

BAB II

LANDASAN TEORI

2.1 Kajian Terhadap Penelitian Sebelumnya

Dalam penelitian (Novian, 2020) dengan judul “Penggunaan Algoritma

Profile Matching Untuk Perekrutan Karyawan” metode ini akan diimplementasikan ke dalam sebuah program sehingga akan menghasilkan sebuah sistem pendukung keputusan yang dapat menghitung dan menampilkan bentuk keluaran berupa perbandingan yang dapat membantu pimpinan dalam memutuskan siapa saja calon karyawan yang akan direkrut.

Dalam jurnal penelitian (Titis, 2020) dengan judul “Rancang Bangun Sistem Pendukung Keputusan Penerimaan Karyawan Baru Menggunakan *Naive Bayes Classifier*” digunakan untuk membantu divisi HRD dalam pengambilan keputusan rekrutmen karyawan yang diterima maupun tidak diterima.

2.2 Metode *Profile Matching*

2.2.1 Pengertian *Profile Matching*

Metode pencocokan profil (*profile matching*) adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel predictor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Dalam *profile matching*, dilakukan identifikasi terhadap kelompok

karyawan yang baik maupun buruk.

2.2.2 Tahapan Perhitungan *Profile Matching*

Tahapan dalam metode *profile matching* dilakukan sebagai berikut:

1. Pembobotan

Padat ahap inidilakukan prosesmenentukan boboty angdihasilkan dari selisih (gap), profil individu dan profil standar yang dijadikan sebagai acuan.

Gap = profil individu - profil standar.

2. Pengelompokan Kriteria

Setelah seluruh kriteria ditetapkan, maka langkah selanjutnya mengelompokkan masing-masing kriteria ke dalam kelompok *Core Factor* dan *Secondary Factor*. *Core Factor* merupakan faktor utama yang menonjol dan dijadikan acuan sehingga menghasilkan kinerja optimal yang dirumuskan, sedangkan *Secondary Factor* merupakan faktor pendukung.

Rumus *core factor*:

$$= \frac{\sum h}{\sum h}$$

Keterangan:

NCF : Nilai rata-rata *core factor*

NC : Jumlah total nilai *core*

factor IC : Jumlah item *core factor*

Rumus *secondary factor*

$$= \frac{\sum}{\sum}$$

$$= \frac{h}{h} \frac{\text{---}}{\text{---}}$$

Keterangan:

NSF : Nilai rata-rata *secondary factor*

NS : Jumlah total nilai *secondary*

factor IS: :Jumlah item *secondaryfactor*

3. Perhitungan NilaiTotal

Setelah didapatkan nilai *core factor* dan *secondary factor* dari setiap kriteria, selanjutnya dilakukan perhitungan nilai total masing-masing kandidat (alternatif), digunakan rumus:

$$= (\quad)\% + (\quad)\%$$

Keterangan:

N : Nilai total tiap aspek

NCF : Nilai rata-rata

corefactor

NSF : Nilai rata-rata *secondaryfactor*

(X)% : Nilai persentase yangdiinputkan

4. Padatahapinidilakukanprosesmengurutkankandidatdariyangpaling direkomendasikan ditempatkan pada bagian atas ranking ditandai dengan nilai paling besar pada perhitungan nilaitotal.

2.3 Algoritma *Naïve Bayes Classification*

2.3.1 Pengertian *Naïve Bayes Classification*

Naive Bayes merupakan metode klasifikasi yang dapat digunakan untuk mendukung keputusan yang termasuk dalam metode data mining dan termasuk dalam *classifer statistic* yang bisa digunakan untuk memprediksi probabilitas keanggotaan *class*. Kelebihan *Naive Bayes*

Classification adalah sederhana tetapi

memiliki tingkat akurasi yang tinggi dengan asumsi independensi (ketidak ketergantungan) yang kuat (naif).

2.3.2 Langkah-langkah Algoritma *Naïve Bayes Classification*

Langkah-langkah dari Algoritma *Naïve Bayes Classification* (Dong-Chul Park, 2016) adalah sebagai berikut:

1. Membaca dataset training.
2. Menghitung jumlah kelas/label.

$$P(C_i) = \frac{S_i}{S}$$

Dimana S_i adalah jumlah training sample pada class C_i , dan S adalah jumlah *training sample*.

3. Menghitung nilai probabilitas hipotesis.

$$P(C_i|X) = \frac{P(C_i) \cdot P(X_1|C_i) \cdot P(X_2|C_i) \cdot P(X_3|C_i)}{P(X)}$$

Keterangan :

$P(C_i|X)$: Probabilitas hipotesis C_i jika diberikan fakta atau *record* X
(*Posterior probability*)

$P(X|C_i)$: Mencari nilai parameter yang memberi kemungkinan yang paling besar (*likelihood*)

$P(C_i)$: *Prior probability* dari X (*Prior probability*)

$P(X)$: Jumlah *probability tuple* yg muncul

2.4 Perangkat Lunak yang Digunakan

2.4.1 Pengertian XAMPP

Xampp adalah sebuah *software web server apache* yang didalamnya sudah tersedia database server *MySQL* dan dapat mendukung pemrograman *PHP*. *XAMPP* merupakan *software* yang mudah digunakan gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah Cuma menginstal satu kali sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHP Support* (*PHP 4* dan *PHP 5*) dan beberapa *module* lainnya.

2.4.2 Pengertian PHP

PHP (*Hypertext Preprocessor*), merupakan bahasa pemrograman yang bekerja dalam sebuah *web server*. *Script-script* *PHP* harus tersimpan dalam sebuah *server* dan dieksekusi atau proses dalam *server* tersebut. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif.

2.4.3 Pengertian MySQL

MySQL merupakan singkatan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. *MySQL* merupakan sistem manajemen *database* yang bersifat *relational*. Artinya, data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan jauh lebih cepat.

2.4.4 PHPMyAdmin

PHP MyAdmin adalah perangkat lunak bebas yang ditulis dalam

bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi
MySQL melalui

wordl wide web. PHP MyAdmin mendukung berbagai operasi *MySQL*,diantaranya mengolah basis data, tabel-tabel, *fields, relasi, indeks, user, permission*, dan lain- lain.

2.5 Permodelan Sistem DenganUML

2.5.1 PengertianUML

UML adalah bahasa standar untuk membuat rancangan *software*. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifakdari *software intensive system*. UML (*unifed modeling language*) sesuai dengan kata- kata terakhir dari kepanjangannya, UML merupakan salah satu bentuk *language* ataubahasa.Menurutpencetusnya,UMLdidefinisikansebagaibahasavisualu ntuk menjelaskan, memberikan spesifikasi, merancang, membuat model dan mendokumentasikan aspek-aspek dari sebuahsistem.

Karena merupakan bahasa visual, UML lebih mengedepankan penggunaan diagramuntukmenggambarkanaspekdarisistemyangsedangdimodelkan.Ba hasa visual lebih ke mental model pikiran kita, sehingga permodelan menggunakan bahasa visual lebih mudah dan lebih cepat dipahami dibandingkan apabila dituliskan dalam sebuah bahasa pemrograman.

Unified Modeling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak, UML menawarkan sebuah standar untuk merancang model sebuah sistem.

2.5.2 Model-model Diagram UML

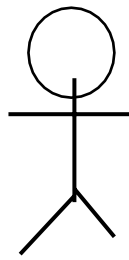
1. Use Case Diagram

Use case diagram merupakan sebuah teknik yang digunakan dalam pengembangan sebuah sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan. *use case* menggambarkan *external view* dari sistem yang akan kita buat modelnya, *use case* diagram menjelaskan interaksi antara “aktor” dan “inisiator” dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah *use case* direpresentasikan dengan urutan langkah yang sederhana.

Fokus dari sebuah *use case* adalah bagaimana mencapai sebuah tujuan atau goal. Setiap *use case* merupakan rangkaian seri yang lengkap dari *event* kejadian jika dilihat dari sudut pandang *actor*. *Actor* merupakan sesuatu atau seseorang yang ada di luar sistem dan ikut berperan serta dalam aktifitas sistem. *Actor* bisa berperan *end user*, perangkat *hardware*, bahkan sistem yang lain.

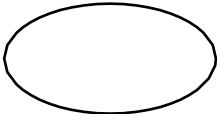
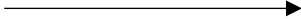
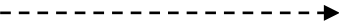
Tabel 2. 1 Komponen *Use Case* Diagram

Komponen <i>Use Case</i>	Penjelasan
--------------------------	------------

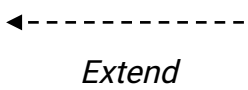


Actor

Actor, adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau

	sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
 <i>Use Case</i>	<i>Use case</i> menggambarkan fungsional yang disediakan sistem sebagai unit-unit yang bertuar pesan antar unit dengan <i>actor</i> , biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
 <i>Association</i>	Asosiasi antara <i>actor</i> dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
 <i>Generalization</i>	Asosiasi antara <i>actor</i> dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila <i>actor</i> berinteraksi secara pasif dengan sistem.
 <i>Include</i>	<i>Include</i> , merupakan di dalam <i>use case</i> lain, (<i>required</i>) atau pemanggilan <i>use case</i> oleh

	h
--	---

	<i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

2. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*attributes*), Operasi (*operations/method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel 2.2 *Multiplicity Class Diagram*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih

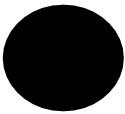
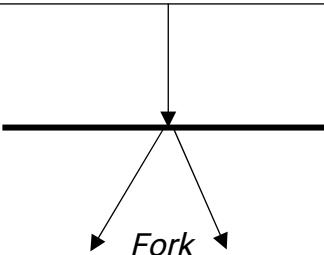
1..*	1 atau lebih
------	--------------

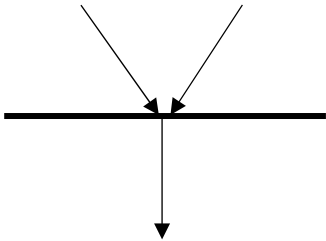
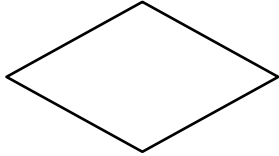
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara, contoh 2..4 mempunyai arti minimal 2 maksimum 4

3. *Activity*Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Komponen-komponen yang digunakan dalam *activity* diagram, yaitu:

Tabel 2.3 Komponen *Activity* Diagram

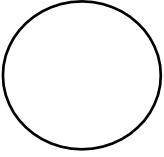
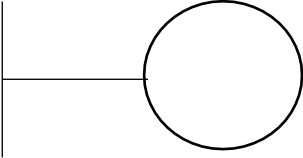
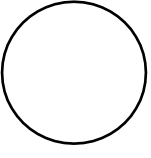
<i>Activity</i> Diagram	Penjelasan
 <i>Start point</i>	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
 <i>End point</i>	<i>End point</i> , akhir aktifitas.
 <i>Activites</i>	<i>Activites</i> , menggambarkan suatu proses/kegiatanbisnis.
 <i>Fork</i>	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau digunakan untuk

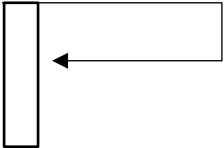
	digabungkan dua kegiatan parallel menjadisatu.
 <i>Join</i>	<i>Join</i> (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
 <i>Decision Points</i>	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
 <i>Swimlane</i>	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.

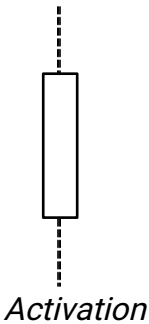
4. *SequenceDiagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Komponen-komponen yang digunakan dalam *sequence diagram*:

Tabel 2.4 Komponen *Sequence Diagram*

Sequence Diagram	Penjelasan
 <i>Entity Class</i>	<i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
 <i>Boundary Class</i>	<i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih <i>actor</i> dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
 <i>Control Class</i>	<i>Control Class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
 <i>Message</i>	<i>Message</i>, simbol mengirim pesan antar <i>class</i>.

 <i>Recursive</i>	<i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
---	---

 The diagram shows a vertical dashed line representing a lifeline. A solid rectangular bar is drawn across this line, representing an activation period. The word "Activation" is written below the bar. <i>Activation</i>	<i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjangkotak ini berbanding lurus dengandurasi aktivitas sebuahoperasi.
 The diagram shows a single vertical dashed line representing a lifeline. The word "Lifeline" is written below the line. <i>Lifeline</i>	<i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis dan Perancangan Sistem

3.1.1 Analisa Permasalahan

Pembuatan aplikasi ini diawali dengan melakukan analisa terhadap kebutuhan perusahaan. Analisis kebutuhan dilakukan dengan mengajukan beberapa pertanyaan mendasar seperti siapa saja yang akan menggunakan aplikasi tersebut, apa saja fungsi dari aplikasi tersebut dan output seperti apa yang diinginkan oleh pengguna pada program tersebut.

Dalam proses penanganan rekrutmen di PT. Rasa Sayang Ais Krim masih dilakukan secara manual, belum menggunakan *system* yang terkomputerisasi baik aplikasi berbasis desktop maupun aplikasi berbasis *website*. Semua proses penanganan rekrutmen karyawan dilakukan secara konvensional melalui tatap muka langsung kepada pihak perusahaan atau dengan cara menulis surat lamaran kepada pihak perusahaan.

Untuk mengatasi permasalahan tersebut, bagian Human Resource Department di perusahaan mendesain langsung suatu program penerimaan karyawan berbasis web. Dengan penerimaan karyawan berbasis web akan mempermudah bagian HRD suatu perusahaan untuk menerima karyawan.

Permasalahan yang sering terjadi pada proses penanganan rekrutmen karyawan pada perusahaan dan pelamar adalah:

A. Permasalahan Pada Perusahaan

1. Dengan sistem manual yang ada, maka akan memperlambat kerja perusahaan dalam tahapan-tahapan rekrutmen.
2. Informasi rekrutmen karyawan belum terdokumentasi dan tersimpan didalam database sehingga mudah hilang.

B. Permasalahan Pada Pelamar

1. Waktu yang digunakan untuk mencari informasi lowongan pekerjaan terbatas hanya pada jam kerja.
2. Belum mempunyai media yang bersifat luas untuk melakukan komunikasi di luar jam kerja dalam hal rekrutmen karyawan.

3.1.2 Analisa Kebutuhan Sistem

Dari permasalahan diatas maka ditentukan kebutuhan pada system baru sebagai berikut:

1. Memberikan kemudahan kepada pelamar untuk mendapatkan informasi yang dibutuhkan.
2. Memberikan kemudahan bagi perusahaan untuk menyebarkan informasi tentang pekerjaan kepada masyarakat luas dengan tepat dan efisien.
3. Memberikan kemudahan pelamar untuk mengikuti tahapan tahapan dalam rekrutmen karyawan tanpa harus datang ke perusahaan.
4. Memberikan kemudahan bagi pelamar untuk mencoba tes yang diadakan oleh pihak perusahaan tanpa harus datang ke perusahaan.

5. Memfasilitasi masyarakat secara umum dan calon pelamar secara khusus dengan pihak perusahaan untuk melakukan komunikasi tanpa melalui tatap muka secara langsung.

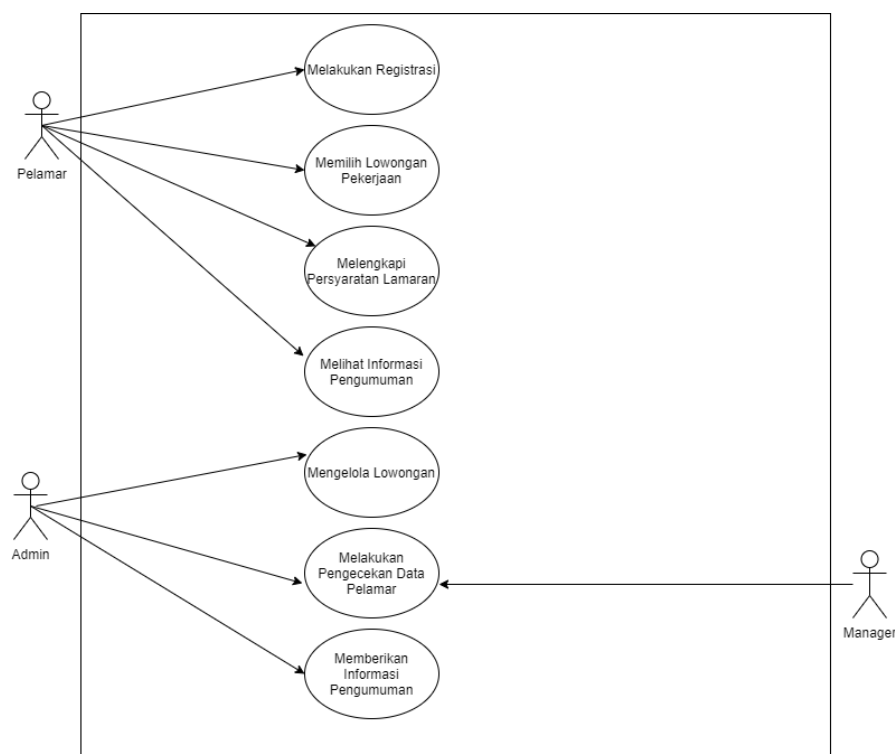
3.2 Perancangan Sistem

3.2.1 Perancangan Model Sistem dengan UML

a. Use Case Diagram

Berikut adalah *use case* diagram dari aplikasi sistem rekomendasi perekrutan karyawan pada PT Rasa Sayang Ais Krim yang mana *use case* diagram

memperlihatkan hubungan antara aktor dan *use case* dalam sistem ini, terdapat tiga aktor diantaranya adalah pelamar, admin dan HRD.



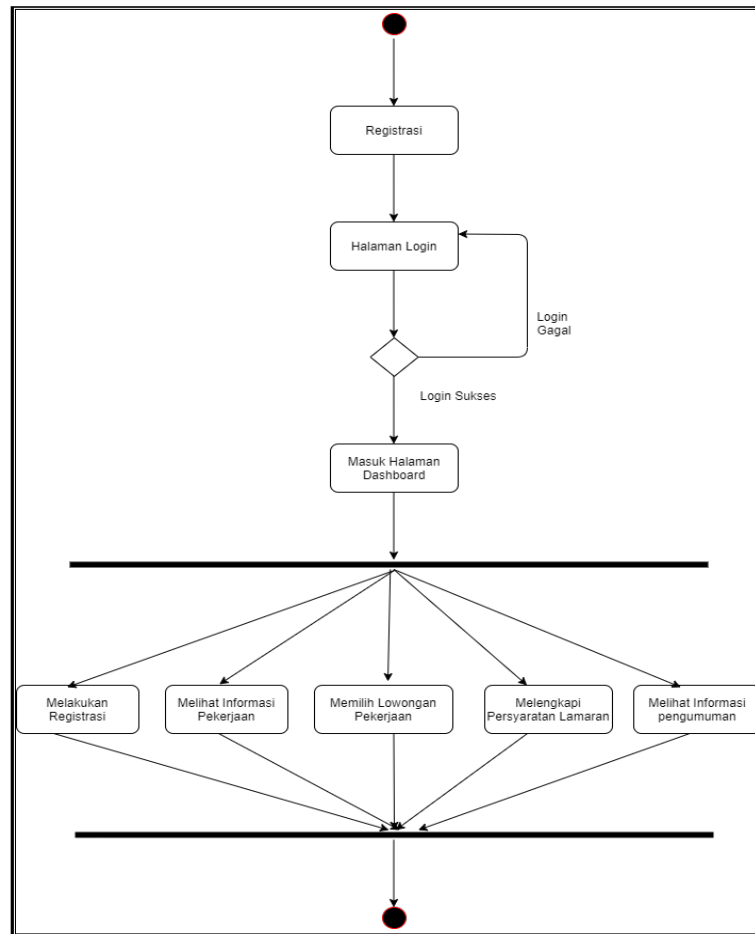
Gambar 3. 1 Use Case Diagram

b. *ActivityDiagram*



Gambar 3. 2 *Activity Diagram Admin*

Dari *Activity* diagram admin dari gambar diatas dapat dilihat bahwa admin harus *login* terlebih dahulu untuk dapat masuk ke halaman utama. Didalam aplikasi admin memiliki hak akses untuk *login*, mengelola lowongan kerja, melakukan pengecekan registrasi pelamar, melakukan pengecekan data pelamar, melakukan persetujuan lamaran, dan memberikan informasi pengumuman.

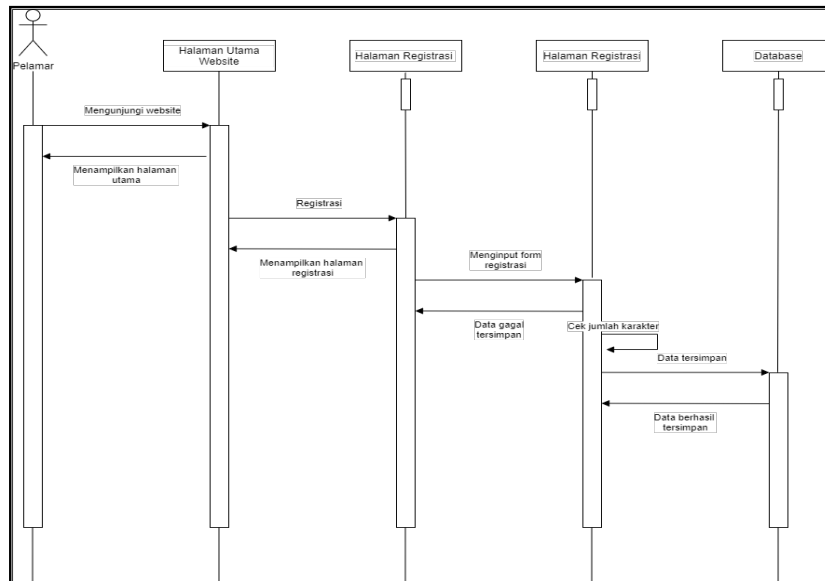


Gambar 3. 3 *Activity* Diagram Pelamar

Dari *activity* diagram pelamar pada gambar dapat dilihat bahwa pelamar harus melakukan registrasi terlebih dahulu untuk mendapatkan akun, kemudian melakukan *login* untuk masuk ke halaman utama. Pada halaman utama pelamar dapat melihat data diri, melakukan pengisian test serta melengkapi persyaratan lamaran dan melihat informasi pengumuman.

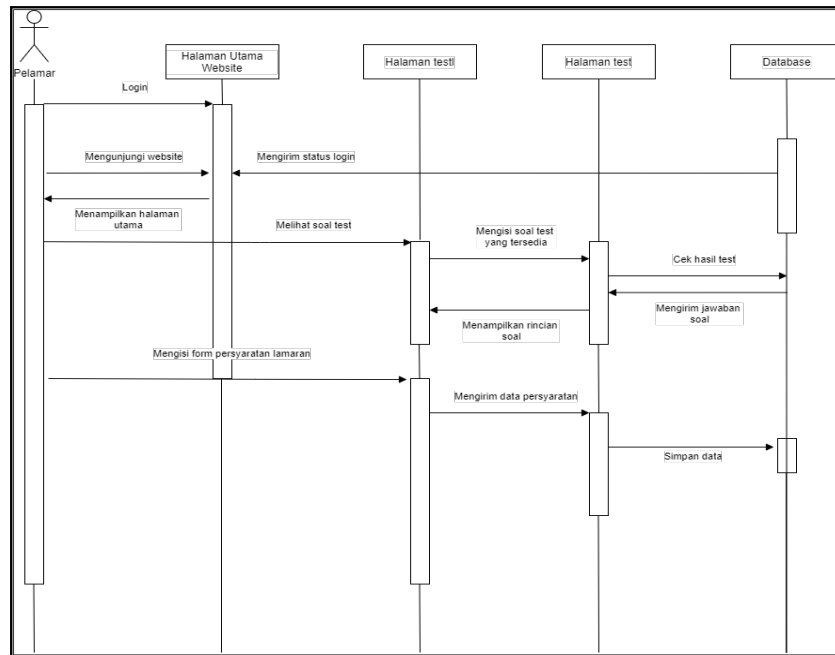
c. *SequenceDiagram*

Sequence diagram merupakan gambaran interaksi antar objek dan mengidentifikasi komunikasi diantara objek-objek. *Sequence* diagram juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu.



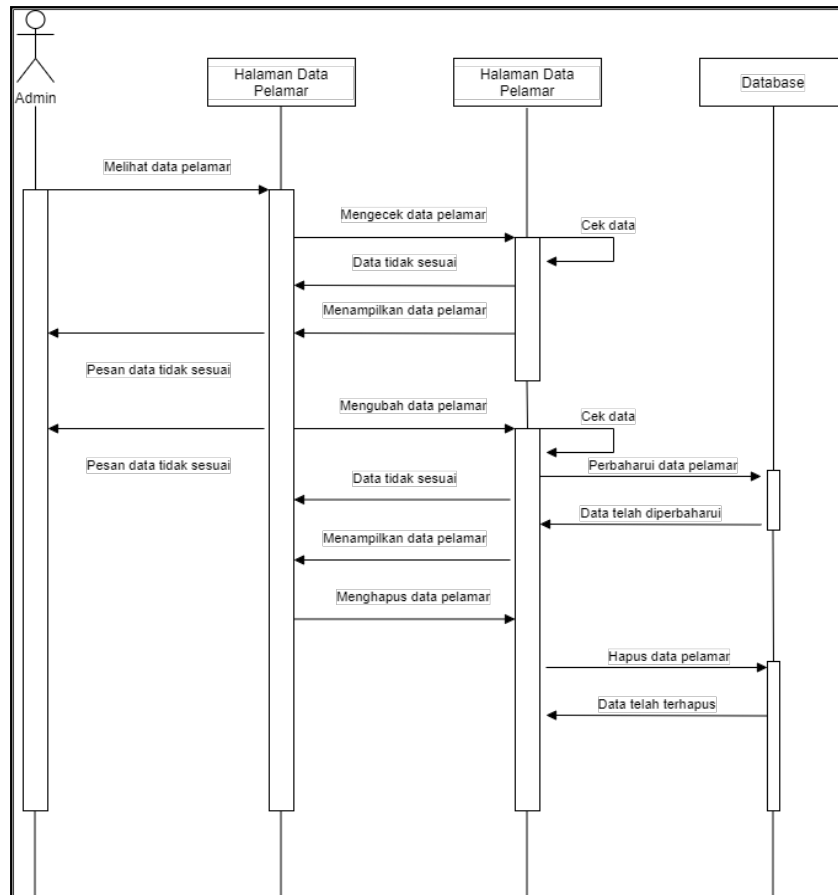
Gambar 3. 4 *Sequence* Diagram Registrasi

Sequence diagram registrasi menjelaskan kepada pelamar untuk melakukan registrasi untuk masuk ke dalam website.



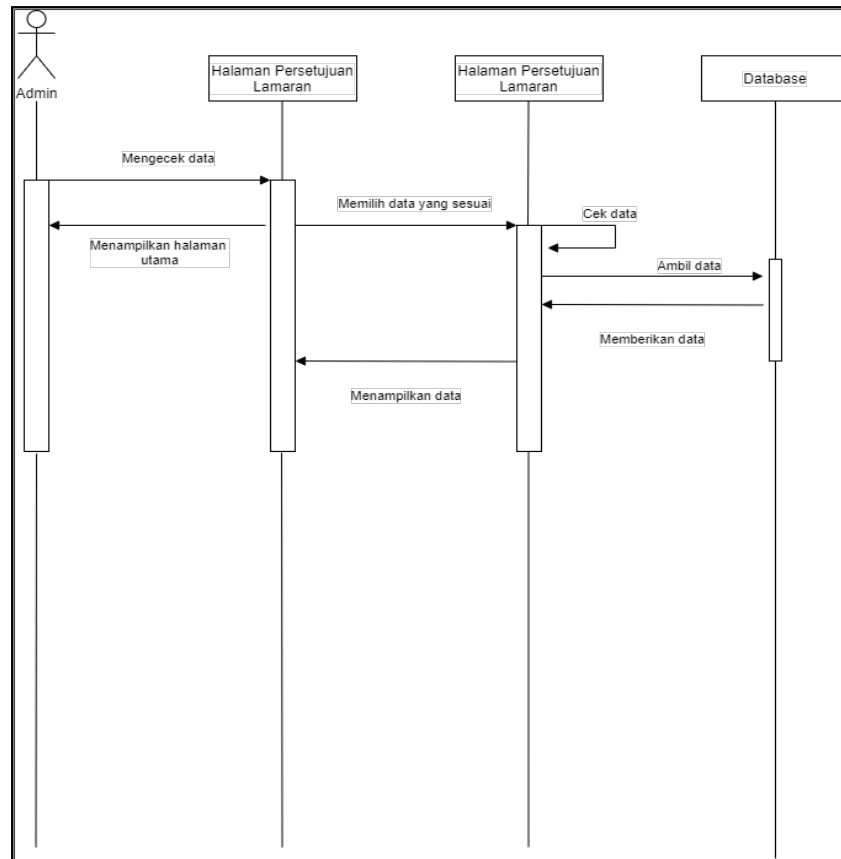
Gambar 3. 5 *Sequence Diagram Pelamar*

Sequencediagram pelamar menjelaskan bahwa saat pelamar masuk ke dalam halaman utama pelamar bisa melihat data diri, melakukan pengisian test yang tersedia dan melengkapi persyaratan lamaran.



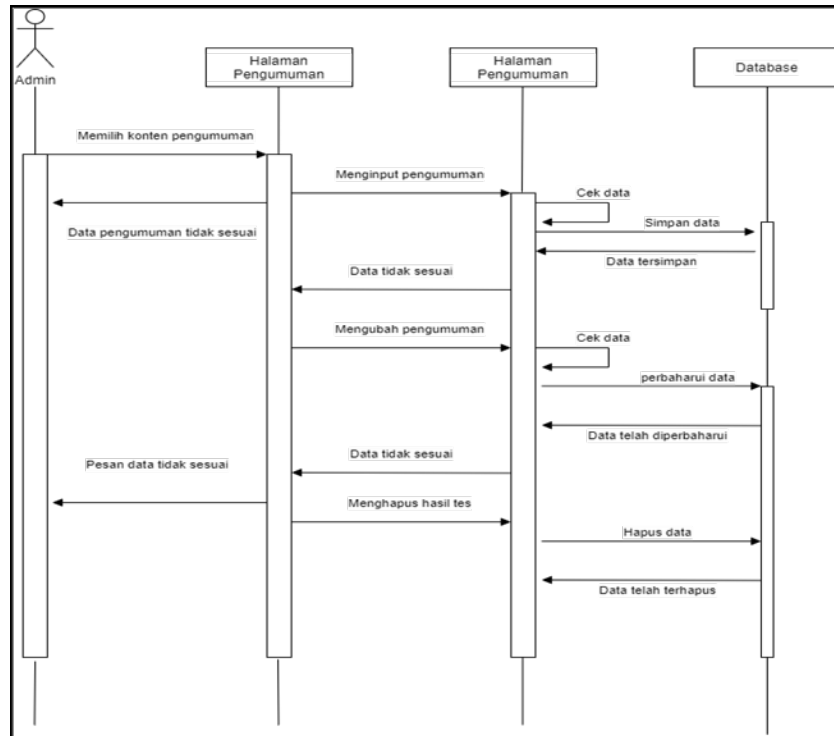
Gambar 3. 6 *Sequence* Diagram Pengecekan Data Pelamar

Sequence diagram pengecekan data pelamar menjelaskan bahwa admin dapat melihat data para pelamar.



Gambar 3. 7 *Sequence Diagram* Persetujuan Lamaran

Sequence diagram persetujuan lamaran menjelaskan bahwa admin harus memilih data yang sesuai setelah melakukan pengecekan data pelamar.



Gambar 3. 8 *Sequence* Diagram Pengumuman

Sequence diagram pengumuman menjelaskan bahwa sebelum melakukan *penginputan* pengumuman admin harus memilih konten pengumuman apa yang akan dibuat.

3.3 PerancanganDatabase

3.3.1 StrukturDatabase

Terdapat 10 (sepuluh) tabel yang dibuat dalam perancangan aplikasi sistem rekondasi perekrutan karyawan da PT Rasa Saya Ais Krim, yaitu tabel user, tabel calon, tabel dokumen, tabel rank, tabel isi, tabel kriteria, tabel matriks, tabel pengumuman, tabel lowongan tabel soal dan tabel jawaban.

Tabel 3. 1 Tabel *User*

No	Field	Type	Length	Primary
1	id_user	int	1	id (primarykey)
2	level	varchar	20	
3	username	varchar	50	
4	password	varchar	50	
5	red_date	datetime		
6	rod_date	datetime		
7	last_login	timestamp		

Tabel 3. 2 Tabel *Calon*

No	Field	Type	Length	Primary
1	id	int	5	id (primarykey)
2	username	varchar	50	
3	nama	varchar	45	
4	gender	enum		
5	no_tlp	varchar	15	
6	alamat	varchar	50	
7	email	varchar	30	
8	tgl_lahir	date		
9	tgl_daftar	date_time		
10	id_lowongan	int	1	
11	password	varchar	35	

Tabel 3. 3 Tabel Dokumen

No	Field	Type	Length	Primary
1	id	int	1	id (primarykey)
2	label	text		
3	dokumen	text		
4	extension	varchar	10	
5	id_calon	int	1	
6	upload_date	datetime		

Tabel 3.4 Tabel Rank

No	Field	Type	Length	Primary
1	alternatif	int	1	
2	nilai_akhir	double		

Tabel 3. 5 Tabel Isi

No	Field	Type	Length	Primary
1	id	int	1	primary key
2	user	int	1	
3	soal	int	1	
4	jawaban	int	1	
5	hasil	varchar	20	
6	reg_time	timestamp		

Tabel 3.6 TabelKriteria

No	Field	Type	Length	Primary
1	id	int	1	primary key
2	nama	varchar	100	
3	sifat	varchar	30	
4	bobot	int	1	

Tabel 3.7 TabelMatriks

No	Field	Type	Length	Primary
1	alternatif	int	1	
2	id_kriteria	int	1	
3	nilai	double		
4	result	varchar	30	

Tabel 3.8 Tabel Pengumuman

No	Field	Type	Length	Primary
1	id	int	1	primary key
2	text	text		
3	type	varchar	20	
4	active	int	1	
5	due_date	date		

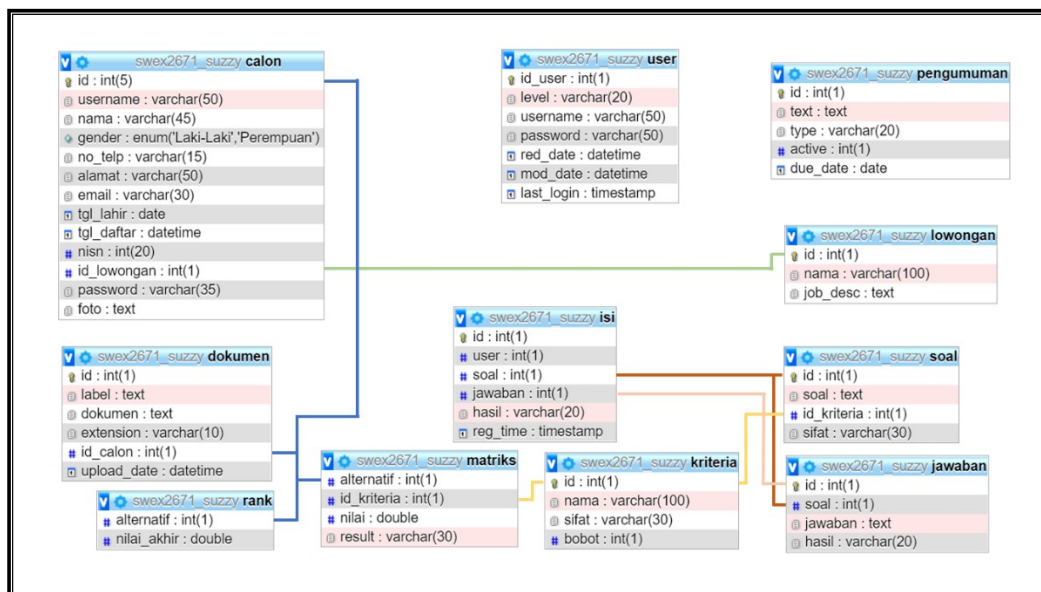
Tabel 3.9 Tabel Lowongan

No	Field	Type	Length	Primary
1	id	int	1	primary key
2	nama	varchar	100	
3	job_desc	text	20	

Tabel 3.10 Tabel Jawaban

No	Field	Type	Length	Primary
1	id	int	1	primary key
2	soal	int	1	
3	jawaban	text		
4	hasil	varchar	20	

3.3.2 Relasi Database



Gambar 3. 9 Relasi Database

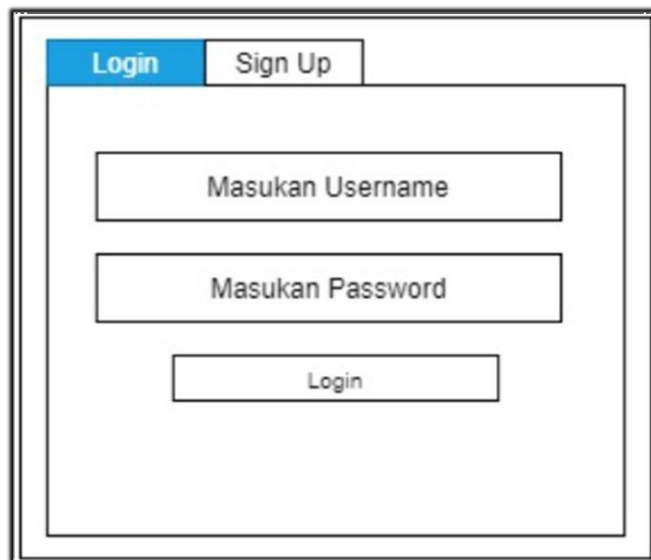
3.4 PerancanganTampilan

Perancangan tampilan yang akan dibuat untuk Sistem Rekomendasi PerekrutanKaryawanpadaPTRasaSayangAisKrim.Yaituberisitampilanadmi
n
yangmerupakantampilanyangdapatdiaksesolehHRD(admin).Halpertamaya
ng harus dilakukan adalah melakukan design atau tampilan dari halaman
aplikasi itu sendiri. Kemudian setelah rancangan selesai, tahap berikutnya
adalah melakukan pembuatan fungsi-fungsi dan juga konten dari aplikasi
tersebut.

3.4.1 Rancangan Tampilan Halaman Hak AksesAdministrator

1. HalamanLogin

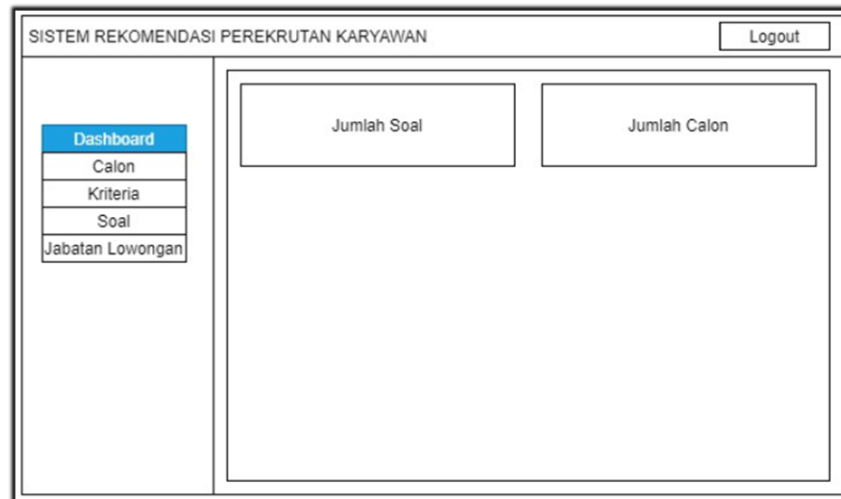
Dibawah ini adalah merupakan tampilan halaman awal dari sistem rekomendasi perekrutan karyawan pada PT Rasa Sayang Ais Krim, yaitu halaman login.



Gambar 3. 10 Rancangan Halaman Login

2. Halaman UtamaAdministrator

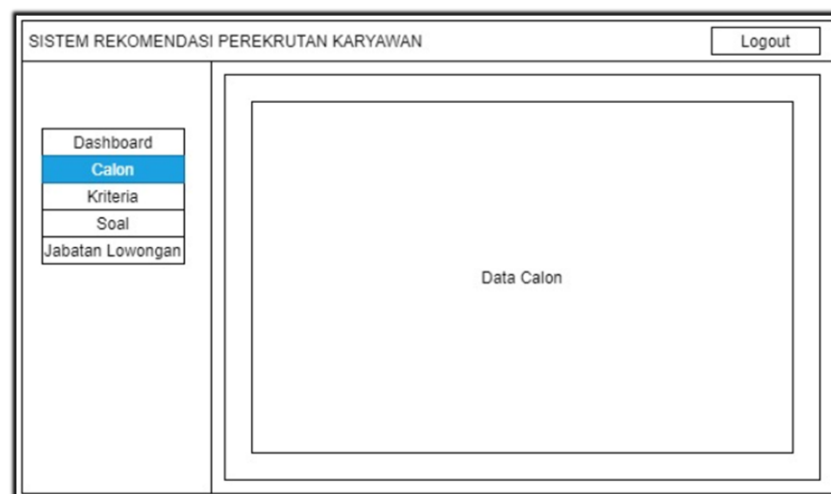
Dibawah ini adalah merupakan tampilan halaman utama dari sistem rekomendasi perekrutan karyawan pada PT Rasa Sayang Ais Krim, yaitu halaman utama.



Gambar 3. 11 Rancangan Halaman Dashboard Admin

3. Halaman Data CalonPelamar

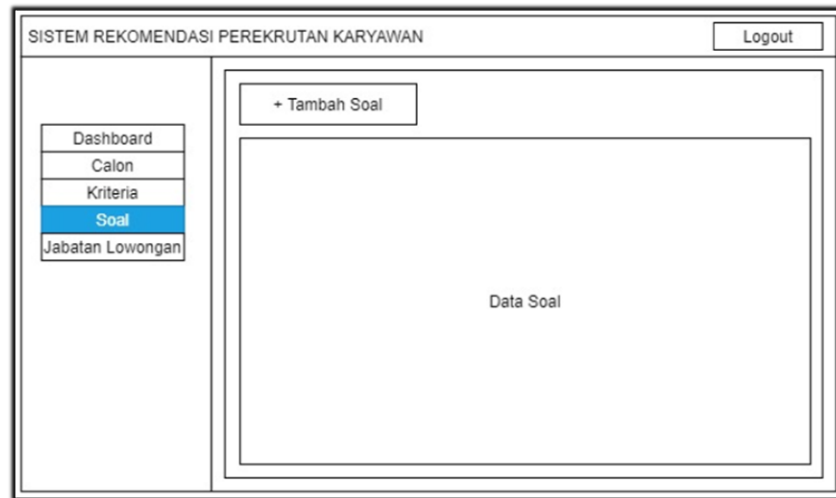
Dibawah ini adalah merupakan tampilan halaman data calon pelamar pada dashboard admin.



Gambar 3. 12 Rancangan Halaman Data Calon

3. Halaman Data Soal

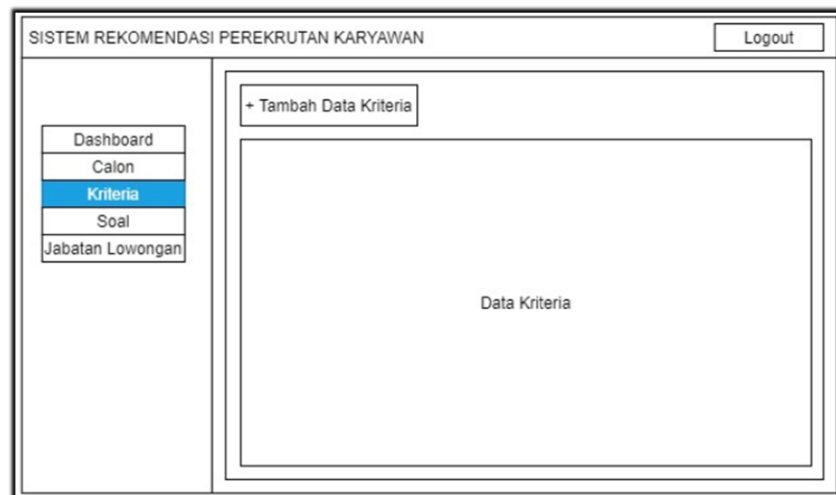
Dibawah ini adalah merupakan tampilan halaman Data Soal



Gambar 3. 13 Rancangan Halaman Data Soal

5. Halaman DataKriteria

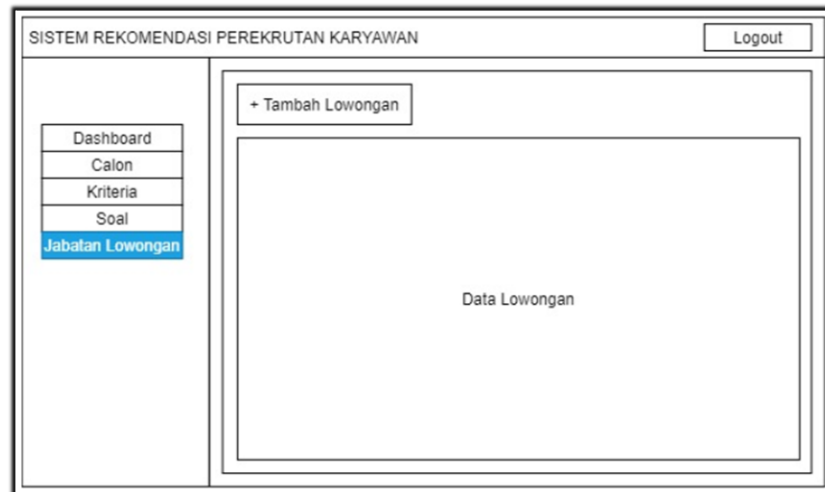
Dibawah ini adalah merupakan tampilan halaman data kriteria



Gambar 3. 14 Rancangan Halaman Kriteria

6. Halaman Jabatan Lowongan

Dibawah ini adalah merupakan tampilan halaman data jabatan lowongan



Gambar 3. 15 Rancangan Halaman Jabatan Lowongan