

**PEMBUATAN APLIKASI *COMPUTER BASED TEST*  
BERBASIS WEB UNTUK SELEKSI  
TES KARYAWAN  
(STUDI KASUS PT RADYA GITA BAHAGI)**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh kelulusan**

**Jenjang Strata Satu (S1)**

**Pada program Studi Teknik Informatika**

**Oleh**

**Wanti Fahrianti**

**361743004**



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER  
INDONESIA MANDIRI**

**2021**

**LEMBAR PENGESAHAN**

**PEMBUATAN APLIKASI *COMPUTER BASED TEST*  
BERBASIS WEB UNTUK SELEKSI TES KARYAWAN  
(STUDI KASUS PT RADYA GITA BAHAGI)**

**Oleh**

**Wanti Fahrianti**

361743004

Skripsi ini telah diterima dan disahkan untuk  
memenuhi persyaratan mencapai gelar

**SARJANA TEKNIK INFORMATIKA**

**Pada**

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
INDONESIA MANDIRI**

Bandung, 9 Februari 2021

Disahkan oleh

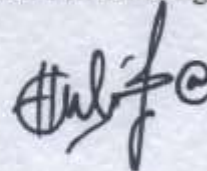
Ketua Program Studi,


**Chalifa Chazar, S.T., M.T**

NIDN. 0421098704

Dosen Pembimbing,



**Chalifa Chazar, S.T., M.T**

NIDN. 0421098704

**LEMBAR PERSETUJUAN REVISI**

**PEMBUATAN APLIKASI *COMPUTER BASED TEST*  
BERBASIS WEB UNTUK SELEKSI TES KARYAWAN  
(STUDI KASUS PT RADYA GITA BAHAGI)**

**Oleh**

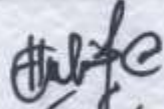
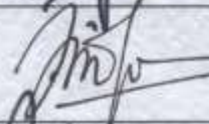
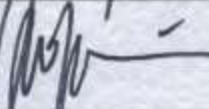
Wanti Fahrianti

361743004

Telah melakukan sidang skripsi dan telah melakukan revisi sesuai dengan perubahan dan perbaikan yang diminta pada saat sidang skripsi.

Bandung, 9 Februari 2021

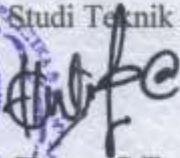
Menyetujui

No	Nama Dosen	Keterangan	Tanda Tangan
1	Chalifa Chazar, S.T., M.T.	Pembimbing	
2	Hendra Gunawan, S.T., M.Kom.	Penguji 1	
3	Novi Rukhviyanti, S.T., M.Si.	Penguji 2	

Mengetahui

Ketua Program Studi Teknik Informatika,



  
Chalifa Chazar, S.T., M.T

NIDN. 0421098704

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

- 1) Naskah Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik, baik di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri maupun perguruan tinggi lainnya.
- 2) Skripsi ini murni merupakan karya penelitian saya sendiri dan tidak menjiplak karya pihak lain. Dalam hal ada bantuan atau arahan dari pihak lain maka telah saya sebutkan identitas dan jenis bantuannya di dalam lembar ucapan terima kasih.
- 3) Seandainya ada karya pihak lain yang ternyata memiliki kemiripan dengan karya saya ini, maka hal ini adalah di luar pengetahuan saya dan terjadi tanpa kesengajaan dari pihak saya.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terbukti adanya kebohongan dalam pernyataan ini, maka saya bersedia menerima sanksi akademik sesuai norma yang berlaku di Sekolah Tinggi Manajemen Informatika dan Komputer Indonesia Mandiri.

Bandung, 9 Februari 2021



Wanti Fahranti

361743004

## ABSTRAK

# PEMBUATAN APLIKASI *COMPUTER BASED TEST* BERBASIS WEB UNTUK SELEKSI TES KARYAWAN (STUDI KASUS PT RADYA GITA BAHAGI)

Oleh

Wanti Fahrianti  
361743004

PT Radya Gita Bahagi memiliki visi untuk dikenal sebagai *Creative Digital Media Agency* yang dapat menciptakan produk yang baik dan inovatif. Untuk itu peran karyawan sangat penting bagi PT Radya Gita Bahagi sehingga rekrutmen karyawan yang dilakukan pun menjadi sangat penting agar mendapatkan karyawan yang memiliki potensi kreatif dan berkualitas bagi perusahaan. Demi mendapatkan karyawan yang berpotensi PT Radya Gita Bahagi menerapkan proses rekrutmen yang sangat panjang. Hal ini membuat perusahaan lama dalam mendapatkan karyawan yang dibutuhkan. Berdasarkan hal tersebut maka penulis membuat sistem aplikasi berbasis *Computer Based Testing (CBT)* yang dapat mempermudah perusahaan dalam proses seleksi tes karyawan dengan memperhatikan faktor *usability end user* dalam pembuatannya, sehingga perusahaan bisa lebih cepat mendapatkan kandidat calon karyawan untuk di proses lebih lanjut. Metode yang digunakan sebagai acuan dalam membangun aplikasi pada penelitian ini menggunakan metode pengembangan sistem aplikasi, yaitu metode *System Development Live Cycle (SDLC)* dengan model *Prototype*. Namun pada penelitian ini penulis hanya melakukan sampai dengan tahap pengujian sistem. Dari penelitian ini dihasilkan aplikasi seleksi tes karyawan yang dapat mempermudah HR dalam menentukan kandidat calon karyawan yang akan maju tahap selanjutnya berdasarkan *score test* yang dihasilkan, sehingga dapat mempercepat proses rekrutmen. Dan juga berdasarkan hasil rekap nilai kuisisioner dalam menentukan faktor *usability* aplikasi, di dapatkan nilai rata-rata keseluruhan di atas 3,5 sehingga aplikasi rekrutmen ini sudah memenuhi aspek *usability*.

**Kata Kunci :** *Computer Based Test, Rekrutmen, Web, Prototype*

## **ABSTRACT**

### **MAKING WEB-BASED COMPUTER BASED TEST APPLICATION FOR EMPLOYEE TEST SELECTION (CASE STUDY PT RADYA GITA BAHAGI)**

**By**

Wanti Fahrianti  
361743004

*PT Radya Gita Bahagi has a vision to be known as a Creative Digital Media Agency that can create good and innovative products. For this reason, the role of employees is very important for PT Radya Gita Bahagi so that employee recruitment is very important in order to get employees who have creative and quality potential for the company. In order to get potential employees, PT Radya Gita Bahagi implements a very long recruitment process. This keeps the old company from getting the employees it needs. Based on this, the authors made an application system based on Computer Based Testing (CBT) which can facilitate the company in the employee test selection process by paying attention to the end user usability factor in making it, so that companies can more quickly get prospective employees for further processing. The method used as a reference in building applications in this study uses the application system development method, namely the System Development Live Cycle (SDLC) method with the Prototype model. However, in this study the authors only did up to the system testing stage. From this research, a employee test selection application is produced that can make it easier for HR to determine prospective employee candidates who will advance to the next stage based on the resulting test score, so as to speed up the recruitment process. And also based on the results of the recapitulation of the questionnaire scores in determining the usability factor of the application, the overall average value is above 3.5 so that this recruitment application has met the usability aspect.*

**Keyword :** *Computer Based Test, Recruitment, Web, Prototype*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Pembuatan Aplikasi Berbasis *Computer Based Test* Berbasis Web Untuk Seleksi Tes Karyawan (Studi Kasus PT Radya Gita Bahagi)”**. Skripsi ini di susun untuk memenuhi persyaratan kelulusan pada Program Studi Sistem Informasi di Sekolah Tinggi Manajemen Informatika & Komputer Indonesia Mandiri (STMIK-IM Bandung).

Penulis menyadari masih banyak kesalahan maupun kekurangan dalam penyusunan skripsi ini, tanpa dorongan, dukungan, motivasi, bimbingan, dan doa maka skripsi ini tidak akan terlaksana. Pada kesempatan kali ini penulis ingin mengucapkan terimakasih kepada seluruh pihak yang berperan dalam penyusunan skripsi ini.

Besar harapan penulis semoga skripsi ini bermanfaat untuk semua pihak yang membutuhkan pada umumnya dan penulis khususnya.

Bandung, 9 Februari 2021

Wanti Fahrianti

## UCAPAN TERIMA KASIH

Dalam penyusunan skripsi ini tidak terlepas dukungan dari berbagai pihak. Penulis secara khusus mengucapkan terima kasih yang sebesar besarnya kepada semua pihak yang telah membantu. Penulis banyak menerima bimbingan, petunjuk dan bantuan serta dorongan dari berbagai pihak baik yang bersifat moral maupun material. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT dengan segala rahmat serta karunia-Nya yang memberikan kekuatan bagi penulis dalam menyelesaikan skripsi ini.
2. Kepada kedua orang tua tercinta yang selama ini telah membantu penulis dalam bentuk perhatian, kasih sayang, semangat, serta doa yang tidak henti-hentinya mengalir demi kelancaran dan kesuksesan penulis dalam menyelesaikan skripsi ini. Kemudian terima kasih banyak untuk adik tercinta Anwariah Syarafina yang telah memberikan dukungan serta perhatian kepada penulis.
3. Ibu Chalifa Chazar, S.T., M.T. selaku pembimbing dan Kaprodi Teknik Informatika yang telah memberikan penulis ilmu dan masukan dalam penyusunan skripsi.
4. Bapak Hendra Gunawan, S.T., M.Kom. dan Ibu Novi Rukhviyanti, S.T., M.Si. selaku penguji pada sidang skripsi yang telah memberikan masukan dalam skripsi.



5. PT Radya Gita Bahagi yang telah memberikan kesempatan penulis untuk melakukan kegiatan penelitian dan membuat web aplikasi rekrutmen guna mendukung kelancaran penyusunan skripsi.
6. Kak Damas dan Kak Allez selaku mentor pada PT Radya Gita Bahagi yang memberikan arahan tentang aplikasi yang dibutuhkan perusahaan.
7. Fitria Wahyuni rekan yang membantu membimbing penulis tentang program disaat penulis kesulitan dalam pembangunan aplikasi.
8. Bapak Dr. Chairuddin, Ir., M.M., M.T. selaku Ketua STMIK-IM Bandung yang telah memberikan ilmu kepada penulis.
9. Bapak & Ibu dosen yang telah memberikan penulis ilmu selama penulis melakukan kegiatan pembelajaran di STMIK-IM Bandung.
10. Kepala dan seluruh staff Administrasi, BAAK, BAUK dan staff Perpustakaan STMIK-IM Bandung.
11. Rekan dan sahabat di STMIK-IM Bandung maupun di PT Radya Gita Bahagi.
12. Serta masih banyak lagi pihak-pihak yang sangat berpengaruh dalam proses penyelesaian skripsi yang yang tidak bisa penulis sebutkan satupersatu.

Semoga Allah SWT senantiasa membalas semua kebaikan yang telah diberikan. Semoga penelitian ini dapat bermanfaat bagi penulis umumnya kepada para pembaca.

Bandung, 9 Februari 2021

Wanti Fahrianti

## DAFTAR ISI

LEMBAR PENGESAHAN .....	<b>Error! Bookmark not defined.</b>
LEMBAR PERSETUJUAN REVISI.....	<b>Error! Bookmark not defined.</b>
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
UCAPAN TERIMA KASIH.....	vii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xv
BAB I .....	1
PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Identifikasi Masalah .....	3
1.3    Tujuan Penulisan.....	3
1.4    Batasan Masalah .....	4
1.5    Metode Penelitian .....	4
1.6    Sistematika Penulisan .....	6
BAB II .....	8
LANDASAN TEORI.....	8
2.1    Profil Perusahaan .....	8
2.2.1    Gambaran Umum Perusahaan .....	8
2.2.2    Struktur Organisasi Perusahaan .....	10
2.2    Pengertian <i>Prototype</i> .....	10
2.3    Pengertian Website.....	12
2.4    Pengertian Aplikasi .....	13
2.5    Pengertian <i>Computer Based Test</i> .....	14
2.6    Pengertian Rekrutmen .....	15
2.7 <i>Web Browser</i> .....	16
2.8 <i>Flowchart</i> .....	17

2.9	<i>Usability</i> .....	18
2.10	UML.....	20
2.11	<i>Blackbox Testing</i> .....	26
BAB III.....		28
ANALISA KEBUTUHAN DAN PERANCANGAN SISTEM .....		28
3.1	Analisis kebutuhan .....	28
3.1.1	Studi Litelatur .....	28
3.1.2	Analisis Sistem Yang Berjalan Saat Ini .....	34
3.1.3	Analisis Masalah .....	36
3.1.4	Sistem Yang Diusulkan .....	36
3.1.5	Analisis Kebutuhan Sistem .....	37
3.1.6	<i>Planning</i> .....	38
3.2	Membangun Prototype.....	39
3.2.1	<i>Use Case Diagram</i> .....	39
3.2.2	<i>Activity Diagram</i> .....	44
3.2.3	<i>Sequence Diagram</i> .....	52
3.2.4	<i>Class Diagram</i> .....	54
3.2.5	<i>Entity Relationship Diagram</i> .....	56
3.2.6	Perancangan Antarmuka.....	58
BAB IV.....		63
IMPLEMENTASI DAN UJI COBA.....		63
4.1	Pengkodean Sistem .....	63
4.1.1	Implementasi.....	63
4.2	Menguji Sistem.....	71
4.2.1	<i>Blackbox Testing</i> .....	71
4.2.2	Kusioner .....	75
BAB V .....		79
PENUTUP.....		79
5.1	Kesimpulan .....	79
5.2	Saran .....	80
Daftar pustaka.....		81

Lampiran.....	84
---------------	----

## DAFTAR GAMBAR

GAMBAR: 1.1 Metode Prototype (Jaya and Widyawati, 2019) .....	5
GAMBAR: 2. 1 Logo PT Radya Gita Bahagi (Rolling Glory, 2012) .....	8
GAMBAR: 2. 2 Struktur Organisasi (Rolling Glory, 2012) .....	10
GAMBAR: 3. 1 <i>Flowchart</i> Proses Rekrutmen PT Radya Gita Bahagi .....	35
GAMBAR: 3. 2 <i>Flowchart</i> Proses Yang Diusulkan .....	37
GAMBAR: 3. 3 <i>Use Case</i> Calon Karyawan .....	40
GAMBAR: 3. 4 <i>Use Case</i> Admin .....	41
GAMBAR: 3. 5 <i>Activity</i> Diagram Register Calon Karyawan .....	44
GAMBAR: 3. 6 <i>Activity</i> Diagram Calon Karyawan Mengisi Test .....	45
GAMBAR: 3. 7 <i>Activity</i> Diagram Admin Login .....	46
GAMBAR: 3. 8 <i>Activity</i> Diagram Admin Logout .....	46
GAMBAR: 3. 9 <i>Activity</i> Diagram Admin Menambah Pertanyaan .....	47
GAMBAR: 3. 10 <i>Activity</i> Diagram Admin Mengedit Pertanyaan .....	48
GAMBAR: 3. 11 <i>Activity</i> Diagram Admin Menghapus Pertanyaan .....	49
GAMBAR: 3. 12 <i>Activity</i> Diagram Admin Melihat Hasil Test .....	50
GAMBAR: 3. 13 <i>Activity</i> Diagram Admin Menghapus Hasil Test .....	51
GAMBAR: 3. 14 <i>Sequence</i> Diagram Calon Karyawan Register .....	52
GAMBAR: 3. 15 <i>Sequence</i> Diagram Calon Karyawan Mengisi Test .....	52
GAMBAR: 3. 16 <i>Sequence</i> Diagram Admin Login .....	53
GAMBAR: 3. 17 <i>Sequence</i> Diagram Admin Mengelola Hasil .....	53
GAMBAR: 3. 18 <i>Sequence</i> Diagram Admin Mengelola Pertanyaan .....	54

GAMBAR: 3. 19 <i>Sequence</i> Diagram Admin Logout .....	54
GAMBAR: 3. 20 <i>Class</i> Diagram Aplikasi Rekrutment.....	55
GAMBAR: 3. 21 <i>Entity Relationship</i> Diagram Aplikasi Rekrutment.....	56
GAMBAR: 3. 22 Antarmuka halaman register versi desktop .....	59
GAMBAR: 3. 23 Antarmuka halaman pembuka versi desktop.....	59
GAMBAR: 3. 24 Antarmuka halaman test versi desktop.....	60
GAMBAR: 3. 25 Antarmuka popup saat memilih jawaban versi desktop .....	60
GAMBAR: 3. 26 Antarmuka halaman register versi mobile.....	61
GAMBAR: 3. 27 Antarmuka halaman pembuka versi mobile .....	61
GAMBAR: 3. 28 Antarmuka halaman test versi mobile .....	62
GAMBAR: 3. 29 Antarmuka popup saat memilih jawaban versi mobile .....	62
GAMBAR: 4. 1 Antarmuka Halaman Selamat Datang Ver Desktop.....	63
GAMBAR: 4. 2 Antarmuka Halaman Register Ver Desktop .....	64
GAMBAR: 4. 3 Antarmuka Halaman Panduan Ver Desktop.....	64
GAMBAR: 4. 4 Antaramuka Halaman Test Ver Desktop.....	65
GAMBAR: 4. 5 Antarmuka Popup Ver Desktop.....	65
GAMBAR: 4. 6 Antarmuka Halaman Selesai Ver Desktop .....	66
GAMBAR: 4. 7 Antarmuka Halaman Penutup Ver Desktop .....	66
GAMBAR: 4. 8 Antarmuka Halaman Selamat Datang Ver Mobile.....	67
GAMBAR: 4. 9 Antarmuka Halaman Register Ver Mobile.....	68
GAMBAR: 4. 10 Antarmuka Halaman Panduan Ver Mobile .....	68
GAMBAR: 4. 11 Antarmuka Halaman Test Ver Mobile .....	69
GAMBAR: 4. 12 Antaarmuka Popup Ver Mobile .....	69

GAMBAR: 4. 13 Antarmuka Halaman Selesai Ver Mobile .....	70
GAMBAR: 4. 14 Antarmuka Halaman Penutup Ver Mobile .....	70

## DAFTAR TABEL

TABEL: 2. 1 Simbol <i>Use Case</i> Diagram (A.S Rosa, 2014) .....	21
TABEL: 2. 2 Simbol <i>Activity</i> Diagram (A.S Rosa, 2014) .....	23
TABEL: 2. 3 Simbol <i>Sequence</i> Diagram (A.S Rosa, 2014) .....	24
TABEL: 2. 4 Simbol <i>Class</i> Diagram (A.S Rosa, 2014).....	25
TABEL: 2. 5 Simbol <i>Class</i> Diagram (A.S Rosa, 2014).....	26
TABEL: 3. 1 Tabel Studi Literatur .....	29
TABEL: 3. 2 Tabel <i>Planning</i> .....	39
TABEL: 3. 3 Tabel Deskripsi Aktor .....	40
TABEL: 3. 4 Tabel Skenario <i>Use Case</i> Melakukan Register .....	41
TABEL: 3. 5 Tabel Skenario <i>Use Case</i> Mengisi Test .....	41
TABEL: 3. 6 Tabel Skenario <i>Use Case</i> Melakukan Login.....	42
TABEL: 3. 7 Tabel Skenario <i>Use Case</i> Melihat Hasil .....	42
TABEL: 3. 8 Tabel Skenario <i>Use Case</i> Menghapus Hasil .....	42
TABEL: 3. 9 Tabel Skenario <i>Use Case</i> Menambah Pertanyaan .....	43
TABEL: 3. 10 Tabel Skenario <i>Use Case</i> Mengedit Pertanyaan .....	43
TABEL: 3. 11 Tabel Skenario <i>Use Case</i> Menghapus Pertanyaan.....	43
TABEL: 3. 12 Tabel Skenario <i>Use Case</i> Melakukan <i>Logout</i> .....	44
TABEL: 3. 13 Tabel Data Admin_User .....	56
TABEL: 3. 14 Tabel Data Dm_Quiz .....	57
TABEL: 3. 15 Tabel Data Session.....	57
TABEL: 3. 16 Tabel Data Session_Quiz .....	58
TABEL: 4. 1 Pengujian <i>Blackbox</i> Pada Register Calon Karyawan.....	71



TABEL: 4. 2 Pengujian <i>Blackbox</i> Pada Pengisian Test.....	72
TABEL: 4. 3 Pengujian <i>Blackbox</i> Pada Login Admin .....	72
TABEL: 4. 4 Pengujian <i>Blackbox</i> Pada Lihat Hasil Test oleh Admin .....	73
TABEL: 4. 5 Pengujian <i>Blackbox</i> Pada Tambah Pertanyaan oleh Admin .....	73
TABEL: 4. 6 Pengujian <i>Blackbox</i> Pada Edit Pertanyaan oleh Admin .....	74
TABEL: 4. 7 Pengujian <i>Blackbox</i> Pada Hapus Pertanyaan oleh Admin .....	75
TABEL: 4. 8 Pengujian <i>Blackbox</i> Pada Logout Admin .....	75
TABEL: 4. 9 Hasil Kuisisioner untuk mengukur faktor <i>Usability</i> .....	76
TABEL: 4. 10 Hasil Kuisisioner untuk mengukur faktor <i>Usability</i> .....	77

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

PT Radya Gita Bahagi merupakan perusahaan *media creative digital* yang berlokasi di Bandung. PT Radya Gita Bahagi memiliki visi untuk dikenal sebagai *Creative Digital Media Agency* yang dapat menciptakan produk yang baik dan inovatif. PT Radya Gita Bahagi juga berharap dapat memberikan kontribusi kepada orang-orang yang membutuhkan *Creative Digital Media Agency* dengan kemampuan dan pengetahuan PT Radya Gita Bahagi sebagai tim *Creative Digital*. PT Radya Gita Bahagi memiliki misi untuk menciptakan produk digital yang dapat berintergrasi, dapat menyampaikan pesan dengan baik, dan dapat menyelesaikan permasalahan yang ada, dengan kualitas yang premium diluar ekspektasi user. Untuk itu peran karyawan sangat penting bagi PT Radya Gita Bahagi sehingga rekrutmen karyawan yang dilakukan pun menjadi sangat ketat untuk mendapatkan karyawan yang sesuai dengan perusahaan.

Dalam proses rekrutmen, PT Radya Gita Bahagi sangat memperhatikan *soft skill* yang dimiliki oleh calon karyawan. Karena selain memiliki kemampuan yang dibutuhkan, penting bagi calon karyawan untuk mempunyai *soft skill* yang baik yang sesuai dengan *value* perusahaan saat ini. Proses rekrutmen menjadi sangat penting agar mendapatkan karyawan yang memiliki potensi kreatif dan berkualitas bagi perusahaan.

Demi mendapatkan karyawan yang berpotensi PT Radya Gita Bahagi menerapkan proses rekrutmen yang sangat panjang dan sangat berbeda dengan proses rekrutmen pada umumnya. Proses yang digunakan saat ini terdiri dari beberapa tahap yaitu *screening CV*, pengisian *google form*, *screening* karyawan, *technical test*, *conference call* interview sesi 1, *onsite* interview sesi 2, *onsite* interview sesi 3 dan *onsite* interview sesi 4 dan karena *pandemic* yang sedang berlangsung di Indonesia maka perusahaan menerapkan *Work From Home* (WFH) yang menjadikan interview *onsite* pun diganti menjadi *interview conference call* tanpa mengurangi tahap seleksi. Proses yang panjang ini perusahaan memerlukan waktu kurang lebih 2 minggu dalam proses rekrutment untuk 1 orang calon karyawan dan tidak jarang calon karyawan gagal setelah melewati proses interview sesi 4, sehingga membuat perusahaan lama dalam mendapatkan karyawan yang dibutuhkan.

Dengan keadaan perusahaan yang sedang sangat berkembang saat ini, perusahaan semakin membutuhkan banyak karyawan dengan cepat. Berdasarkan hal tersebut maka dibutuhkan sebuah sistem aplikasi yang dapat mempermudah perusahaan dalam proses seleksi karyawan . Maka penulis membuat sistem aplikasi berbasis *Computer Based Testing* (CBT) yang dapat mempermudah perusahaan dalam proses tes seleksi karyawan dengan memperhatikan faktor *usability end user* dalam pembuatannya. Sistem aplikasi berbasis CBT ini tidak menghapuskan proses *technical test* dan interview yang sudah ada, namun aplikasi ini dapat dijadikan sistem yang mendiagnosa calon karyawan pada awal proses apakah calon karyawan berpotensi dan sudah sesuai dengan yang dibutuhkan perusahaan. Sehingga pihak

perusahaan dapat terfokus secara detail melakukan *technical test* dan interview hanya kepada kandidat calon karyawan yang sudah sesuai dengan kualifikasi perusahaan dan dapat segera menggugurkan calon karyawan jika tidak sesuai dengan yang dibutuhkan perusahaan, hal ini akan memangkas waktu, maka perusahaan bisa lebih cepat mendapatkan kandidat calon karyawan untuk di proses lebih lanjut.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang sudah dipaparkan sebelumnya, maka dapat diidentifikasi masalah yang ada yaitu :

1. Bagaimana membuat aplikasi untuk proses tes awal seleksi karyawan?
2. Bagaimana membuat sebuah aplikasi yang nyaman dan memudahkan *end user*?

## 1.3 Tujuan Penulisan

Tujuan penelitian yang dilakukan oleh penulis adalah :

1. Membuat web aplikasi berbasis *Computer Based Testing* untuk proses tes awal seleksi karyawan.
2. Membuat web aplikasi dengan memperhatikan faktor *usability* untuk kenyamanan dan memudahkan *end user* dalam menggunakannya.

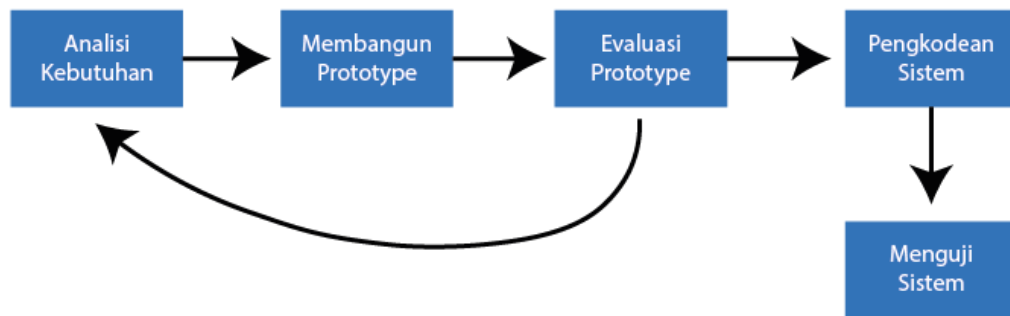
#### 1.4 Batasan Masalah

Batasan masalah dalam penelitian ini dibutuhkan agar dapat fokus dan tidak terlalu meluas maupun keluar dari batas konteks yang ada, maka batasan-batasan penelitian ini yaitu :

1. Aplikasi CBT berbasis web yang dibuat pada penelitian berdasarkan studi kasus PT Radya Gita Bahagi.
2. Aplikasi CBT berbasis web yang dibuat pada penelitian terfokus pada proses seleksi tes awal proses rekrutmen pada perusahaan.
3. Aplikasi CBT berbasis web ini hanya untuk proses tes general bukan untuk setiap posisi/kategori yang berbeda.
4. Penelitian ini hanya terfokus pada factor *usability end user* dalam pembuatannya.

#### 1.5 Metode Penelitian

Metode yang digunakan sebagai acuan dalam membangun aplikasi pada penelitian ini menggunakan metode pengembangan sistem aplikasi, yaitu metode *System Development Live Cycle* (SDLC), metode SDLC yang digunakan adalah model *Prototype*. Namun pada penelitian ini penulis hanya melakukan sampai dengan tahap pengujian sistem. Adapun model *prototype* mempunyai tahapan-tahapan pada Gambar 1.1 :



GAMBAR: 1.1 Metode Prototype (Jaya and Widyawati, 2019)

a. Analisis Kebutuhan

Dalam tahap penelitian ini akan dilakukan literatur dan metode untuk mengumpulkan data, informasi, permasalahan, dan kebutuhan untuk sistem yang dibangun.

b. Membangun *Prototype*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus pada penyajian kepada perusahaan.

c. Evaluasi *Prototype*

Pada tahap ini perusahaan akan melakukan evaluasi apakah *prototype* sudah sesuai atau belum.

d. Pengkodean Sistem

Pada tahap ini *prototype* yang sudah disetujui akan diubah ke dalam bahasa pemrograman.

e. Menguji sistem

Di tahap ini dilakukan pengujian aplikasi yang sudah dibuat. Pengujian dilakukan dengan *Black Box Testing* dan melakukan percobaan kepada beberapa responden lalu disediakan form evaluasi untuk mengetahui apakah sistem aplikasi sudah memenuhi faktor *usability*.

## 1.6 Sistematika Penulisan

Sistematika penulisan ini meliputi lima bagian yaitu terdiri dari:

### BAB I : PENDAHULUAN

Pada bab ini dibahas mengenal latar belakang, identifikasi masalah, tujuan penelitian, batasan masalah, metode penelitian, dan sistematika penulisan.

### BAB II : LANDASAN TEORI

Pada bab ini akan dijelaskan beberapa teori yang mendasari pemecahan masalah dan teori-teori yang berkaitan langsung dengan pemecahan masalah serta pembahasannya diperoleh dari sumber-sumber referensi dan catatan-catatan lainnya yang dikembangkan.

### BAB III : ANALISI MASALAH DAN PERANCANGAN PROGRAM

Pada bab ini memuat tentang pembahasan masalah yang akan dipecahkan berdasarkan metode *prototype* dan metode pemograman berorientasi objek dengan *tools Unified Modeling Language* (UML) yang meliputi *use case* diagram, *activity*

diagram, *class* diagram dan *sequence* diagram. Serta pada bab ini juga akan membahas mengenai *user interface*.

#### BAB IV : IMPLEMENTASI DAN UJI COBA

Pada bab ini berisi tentang penjelasan dalam pengoperasian program secara bertahap dan pengujian akhir dengan metode *blackbox*, kuisioner, serta tampilan layout aplikasi.

#### BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisikan kesimpulan dan saran-saran dari hasil analisis dan pembahasan yang dapat diajukan sehubungan dengan permasalahan skripsi.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Profil Perusahaan**

##### **2.2.1 Gambaran Umum Perusahaan**



GAMBAR: 2. 1 Logo PT Radya Gita Bahagi (Rolling Glory, 2012)

PT. Radya Gita Bahagi atau dikenal dengan Rolling Glory didirikan awal tahun 2012, oleh empat orang pemuda berumur 20 tahun-an, dengan semangat dalam programming dan seni, mereka memutuskan untuk membentuk *creative digital* mereka sendiri. Pada tanggal 29 Febuari 2012 Rolling Glory *Creative Digital Media Agency* dinyatakan resmi dan mempunyai kantor tetap. Bermodalkan dengan semangat positif dan kreatif, meskipun dengan waktu yang sedikit, Rolling Glory telah mendapat dan menyelesaikan banyak proyek. Rolling Glory telah melayani baik klien dari dalam negeri maupun dari luar negeri.

**Visi :**

Rolling Glory memiliki visi untuk dikenal sebagai *Creative Digital Media Agency* yang dapat menciptakan produk yang baik dan inovatif Rolling Glory juga berharap dapat memberikan kontribusi kepada orang-orang yang membutuhkan *Creative Digital Media Agency* dengan kemampuan dan pengetahuan Rolling Glory sebagai tim *Creative Digital* (Rolling Glory, 2012).

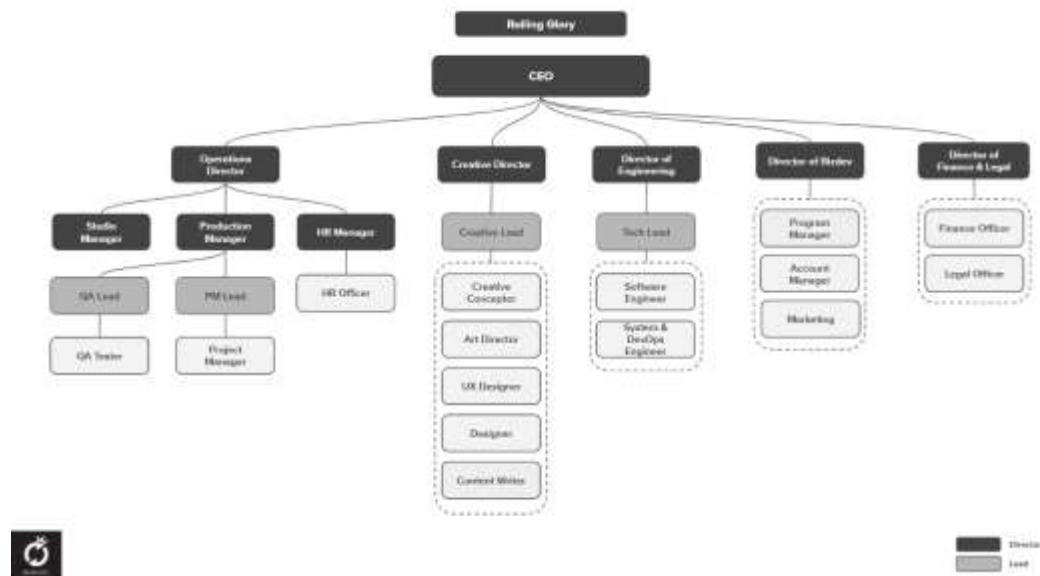
**Misi :**

Rolling Glory memiliki misi untuk menciptakan produk digital yang dapat berintergrasi, dapat menyampaikan pesan dengan baik, dan dapat menyelesaikan permasalahan yang ada, dengan kualitas yang premium diluar ekspektasi user. Rolling Glory memiliki tujuan untuk menciptakan semangat kerja dan kebahagiaan di dalam suasana tempat kerja.

Rolling Glory juga memiliki keinginan untuk belajar dari pengalaman yang ada dan menjadikan pengalaman tersebut sebagai landasan untuk melompat lebih tinggi dari sebelumnya. Rolling Glory ingin memberikan kontribusi yang terbaik dengan memberikan kualitas premium terbaik (Rolling Glory, 2012).

## 2.2.2 Struktur Organisasi Perusahaan

Struktur organisasi PT Radya Gita Bahagi dapat dilihat pada bagan dibawah ini :



GAMBAR: 2. 2 Struktur Organisasi (Rolling Glory, 2012)

## 2.2 Pengertian *Prototype*

Metode *Prototype* merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC.

*Prototype* adalah bagian dari produk yang mengekspresikan logika maupun fisik antarmuka eksternal yang ditampilkan (Janner Simarmata, 2010).

*Prototype* adalah salah satu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai (Deni Darmawan, 2013).

*Prototype* didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkap, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping* (Raymond McLeod, 2013).

*Protoype is a toy implementation of system* (Prototipe adalah sebuah implementasi tiruan dari sebuah sistem) (Thiemann, 2012).

#### Tahapan Metode *Prototype* / *Prototyping*

##### 1. Analisa Kebutuhan

*User* dan pengembang bersama-sama mendefinisikan format software, mengidentifikasi kebutuhan dan sistem yang dibuat.

##### 2. Membangun *Prototype*

Membangun *prototyping* dengan membuat perancangan sementara yang berfokus penyajian kepada user (contoh membuat input dan format output).

##### 3. Evaluasi *Prototype*

Tahap ini dilakukan oleh *user*, apakah *prototyping* yang dibangun, sudah sesuai dengan keinginan dan kebutuhan pelanggan atau belum. Jika tidak sesuai, *prototyping* akan direvisi dengan mengulangi langkah-langkah sebelumnya. Tapi jika sudah sesuai, maka langkah selanjutnya akan dilaksanakan.

##### 4. Pengkodean Sistem

Di tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

#### 5. Menguji Sistem

Setelah sistem sudah menjadi suatu software yang siap pakai, maka software harus di tes dahulu sebelum digunakan. Hal ini bertujuan untuk meminimalisir kesalahan software tersebut. Pengujian dilakukan dengan Black Box, Kuisisioner dan lain-lain.

### 2.3 Pengertian Website

Website atau web merupakan suatu dokumen berupa sekumpulan halaman yang berisi berbagai informasi berbentuk digital. Informasi itu bisa berupa teks, gambar, animasi, video atau gabungan dari semuanya yang disediakan melalui internet dan dapat diakses oleh banyak orang di seluruh dunia selama memiliki koneksi internet. Website dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet (Abdullah, 2015). Website adalah sejumlah halaman web yang memiliki topik saling terkait antara halaman yang satu dengan halaman lain, yang biasanya ditempatkan pada sebuah *server web* yang dapat diakses melalui jaringan internet ataupun jaringan wilayah lokal (LAN) (Susilowati, 2019).

Website dibuat untuk dapat diakses secara luas melalui sebuah aplikasi peramban menggunakan URL (*Uniform Resource Locator*). Dalam suatu halaman

web, biasanya terdapat berbagai macam jenis informasi dalam bentuk teks, video, gambar, suara, dan lain-lain.

## 2.4 Pengertian Aplikasi

Aplikasi yang biasa disebut juga sebagai program aplikasi atau perangkat lunak aplikasi, adalah paket perangkat lunak komputer yang menjalankan fungsi tertentu secara langsung untuk end user atau dalam beberapa kasus untuk aplikasi lain. Aplikasi dapat berdiri sendiri atau bisa dari gabungan program. Program adalah sekumpulan operasi yang menjalankan aplikasi untuk user.

Di antara banyak lainnya, program aplikasi meliputi :

- *Email*
- *Web Browsers*
- *Games*
- *Word Processors*
- *Enterprise Software*
- *Accounting Software*
- *Graphics Software*
- *Media Players*
- *Database Management*

Perangkat lunak aplikasi dan sistem perangkat lunak adalah dua jenis perangkat lunak utama yang tersedia. Sistem perangkat lunak mengelola operasi internal komputer, terutama melalui sistem operasi. Sistem perangkat lunak

mengelola periferal seperti perangkat penyimpanan, printer, dan monitor juga. Sebaliknya, perangkat lunak aplikasi atau program aplikasi memandu komputer untuk menjalankan instruksi yang diberikan oleh user.

Beberapa contoh program aplikasi meliputi:

- *Application suite* : Berisi berbagai aplikasi yang dikemas bersama
- *Enterprise software* : Mengatasi aliran data dan persyaratan proses organisasi, yang mencakup seluruh departemen
- *Information worker software* : Mengizinkan pengguna untuk membuat dan mengelola informasi
- *Content access software* : Digunakan terutama untuk mendapatkan akses ke konten tanpa mengedit
- *Media development software* : Membuat media elektronik dan cetak
- *Educational software* : Berisi konten dan/atau fitur yang ditujukan untuk siswa atau pendidik

*Product engineering software*: Mengembangkan produk perangkat lunak dan perangkat keras.

## **2.5 Pengertian *Computer Based Test***

*Computer Based Test* atau Tes Berbasis Komputer adalah tes dengan sistem pelaksanaan menggunakan komputer sebagai media untuk melakukan tes. *Computer Based Test* merupakan ujian yang dikerjakan di komputer sehingga tidak memerlukan kertas, pena maupun pensil untuk menjawab pertanyaannya.

Saat ini, penggunaan *computer based testing* pun cukup marak digunakan dalam dunia perusahaan. Khususnya adalah pada proses rekrutmen, di mana pelamar diuji dan hasil dari tahapan ujian tersebut dapat dengan cepat diperoleh (Al-Amri, 2011).

Peluang untuk menggantikan ujian berbasis kertas dengan ujian berbasis komputer dengan memperhatikan unsur-unsur teknis seperti keamanan, kemudahan penggunaan dan kemampuan dasar pengguna komputer (Yulianto, Wahyuni and Eka, 2016).

## **2.6 Pengertian Rekrutmen**

Rekrutmen adalah proses mencari dan menarik pelamar yang berkemampuan untuk diseleksi menjadi karyawan sesuai dengan posisi yang dibutuhkan. Adapun pengertian rekrutmen menurut para ahli :

- Menurut Mardianto, pengertian rekrutmen adalah proses untuk mendapatkan calon karyawan yang memiliki kemampuan yang relevan dengan kualifikasi dan kebutuhan perusahaan atau organisasi (Mardianto, 2014).
- Menurut Rivai, pengertian rekrutmen adalah proses untuk memperoleh sumber daya manusia (karyawan) yang memiliki kualitas untuk menduduki jabatan atau pekerjaan didalam perusahaan (Rivai, 2004: 158).
- Menurut Simamora, pengertian rekrutmen adalah serangkaian kegiatan untuk mencari dan menarik pelamar kerja dengan motivasi, kemampuan,



keahlian serta pengetahuan yang dibutuhkan dengan tujuan untuk menutupi kekurangan yang teridentifikasi di dalam rencana kepegawaiaan (Simamora, 2015).

- Menurut Robert L.Mathis, rekrutmen adalah proses menghasilkan suatu kelompok pelamar yang memenuhi syarat untuk pekerjaan (L. Mathis, Robert & H. Jackson, 2011).

Tujuan adanya proses rekrutmen ini diharapkan perusahaan dapat memenuhi harapan perusahaan untuk mendapatkan Sumber Daya Manusia yang berkualitas yang sesuai dengan harapan.

## 2.7 *Web Browser*

Dikutip dari Techopedia *Web Browser* adalah program perangkat lunak yang memungkinkan pengguna untuk mencari, mengakses, dan menampilkan halaman web. Dalam penggunaan umum, *browser web* biasanya disingkat menjadi “*browser*”. *Web browser* menerjemahkan halaman situs web yang dikirim menggunakan HTTP/HTTPS (*Hypertext Transfer Protocol* ) ataupun FTP (*File Transfer Protocol*) menjadi konten yang dapat dibaca manusia. Terdapat beberapa fungsi dari *web browser*:

- Untuk menampilkan halaman atau situs-situs website
- Menerjemahkan *script* menjadi tampilan yang menarik
- Menjamin keamanan situs *website* yang dikunjungi
- Mendukung penyimpanan data halaman secara *offline*.

- Memberikan dukungan untuk akses ke mesin-mesin pencari seperti google, dan sebagainya.

## 2.8 *Flowchart*

*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

*Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

Untuk pengolahan data dengan komputer, dapat dirangkum urutan dasar untuk pemecahan suatu masalah, yaitu;

- *START*: Berisi intruksi untuk persiapan peralatan yang di perlukan sebelum menangani pemecahan masalah
- *READ*: Berisi instruksi untuk membaca data dari suatu peralatan input.
- *PROCESS*: Berisi kegiatan yang berkaitan dengan pemecahan persoalan sesuai dengan data yang dibaca.
- *WRITE*: Berisi instruksi untuk merekam hasil kegiatan ke perlatan output.
- *END*: Mengakhiri kegiatan pengolahan

*Flowchart* terbagi atas 5 jenis, yaitu :

- Bagan alir sistem (*systems flowchart*).
- Bagan alir dokumen (*document flowchart*).
- Bagan alir skematik (*schematic flowchart*).

- Bagan alir program (*program flowchart*).
- Bagan alir proses (*process flowchart*).

## 2.9 Usability

Menurut ISO 9241 – 11 (1998) , *Usability* adalah sejauh mana sebuah produk dapat digunakan oleh pengguna tertentu untuk mencapai tujuan tertentu dengan efektif, efisiensi dan pengguna menjadi puas dalam konteks penggunaan. Secara umum pengertian *Usability* adalah atribut dari kualitas yang digunakan untuk mengevaluasi bagaimana mudahnya sebuah antar muka digunakan.

Menurut Nielsen, *Usability* didefinisikan dalam 5 komponen kualitas yaitu:

- a. *Learnability*, yaitu ketika pengguna baru mempelajari/berinteraksi dengan design seberapa mudah pengguna menyelesaikan tugas-tugas dasar
- b. *Efficiency*, setelah pengguna mempelajari *design*, seberapa cepat pengguna melakukan tugas
- c. *Memorability*, seberapa mudah pengguna mahir kembali dalam menggunakan design ketika pengguna tidak menggunakan *design* dalam rentang waktu yang lama.
- d. *Errors*, yaitu seberapa banyak kesalahan, atau seberapa parah kesalahan yang dilakukan oleh pengguna dan seberapa mudah pengguna memperbaiki kesalahan tersebut.
- e. *Satisfaction*, yaitu seberapa menyenangkan pengguna menggunakan *design* tersebut.

Ada alasan mengapa *usability* itu penting diantaranya

- Memudahkan Pengguna

Sebuah aplikasi harus memiliki *usability* yang baik. Dengan memiliki *usability* yang baik akan memudahkan pengguna dalam mengoperasikan aplikasi. Karena pada prinsipnya aplikasi dibuat untuk memudahkan penggunanya.

- Mudah Dipelajari

Selain mudah digunakan, *usability* juga memastikan aplikasi mudah untuk dipelajari. Sebisa mungkin pengguna langsung paham dan mudah memahami penggunaan aplikasi saat pertama kali mencoba. Atau harus bisa memberikan pandangan pertama yang menakjubkan.

- Berjalan Sesuai Fungsinya

*Usability* memastikan fungsi aplikasi berjalan sesuai dengan tujuannya. Jika sebuah aplikasi berfungsi untuk transaksi pembayaran, maka aplikasi tersebut harus bisa digunakan sesuai dengan fungsinya.

- Mempengaruhi kepada faktor kesuksesan

*Usability* juga berdampak pada faktor kesuksesan dalam sebuah aplikasi jika aplikasi tersebut susah diakses oleh pengguna maka pengguna akan meninggalkan aplikasi tersebut dan berpindah aplikasi yang mudah digunakan.

- Menarik minat pengguna

Aplikasi harus mempunyai *usability* yang baik agar menarik minat pengguna untuk menggunakan aplikasi tersebut sampai seterusnya.

## 2.10 UML

*Unified Modelling Language* (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual (Braun, et. al. 2001). Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, Jeffrey L., 2014). UML (*Unified Modeling Language*) adalah perangkat lunak yang berparadigma “berorientasi objek”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010). Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program.

UML juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML. UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada Januari 1997.

### 1. *Use Case* Diagram

*Use case* diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Diagram ini menggambarkan kumpulan *use case*, aktor, dan hubungan mereka. *Use case* adalah

hubungan antara fungsionalitas sistem dengan aktor internal/eksternal dari sistem.

TABLE: 2. 1 Simbol *Use Case* Diagram (A.S Rosa, 2014)










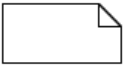
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .


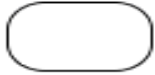



TABLE: 2. 2 Simbol *Use Case* Diagram (A.S Rosa, 2014)

NO	GAMBAR	NAMA	KETERANGAN
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

## 2. Activity Diagram

*Activity Diagram*, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi. Pada pemodelan UML, *activity diagram* dapat digunakan untuk menjelaskan bisnis dan alur kerja operasional secara *step-by-step* dari komponen suatu sistem.

TABLE: 2. 2 Simbol *Activity Diagram* (A.S Rosa, 2014)

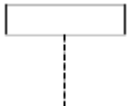


NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran



### 3. *Sequence* Diagram

*Sequence* diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. Diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.

TABLE: 2. 3 Simbol *Sequence* Diagram (A.S Rosa, 2014)

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

#### 4. Class Diagram

*Class* diagram merupakan diagram yang selalu ada di pemodelan sistem berorientasi objek. Diagram ini terdiri dari *class*, *interface*, *association*, dan *collaboration*. Diagram ini menggambarkan objek - objek yang ada di sistem.

TABLE: 2. 4 Simbol *Class* Diagram (A.S Rosa, 2014)



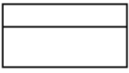

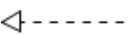


NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

TABLE: 2. 5 Simbol *Class Diagram* (A.S Rosa, 2014)

NO	GAMBAR	NAMA	KETERANGAN
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

### 2.11 *Blackbox Testing*

*Blackbox testing* adalah suatu metode pengujian dimana tester hanya fokus pada apa yang seharusnya dilakukan oleh system (Rex, 2009). Pengujian *Blackbox* terutama berfokus pada input dan output aplikasi perangkat lunak dan sepenuhnya didasarkan pada persyaratan dan spesifikasi perangkat lunak. Ini juga dikenal sebagai Pengujian Perilaku. *Blackbox Testing* ini lebih menguji ke Tampilan Luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *Customer*.

Keuntungan dari *Blackbox Testing*:

- Penguji tidak perlu memiliki pengetahuan tentang bahasa pemograman tertentu
- Pengujian yang dilakukan berdasarkan sudut pandang user agar dapat mengungkapkan inkosistensi atau ambiguitas dalam spesifikasi.

- Programmer dan tester memiliki ketergantungan satu sama lain

Kekurangan *Blackbox Testing*:

- Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer
- Beberapa bagian back end tidak diuji sama sekali.

## **BAB III**

### **ANALISA KEBUTUHAN DAN PERANCANGAN SISTEM**

Dalam penelitian ini digunakan metode pengembangan SDLC (*System Development Life Cycle*) dengan Model *Prototype*. Ada tujuh tahap dalam pengembangannya namun dalam penelitian ini dibatasi sampai dengan tahap lima, kelima tahap tersebut yaitu, analisis kebutuhan, membangun prototyping, evaluasi prototyping, mengkodekan sistem, dan pengujian sistem.

#### **3.1 Analisis kebutuhan**

Pada tahap ini dilakukan proses pengumpulan informasi berupa data-data yang berkaitan dengan penelitian, dan melakukan analisis permasalahan yang dihadapi.

##### **3.1.1 Studi Litelatur**

Studi Literatur merupakan teknik yang digunakan untuk memperoleh informasi atau data dengan mempelajari buku-buku serta jurnal yang berkaitan dengan perancangan aplikasi rekrutmen karyawan. Berikut adalah tabel 3.1 yang menjadi referensi penelitian

TABEL: 3. 1 Tabel Studi Literatur

No	Penulis	Judul	Pembahasan
1	Ferry Putrawansyah	Aplikasi <i>Computer Assisted Test</i> (CAT) Pada Penerimaan Mahasiswa Baru Sekolah Tinggi Teknologi Pagar Alam (STTP)	Aplikasi Multimedia <i>Computer Assisted Test</i> (CAT) pada test penerimaan mahasiswa baru pada Sekolah Tinggi Teknologi Pagaralam untuk menguji kevalidan dan uji one to one untuk mengetahui keraktisan. Berdasarkan hasil <i>Expert Review</i> produk ini di nyatakan valid dilihat dari nilai rata-hasil penilaian ahli media, ahli desain dan ahli evaluasi dengan rata-rata skor 3,9 sehingga kategori produk ini yaitu valid. Selanjutnya hasil uji kepraktisan produk melalui one to one dinilai oleh 3 orang responden dengan hasil rata-rata nilai 3,9 dengan kriteria praktis. (Putrawansyah, 2017)

TABEL: 3. 1 Tabel Studi Literatur

No	Penulis	Judul	Pembahasan
2	M. Fadli	Rancang Bangun Aplikasi Rekrutmen Karyawan Pada PT. Kasa Husada Wira Jatim Berbasis Website	<p>Permasalahan dalam proses penilaian kinerja di perusahaan adalah lamaran yang masuk masih menggunakan data fisik. Bagian HRD harus memilah satu persatu data pelamar yang masuk apakah memenuhi kriteria perusahaan, proses ini membutuhkan waktu kurang lebih 14 hari. Untuk mengatasi masalah di atas, maka diberikan solusi untuk dibuatkan rancang bangun aplikasi rekrutmen karyawan berbasis website. Berdasarkan hasil uji coba aplikasi ini mempermudah bagian HRD dalam melakukan proses pembuatan lowongan, perhitungan nilai seleksi karyawan dan pembuatan laporan dan proses yang semula 14 hari bisa lebih singkat menjadi kurang lebih 5 hari. (Fadli, 2020)</p>

TABEL: 3. 1 Tabel Studi Literatur

No	Penulis	Judul	Pembahasan
3	Iswanto, Abdul Azis Muslim	Perancangan Dan Implementasi Rekrutmen Karyawan Menggunakan Metode Saw Berbasis Web (Studi Kasus PT. Ateja)	Departemen HRD di PT Ateja masih menggunakan sistem manual sehingga memperlambat kerja perusahaan dalam tahapan rekrutmen, data tentang rekrutmen belum terdokumentasi dan tersimpan dalam database sehingga mudah hilang. Dengan menggunakan komputer proses pengelolaan administrasi akan lebih cepat, informasi lebih akurat, efektivitas dan efisiensi dapat ditingkatkan serta human error dapat diminimalisasi. (Iswanto, 2018)



TABEL: 3. 1 Tabel Studi Literatur

No	Penulis	Judul	Pembahasan
4	Liza Trisnawati, Evi Syafriзал	Rancangan Sistem Rekrutmen Karyawan Berbasis Web Pada PT. Fast Food Indonesia Region Pekanbaru	<p>Pada sistem yang berjalan saat ini, staff HRD membutuhkan waktu yang lama dalam mengumpulkan informasi-informasi tentang pelamar yang masuk, sehingga proses penyeleksian pun tidak benar-benar memiliki objektivitas yang tinggi karena waktu dan tenaga sudah terbuang pada saat pengumpulan informasi. Dengan menggunakan sistem konvensional, memakan waktu, biaya dan menghasilkan tumpukan berkas yang banyak. Untuk mengatasi permasalahan tersebut, maka dibutuhkan sistem rekrutmen dengan menggunakan teknologi komputer, yaitu berbasis web. Hal tersebut menjadi dasar untuk memudahkan calon pelamar dalam melakukan pengiriman berkas lamaran dan mengikuti tes penerimaan masuk ke perusahaan secara online. (Trisnawati and Syafriзал, 2016)</p>

TABEL: 3. 1 Tabel Studi Literatur

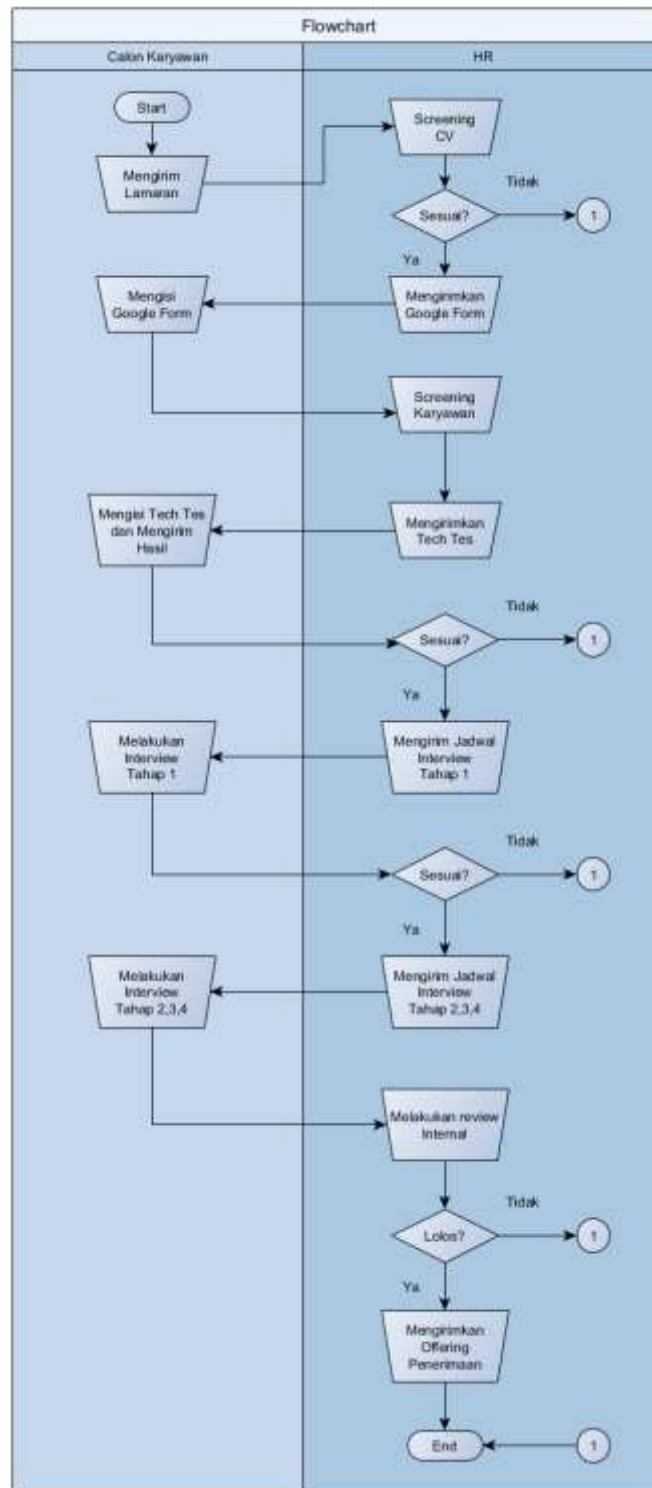
No	Penulis	Judul	Pembahasan
5	Jauhari Maulani, S.Kom, M.Kom	Rekrutmen Karyawan CV. Annisa Menggunakan Aplikasi Test Psikotes Dan Kepribadian Berdasarkan Delphi	Proses tes psikotes dan kepribadian dilakukan oleh perusahaan CV. Annisa selama ini masih secara manual, tes psikotes bisa dilakukan sewaktu – waktu sesuai dengan kebutuhan dari perusahaan tersebut. Hambatan lainnya ialah apabila pemeriksa sedang berhalangan, maka tes psikotes tidak dapat diproses dalam waktu yang sulit untuk ditentukan, hal ini membuat ketergantungan yang sangat besar pada staf penilai psikotes. Untuk itu diusulkan membuat aplikasi test psikotes dan test kepribadian menggunakan program delphi guna mempercepat proses perekrutan calon karyawan pada CV. Annisa agar pengerjaan dan hasilnya bisa lebih efisien serta penyimpanan data juga tersimpan dengan baik. (Maulani, Kom and Kom, 2016)

Perbedaan penelitian ini dengan penelitian sejenis :

1. Studi kasus pada perusahaan yang berbeda
2. *Web* aplikasi ini dibuat menggunakan framework *Vuejs*
3. Pada penelitian ini aplikasi dibuat untuk proses awal seleksi tes karyawan di PT Radya Gita Bahagi.
4. Pada pembuatan aplikasi ini penulis memperhatikan faktor *usability end user*.

### **3.1.2 Analisis Sistem Yang Berjalan Saat Ini**

Analisis sistem yang sedang berjalan saat ini akan menggambarkan prosedur rekrutmen pada PT Radya Gita Bahagi. Proses rekrutmen yang digunakan saat ini terdiri dari beberapa tahap yaitu *screening CV*, pengisian google form, *screening* karyawan, *technical test*, dan interview yang terbagi menjadi 4 sesi. Proses ini akan digambarkan ke dalam diagram flowchart sebagai berikut :



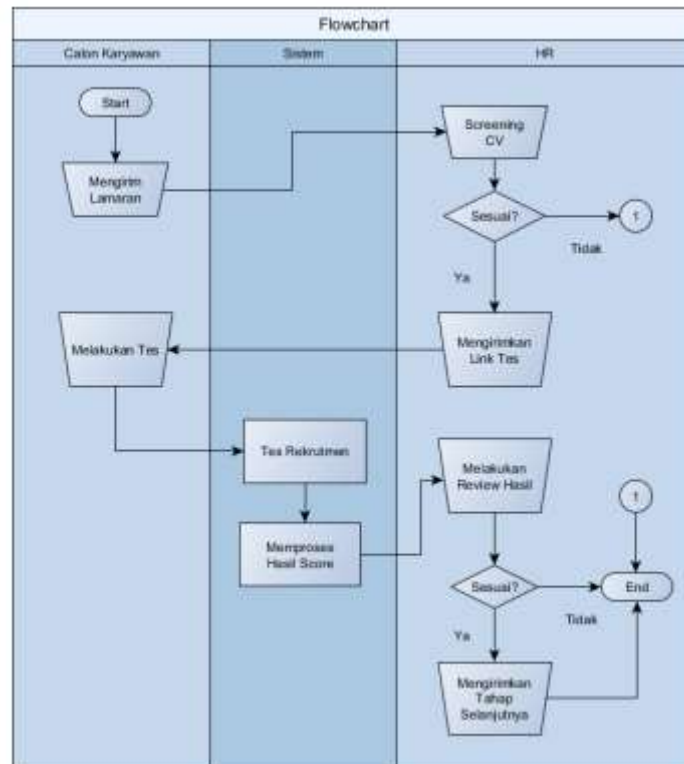
GAMBAR: 3. 1 *Flowchart* Proses Rekrutmen PT Radya Gita Bahagi

### **3.1.3 Analisis Masalah**

Berdasarkan analisis sistem yang sedang berjalan saat ini di PT Radya Gita Bahagi proses rekrutmen yang saat ini sangat panjang, sehingga membuat perusahaan lama dalam mendapatkan karyawan yang dibutuhkan. Berdasarkan hal tersebut maka dibutuhkan sebuah sistem aplikasi yang dapat membantu perusahaan dalam proses seleksi karyawan pada perusahaan.

### **3.1.4 Sistem Yang Diusulkan**

Berdasarkan analisis masalah tersebut maka penulis membuat sistem aplikasi berbasis Computer Based Testing (CBT) yang dapat mempermudah perusahaan dalam proses seleksi karyawan dengan memperhatikan faktor usability end user dalam pembuatannya. Aplikasi ini digunakan pada awal proses rekrutmen untuk mendiagnosa apakah calon karyawan berpotensi dan sudah sesuai dengan yang dibutuhkan perusahaan. Berikut gambaran proses yang diusulkan:



GAMBAR: 3. 2 Flowchart Proses Yang Diusulkan

### 3.1.5 Analisis Kebutuhan Sistem

Sistem yang akan dibangun adalah berupa web aplikasi menggunakan bahasa pemrograman php dan javascript, sehingga diperlukan beberapa kebutuhan untuk melakukan pembangunan web aplikasi. Berikut adalah kebutuhan yang dibutuhkan :

- *Hardware* (Perangkat Keras) untuk pembuatan web aplikasi
  1. Komputer/ Laptop dengan spesifikasi :
    - Processor Intel Core i5
    - Ram 8Gb
  2. *Keyboard*

3. *Mouse*
4. *Monitor*
- *Software* (Perangkat Lunak) untuk pembuatan web aplikasi
  1. Xampp versi 3.2.4
  2. PHP versi 7.3.25
  3. Node.js
  4. Visual Studio Code
  5. Browser Google Chrome, Mozilla Firefox, Internet Explorer

### **3.1.6 Planning**

Dalam tahapan penelitian ini memfokuskan pada penjadwalan pengerjaan penelitian. Pada penelitian ini terdapat beberapa proses yang harus dilakukan dari tahap analisis kebutuhan sampai pengujian sistem, maka diperlukan penjadwalan yang tepat agar penelitian ini dapat selesai tepat waktu, berikut penjadwalan penelitian berdasarkan aktifitas yang dilakukan dalam skala waktu per-minggu.

TABEL: 3. 2 Tabel *Planning*

No	Nama Kegiatan	November				Desember				Januari	
		1	2	3	4	1	2	3	4	1	2
1	Analisis Kebutuhan										
	a. Metode Studi Literatur										
	b. Analisis Masalah										
	c. Analisis Kebutuhan Sistem										
	d. Penjadwalan										
5	Membangun Prototype dan Evaluasi Prototype										
6	Pengkodean Sistem										
7	Pengujian Sistem										

### 3.2 Membangun Prototype

Setelah melakukan analisis kebutuhan sistem, tahap selanjutnya adalah tahap prototype. Dimana pada tahap ini membuat perancangan sementara yang berfokus pada penyajian kepada perusahaan.

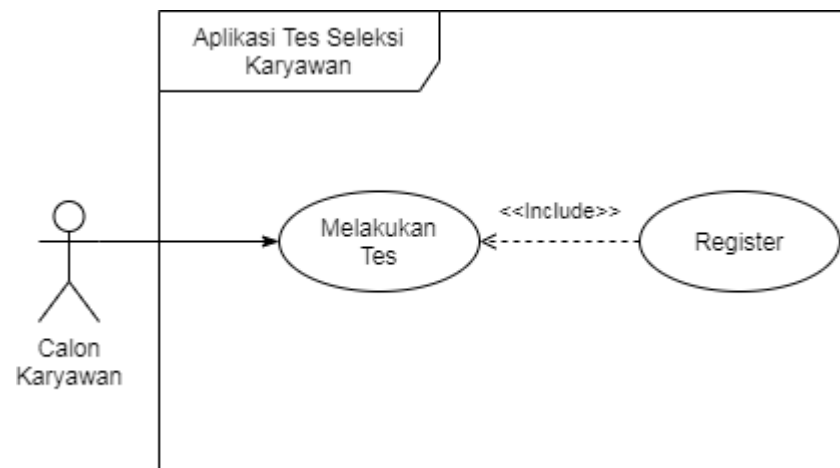
#### 3.2.1 Use Case Diagram

*Use Case* diagram menggambarkan proses dari sistem, hubungan antara *use case* dan aktor berdasarkan kebutuhan sistem dan menggambarkan fungsionalitas yang diharapkan dari sebuah sistem.

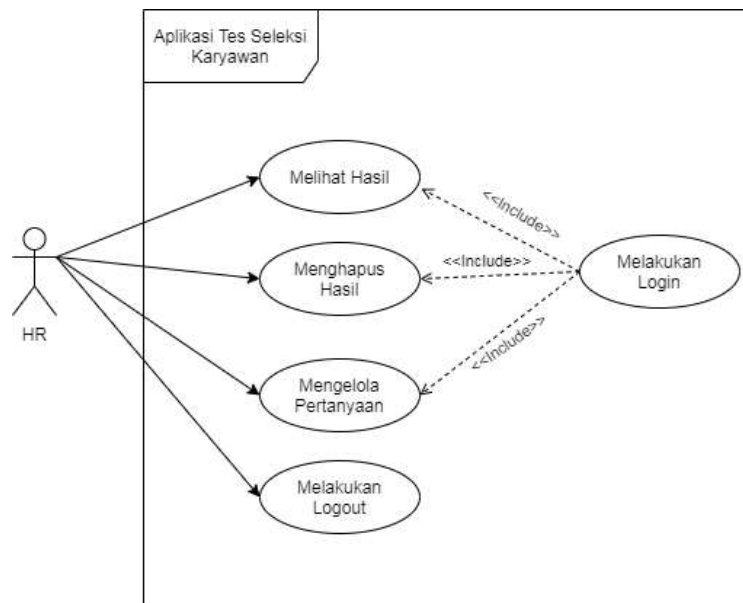


TABEL: 3. 3 Tabel Deskripsi Aktor

No	Aktor	Deskripsi
1	Calon Karyawan	Seseorang yang dapat mengakses halaman <i>frontend</i> aplikasi.
2	HR	Seseorang yang dapat mengakses dan mengelola bagian <i>backoffice</i> aplikasi.



GAMBAR: 3. 3 Use Case Calon Karyawan



GAMBAR: 3. 4 Use Case Admin

TABEL: 3. 4 Tabel Skenario Use Case Melakukan Register

Skenario Use Case	
Nama Use Case	Melakukan Register
Aktor	Calon Karyawan
Skenario	1) Memasukkan Nama Lengkap, Nomor Telepon dan Email 2) Menekan Tombol Mulai

TABEL: 3. 5 Tabel Skenario Use Case Mengisi Test

Skenario Use Case	
Nama Use Case	Mengisi Test
Aktor	Calon Karyawan
Skenario	1) Menekan Tombol Mulai 2) Menekan Tombol Option Jawaban 3) Menekan Tombol Selesai

TABEL: 3. 6 Tabel Skenario *Use Case* Melakukan Login

Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Melakukan Login
Aktor	Admin
Skenario	1) Memasukkan <i>Username</i> dan <i>Password</i> 2) Menekan Tombol <i>Login</i>

TABEL: 3. 7 Tabel Skenario *Use Case* Melihat Hasil

Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Melihat Hasil
Aktor	Admin
Skenario	1) Menekan Menu <i>User Quiz</i> 2) Menekan Tombol <i>Action Show</i>

TABEL: 3. 8 Tabel Skenario *Use Case* Menghapus Hasil

Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Menghapus Hasil
Aktor	Admin
Skenario	1) Menekan Menu <i>User Quiz</i> 2) Memilih session dengan mencentang <i>checkbox</i> 3) Menekan Tombol Hapus 4) Menekan Tombol pilihan 'Ya'

TABEL: 3. 9 Tabel Skenario *Use Case* Menambah Pertanyaan

Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Menambah Pertanyaan
Aktor	Admin
Skenario	1) Menekan Menu <i>Upload Quiz</i> 2) Menekan Tombol <i>New</i> 3) Mengisi form lalu menekan tombol <i>Submit</i>

TABEL: 3. 10 Tabel Skenario *Use Case* Mengedit Pertanyaan

Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Mengedit Pertanyaan
Aktor	Admin
Skenario	1) Menekan Menu <i>Upload Quiz</i> 2) Menekan Action Edit 3) Mengedit data lalu menekan tombol <i>Submit</i>

TABEL: 3. 11 Tabel Skenario *Use Case* Menghapus Pertanyaan

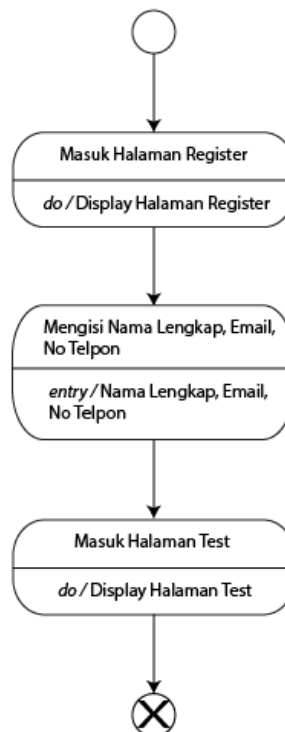
Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Menghapus Pertanyaan
Aktor	Admin
Skenario	1) Menekan Menu <i>Upload Quiz</i> 2) Memilih pertanyaan dengan mencentang <i>checkbox</i> 3) Menekan Tombol Hapus 4) Menekan Tombol pilihan 'Ya'

TABEL: 3. 12 Tabel Skenario *Use Case* Melakukan *Logout*

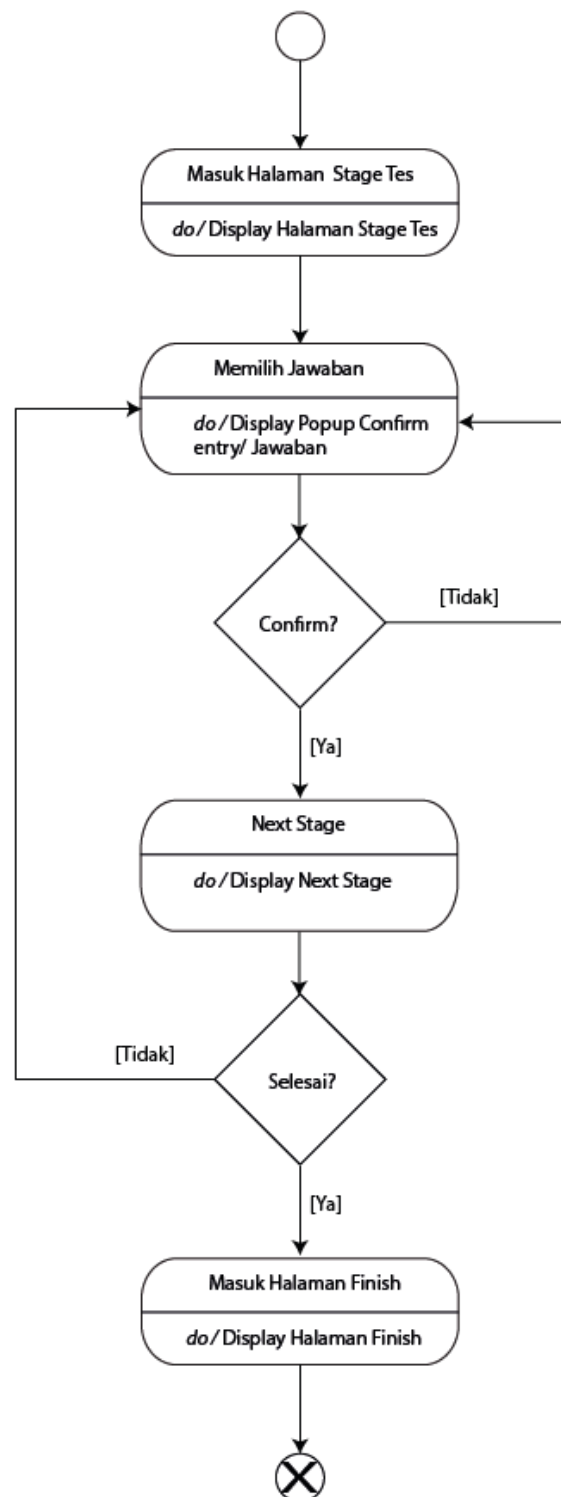
Skenario <i>Use Case</i>	
Nama <i>Use Case</i>	Melakukan <i>Logout</i>
Aktor	Admin
Skenario	1) Menekan Menu <i>User</i> 2) Menekan Tombol <i>Logout</i>

### 3.2.2 Activity Diagram

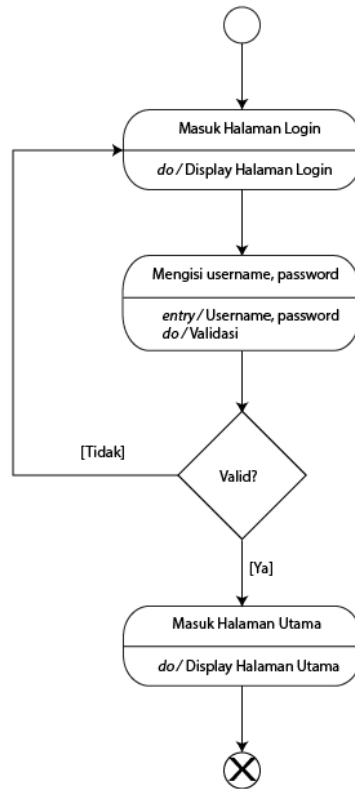
*Activity* diagram adalah diagram yang menunjukkan aktifitas dari setiap fungsi yang ada, biasanya menggambarkan *workflow* atas proses bisnis, bisa juga menggambarkan aktifitas menu yang ada pada aplikasi.



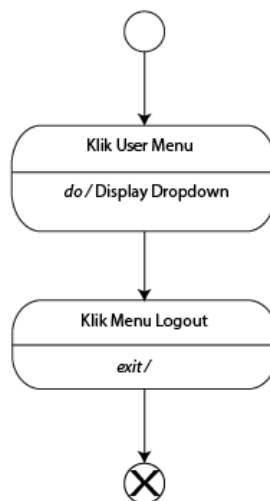
GAMBAR: 3. 5 Activity Diagram Register Calon Karyawan



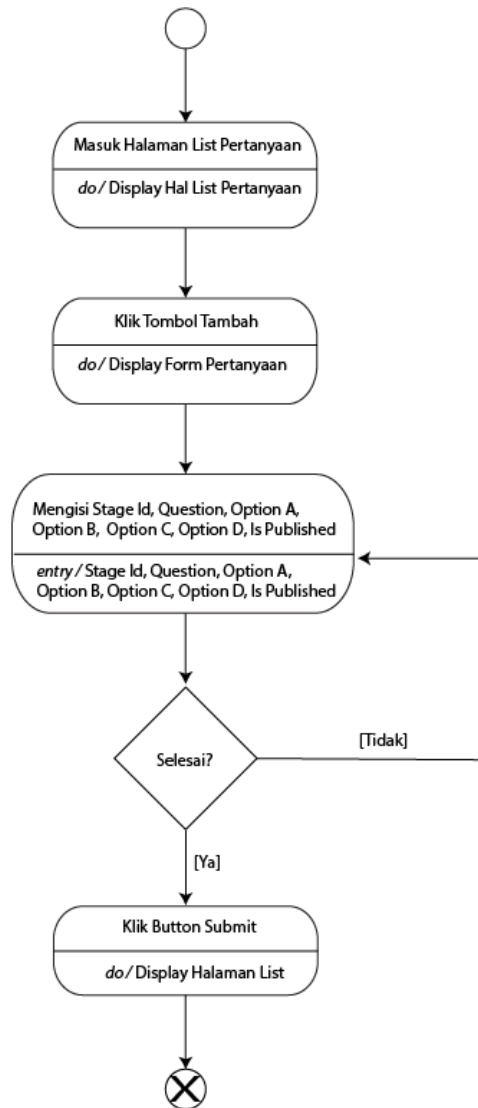
GAMBAR: 3. 6 Activity Diagram Calon Karyawan Mengisi Test



GAMBAR: 3. 7 Activity Diagram Admin Login

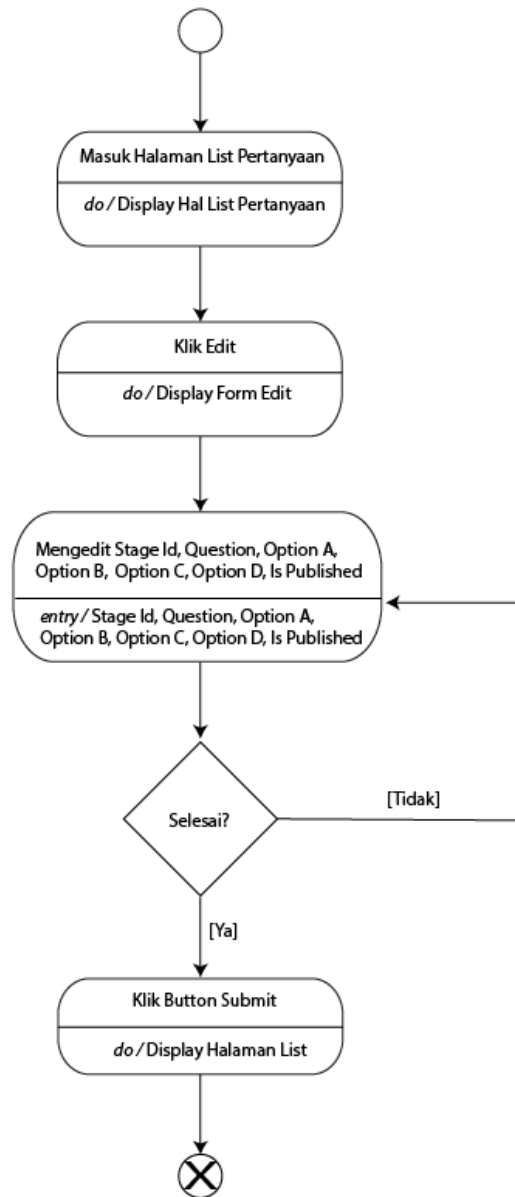


GAMBAR: 3. 8 Activity Diagram Admin Logout

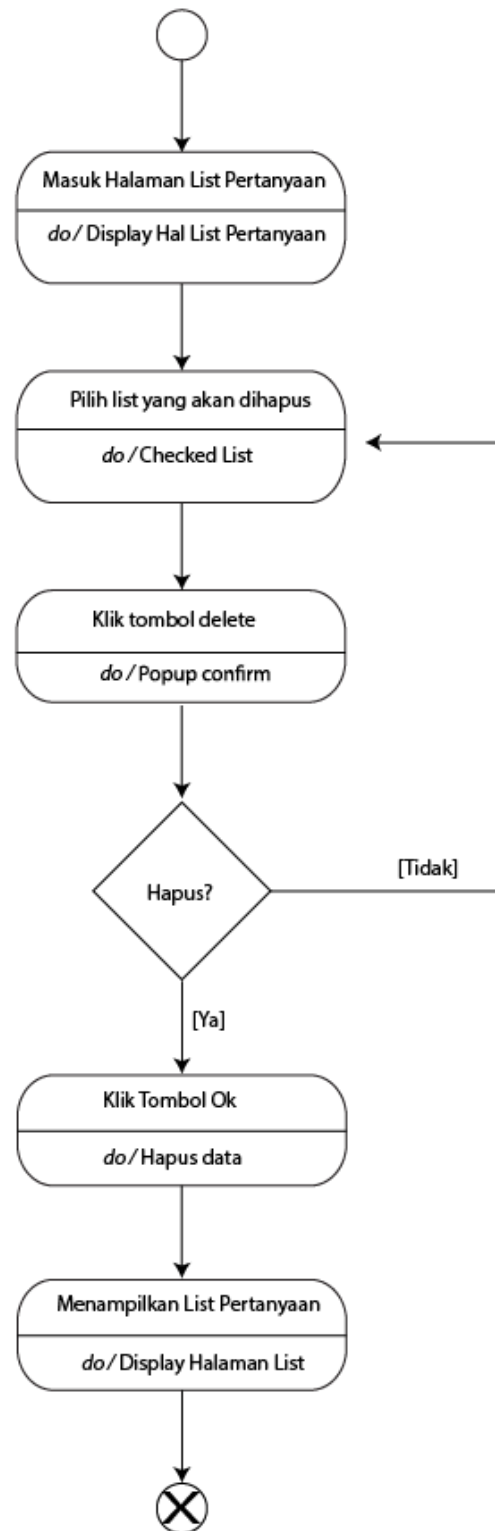


GAMBAR: 3. 9 Activity Diagram Admin Menambah Pertanyaan

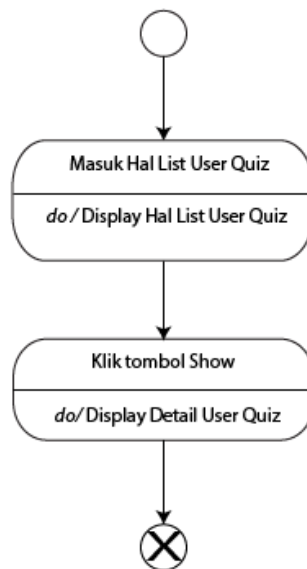




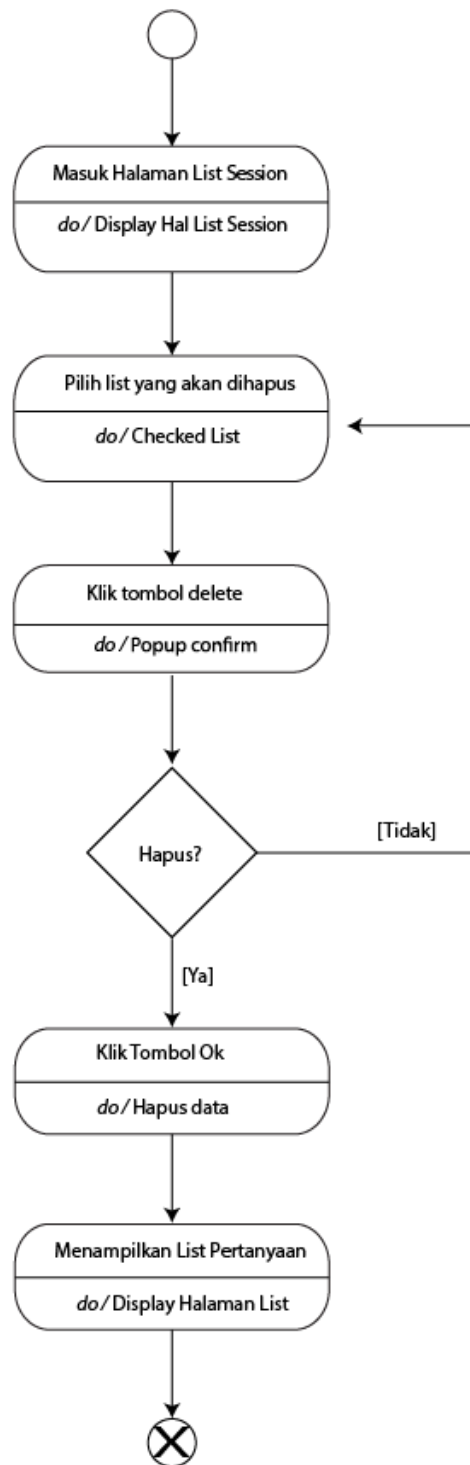
GAMBAR: 3. 10 *Activity* Diagram Admin Mengedit Pertanyaan



GAMBAR: 3. 11 *Activity* Diagram Admin Menghapus Pertanyaan



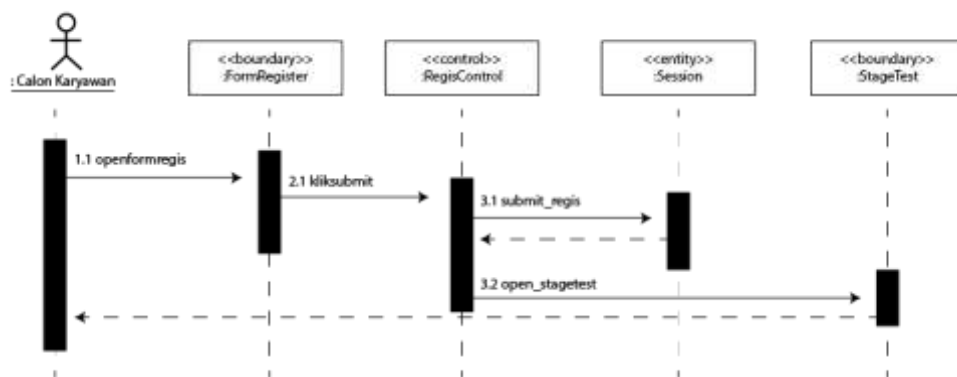
GAMBAR: 3. 12 *Activity* Diagram Admin Melihat Hasil Test



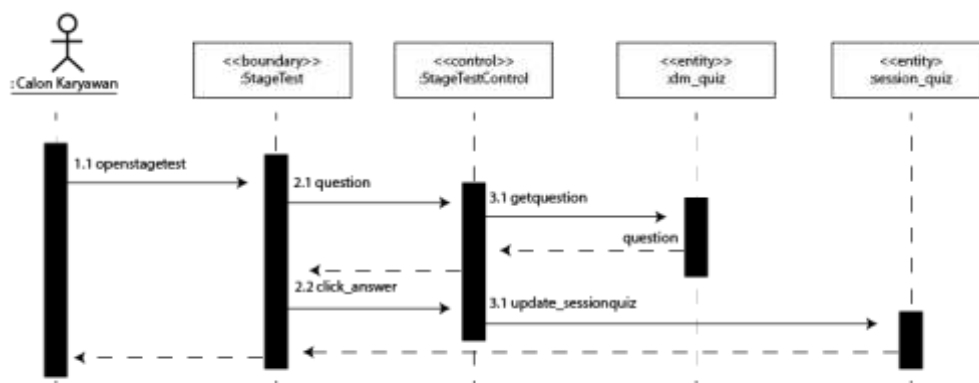
GAMBAR: 3. 13 Activity Diagram Admin Menghapus Hasil Test

### 3.2.3 Sequence Diagram

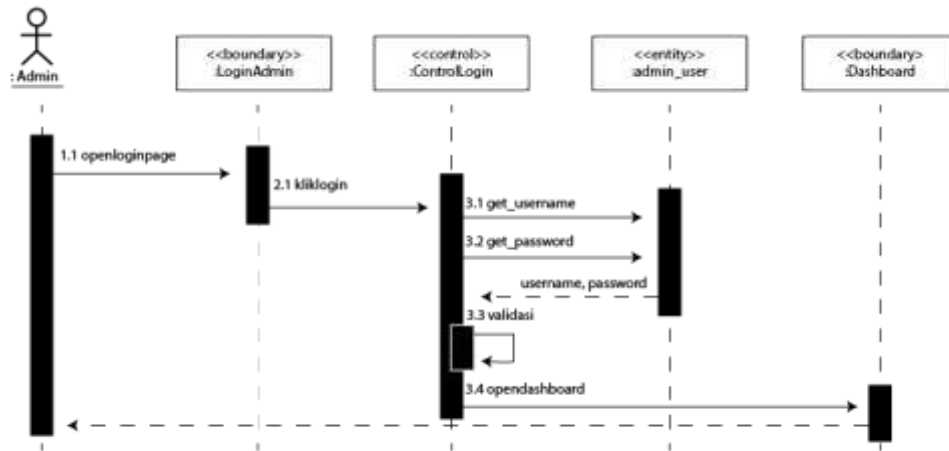
Diagram *sequence* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dengan message yang dikirimkan dan diterima antar objek (A.S Rosa, 2014).



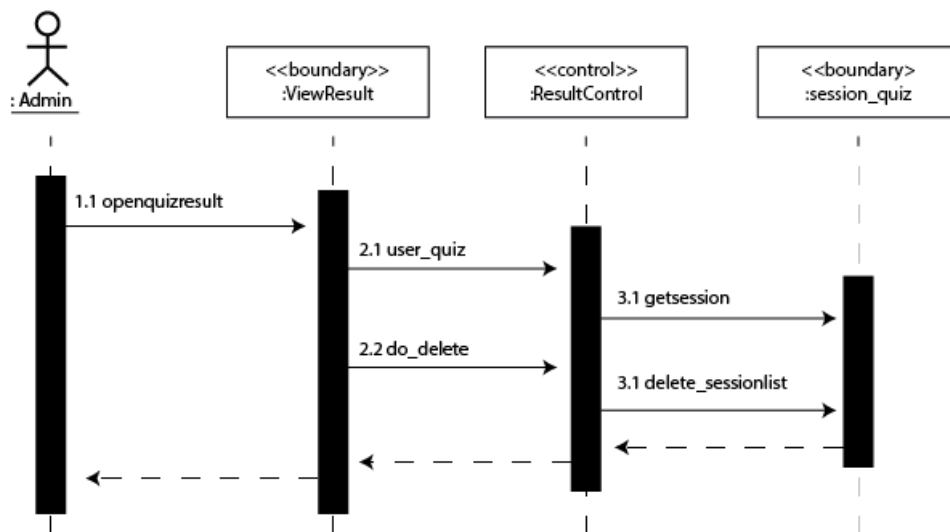
GAMBAR: 3. 14 Sequence Diagram Calon Karyawan Register



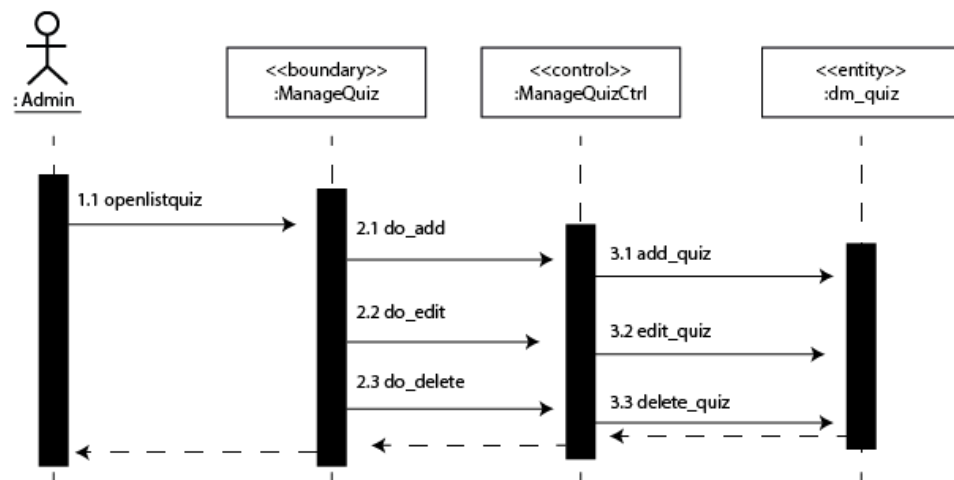
GAMBAR: 3. 15 Sequence Diagram Calon Karyawan Mengisi Test



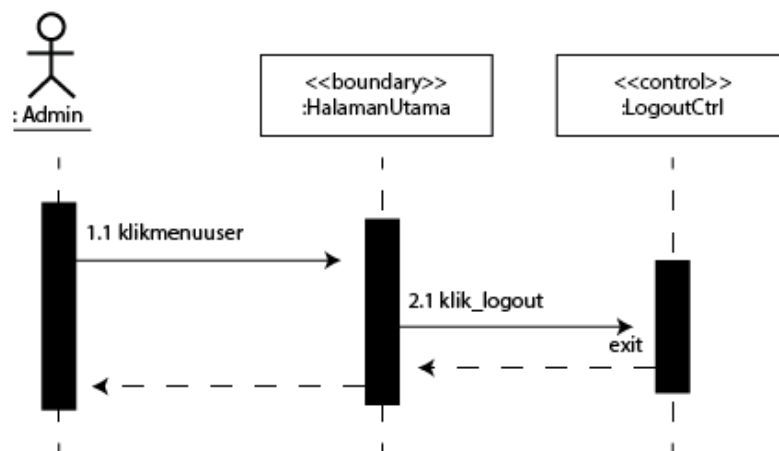
GAMBAR: 3. 16 *Sequence* Diagram Admin Login



GAMBAR: 3. 17 *Sequence* Diagram Admin Mengelola Hasil



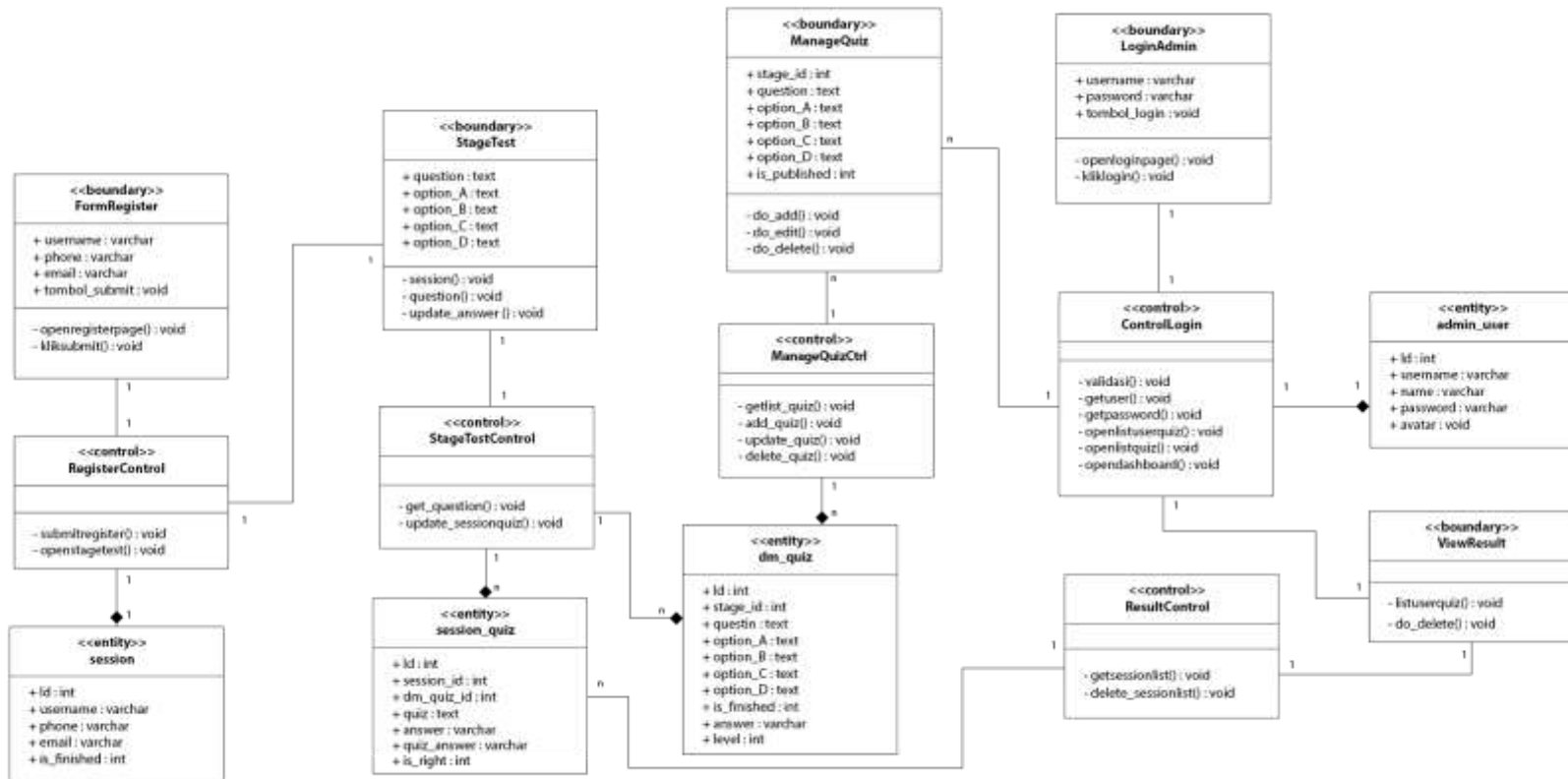
GAMBAR: 3. 18 *Sequence Diagram Admin Mengelola Pertanyaan*



GAMBAR: 3. 19 *Sequence Diagram Admin Logout*

### 3.2.4 Class Diagram

*Class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi.

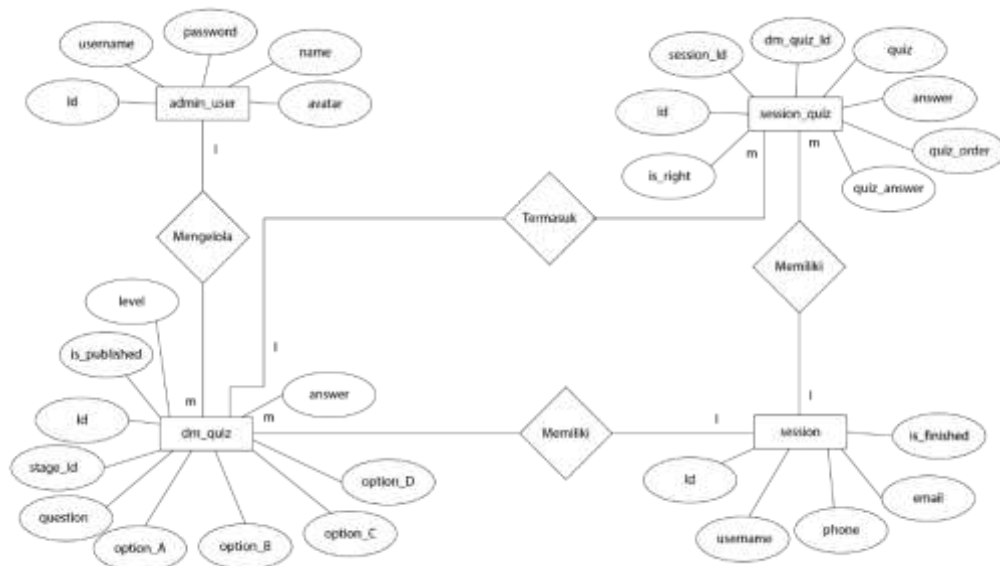


GAMBAR: 3. 20 Class Diagram Aplikasi Rekrutment



### 3.2.5 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah salah satu model yang digunakan untuk mendesign database dengan tujuan menggambarkan data yang berelasi pada sebuah database.



GAMBAR: 3. 21 Entity Relationship Diagram Aplikasi Rekrutment

Terdapat empat tabel yang ada pada aplikasi rekrutment yaitu tabel admin\_user, dm\_quiz, session dan session\_quiz.

TABEL: 3. 13 Tabel Data Admin\_User

Nama	Type Data
Id	Int
Username	Varchar(190)
Password	Varchar(60)
Name	Varchar(255)
Avatar	Varchar(255)

TABEL: 3. 14 Tabel Data Dm\_Quiz

<b>Nama</b>	<b>Type Data</b>
Id	Int
Stage_id	Int
Question	Text
Option_a	Text
Option_b	Text
Option_c	Text
Option_d	Text
Is_published	Int
answer	Varchar(1)
level	int

TABEL: 3. 15 Tabel Data Session

<b>Nama</b>	<b>Type Data</b>
Id	Int
Username	Varchar(255)
Phone	Varchar(255)
Email	Varchar(255)
Is_finished	Int

TABEL: 3. 16 Tabel Data Session\_Quiz

<b>Nama</b>	<b>Type Data</b>
Id	Int
Session_Id	Int
Dm_quiz_id	Int
Quiz	Text
Answer	Varchar(1)
Quiz_order	Int(11)
Quiz_answer	Varchar(1)
Is_right	int

### **3.2.6 Perancangan Antarmuka**

Perancangan antarmuka bertujuan sebagai acuan pengemban dalam membangun web aplikasi rekrutmen. Perancangan antarmuka aplikasi akan terdiri dari halaman untuk end user saja yang terdiri dari desain versi desktop dan versi mobile.

### 3.2.6.1 Antarmuka versi desktop



The screenshot shows a desktop registration form. At the top left is the text "LOGO". The main heading is "Data Diri". Below the heading are three input fields labeled "Nama Lengkap", "No. Telp", and "Email". At the bottom of the form is a "Submit" button. The footer text is "Rolling Gary 2020".

GAMBAR: 3. 22 Antarmuka halaman register versi desktop



The screenshot shows a desktop landing page. At the top left is the text "LOGO". The main heading is "Personal Test". Below the heading is a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco". At the bottom of the form is a "Mulai" button. The footer text is "Rolling Gary 2020".

GAMBAR: 3. 23 Antarmuka halaman pembuka versi desktop



GAMBAR: 3. 24 Antarmuka halaman test versi desktop



GAMBAR: 3. 25 Antarmuka popup saat memilih jawaban versi desktop

### 3.2.6.2 Antarmuka versi mobile



LOGO

Data Diri

Nama Lengkap

No Telepon

Email

Submit

Daftar Sekarang

The image shows a mobile registration form. At the top, there is a 'LOGO' label. Below it is a rounded rectangular container titled 'Data Diri'. Inside this container, there are three input fields labeled 'Nama Lengkap', 'No Telepon', and 'Email'. Below the input fields is a dark 'Submit' button. At the bottom of the page, there is a small text label 'Daftar Sekarang'.

GAMBAR: 3. 26 Antarmuka halaman register versi mobile



LOGO

Personal Test

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Hubungi Kami

Hubungi Kami

The image shows a mobile home page. At the top, there is a 'LOGO' label. Below it is a rounded rectangular container titled 'Personal Test'. Inside this container, there is a paragraph of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.' Below the text is a dark button labeled 'Hubungi'. At the bottom of the page, there is a small text label 'Hubungi Kami'.

GAMBAR: 3. 27 Antarmuka halaman pembuka versi mobile



GAMBAR: 3. 28 Antarmuka halaman test versi mobile



GAMBAR: 3. 29 Antarmuka popup saat memilih jawaban versi mobile

## BAB IV

### IMPLEMENTASI DAN UJI COBA

#### 4.1 Pengkodean Sistem

Dalam tahap ini peneliti berfokus pada pengkodean menggunakan bahasa pemrograman javascript dengan menggunakan *framework VueJs* untuk halaman depan dan PHP untuk halaman *backoffice*, melakukan pengujian hasil menggunakan metode *blackbox testing* untuk pengujian fungsi dan menggunakan kuisisioner untuk pengujian faktor *usability end user*.

##### 4.1.1 Implementasi

Berikut adalah antarmuka bagian *end user* aplikasi rekrutmen PT Radya Gita Bahagi yang telah dibuat pada tahap sebelumnya.

1. Halaman Selamat Datang versi desktop



GAMBAR: 4. 1 Antarmuka Halaman Selamat Datang Ver Desktop



## 2. Halaman Register versi desktop



The screenshot shows a registration form on a dark background. At the top left, there is a logo with a crown and the text "WE ARE ROLLING GLORY". The form is a white rectangle with the following fields: "Nama Lengkap" with a sub-label "Masukkan Nama Lengkap", "No Telpun" with a sub-label "Masukkan No Telpun", and "Email" with a sub-label "Masukkan Email". A "Mula" button is located at the bottom center of the form. At the bottom of the page, the text "Rolling Glory ©2020" is visible.

GAMBAR: 4. 2 Antarmuka Halaman Register Ver Desktop

## 3. Halaman Panduan Test versi desktop



The screenshot shows a test guide page on a dark background. At the top left, there is a logo with a crown and the text "WE ARE ROLLING GLORY". The page is a white rectangle with the following content: "Sesi 1 - General" and "Panduan" in bold. Below this, there is a paragraph of text: "Pada setiap sesi terdiri dari 10 pertanyaan dengan 4 opsi berupa pilihan ganda. Silahkan pilih salah satu jawaban dengan mengklik opsi tersebut...". A "Mula" button is located at the bottom center of the page. At the bottom of the page, the text "Rolling Glory ©2020" is visible.

GAMBAR: 4. 3 Antarmuka Halaman Panduan Ver Desktop

#### 4. Halaman Test versi desktop



GAMBAR: 4. 4 Antarmuka Halaman Test Ver Desktop

#### 5. Popup saat memilih jawaban



GAMBAR: 4. 5 Antarmuka Popup Ver Desktop

## 6. Halaman Selesai versi desktop



GAMBAR: 4. 6 Antarmuka Halaman Selesai Ver Desktop

## 7. Halaman Penutup versi desktop



GAMBAR: 4. 7 Antarmuka Halaman Penutup Ver Desktop

## 8. Halaman Selamat Datang versi mobile



GAMBAR: 4. 8 Antarmuka Halaman Selamat Datang Ver Mobile

## 9. Halaman Registrasi versi mobile



The screenshot shows a mobile registration form with a black background. At the top left is the logo 'WE ARE ROLLING GLORY' with a crown icon. The form contains three input fields: 'Nama Lengkap' (Full Name), 'No Telepon' (Phone Number), and 'Email'. Each field has a placeholder text: 'Masukkan Nama Lengkap', 'Masukkan No Telepon', and 'Masukkan Email'. Below the fields is a dark 'Mulai' (Start) button. At the bottom, it says 'Rolling Glory ©2020'.

GAMBAR: 4. 9 Antarmuka Halaman Register Ver Mobile

## 10. Halaman Panduan Test versi mobile



GAMBAR: 4. 10 Antarmuka Halaman Panduan Ver Mobile

## 11. Halaman Test versi mobile



GAMBAR: 4. 11 Antarmuka Halaman Test Ver Mobile

## 12. Popup saat memilih jawaban versi mobile



GAMBAR: 4. 12 Antaarmuka Popup Ver Mobile

## 13. Halaman Selesai versi mobile



GAMBAR: 4. 13 Antarmuka Halaman Selesai Ver Mobile

## 14. Halaman Penutup versi mobile



GAMBAR: 4. 14 Antarmuka Halaman Penutup Ver Mobile

## 4.2 Menguji Sistem

Pengujian dilakukan dengan menjalankan aplikasi bagian frontend dan backoffice, lalu menjalankan fitur-fitur yang telah dibuat.

### 4.2.1 *Blackbox Testing*

Berikut adalah pengujian menggunakan *blackbox testing* :

TABEL: 4. 1 Pengujian *Blackbox* Pada Register Calon Karyawan

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Menguji Register jika Nama tidak diisi	Nama Kosong	Menampilkan pesan 'Nama Wajib diisi'	Sesuai
2	Menguji Register jika Telepon tidak diisi	Telepon Kosong	Menampilkan pesan 'Nomor Telepon Wajib diisi'	Sesuai
3	Menguji Register jika Email tidak diisi	Email Kosong	Menampilkan pesan 'Email Wajib diisi'	Sesuai
4	Mengisi Semua Form	Isi Nama, Isi Telepon, Isi Email	Masuk pada halaman panduan	Sesuai



TABEL: 4. 2 Pengujian *Blackbox* Pada Pengisian Test

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik salah satu jawaban	Klik jawaban	Memunculkan popup konfirmasi	Sesuai
2	Memilih tombol 'Ya' pada popup	Klik tombol 'Ya'	Menampilkan pertanyaan selanjutnya	Sesuai
3	Memilih tombol 'Tidak' pada popup	Klik tombol 'Tidak'	Menutup popup dan masih di pertanyaan yang sama	Sesuai

TABEL: 4. 3 Pengujian *Blackbox* Pada Login Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengosongkan username	Username Kosong	Menampilkan pesan ' <i>The username field is required</i> '	Sesuai
2	Mengosongkan password	Password Kosong	Menampilkan pesan ' <i>The username field is required</i> '	Sesuai
3	Mengisi username yang salah	Username salah	Menampilkan pesan ' <i>These credentials do not match our records</i> '	Sesuai
4	Mengisi Semua Form	Isi Username dan password dengan benar	Masuk pada halaman dashboard	Sesuai

TABEL: 4. 4 Pengujian *Blackbox* Pada Lihat Hasil Test oleh Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik menu User Quiz	Klik Menu User Quiz	Menampilkan halaman list User yang telah melakukan register	Sesuai
2	Mengklik tombol action Show	Klik tombol action Show	Menampilkan halaman detail user yang berisi data user dan hasil test user	Sesuai

TABEL: 4. 5 Pengujian *Blackbox* Pada Tambah Pertanyaan oleh Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik menu Upload Quiz	Klik Menu Upload Quiz	Menampilkan halaman list pertanyaan	Sesuai
2	Mengklik tombol +New	Klik tombol +New	Menampilkan halaman tambah pertanyaan	Sesuai
3	Menguji tambah data jika mengosongkan form Stage Id	Stage Id Kosong	Menampilkan pesan 'Stage Id harus diisi'	Sesuai
4	Menguji tambah data jika mengosongkan form Question	Question Kosong	Menampilkan pesan 'Question harus diisi'	Sesuai
5	Menguji tambah data jika mengosongkan form Option	Option Kosong	Menampilkan pesan 'Option harus diisi'	Sesuai

TABEL: 4. 5 Pengujian *Blackbox* Pada Tambah Pertanyaan oleh Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
6	Menguji tambah data jika mengisi semua form	Mengisi semua form	Data tersimpan dan muncul pada halaman list pertanyaan	Sesuai

TABEL: 4. 6 Pengujian *Blackbox* Pada Edit Pertanyaan oleh Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik menu Upload Quiz	Klik Menu Upload Quiz	Menampilkan halaman list pertanyaan	Sesuai
2	Mengklik tombol action Edit pada salah satu list	Klik tombol action Edit	Menampilkan halaman edit pertanyaan	Sesuai
3	Menguji edit data jika mengosongkan form Stage Id	Stage Id Kosong	Menampilkan pesan 'Stage Id harus diisi'	Sesuai
4	Menguji edit data jika mengosongkan form Question	Question Kosong	Menampilkan pesan 'Question harus diisi'	Sesuai
5	Menguji edit data jika mengosongkan form Option	Option Kosong	Menampilkan pesan 'Option harus diisi'	Sesuai
6	Menguji edit data jika mengisi semua form	Mengisi semua form	Data terupdate dan muncul pada halaman list pertanyaan	Sesuai

TABEL: 4. 7 Pengujian *Blackbox* Pada Hapus Pertanyaan oleh Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik menu Upload Quiz	Klik Menu Upload Quiz	Menampilkan halaman list pertanyaan	Sesuai
2	Mengklik list yang akan di hapus pada checkbox	Klik checkbox	Checkbox tercentang	Sesuai
3	Menekan tombol Batch Detelet	Klik tombol Batch Delete	Menampilkan popup konfirmasi	Sesuai
4	Menekan tombol Confirm pada popup	Klik tombol Confirm	Menghapus data	Sesuai
5	Menekan tombol Cancel pada popup	Klik tombol Cancel	Menutup popup	Sesuai

TABEL: 4. 8 Pengujian *Blackbox* Pada Logout Admin

No	Deskripsi Pengujian	Masukan	Hasil Yang Diharapkan	Hasil Yang Didapat
1	Mengklik menu User	Klik Menu User	Menampilkan dropdown menu	Sesuai
2	Mengklik tombol Logout	Klik tombol Logout	Keluar dari halaman admin dan menuju halaman login	Sesuai

#### 4.2.2 Kusioner

Pengujian ini dilakukan untuk menguji faktor *usability* aplikasi rekrutmen dari sisi *end user* pada halaman test rekrutmen. Faktor *usability* memperhatikan 5 komponen *usability* yaitu *Learnability*, *Efficiency*, *Memorability*, *Errors*, dan

*Satisfaction*. Pengujian ini dilakukan kepada 10 responden acak. Berikut adalah hasil dari kuisisioner:

TABEL: 4. 9 Hasil Kuisisioner untuk mengukur faktor *Usability*

NO	Pertanyaan	Nilai
<b>Learning</b>		
1.	Apakah aplikasi mudah di akses di perangkat yang kamu gunakan?	4.7
2.	Apakah kamu mengalami kesulitan dalam menggunakan aplikasi ini?	4.6
3.	Apakah informasi yang disajikan dapat kamu pahami dengan sangat baik?	4.1
4.	Apakah tampilan aplikasi terasa nyaman saat digunakan?	4.7
<b>Efficiency</b>		
1.	Apakah kamu dapat menyelesaikan test dengan baik?	3.8
2.	Apakah navigasi dan informasi saat test berlangsung dapat membantu dengan baik?	4.6
<b>Memorability</b>		
1.	Apakah tampilan aplikasi mudah di kenali?	4.7
2.	Apakah tampilan dan alur mudah diingat?	4.5
<b>Errors</b>		
1.	Apakah kamu kesulitan dengan alur yang ada di aplikasi?	2.2
2.	Apakah kamu sempat mengalami kesalahan dalam menjalankan aplikasi? Jika ya sebutkan pada bagian mana km mengalami kesalahan	1.4
<b>Satisfaction</b>		
1.	Apakah kamu puas dengan aplikasi ini?	4.6
2.	Secara keseluruhan seberapa nyaman kamu dengan aplikasi ini?	4.5

Dengan nilai rata-rata per atribut sebagai berikut

TABEL: 4. 10 Hasil Kuisisioner untuk mengukur faktor *Usability*

Nilai Rata-rata	
<b>Learning</b>	4.5
<b>Efficiency</b>	4.2
<b>Memorability</b>	4.6
<b>Errors</b>	1.8
<b>Satisfaction</b>	4.6

Tabel 4.9 menunjukkan nilai-nilai kepuasan atau penerimaan user terhadap masing-masing atribut. Dapat dilihat bahwa untuk atribut kenyamanan aplikasi dan aplikasi yang mudah dikenali memiliki nilai penerimaan *usability* oleh user sebesar 4,7 (sudah berada diatas nilai 3) dalam skala 5. Hal ini dapat diartikan bahwa aplikasi rekrutmen yang telah dibuat membuat nyaman user dan mudah dikenali oleh user dari segi antarmuka.

Apabila disesuaikan kembali hubungannya dengan masing-masing aspek usability dalam Tabel 4.10, dapat dikatakan bahwa perangkat lunak aplikasi android yang telah dibuat telah memiliki nilai *Usability*, yaitu: *Learnability*, *Efficiency*, *Memorability*, dan *Satisfaction* yang sangat baik, namun pada aspek *Errors* masih belum mencapai nilai 3. Hal ini ditunjukkan dengan nilai hasil usability pada kelima atribut, sebagai berikut :

- a. Pada atribut "kemudahan dan nyaman antarmuka aplikasi" memiliki nilai sebesar 4,5 ini menunjukkan bahwa aplikasi memenuhi aspek *Learning*

- b. Pada atribut "kemudahan pengoperasian aplikasi" memiliki nilai sebesar 4,2 ini menunjukkan bahwa aplikasi memenuhi aspek *Efficiency*.
- c. Pada atribut "kemudahan antarmuka dan alur aplikasi" memiliki nilai sebesar 4,6 ini menunjukkan bahwa aplikasi memenuhi aspek *Memorability*.
- d. Pada atribut "seberapa sering mengalami kesalahan" ini memiliki nilai 1,8 menunjukkan bahwa aplikasi belum memenuhi aspek *Errors*, namun berdasarkan keterangan yang responden berikan hal ini bukan karna seberapa sering mereka melakukan kesalahan dalam menjalankan aplikasi tetapi karena pertanyaan yang cukup sulit untuk dijawab.
- e. Dan pada keseluruhan aplikasi memiliki nilai 4,6 hal ini menunjukkan jika aplikasi rekrutmen telah mempunyai aspek *Satisfaction* dengan sangat baik.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan seluruh hasil tahapan penelitian yang telah dilakukan pada Pembuatan Aplikasi *Computer Based Test* Berbasis Web Untuk Proses Rekrutmen (Studi Kasus PT Radya Gita Bahagi) dapat disimpulkan sebagai berikut,

1. Pembuatan *aplikasi computer based test* berbasis web untuk proses awal rekrutmen ini dapat mempermudah proses tes seleksi karyawan pada PT Radya Gita Bahagi, karena dengan adanya aplikasi ini akan mempermudah HR dalam menentukan kandidat calon karyawan yang akan maju tahap selanjutnya berdasarkan *score test* yang dihasilkan.
2. Hasil Rekap Nilai *Usability* menunjukkan keempat aspek memiliki nilai penerimaan usability oleh user, rata-rata diatas nilai 4, namun pada salah satu aspek mendapatkan nilai di bawah 3 dikarenakan *user* memiliki kesulitan dalam menjawab. Tetapi dapat dikatakan bahwa aplikasi rekrutmen yang telah dibuat sudah memiliki nilai aspek *Usability*, sehingga memudahkan *user* untuk menggunakan dalam perangkat apapun dan membuat *user* nyaman dalam menggunakannya.



## 5.2 Saran

Berdasarkan adanya keterbatasan dalam pembuatan aplikasi rekrutmen ini, maka terdapat beberapa saran yang disampaikan sebagai pengembangan aplikasi rekrutmen kedepannya yaitu:

1. Menambahkan fitur tes yang dapat dikelompokkan berdasarkan posisi/kategori.
2. Menambahkan fitur untuk technical tes pada aplikasi sehingga tidak perlu lagi mengirim melalui email.
3. Jika fitur sudah berkembang, dapat menambahkan fitur login untuk calon karyawan yang sedang proses rekrutmen.

## DAFTAR PUSTAKA

A.S Rosa, dan M. S. (2014) *Rekayasa Perangkat Lunak Struktur dan Berorientasi Objek*. Bandung: Informatika.

Abdullah, R. (2015) *Web Programing is Easy*. Jakarta: Elek Media Komputindo.

Al-Amri (2011) 'Computer-Based Testing vs. Paper-Based Testing: A Comprehensive Approach to Examining the Comparability of Testing Modes', *Essex Graduate Student Papers in Language & Linguistics*, Vol. 10, p. hal. 22-44.

Deni Darmawan (2013) *Sistem Informasi Manajemen*. Bandung: PT. Remaja Rosdakarya.

Fadli, M. (2020) 'Website, Rancang Bangun Aplikasi Rekrutmen Karyawan Pada PT. Kasa Husada Wira Jatim Berbasis', *Journal of Chemical Information and Modeling*, 21(1), pp. 1–9.

Iswanto, A. aziz muslim (2018) 'Perancangan dan implementasi rekrutmen karyawan menggunakan metode saw berbasis web ( studi kasus PT. Ateja)', *Jurnal teknologi informasi dan komunikasi (FIKI)*, VIII(2).

Janner Simarmata (2010) *Rekayasa Perangkat Lunak*. Yogyakarta: Penerbit Andi.

Jaya, T. S. and Widyawati, D. K. (2019) 'Pengembangan E-Market Place Pertanian Dengan Metode Prototype Development of Agricultural E-Marketplace By Prototype Method'.

L. Mathis, Robert & H. Jackson, J. (2011) *Human Resource Management (edisi 10)*. Jakarta: Salemba Empat.

Mardianto, A. (2014) *Management Recruitmen*. Jakarta: Pinasthika

publisher.

Maulani, J., Kom, S. and Kom, M. (2016) 'Aplikasi Test Psikotes Dan Kepribadian Berbasis Delphi', 7(2), pp. 117–121.

Nugroho, A. (2010) *Rekayasa Perangkat Lunak Menggunakan UML & Java*. Yogyakarta: Andi Offset.

Putrawansyah, F. (2017) 'Aplikasi Computer Assisted Test (CAT) Pada Penerimaan Mahasiswa Baru Sekolah Tinggi Teknologi Pagar Alam (STTP)', 1(1), pp. 1–8.

Rex, B. (2009) *Managing The Testing Process: Practical Tools and Techniques for Managing Hardware and Software Testing*. Indiana: Wiley Publishing.Inc.

Rolling Glory (2012) *Misi & Visi*. Available at: <https://shrine.rollingglory.com/rolling-glory/misi-and-visi>.

Simamora, H. (2015) *Manajemen Sumber Daya Manusia*. Yogyakarta: STIEY.

Susilowati, Y. (2019) *Modul E-Commerce - Teaching Factory For Students*. Mutiara Publisher.

Thiemann, T. S. and P. (2012) *Functional and Logic Programming*.

Trisnawati, L. and Syafrizal, E. (2016) 'Rancangan Sistem Rekrutmen Karyawan Berbasis Web Pada Pt. Fast Food Indonesia Region Pekanbaru', *Jurnal Teknologi dan Sistem Informasi UNIVRAB*, 1(1), pp. 25–30.

Whitten, Jeffrey L., E. a. (2014) *System Analysis and Design Methods*. 6th edn. New York: Mc Graw-Hill.

Yulianto, H., Wahyuni, T. and Eka, Y. (2016) 'Ujian on Line Dalam E-Learning: Perbandingan Ujian Online (Computer Based) Terhadap Ujian

Tradisional (Paper Based)', *Jurnal Vokasi Indonesia*, 4(2). doi:  
10.7454/jvi.v4i2.100.

## **LAMPIRAN**

## Lampiran 1

### Daftar Pertanyaan Kuisisioner Penggunaan Web Aplikasi Rekrutmen

NO	Pertanyaan
	Apakah perangkat yang kamu gunakan? Komputer Desktop/Laptop atau Smartphone?
	<b>Learning</b> Learnability, yaitu tingkat kemudahan pengguna dalam mempelajari produk untuk menyelesaikan tugas-tugas mereka.
1.	Apakah aplikasi mudah di akses di perangkat yang kamu gunakan?
2.	Apakah kamu mengalami kesulitan dalam menggunakan aplikasi ini?
3.	Apakah informasi yang disajikan dapat kamu pahami dengan sangat baik?
4.	Apakah tampilan aplikasi terasa nyaman saat digunakan?
	<b>Efficiency</b> Menjelaskan seberapa cepat pengguna dapat menyelesaikan tugas-tugas yang ada saat mereka pertama kali mempelajari sistem tersebut.
1.	Apakah kamu dapat menyelesaikan test dengan baik?
2.	Apakah navigasi dan informasi saat test berlangsung dapat membantu dengan baik?
	<b>Memorability</b> Menjelaskan tentang tingkat kemudahan pengguna atau user dalam menggunakan sistem dengan baik, setelah beberapa lama tidak menggunakannya.

1.	Apakah tampilan aplikasi mudah di kenali?
2.	Apakah tampilan dan alur mudah diingat?
	<p><b>Errors</b></p> <p>Menjelaskan kemungkinan terjadinya error atau kesalahan yang dilakukan oleh pengguna dan seberapa mudah mereka dapat mengatasinya.</p>
1.	Apakah kamu kesulitan dengan alur yang ada di aplikasi?
2.	Apakah kamu sempat mengalami kesalahan dalam menjalankan aplikasi? Jika ya sebutkan pada bagian mana km mengalami kesalahan
	<p><b>Satisfaction</b></p> <p>Seberapa nyaman pengguna dengan aplikasi ini</p>
1.	Apakah kamu puas dengan aplikasi ini?
2.	Secara keseluruhan seberapa nyaman kamu dengan aplikasi ini?
	<b>Pertanyaan tidak wajib di isi</b>
	Mohon berikan masukan jika masih banyak kekurangan dalam aplikasi (tidak wajib)

## Lampiran 2

### Kode Program

#### Halaman Selamat Datang

```
<template>
  <div>
    <div
      class="text-welcome"
    >
      <h3>Selamat Datang</h3>
      <h4>di Rolling Rekrutmen</h4>
      <p>Ini adalah sebuah proses tes rekrutmen pada perusahaan kami
.</p>
      <p>Selamat Mengerjakan</p>
      <router-link to="/profile">
        <b-button
          pill
          variant="primary"
        >
          Mulai
        </b-button>
      </router-link>
    </div>
  </div>
</template>

<style lang="scss" scoped>

.text-welcome {
  margin-top: 30px;
  padding: 30px;
}
</style>
```



## Halaman Panduan dan Halaman Selesai

```
<template>
  <div>
    <div class="text-welcome">
      <div class="test-title">
        <h3>{{ this.$route.path !== '/stagetest' ? 'Sesi 1 - General'
          : (currentQuestion.step + 2 == finishNumber ? 'Selesai': `Sesi ${currentQuestion.step + 2} - `) }}</h3>
        <span
          v-
          if="this.$route.path == '/stagetest' && currentQuestion.step + 2 !== finishNumber"
          >
          <span v-
            if="currentQuestion.step + 2 == 2"> Personal</span>
          <span v-
            if="currentQuestion.step + 2 == 3"> Cultural Fit</span>
          </span>
        </div>
        <div v-
          if="this.$route.path== '/stagetest' && currentQuestion.step + 2 == finishNumber">
          <ScoreFinished />
        </div>

        <div v-else>
          <h4>Panduan</h4>
          <p>
            Pada setiap sesi terdiri dari 10 pertanyaan dengan 4 opsi berupa pilihan ganda. Silahkan pilih salah satu jawaban dengan mengklik opsi tersebut.
          </p>
        </div>

        <router-link
          v-if="this.$route.path !== '/stagetest'"
          to="/stagetest"
          >
          <b-button
            pill
            variant="primary"
            >
            Mulai
          </b-button>

```

```

</router-link>
<template v-else>
  <b-button
    v-if="currentQuestion.step + 2 != finishNumber"
    pill
    variant="primary"
    @click="nextStep"
  >
    Mulai
  </b-button>
  <b-button
    v-else
    pill
    variant="primary"
    to="/finishstage"
  >
    Selesai
  </b-button>
</template>
</div>
</div>
</template>

<script>
import ScoreFinished from './test/ScoreFinished.vue';

export default {
  components: {
    ScoreFinished,
  },
  props: ['currentQuestion'],
  data() {
    return {
      finishNumber: 4,
      test: this.currentQuestion, // jumlah semua stage tambah 1
    };
  },
  watch: {
    test() {
      if(this.test.step + 2 === finishNumber) {
        console.log('Done');
      } else {
        console.log('not done');
      }
    }
  }
}

```

```
    },  
    methods: {  
      nextStep() {  
        this.$emit('nextStep');  
      },  
    },  
  },  
};  
</script>
```

```
<style lang="scss" scoped>  
h3 {  
  font-weight: bold;  
}  
.text-welcome {  
  margin-top: 30px;  
  padding: 30px;  
}  
.test-title {  
  font-size: 1.575rem !important;  
  display: flex;  
  justify-content: center;  
  line-height: 1.2;  
  font-weight: bold;  
}  
.total-score {  
  font-size: 28px;  
  font-weight: 600;  
}  
</style>
```

## Halaman Test

```
<template>
  <div>
    <StageResumeAlert
      v-if="!isFinish"
      v-model="modalShow"
    />
    <div
      v-show="!isFinish"
      class="container-fluid"
    >
      <div v-if="!showStageResult">
        <Question
          :list-questions="listQuestions"
          :current-question="currentQuestion"
        />
        <RadioButton
          :list-questions="listQuestions"
          :list-answer="listAnswer"
          :current-question="currentQuestion"
          :confirm-answer="confirmAnswer"
          :is-true="isTrue"
          @confirmAnswerUser="confirmAnswerUser"
        />
        <BoxConfirm
          :confirm-answer="confirmAnswer"
          @cekAnswer="cekAnswer"
          @dismissAnswer="dismissAnswer"
        />
      </div>
      <div v-else>
        <StageResult
          :list-questions="listQuestions"
          :list-answer="listAnswer"
          :current-question="currentQuestion"
          @nextStep="nextStep"
        />
      </div>
    </div>
    <div v-show="isFinish">
      <FinishStage />
    </div>
  </div>
</template>
```

```

<script>
import { mapGetters } from 'vuex';
import RadioButton from './RadioButton.vue';
import Question from './Question.vue';
import StageResult from './StageResult.vue';
import StageResumeAlert from './StageResumeAlert.vue';
import BoxConfirm from './confirm/BoxConfirm.vue';
import FinishStage from './FinishStage.vue';

import * as type from '../..//store/types';

export default {
  components: {
    RadioButton,
    Question,
    BoxConfirm,
    StageResult,
    StageResumeAlert,
    FinishStage
  },

  data() {
    return {
      currentQuestion: { number: 0, step: 0 },
      listQuestions: null,
      listAnswer: null,
      isLoading: true,
      isTrue: null,
      showStageResult: false,
      userData: null,
      confirmAnswer: {
        idQuestion: null,
        status: false,
        indexAnswer: null,
        optionChoice: null,
      },
      isFinish: false,
      modalShow: false,
    };
  },
  computed: {

  },

  mounted() {

```

```

        this.getQuizQuestion();
    },
    methods: {
        async getQuizQuestion() {
            console.log(localStorage.getItem('userData'));

            if (localStorage.getItem('userData') === null) {
                this.$router.push('profile');
            } else {
                const loader = this.$loading.show();
                const userData = JSON.parse(localStorage.getItem('userData'));

                this.userData = userData;
                this.$store
                    .dispatch(type.ACT_DATA_QUIZ, userData.id)
                    .then((res) => {
                        this.listQuestions = res.listQuestions;
                        if (localStorage.getItem('userAnswer') === null) {
                            this.listAnswer = res.listAnswer;
                        } else {
                            this.listAnswer = JSON.parse(localStorage.getItem('userAnswer'));
                        }
                        this.cekSession();
                        this.isLoading = false;
                        loader.hide();
                    })
                    .catch((err) => {
                        console.log(err);
                        loader.hide();
                    });
            }
        },

        confirmAnswerUser(idxQuestion) {

            const userAnswer = this.listQuestions[this.currentQuestion.step][
                this.currentQuestion.number
            ].options[idxQuestion];

            this.confirmAnswer.idQuestion = this.listQuestions[
                this.currentQuestion.step
            ]

```

```

    ][this.currentQuestion.number].qId;
    this.confirmAnswer.status = true;
    this.confirmAnswer.indexAnswer = idxQuestion;
    this.confirmAnswer.optionChoice = userAnswer;
  },

  dismissAnswer() {
    this.confirmAnswer.status = false;
    this.confirmAnswer.indexAnswer = null;
    this.confirmAnswer.optionChoice = null;
    this.confirmAnswer.idQuestion = null;
  },

  cekAnswer(answer) {
    localStorage.removeItem('lastSession');
    localStorage.removeItem('userAnswer');

    const userAnswer = {
      id: answer.idQuestion,
      quiz_answer: answer.optionChoice.no,
    };
    const loader = this.$loading.show();
    this.$store
      .dispatch(type.ACT_CEK_ANSWER_QUIZ, userAnswer)
      .then((res) => {

        this.isTrue = null;
        this.nextQuestion();

        loader.hide();
        localStorage.setItem('userAnswer', JSON.stringify(
y(this.listAnswer)));
      })
      .catch((err) => {
        loader.hide();
        console.log(err);
      });
    localStorage.setItem('lastSession', JSON.stringify(this.
currentQuestion));

    this.confirmAnswer.status = false;
  },

  nextQuestion() {
    this.confirmAnswer.indexAnswer = null;

```

```

        this.confirmAnswer.optionChoice = null;
        this.confirmAnswer.idQuestion = null;
        if (
            this.currentQuestion.number + 1 ===
this.listQuestions[this.currentQuestion.step].length
        ) {
            this.showStageResult = true;
        } else if (
            this.currentQuestion.number + 1 <
this.listQuestions[this.currentQuestion.step].length
        ) {
            this.currentQuestion.number += 1;
            this.isTrue = null;
            this.showStageResult = false;
        }
    },
    async nextStep() {

        if (this.currentQuestion.step + 1 < this.listQuestions.l
length) {

            this.currentQuestion.step += 1;
            this.currentQuestion.number = 0;
            this.showStageResult = false;
            this.isTrue = null;
            localStorage.setItem(
                'lastSession',
                JSON.stringify(this.currentQuestion)
            );
        } else if (this.currentQuestion.step + 1 === this.listQu
estions.length) {
            this.isFinish = true;
            this.getFinishScore();
        }
    },

    cekSession() {
        if (localStorage.getItem('lastSession') === null) {
            this.currentQuestion = { number: 0, step: 0 };
        } else {
            const userDataSession = JSON.parse(localStorage.getI
tem('lastSession'));
            this.currentQuestion = {
                number:
                    userDataSession.number <
                    this.listQuestions[userDataSession.step].length - 1
            };
        }
    }
};

```



```

        ? userDataSession.number + 1
        : userDataSession.number,
        step: userDataSession.step,
    });

    if (
        userDataSession.number ===
        this.listQuestions[userDataSession.step].length - 1
    ) {
        this.showStageResult = true;
        this.isTrue = 'resume';
    }

    if (userDataSession.step === this.listQuestions.leng
th - 1) {
        this.getFinishScore();
        this.showStageResult = true;
        this.isTrue = 'finish';
        this.isFinish = true;
    }
    },
    downloadData() {
        localStorage.removeItem('lastSession');
        localStorage.removeItem('userAnswer');
        localStorage.removeItem('userData');
        this.$router.push('profile');
    },
    },
};
</script>

```

```

<style lang="scss">
    .finish-step {
        margin: 60px 0;
        h3 {
            font-weight: bold;
        }
    }
</style>

```

### Kode test.js

```
import axios from 'axios';
import jsonApi from 'jsonapi-serializer';
import moment from 'moment';

import * as type from './types';

const JSONAPIDeserializer = jsonApi.Deserializer;
const JSONAPISerializer = jsonApi.Serializer;

// STATE
const state = {
  finishScore: null,
};

// GETTERS
const getters = {
  finishScore: (state) => state.finishScore,
};

// MUTATIONS
const mutations = {
  [type.MUTATE_SET_FINISH_SCORE](state, finishData) {
    console.log('Ulala');
    state.finishScore = finishData;
    console.log(state.finishScore);
  }
};

const actions = {

  [type.ACT_POST_USER_DATA]({commit}, userData) {
    const UserSerializer = new JSONAPISerializer('sessions', {
      attributes: ['username', 'email', 'phone', 'is_finished', 'start_date']
    });

    const userDataReg = UserSerializer.serialize(userData);

    const resultConvertSymbol = JSON.stringify(userDataReg).replace(
      /-/g,
      '-'
    );
  }
};
```

```

return new Promise((resolve, reject) => {
  axios({
    method: 'post',
    url: '/api/v1/sessions',
    headers: {
      'Content-Type' : 'application/json'
    },
    data: JSON.parse(resultConvertSymbol)
  })
  .then(res => {
    new JSONAPIDeserializer().deserialize(res.data,
function fn(
    err,
    userDataDesrlz
  ){
    const dateTime = moment().format('YYYY-MM-
DD h:mm:ss');

    const saveData = {
      id: userDataDesrlz.id,
      name: userDataDesrlz.username,
      email: userDataDesrlz.email,
      phone: userDataDesrlz.phone,
      dateSession: dateTime
    };

    localStorage.setItem(
      'userData',
      JSON.stringify(saveData)
    );
  });

  resolve(res);
})
.catch(function fn(error){
  const {errors} = error.response.data;
  let message = '';
  errors.forEach(el => {
    message += `${el.detail }\n`;
  });
  reject(message);
});
});
},

// get data list question quiz by session ID

```

```

[type.ACT_DATA_QUIZ]({ commit }, sessionID) {
  return new Promise((resolve, reject) => {
    axios
      .get(`/api/v1/sessions/${ sessionID}`)
      .then(res => {
        new JSONAPIDeserializer().deserialize(res.data,
function fn(
          err,
          quizzes
        ) {
          const dataStage = [];
          quizzes.questions.forEach((quiz, index) => {
            const spliceData = quizzes.questions.splice(
ice(0, 10);

              dataStage[index] = spliceData;
            });

            const dataQuestion = [];
            dataStage.forEach((stage, index) => {
              const dataOption = [];
              stage.forEach((st, x) => {
                const options = [];
                let i = 0;
                const alphabet = ['A', 'B', 'C', 'D']

                for (const key in st.quiz) {
                  if (key.includes('option')) {
                    options[i] = {
                      option: st.quiz[key],
                      no: alphabet[i]
                    };
                    i += 1;
                  }
                }
                dataOption[x] = {
                  options,
                  qId: st.id,
                  qnumber: st['quiz-order'],
                  qtext: st.quiz.question,
                  trueAnswer: st.answer
                };
              });
              dataQuestion[index] = dataOption;
            });
          });
        });
      });
    });
  });
};

```

```

        // make asnwer data
        const dataAnswer = [];
        for (let i = 0; i < dataQuestion.length; i +
= 1) {
            const detail = [];
            for (
                let i2 = 0;
                i2 < dataQuestion[i].length;
                i2 += 1
            ) {
                detail[i2] = {
                    no: dataQuestion[i][i2].qnumber,
                    answer: null,
                    answerNumber: null,
                    trueAnswer: null
                };
            }
            dataAnswer[i] = { stageScore: 0, data: d
etail };
        }

        // final data
        const dataQuiz = {
            listQuestions: dataQuestion,
            listAnswer: dataAnswer
        };
        resolve(dataQuiz);
    });
}
.catch(err => console.log(err));
});
},

// CEK TRUE/FALSE ANSWER
[type.ACT_CEK_ANSWER_QUIZ]({ commit }, asnwerData) {
    return new Promise((resolve, reject) => {
        const AnswerSerializer = new JSONAPISerializer('session-
quizzes', {
            attributes: ['quiz_answer']
        });
        const userAnswer = AnswerSerializer.serialize(asnwerData
);
        axios({
            method: 'patch',
            url: `/api/v1/session-quizzes/${ asnwerData.id}`,

```

```

        headers: {
            'Content-Type': 'application/json'
        },
        data: userAnswer
    })
    .then(res => {
        return new JSONAPIDeserializer()
            .deserialize(res.data)
            .then(data => {
                const returnData = {
                    id: data.id,
                    isRight: true,
                    quizOrder: data['quiz-order'],
                    sessionID: data['session-id'],
                    trueAnswer: data.answer
                };
                resolve(returnData);
            });
    });
    .catch(err => console.log(err));
});
},

// GET FINISH SCORE
[type.ACT_GET_FINISH_SCORE]({ commit }) {
    return new Promise((resolve, reject) => {
        let userSession = localStorage.getItem('userData');
        if (userSession !== null) {
            userSession = JSON.parse(userSession);
            const data = {
                id: userSession.id,
                is_finished: 1,
                start_date: userSession.dateSession,
                // end_date: '',
            };
            const AnswerSerializer = new JSONAPISerializer('sessions', {
                attributes: ['is_finished', 'start_date']
            });
            const dataSession = AnswerSerializer.serialize(data);

            axios({
                method: 'patch',
                url: `/api/v1/sessions/${ userSession.id}`,
            })
        }
    });
}

```

```

        headers: {
            'Content-Type': 'application/json'
        },
        data: dataSession
    })
    .then(res => {
        return new JSONAPIDeserializer()
            .deserialize(res.data)
            .then(dataDesrlz => {
                const returnData = {
                    name: dataDesrlz.username,
                    email: dataDesrlz.email,
                    phone: dataDesrlz.phone,
                    score: dataDesrlz.score
                };
                console.log(returnData);
                commit(
                    type.MUTATE_SET_FINISH_SCORE,
                    returnData
                );
            });
    })
    .catch(err => console.log(err));
    });
}
});
};

export default {
    actions,
    getters,
    mutations,
    state,
};

```