGraw-Hill.
- Pressman, Roger S. (2012). *Rekayasa Perangkat Lunak-Buku Satu, Pendekatan Praktisi (Edisi 7)*. (Alih bahasa: Adi Nugroho, George John L.N, Theresia Herlina R, Ike Kurniawati W). Yogyakarta : ANDI.
- Riduwan & Sunarto. (2012). *Pengantar Statistika untuk Penelitian Pendidikan, Sosial, Ekonomi, Komunikasi, dan Bisnis*. Bandung: Alfabeta.
- Romi Satria Wahono. (2006). *Teknik Pengukuran Kualitas Perangkat Lunak*. Diakses dari <http://romisatriawahono.net/2006/06/05/teknik-pengukuran-kualitas-perangkat-lunak-pada-tanggal-17-oktober-2013>.
- Rosa & Muhammad Salahuddin. (2011). *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur Dan Berorientasi Objek)*. Bandung : Modula.
- Rusman, Deni Kurniawan & Cepi Riyana. (2012). *Pembelajaran Berbasis Teknologi Informasi dan Komunikasi Mengembangkan Profesionalitas Guru*. Jakarta : PT RajaGrafindo Persada.
- Sandjaja & Albertus Herianto. (2006). *Panduan Penelitian*. Jakarta : Prestasi Pustakaraya.
- Sugiyono. (2010). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung : Alfabeta.
- _____. (2013). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung : Alfabeta.

- Suharsimi Arikunto. (2010). *Prosedur Penelitian Suatu Pendekatan Praktik*. Jakarta : Rineka Cipta.
- Trianto. (2011). *Desain Pengembangan Pembelajaran Tematik Bagi Anak Usia Dini TK/KA & Anak Usia Kelas Awal SD/MI*. Jakarta : Kencana Prenada Media Group.
- Yuni Sugiarti. (2013). *Analisis dan Perancangan UML (Unified Modeling Language) Generated VB.6*. Yogyakarta : Graha Ilmu.
- Zulfiar Sani. (2010). *Transportasi (Suatu Pengantar)*. Jakarta : Universitas Indonesia.
- Zainal Mustafa. (2009). *Mengurai Variabel Hingga Instrumentasi*. Yogyakarta: Graha Ilmu

LAMPIRAN

Lampiran 1. Surat Izin Penelitian

	<p>KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS NEGERI YOGYAKARTA FAKULTAS TEKNIK</p> <p>Alamat : Kampus Karangmalang, Yogyakarta, 55281 Telp. (0274) 586168 psw. 276,289,292 (0274) 586734 Fax. (0274) 586734 website : http://ft.uny.ac.id e-mail: ft@uny.ac.id ; teknik@uny.ac.id</p>											
<hr/>												
Nomor : 1584/H34/PL/2014	19 Mei 2014											
Lamp. :	-											
Hal :	Ijin Penelitian											
Yth.												
<ol style="list-style-type: none">1. Gubernur DIY c.q. Ka. Biro Adm. Pembangunan Setda DIY2. Gubernur Provinsi DIY c.q. Ka. Bappeda Provinsi DIY3. Walikota Kota Yogyakarta c.q. Kepala Badan Pelayanan Terpadu Kota Yogyakarta4. Kepala Dinas Pendidikan, Pemuda, dan Olahraga Provinsi DIY5. Kepala Dinas Pendidikan, Pemuda, dan Olahraga Kota Yogyakarta6. Kepala SDN Pujokusuman I												
<p>Dalam rangka pelaksanaan Tugas Akhir Skripsi kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan penelitian dengan judul Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun, bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta tersebut di bawah ini:</p>												
<table border="1"><thead><tr><th>No.</th><th>Nama</th><th>NIM</th><th>Jurusan</th><th>Lokasi</th></tr></thead><tbody><tr><td>1</td><td>Anung Budianto</td><td>10520241019</td><td>Pend. Teknik Informatika - S1</td><td>SDN Pujokusuman I</td></tr></tbody></table>	No.	Nama	NIM	Jurusan	Lokasi	1	Anung Budianto	10520241019	Pend. Teknik Informatika - S1	SDN Pujokusuman I		
No.	Nama	NIM	Jurusan	Lokasi								
1	Anung Budianto	10520241019	Pend. Teknik Informatika - S1	SDN Pujokusuman I								
<p>Dosen Pembimbing/Dosen Pengampu :</p> <p>Nama : Dr. Eko Marpanaji, MT NIP : 19670608 199303 1 001</p> <p>Adapun pelaksanaan penelitian dilakukan mulai Mei 2014 s/d Juli 2014.</p> <p>Demikian permohonan ini, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.</p>												
<p style="text-align: right;"> Wakil Dekan I Dr. Sunaryo Soenarto NIP. 19580630 198601 1 0014</p> <p style="text-align: center;"></p>												
Tembusan : Ketua Jurusan												



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS NEGERI YOGYAKARTA
FAKULTAS TEKNIK

Alamat : Kampus Karangmatang, Yogyakarta, 55281

Telp. (0274) 586168 psw. 276.289.292 (0274) 586734 Fax. (0274) 586734

website : <http://ft.uny.ac.id> e-mail: ft@uny.ac.id ; teknik@uny.ac.id



Nomor : 1585/H34/PL/2014

19 Mei 2014

Lamp. : -

Hal : Ijin Penelitian

Yth.

1. Gubernur DIY c.q. Ka. Biro Adm. Pembangunan Setda DIY
2. Gubernur Provinsi DIY c.q. Ka. Bappeda Provinsi DIY
3. Bupati Kabupaten Sleman c.q. Kepala Badan Pelayanan Terpadu Kabupaten Sleman
4. Kepala Dinas Pendidikan, Pemuda, dan Olahraga Provinsi DIY
5. Kepala Dinas Pendidikan, Pemuda, dan Olahraga Kabupaten Sleman
6. Kepala Kepolisian Negara RI Resort Sleman Sektor Bulaksumur

Dalam rangka pelaksanaan Tugas Akhir Skripsi kami mohon dengan hormat bantuan Saudara memberikan ijin untuk melaksanakan penelitian dengan judul Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun, bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta tersebut di bawah ini:

No.	Nama	NIM	Jurusan	Lokasi
1	Anung Budianto	10520241019	Pend. Teknik Informatika - SI	Kepolisian Negara RI Resort Sleman Sektor Bulaksumur

Dosen Pembimbing/Dosen Pengampu :

Nama : Dr. Eko Marpanaji, MT

NIP : 19670608 199303 1 001

Adapun pelaksanaan penelitian dilakukan mulai Mei 2014 s/d Juli 2014.

Demikian permohonan ini, atas bantuan dan kerjasama yang baik selama ini, kami mengucapkan terima kasih.



Dekan,
Wakil Dekan I

Dr. Sunaryo Soenarto

NIP. 19580630 198601 1 001

Tembusan :
Ketua Jurusan



PEMERINTAH DAERAH DAERAH ISTIMEWA YOGYAKARTA
SEKRETARIAT DAERAH
Kompleks Kepatihan, Danurejan, Telepon (0274) 562811 - 562814 (Hunting)
YOGYAKARTA 55213

SURAT KETERANGAN / IJIN
070/REG/VI/468/5/2014

Membaca Surat : **WAKIL DEKAN I FAKULTAS TEKNIK** Nomor : **1583/H34/PL/2014**
Tanggal : **19 MEI 2014** Penhal : **IJIN PENELITIAN/RISET**

Mengingat : 1. Peraturan Pemerintah Nomor 41 Tahun 2006, tentang Perizinan bagi Perguruan Tinggi Asing, Lembaga Penelitian dan Pengembangan Asing, Badan Usaha Asing dan Orang Asing dalam melakukan Kegiatan Penelitian dan Pengembangan di Indonesia;
2. Peraturan Menteri Dalam Negeri Nomor 20 Tahun 2011, tentang Pedoman Penelitian dan Pengembangan di Lingkungan Kementerian Dalam Negeri dan Pemerintah Daerah;
3. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 37 Tahun 2008, tentang Rincian Tugas dan Fungsi Satuan Organisasi di Lingkungan Sekretariat Daerah dan Sekretariat Dewan Perwakilan Rakyat Daerah;
4. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor 18 Tahun 2009 tentang Pedoman Pelayanan Perizinan, Rekomendasi Pelaksanaan Survei, Penelitian, Pendataan, Pengembangan, Pengkajian, dan Studi Lapangan di Daerah Istimewa Yogyakarta.

DIJINKAN untuk melakukan kegiatan survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan kepada:

Nama : **ANUNG BUDIANTO** NIP/NIM : **10520241019**
Alamat : **FAKULTAS TEKNIK, PENDIDIKAN TEKNIK INFORMATIKA, UNIVERSITAS NEGERI YOGYAKARTA**
Judul : **ANALISIS DAN PERANCANGAN GAME EDUKASI "NEED FOR SAFETY" SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS UNTUK ANAK USIA 6-12 TAHUN**
Lokasi : **KOTA YOGYAKARTA & KABUPATEN SLEMAN**
Waktu : **20 MEI 2014 s/d 20 AGUSTUS 2014**

Dengan Ketentuan

1. Menyerahkan surat keterangan/ijin survei/penelitian/pendataan/pengembangan/pengkajian/studi lapangan *) dan Pemerintah Daerah DIY kepada Bupati/Walikota melalui institusi yang berwenang mengeluarkan ijin dimaksud;
2. Menyerahkan soft copy hasil penelitiannya baik kepada Gubernur Daerah Istimewa Yogyakarta melalui Biro Administrasi Pembangunan Setda DIY dalam compact disk (CD) maupun mengunggah (upload) melalui website adbang.jogjaprov.go.id dan menunjukkan cetakan asli yang sudah disahkan dan dibubuhi cap institusi;
3. Ijin ini hanya dipergunakan untuk keperluan ilmiah, dan pemegang ijin wajib mentaati ketentuan yang berlaku di lokasi kegiatan;
4. Ijin penelitian dapat diperpanjang maksimal 2 (dua) kali dengan menunjukkan surat ini kembali sebelum berakhir waktunya setelah mengajukan perpanjangan melalui website adbang.jogjaprov.go.id;
5. Ijin yang diberikan dapat dibatalkan sewaktu-waktu apabila pemegang ijin ini tidak memenuhi ketentuan yang berlaku.

Dikeluarkan di Yogyakarta
Pada tanggal **20 MEI 2014**
A.n Sekretaris Daerah
Asisten Perencanaan dan Pembangunan
Ub.
Kepala Biro Administrasi Pembangunan



Hendar Supriatno, SH
19580120199032003

Tembusan :

1. GUBERNUR DAERAH ISTIMEWA YOGYAKARTA (SEBAGAI LAPORAN)
2. WALIKOTA YOGYAKARTA C.Q DINAS PERIJINAN KOTA YOGYAKARTA
3. BUPATI SLEMAN C.Q KA. BAKESBANGLINMAS SLEMAN
4. DINAS PENDIDIKAN, PEMUDA DAN OLAHRAGA DIY
5. WAKIL DEKAN I FAKULTAS TEKNIK, UNIVERSITAS NEGERI YOGYAKARTA
6. YANG BERSANGKUTAN



PEMERINTAH KOTA YOGYAKARTA
DINAS PERIZINAN

Jl. Kenari No. 56 Yogyakarta Kode Pos : 55165 Telp. (0274) 555241, 515865, 515866, 562682
Fax (0274) 555241
EMAIL : perizinan@jogjakota.go.id
HOT LINE SMS : 081227625000 HOT LINE EMAIL : upik@jogjakota.go.id
WEBSITE : www.perizinan.jogjakota.go.id

SURAT IZIN

NOMOR : 070/1779
3337/34

Dasar : Surat izin / Rekomendasi dari Gubernur Kepala Daerah Istimewa Yogyakarta
Nomor : 070/REG/V/466/5/2014 Tanggal : 20/05/2014
Mengingat : 1. Peraturan Daerah Kota Yogyakarta Nomor 10 Tahun 2008 tentang Pembentukan, Susunan, Kedudukan dan Tugas Pokok Dinas Daerah
2. Peraturan Walikota Yogyakarta Nomor 85 Tahun 2008 tentang Fungsi, Rincian Tugas Dinas Perizinan Kota Yogyakarta;
3. Peraturan Walikota Yogyakarta Nomor 29 Tahun 2007 tentang Pemberian Izin Penelitian, Praktek Kerja Lapangan dan Kuliah Kerja Nyata di Wilayah Kota Yogyakarta;
4. Peraturan Walikota Yogyakarta Nomor 18 Tahun 2011 tentang Penyelenggaraan Perizinan pada Pemerintah Kota Yogyakarta;
5. Peraturan Gubernur Daerah Istimewa Yogyakarta Nomor: 18 Tahun 2009 tentang Pedoman Pelayanan Perizinan, Rekomendasi Pelaksanaan Survei, Penelitian, Pendataan, Pengembangan, Pengembangan, Pengkajian dan Studi Lapangan di Daerah Istimewa Yogyakarta;

Dijinkan Kepada : Nama : ANUNG BUDIANTO NO MHS / NIM : 10520241019
Pekerjaan : Mahasiswa Fak. Teknik - UNY
Alamat : Kampus Karangmalang, Yogyakarta
Penanggungjawab : Dr. Eko Marpanaji
Keperluan : Melakukan Penelitian dengan judul Proposal : ANALISIS DAN PERANCANGAN GAME EDUKASI "NEED FOR SAFETY" SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS UNTUK ANAK USIA 6-12 TAHUN

Lokasi/Responden : Kota Yogyakarta
Waktu : 20/05/2014 Sampai 20/08/2014
Lampiran : Proposal dan Daftar Pertanyaan
Dengan Ketentuan : 1. Wajib Memberi Laporan hasil Penelitian berupa CD kepada Walikota Yogyakarta (Cq. Dinas Perizinan Kota Yogyakarta)
2. Wajib Menjaga Tata tertib dan mentaati ketentuan-ketentuan yang berlaku setempat
3. Izin ini tidak disalahgunakan untuk tujuan tertentu yang dapat mengganggu kestabilan Pemerintah dan hanya diperlukan untuk keperluan ilmiah
4. Surat izin ini sewaktu-waktu dapat dibatalkan apabila tidak dipenuhinya ketentuan-ketentuan tersebut diatas
Kemudian diharap para Pejabat Pemerintah setempat dapat memberi bantuan seperlunya

Tanda tangan
Pemegang Izin

ANUNG BUDIANTO

Dikeluarkan di : Yogyakarta
pada Tanggal : 20-5-2014

Art. Kepala Dinas Perizinan
Sekretaris



Tembusan Kepada :

- Yth. 1. Walikota Yogyakarta (sebagai laporan)
2. Ka. Biro Administrasi Pembangunan Setda DIY
3. Ka. Dinas Pendidikan Kota Yogyakarta

Lampiran 2. Validasi Instrumen Penelitian

Surat Permohonan Validasi Instrumen Penelitian

Hal : Permohonan Validasi Instrumen TAS
Lampiran : 1 Bendel

Kepada Yth,
Bapak Muhammad Munir, M.Pd
Dosen Jurusan Pendidikan Teknik Elektronika
di Fakultas Teknik UNY

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

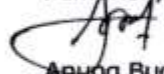
Nama : Anung Budianto
NIM : 10520241019
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun

dengan hormat mohon 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 Ibu diucapkan terima kasih.

Yogyakarta, 15 April 2014

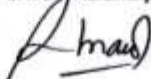
Pemohon,



Anung Budianto
NIM. 10520241019

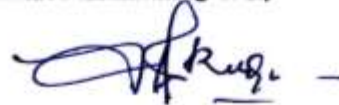
Mengetahui,

Kaprodi Pendidikan Teknik
Informatika,



Dr. Ratna Wardani
NIP. 19701218 200501 2 001

Dosen Pembimbing TAS,



Dr. Eko Marpanaji
NIP. 19670608 199303 1 001

**SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI**

Saya yang bertanda tangan dibawah ini:

Nama : Muhammad Munir, M.Pd
NIP : 19630512 198901 1 001
Jurusan : Pendidikan Teknik Elektronika

menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Anung Budianto
NIM : 10520141019
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*"
Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas
Untuk Anak Usia 6-12 Tahun

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, 17/4/14

Validator,

Muhammad Munir, M.Pd


NIP. 19630512 198901 1 001

Catatan:


☐ Beri tanda ✓

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Anung Budianto NIM : 10520241019
 Judul TAS : Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun

No.	Variabel	Saran/Tanggapan
		
	Komentar Umum/Lain-lain:	

Yogyakarta, 17.4.24.....
 Validator,


 Muhammad Munir M.Pd
 NIP. 19630512 198901 1 001

Surat Permohonan Validasi Instrumen Penelitian

Hal : Permohonan Validasi Instrumen TAS

Lampiran : 1 Bendel

Kepada Yth,

Bapak Suparman, M.Pd

Dosen Jurusan Pendidikan Teknik Elektronika

di Fakultas Teknik UNY

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

Nama : Anung Budianto

NIM : 10520241019

Program Studi : Pendidikan Teknik Informatika

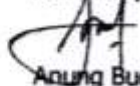
Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun

dengan hormat mohon 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 Ibu diucapkan terima kasih.

Yogyakarta, 15 April 2014

Pemohon,

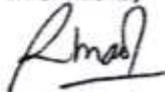


Anung Budianto

NIM. 10520241019

Mengetahui,

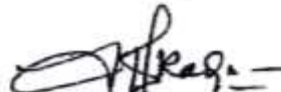
Kaprodi Pendidikan Teknik
Informatika,



Dr. Ratna Wardani

NIP. 19701218 200501 2 001

Dosen Pembimbing TAS,



Dr. Eko Marpanaji

NIP. 19670608 199303 1 001

**SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI**

Saya yang bertanda tangan dibawah ini:

Nama : Suparman, M.Pd
NIP : 19491231 197803 1 004
Jurusan : Pendidikan Teknik Elektronika

menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Anung Budianto
NIM : 10520241019
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis dan Perancangan Game Edukasi "Need For Safety"
Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas
untuk Anak Usia 6-12 Tahun

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,

Validator,



Suparman, M.Pd
NIP. 19491231 197803 1 004

Catatan:

☐ Beri tanda ✓

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Anung Budianto NIM : 10520241019
Judul TAS : Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun.

No.	Variabel	Saran/Tanggapan
	Komentar Umum/Lain-lain:	

Yogyakarta,

Validator,



Suparman, M.Pd
NIP. 19491231 197803 1 004

Surat Permohonan Validasi Instrumen Penelitian

Hal : Permohonan Validasi Instrumen TAS

Lampiran : 1 Bendel

Kepada Yth,

Ibu Dessy Irmawati, M.T.

Dosen Jurusan Pendidikan Teknik Elektronika

di Fakultas Teknik UNY

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

Nama : Anung Budianto

NIM : 10520241019

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun

dengan hormat mohon 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 Ibu diucapkan terima kasih.

Yogyakarta, 15 April 2014

Pemohon,



Anung Budianto
NIM. 10520241019

Mengetahui,

Kaprodi Pendidikan Teknik
Informatika,



Dr. Ratna Wardani
NIP. 19701218 200501 2 001

Dosen Pembimbing TAS,



Dr. Eko Marpanaji
NIP. 19670608 199303 1 001

**SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI**

Saya yang bertanda tangan dibawah ini:

Nama : Dessy Irmawati, M.T.
NIP : 19791214 201012 2 002
Jurusan : Pendidikan Teknik Elektronika

menyatakan bahwa Instrumen penelitian TAS atas nama mahasiswa:

Nama : Anung Budianto
NIM : 10520141019
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis dan Perancangan Game Edukasi "*Need For Safety*"
Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas
Untuk Anak Usia 6-12 Tahun

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan:

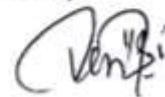
- ☒ Layak digunakan untuk penelitian
☐ Layak digunakan dengan perbaikan
☐ Tidak layak digunakan untuk penelitian yang bersangkutan

dengan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta, 16 April 2014

Validator,



Dessy Irmawati, M.T.

NIP. 19791214 201012 2 002

Catatan:

☐ Beri tanda ✓

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Anung Budianto NIM : 10520241019
 Judul TAS : Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Penguatan Rambu-Rambu Lalu Lintas Untuk Anak Usia 6-12 Tahun

No.	Variabel	Saran/Tanggapan
	Komentar Umum/Lain-lain: Perlu ditambahkan ukuran layar pada sub indikator Variabel portability	

Yogyakarta, 16 April 2014

Validator,



Dessy Imawati, M.T.

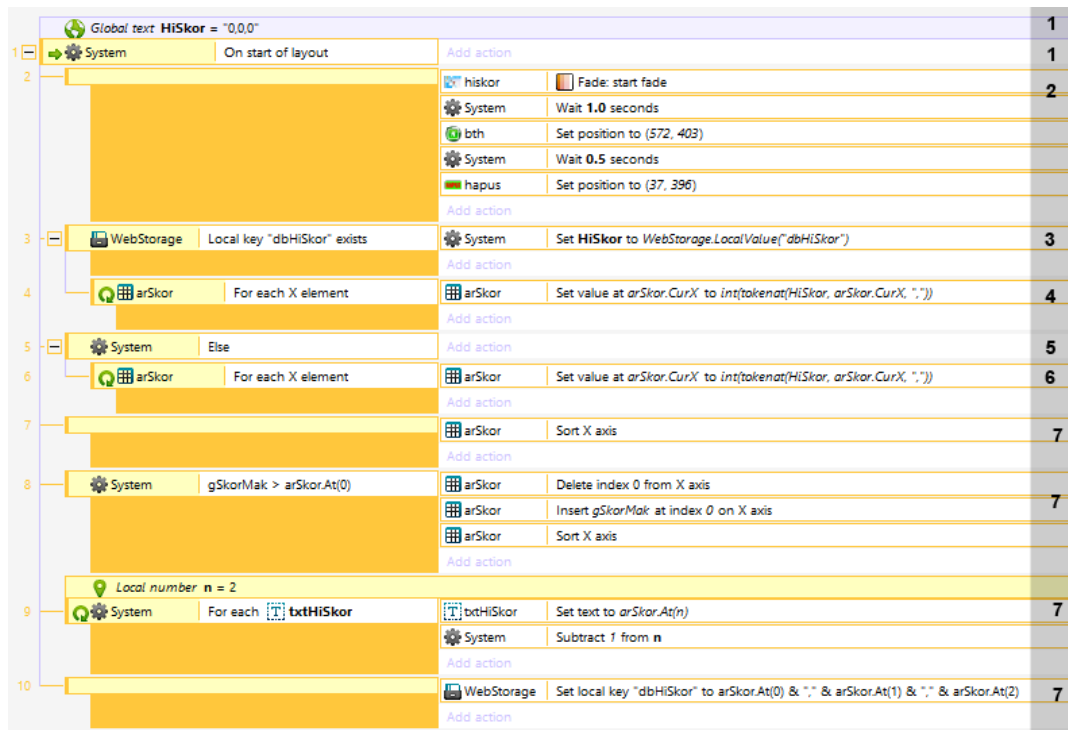
NIP. 19791214 201012 2 002

Lampiran 3. Pengujian White Box

Data hasil pengujian **white box** adalah sebagai berikut:

a. Pengujian *White Box* Menu Skor

Pengujian : Fungsi Simpan_Skor



Gambar 66. *Event* Fungsi Simpan_Skor

Node yang sudah ditunjukkan pada setiap statemen pada fungsi *event* di atas, selanjutnya akan digambarkan dalam bentuk diagram alir.

Tabel 79. *Flow Graph* Fungsi Simpan_Skor

Flow Graph	Cyclomatic Complexity	Independent Path
	1. Jumlah region = 2 2. $V(G) = 7 - 7 + 2 = 2$ 3. $V(G) = 1$ Node Predikat + 1 = 2	1. 1-2-3-4-7 2. 1-2-5-6-7

Perhitungan nilai kompleksitas siklomatik (*cyclomatic complexity*) untuk *flow graph* fungsi Simpan_Skor dapat menggunakan beberapa rumus, yaitu:

- d) Menghitung jumlah *region* yang berhubungan dengan kompleksitas siklomatik. *Flow graph* pada Tabel 79 menggambarkan jumlah *region* sebanyak 2 sehingga nilai kompleksitas siklomatiknya adalah 2.
- e) Menggunakan rumus $V(G) = E - N + 2$, dimana E adalah jumlah *edge* yang menunjukkan aliran kendali atau perjalanan logika dan N adalah jumlah node grafik alir. *Flow graph* pada Tabel 79 menggambarkan jumlah *edge* sebanyak 7 dan jumlah node sebanyak 7, sehingga $V(G) = 7 - 7 + 2 = 2$.
- f) Menggunakan rumus $V(G) = P + 1$, dimana P adalah jumlah node predikat yang terdapat dalam grafik alir. Node predikat merupakan node yang memiliki percabangan *edge* atau aliran kendali. *Flow graph* pada Tabel 79 menggambarkan jumlah node predikat sebanyak 1, sehingga kompleksitas siklomatik atau $V(G) = 1 + 1 = 2$.

Tabel 80. Uji *Test Case* Fungsi Simpan_Kotak

Path	Input	Output	Hasil Yang Diharapkan
1	Memainkan permainan dan menghasilkan nilai	Nilai disimpan dalam webstorage dengan nama dbHiSkor, Nilai diurutkan dari yang terbesar ke terkecil dan ditampilkan di menu Skor	Sesuai
2	Belum pernah memainkan permainan dan tidak ada nilai yang disimpan	Nilai yang ditampilkan adalah angka "nol"	Sesuai

Pengujian kasus untuk fungsi Simpan_Kotak pada tabel di atas mendapatkan hasil yang sudah sesuai dengan yang diharapkan. *Output* program sudah sesuai dengan *input* yang diberikan.

b. Pengujian *White Box* Alur Masuk Menu Ayo Belajar

Tabel 81. *Flow Graph* Alur Menu Ayo Belajar Aplikasi *Need For Safety*

<i>Flow Graph</i>	Keterangan
	<ol style="list-style-type: none"> Node 1 : Mulai Node 2 : Tampil Submenu Ayo Belajar Node 3 : Submenu Pengertian Rambu Node 4 : Tampil Pengertian Rambu Node 5 : Submenu Rambu Peringatan Node 6 : Tampil Pengertian Rambu Peringatan Node 7 : Tampil Macam Rambu Peringatan Node 8 : Submenu Rambu Larangan Node 9 : Tampil Pengertian Rambu Larangan Node 10 : Tampil Macam Rambu larangan Node 11 : Submenu Rambu Perintah Node 12 : Tampil Pengertian Rambu Perintah Node 13 : Tampil Macam Rambu Perintah Node 14 : Submenu Rambu Petunjuk Node 15 : Tampil Pengertian Rambu Petunjuk Node 16 : Tampil Macam Rambu Petunjuk Node 17 : Selesai

Tabel 82. Perhitungan *Cyclomatic Complexity* dan *Independent Path* Alur Menu Ayo Belajar Aplikasi *Need For Safety*

<i>Cyclomatic Complexity</i>	<i>Independent Path</i>
<ol style="list-style-type: none"> Jumlah region = 10 $V(G) = 25 - 17 + 2 = 10$ $V(G) = 9 \text{ Node Predikat} + 1 = 10$ 	<ol style="list-style-type: none"> 1-2-3-4-17 1-2-3-5-6-7-17 1-2-3-5-6-17 1-2-3-5-8-9-10-17 1-2-3-5-8-9-17 1-2-3-5-8-11-12-13-17 1-2-3-5-8-11-12-17 1-2-3-5-8-11-14-15-16-17 1-2-3-5-8-11-14-15-17 1-2-3-5-8-11-14-17

Perhitungan nilai kompleksitas siklomatik (*cyclomatic complexity*) untuk *flow graph* pada Tabel 81 dapat menggunakan beberapa rumus, yaitu:

a) Menghitung jumlah *region* yang berhubungan dengan kompleksitas siklomatik. *Flow graph* pada Tabel 81 menggambarkan jumlah *region* sebanyak 10 sehingga nilai kompleksitas siklomatiknya adalah 10.

b) Menggunakan rumus $V(G) = E - N + 2$. *Flow graph* pada Tabel 82 menggambarkan jumlah edge (E) sebanyak 25 dan jumlah node (N) sebanyak 17, sehingga $V(G) = 25 - 17 + 2 = 10$.

c) Menggunakan rumus $V(G) = P + 1$. *Flow graph* pada Tabel 82 menggambarkan jumlah node predikat (P) sebanyak 9, sehingga kompleksitas siklomatik atau $V(G) = 9 + 1 = 10$.

Tabel 83. Uji *Test Case* Alur Menu Ayo Belajar Aplikasi *Need For Speed*

Path	Input	Output	Hasil Yang Diharapkan
1	Pilih Menu Ayo Belajar, Pilih submenu Pengertian Rambu	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu	Sesuai
2	Pilih Menu Ayo Belajar, Pilih submenu Rambu Peringatan, pilih tombol "right"	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu peringatan, tampil contoh rambu peringatan	Sesuai
3	Pilih Menu Ayo Belajar, Pilih submenu Rambu Peringatan	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu peringatan	Sesuai
4	Pilih Menu Ayo Belajar, Pilih submenu Rambu Larangan, pilih tombol "right"	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Larangan, tampil contoh rambu Larangan	Sesuai
5	Pilih Menu Ayo Belajar, Pilih submenu Rambu Larangan	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Larangan	Sesuai
6	Pilih Menu Ayo Belajar, Pilih submenu Rambu Perintah, pilih tombol "right"	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Perintah, tampil contoh rambu Perintah	Sesuai
7	Pilih Menu Ayo Belajar, Pilih submenu Rambu Perintah	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Perintah	Sesuai
8	Pilih Menu Ayo Belajar, Pilih submenu Rambu Petunjuk, pilih tombol "right"	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Petunjuk, tampil contoh rambu Petunjuk	Sesuai
9	Pilih Menu Ayo Belajar, Pilih submenu Rambu Petunjuk	Tampil Submenu Ayo Belajar, tampil Pengertian Rambu Petunjuk	Sesuai
10	Pilih Menu Ayo Belajar	Tampil Submenu Ayo Belajar	Sesuai

Pengujian *white box* pada alur masuk menu ayo belajar aplikasi *Need For Safety* mendapatkan hasil yaitu output sudah sesuai dengan yang diharapkan. Masing-masing *input* yang diberikan telah diproses dan menghasilkan *output* yang sesuai.

c. Pengujian White Box Alur Aplikasi *Need For Safety*

Flow Graph Alur Aplikasi “*Need For Safety*” dari proses memulai aplikasi hingga keluar aplikasi adalah:

Tabel 84. *Flow Graph* Alur Aplikasi *Need For Safety*

Flow Graph	Keterangan
	<ol style="list-style-type: none"> Node 1 : Mulai Node 2 : Halaman Home Node 3 : Off Musik Node 4 : Musik Mati Node 5 : On Musik Node 6 : Musik Hidup Node 7 : Menu Skor Node 8 : Tampil Skor Node 9 : Menu Mulai Bermain Node 10 : Tampil Permainan Level 1 Node 11 : Menu Ayo Belajar Node 12 : Tampil Submenu Ayo Belajar Node 13 : Menu Bantuan Node 14 : Tampil Bantuan Node 15 : Keluar Node 16 : Selesai

Flow graph yang telah dibuat kemudian dihitung nilai kompleksitas siklomatik dan jalur dasarnya.

Tabel 85. Nilai *Cyclomatic Complexity* alur aplikasi *Need For Safety*

Cyclomatic Complexity	Independent Path
<ol style="list-style-type: none"> Jumlah region = 7 $V(G) = 21 - 16 + 2 = 7$ $V(G) = 6 \text{ Node Predikat} + 1 = 7$ 	<ol style="list-style-type: none"> 1-2-3-4-15-16 1-2-3-5-6-15-16 1-2-3-5-7-8-15-16 1-2-3-5-7-9-10-15-16 1-2-3-5-7-9-11-12-15-16 1-2-3-5-7-9-11-13-14-15-16 1-2-3-5-7-9-11-13-15-16

Perhitungan nilai kompleksitas siklomatik (*cyclomatic complexity*) untuk *flow graph* pada Tabel 85 dapat menggunakan beberapa rumus sebagai berikut:

- d) Menghitung jumlah *region* pada *flow graph* alur aplikasi *Need For Safety*, yaitu berjumlah 7 sehingga nilai kompleksitas siklomatiknya adalah 7.

- e) Menggunakan rumus $V(G) = E - N + 2$. *Flow graph* alur aplikasi *Need For Safety* menggambarkan jumlah edge (E) sebanyak 21 dan jumlah node (N) sebanyak 16, sehingga kompleksitas siklomatik atau $V(G) = 21 - 16 + 2 = 7$.
- f) Menggunakan rumus $V(G) = P + 1$. *Flow graph* alur aplikasi *Need For Safety* menggambarkan jumlah node predikat (P) sebanyak 6, sehingga kompleksitas siklomatik atau $V(G) = 6 + 1 = 7$.

Tabel 86. Uji *Test Case* Alur Aplikasi *Need For Speed*

Path	Input	Output	Hasil Yang Diharapkan
1	Pilih tombol mulai, pilih Off musik, keluar	Masuk Menu Home, Suara musik mati, dan keluar dari game	Sesuai
2	Pilih tombol mulai, pilih On musik, keluar	Masuk Menu Home, Suara musik hidup, dan keluar dari game	Sesuai
3	Pilih tombol mulai, pilih tombol skor, keluar	Masuk Menu Home, Suara musik hidup, masuk menu lihat skor, dan keluar dari game	Sesuai
4	Pilih tombol mulai, pilih tombol mulai bermain, keluar	Masuk Menu Home, Suara musik hidup, masuk menu permainan level 1, dan keluar dari game	Sesuai
5	Pilih tombol mulai, pilih tombol ayo belajar, keluar	Masuk Menu Home, Suara musik hidup, masuk submenu ayo belajar (submenu pengertian rambu, submenu rambu peringatan, submenu rambu larangan, submenu rambu perintah, submenu rambu petunjuk), dan keluar dari game	Sesuai
6	Pilih tombol mulai, pilih tombol bantuan, keluar	Masuk Menu Home, Suara musik hidup, masuk menu bantuan, dan keluar dari game	Sesuai
7	Pilih tombol mulai, seluar	Masuk Menu Home, dan keluar	Sesuai

Pengujian kasus untuk alur aplikasi *Need For Safety* secara keseluruhan yang telah dilakukan mendapatkan hasil yang sudah sesuai dengan yang diharapkan. *Output* program sudah sesuai dengan masing-masing *input* yang diberikan. Data pengujian *white box* secara keseluruhan menunjukkan bahwa setiap node sudah dieksekusi minimal satu kali oleh sistem dan menghasilkan keluaran (*output*) yang sesuai dengan harapan pengembang.

Lampiran 4. Pengujian Alpha oleh Ahli

Yogyakarta, 30 April 2014

Kepada

Yth. Drs. Priyanto, M.Kom

Di Universitas Negeri Yogyakarta

Melalui surat ini saya memohon dengan hormat kepada Bapak untuk sedianya menjadi penilai ahli media terhadap perangkat lunak sebagai hasil dari skripsi saya dengan judul **"Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun"**.

Demikian surat permohonan dari saya. Atas kesempatan yang diberikan untuk mengevaluasi media tersebut, saya mengucapkan terima kasih.

Mengetahui

Dosen Pembimbing,



Dr. Eko Marpanaji

NIP. 19670608 199303 1 001

Pemohon,



Anung Budianto

NIM. 10520241019

INSTRUMEN PENELITIAN

ANALISIS DAN PERANCANGAN GAME EDUKASI "*NEED FOR SAFETY*" SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS UNTUK ANAK USIA 6-12 TAHUN

Nama : Drs. Priyanto, M.Kom

Bidang Keahlian : Dosen

Petunjuk :

1. Pertanyaan dibawah ini diisi dan dijawab dengan memberi *checklist* (✓) pada jawaban yang anda pilih.
2. Komentar ataupun saran mohon ditulis pada lembar yang telah disediakan.
3. Keterangan alternatif jawaban untuk tabel uji kelayakan :

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

Atas kesediaan Bapak/Ibu Dosen Ahli Rekayasa Perangkat Lunak untuk mengisi angket uji kelayakan ini, saya ucapkan terima kasih.

Spesifikasi Uji Membuka Aplikasi

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Tampil halaman <i>loading</i>	✓	
2	User memilih tombol "masuk"	Tampil halaman menu <i>game</i> (menu skor, menu bantuan, menu mulai bermain, menu ayo belajar, dan menu <i>On/Off</i> musik)	✓	

Spesifikasi Uji Menu *Game*

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Menu <i>Game</i>	Menu <i>game</i> dapat terbuka setelah <i>user</i> menekan tombol "masuk" pada halaman <i>loading</i>	✓	
		<i>User</i> dapat masuk ke semua menu dengan memilih tombol menu-menu yang ada	✓	
		<i>User</i> dapat menonaktifkan musik <i>game</i> dengan memilih tombol "off musik" pada menu awal	✓	
		<i>User</i> dapat mengaktifkan musik setelah musik mati dengan memilih tombol "on musik" pada menu awal	✓	
		<i>User</i> dapat kembali ke menu awal dengan memilih tombol "home"	✓	
		<i>User</i> dapat keluar aplikasi dengan memilih tombol "back" pada <i>handphone</i>	✓	

Spesifikasi Uji Menu Mulai Bermain

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Mulai Bermain	<i>User</i> dapat masuk ke menu "mulai bermain"	✓	
		Muncul permainan level 1	✓	
		Setelah <i>user</i> menyelesaikan level 1 langsung menuju level 2	✓	
		Setelah <i>user</i> menyelesaikan level 2 langsung menuju level 3	✓	
		Setelah <i>user</i> menyelesaikan level 3 langsung menuju level 4	✓	
		Setelah <i>user</i> menyelesaikan level 4 langsung muncul informasi nilai	✓	

2	Tampil skor nilai	<i>User</i> dapat menuju halaman skor setelah memilih tombol "skor"	✓	
		<i>User</i> dapat memainkan permainan kembali dari level 1 setelah memilih tombol " <i>left</i> /arah kiri"	✓	

Spesifikasi Uji Menu Ayo Belajar

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Ayo Belajar	<i>User</i> dapat masuk ke menu "ayo belajar"	✓	
		Tampil submenu ayo belajar	✓	
2	Pengertian Rambu	<i>User</i> dapat masuk ke halaman pengertian rambu setelah memilih submenu pengertian rambu	✓	
3	Rambu Peringatan	<i>User</i> dapat masuk ke halaman pengertian rambu peringatan setelah memilih submenu rambu peringatan	✓	
		Menuju halaman contoh rambu peringatan setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	✓	
		Pada halaman contoh rambu peringatan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	
5	Rambu Larangan	<i>User</i> dapat masuk ke halaman pengertian rambu larangan setelah memilih submenu rambu larangan	✓	
		Menuju halaman contoh rambu larangan setelah <i>user</i> memilih tombol " <i>right</i> /arah kanan"	✓	
		Pada halaman contoh rambu larangan, <i>user</i> dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	

6	Rambu Perintah	User dapat masuk ke halaman pengertian rambu perintah setelah memilih submenu rambu perintah	✓	
		Menuju halaman contoh rambu perintah setelah user memilih tombol "right/darah kanan"	✓	
		Pada halaman contoh rambu perintah, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	
7	Rambu Petunjuk	User dapat masuk ke halaman pengertian rambu petunjuk setelah memilih submenu rambu petunjuk	✓	
		Menuju halaman contoh rambu petunjuk setelah user memilih tombol "right/darah kanan"	✓	
		Pada halaman contoh rambu petunjuk, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	

Uji Kelayakan Perangkat Lunak Aspek *Functionality*, *Efficiency*, dan *Usability*

No	Pernyataan	Alternatif Jawaban				
		STS	TS	N	S	SS
Variabel <i>Functionality</i>						
1	Perangkat lunak dapat menampilkan skor/nilai permainan					✓
2	Perangkat lunak dapat menampilkan permainan mencocokkan gambar					✓
3	Perangkat lunak dapat menampilkan definisi macam-macam rambu lalu lintas dan contohnya masing-masing					✓
4	Perangkat lunak dapat menampilkan halaman bantuan berisi pengertian perangkat lunak dan cara memainkan permainannya					✓

5	Setiap fitur/menu berfungsi sesuai dengan fungsinya masing-masing					✓
6	Perangkat lunak mampu bekerja memadukan fitur webstorage untuk penyimpanan data skor					✓
Variabel Efficiency						
7	Tiap proses membutuhkan jeda waktu yang singkat					✓
8	Respon dari setiap proses sesuai dengan fungsinya masing-masing					✓
Variabel Usability						
9	Kemudahan dalam mempelajari fitur-fitur yang ada				✓	
10	Kejelasan dalam mengoperasikan fitur-fitur yang ada					✓
11	Memberikan informasi yang mudah dipahami					✓
12	Penggunaan tampilan per menu/fitur mudah dipelajari					✓
13	Permainan yang mudah dipelajari/dimainkan					✓
14	Tampilan per menu/fitur menggunakan objek gambar yang menarik					✓
15	Kelengkapan menu operasi				✓	
16	Animasi perpindahan menu/fitur yang ada menarik				✓	

SARAN/KOMENTAR :

KESIMPULAN :

Perangkat Lunak ini dinyatakan : (Lingkari salah satu)

- ☒ 1. Memiliki unjuk kerja yang baik
- ☐ 2. Memiliki unjuk kerja yang buruk
- ☐ 3. Tidak berfungsi

Penguji,



Dr. Privanto, M.Kom

NIP. 19620625 198503 1 002

Yogyakarta, 30 April 2014

Kepada

Yth. Herman Dwi Surjono.,M.Sc.,M.T.,Ph.D

Di Universitas Negeri Yogyakarta

Melalui surat ini saya memohon dengan hormat kepada Bapak untuk sedianya menjadi penilai ahli media terhadap perangkat lunak sebagai hasil dari skripsi saya dengan judul **"Analisis dan Perancangan Game Edukasi "Need For Safety" Sebagai Sarana Pengenalan Rambu-Rambu Lalu Lintas untuk Anak Usia 6-12 Tahun"**.

Demikian surat permohonan dari saya. Atas kesempatan yang diberikan untuk mengevaluasi media tersebut, saya mengucapkan terima kasih.

Mengetahui

Dosen Pembimbing,



Dr. Eko Marpanaji

NIP. 19670608 199303 1 001

Pemohon,



Anung Budianto

NIM. 10520241019

INSTRUMEN PENELITIAN

**ANALISIS DAN PERANCANGAN GAME EDUKASI "NEED FOR SAFETY"
SEBAGAI SARANA PENGENALAN RAMBU-RAMBU LALU LINTAS
UNTUK ANAK USIA 6-12 TAHUN**

Nama : Herman Dwi Surjono.,M.Sc.,M.T.,Ph.D

Bidang Keahlian : Dosen

Petunjuk :

1. Pertanyaan dibawah ini diisi dan dijawab dengan memberi *checklist* (✓) pada jawaban yang anda pilih.
2. Komentar ataupun saran mohon ditulis pada lembar yang telah disediakan.
3. Keterangan alternatif jawaban untuk tabel uji kelayakan :

STS : Sangat Tidak Setuju

TS : Tidak Setuju

N : Netral

S : Setuju

SS : Sangat Setuju

Atas kesediaan Bapak/Ibu Dosen Ahli Rekayasa Perangkat Lunak untuk mengisi angket uji kelayakan ini, saya ucapkan terima kasih.

Spesifikasi Uji Membuka Aplikasi

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Membuka Aplikasi	Tampil halaman <i>loading judul (title page)</i>	✓	
2	User memilih tombol "masuk"	Tampil halaman menu <i>game</i> (menu skor, menu bantuan, menu mulai bermain, menu ayo belajar, dan menu <i>On/Off</i> musik)	✓	

Spesifikasi Uji Menu *Game*

No	Aktivitas/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Menu <i>Game</i>	Menu <i>game</i> dapat terbuka setelah <i>user</i> menekan tombol "masuk" pada halaman <i>loading</i>	✓	
		<i>User</i> dapat masuk ke semua menu dengan memilih tombol menu-menu yang ada	✓	
		<i>User</i> dapat menonaktifkan musik <i>game</i> dengan memilih tombol "off musik" pada menu awal	✓	
		<i>User</i> dapat mengaktifkan musik setelah musik mati dengan memilih tombol "on musik" pada menu awal	✓	
		<i>User</i> dapat kembali ke menu awal dengan memilih tombol "home"	✓	
		<i>User</i> dapat keluar aplikasi dengan memilih tombol "back" pada <i>handphone</i>	✓	

Spesifikasi Uji Menu Mulai Bermain

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Mulai Bermain	<i>User</i> dapat masuk ke menu "mulai bermain"	✓	
		Muncul permainan level 1	✓	
		Setelah <i>user</i> menyelesaikan level 1 langsung menuju level 2	✓	
		Setelah <i>user</i> menyelesaikan level 2 langsung menuju level 3	✓	
		Setelah <i>user</i> menyelesaikan level 3 langsung menuju level 4	✓	
		Setelah <i>user</i> menyelesaikan level 4 langsung muncul informasi nilai	✓	

2	Tampil skor nilai	User dapat menuju halaman skor setelah memilih tombol "skor"	✓	
		User dapat memainkan permainan kembali dari level 1 setelah memilih tombol " <i>left</i> /arah kiri"	✓	

Spesifikasi Uji Menu Ayo Belajar

No	Aksi/Menu	Hasil yang diharapkan	Ketercapaian	
			Ya	Tidak
1	Masuk Menu Ayo Belajar	User dapat masuk ke menu "ayo belajar"	✓	
		Tampil submenu ayo belajar	✓	
2	Pengertian Rambu	User dapat masuk ke halaman pengertian rambu setelah memilih submenu pengertian rambu	✓	
3	Rambu Peringatan	User dapat masuk ke halaman pengertian rambu peringatan setelah memilih submenu rambu peringatan	✓	
		Menuju halaman contoh rambu peringatan setelah user memilih tombol " <i>right</i> /arah kanan"	✓	
		Pada halaman contoh rambu peringatan, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	
5	Rambu Larangan	User dapat masuk ke halaman pengertian rambu larangan setelah memilih submenu rambu larangan	✓	
		Menuju halaman contoh rambu larangan setelah user memilih tombol " <i>right</i> /arah kanan"	✓	
		Pada halaman contoh rambu larangan, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	

6	Rambu Perintah	User dapat masuk ke halaman pengertian rambu perintah setelah memilih submenu rambu perintah	✓	
		Menuju halaman contoh rambu perintah setelah user memilih tombol "right/arah kanan"	✓	
		Pada halaman contoh rambu perintah, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	
7	Rambu Petunjuk	User dapat masuk ke halaman pengertian rambu petunjuk setelah memilih submenu rambu petunjuk	✓	
		Menuju halaman contoh rambu petunjuk setelah user memilih tombol "right/arah kanan"	✓	
		Pada halaman contoh rambu petunjuk, user dapat melihat arti masing-masing rambu dengan memilih ikon rambu tersebut	✓	

Uji Kelayakan Perangkat Lunak Aspek *Functionality*, *Efficiency*, dan *Usability*

No	Pernyataan	Alternatif Jawaban				
		STS	TS	N	S	SS
Variabel <i>Functionality</i>						
1	Perangkat lunak dapat menampilkan skor/nilai permainan				✓	
2	Perangkat lunak dapat menampilkan permainan mencocokkan gambar				✓	
3	Perangkat lunak dapat menampilkan definisi macam-macam rambu lalu lintas dan contohnya masing-masing				✓	
4	Perangkat lunak dapat menampilkan halaman bantuan berisi pengertian perangkat lunak dan cara memainkan permainannya				✓	

5	Setiap fitur/menu berfungsi sesuai dengan fungsinya masing-masing				✓	
6	Perangkat lunak mampu bekerja memadukan fitur webstorage untuk penyimpanan data skor				✓	
Variabel <i>Efficiency</i>						
7	Tiap proses membutuhkan jeda waktu yang singkat		✓			
8	Respon dari setiap proses sesuai dengan fungsinya masing-masing				✓	
Variabel <i>Usability</i>						
9	Kemudahan dalam mempelajari fitur-fitur yang ada				✓	
10	Kejelasan dalam mengoperasikan fitur-fitur yang ada				✓	
11	Memberikan informasi yang mudah dipahami				✓	
12	Penggunaan tampilan per menu/fitur mudah dipelajari				✓	
13	Permainan yang mudah dipelajari/dimainkan				✓	
14	Tampilan per menu/fitur menggunakan objek gambar yang menarik				✓	
15	Kelengkapan menu operasi				✓	
16	Animasi perpindahan menu/fitur yang ada menarik				✓	

SARAN/KOMENTAR :

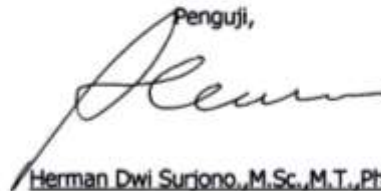
- Di title page perlu identitas pengunby (nama, Min, prodi)
- perlu tombol back pd sub materi /
- pada menu materi, perlu diberi ilustrasi yg menarik masing2 menu.

KESIMPULAN :

Perangkat Lunak ini dinyatakan : (Lingkari salah satu)

- ☒ 1. Memiliki unjuk kerja yang baik
- ☐ 2. Memiliki unjuk kerja yang buruk
- ☐ 3. Tidak berfungsi

Penguji,



Herman Dwi Surjono, M.Sc., M.T., Ph.D

NIP. 19640205 198703 1 001

Lampiran 5. Kamus Event Construct2

Behaviour Reference

Nama	Keterangan
8 direction	Properties
	<i>Acceleration</i> Pengaturan gerakan mempercepat gerakan objek
	<i>Deceleration</i> Pengaturan gerakan memperlambat gerakan objek
	<i>Default controls</i> Secara default kontrol gerakan menggunakan arah pada <i>keyboard</i>
	<i>Direction</i> Mengatur banyaknya arah gerakan objek
	<i>Max Speed</i> Kecepatan maksimal dari objek dapat berjalan di beberapa arah (<i>pixels/second</i>)
	<i>Set angle</i> Pengaturan nilai sudut untuk sebuah objek
	Conditions
	<i>Compare speed</i> Membandingkan kecepatan objek
	<i>Is moving</i> Nilai " <i>True</i> " jika objek bergerak
	Actions
	<i>Reverse</i> Kebalikan dari gerakan <i>motion</i>
	<i>Set acceleration</i> <i>Set deceleration</i> <i>Set max speed</i> Pengaturan <i>properties</i>
	<i>Set enabled</i> Pengaturan aktif/tidak aktif gerakan " <i>8 direction</i> "
	<i>Set ignoring input</i> Pengaturan <i>input</i> " <i>8 direction</i> "
	<i>Set speed</i> Pengaturan kecepatan " <i>8 direction</i> "
	<i>Simulate control</i> Simulasi salah satu gerakan " <i>8 direction</i> "
	<i>Stop</i> Pengaturan kecepatan ke nilai 0 (berhenti bergerak)
	Expressions
	<i>Acceleration, deceleration, max speed</i> Pengaturan <i>properties</i>
	<i>movingAngle</i> Mendapatkan nilai sudut dari gerakan objek
	<i>Speed</i> Mendapatkan nilai dari kecepatan objek
	<i>VectorX, vectorY</i> Mendapatkan kecepatan objek pada masing-masing sumbu (Koordinat)
Anchor	Conditions, actions, and expressions
	<i>Set enabled</i> Pengaturan kondisi aktif/non aktif <i>behaviour</i> (<i>anchor</i>)
Bound Of Layout	Properties
	<i>Bound by</i> Pengaturan tepi sebuah objek terhadap <i>layout</i>
Bullet	Properties
	<i>Acceleration</i> Pengaturan mempercepat gerakan objek
	<i>Bounce off solid</i> Pengaturan gerakan memantul
	<i>Gravity</i> Membuat gerak gravitasi, jatuh ke bawah
	<i>Set angle</i> Pengaturan nilai sudut suatu objek
	<i>Speed</i> Pengaturan kecepatan objek
	Conditions
	<i>Compare distance travelled</i> Membandingkan total jarak " <i>bullet</i> " selama bergerak
	<i>Compare speed</i> Membandingkan kecepatan " <i>bullet</i> " dengan nilai tertentu
	Actions
	<i>Bounce off object</i> Pengaturan gerakan memantul sebuah objek
	<i>Set acceleration, Set angle</i> Penambahan nilai (<i>properties</i>)

	<i>of motion, Set enabled, Set gravity, Set speed</i>	
Fade	Properties	
	<i>Activate at start</i>	Animasi fade aktif saat <i>layout</i> dijalankan
	<i>Destroy</i>	Objek hancur setelah <i>fade</i> "after fade out"
	<i>Fade in time</i>	<i>Fade in</i> (animasi fade dari <i>visibility</i> 0 ke 1)
	<i>Fade out time</i>	<i>Fade out</i> (animasi fade dari <i>visibility</i> 1 ke 0)
	<i>Wait time</i>	Jeda waktu <i>fade in</i> dan <i>fade out</i>
	Conditions	
	<i>On fade out finished</i>	Kondisi dijalankan setelah animasi <i>fade</i> selesai
	Actions	
	<i>Restart fade</i>	Memulai animasi <i>fade</i> dari awal
Flash	<i>Start fade</i>	Jika "active on start" disetting "no", <i>fade</i> akan dijalankan
	Conditions	
	<i>Flash</i>	Membuat objek menjalankan animasi <i>flash</i>
	<i>Stop flashing</i>	Menghentikan animasi <i>flash</i>

Plugin Reference

Nama	Keterangan	
Array	Properties	
	<i>Width</i>	Dimensi X
	<i>Height</i>	Dimensi Y
	<i>Depth</i>	Dimensi Z
	Conditions	
	<i>Compare at X</i>	Membandingkan nilai posisi di <i>array</i>
	<i>Compare at XY</i>	Membandingkan nilai posisi di <i>array</i>
	<i>Compare at XYZ</i>	Membandingkan nilai posisi di <i>array</i>
	<i>Compare current value</i>	Membandingkan nilai dalam <i>array</i>
	<i>Contains value</i>	Menyertakan nilai <i>array</i>
	<i>For each element</i>	Pengulangan kondisi
	<i>Is empty</i>	Kondisi <i>array</i> kosong
	Actions	
	<i>Clear</i>	Pengaturan nilai <i>array</i> menjadi kosong
	<i>Set X, set XY, set XYZ</i>	Pengaturan nilai X,Y,Z dalam <i>array</i>
	<i>Set size</i>	Pengaturan besar <i>array</i>
	<i>Sort</i>	Mengurutkan element dalam <i>array</i>
Audio	Properties	
	<i>Timescale audio</i>	Pengaturan mempercepat atau memperlambat <i>playback</i> .
	Conditions	
	<i>On ended</i>	Kondisi dijalankan ketika suara berhenti dimainkan
	Actions	
	<i>Play</i>	Mulai memainkan file audio
	<i>Preload</i>	Mulai mengunduh file audio
	<i>Set looping</i>	Pengaturan pengulangan file audio yang dimainkan
	<i>Set muted</i>	Pengaturan mode <i>silent</i>
	<i>Set silent</i>	Pengaturan mode <i>silent</i> untuk semua audio yang dimainkan
	<i>Set volume</i>	Pengaturan <i>volume</i> suara

	<i>Stop</i>	Menghentikan suara
Keyboard	Conditions	
	<i>Key is down</i>	Kondisi ketika keyboard ditekan
	<i>On any key pressed</i>	Ketika salah satu keyboard ditekan
	<i>On key pressed</i>	Ketika spesifik keyboard ditekan
	<i>On key released</i>	Ketika keyboard dilepas
Mouse	Conditions	
	<i>Cursor is over object</i>	Kondisi "True" ketika kursor berada diatas objek
	<i>On any click</i>	Kondisi ketika tombol mouse ditekan
	<i>On button released</i>	Kondisi ketika tombol mouse dilepas
	<i>On click</i>	Kondisi ketika spesifik tombol mouse ditekan
	<i>On mouse wheel</i>	Kondisi ketika menggunakan <i>scroll</i> pada mouse
	<i>On object clicked</i>	Kondisi ketika objek dipilih/ditekan
Sprite	Properties	
	<i>Animations</i>	Pengaturan animasi objek
	<i>Size</i>	Pengaturan ukuran objek
	<i>Initial visibility</i>	Pengaturan <i>visibility</i> objek
	<i>Initial frame</i>	Pengaturan frame
	Conditions	
	<i>Set animation</i>	Mengubah animasi objek (<i>sprite</i>)
	<i>Set flipped</i>	Mengubah tampilan objek secara normal atau <i>flipped</i>
	<i>Set mirrored</i>	Menampilkan <i>mirror</i> objek
	<i>Start</i>	<i>Start animation</i>
	<i>Stop</i>	<i>Stop animation</i>
Text	Properties	
	<i>Color</i>	Pengaturan warna teks
	<i>Font</i>	Pengaturan jenis <i>font</i> teks
	<i>Initial visibility</i>	Pengaturan <i>visibility</i> objek teks
	<i>Text</i>	Penambahan teks
	Conditions	
	<i>Compare text</i>	Membandingkan nilai " <i>string</i> "
	Actions	
	<i>Set font color</i>	Mengubah warna teks
	<i>Set font face</i>	Mengubah tampilan teks
	<i>Set font size</i>	Mengubah ukuran teks
Textbox	Properties	
	<i>Text</i>	Pengaturan teks yang akan ditampilkan
	Conditions	
	<i>Compare text</i>	Membandingkan nilai " <i>string</i> "
	<i>On clicked</i>	Dijalankan ketika <i>user</i> menekan area teks
	<i>On double-clicked</i>	Dijalankan ketika <i>user</i> menekan dua kali area teks
	<i>On text changed</i>	Dijalankan ketika teks dimodifikasi
	Actions	
	<i>Set enabled</i>	Pengaturan aktif/tidak aktif objek teks
Tile Background	Properties	
	<i>Image</i>	Menampilkan fungsi <i>edit</i> pada objek gambar
	<i>Initial visibility</i>	Pengaturan <i>visibility</i> objek gambar
Touch	Conditions	
	<i>Compare touch speed</i>	Membandingkan kecepatan dalam menekan objek
	<i>Is in touch</i>	kondisi saat <i>user</i> menyentuh layar
	<i>Is touching objek</i>	kondisi jika ada sentuhan pada layar

	<i>On touch end</i>	Kondisi ketika sentuhan berakhir
	<i>On touch start</i>	Kondisi ketika sentulan berawal
	<i>On touched object</i>	Kondisi ketika objek disentuh
Webstorage	Conditions	
	<i>Local/session key exist</i>	Mengecek apakah ada nilai atau data yang tersimpan dalam webstorage dengan inisial nama tertentu
	Actions	
	<i>Clear local/session storage</i>	mengosongkan nilai dalam webstorage ke nilai awal
	<i>Remove local/session value</i>	Menghapus nilai dalam webstorage
	<i>Set local/session value</i>	Menyimpan nilai dalam webstorage
System	Conditions	
	<i>Compare two values</i>	Membandingkan beberapa ekspresi logika
	<i>Every tick</i>	Kondisi yang selalu "true"
	<i>Is between values</i>	Mengecek jika suatu angka berada di antara dua nilai
	<i>For, For each, While</i>	Kondisi pengulangan
	<i>Repeat</i>	Mengulang fungsi tertentu
	<i>Else</i>	Menjalankan <i>event</i> ketika <i>event</i> sebelumnya tidak dijalankan
	<i>On start of layout</i>	kondisi ketika <i>layout</i> mulai dijalankan
	<i>On end of layout</i>	Kondisi ketika <i>layout</i> ditutup
	<i>Compare time</i>	Membandingkan waktu
	Actions	
	<i>Create object</i>	Membuat "new instance" dari sebuah objek
	<i>Go to layout</i>	Aksi untuk menuju <i>layout</i> tertentu
	<i>Restart layout</i>	Menjalankan kembali <i>layout</i> yang dimaksud
	<i>Stop loop</i>	Menghentikan perulangan
	<i>Reset global variables</i>	Reset nilai <i>global variables</i> ke nilai awal
	<i>Add to, set value, subtract from</i>	Penambahan nilai untuk lokal variabel atau global variabel
	<i>Set layer opacity</i>	Pengaturan fungsi transparan pada <i>layout</i>
	<i>Set layer visible</i>	Pengaturan aktif/tidak aktif <i>layout</i>

Manual Construct 2 selengkapnya dapat diunduh pada site
https://www.scirra.com/manual/1/construct-2?_e_pi_=7%2CPAGE_ID10%2C6633796663