

**PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING YANG
TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE**

TUGAS AKHIR SKRIPSI

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



Oleh:

Jilly Ratria Sari
NIM 10520240133

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2018

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING YANG TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE

Disusun Oleh:

Jilly Ratria Sari
NIM 10520241033

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

Yogyakarta, 19 Desember 2017

Mengetahui,

Disetujui,

Ketua Program Studi
Pendidikan Teknik Informatika,

Dosen Pembimbing,



Handaru Jati, Ph.D.
NIP. 19740511 199903 1 002



Handaru Jati, Ph.D.
NIP. 19740511 199903 1 002

HALAMAN PENGESAHAN

Tugas Akhir Skripsi




PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING YANG TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE

Disusun Oleh:

Jilly Ratria Sari
NIM 10520241033

Telah dipertahankan di depan tim penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta
pada tanggal 3 Januari 2018.

TIM PENGUJI

Nama/Jabatan	Tanda Tangan	Tanggal
Handaru Jati, Ph.D. Ketua Penguji/Pembimbing		17 Januari 2018
Nur Hasanah, M.Cs. Sekretaris		17 Januari 2018
Nurkhamid, Ph.D. Penguji		17 Januari 2018

Yogyakarta, 18 Januari 2018

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,



Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001

HALAMAN PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Jilly Ratria Sari

NIM : 10520241033

Program Studi: Pendidikan Teknik Informatika

Judul TAS : Pengembangan Perangkat Lunak Ujian Daring yang
: Terintegrasi dengan Layanan Percakapan LINE

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

Yogyakarta, 29 November 2017

Yang menyatakan,



Jilly Ratria Sari

NIM. 10520241033

MOTTO

"The one who matter the most is yourself."

"Everyone you meet is fighting a battle you know nothing about. Be kind.

Always."

HALAMAN PERSEMBAHAN

I can not thank more to myself who still believe and want to finish this thing.

Congratulation, Jilly. You did (almost) a good job.

To my parents, I'm so sorry for making this so late.

*I would find another thing to make you proud of me, but I hope, for this time,
this is enough.*

PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING YANG TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE

Oleh:

Jilly Ratria Sari
NIM. 10520241033

ABSTRAK

Penelitian bertujuan (1) mengembangkan perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE, (2) mengetahui kualitas dari perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE berdasarkan standar pengujian kualitas perangkat lunak ISO/IEC 25010 pada aspek *functional suitability*, *performance efficiency*, *usability*, *security*, dan *compatibility*.

Metode yang digunakan adalah *Research and Development* dengan *waterfall* sebagai model pengembangan perangkat lunak yang terdiri dari tahapan analisis kebutuhan, desain, pengodean dan pengujian. Pengembangan perangkat lunak menggunakan ExpressJS dan AngularJS.

Hasil dari penelitian ini adalah 1) perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE yang memungkinkan pengguna LINE mengakses daftar ujian, melihat informasi melalui detail ujian, mengerjakan soal sesuai dengan ujian yang dipilih, mengetahui nilai ujian setelah berhasil mengumpulkan, dan melihat peringkat diri dibanding semua peserta serta pembahasan soal setelah masa ujian telah selesai, dan 2) hasil pengujian menunjukkan perangkat lunak telah memenuhi standar ISO/IEC 25010 pada aspek (1) *functional suitability* fungsi berjalan 100% (telah memenuhi standar), (2) *performance efficiency* dengan rata-rata waktu untuk memuat halaman 1,8 detik (memenuhi standar), (3) *usability* diperoleh nilai alpha cronbach 0,864 (good) dengan presentase sebesar 82,00% (sangat layak), (4) *security* dengan tingkat keparahan keamanan level 1 (tingkat keparahan keamanan rendah), dan (5) *compatibility*, pengujian tidak dilakukan karena tidak ada *resources* yang dibagi bersama perangkat keras dan perangkat lunak lain.

Kata kunci: ujian daring , ujian online, *mobile web app*, ISO/IEC 25010

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiratan Yang Maha Kuasa yang telah melimpahkan kasih dan sayang-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengembangan Perangkat Lunak Ujian Daring yang Terintegrasi dengan Layanan Percakapan LINE".

Maksud dari penyusunan skripsi ini adalah untuk memenuhi salah satu syarat dalam menempuh ujian sarjana pendidikan pada Fakultas Teknik Program Studi Pendidikan Teknik Informatika di Universitas Negeri Yogyakarta.

Dalam penyusunan skripsi ini, banyak pihak yang sangat membantu penulis dalam berbagai hal. Oleh karena itu, penulis sampaikan rasa terima kasih yang sedalam-dalamnya kepada:

1. Handaru Jati, Ph.D selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi ini
2. Muhammad Iman Usman, Alvin Francis Tamie, dan Surya Pradhana selaku validator instrumen TAS yang memberikan saran/masukan perbaikan sehingga penelitian TAS dapat terlaksana sesuai dengan tujuan.
3. Handaru Jati, Ph.D., Nur Hasanah, M.Cs., Nurkhamid, Ph.D, selaku Ketua Penguji, Sekretaris dan Penguji yang memberikan koreksi perbaikan secara komprehensif terhadap TAS ini.
4. Dr. Fatchul Arifin, S.T, M.T. selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Handaru Jati, Ph.D selaku Ketua Program Studi Pendidikan Teknik Informatika beserta dosen dan staf yang telah memberikan bantuan dan

fasilitas selama proses penyusunan pra proposal sampai dengan selesainya Tugas Akhir Skripsi ini

5. Dr. Widarto, M.Pd., selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang memberikan persetujuan pelaksanaan Tugas Akhir Skripsi
6. Dr. Eko Marpanaji, selaku Dosen Penasihat Akademik yang selalu memberikan arahan dan bimbingan dalam menempuh studi ini
7. Ruangguru dan tim yang memberikan saya kesempatan untuk mengangkat salah satu produknya untuk Tugas Akhir Skripsi ini
8. Tim Tech Ruangguru yang selalu menyediakan bantuan dan dukungan untuk saya menyelesaikan Tugas Akhir Skripsi
9. Lissa, Zum, Mas Banu, Kak Ripin, Dayan, Mirza, Thoriq yang memberikan dukungan dan menjadi panutan untuk menyelesaikan Tugas Akhir Skripsi ini
10. Semua pihak, secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu atas bantuan dan perhatiannya selama penyusunan Tugas Akhir Skripsi ini

Yogyakarta, 29 November 2017

Penulis,



Jilly Ratria Sari

NIM 10520241033

DAFTAR ISI

LEMBAR PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	iv
MOTTO	v
HALAMAN PERSEMBAHAN.....	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Identifikasi Masalah	4
C. Batasan Masalah	4
D. Rumusan Masalah	5
E. Tujuan Penelitian	5
F. Spesifikasi Produk yang Dikembangkan.....	5
G. Manfaat Penelitian.....	6
BAB II KAJIAN PUSTAKA.....	7
A. Kajian Teori.....	7
1. Ujian Daring	7

2. Ruangguru	7
3. LINE	8
4. Perangkat Lunak.....	9
5. <i>Software Development Life Cycle</i> (SDLC)	9
6. Kualitas Perangkat Lunak.....	15
B. Hasil Penelitian yang Relevan	21
C. Kerangka Pikir	23
D. Pertanyaan Penelitian.....	24
BAB III METODE PENELITIAN	25
A. Model Pengembangan	25
B. Prosedur Pengembangan.....	25
1. Analisis Kebutuhan.....	25
2. Desain.....	26
3. Pengodean	26
4. Pengujian	26
C. Subjek Penelitian	28
D. Metode Pengumpul Data	28
1. Observasi	28
2. Wawancara	28
3. Angket	28
4. Pengukuran Perangkat Lunak.....	28
E. Instrumen Penelitian	29
1. Instrumen <i>Functional Suitability</i>	29
2. Instrumen <i>Performance Efficiency</i>	31

3. Instrumen <i>Usability</i>	32
4. Instrumen <i>Security</i>	33
5. Instrumen <i>Compatibility</i>	34
F. Teknik Analisis Data	34
1. Analisis Faktor Kualitas <i>Functional Suitability</i>	34
2. Analisis Faktor Kualitas <i>Performance Efficiency</i>	35
3. Analisis Faktor Kualitas <i>Usability</i>	37
4. Analisis Faktor Kualitas <i>Security</i>	40
5. Analisis Faktor Kualitas <i>Compatibility</i>	40
BAB IV HASIL DAN PEMBAHASAN.....	41
A. Tahap Analisis Kebutuhan	41
1. Analisis Kebutuhan Fungsional	41
2. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak	42
B. Tahap Desain	42
1. Desain UML (<i>Unified Modeling Language</i>)	42
2. Desain Antarmuka.....	51
C. Tahap Pengodean (Implementasi)	60
1. Implementasi Antarmuka.....	60
2. Implementasi Sistem.....	71
D. Tahap Pengujian	72
1. Hasil Pengujian <i>Functional Suitability</i>	72
2. Hasil Pengujian <i>Performance Efficiency</i>	76
3. Hasil Pengujian <i>Usability</i>	85
4. Hasil Pengujian <i>Security</i>	88

5. Hasil Pengujian <i>Compatibility</i>	91
E. Pembahasan Hasil Penelitian.....	91
1. Pembahasan Hasil Pengujian <i>Functional Suitability</i>	91
2. Pembahasan Hasil Pengujian <i>Performance Efficiency</i>	92
3. Pembahasan Hasil Pengujian <i>Usability</i>	92
4. Pembahasan Hasil Pengujian <i>Security</i>	92
5. Pembahasan Hasil Pengujian <i>Compatibility</i>	92
BAB V KESIMPULAN DAN SARAN	93
A. Kesimpulan	93
B. Keterbatasan Produk	94
C. Pengembangan Produk Lebih Lanjut.....	94
D. Saran.....	94
DAFTAR PUSTAKA.....	95
LAMPIRAN	98

DAFTAR GAMBAR

Gambar 1. Ilustrasi Model Waterfall (Rosa dan Shlahudin, 2013)	10
Gambar 2. Diagram Tipe-Tipe UML (Fowler, 2013)	12
Gambar 3. Model Kualitas Produk ISO/IEC 25010 (Wagner, 2013)	16
Gambar 4. Diagram Dimensions of Quality Pressmann (Pressmann, 2015)	17
Gambar 5. Diagram Irisan Kualitas ISO dan Dimensions of Quality Pressmann..	18
Gambar 6. Kerangka Pikir	23
Gambar 7. Use Case Diagram Sistem Perangkat Lunak Ujian Daring	43
Gambar 8. Activity Diagram Login	44
Gambar 9. Activity Diagram Memilih Tryout Dibuka.....	46
Gambar 10. Activity Diagram Memilih Tryout Ditutup	47
Gambar 11. Sequence Diagram Melakukan Login.....	48
Gambar 12. Sequence Diagram Memilih Tryout	49
Gambar 13. Sequence Diagram Memilih Paket Soal.....	49
Gambar 14. Sequence Diagram Melakukan Ujian	50
Gambar 15. Sequence Diagram Melihat Hasil.....	51
Gambar 16. Desain Antarmuka Halaman Login	52
Gambar 17. Desain Antarmuka Halaman Daftar Tryout	53
Gambar 18. Desain Antarmuka Halaman Detail Tryout Dibuka	54
Gambar 19. Desain Antarmuka Halaman Detail Tryout Ditutup	54
Gambar 20. Desain Antarmuka Halaman Detail Paket Soal	55
Gambar 21. Desain Antarmuka Halaman Pengerjaan Ujian	56
Gambar 22. Desain Antarmuka Halaman Mengakhiri Ujian.....	57

Gambar 23. Desain Antarmuka Halaman Hasil Ujian (Tryout Dibuka)	58
Gambar 24. Desain Antarmuka Halaman Hasil Ujian (Tryout Ditutup).....	58
Gambar 25. Desain Antarmuka Halaman Pembahasan Ujian	59
Gambar 26. Desain Antarmuka Halaman Peringkat Ujian.....	60
Gambar 27. Implementasi Halaman Login	61
Gambar 28. Implementasi Halaman Daftar Tryout Dibuka	62
Gambar 29. Implementasi Halaman Daftar Tryout Ditutup	62
Gambar 30. Implementasi Halaman Detail Tryout Dibuka	63
Gambar 31. Implementasi Halaman Detail Tryout Ditutup	64
Gambar 32. Implementasi Halaman Detail Paket Soal	65
Gambar 33. Implementasi Halaman Pengerjaan Ujian	66
Gambar 34. Implementasi Halaman Mengakhiri Ujian	67
Gambar 35. Implementasi Halaman Hasil Ujian Tryout Dibuka.....	68
Gambar 36. Implementasi Halaman Hasil Ujian Tryout Ditutup	68
Gambar 37. Implementasi Halaman Pembahasan Jawaban Benar	69
Gambar 38. Implementasi Halaman Pembahasan Jawaban Salah.....	70
Gambar 39. Implementasi Halaman Peringkat	71
Gambar 40. Laporan GTMetrix Halaman Integrasi LINE.....	76
Gambar 41. Laporan GTMetrix Halaman Onboard	77
Gambar 42 Laporan GTMetrix Halaman Register (1)	77
Gambar 43. Laporan GTMetrix Halaman Register (2)	78
Gambar 44. Laporan GTMetrix Halaman Login.....	79
Gambar 45. Laporan GTMetrix Halaman Daftar Tryout	79
Gambar <u>46</u> . Laporan GTMetrix Halaman Detail Tryout - Daftar Paket Soal	80

Gambar 47. Laporan GTMetrix Halaman Hasil (Tryout Dibuka).....	81
Gambar 48. Laporan GTMetrix Halaman Hasil (Tryout Ditutup).....	82
Gambar 49. Laporan GTMetrix Halaman Pembahasan	82
Gambar 50. Laporan GTMetrix Halaman Peringkat.....	83
Gambar 51. Laporan GTMetrix Halaman Profil	84
Gambar 52. Hasil perhitungan Alpha Cronbach menggunakan SPSS.....	88
Gambar 53. Laporan Pengujian Security dengan Acunetix: Scan Stats & Info....	89
Gambar 54. Tampilan Pengujian Security dengan Acunetix: Vulnerabilities	89
Gambar 55. Tampilan Pengujian Security dengan Acunetix: Site Structure.....	90
Gambar 56. Tampilan Pengujian Security dengan Acunetix: Events.....	91

DAFTAR TABEL

Tabel 1. Instrumen Functional Completeness dan Functional Appropriateness ..	29
Tabel 2. Instrumen Functional Correctness.....	31
Tabel 3. Instrumen Usability	32
Tabel 4. Instrumen Security	34
Tabel 5. Aturan Audit YSlow (Performance Efficiency)	35
Tabel 6. Aturan Audit PageSpeed Insights (Performance Efficiency)	36
Tabel 7 Kategori Penilaian Usability (Guritno, Sudaryono & Rahardja, 2011)	39
Tabel 8 Konsistensi Alpha Cronbach (Gliem dan Gliem, 2003)	40
Tabel 9 Hasil Pengujian Subkarakteristik Functional Appropriateness.....	73
Tabel 10 Hasil Pengujian Subkarakteristik Functional Correctness	75
Tabel 11. Tabel Hasil Pengujian Performance Efficiency	85
Tabel 12. Hasil Pengujian Usability	86

DAFTAR LAMPIRAN

Lampiran 1. Surat Pernyataan Validasi Instrumen

Lampiran 2. Pengujian Aspek *Functional Suitability*

Lampiran 3. Pengujian Aspek *Usability*

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Menurut Tjahjo Kumolo, Menteri Dalam Negeri RI, jumlah penduduk Indonesia pada Juni 2016 telah mencapai 257.912.349 jiwa. Hal ini membuat Indonesia masih berada di peringkat ke empat populasi terbanyak di dunia (Purnama, 2017). Namun, jumlah penduduk yang besar tidak sejalan dengan kualitas manusianya. Salah satu indikatornya dapat dilihat dalam penelitian *Programme for International Student Assessment* (PISA) yang diselenggarakan oleh *Organization for Economic Co-operation and Development* (OECD) pada tahun 2015. Hasil penelitian ini, Indonesia berada di tingkatan bawah dari 45 negara di bidang ilmu pengetahuan, matematika dan membaca yang diteliti dalam PISA (OECD PISA, 2015).

Pada tahun 2014, Kementerian Komunikasi dan Informatika merilis siaran pers tentang riset Kementerian Komunikasi dan Informatika dan UNICEF mengenai perilaku anak dan remaja dalam menggunakan internet. Dari riset tersebut diketahui bahwa setidaknya 30 juta anak-anak dan remaja di Indonesia merupakan pengguna internet dan media digital. Dari riset tersebut diketahui juga ada tiga motivasi utama anak-anak dan remaja mengakses internet, yaitu: untuk mencari informasi, untuk terhubung dengan teman dan untuk hiburan. Selain anak-anak dan remaja, orang tua dan guru juga telah memanfaatkan kemajuan teknologi. Mereka memanfaatkan media digital untuk mendukung pendidikan dan pembelajaran anak, misalnya dengan menugaskan para siswa untuk mencari dan mengumpulkan informasi dari internet untuk mengerjakan berbagai tugas.

Sejalan dengan hal itu, ruangguru sebagai perusahaan digital di bidang pendidikan di Indonesia membuat produk dan layanan pendidikan dengan memanfaatkan teknologi. Sejumlah produk yang telah dirilis, antara lain: *ruangles*, *marketplace* pencarian guru privat, di mana Ruangguru menghubungkan calon murid dan calon guru; *ruanglesonline*, aplikasi *mobile* yang membantu murid untuk mendapatkan solusi permasalahan belajar secara instan; *ruangkkelas*, platform *learning management system* yang memungkinkan guru membuat kelas virtual yang berisi tugas-tugas dan murid dapat mengikuti kelas-kelasnya dengan undangan; *ruangbelajar*, ribuan video materi yang berisi penjelasan dari sebuah konsep pembelajaran dan atau pembahasan, latihan topik dan rangkuman; dan terakhir *ruanguji*, platform ujian daring yang digunakan sebagai persiapan dalam menghadapi UN, UAS, SBMPTN bahkan UKG. Di *ruanguji* pengguna dapat melihat pembahasan soal serta analisa tingkat kemampuannya (Ruangguru, 2017).

Dipaparkan di paragraf sebelumnya bahwa anak-anak muda dan remaja Indonesia adalah pengguna media digital dan sosial. Cukup banyak media sosial dan layanan percakapan yang populer di Indonesia, salah satu di antaranya adalah LINE, sebuah aplikasi layanan percakapan dari Jepang. *Managing director* LINE Indonesia, Ongki Kurniawan, menyebutkan pengguna LINE di Indonesia mencapai 90 juta dengan pengguna aktif bulanan sebesar 72 juta (Bohang, 2016). Dari keseluruhan pengguna LINE di Indonesia, 59% pengguna berada di rentang usia remaja (di bawah 17 tahun) hingga dewasa muda (23 tahun).

Sebagai upaya untuk mendekatkan jarak dengan penggunanya, LINE bekerja sama dengan Ruangguru membentuk layanan turunan LINE yang berisi konten-konten pembelajaran, yaitu LINE Academy. Di Indonesia, prestasi belajar siswa

secara periodik dipantau dengan melihat hasil ujian, baik di tengah semester, akhir semester, ujian kenaikan kelas maupun ujian akhir sekolah yang menentukan kelulusan. Dengan harapan membantu para pelajar, LINE dan Ruangguru memutuskan untuk membuat *platform* ruanguji yang dimiliki Ruangguru dapat diakses melalui aplikasi LINE. Aplikasi ujian daring ini diharapkan dapat memberikan media belajar baru dan membantu persiapan ujian untuk para pelajar pengguna LINE.

Setelah pengembangan, perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE belum diketahui bagaimana kualitasnya. Padahal dalam menerapkan berbagai teknologi dalam pengembangan perangkat lunak ujian daring ini membutuhkan waktu, tenaga, dan biaya yang tidak sedikit, sehingga akan menjadi percuma apabila perangkat lunak yang telah dikembangkan gagal ketika didistribusikan kepada pengguna. Sebuah aplikasi yang akan dirilis secara publik haruslah minim *bugs*, atau celah-celah kesalahan yang muncul selama proses pengembangan secara teknis maupun non teknik. Semakin baik kualitas sebuah perangkat lunak, semakin sedikit kesalahan yang ditemukan. Untuk mengetahui kualitas sebuah perangkat lunak dibutuhkan berbagai pengujian. Dengan pengujian yang melalui ukuran dan metode tertentu diharapkan diketahui kualitas suatu perangkat lunak. Salah satu standar pengujian perangkat lunak adalah ISO/IEC 25010. ISO/IEC 25010 dibuat oleh *International Organization for Standardization and International Electrotechnical Commission* sebagai revisi atau pengganti ISO/IEC 9126 (*International Organization for Standardization, 2011*). Dengan itu maka ISO/IEC 25010 merupakan standar terbaru dan yang paling

relevan untuk digunakan sebagai standar tolak ukur kualitas suatu perangkat lunak.

B. Identifikasi Masalah

Dari uraian latar belakang masalah di atas, dapat diidentifikasi beberapa masalah sebagai berikut:

1. Kualitas pendidikan di Indonesia cukup rendah
2. Penggunaan media sosial dan digital yang minim konten pembelajaran menjadi bagian yang menyatu dalam kehidupan sehari-hari anak-anak dan remaja di Indonesia
3. LINE, sebagai layanan percakapan yang merambah ke media sosial memiliki banyak pengguna di usia belajar, belum memiliki konten atau layanan yang dapat mendukung pembelajaran
4. Belum adanya perangkat lunak ujian daring yang dapat diakses di layanan percakapan LINE
5. Tidak semua perangkat lunak/aplikasi yang dipublikasi telah memenuhi standar perangkat lunak

C. Batasan Masalah

Ruang lingkup pada permasalahan yang ada pada penelitian ini cukup luas, sehingga perlu dilakukan pembatasan masalah. Permasalahan yang dibahas meliputi:

1. LINE, sebagai layanan percakapan yang merambah ke media sosial memiliki banyak pengguna di usia belajar, tidak memiliki *platform* ujian daring
2. Perangkat lunak ujian daring yang akan digunakan oleh publik tidak diketahui kualitasnya

D. Rumusan Masalah

Berdasarkan uraian latar belakang dan identifikasi masalah serta batasan masalah di atas, maka dirumuskan masalah sebagai berikut:

1. Bagaimana *platform* ujian daring dapat dikembangkan di layanan percakapan LINE?
2. Bagaimana kualitas perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE?

E. Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengembangkan *platform* ujian daring yang terintegrasi dengan layanan percakapan LINE
2. Untuk mengetahui kualitas perangkat lunak ujian daring dari ruangguru (ruanguji) yang terintegrasi dengan layanan percakapan LINE sebelum digunakan ke publik

F. Spesifikasi Produk yang Dikembangkan

Produk yang akan dikembangkan adalah perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE. Aplikasi *mobile web* ini memungkinkan pengguna LINE yang terdaftar dalam *channel* LINE Academy mengakses daftar ujian, melihat informasi melalui detail ujian, mengerjakan soal sesuai dengan ujian yang dipilih, mengetahui nilai ujian setelah berhasil mengumpulkan, dan melihat peringkat diri dibanding semua peserta serta pembahasan soal setelah masa ujian telah selesai. Aplikasi ujian daring yang

terintegrasi dengan layanan percakapan LINE ini merupakan turunan dari *platform* ujian daring dari ruangguru (ruanguji) yang telah dikembangkan sebelumnya.

G. Manfaat Penelitian

Penelitian diharapkan dapat memberikan manfaat, antara lain:

1. Manfaat Empiris

Manfaat empiris yang didapat penulis melalui penelitian ini adalah proses, hasil, dan hasil uji pengembangan aplikasi ujian daring yang terintegrasi dengan layanan percakapan LINE berdasarkan standar pengujian perangkat lunak

2. Manfaat Praktis

Manfaat praktis dari penelitian ini adalah terciptanya *platform* ujian daring untuk pelajar yang menggunakan layanan percakapan LINE

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. Ujian Daring

Definisi ujian menurut Kamus Besar Bahasa Indonesia (Tim Penyusun Kamus Pusat, 2013) adalah sesuatu yang dipakai untuk menguji mutu sesuatu (kepandaian, kemampuan, hasil belajar, dsb). Sedangkan daring adalah akronim dari dalam jaring(an), terhubung melalui jejaring komputer, internet, dsb.

Dari definisi-definisi di atas, dapat dituliskan bahwa ujian daring adalah sesuatu yang dipakai untuk menguji mutu kemampuan/hasil belajar yang terhubung melalui jejaring komputer / internet.

2. Ruangguru

Ruangguru adalah perusahaan digital di bidang pendidikan di Indonesia yang membuat produk dan layanan pendidikan serta menyediakan konten pendidikan dengan memanfaatkan teknologi. Sejumlah produk yang telah dirilis, antara lain: *ruangles*, *marketplace* pencarian guru privat, di mana Ruangguru menghubungkan calon murid dan calon guru; *ruanglesonline*, aplikasi *mobile* yang membantu murid untuk mendapatkan solusi permasalahan belajar secara instan; *ruangkelas*, platform *learning management system* yang memungkinkan guru membuat kelas virtual yang berisi tugas-tugas dan murid dapat mengikuti kelas-kelasnya dengan undangan; *ruangbelajar*, ribuan video materi yang berisi penjelasan dari sebuah konsep pembelajaran dan atau pembahasan, latihan topik dan rangkuman; dan terakhir *ruanguji*, platform ujian daring yang digunakan sebagai persiapan dalam

menghadapi UN, UAS, SBMPTN bahkan UKG. Di ruanguji pengguna dapat melihat pembahasan soal serta analisa tingkat kemampuannya (Ruangguru, 2017).

Produk layanan yang akan digunakan dalam penelitian ini adalah ruanguji. Ruanguji terdiri dari dua pengguna utama, yaitu: admin dan pengguna biasa. Admin bertugas untuk membuat soal, membuat paket soal dan membuat tryout. Sedangkan pengguna biasa dapat mengakses daftar tryout, melihat informasi melalui detail tryout, mengerjakan soal sesuai dengan paket soal yang dipilih, mengetahui nilai ujian setelah berhasil mengumpulkan dan melihat peringkat diri dibanding semua peserta serta pembahasan soal setelah masa ujian telah selesai. Ruanguji menyediakan layanan API untuk ke dua pengguna utamanya. Namun, dalam penelitian ini, aplikasi hanya menggunakan API untuk pengguna biasa yang kemudian hanya disebut pengguna.

3. LINE

LINE adalah sebuah aplikasi pengirim pesan instan gratis yang dapat digunakan pada berbagai platform seperti telepon cerdas, tablet dan komputer. LINE difungsikan dengan menggunakan jaringan internet sehingga pengguna LINE dapat melakukan aktivitas seperti mengirim pesan teks, mengirim gambar, video, pesan suara, dan lain lain. LINE diklaim sebagai aplikasi pengirim pesan instan terlaris di 42 negara (Wikipedia, 2017).

Selain sebagai aplikasi pengirim pesan, LINE juga memiliki beberapa layanan turunan, diantaranya LINE Channel. Layanan LINE Channel memungkinkan aplikasi dan layanan web non LINE digabungkan dengan LINE. Dengan demikian, pengguna LINE dapat ditawarkan berbagai layanan (LINE, 2017).

4. Perangkat Lunak

Dalam bukunya "*Software Engineering: A Practitioner's Approach*" (2015: 4), Roger S. Pressman menuliskan bahwa perangkat lunak adalah instruksi (program komputer) yang bila dieksekusi memberikan fitur, fungsi dan kinerja yang diinginkan; struktur data yang memungkinkan program memanipulasi informasi secara memadai; dan dokumen yang menggambarkan operasi dan penggunaan program.

Dari buku yang sama didapatkan tujuh kategori terkait perangkat lunak, diantaranya: *system software, application software, engineering and scientific software, embedded software, web-based software, product-line software*, dan *artificial intelligence software*.

Perangkat lunak yang dikembangkan dalam penelitian ini adalah *web-based software* atau bisa disebut dengan *web-apps*. *Web-based software* adalah perangkat lunak berbasis *website*.

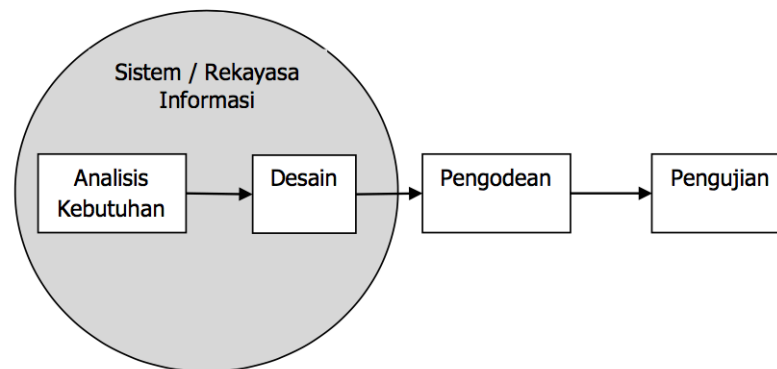
5. **Software Development Life Cycle (SDLC)**

Software Development Life Cycle adalah proses pengembangan atau mengubah suatu sistem perangkat lunak menggunakan metode-metode atau metodologi yang digunakan untuk mengembangkan sistem perangkat lunak sebelumnya (Rosa dan Shlahudin, 2013, p.26). Ada beberapa model SDLC, di antaranya: *waterfall model, spiral model* dan *incremental/iterative model*.

SDLC yang digunakan dalam penelitian ini adalah *waterfall model*. Diketahui bahwa produk yang akan dikembangkan merupakan *porting* dari *platform* ruanguji yang telah ada ke *platform* baru yang terintegrasi dengan layanan percakapan

LINE, jadi sebaiknya menggunakan SDLC dengan *waterfall model* (Alshamrani, & Bahattab, 2015).

Model *waterfall* disebut juga *classic life cycle* atau sekuensial linear merupakan model pengembangan perangkat lunak yang tahapannya dilakukan secara terurut atau sekuensial. Tahapan pengembangan perangkat lunak menggunakan model *waterfall* adalah sebagai berikut:



Gambar 1. Ilustrasi Model Waterfall (Rosa dan Shlahudin, 2013)

a. Analisis kebutuhan

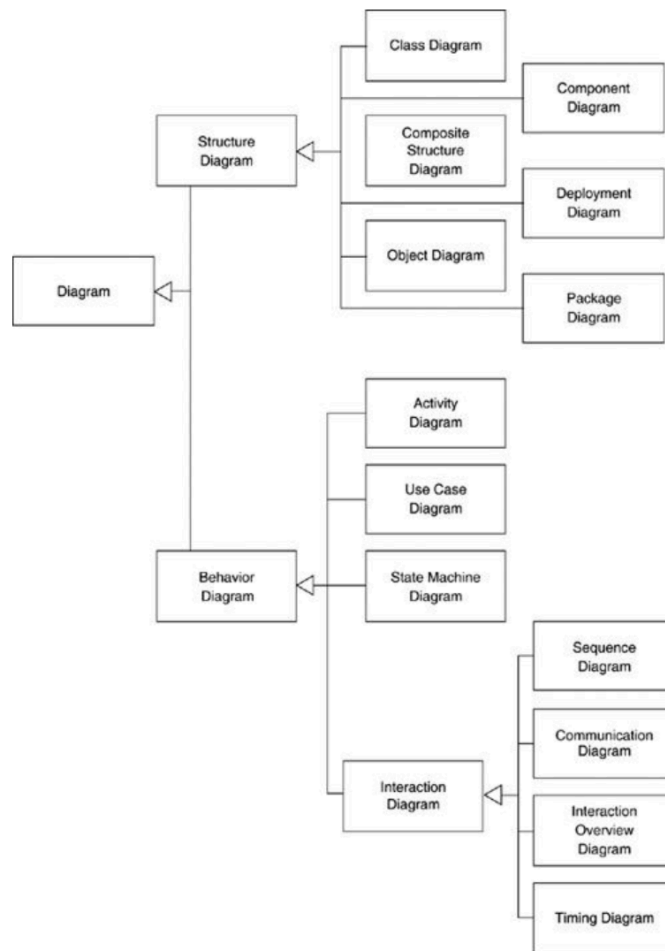
Menganalisis dan mendokumentasikan spesifikasi kebutuhan perangkat lunak agar perangkat lunak sesuai dengan apa yang dibutuhkan pengguna.

b. Desain

Pada tahap ini spesifikasi kebutuhan yang telah dianalisis diterjemahkan ke representasi desain seperti struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengodean agar mudah diimplementasikan ke tahap selanjutnya. Bahasa pemodelan sistem yang digunakan dalam penelitian ini adalah UML (*Unified Modelling Language*). UML adalah sebuah bahasa pemodelan dapat berupa *pseudo-code*, *actual code*, gambar, diagram atau bagian panjang deskripsi yang membantu

untuk menjabarkan sebuah sistem (Hamilton dan Miles, 2006). Sedangkan menurut Fowler (2003) dalam bukunya "UML Distilled 3rd Edition", UML adalah keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP).

UML mempunyai banyak tipe diagram yang dibagi menjadi dua kategori, yaitu kategori diagram yang mempresentasikan struktur informasi dan yang lain mempresentasikan *behaviour* (sifat). Diagram tipe-tipe diagram UML dapat dilihat pada Gambar 2.



Gambar 2. Diagram Tipe-Tipe UML (Fowler, 2013)

Desain UML yang digunakan dalam penelitian ini adalah *use case diagram*, *activity diagram*, dan *sequence diagram*.

1) *Use case diagram*

Use case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

2) *Activity diagram*

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. *Activity diagram* memungkinkan siapa pun yang melakukan proses untuk memilih urutan dalam melakukannya. Dengan kata lain diagram hanya menyebutkan aturan-aturan rangkaian dasar yang harus diikuti. Hal ini penting untuk pemodelan bisnis karena proses-proses sering muncul secara paralel. Ini juga berguna pada algoritma yang bersamaan, di mana urutan-urutan independen dapat melakukan hal-hal secara paralel.

3) *Sequence diagram*

Sequence diagram menunjukkan bagaimana kelompok-kelompok obyek saling berkolaborasi dalam beberapa *behaviour*. *Sequence diagram* secara khusus menjabarkan *behaviour* sebuah skenario tunggal yang menunjukkan sejumlah obyek.

c. Pengodean

Pada tahap pengodean atau implementasi ini, desain yang telah dibuat diterjemahkan menjadi perangkat lunak. Perangkat lunak dibantu dengan penyedia data dan bahasa pemrograman. Penyedia data yang digunakan dalam penelitian ini adalah API ruanguji (ujian daring yang disediakan oleh ruangguru). API adalah cara dua aplikasi komputer untuk berkomunikasi melalui internet menggunakan bahasa yang keduanya mengerti (Jacobson, dkk. 2012:5).

Dalam penelitian ini, jenis API yang digunakan adalah RESTful API, dengan kembalian data berbentuk JSON. *JSON is a standard for representing*

simple data structures in plain text (Richardson & Amundsen, 2013). Diterjemahkan menjadi JSON adalah standar untuk menggambarkan struktur data sederhana dalam teks biasa. JSON menggunakan tanda petik untuk menggambarkan *string* (contoh: "ini adalah *string*"), menggunakan tanda kurung siku untuk menggambarkan *lists* (contoh: [1, 2, 3]) dan menggunakan tanda kurung kurawal untuk menggambarkan *objects* (contoh: {"key": "value"}).

Dalam API web, protokol yang digunakan adalah HTTP. *API Client* dapat berinteraksi dengan penyedia API dengan mengirimkan beberapa jenis pesan HTTP yang berbeda. Ada delapan jenis pesan yang digunakan dalam komunikasi, namun hanya empat yang paling sering digunakan, yaitu: GET, POST, UPDATE dan DELETE.

Bahasa pemrograman yang digunakan adalah ExpressJs dan AngularJS. Express.js, atau hanya Express, adalah kerangka aplikasi web untuk Node.js, yang dirilis sebagai perangkat lunak bebas dan open-source di bawah Lisensi MIT (Wikipedia, 2017). Sedangkan, AngularJS adalah kerangka kerja aplikasi open-source front-end berbasis JavaScript yang dikelola oleh Google dan oleh komunitas individu dan perusahaan untuk mengatasi banyak tantangan yang dihadapi dalam pengembangan aplikasi satu halaman (Wikipedia, 2017).

d. Pengujian

Perangkat lunak yang telah selesai dikembangkan diuji untuk mengurangi kesalahan yang mengakibatkan perangkat lunak tidak sesuai dengan kebutuhan pengguna.

6. Kualitas Perangkat Lunak

Tujuan dari pembuatan perangkat lunak adalah dapat menciptakan suatu perangkat lunak yang berkualitas. Oleh karena itu, kualitas merupakan salah satu faktor penting dalam pengembangan suatu produk perangkat lunak (Farooq & Azmat, 2009).

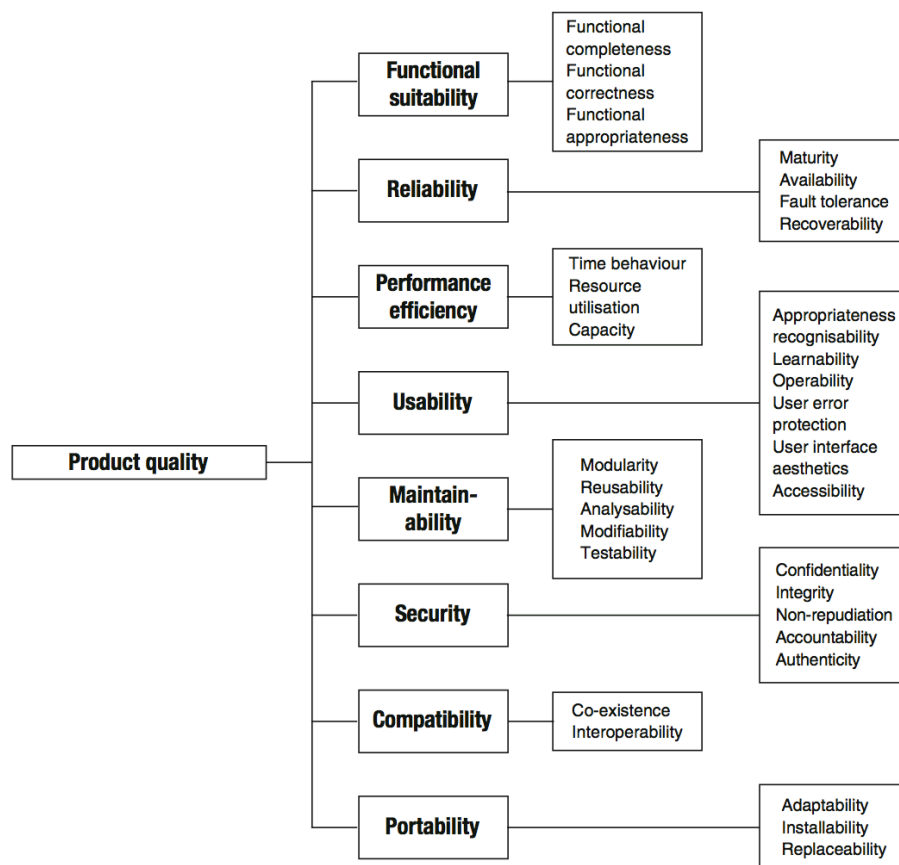
Kualitas adalah karakteristik atau atribut dari sesuatu di mana karakteristik atau atribut tersebut dapat diukur dan dapat dibandingkan dengan suatu standar yang ada. *The Institute of Electrenical and Electronic Engineers* (IEEE) mendefinisikan kualitas sebagai tingkatan pada sistem, komponen, atau proses yang sesuai dengan kebutuhan pengguna (Farooq & Azmat, 2009).

Pengertian di atas memberikan definisi bahwa untuk dapat menilai kualitas perangkat lunak didasarkan pada karakteristik perangkat lunak itu sendiri dengan dibandingkan pada suatu standar pengukuran dan didasarkan pada aspek pemenuhan kebutuhan pengguna perangkat lunak. Dari penjelasan tersebut dapat dipahami bahwa dalam menilai atau melakukan analisis kualitas perangkat lunak dilakukan dengan pengujian terhadap perangkat lunak tersebut dan terhadap penggunaannya.

Ada beberapa standar pengukuran kualitas perangkat lunak, di antaranya: McCall, Boehm, FURPS, Dromey, BBN, Kazman, Star, *International Organization of Standardization* (ISO), dan IEEE. Standar yang paling terkenal dan paling banyak digunakan dari kualitas perangkat lunak adalah model kualitas ISO/IEC 9126, yang sekarang telah digantikan oleh ISO/IEC 25010 (Wagner, 2013). Oleh karena itu,

penelitian ini menggunakan model ISO 25010 dalam melakukan analisis perangkat lunak.

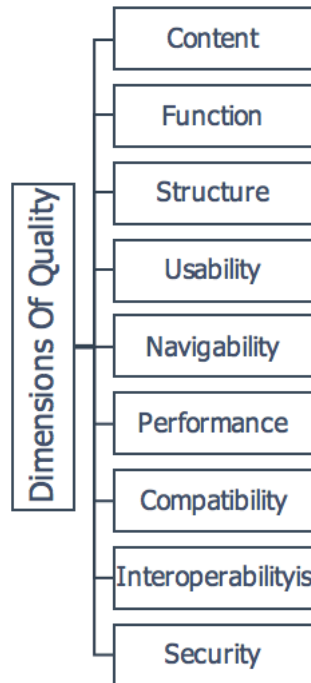
Pada ISO/IEC 25010 terdapat karakteristik tambahan dan subkarakteristik yang dipindahkan ke karakteristik lain. Standar pengujian ini dianggap sebagai standar pengujian perangkat lunak yang sesuai dengan perubahan teknologi dan informasi saat ini. Karakteristik model ISO/IEC 25010 dapat dilihat di diagram pada Gambar 3 di bawah ini:



Gambar 3. Model Kualitas Produk ISO/IEC 25010 (Wagner, 2013)

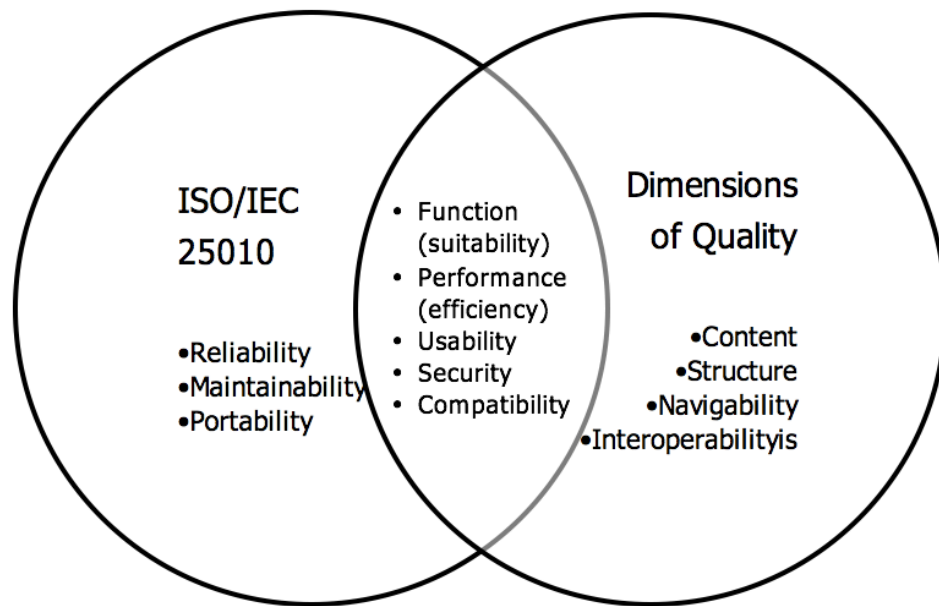
Pressmann mengatakan pengujian adalah proses yang bertujuan untuk menemukan dan memperbaiki kesalahan. Untuk memahami tujuan pengujian dalam konteks rekayasa *website* diperlukan pertimbangan beberapa dimensi

kualitas *webapp* (2015). Dimensi kualitas (*Dimensions of quality*) menurut Pressmann dapat dilihat dalam diagram pada Gambar 4 di bawah ini:



Gambar 4. Diagram *Dimensions of Quality Pressmann* (Pressmann, 2015)

Keterbatasan waktu dan sumber daya adalah alasan utama penulis melakukan pengujian kualitas perangkat lunak menggunakan ISO/IEC 25010 yang ada dalam *Dimensions of Quality*-nya Pressmann, yaitu *functional suitability*, *performance efficiency*, *usability*, *security* dan *compatibility*. Irisan aspek-aspek tersebut dapat dilihat dalam diagram *venn* pada Gambar 5 berikut ini:



Gambar 5. Diagram Irisan Kualitas ISO dan Dimensions of Quality Pressmann

a. Functional suitability

Sejauh mana perangkat lunak mampu menyediakan fungsi yang memenuhi kebutuhan yang dapat digunakan dalam kondisi tertentu. Karakteristik ini dibagi menjadi beberapa sub, yaitu:

- 1) Functional completeness, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik
- 2) *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan
- 3) *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

b. Performance efficiency

Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi beberapa sub yaitu:

- 1) *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi
- 2) *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- 3) *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan

c. *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh pengguna tertentu untuk mencapai tujuan dengan efektif, efisien, dan kepuasan tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa sub, yaitu:

- 1) *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
- 2) *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu
- 3) *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol
- 4) *User error protection*, sejauh mana produk atau sistem melindungi pengguna dalam membuat kesalahan

- 5) *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna
- 6) *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

Penentuan jumlah sampel responden untuk penelitian ini pada pengujian *usability* mengacu pada Jakob Nielsen (2012) yang mengemukakan bahwa uji pengguna untuk penelitian kuantitatif setidaknya paling sedikit 20 responden.

d. *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa sub, yaitu:

- 1) *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses
- 2) *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- 3) *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.

- 4) *Accountability*, sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim

e. *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan *environment* perangkat lunak yang sama. Karakteristik ini dibagi menjadi dua sub, yaitu:

- 1) *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.
- 2) *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut.

B. Hasil Penelitian yang Relevan

Beberapa penelitian relevan dengan penelitian ini, antara lain:

1. Pengembangan dan Analisis Kualitas Aplikasi Simulasi dan Pembahasan Ujian Nasional untuk Siswa Sekolah Menengah Pertama (SMP) oleh Neutrina Nilamsari (2014). Penelitian ini bertujuan untuk mengembangkan dan menganalisis kualitas aplikasi simulasi dan pembahasan ujian nasional. Hasilnya, aplikasi android yang dikembangkan layak untuk digunakan oleh siswa Sekolah Menengah Pertama (SMP). Relevansi dapat dilihat dari segi pengembangan ujian daring (*online*). Perbedaannya dengan penelitian ini

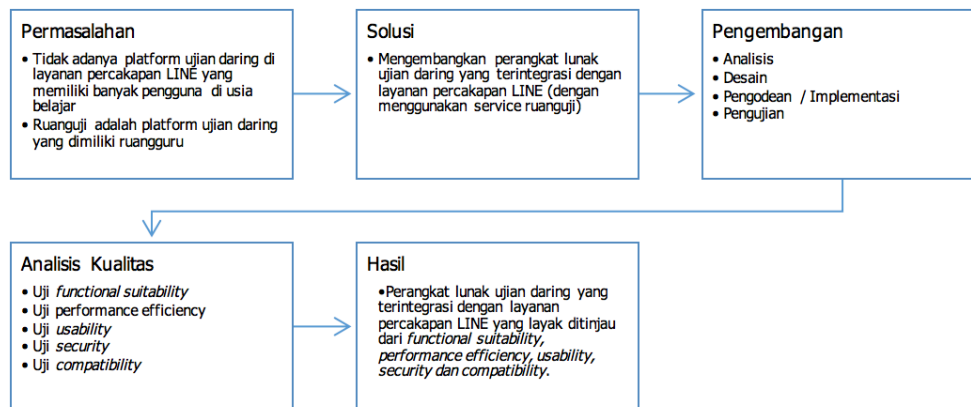
adalah Neutrina mengembangkan aplikasi berbasis android pada penelitiannya, sedangkan penelitian ini berbasis *web apps*, yang lebih *portable*.

2. Pengembangan Sistem Ujian Online Berbasis Web pada Mata Pelajaran Teknik Listrik di Sekolah Menengah Kejuruan Yogyakarta oleh Febrianto (2016). Penelitian ini bertujuan untuk merancang dan membuat perangkat tes online untuk Sekolah Menengah Kejuruan (SMK), dan mengetahui kualitas perangkat tes online berdasarkan standar ISO-9126 pada aspek *functionality*, *efficiency*, *usability*, *realibility* dan *portability*. Hasilnya, aplikasi berhasil dibuat dan memiliki kualitas yang baik dari semua aspek yang diujikan. Relevansi dapat dilihat dari segi pengembangan ujian daring yang berbasis *website*. Perbedaannya dengan penelitian ini adalah standar kualitas perangkat lunak yang digunakan, Febrianto menggunakan ISO-9126, sedangkan peneliti menggunakan ISO/IEC 25010 yang lebih baru.
3. Pengembangan Aplikasi Pengelolaan Data Prestasi Mahasiswa Universitas Negeri Yogyakarta oleh Yanuar Arifin (2015). Penelitian ini bertujuan untuk mengembangkan aplikasi pengelolaan data prestasi mahasiswa yang mampu menampilkan data prestasi mahasiswa tiap tahunnya dan mengetahui kualitas aplikasinya berdasarkan standar pengujian kualitas perangkat lunak ISO/IEC 25010. Hasilnya, aplikasi berhasil dibuat dan memiliki kualitas yang baik dari semua aspek yang diujikan. Relevansi dapat dilihat dari segi pengembangan yang berbasis *website* dan melakukan pengujian perangkat lunak dengan menggunakan standar ISO/IEC 25010. Perbedaannya dengan penelitian ini adalah spesifikasi penggunaannya, Arifin membuat aplikasi untuk pengelolaan

data prestasi mahasiswa UNY, sedangkan peneliti membuat aplikasi untuk publik yang menggunakan layanan percakapan LINE.

C. Kerangka Pikir

Kerangka pikir dijelaskan dari latar belakang permasalahan bahwa tidak adanya *platform* ujian daring di layanan percakapan LINE yang memiliki banyak pengguna di usia belajar. Berasal dari latar belakang tersebut dibuat alternatif pemecahan masalah dengan mengembangkan perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE. Pengembangan sistem menggunakan metode *waterfall* dengan tahapan analisis kebutuhan, desain, implementasi, dan pengujian. Setelah pengembangan selesai, dilakukan analisis kualitas dengan menggunakan model ISO 25010 meliputi aspek *functional suitability*, *performance efficiency*, *usability*, *security* dan *compatibility*. Kemudian tahap terakhir membuat kesimpulan dari hasil penelitian dan pengembangan yang didapatkan. Kerangka pikir secara sederhana dapat dilihat pada Gambar 6 berikut ini:



Gambar 6. Kerangka Pikir

D. Pertanyaan Penelitian

Pertanyaan penelitian dari pengembangan perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE sebagai berikut:

1. Bagaimana mengaplikasikan *platform* ujian daring yang telah tersedia ke dalam layanan percakapan LINE?
2. Apakah perangkat lunak ujian yang terintegrasi dengan layanan percakapan LINE memenuhi aspek *functional suitability*?
3. Apakah perangkat lunak ujian yang terintegrasi dengan layanan percakapan LINE memenuhi aspek *performance efficiency*?
4. Apakah perangkat lunak ujian yang terintegrasi dengan layanan percakapan LINE memenuhi aspek *usability*?
5. Apakah perangkat lunak ujian yang terintegrasi dengan layanan percakapan LINE memenuhi aspek *security*?
6. Apakah perangkat lunak ujian yang terintegrasi dengan layanan percakapan LINE memenuhi aspek *compatibility*?

BAB III

METODE PENELITIAN

A. Model Pengembangan

Pengembangan perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE merupakan penelitian *Research and Development* (R&D). Penelitian R&D adalah penelitian yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut (Sugiyono, 2015).

Perangkat lunak ini merupakan *porting* dari *platform* ruanguji yang telah ada ke *platform* baru yang terintegrasi dengan layanan percakapan LINE. Menurut Alshamrani & Bahattab (2015), metode pengembangan yang sebaiknya digunakan adalah metode *waterfall* karena kebutuhan produk telah diketahui dengan sangat baik.

B. Prosedur Pengembangan

Prosedur pengembangan menggunakan model pengembangan perangkat lunak *waterfall* dengan tahapan seperti berikut:

1. Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan untuk mengumpulkan segala informasi mengenai sistem/perangkat lunak yang akan dikembangkan secara spesifik. Analisis kebutuhan meliputi analisis kebutuhan fungsionalitas, analisis kebutuhan perangkat keras, dan analisis kebutuhan perangkat lunak. Analisis kebutuhan diperlukan untuk mendapatkan kebutuhan spesifik yang diperlukan sesuai dengan kebutuhan pengguna dan *stakeholder*.

Proses pengumpulan informasi untuk analisis kebutuhan dilakukan dengan observasi, wawancara, studi literatur mengenai perangkat lunak dan standar

kualitas perangkat lunak. Observasi dilakukan dengan mengamati dan meneliti produk ruangguji yang telah ada, sedangkan wawancara dilakukan secara langsung kepada *Chief Product Officer* Ruangguru, *Product Lead Ruangguru* dan wawancara tidak langsung dengan tim *developer* LINE Indonesia.

2. Desain

Langkah ini akan menerjemahkan hasil dari analisis kebutuhan sistem menjadi sebuah rancangan perangkat lunak sebelum dilakukan proses penulisan kode. Penulis menggambarkan rancangan sistem informasinya dalam sebuah pemodelan dan rancangan antarmuka. Pemodelan yang digunakan adalah *Unified Modeling Language* (UML).

3. Pengodean

Tahap ini adalah tahap implementasi. Tahap yang menerjemahkan pemodelan, desain antar muka, desain pengalaman ke dalam kode agar menjadi satu aplikasi yang dapat digunakan oleh pengguna. Bahasa pemrograman yang digunakan adalah *framework* javascript (ExpressJS dan AngularJS) dan SASS.

4. Pengujian

Pada tahap ini dilakukan pengujian kualitas terhadap perangkat lunak yang telah dikembangkan sesuai dengan standar ISO/IEC 25010 untuk mengetahui kelayakan perangkat lunak tersebut digunakan oleh pengguna. Pengujian dilakukan pada aspek *functional suitability*, *performance efficiency*, *usability*, *security*, dan *compatibility*.

a. Pengujian *functional suitability*

Pengujian *functional suitability* menggunakan *checklist* pada *test case* yang berisi fungsi-fungsi berdasarkan kebutuhan-kebutuhan (tujuan fungsi) yang

telah terdefinisi. *Test case* berguna untuk memastikan tidak ada kesalahan dalam perangkat lunak yang dikembangkan. Pengujian ini dilakukan oleh responden ahli.

b. Pengujian *performance efficiency*

Pengujian *performance efficiency* dilakukan dengan menggunakan perangkat lunak yang memiliki fungsi mengukur performa dari perangkat lunak ujian daring yang dikembangkan. Pengujian dilakukan dengan mengukur kecepatan akses, kecepatan *request* dan *response* data.

c. Pengujian *usability*

Pengujian *usability* menggunakan instrumen USE *questionnaire* dari Arnold M. Lund (2001). Instrumen tersebut telah banyak dipakai untuk melakukan pengujian sehingga sudah terjamin valid. Instrumen USE *questionnaire* terbagi menjadi 4 kriteria yaitu *usefulness*, *ease of use*, *ease of learning*, dan *satisfaction*.

d. Pengujian *security*

Pengujian *security* menggunakan perangkat lunak khusus untuk menguji kualitas keamanan aplikasi *web* terhadap *vulnerabilities*, *attack*, *threat* dan *countermeasures*.

e. Pengujian *compatibility*

Pengujian *compatibility* dilakukan dengan cara menjalankan produk, sistem atau komponen lain yang terdapat pada lingkungan yang sama secara bersamaan dengan perangkat lunak yang diuji untuk menguji efisiensi fungsi yang dibutuhkan ketika berbagi *resource* atau lingkungan perangkat lunak dengan produk, sistem atau komponen lain.

C. Subjek Penelitian

Subjek penelitian digunakan untuk menguji aspek *functional suitability* dan *usability* dari perangkat lunak yang dikembangkan. Pengujian *functional suitability* menggunakan lima responden yang merupakan ahli dalam pengembangan aplikasi website di Ruangguru (PT Ruang Raya Indonesia). Pengujian *usability* menggunakan 20 responden yang merupakan pengguna ujian daring yang terintegrasi dengan layanan percakapan LINE.

D. Metode Pengumpul Data

Metode pengumpul data yang digunakan dalam penelitian ini, antara lain:

1. Observasi

Observasi adalah melakukan pengamatan secara langsung kepada obyek penelitian untuk menganalisis kualitas perangkat lunak yang akan dikembangkan

2. Wawancara

Teknik ini merupakan teknik pengumpulan data dengan memberi beberapa pertanyaan kepada calon pengguna dan *stakeholder* untuk menentukan kebutuhan awal sistem dan menguatkan hasil kuesioner.

3. Angket

Angket atau kuesioner digunakan untuk mengumpulkan data dari aspek *functional suitability* dan *usability*.

4. Pengukuran Perangkat Lunak

Perangkat lunak pengukuran yang digunakan, antara lain: GTMetrix (*performance efficiency*), dan Acunetix Online Vulnerability Scanner 11 (*security*).

E. Instrumen Penelitian

Instrumen penelitian terdiri dari instrumen pengujian perangkat lunak berdasarkan aspek *functional suitability*, *performance efficiency*, *usability*, *security*, dan *compatibility*.

1. Instrumen *Functional Suitability*

Instrumen ini digunakan untuk mengukur sejauh mana suatu sistem menyediakan fungsi-fungsi yang telah ditetapkan. Instrumen ini berupa *checklist* pada *test case* yang berisi daftar fungsi aplikasi yang dijabarkan sesuai kebutuhan fungsional yang telah terdefinisi. Pengujian *test case* dilakukan oleh responden ahli di bidang pemrograman. *Test case* yang digunakan dalam penelitian ini dilampirkan pada Tabel 1 dan Tabel 2 berikut ini:

Tabel 1. Instrumen *Functional Completeness* dan *Functional Appropriateness*

No.	Fungsi	Hasil yang diharapkan
1.	Integrasi LINE	Fungsi <i>register</i> dan <i>login</i> otomatis dengan menggunakan akun LINE berjalan dengan benar
2.	<i>Onboard</i>	Fungsi untuk menampilkan kilasan tentang ujian daring yang terintegrasi dengan LINE berjalan dengan benar
3.	Daftar	Fungsi untuk mendaftar menjadi pengguna ujian daring berjalan dengan benar
4.	Masuk	Fungsi untuk masuk ke dalam dasbor aplikasi berjalan dengan benar
5.	Data daftar <i>tryout</i>	Fungsi untuk melihat daftar <i>tryout</i> berjalan dengan benar

No.	Fungsi	Hasil yang diharapkan
6.	Data detail <i>tryout</i>	Fungsi untuk melihat detail <i>tryout</i> yang berisi paket-paket ujian berjalan dengan benar
7.	Data detail paket soal	Fungsi untuk melihat detail paket soal berjalan dengan benar
8.	Mulai ujian	Fungsi untuk memulai ujian berjalan dengan benar
9.	Data soal	Fungsi untuk menampilkan soal-soal sesuai dengan paket data yang dipilih berjalan dengan benar
10.	Mengumpulkan soal	Fungsi untuk mengumpulkan soal berjalan dengan benar
11.	Data hasil ujian	Fungsi untuk menampilkan hasil ujian berjalan dengan benar
12.	Data pembahasan ujian	Fungsi untuk menampilkan pembahasan dari ujian yang telah dikerjakan berjalan dengan benar
13.	Data peringkat peserta	Fungsi untuk menampilkan peringkat pengguna dibanding dengan keseluruhan peserta yang mengerjakan ujian berjalan dengan benar
14.	Data profil	Fungsi untuk menampilkan data profil pengguna yang sedang aktif berjalan dengan benar
15.	Keluar	Fungsi untuk keluar dari dasbor sudah berjalan dengan benar

Tabel 2. Instrumen *Functional Correctness*

No	Fungsi	Hasil yang diharapkan
1.	Identifikasi <i>email</i> dan <i>password</i>	Fungsi untuk masuk sistem sesuai dengan pengguna sudah berjalan dengan benar
2.	Menampilkan data <i>tryout</i>	Fungsi untuk menampilkan data <i>tryout</i> sudah berjalan dengan benar
3.	Menampilkan detail <i>tryout</i>	Fungsi untuk menampilkan data detail <i>tryout</i> yang berisi paket-paket soal sudah berjalan dengan benar
4.	Menampilkan paket soal	Fungsi untuk menampilkan detail paket soal sudah berjalan dengan benar
5.	Memulai ujian	Fungsi untuk memulai ujian sudah berjalan dengan benar
6.	Menampilkan soal-soal	Fungsi untuk menampilkan soal-soal sesuai dengan paket ujian yang dipilih sudah berjalan dengan benar
7.	Mengumpulkan ujian	Fungsi untuk mengumpulkan ujian sudah berjalan dengan benar
8.	Menampilkan hasil ujian	Fungsi untuk menampilkan hasil ujian sudah berjalan dengan benar
9.	Menampilkan pembahasan	Fungsi untuk menampilkan pembahasan ujian sudah berjalan dengan benar
10.	Menampilkan peringkat	Fungsi untuk menampilkan peringkat peserta sudah berjalan dengan benar

2. Instrumen *Performance Efficiency*

Pengujian aspek *performance efficiency* menggunakan aplikasi GTMetrix. GTMetrix melakukan pengujian dengan berdasarkan kecepatan pemuatan

halaman sesuai dengan aturan YSlow dan PageSpeed. YSlow dikembangkan oleh Yahoo Developer Network, sedangkan PageSpeed dikembangkan oleh Google.

3. Instrumen *Usability*

Pengujian *usability* dilakukan dengan menggunakan angket USE Questionnaire oleh Arnold M. Lund yang berjumlah 30 pernyataan yang dibagi ke dalam empat kriteria, yaitu: *usefulness*, *easy of use*, *ease of learning*, dan *satisfaction*.

Instrumen USE Questionnaire terdapat pada Tabel 3 di bawah ini:

Tabel 3. Instrumen *Usability*

No	Instrumen
<i>Usefulness</i>	
1.	Aplikasi ini membantu saya menjadi lebih efektif dalam mengerjakan ujian
2.	Aplikasi ini membantu saya menjadi lebih produktif dalam belajar
3.	Aplikasi ini bermanfaat
4.	Aplikasi ini memberi saya dampak yang besar terhadap tugas yang saya lakukan dalam mengerjakan soal-soal ujian daring
5.	Aplikasi ini memudahkan saya mencapai hal-hal yang saya inginkan dalam mengerjakan ujian daring
6.	Aplikasi ini menghemat waktu ketika saya menggunakannya
7.	Aplikasi ini sesuai dengan kebutuhan saya
8.	Aplikasi ini bekerja sesuai dengan apa yang saya harapkan
<i>Easy of Use</i>	
9.	Aplikasi ini mudah digunakan
10.	Aplikasi ini praktis untuk digunakan
11.	Aplikasi ini mudah dipahami
12.	Aplikasi ini memerlukan langkah-langkah yang praktis untuk mencapai apa yang saya kerjakan
13.	Aplikasi ini dapat disesuaikan dengan kebutuhan
14.	Tidak kesulitan menggunakan aplikasi ini

Sambungan Tabel 3	
15.	Saya dapat menggunakan tanpa instruksi tertulis
16.	Saya tidak melihat adanya ketidakkonsistenan selama saya menggunakannya
17.	Pengguna yang jarang maupun rutin menggunakan akan menyukai sistem ini
18.	Saya dapat kembali dari kesalahan dengan cepat dan mudah
19.	Saya dapat menggunakan sistem ini dengan berhasil
<i>Easy of learning</i>	
20.	Saya belajar menggunakan aplikasi ini dengan cepat
21.	Saya mudah mengingat bagaimana cara menggunakan aplikasi ini
22.	Aplikasi ini mudah untuk dipelajari cara menggunakannya
23.	Saya cepat menjadi terampil dengan aplikasi ini
<i>Satisfaction</i>	
24.	Saya puas dengan aplikasi ini
25.	Saya akan merekomendasikan sistem informasi ini kepada teman
26.	Aplikasi ini menyenangkan untuk digunakan
27.	Aplikasi ini bekerja seperti yang saya inginkan
28.	Aplikasi ini sangat bagus
29.	Saya merasa saya harus memiliki aplikasi ini
30.	Aplikasi ini nyaman untuk digunakan.

4. Instrumen *Security*

Pengujian aspek *security* menggunakan aplikasi Acunetix Web Vulnerability Scanner. Adapun hal-hal yang diuji oleh aplikasi Acunetix, dapat dilihat pada Tabel 4 berikut:

Tabel 4. Instrumen *Security*

No.	Parameter
1	<i>Blind SQL Injection</i>
2	<i>Cross Site Scripting</i>
3	<i>Googling Hacking Database (GHDB)</i>
4	<i>Microsoft IIS tilde directory enumeration</i>
5	<i>SQL Injection</i>
6	<i>Weak password</i>
7	<i>Directory Traversal</i>
8	<i>Application error message</i>
9	<i>Script Source Code Disclosure</i>
10	<i>HTML Form without CSRF protection</i>
11	<i>User credentials are sent in clear text</i>
12	<i>ASP.NET version disclosure</i>
13	<i>Clickjacking: X-Frame-Options header missing</i>
14	<i>Cookie without Http Only flag set</i>
15	<i>Cookie without Secure flag set</i>
16	<i>Login page password-guessing attack</i>
17	<i>OPTIONS method is enabled</i>
18	<i>Broken links</i>
19	<i>Microsoft IIS version disclosure</i>
20	<i>Password type input with auto-complete enabled</i>

5. Instrumen Compatibility

Pengujian aspek *compatibility* tidak dilakukan karena perangkat keras atau lingkungan perangkat lunak hanya digunakan oleh aplikasi ujian daring ini dalam satu waktu.

F. Teknik Analisis Data

1. Analisis Faktor Kualitas *Functional Suitability*

Pengukuran untuk mengukur pengujian *functional suitability* menggunakan skala Guttman. Menurut Sugiyono, skala Guttman digunakan untuk mendapatkan jawaban yang tegas seperti "YA" dan "TIDAK". Hasil pengujian *functional suitability* dihitung menggunakan rumus dari matriks *Feature Completeness* (Acharya dan

Sinha, 2013). Matriks *Feature Completeness* adalah matriks untuk mengukur sejauh mana fitur yang ada di desain dapat benar-benar diimplementasikan. Rumus yang dimaksud sebagai berikut:

$$X = \frac{I}{P}$$

Keterangan :

P = Jumlah fitur yang dirancang

I = Jumlah fitur yang berhasil diimplementasikan

Dalam pengujian ini P adalah jumlah semua fungsi berdasarkan analisis kebutuhan, sedangkan I merupakan jumlah fungsi yang benar-benar berhasil diimplementasikan ke perangkat lunak. Interpretasi pengukuran yang digunakan berasal dari matriks *Feature Completeness* yaitu nilai yang mendekati 1 mengindikasikan banyaknya fitur yang berhasil diimplementasikan. Dalam pengujian ini perangkat lunak dikatakan baik dalam aspek *functional suitability* jika nilai X mendekati 1.

2. Analisis Faktor Kualitas *Performance Efficiency*

Analisis *performance efficiency* menggunakan aplikasi GTMetrix untuk mengukur *performance* dari perangkat lunak. Aturan yang digunakan oleh GTMetrix adalah YSlow dan PageSpeed Insights. Adapun aturan YSlow dapat dilihat pada Tabel 5 berikut:

Tabel 5. Aturan Audit YSlow (Performance Efficiency)

No	Aturan
1.	<i>Make fewer HTTP</i>
2.	<i>Use a Content Delivery Network</i>
3.	<i>Avoid empty src or href</i>
4.	<i>Add Expires headers</i>
5.	<i>Compress components with gzip</i>

6.	<i>Put CSS at top</i>
7.	<i>Put JavaScript at bottom</i>
8.	<i>Avoid CSS expressions</i>
9.	<i>Make JavaScript and CSS external</i>
10.	<i>Reduce DNS lookups</i>
11.	<i>Minify JavaScript and CSS</i>
12.	<i>Avoid URL redirects</i>
13.	<i>Remove duplicate JavaScript and CSS</i>
14.	<i>Configure entity tags</i>
15.	<i>Make AJAX cacheable</i>
16.	<i>Use GET for AJAX requests</i>
17.	<i>Reduce the number of DOM elements</i>
18.	<i>Avoid HTTP 404 (Not Found) error</i>
19.	<i>Reduce cookie size</i>
20.	<i>Use cookie-free domains</i>
21.	<i>Avoid Alphasort filter</i>
22.	<i>Do not scale images in HTML</i>
23.	<i>Make favicon small and cacheable</i>

Sedangkan aturan Pagespeed Insights dapat dilihat pada Tabel 6 seperti berikut:

Tabel 6. Aturan Audit PageSpeed Insights (*Performance Efficiency*)

No	Aturan
1.	<i>Specify a Vary: Accept-Encoding header</i>
2.	<i>Minify HTML</i>
3.	<i>Minify CSS</i>
4.	<i>Minify Javascript</i>
5.	<i>Avoid bad requests</i>
6.	<i>Avoid a character set in the meta tag</i>
7.	<i>Avoid landing page redirects</i>
8.	<i>Defer parsing of JavaScript</i>
9.	<i>Enable gzip compression</i>
10.	<i>Enable Keep-Alive</i>
11.	<i>Inline small CSS</i>
12.	<i>Inline small Javascript</i>
13.	<i>Leverage browser caching</i>
14.	<i>Minimize redirects</i>
15.	<i>Minimize request size</i>
16.	<i>Optimize images</i>

17.	<i>Optimize the order of styles and scripts</i>
18.	<i>Put CSS in the document head</i>
19.	<i>Remove query strings from static resources</i>
20.	<i>Serve resources from a consistent URL</i>
21.	<i>Serve scaled images</i>
22.	<i>Specify a cache validator</i>
23.	<i>Specify a character set early</i>
24.	<i>Specify image dimensions</i>
25.	<i>Avoid CSS @import</i>
26.	<i>Combine images using CSS sprites</i>
27.	<i>Prefer asynchronous resources</i>

Aturan-aturan di tabel-tabel di atas digunakan untuk menentukan skor *performance efficiency* dari perangkat lunak yang diuji, semakin tinggi skor maka semakin baik kualitas *performance efficiency* dari perangkat lunak tersebut.

3. Analisis Faktor Kualitas *Usability*

Pada pengujian *usability* pengukuran instrumen yang digunakan adalah skala Likert. Menurut Sugiyono, Skala Likert digunakan untuk mengukur sikap pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Jawaban setiap item instrumen yang menggunakan skala Likert mempunyai gradasi dari sangat positif sampai sangat negatif.

Skala Likert pada *USE questionnaire* dapat menggunakan 7 skala atau 5 skala. Penelitian ini menggunakan skala 5 seperti yang dilakukan oleh Muderedzwa & Nyakwende (2010) dalam penelitiannya di bidang teknologi informasi. Menurut Losby dan Wetmore (2012), skala 5 merupakan skala yang paling sering digunakan. Pada penelitian ini, jawaban yang disediakan berupa persetujuan terhadap item yang digunakan. Menurut Vagias (2006), jawaban untuk persetujuan atau ia sebut sebagai *Level of Agreement* sebagai berikut :

- a. Sangat Tidak Setuju
- b. Tidak Setuju
- c. Ragu-ragu
- d. Setuju
- e. Sangat Setuju

Untuk keperluan analisis kuantitatif, maka jawaban di atas dapat diberi skor (Sugiyono, 2015, p.93-94) sebagai berikut :

- a. Sangat Tidak Setuju (STS) diberi skor 1
- b. Tidak Setuju (TS) diberi skor 2
- c. Ragu-ragu (RR) diberi skor 3
- d. Setuju (S) diberi skor 4
- e. Sangat Setuju (SS) diberi skor 5

Data hasil dari pengujian *usability* dianalisis dengan menghitung rata-rata jawaban berdasarkan skor setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut :

$$Skor_{total} = (J_{SS} \times 5) + (J_S \times 4) + (J_{RR} \times 3) + (J_{TS} \times 2) + (J_{STS} \times 1)$$

Keterangan :

J_{SS} = jumlah responden menjawab Sangat Setuju

J_S = jumlah responden menjawab Setuju

J_{RR} = jumlah responden menjawab Ragu-ragu

J_{TS} = jumlah responden menjawab Tidak Setuju

J_{STS} = jumlah responden menjawab Sangat Tidak Setuju

Setelah skor total didapatkan kemudian mencari persentase skor untuk mendapatkan interpretasi hasil pengujian *usability* untuk kemudian dibandingkan dengan kategori penilaian pada Tabel 7, menggunakan rumus :

$$P_{skor} = \frac{Skor_{total}}{i \times r \times 5} \times 100\%$$

Keterangan :

$Skor_{total}$ = Skor total hasil responden menjawab

i = Jumlah Pertanyaan

r = Jumlah Responden

Tabel 7. Kategori Penilaian *Usability* (Guritno, Sudaryono & Rahardja, 2011)

Presentase Skor	Interpretasi
0% – 20%	Sangat Tidak Layak
21% – 40%	Tidak Layak
41% – 60%	Cukup Layak
61% – 80%	Layak
81% – 100%	Sangat Layak

Hasil pengujian *usability* juga dihitung nilai konsistensinya menggunakan *tool* SPSS dengan perhitungan *alpha cronbach*. Hasil dari perhitungan tersebut lalu dibandingkan dengan tabel konsistensi *alpha cronbach* pada Tabel 8 berikut ini :

Tabel 8. Konsistensi *Alpha Cronbach* (Gliem dan Gliem, 2003)

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

4. Analisis Faktor Kualitas *Security*

Pengujian *security* menggunakan aplikasi Acunetix Online Vulnerability Scanner sehingga hasil analisis data untuk aspek *security* diperoleh dari hasil pengujian menggunakan aplikasi Acunetix Online Vulnerability Scanner.

5. Analisis Faktor Kualitas *Compatibility*

Analisis data untuk *compatibility* tidak dilakukan karena tidak dilakukan pengujian.

BAB IV

HASIL DAN PEMBAHASAN

A. Tahap Analisis Kebutuhan

Tahap analisis kebutuhan adalah tahap awal yang dilakukan dalam penelitian pengembangan perangkat lunak, tujuannya adalah untuk menentukan hal-hal yang dibutuhkan dalam pengembangan sistem.

1. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisis fungsi yang diperlukan dalam aplikasi berdasarkan hasil observasi dan wawancara kepada *stakeholder* dan pengguna yang telah dilakukan. Fungsi yang diperlukan, meliputi:

- a. Pengguna dapat mendaftar jika belum memiliki akun
- b. Pengguna dapat masuk jika telah memiliki akun
- c. Pengguna dapat terdaftar/masuk secara otomatis melalui email LINE
- d. Pengguna dapat melihat daftar *tryout* yang sedang aktif
- e. Pengguna dapat memilih *tryout* yang ingin diikuti
- f. Pengguna dapat melihat detail *tryout* dan daftar paket-paket soal di dalamnya
- g. Pengguna dapat memilih paket soal yang ingin dikerjakan
- h. Pengguna dapat melihat detail paket soal dengan waktu pengerjaan yang disediakan dan jumlah soal yang tersedia
- i. Pengguna dapat mengerjakan ujian dengan navigasi yang mudah
- j. Pengguna dapat mengumpulkan ujian ketika telah selesai mengerjakan dan atau ketika waktu telah habis

- k. Pengguna dapat melihat hasil ujian sesaat setelah mengumpulkan ujian (dengan nilai, jumlah jawaban benar, jumlah jawaban salah, dan jumlah jawaban yang tidak dijawab)
- l. Pengguna dapat melihat pembahasan soal ketika tenggat waktu *tryout* telah habis
- m. Pengguna dapat melihat peringkatnya dan sepuluh besar peringkat peserta *tryout*
- n. Pengguna dapat melihat profil dirinya
- o. Pengguna dapat keluar dari aplikasi

2. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Dari hasil analisis kebutuhan untuk perangkat keras dan perangkat lunak, didapat hasil untuk dapat menjalankan aplikasi ini pengguna harus memiliki:

- a. Akun LINE
- b. Aplikasi LINE
- c. Terdaftar dalam *channel* LINE Academy
- d. *Smartphone* dengan internet

B. Tahap Desain

Dalam tahap ini, proses perancangan dilakukan yang meliputi Desain UML dan desain antarmuka.

1. Desain UML (*Unified Modeling Language*)

- a. *Use Case Diagram*

Perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE hanya memiliki satu aktor, yaitu pengguna. Pengguna

dapat mengakses semua fungsi di dalam perangkat lunak ujian daring. Adapun fungsi-fungsi di perangkat lunak ini, antara lain register, login, mengakses daftar ujian, melihat informasi melalui detail ujian, mengerjakan soal sesuai dengan ujian yang dipilih, mengetahui nilai ujian setelah berhasil mengumpulkan, dan melihat peringkat diri dibanding semua peserta serta pembahasan soal setelah masa ujian telah selesai.

Diagram *use case* menggambarkan bagaimana pengguna berhubungan dengan fungsi-fungsi di dalam sistem. Adapun diagramnya dapat dilihat pada Gambar 7 seperti berikut:



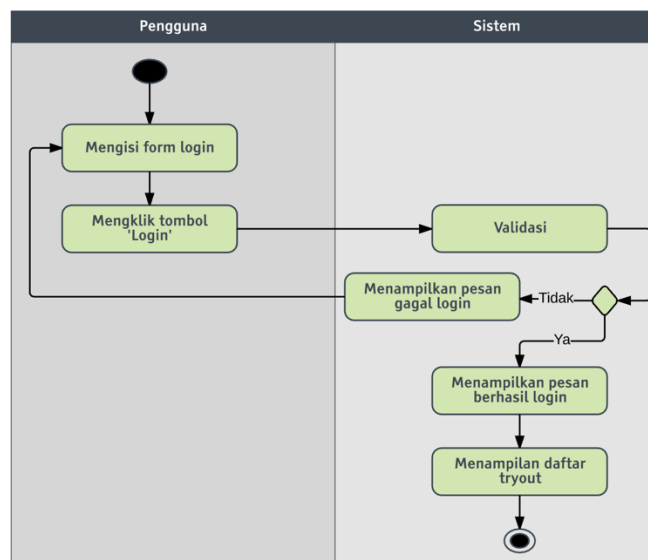
Gambar 7. *Use Case Diagram* Sistem Perangkat Lunak Ujian Daring

b. *Activity Diagram*

Berikut ini merupakan *activity diagram* pada perangkat lunak ujian daring:

1) Login

Gambar 9 menampilkan *activity diagram login*, awalnya pengguna disediakan *form* login yang berisi email dan password. Pengguna mengisi form login, kemudian memilih tombol 'Login'. Setelah itu sistem memvalidasi data dari form yang telah diisi pengguna. Jika valid, sistem akan menampilkan pesan berhasil login dan mengarahkan pengguna ke halaman daftar tryout. Namun, apabila validasi gagal, sistem akan menampilkan pesan gagal login dan meminta pengguna melakukan langkah dari awal lagi.



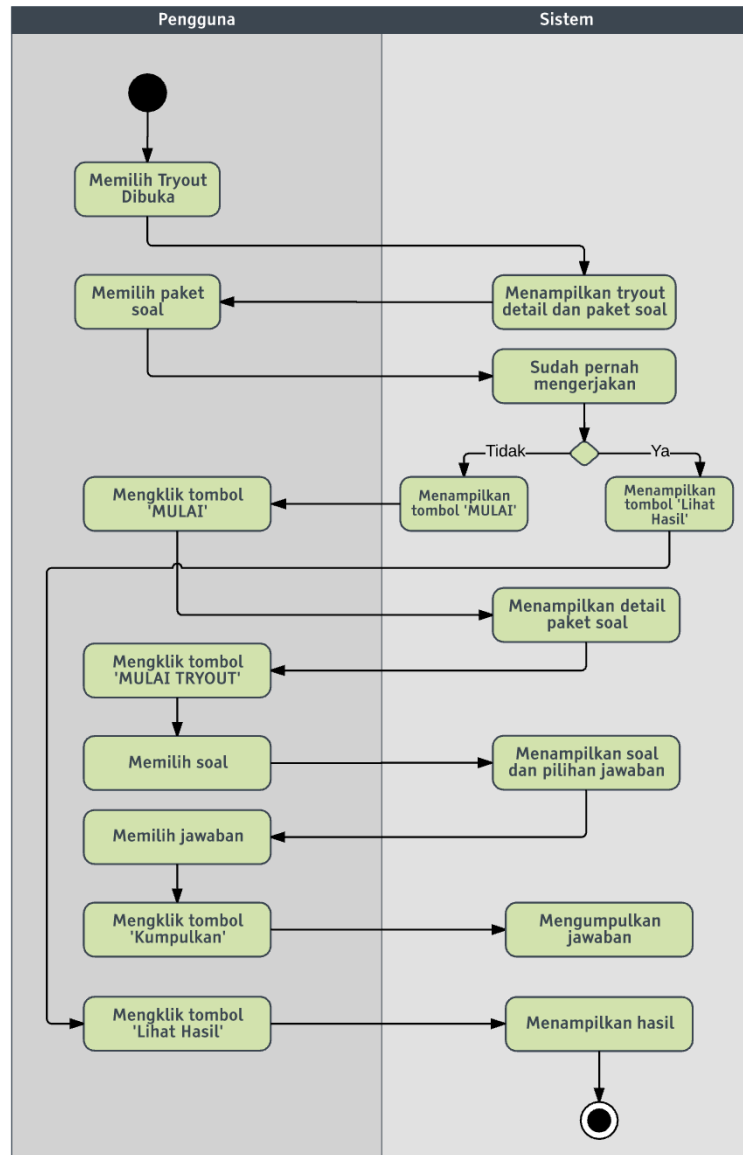
Gambar 8. *Activity Diagram Login*

2) Memilih tryout dibuka

Dijelaskan pada Gambar 8 bahwa pengguna diberikan tampilan daftar tryout, pada gambar 9 merupakan *activity diagram* apabila pengguna memilih tryout yang dibuka pada tampilan daftar tryout.

Setelah pengguna memilih tryout, sistem akan menampilkan tryout detail dan paket soal untuk dipilih pengguna. Sistem akan mengecek apakah pengguna pernah mengerjakan paket soal sebelumnya. Jika pengguna pernah mengerjakan paket soal tersebut sebelumnya, tombol yang ditampilkan adalah tombol 'Lihat Hasil' yang kemudian jika dipilih pengguna, akan membawa pengguna ke halaman hasil (tanpa peringkat).

Sedangkan apabila pengguna belum pernah mengerjakan paket soal sebelumnya, tombol yang ditampilkan adalah tombol 'Mulai'. Dengan pengguna memilih tombol 'Mulai', sistem akan menampilkan detail paket soal dengan tombol 'Mulai Tryout' yang harus dipilih pengguna. Kemudian pengguna meng-klik tombol 'Mulai Tryout', sistem menampilkan pilihan soal yang dapat dipilih oleh pengguna akan mengerjakan soal yang mana saja terlebih dahulu. Sistem akan menampilkan uraian soal dan pilihan jawaban sesuai nomor soal yang dipilih pengguna. Ketika pengguna memutuskan untuk mengumpulkan dengan menekan tombol 'Kumpulkan', sistem akan memproses data dan kemudian menampilkan hasil sesuai dengan tryout yang diikuti pengguna.

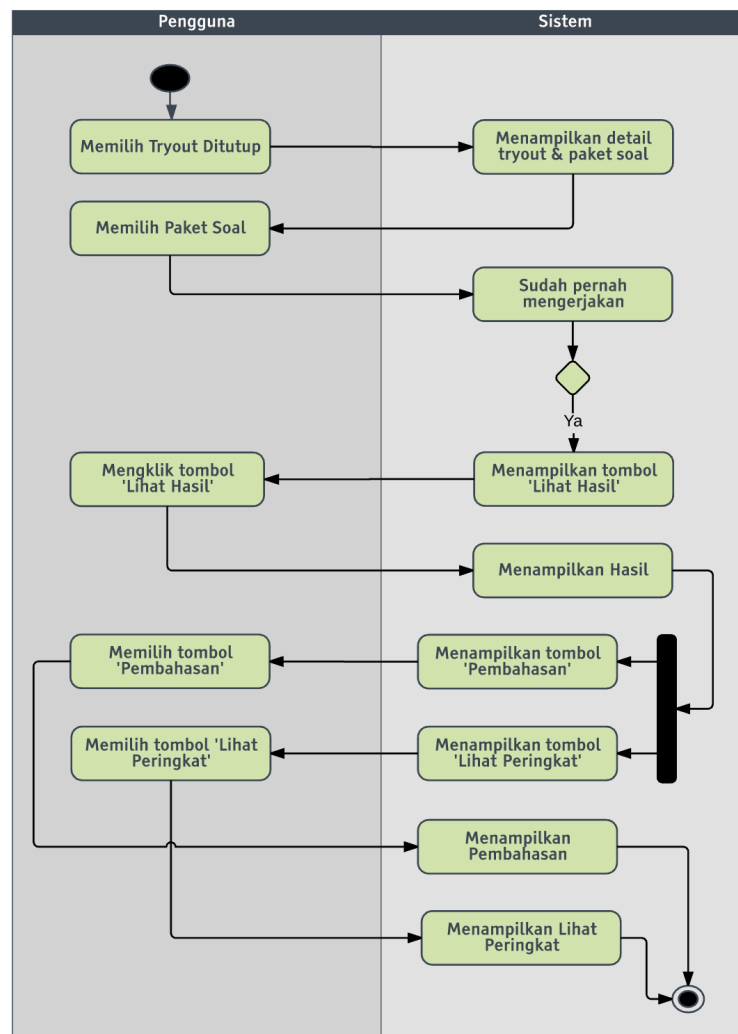


Gambar 9. *Activity Diagram* Memilih Tryout Dibuka

3) Memilih tryout ditutup

Dijelaskan pada Gambar 8 bahwa pengguna diberikan tampilan daftar tryout, pada gambar 10 merupakan *activity diagram* apabila pengguna memilih tryout yang dibuka pada tampilan daftar tryout. Setelah pengguna memilih tryout, sistem akan menampilkan tryout detail dan paket soal untuk dipilih pengguna. Sistem akan mengecek apakah

pengguna pernah mengerjakan paket soal sebelumnya. Jika pengguna pernah mengerjakan paket soal tersebut sebelumnya, tombol yang ditampilkan adalah tombol 'Lihat Hasil' yang kemudian jika dipilih pengguna, akan membawa pengguna ke halaman hasil dengan tombol 'Pembahasan' dan 'Lihat Peringkat'. Apabila pengguna memilih tombol 'Pembahasan', maka sistem akan menampilkan halaman pembahasan. Begitu pula jika pengguna memilih tombol 'Lihat Peringkat', maka sistem akan menampilkan halaman peringkat.

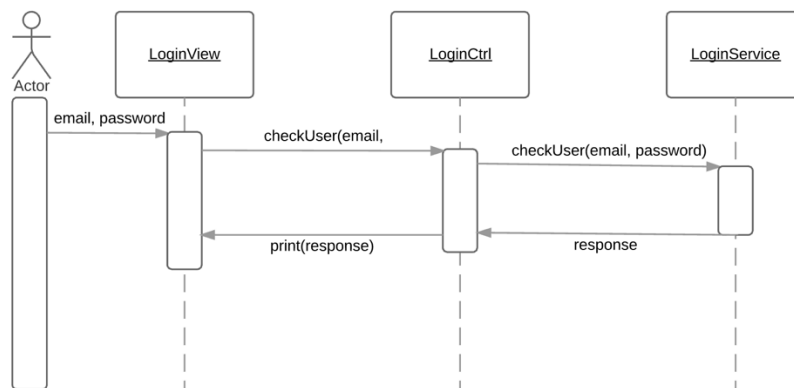


Gambar 10. Activity Diagram Memilih Tryout Ditutup

c. *Sequence Diagram*

1) Melakukan login

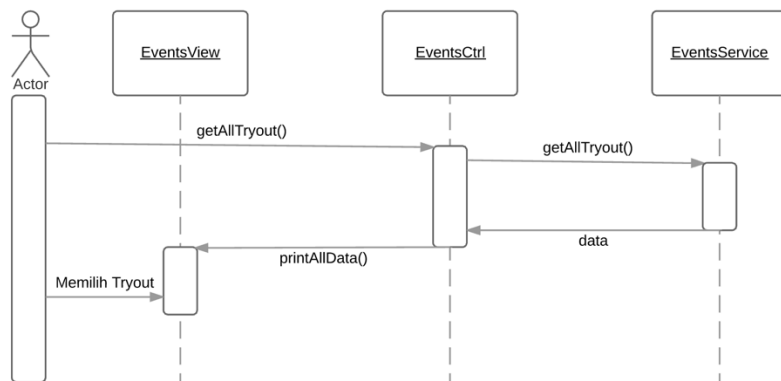
Gambar 11 di bawah ini merupakan *sequence diagram* untuk fungsi melakukan login. Pada *sequence diagram* melakukan login, yang berperan sebagai actor adalah pengguna. *Class* yang terlibat pada *sequence diagram* ini adalah LoginView, LoginService dan LoginCtrl. LoginView berperan sebagai view yang mengatur tampilan, LoginService sebagai model yang terhubung dengan API, dan LoginCtrl yang menghubungkan keduanya.



Gambar 11. *Sequence Diagram* Melakukan Login

2) Memilih tryout

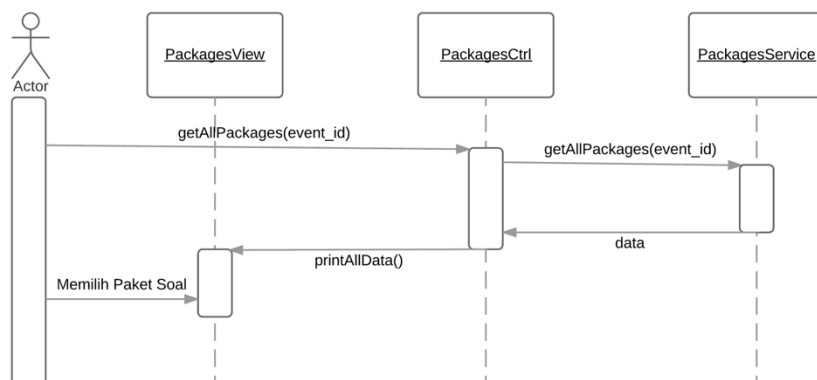
Gambar 12 di bawah ini merupakan *sequence diagram* untuk fungsi memilih tryout. Pada *sequence diagram* memilih tryout, yang berperan sebagai actor adalah pengguna. *Class* yang terlibat pada *sequence diagram* ini adalah EventsView, EventsService dan EventsCtrl. EventsView berperan sebagai view yang mengatur tampilan, EventsService sebagai model yang terhubung dengan API, dan EventsCtrl yang menghubungkan keduanya.



Gambar 12. Sequence Diagram Memilih Tryout

3) Memilih paket soal

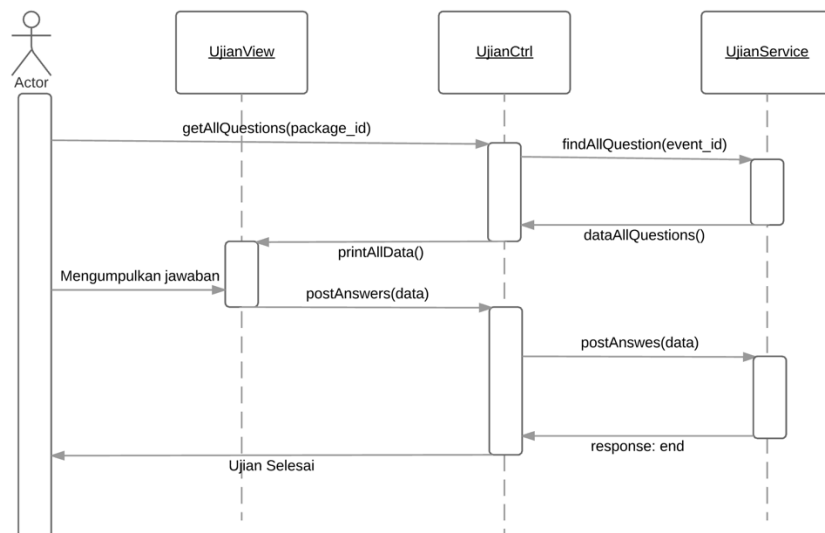
Gambar 13 di bawah ini merupakan *sequence diagram* untuk fungsi memilih paket soal. Pada *sequence diagram* memilih paket soal, yang berperan sebagai actor adalah pengguna. *Class* yang terlibat pada *sequence diagram* ini adalah PackagesView, PackagesService dan PackagesCtrl. PackagesView berperan sebagai view yang mengatur tampilan, PackagesService sebagai model yang terhubung dengan API, dan PackagesCtrl yang menghubungkan keduanya.



Gambar 13. Sequence Diagram Memilih Paket Soal

4) Melakukan ujian

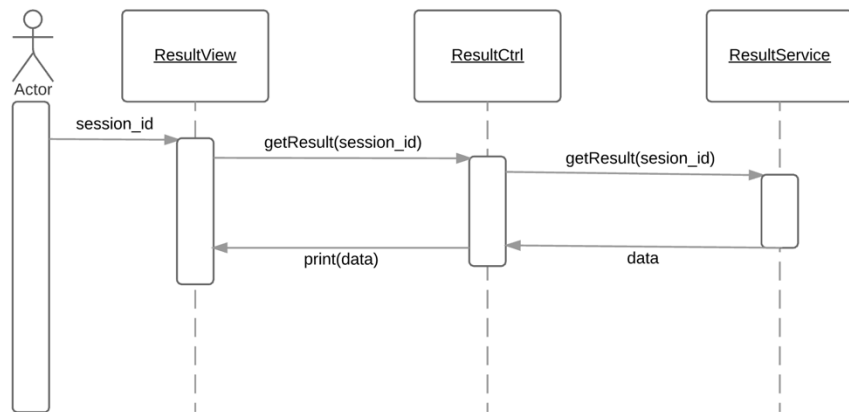
Gambar 14 di bawah ini merupakan *sequence diagram* untuk fungsi melakukan ujian. Pada *sequence diagram* melakukan ujian, yang berperan sebagai actor adalah pengguna. *Class* yang terlibat pada *sequence diagram* ini adalah *UjianView*, *UjianService* dan *UjianCtrl*. *UjianView* berperan sebagai view yang mengatur tampilan, *UjianService* sebagai model yang terhubung dengan API, dan *UjianCtrl* yang menghubungkan keduanya.



Gambar 14. *Sequence Diagram* Melakukan Ujian

5) Melihat hasil

Gambar 15 di bawah ini merupakan *sequence diagram* untuk fungsi melihat hasil. Pada *sequence diagram* melihat hasil, yang berperan sebagai actor adalah pengguna. *Class* yang terlibat pada *sequence diagram* ini adalah *ResultView*, *ResultService* dan *ResultCtrl*. *ResultView* berperan sebagai view yang mengatur tampilan, *ResultService* sebagai model yang terhubung dengan API, dan *ResultCtrl* yang menghubungkan keduanya.



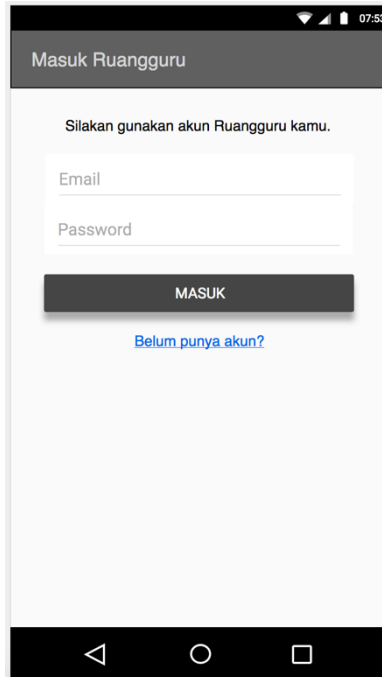
Gambar 15. *Sequence Diagram* Melihat Hasil

2. Desain Antarmuka

Berikut ini merupakan hasil dari rancangan desain antarmuka perangkat lunak ujian daring:

a. Halaman *login*

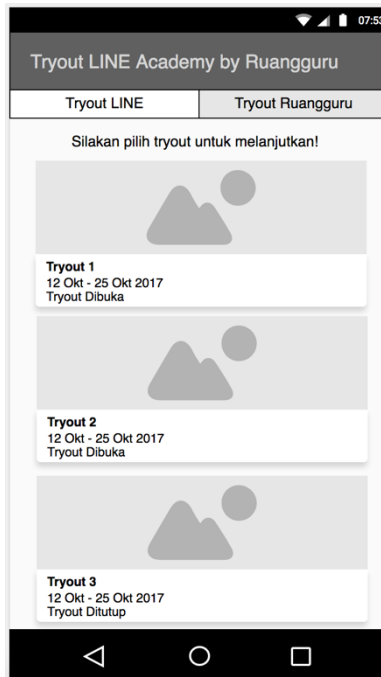
Desain antarmuka halaman *login* untuk sistem ini digambarkan pada Gambar 16. Untuk *login*, pengguna diharuskan telah mempunyai akun ruangguru, yang minimal terdiri dari *email* dan *password*.



Gambar 16. Desain Antarmuka Halaman Login

b. Halaman daftar *tryout*

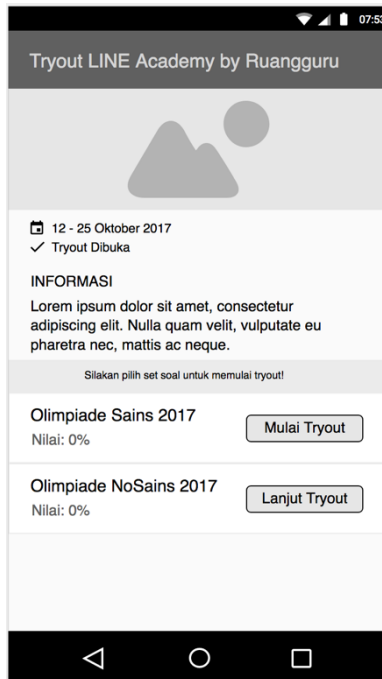
Desain antarmuka halaman daftar *tryout* untuk sistem ini digambarkan pada Gambar 17. Halaman ini muncul setelah pengguna melakukan *login*. Tryout yang tersedia adalah tryout yang sedang aktif, yaitu tryout yang bisa diikuti oleh pengguna. Selain tryout yang sedang aktif, bagian bawah dimunculkan tryout-tryout yang pernah diikuti peserta.



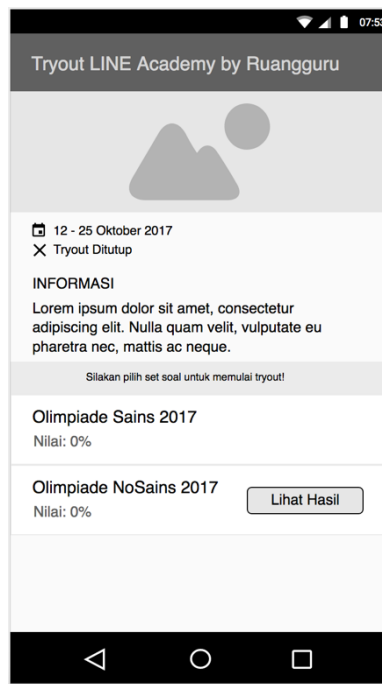
Gambar 17. Desain Antarmuka Halaman Daftar Tryout

c. Halaman detail tryout – daftar paket soal

Desain antarmuka halaman detail tryout dan daftar paket soal untuk sistem ini digambarkan pada Gambar 18 dan Gambar 19. Halaman ini muncul setelah pengguna memilih salah satu tryout di halaman daftar tryout. Detail tryout dapat dilihat dari banner, tanggal mulai dan berakhirnya tryout, status tryout (dibuka atau ditutup) dan informasi singkat tryout. Daftar paket soal dapat dilihat di halaman yang sama. Apabila tryout masih dibuka, pengguna dapat memulai tryout atau melanjutkan tryout yang pernah dikerjakan sebelumnya (selama waktu masih tersedia). Jika tryout telah ditutup, pengguna dapat melihat hasil dari tryout yang pernah diikutinya.



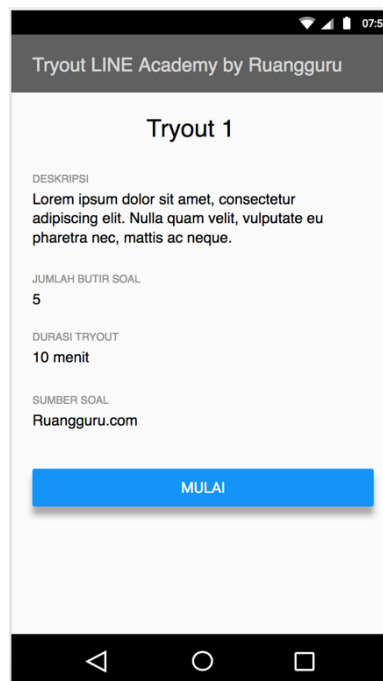
Gambar 18. Desain Antarmuka Halaman Detail Tryout Dibuka



Gambar 19. Desain Antarmuka Halaman Detail Tryout Ditutup

d. Halaman detail paket soal

Desain antarmuka halaman detail paket soal untuk sistem ini digambarkan pada Gambar 20. Halaman ini akan muncul jika pengguna memilih Mulai tryout atau Lanjut tryout di halaman pada Gambar 19. Pada halaman detail paket soal tersedia informasi singkat tentang paket soal yang akan dikerjakan pengguna, diantaranya: deskripsi, jumlah butir soal, durasi tryout, dan sumber soal.

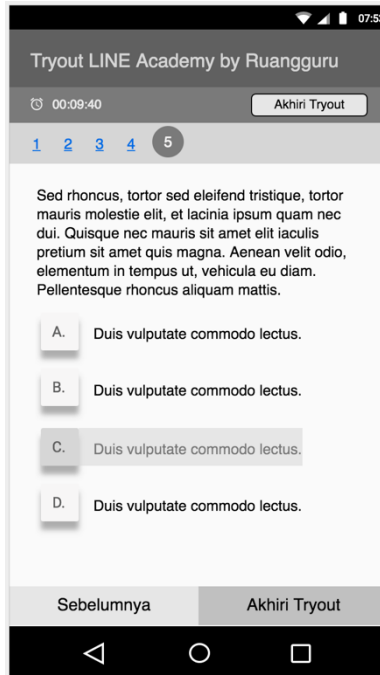


Gambar 20. Desain Antarmuka Halaman Detail Paket Soal

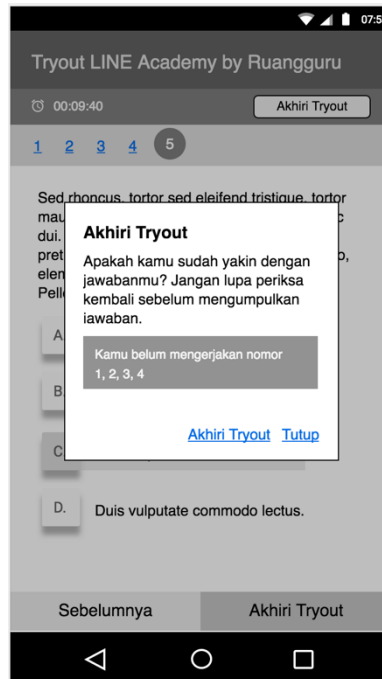
e. Halaman pengerjaan ujian

Desain antarmuka halaman pengerjaan ujian untuk sistem ini digambarkan pada Gambar 21. Ketika pengguna ada pada halaman ini, *timer countdown* ujian langsung berjalan. Jumlah angka di bawah *timer* menunjukkan jumlah soal yang harus dikerjakan pengguna. Pengguna dapat mengarahkan soal dengan tombol 'Sebelumnya' dan 'Sesudahnya' di bagian bawah layar, atau cukup dengan menggeser ke kiri dan ke kanan

pada layar sentuh. Tombol 'Sesudahnya' akan berganti menjadi 'Akhiri tryout' pada soal terakhir. Tombol 'Akhiri tryout' juga berada di bagian atas layar seperti terlihat pada Gambar 21. Apabila pengguna menekan tombol 'Akhiri tryout' namun ada soal yang belum terjawab, sistem akan memberikan peringatan seperti terlihat pada Gambar 22.



Gambar 21. Desain Antarmuka Halaman Pengerjaan Ujian



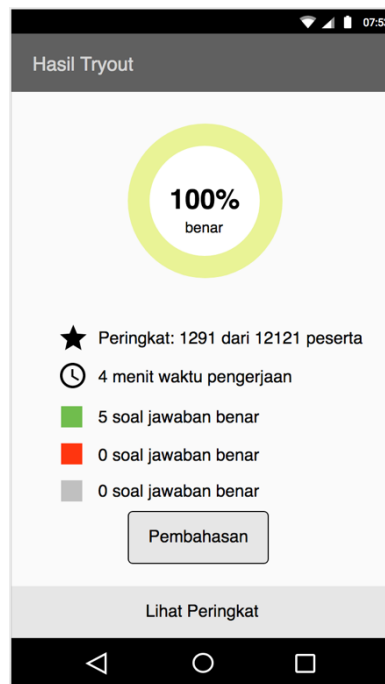
Gambar 22. Desain Antarmuka Halaman Mengakhiri Ujian

f. Halaman hasil ujian

Desain antarmuka halaman hasil ujian untuk sistem ini digambarkan pada Gambar 23 dan Gambar 24. Setelah berhasil mengakhiri ujian, pengguna akan dibawa ke halaman hasil ujian pada Gambar 23. Angka yang berada dalam lingkaran merupakan nilai ujian peserta dalam persentase. Selain menampilkan nilai ujian, sistem juga menampilkan peringkat pengguna (ketika tryout telah ditutup), waktu pengerjaan, jumlah jawaban benar, jumlah jawaban salah dan jumlah jawaban yang tidak diisi oleh pengguna. Apabila tryout telah ditutup, akan muncul tombol 'Pembahasan' dan 'Lihat Peringkat' seperti terlihat pada Gambar 24. Halaman pada Gambar 24 biasanya muncul setelah tombol 'Lihat Hasil' pada Gambar 19 dipilih.



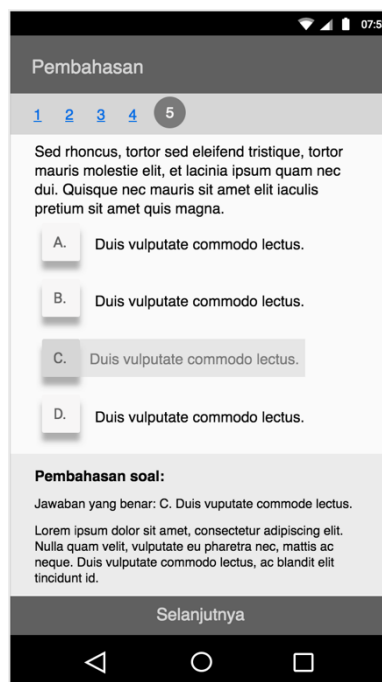
Gambar 23. Desain Antarmuka Halaman Hasil Ujian (Tryout Dibuka)



Gambar 24. Desain Antarmuka Halaman Hasil Ujian (Tryout Ditutup)

g. Halaman pembahasan

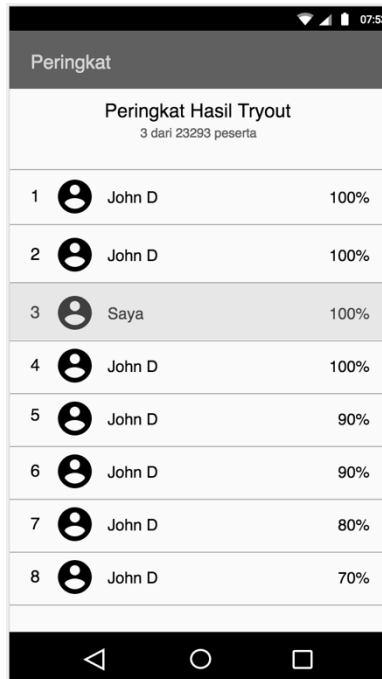
Desain antarmuka halaman pembahasan untuk sistem ini digambarkan pada Gambar 25. Halaman pembahasan terdiri dari beberapa informasi, diantaranya: nomor soal, soal, pilihan jawaban yang telah dipilih peserta, pilihan jawaban benar, dan pembahasan soal. Peserta tryout dapat memilih soal yang ingin dilihat pembahasannya dengan tombol di bagian bawah, atau cukup dengan menggeser ke kiri dan ke kanan halaman pada layar sentuh.











Gambar 25. Desain Antarmuka Halaman Pembahasan Ujian

h. Halaman peringkat

Desain antarmuka halaman peringkat untuk sistem ini digambarkan pada Gambar 26. Halaman peringkat menampilkan informasi peringkat peserta saat ini, jumlah total peserta, peringkat, nama dan nilai peserta-peserta yang masuk 15 besar (plus 1, apabila peserta saat ini tidak masuk 15 besar)



Peringkat		
Peringkat Hasil Tryout 3 dari 23293 peserta		
1	 John D	100%
2	 John D	100%
3	 Saya	100%
4	 John D	100%
5	 John D	90%
6	 John D	90%
7	 John D	80%
8	 John D	70%

Gambar 26. Desain Antarmuka Halaman Peringkat Ujian

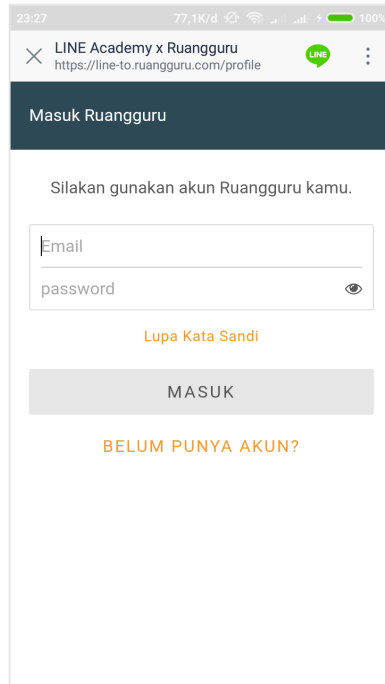
C. Tahap Pengodean (Implementasi)

1. Implementasi Antarmuka

Berikut ini hasil implementasi antarmuka sesuai dengan rancangan yang telah dibuat:

a. Halaman *login*

Implementasi antarmuka halaman *login* sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 27. Untuk *login*, pengguna diharuskan telah mempunyai akun ruangguru, yang minimal terdiri dari *email* dan *password*.



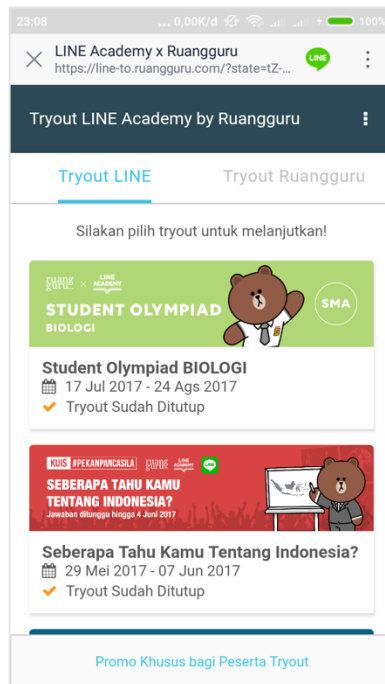
Gambar 27. Implementasi Halaman Login

b. Halaman daftar *tryout*

Implementasi antarmuka halaman daftar *tryout* sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 28 dan Gambar 29. Halaman ini muncul setelah pengguna melakukan *login*. Tryout yang tersedia adalah tryout yang sedang aktif terlihat pada Gambar 28, yaitu tryout yang bisa diikuti oleh pengguna. Selain tryout yang sedang aktif, bagian bawah dimunculkan tryout-tryout yang telah ditutup namun pernah diikuti peserta dapat dilihat pada Gambar 29.



Gambar 28. Implementasi Halaman Daftar Tryout Dibuka



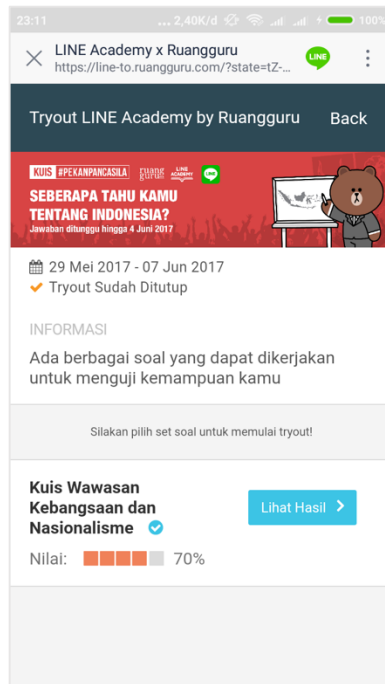
Gambar 29. Implementasi Halaman Daftar Tryout Ditutup

c. Halaman detail tryout – daftar paket soal

Implementasi antarmuka halaman detail tryout dan daftar paket soal sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 30 dan Gambar 31. Halaman ini muncul setelah pengguna memilih salah satu tryout di halaman daftar tryout. Detail tryout dapat dilihat dari banner, tanggal mulai dan berakhirnya tryout, status tryout (dibuka atau ditutup) dan informasi singkat tryout. Daftar paket soal dapat dilihat di halaman yang sama. Apabila tryout masih dibuka, pengguna dapat memulai tryout atau melanjutkan tryout yang pernah dikerjakan sebelumnya (selama waktu masih tersedia). Jika tryout telah ditutup, pengguna dapat nilai dan melihat hasil dari tryout yang pernah diikutinya.



Gambar 30. Implementasi Halaman Detail Tryout Dibuka



Gambar 31. Implementasi Halaman Detail Tryout Ditutup

d. Halaman detail paket soal

Implementasi antarmuka halaman detail paket soal untuk sistem ini digambarkan pada Gambar 32. Halaman ini akan muncul jika pengguna memilih Mulai tryout atau Lanjut tryout di halaman pada Gambar 30. Pada halaman detail paket soal tersedia informasi singkat tentang paket soal yang akan dikerjakan pengguna, diantaranya: deskripsi, jumlah butir soal, durasi tryout, dan sumber soal.

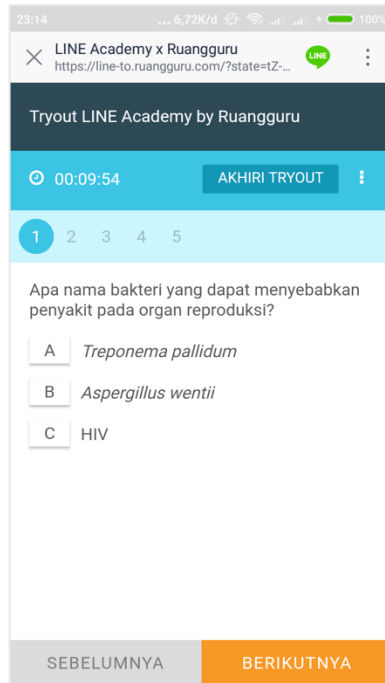


Gambar 32. Implementasi Halaman Detail Paket Soal

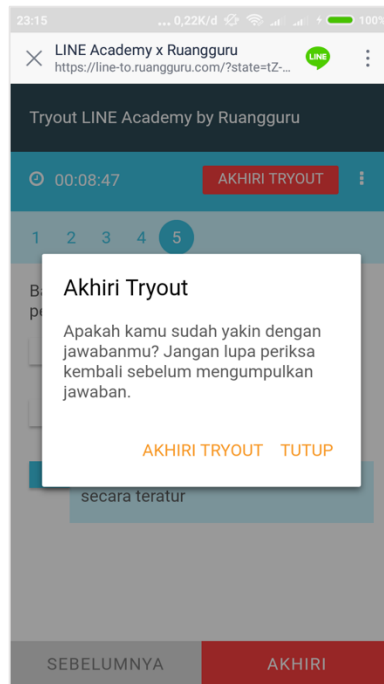
e. Halaman pengerjaan ujian

Implementasi antarmuka halaman pengerjaan ujian sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 33. Ketika pengguna ada pada halaman ini, *timer countdown* ujian langsung berjalan. Jumlah angka di bawah *timer* menunjukkan jumlah soal yang harus dikerjakan pengguna. Pengguna dapat mengarahkan soal dengan tombol 'Sebelumnya' dan 'Sesudahnya' di bagian bawah layar, atau cukup dengan menggeser ke kiri dan ke kanan pada layar sentuh. Tombol 'Sesudahnya' akan berganti menjadi 'Akhir tryout' pada soal terakhir. Secara *default* warna nomor soal adalah abu-abu, ini berarti soal belum dijawab, nomor soal akan berganti biru ketika soal telah dijawab. Tombol 'Akhir tryout' juga berada di bagian atas layar seperti terlihat pada Gambar 33. Apabila pengguna menekan tombol 'Akhir tryout' namun ada soal yang belum

terjawab, sistem akan memberikan peringatan seperti terlihat pada Gambar 34.



Gambar 33. Implementasi Halaman Pengerjaan Ujian



Gambar 34. Implementasi Halaman Mengakhiri Ujian

f. Halaman hasil ujian

Implementasi antarmuka halaman hasil ujian sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 35 dan Gambar 36. Setelah berhasil mengakhiri ujian, pengguna akan dibawa ke halaman hasil ujian pada Gambar 35. Angka yang berada dalam lingkaran merupakan nilai ujian peserta dalam persentase. Selain menampilkan nilai ujian, sistem juga menampilkan peringkat pengguna (ketika tryout telah ditutup), waktu pengerjaan, jumlah jawaban benar, jumlah jawaban salah dan jumlah jawaban yang tidak diisi oleh pengguna. Apabila tryout telah ditutup, akan muncul tombol 'Pembahasan' dan 'Lihat Peringkat' seperti terlihat pada Gambar 36. Halaman pada Gambar 36 biasanya muncul setelah tombol 'Lihat Hasil' pada Gambar 31 dipilih.



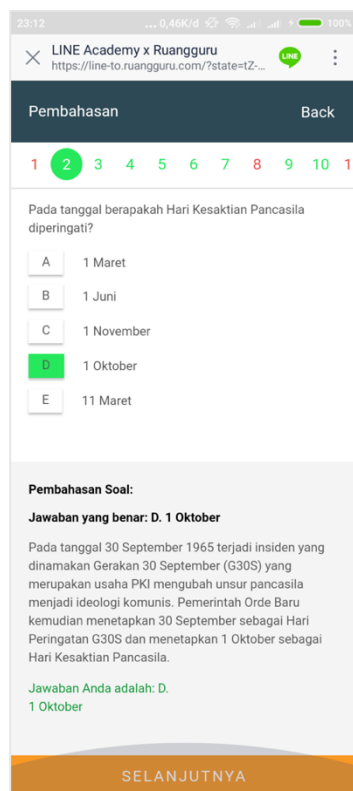
Gambar 35. Implementasi Halaman Hasil Ujian Tryout Dibuka



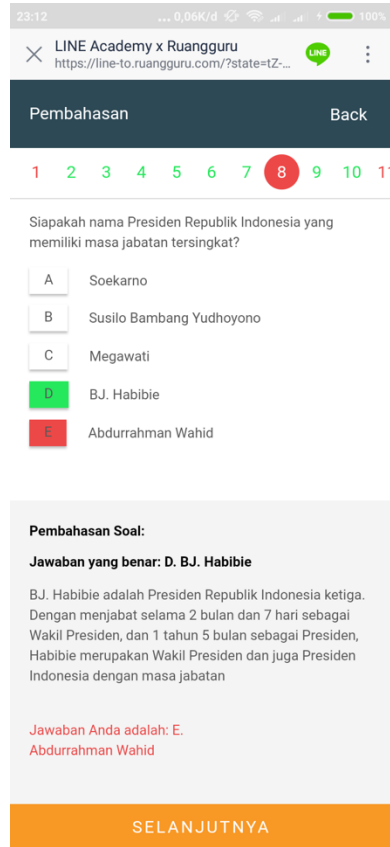
Gambar 36. Implementasi Halaman Hasil Ujian Tryout Ditutup

g. Halaman pembahasan ujian

Implementasi antarmuka halaman pembahasan ujian sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 37 dan Gambar 38. Halaman pembahasan terdiri dari beberapa informasi, diantaranya: nomor soal, soal, pilihan jawaban yang telah dipilih peserta, pilihan jawaban benar, dan pembahasan soal. Warna hijau pada nomor soal berarti soal telah dijawab benar oleh peserta, sedangkan warna merah berarti soal dijawab salah oleh peserta. Peserta tryout dapat memilih soal yang ingin dilihat pembahasannya dengan tombol di bagian bawah, atau cukup dengan menggeser ke kiri dan ke kanan halaman pada layar sentuh.



Gambar 37. Implementasi Halaman Pembahasan Jawaban Benar



Gambar 38. Implementasi Halaman Pembahasan Jawaban Salah

h. Halaman peringkat

Implementasi antarmuka halaman peringkat sesuai dengan desain antarmuka untuk sistem ini digambarkan pada Gambar 39. Halaman peringkat menampilkan informasi peringkat peserta saat ini, jumlah total peserta, peringkat, nama dan nilai peserta-peserta yang masuk 15 besar (plus 1, apabila peserta saat ini tidak masuk 15 besar)

Peringkat Hasil Tryout dari 45.800 peserta		
1	Muhammad I.	100%
2	Devi P.	100%
3	Ranisa C.	100%
4	Amalia	100%
5	Muhammad I.	100%
6	Ahmad R.	100%
7	Dewi S.	100%
8	MUHAMMAD H.	100%
9	Rafi F.	100%
10	Al D.	100%
11	Makrub	100%
12	Kevin	100%
13	risa k.	100%
14	Makrub	100%
15	Ahmad R.	100%
...		
15.480	ab	75%

Gambar 39. Implementasi Halaman Peringkat

2. Implementasi Sistem

Proses pengembangan perangkat lunak ujian daring menggunakan kode editor *Sublime Text 3* dan *framework* bahasa pemrograman javascript *AngularJS*. *AngularJS* menggunakan pola *Model-View-Controller* (MVC) yang memiliki 3 komponen, yaitu *model*, *view* dan *controller*. *Model* bertanggungjawab mengenai data (memproses data dari API), *view* berhubungan dengan tampilan yang muncul, sedangkan *controller* yang menghubungkan *model* dan *view*.

API yang digunakan berasal dari dua sumber, yaitu API ujian daring dari ruangguru, dan API LINE Login. API ujian daring dari ruangguru memungkinkan pengguna untuk melakukan pendaftaran, masuk dalam sistem, mendapatkan daftar event/ tryout, melihat paket soal, mengerjakan soal, mengumpulkan jawaban, mendapatkan hasil, mengetahui peringkat diri dibanding peserta lainnya

dan mengetahui pembahasan dari soal-soal yang telah dikerjakannya. API LINE Login digunakan sebagai *authorization* utama bagi penggunanya, dengan menggunakan API ini, pengguna tidak perlu melakukan pendaftaran dan masuk karena akan diotomatisasi oleh API ini.

D. Tahap Pengujian

Pengujian dilakukan menggunakan instrumen sesuai dengan ISO/IEC 25010 dan *Dimensions of Quality Pressmann* yang meliputi aspek *functional suitability*, *performance efficiency*, *usability*, *security* dan *compatibility*.

1. Hasil Pengujian *Functional Suitability*

Pengujian *functional suitability* diujikan kepada 5 orang yang ahli dalam pengembangan website dari PT Ruang Raya Indonesia. Instrumen yang diujikan memenuhi subkarakteristik *functional completeness*, *functional correctness*, dan *functional appropriateness*. Hasil pengujian *functional suitability* adalah sebagai berikut:

Tabel 9. Hasil Pengujian Subkarakteristik Functional Appropriateness

Nomor Pertanyaan	Ya	Tidak
1	5	0
2	5	0
3	5	0
4	5	0
5	5	0
6	5	0
7	5	0
8	5	0
9	5	00
10	5	0
11	5	0
12	5	0
13	5	0
14	5	0
15	5	0
Total	75	0

Pengujian subkarakteristik *functional appropriateness* menggunakan rumus dari Acharya dan Sinha (2013), yaitu:

$$X = \frac{I}{P}$$

Keterangan:

I = Jumlah fitur yang berhasil diimplementasikan (Ya)

P = Jumlah fitur yang dirancang

Berikut ini perhitungan subkarakteristik *functiona appropriateness* dilakukan berdasarkan tabel hasil pengujian *functional appropriateness*:

$I_{total} = \text{jumlah fitur yang berhasil diimplementasikan} \times \text{jumlah penguji}$

$$I_{total} = 15 \times 5$$

$$I_{total} = 75$$

$P_{total} = \text{jumlah fitur yang dirancang} \times \text{jumlah penguji}$

$$P_{total} = 15 \times 5$$

$$P_{total} = 75$$

Sehingga:

$$X = \frac{I_{total}}{P_{total}}$$

$$X = \frac{75}{75}$$

$$X = 1$$

Hasil perhitungan di atas juga berlaku untuk pengujian subkarakteristik *functional completeness* karena instrumen yang diujikan merupakan instrumen hasil analisis kebutuhan yang mencakup semua tugas dan tujuan pengguna secara spesifik. Berdasarkan dari hasil perhitungan di atas dapat disimpulkan bahwa perangkat lunak ujian daring ini dinyatakan baik dalam subkarakteristik *functional completeness* dan *functional appropriateness*.

Untuk hasil pengujian subkarakteristik *functional correctness* terdapat pada tabel berikut ini:

Tabel 10. Hasil Pengujian Subkarakteristik Functional Correctness

Nomor Pertanyaan	Ya	Tidak
1	5	0
2	5	0
3	5	0
4	5	0
5	5	0
6	5	0
7	5	0
8	5	0
9	5	0
10	5	0
Total	50	0

Rumus yang digunakan untuk menghitung hasil pengujian subkarakteristik *functional correctness* sama dengan rumus yang digunakan untuk menghitung hasil pengujian subkarakteristik *functional completeness* dan *functional appropriateness*. Perhitungannya adalah seperti berikut:

$$I_{total} = \text{jumlah fitur yang berhasil diimplementasikan} \times \text{jumlah penguji}$$

$$I_{total} = 10 \times 5$$

$$I_{total} = 50$$

$$P_{total} = \text{jumlah fitur yang dirancang} \times \text{jumlah penguji}$$

$$P_{total} = 10 \times 5$$

$$P_{total} = 50$$

Sehingga:

$$X = \frac{I_{total}}{P_{total}}$$

$$X = \frac{50}{50}$$

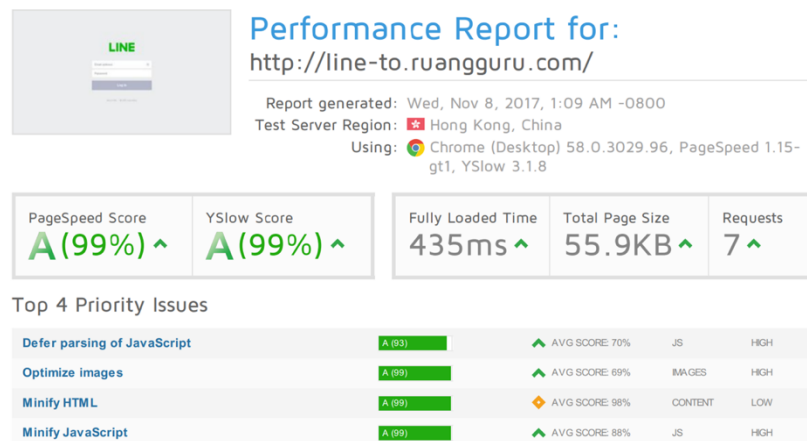
$$X = 1$$

Hasil 1 untuk pengujian subkarakteristik *functional correctness* menunjukkan bahwa perangkat lunak ujian daring ini baik dalam subkarakteristik *functional correctness*. Berdasarkan hasil perhitungan dari subkarakteristik *functional completeness*, *functional appropriateness* dan *functional correctness* dapat disimpulkan bahwa perangkat lunak ujian daring telah memenuhi aspek *functional suitability* ISO/IEC 25010.

2. Hasil Pengujian *Performance Efficiency*

Pengujian *performance efficiency* menggunakan *tool* GTMetrix. Hasilnya adalah seperti berikut:

a. Integrasi LINE login

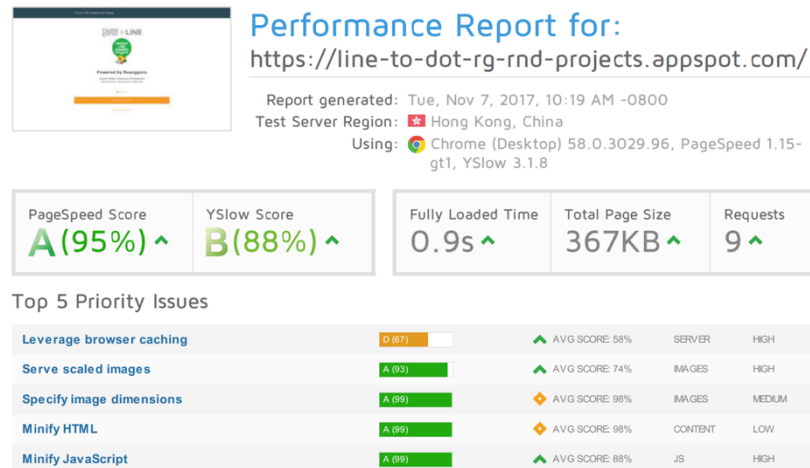


Gambar 40. Laporan GTMetrix Halaman Integrasi LINE

Pengujian *performance efficiency* halaman integrasi LINE pada Gambar 40 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(99%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource*

software yang diujikan sangat baik. Waktu untuk memuat halaman selama 0,435 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

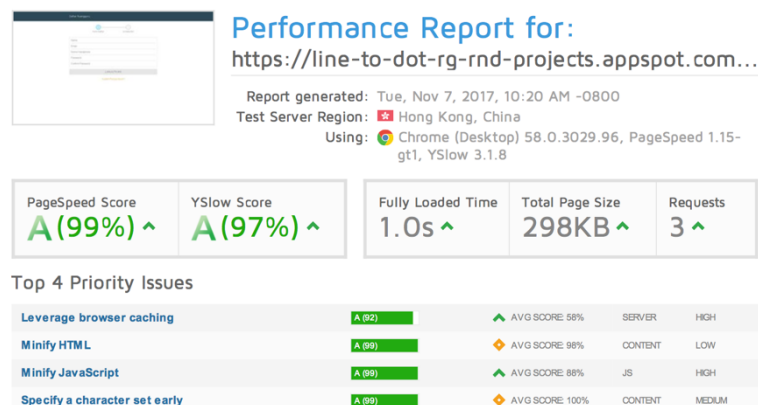
b. Halaman onboard



Gambar 41. Laporan GTMetrix Halaman Onboard

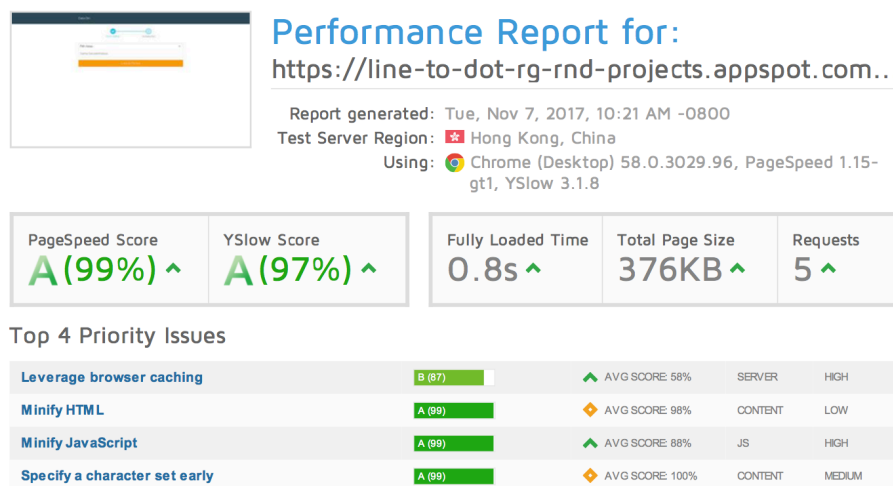
Pengujian *performance efficiency* halaman Onboard pada Gambar 41 menunjukkan grade yang didapatkan adalah A(95%) untuk PageSpeed dan A(88%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan baik. Waktu untuk memuat halaman selama 0,9 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

c. Halaman register



Gambar 42 Laporan GTMetrix Halaman Register (1)

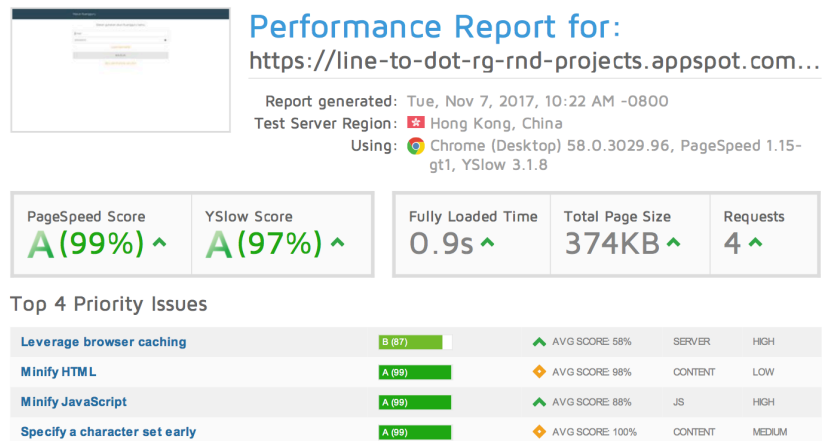
Pengujian *performance efficiency* halaman register(1) pada Gambar 42 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 1 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.



Gambar 43. Laporan GTMetrix Halaman Register (2)

Pengujian *performance efficiency* halaman register(2) pada Gambar 43 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 0,8 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

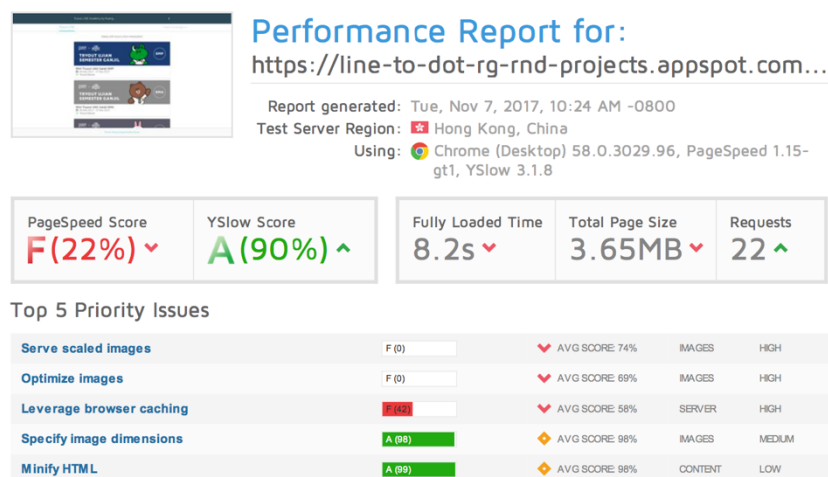
d. Halaman login



Gambar 44. Laporan GTMetrix Halaman Login

Pengujian *performance efficiency* halaman login pada Gambar 44 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 0,9 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

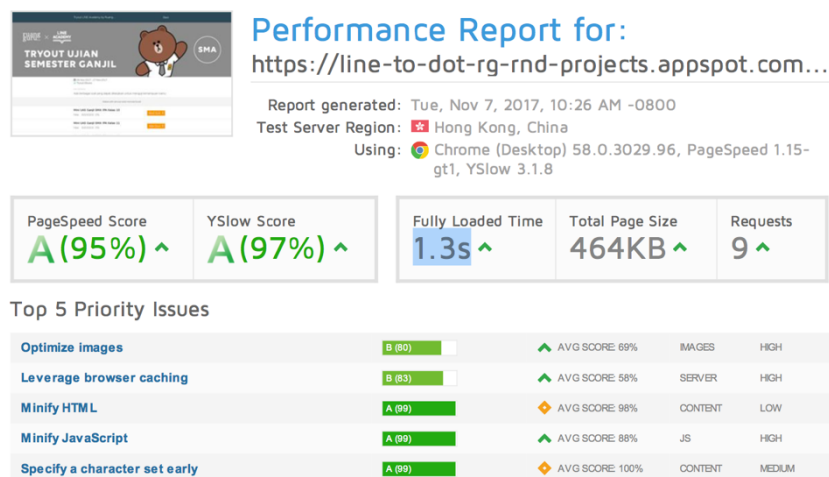
e. Halaman daftar tryout



Gambar 45. Laporan GTMetrix Halaman Daftar Tryout

Pengujian *performance efficiency* halaman daftar tryout pada Gambar 45 menunjukkan grade yang didapatkan adalah F(22%) untuk PageSpeed dan A(90%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan cukup rendah. Hal ini dikarenakan banyaknya event yang pernah dikerjakan pengguna, sehingga waktu untuk memuat gambar banner meningkat. Ditambah dengan tidak adanya sistem *caching image* dalam aplikasi ini. Waktu untuk memuat halaman selama 8,2 detik, tidak cukup memenuhi standar Aptimize (2010), yaitu di bawah 7 detik

f. Halaman detail tryout – daftar paket soal



Gambar 46. Laporan GTMetrix Halaman Detail Tryout - Daftar Paket Soal

Pengujian *performance efficiency* halaman Detail Tryout - Daftar Paket Soal pada Gambar 46 menunjukkan grade yang didapatkan adalah A(95%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 1,3 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

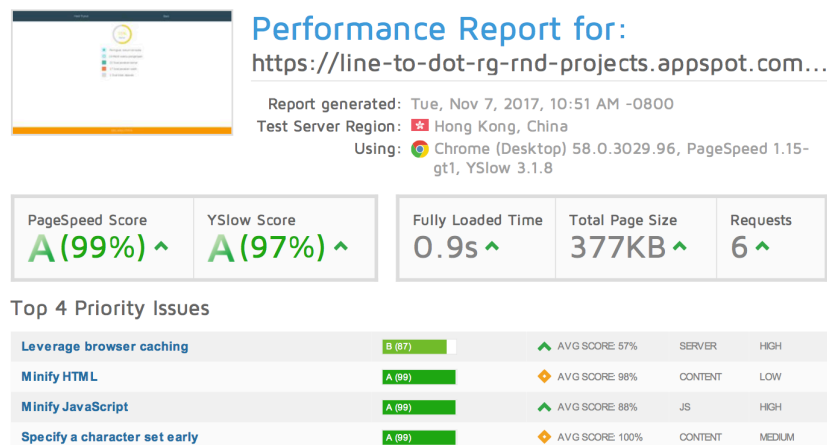
g. Halaman detail paket soal

Pengujian performance ini tidak dilakukan karena data di halaman ini menggunakan *localStorage*, sedangkan pengujian *performance efficiency* oleh GTMetrix menyebabkan penghapusan *localStorage*.

h. Halaman pengerjaan ujian

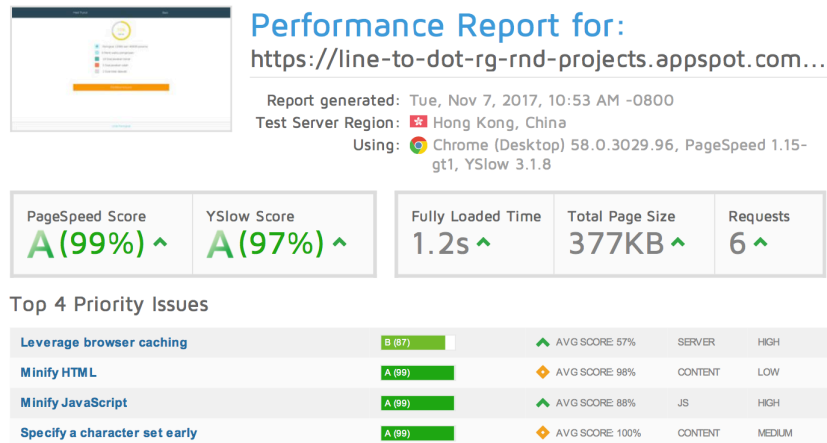
Pengujian performance ini tidak dilakukan karena data di halaman ini menggunakan *localStorage*, sedangkan pengujian *performance efficiency* oleh GTMetrix menyebabkan penghapusan *localStorage*.

i. Halaman hasil ujian



Gambar 47. Laporan GTMetrix Halaman Hasil (Tryout Dibuka)

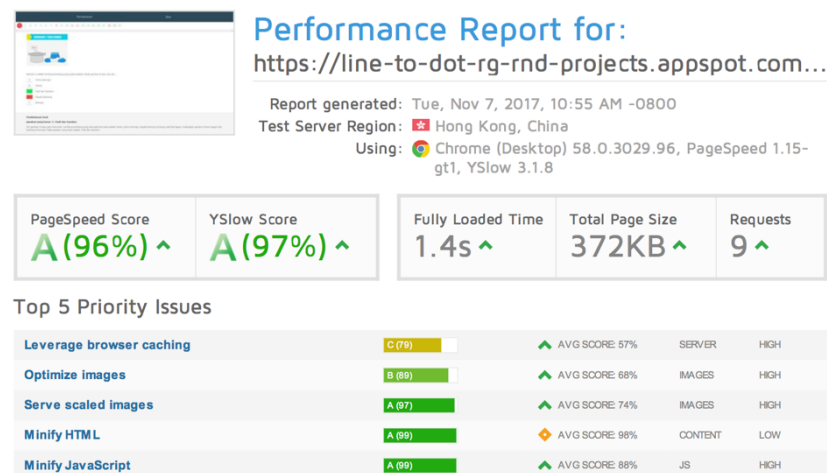
Pengujian *performance efficiency* halaman hasil (tryout dibuka) pada Gambar 47 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 0,9 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.



Gambar 48. Laporan GTMetrix Halaman Hasil (Tryout Ditutup)

Pengujian *performance efficiency* halaman hasil (tryout ditutup) pada Gambar 48 menunjukkan grade yang didapatkan adalah A(99%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 1,2 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

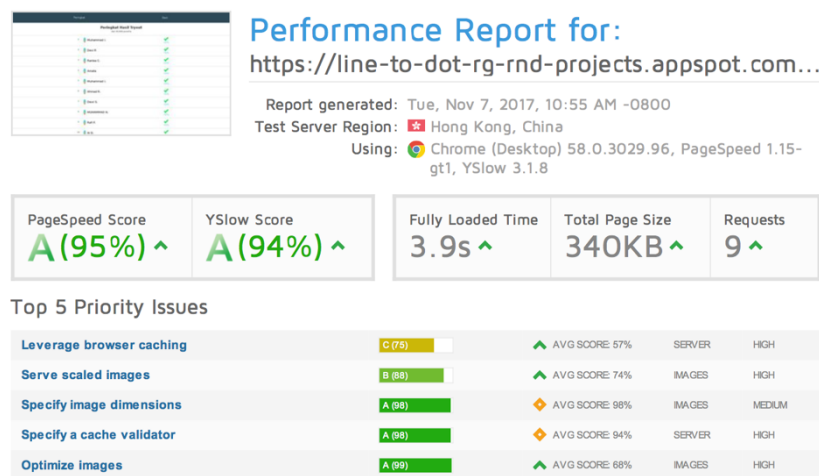
j. Halaman pembahasan



Gambar 49. Laporan GTMetrix Halaman Pembahasan

Pengujian *performance efficiency* halaman login pada Gambar 49 menunjukkan grade yang didapatkan adalah A(96%) untuk PageSpeed dan A(97%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 1,4 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

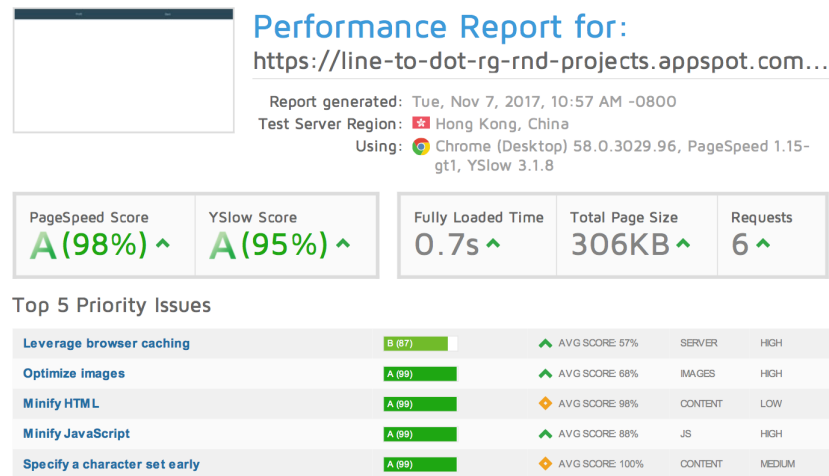
k. Halaman peringkat



Gambar 50. Laporan GTMetrix Halaman Peringkat

Pengujian *performance efficiency* halaman login pada Gambar 50 menunjukkan grade yang didapatkan adalah A(95%) untuk PageSpeed dan A(94%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 3,9 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

I. Halaman profil



Gambar 51. Laporan GTMetrix Halaman Profil

Pengujian *performance efficiency* halaman login pada Gambar 51 menunjukkan grade yang didapatkan adalah A(98%) untuk PageSpeed dan A(95%) untuk YSlow. Grade ini menunjukkan bahwa tingkat efisiensi *resource software* yang diujikan sangat baik. Waktu untuk memuat halaman selama 0,7 detik, berarti telah memenuhi standar Aptimize (2010), yaitu di bawah 7 detik.

Dari 10 hasil pengujian *performance efficiency* di atas didapatkan hasil yang berbeda-beda di setiap halamannya. Untuk mendapatkan hasil secara keseluruhan sistem, maka dicari rata-ratanya. Adapun rangkuman hasil pengujian di atas dan rata-ratanya dapat dilihat pada Tabel 11 berikut ini:

Tabel 11. Tabel Hasil Pengujian Performance Efficiency

No	Halaman	Page Load (detik)	Page Size (kB)	PageSpeed Grade	YSlow Grade
1	Integrasi LINE	0,4	55,9	A (99%)	A (99%)
2	Onboard	0,9	367	A (95%)	B (88%)
3	Register I	1,0	293	A (99%)	A (99%)
4	Register II	0,8	376	A (99%)	A (99%)
5	Login	0,9	374	A (99%)	A (97%)
6	Daftar Tryout	8,2	3645	F (22%)	A (90%)
7	Detail Tryout	1,3	464	A (95%)	A (97%)
8	Hasil (TO Dibuka)	0,9	377	A (99%)	A (97%)
9	Hasil (TO Ditutup)	1,2	377	A (99%)	A (97%)
10	Pembahasan	1,4	372	A (96%)	A (97%)
11	Peringkat	3,9	340	A (95%)	A (94%)
12	Profil	0,7	306	A (98%)	A (95%)
Rata-rata		1,8	662,8	A (91%)	A (96%)

3. Hasil Pengujian *Usability*

Pengujian *usability* menggunakan angke USE Questionnaire yang berjumlah 30 butir dengan menggunakan skala Likert dan diberikan kepada 20 responden. Hasil dari pengujian dapat dilihat di Tabel 12 berikut ini:

Tabel 12. Hasil Pengujian Usability

Pertanyaan	STS	TS	R	S	SS
1	0	4	5	8	3
2	0	0	6	10	4
3	0	5	0	8	7
4	0	1	6	8	5
5	0	0	8	7	5
6	0	0	1	14	5
7	0	0	6	8	6
8	0	4	3	11	2
9	0	3	2	7	8
10	0	0	0	9	11
11	0	2	0	11	7
12	0	4	3	7	6
13	0	0	1	10	9
14	0	0	5	9	6
15	0	0	6	11	3
16	0	0	4	8	8
17	0	1	4	10	5
18	0	0	4	10	6
19	0	0	0	11	9
20	0	0	3	5	12
21	0	0	2	12	6
22	0	0	2	12	6
23	0	0	4	7	9
24	0	0	2	7	11
25	0	0	3	8	9
26	0	0	3	10	7
27	0	0	7	7	6
28	0	0	0	15	5
29	0	0	6	6	8
30	0	0	0	10	10
TOTAL	0	24	96	276	204

Berdasarkan tabel di atas dapat diketahui total jawaban sangat setuju (**SS**) berjumlah **204**, setuju (**S**) berjumlah **276**, ragu-ragu (**R**) berjumlah **96**, tidak

setuju (**TS**) berjumlah **24**, dan sangat tidak setuju berjumlah (**0**). Untuk mendapatkan interpretasi hasil pengujian aspek *usability* (P_{skor}), data-data di atas dihitung dengan rumus sebagai berikut:

$$P_{skor} = \frac{Skor_{total}}{i \times r \times 5} \times 100\%$$

dengan **J** adalah jumlah jawaban responden masing-masing *level of agreement*, **i** adalah jumlah pertanyaan, dan **r** adalah jumlah total responden, maka rumus disubstitusi menjadi sebagai berikut:

$$P_{skor} = \frac{(J_{ss} \times 5) + (J_s \times 4) + (J_R \times 3) + (J_{TS} \times 2) + (J_{STS} \times 1)}{i \times r \times 5} \times 100\%$$

$$P_{skor} = \frac{(204 \times 5) + (276 \times 4) + (96 \times 3) + (24 \times 2) + (0 \times 1)}{30 \times 20 \times 5} \times 100\%$$

$$P_{skor} = \frac{1020 + 1104 + 288 + 48 + 0}{3000} \times 100\%$$

$$P_{skor} = \frac{2460}{3000} \times 100\%$$

$$P_{skor} = 82 \%$$

Dari penghitungan di atas didapat bahwa interpretasi hasil pengujian adalah **82%**. Jika dibandingkan dengan Tabel 7 Kategori Penilaian Usability, maka dapat disimpulkan hasil pengujian *usability* ini adalah **sangat layak**.

Selanjutnya, agar angket / kuesioner yang digunakan benar-benar dapat dipercaya sebagai alat pengumpul data, maka perlu diuji reliabilitas atau tingkat kepercayaannya, untuk mengetahui kekonsistensian angket/kuesioner tersebut. Pengujian reliabilitas angket di sini menggunakan tool SPSS dengan perhitungan alpha cronbach. Adapun hasil pengujian SPSS dengan alpha cronbach dapat dilihat pada Gambar 61 seperti berikut:

Case Processing Summary			
		N	%
Cases	Valid	20	100,0
	Excluded ^a	0	,0
	Total	20	100,0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics	
Cronbach's Alpha	N of Items
,864	30

Gambar 52. Hasil perhitungan Alpha Cronbach menggunakan SPSS

Dari gambar 52 didapat bahwa hasil uji reliabilitas angket adalah 0,864. Apabila dibandingkan dengan Tabel 7, didapatkan *internal consistency* angket masuk dalam kategori **Good**.

Jika menggunakan tabel r product moment, dengan taraf signifikansi 1% dan nilai N = 20, didapatkan angka sebesar 0,561. Selanjutnya, diketahui bahwa jika nilai Alpha lebih besar rtabel maka item-item angket yang digunakan dinyatakan reliabel. Dari pernyataan itu, dapat disimpulkan item-item angket dengan menggunakan USE Questionnaire yang digunakan dalam penelitian ini adalah reliabel, karena nilai Alpha (0,864) lebih besar dari nilai rtabel (0,561).

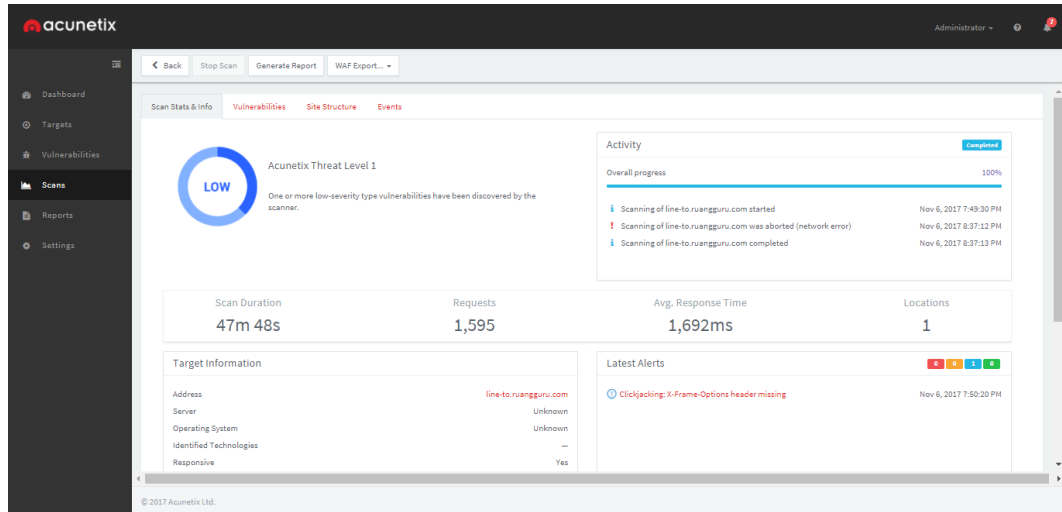
4. Hasil Pengujian *Security*

Pengujian *security* menggunakan aplikasi *Acunetix 11* dan dilakukan dengan metode *scanning* (pemindaian). Setelah pemindaian selesai, Acunetix akan menampilkan hasil pemindaian secara langsung. Hasil pemindaian menunjukkan tanggal mulai dan akhir pemindaian, durasi pemindaian dan semua tanda yang telah diidentifikasi selama pemindaian. Logo AcuSensor juga ditampilkan saat pemindaian mendeteksi dan menggunakan AcuSensor selama pemindaian web.

Hasil pemindaian terdiri dari empat bagian, yaitu:

a. Scan Stats & Info

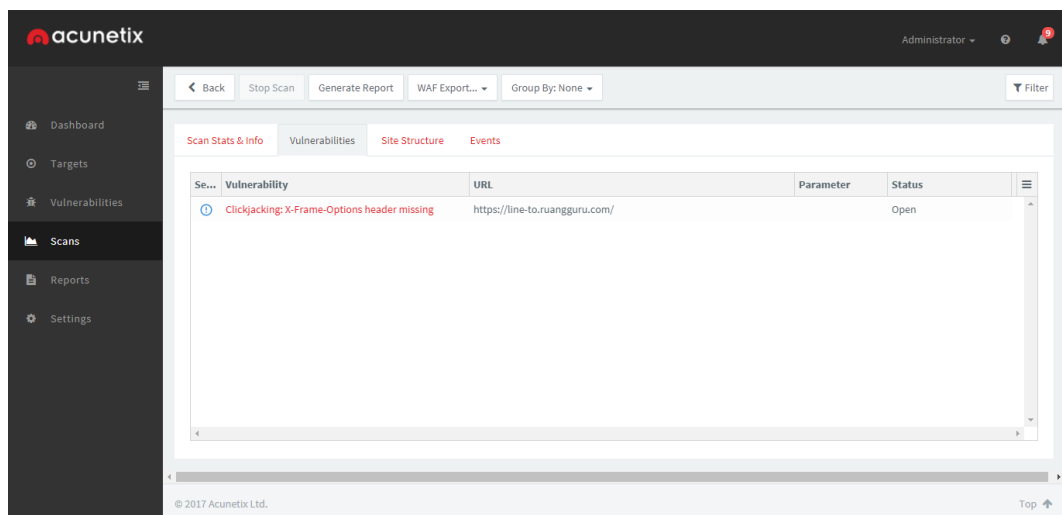
Gambar 53 memberikan gambaran umum tentang *Target* yang terdeteksi oleh pemindaian, dan informasi tentang *Scan*, seperti durasi pemindaian, waktu respons rata-rata dan jumlah file yang dipindai.



Gambar 53. Laporan Pengujian Security dengan Acunetix: Scan Stats & Info

b. Vulnerabilities

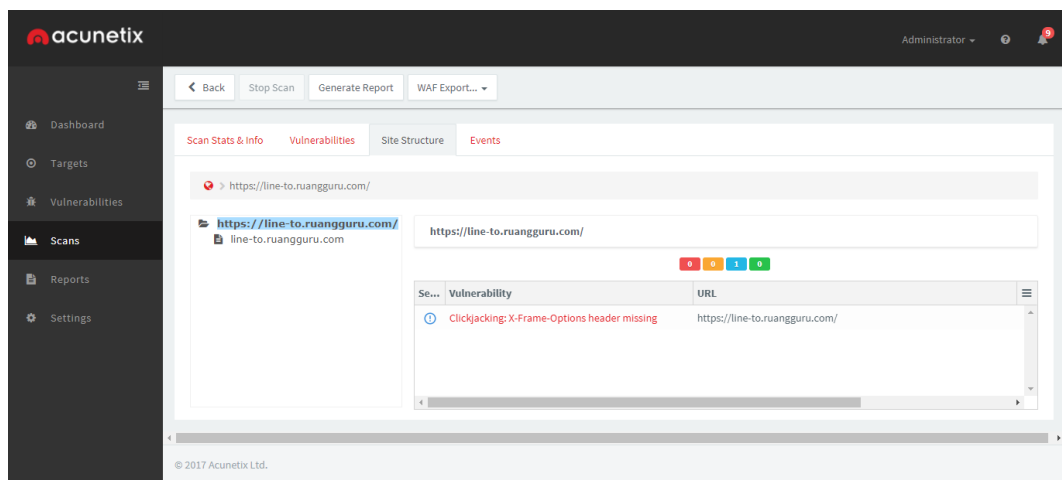
Gambar 54 menampilkan daftar kerentanan yang terdeteksi yang diurutkan dari tingkat keparahan.



Gambar 54. Tampilan Pengujian Security dengan Acunetix: Vulnerabilities

c. Site Structure

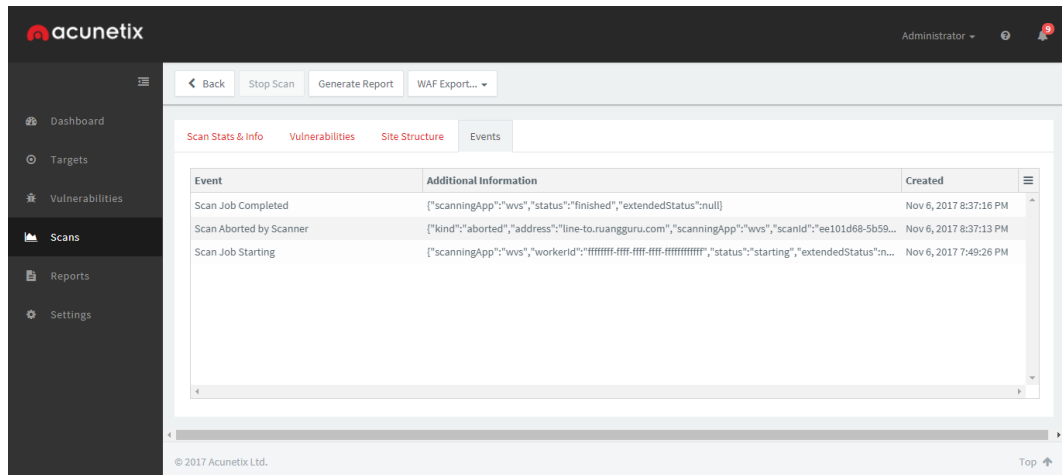
Site Structure digunakan untuk memastikan bahwa Acunetix telah mencakup semua situs, dan untuk mengidentifikasi kerentanan yang mempengaruhi file atau folder spesifik dari situs yang dipindai. Hasil pengujian *site structure* dapat dilihat pada Gambar 55 berikut ini:



Gambar 55. Tampilan Pengujian Security dengan Acunetix: Site Structure

d. Events

Gambar 56 menunjukkan daftar kejadian yang terkait selama pemindaian. Ini akan menunjukkan kapan pemindaian dimulai dan selesai, dan jika ada kesalahan yang ditemukan selama pemindaian.



Gambar 56. Tampilan Pengujian Security dengan Acunetix: Events

Dari data-data yang didapat di atas, disimpulkan bahwa perangkat lunak ujian daring memiliki tingkat keparahan yang rendah, yaitu *Low Risk Alert Level 1*. Tingkat keparahan keamanan di level 1 ini menandakan adanya kerentanan berasal dari kurangnya enkripsi lalu lintas data atau pengungkapan jalur direktori (Acunetix Product Manual, 2017).

5. Hasil Pengujian *Compatibility*

Pengujian *compatibility* tidak dilakukan karena seluruh informasi/sumber daya digunakan oleh perangkat lunak ujian daring itu sendiri.

E. Pembahasan Hasil Penelitian

Berikut ini akan dibahas mengenai hasil uji kualitas perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE pada aspek *functional suitability*, *performance efficiency*, *usability*, *security*, dan *compatibility* dari data yang sudah diambil dan diolah.

1. Pembahasan Hasil Pengujian *Functional Suitability*

Berdasarkan hasil pengujian kualitas perangkat lunak pada aspek *functional suitability* didapatkan tingkat keberhasilan sebesar 100%, dengan hasil

pengukuran menunjukkan nilai $X = 1$ sehingga disimpulkan perangkat lunak memiliki kualitas yang tinggi dan telah memenuhi aspek *functional suitability*.

2. Pembahasan Hasil Pengujian *Performance Efficiency*

Berdasarkan hasil pengujian kualitas perangkat lunak pada aspek *performance efficiency* didapatkan bahwa perangkat lunak ujian daring telah memenuhi aspek *performance efficiency*, karena rata-rata waktu yang dibutuhkan untuk memuat halaman adalah 1,8 detik, di atas standar PageSpeed dan YSlow.

3. Pembahasan Hasil Pengujian *Usability*

Berdasarkan hasil pengujian kualitas perangkat lunak pada aspek *usability* didapatkan hasil sebesar 82% dan memiliki kualitas **sangat layak**. Sedangkan item angketnya memiliki konsistensi yang masuk dalam kategori **good** karena mendapatkan nilai sebesar 0,864 pada perhitungan alpha cronbach.

4. Pembahasan Hasil Pengujian *Security*

Berdasarkan hasil pengujian kualitas perangkat lunak pada aspek *security* didapatkan hasil bahwa perangkat lunak ujian daring memiliki tingkat keparahan keamanan yang rendah, yaitu *Low Risk Alert Level 1*.

5. Pembahasan Hasil Pengujian *Compatibility*

Pengujian *compatibility* tidak dilakukan karena perangkat lunak ujian daring tidak berbagi informasi atau *resource* dengan produk, sistem, atau komponen lain. Jadi bisa ditarik kesimpulan bahwa tingkat keberhasilan pada aspek *compatibility* adalah 100%, sehingga memenuhi aspek *compatibility*.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

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

1. Penelitian ini menghasilkan perangkat lunak berbasis ruanguji (ruangguru) yang menyediakan fasilitas mengerjakan ujian daring bagi pengguna layanan percakapan LINE. Perangkat lunak ini memungkinkan pengguna LINE mengakses daftar ujian, melihat informasi melalui detail ujian, mengerjakan soal sesuai dengan ujian yang dipilih, mengetahui nilai ujian setelah berhasil mengumpulkan, dan melihat peringkat diri dibanding semua peserta serta pembahasan soal setelah masa ujian telah selesai.
2. Perangkat lunak ujian daring yang terintegrasi dengan layanan percakapan LINE diuji berdasarkan standar kualitas perangkat lunak ISO/IEC 25010 dengan aspek-aspek yang beririsan dengan Dimensions of Quality Pressmann, yang meliputi aspek *functional suitability*, *performance efficiency*, *usability*, *security* dan *compatibility*. Pada aspek *functional suitability*, perangkat lunak **telah memenuhi standar** karena fungsi berjalan 100%. Pada aspek *performance efficiency*, perangkat lunak **telah memenuhi standar** dengan rata-rata waktu untuk memuat halaman 1,8 detik. Pada aspek *usability*, perangkat lunak telah memenuhi standar dengan tingkat persetujuan pengguna sebesar 82% (**sangat layak**) dan nilai *alpha cronbach* 0,864 (**good**). Pada aspek *security*, perangkat lunak telah memenuhi standar dengan memiliki **tingkat keparahan keamanan yang rendah**, yaitu *Low Risk Alert*

Level 1. Tingkat keberhasilan pada aspek *compatibility* adalah 100% karena tidak ada informasi atau sumber daya lain yang dibagi di saat bersamaan dengan perangkat lunak ujian daring ini.

B. Keterbatasan Produk

Perangkat lunak yang dikembangkan masih memiliki keterbatasan dalam hal kelengkapan fitur seperti:

1. Pengguna hanya bisa mengerjakan soal yang disediakan oleh ruangguru, tidak ada fitur untuk membuat soal dari pengguna lain (guru)
2. Tidak adanya pemberitahuan langsung kepada pengguna ketika tryout ketika tryout telah selesai, sehingga pengguna tidak dapat melihat peringkat dan pembahasan dengan segera.

C. Pengembangan Produk Lebih Lanjut

Pengembangan perangkat lunak selanjutnya dapat menambahkan fitur membuat soal dari pengguna lain (guru), dan pemberitahuan langsung kepada pengguna ketika tryout yang diikuti telah berakhir baik melalui surat elektronik atau layanan pesan instan.

D. Saran

Mengingat keterbatasan yang dimiliki penulis baik dari segi pikiran maupun waktu, maka penulis menyarankan untuk pengembangan :

1. Fitur membuat soal dari pengguna lain (guru), dan pemberitahuan langsung kepada pengguna ketika tryout yang diikuti telah berakhir baik melalui surel atau SMS
2. Teknik pengujian kualitas perangkat lunak yang lebih lengkap dan beragam

DAFTAR PUSTAKA

- Acunetix. (2013). Acunetix User Manual Book. Retrieved March 25, 2015, from <https://http://www.acunetix.com/resources/wvsmanual.pdf>
- Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, Volume 12.
- Arifin, Yanuar. (2015). Pengembangan Aplikasi Pengelolaan Data Prestasi Mahasiswa Universitas Negeri Yogyakarta. Skripsi. Universitas Negeri Yogyakarta.
- A.S, Rosa, & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika
- Bahasa, Tim Penyusun Kamus Pusat. (Ed.) (2013) *Kamus Besar Bahasa Indonesia*. Jakarta: Pusat Bahasa.
- Bohang, Fatimah K. (2016). Di Indonesia Jumlah Pengguna LINE Pepet Facebook. Diakses dari <http://tekno.kompas.com/read/2016/09/03/09490637/di.indonesia.jumlah.pengguna.line.pepet.facebook>. pada tanggal 12 Agustus 2017
- Developers, Google. (2013). PageSpeed Insights Rules. Retrieved March 03, 2015, from <https://developers.google.com/speed/docs/insights/rules>
- Farooq, Umar., Azmat, Usman. (2009). *Testing Challenges in Web-based Applications with Respect to Interoperability and Integration*. Sweden : Blekinge Institute of Technology.
- Fowler, Martin. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. San Fransisco: Addison-Wesley Professional.
- Hamilton, Kim, & Miles, Russell. (2006). *Learning UML 2.0*. Sebastopol: O'Reilly Media.
- ISO. (2011). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models ISO/IEC 25010:2011*. Switzerland.

Jacobson, D., Brail, G., & Woods, D. (2012). *APIs: A Strategy Guide*. Sebastopol: O'Reilly Media.

LINE. (2017). LINE. Diakses dari <https://line.me/en/>. Pada tanggal 12 Agustus 2017.

Losby, Jan, & Wetmore, Anne. (2012). CDC Coffee Break: Using Likert Scales in Evaluation Survey Work. Diakses dari http://www.cdc.gov/dhds/pubs/docs/CB_February_14_2012.pdf. pada tanggal 12 Oktober 2017

Lund, Arnold M. (2001). Measuring Usability with the USE Questionnaire. Diakses dari https://www.researchgate.net/profile/Arnold_Lund/publication/230786746_Measuring_Usability_with_the_USE_Questionnaire/links/56e5a90e08ae98445c21561c/Measuring-Usability-with-the-USE-Questionnaire.pdf. pada tanggal 12 Oktober 2017.

Muderedzwa, Meshack, & Nyakwende, Emanuel. (2010). The effectiveness of online employment background screening systems. *African Journal of Business Management*, 4(17).

OECD. (2016). Science Performance (PISA). Diakses dari <https://data.oecd.org/pisa/science-performance-pisa.htm>. pada tanggal 12 Agustus 2017.

Pressman, Roger S. (2015). *Software Engineering A Practitioner's Approach* (8th ed.). New York: McGraw-Hill Education.

Purnama, A. (2017). Hingga Juli 2017, Jumlah Penduduk Indonesia Bertambah Jadi 262 Juta Jiwa Lebih. Diakses dari <http://jogja.tribunnews.com/2017/08/02/hingga-juli-2017-jumlah-penduduk-indonesia-bertambah-jadi-262-juta-jiwa-lebih>. pada tanggal 12 Agustus 2017.

Richardson, L. & Amundsen, M. (2013). *RESTful Web APIs*. Sebastopol: O'Reilly Media.

Ruangguru. (2017). Ruangguru. Diakses dari <https://ruangguru.com>. Pada tanggal 12 Agustus 2017.

- S.Gatot, B.Dewa, (2014). Siaran Pers tentang Riset Kominfo dan UNICEF Mengenai Perilaku Anak dan Remaja dalam Menggunakan Internet. https://www.kominfo.go.id/content/detail/3834/siaran-pers-no-17pihkominfo22014-tentang-ri-set-kominfo-dan-unicef-mengenai-perilaku-anak-dan-remaja-dalam-menggunakan-internet/0/siaran_pers. Pada 12 Agustus 2017.
- Sugiyono. (2015). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Bandung: Alfabeta.
- Vagias, Wade M. (2006). Likert-Type Scale Response Anchors. Diakses dari <http://www.clemson.edu/centers-institutes/tourism/documents/sample-scales.pdf>. pada 11 Oktober 2017
- Wagner, Stefan. (2013). *Software Product Quality Control*. Berlin: Springer.
- Wikipedia. (2017). AngularJS. Diakses dari <https://en.wikipedia.org/wiki/AngularJS> Pada tanggal 12 Agustus 2017.
- Wikipedia. (2017). ExpressJS. Diakses dari <https://en.wikipedia.org/wiki/Express.js>. Pada tanggal 12 Agustus 2017.
- Wikipedia. (2017). Line (software). Diakses dari [https://en.wikipedia.org/wiki/Line_\(software\)](https://en.wikipedia.org/wiki/Line_(software)). Pada tanggal 12 Agustus 2017.
- Yahoo. (2012). YSlow Ruleset Matrix. Retrieved March 03, 2015, from <http://yslow.org/ruleset-matrix/>

LAMPIRAN

Lampiran 1. Surat Pernyataan Validasi Instrumen

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

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Iman Usman

Jabatan : *Chief Product Officer (CPO) of Ruangguru*

menyatakan bawah instrumen penelitian TAS atas nama mahasiswa

Nama : Jilly Ratria Sari

NIM : 10520241033

Program Studi : Pendidikan Teknik Informatika


Judul TAS : Pengembangan Perangkat Lunak Ujian Daring yang
: Terintegrasi dengan Layanan Percakapan LINE

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.

Jakarta, 23 Oktober 2017

Validator,


Muhammad Iman Usman

Catatan:

- ☐ Beri tanda ✓

Lampiran 2. Pengujian Aspek *Functional Suitability*

INSTRUMEN PENELITIAN ASPEK *FUNCTIONAL SUITABILITY*
PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING
YANG TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE
BERDASARKAN ISO/IEC 25010

Nama : HASKA FIKRI FAUHI
 Profesi : SI FRONTEND DEVELOPER
 Instansi : RUANGGURU

Berilah tanda centang (✓) pada kolom yang disediakan sesuai dengan penilaian untuk pengujian *functional suitability*, dengan ketentuan penilaian sebagai berikut:

Ya = Jika fungsi **berfungsi** dengan benar

Tidak = Jika fungsi **tidak berfungsi** dengan benar

1. Instrumen *Functional Completeness* dan *Functional Appropriateness*

No.	Fungsi	Hasil yang diharapkan	Ya	Tidak
1.	Integrasi LINE	Fungsi <i>register</i> dan <i>login</i> otomatis dengan menggunakan akun LINE berjalan dengan benar	✓	
2.	<i>Onboard</i>	Fungsi untuk menampilkan kilasan tentang ujian daring yang terintegrasi dengan LINE berjalan dengan benar	✓	
3.	Daftar	Fungsi untuk mendaftar menjadi pengguna ujian daring berjalan dengan benar	✓	
4.	Masuk	Fungsi untuk masuk ke dalam dasbor aplikasi berjalan dengan benar	✓	
5.	Data daftar <i>tryout</i>	Fungsi untuk melihat daftar <i>tryout</i> berjalan dengan benar	✓	
6.	Data detail <i>tryout</i>	Fungsi untuk melihat detail <i>tryout</i> yang berisi paket-paket ujian berjalan dengan benar	✓	
7.	Data detail paket soal	Fungsi untuk melihat detail paket soal berjalan dengan benar	✓	

No.	Fungsi	Hasil yang diharapkan	Ya	Tidak
8.	Mulai ujian	Fungsi untuk memulai ujian berjalan dengan benar	✓	
9.	Data soal	Fungsi untuk menampilkan soal-soal sesuai dengan paket data yang dipilih berjalan dengan benar	✓	
10.	Mengumpulkan soal	Fungsi untuk mengumpulkan soal berjalan dengan benar	✓	
11.	Data hasil ujian	Fungsi untuk menampilkan hasil ujian berjalan dengan benar	✓	
12.	Data pembahasan ujian	Fungsi untuk menampilkan pembahasan dari ujian yang telah dikerjakan berjalan dengan benar	✓	
13.	Data peringkat peserta	Fungsi untuk menampilkan peringkat pengguna dibanding dengan keseluruhan peserta yang mengerjakan ujian berjalan dengan benar	✓	
14.	Data profil	Fungsi untuk menampilkan data profil pengguna yang sedang aktif berjalan dengan benar	✓	
15.	Keluar	Fungsi untuk keluar dari dasbor sudah berjalan dengan benar	✓	

2. Instrumen *Functional Correctness*

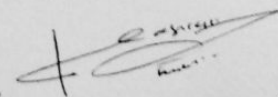
No	Fungsi	Hasil yang diharapkan	Ya	Tidak
1.	Identifikasi <i>email</i> dan <i>password</i>	Fungsi untuk masuk sistem sesuai dengan pengguna sudah berjalan dengan benar	✓	
2.	Menampilkan data <i>tryout</i>	Fungsi untuk menampilkan data <i>tryout</i> sudah berjalan dengan benar	✓	
3.	Menampilkan detail <i>tryout</i>	Fungsi untuk menampilkan data detail <i>tryout</i> yang berisi paket-paket soal sudah berjalan dengan benar	✓	

No.	Fungsi	Hasil yang diharapkan	Ya	Tidak
4.	Menampilkan paket soal	Fungsi untuk menampilkan detail paket soal sudah berjalan dengan benar	✓	
5.	Memulai ujian	Fungsi untuk memulai ujian sudah berjalan dengan benar	✓	
6.	Menampilkan soal-soal	Fungsi untuk menampilkan soal-soal sesuai dengan paket ujian yang dipilih sudah berjalan dengan benar	✓	
7.	Mengumpulkan ujian	Fungsi untuk mengumpulkan ujian sudah berjalan dengan benar	✓	
8.	Menampilkan hasil ujian	Fungsi untuk menampilkan hasil ujian sudah berjalan dengan benar	✓	
9.	Menampilkan pembahasan	Fungsi untuk menampilkan pembahasan ujian sudah berjalan dengan benar	✓	
10.	Menampilkan peringkat	Fungsi untuk menampilkan peringkat peserta sudah berjalan dengan benar	✓	

Saran:

Jakarta, 24 Oktober 2017

Penguji,


 (Hasfa Fikri Tawhid)

Lampiran 3. Pengujian Aspek *Usability*

INSTRUMEN PENELITIAN ASPEK *USABILITY*
APLIKASI PENGEMBANGAN PERANGKAT LUNAK UJIAN DARING
YANG TERINTEGRASI DENGAN LAYANAN PERCAKAPAN LINE
BERDASARKAN ISO/IEC 25010

Nama : Nia fadhia yunitami
 Profesi : Pelajar
 Instansi : SMP NEGERI 5 WATES

Lingkarilah angka pada kolom skala yang disediakan sesuai dengan penilaian dengan pengujian *usability*, dengan ketentuan sebagai berikut:

- 1 = Sangat tidak setuju**
2 = Tidak setuju
3 = Ragu-ragu
4 = Setuju
5 = Sangat setuju

No	Instrumen	Skala				
Usefulness						
1.	Aplikasi ini membantu saya menjadi lebih efektif dalam mengerjakan ujian	1	2	3	④	5
2.	Aplikasi ini membantu saya menjadi lebih produktif dalam belajar	1	2	3	4	⑤
3.	Aplikasi ini bermanfaat	1	2	3	4	⑤
4.	Aplikasi ini memberi saya dampak yang besar terhadap tugas yang saya lakukan dalam mengerjakan soal-soal ujian daring	1	2	3	④	5
5.	Aplikasi ini memudahkan saya mencapai hal-hal yang saya inginkan dalam mengerjakan ujian daring	1	2	3	4	⑤
6.	Aplikasi ini menghemat waktu ketika saya menggunakannya	1	2	3	4	⑤
7.	Aplikasi ini sesuai dengan kebutuhan saya	1	2	③	4	5

No	Instrumen	Skala				
8.	Aplikasi ini bekerja sesuai dengan apa yang saya harapkan	1	2	3	4	5
<i>Easy of Use</i>						
9.	Aplikasi ini mudah digunakan	1	2	3	4	5
10.	Aplikasi ini praktis untuk digunakan	1	2	3	4	5
No	Instrumen					
11.	Aplikasi ini mudah dipahami	1	2	3	4	5
12.	Aplikasi ini memerlukan langkah-langkah yang praktis untuk mencapai apa yang saya kerjakan	1	2	3	4	5
13.	Aplikasi ini dapat disesuaikan dengan kebutuhan	1	2	3	4	5
14.	Tidak kesulitan menggunakan aplikasi ini	1	2	3	4	5
15.	Saya dapat menggunakan tanpa instruksi tertulis	1	2	3	4	5
16.	Saya tidak melihat adanya ketidakonsistenan selama saya menggunakannya	1	2	3	4	5
17.	Pengguna yang jarang maupun rutin menggunakan akan menyukai sistem ini	1	2	3	4	5
18.	Saya dapat kembali dari kesalahan dengan cepat dan mudah	1	2	3	4	5
19.	Saya dapat menggunakan sistem ini dengan berhasil	1	2	3	4	5
<i>Easy of learning</i>						
20.	Saya belajar menggunakan aplikasi ini dengan cepat	1	2	3	4	5
21.	Saya mudah mengingat bagaimana cara menggunakan aplikasi ini	1	2	3	4	5
22.	Aplikasi ini mudah untuk dipelajari cara menggunakannya	1	2	3	4	5
23.	Saya cepat menjadi terampil dengan aplikasi ini	1	2	3	4	5
<i>Satisfaction</i>						
24.	Saya puas dengan aplikasi ini	1	2	3	4	5
25.	Saya akan merekomendasikan sistem informasi ini kepada teman	1	2	3	4	5
26.	Aplikasi ini menyenangkan untuk digunakan	1	2	3	4	5
27.	Aplikasi ini bekerja seperti yang saya inginkan	1	2	3	4	5

No	Instrumen	Skala				
28.	Aplikasi ini sangat bagus	1	2	3	4	5
29.	Saya merasa saya harus memiliki aplikasi ini	1	2	3	4	5
30.	Aplikasi ini nyaman untuk digunakan.	1	2	3	4	5

Saran:

DEYAKARTA, Oktober 2017

Penguji



(NIA FADILA Y)