

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Hasil**

Berikut ini adalah pembahasan dari hasil penelitian yang dilakukan dengan metode pengembangan sistem *Waterfall*.

##### **4.1.1 Komunikasi**

Pada tahap komunikasi ini, penulis melakukan tahapan wawancara kepada pimpinan, staff alumni dan alumni pada IIB DARMAJAYA, dan mendapatkan kesimpulan bahwa alumni memiliki kesulitan untuk memilih lowongan kerja yang sesuai dengan kemampuan dan minatnya. Selain itu dengan penyesuaian minat dan kemampuan diawal sebelum melamar pekerjaan maka HRD akan lebih mudah untuk melakukan seleksi. Sehingga diperlukan sebuah sistem yang akan menjadi jembatan penyelesaian masalah tersebut, sistem akan memberikan rekomendasi pekerjaan kepada alumni sesuai kemampuan dan minatnya, melakukan lamaran kerja dan akan mempermudah HRD untuk melakukan seleksi dari segi minat dan kemampuan. HRD akan bisa lebih fokus pada proses wawancara lanjutan.

##### **4.1.2 Perencanaan**

Setelah melakukan wawancara, penulis selanjutnya melakukan perencanaan kebutuhan sistem sesuai dengan hasil wawancara tersebut. Perencanaan kebutuhan dibagi dalam 2 bagian, yaitu kebutuhan fungsional sistem dan kebutuhan non fungsional sistem. Berikut penjelasan dari hasil perencanaan yang dilakukan

###### **1. Kebutuhan Fungsional**

Kebutuhan fungsional merupakan apa saja yang menjadi syarat agar sistem yang dibangun sesuai dengan kebutuhan yang direncanakan. Berikut adalah kebutuhan fungsionalnya

a. Pelamar Kerja

- Pelamar kerja melakukan pendaftaran akun pencari kerja serta mengisi data diri secara benar.
- Pelamar kerja melengkapi dokumen pendidikan yang telah ditempuh.
- Pelamar kerja mengisi estimasi gaji yang diharapkan.
- Pelamar kerja mengisi skill yang dimiliki dan juga minat yang sesuai dengan dirinya.
- Pelamar kerja mendapatkan rekomendasi beberapa lowongan kerja yang sesuai dengan kriterianya.
- Pelamar kerja dapat melakukan lamaran cepat pada suatu lowongan pekerjaan.
- Pelamar dapat melihat kemajuan lamaran kerjanya.

b. Staff

- Staff dapat menambahkan data minat, pendidikan, skill yang secara umum ada dalam dunia kerja.
- Staff dapat menambahkan data lowongan kerja.
- Staff dapat mengubah status pendaftaran pelamar kerja.
- Staff dapat mengubah status lolos atau tidaknya pelamar kerja

c. Pimpinan

- Pimpinan dapat melihat data pelamar kerja
- Pimpinan dapat melihat data lowongan kerja dan siapa saja yang melakukan lamaran pada posisi tersebut
- Pimpinan dapat mengubah status pelamar

2. Kebutuhan non Fungsional

Analisis kebutuhan non fungsional adalah mengidentifikasi batasan fasilitas yang disediakan oleh sistem. Sistem aplikasi dan data base dilengkapi password

### **4.1.3 Modeling**

Setelah melakukan perencanaan, penulis membuat permodelan sederhana dengan metode Promethee untuk memudahkan dalam memahami alur penentuannya.

#### **4.1.3.1 Penentuan Kriteria**

Kriteria yang akan digunakan dalam penentuan lowongan kerja paling sesuai untuk pelamar kerja antara lain

1. Usia pelamar, usia pelamar menjadi salah satu kriteria yang diperlukan dalam kesesuaian lowongannya, kriteria usia yang bernilai paling besar adalah yang memiliki usia paling minimal. Misalnya ada 2 lowongan A dan B, dan pelamar berusia 30 tahun. Lowongan A memiliki persyaratan usia 18-30 tahun dan lowongan B memiliki persyaratan usia 25-35 tahun, maka pelamar akan lebih memiliki bobot maksimal pada lowongan B dibandingkan lowongan A.
2. Estimasi gaji yang diharapkan oleh pelamar. Misalkan lowongan A memiliki gaji 3.000.000-4.000.000 dan lowongan B memiliki gaji 1.500.000-2.500.000 sedangkan pelamar memiliki standar gaji yang diinginkan yaitu 3.500.000. Maka secara rekomendasi maka sistem akan memberikan lowongan A dari kriteria gaji yang diharapkan.
3. Skill yang dimiliki oleh pelamar, kriteria ini dilihat dari jumlah skill yang dibutuhkan oleh masing-masing lowongan, misalnya pelamar memiliki 4 skill (S1,S2,S3,S4) lowongan A membutuhkan skill S2,S4 dan lowongan B membutuhkan skill S1,S2,S3,S5,S6. Skill pelamar pada lowongan A adalah 2 skill yang sesuai sedangkan pada lowongan B ada 3 skill yang sesuai, maka sistem akan memberikan rekomendasi lowongan B pada pelamar.
4. Minat pelamar yang sesuai dengan lowongan, misalnya pelamar memiliki 5 minat yaitu M1,M2,M3,M4,M5. Lowongan A diisi dengan 4 minat yaitu

M1,M3,M4,M6 sedangkan lowongan B diisi dengan 2 minat yaitu M2,M6. Dari contoh tersebut, pelamar memiliki kesesuaian 3 minat dengan lowongan A dan 1 minat dengan lowongan B, dari perhitungan itu sistem akan memberikan rekomendasi untuk mengambil lowongan A.

5. Pendidikan, tidak dipungkiri lagi ini adalah kriteria utama yang harus memiliki kesesuaian antara pelamar dan lowongan kerja. Pelamar yang memiliki kesesuaian dengan lowongan kerja dibidang pendidikan minimal akan diberikan nilai sedangkan yang tidak sesuai diberikan nilai 0. Jenjang pendidikan akan memiliki poin yang berbeda, SMA/SMK sederajat memiliki poin 1, S1 memiliki poin 2, S2 memiliki poin 3 dan S3 memiliki poin 4
6. Jurusan, memiliki penilaian yang sama dengan pendidikan

#### 4.1.3.2 Normalisasi Kriteria

Tiap kriteria lowongan akan dinormalisasi sesuai nilai batasnya, perhitungan normalisasi diilustrasikan seperti berikut.

##### a. Spesifikasi Pelamar (P)

No	Kriteria	Value
1	Gaji	4.000.000
2	Usia	25 Tahun
3	Pendidikan	S1
4	Skill	SK1,SK2,SK3
5	Minat	MN1,MN2.MN3
6	Jurusan	Teknik Informatika (TI)

## b. Spesifikasi Lowongan A (LA)

No	Kriteria	Value
1	Gaji	2.000.000 – 10.000.000
2	Usia	20-30 Tahun
3	Pendidikan	S1
4	Skill	SK1,SK2,SK3,SK4,SK5
5	Minat	MN2.MN3,MN5
6	Jurusan	Teknik Informatika (TI), Sistem Informasi (SI)

## c. Spesifikasi Lowongan B (LB)

No	Kriteria	Value
1	Gaji	1.000.000 – 3.000.000
2	Usia	18-30 Tahun
3	Pendidikan	SMK,SMA
4	Skill	SK3,SK5
5	Minat	MN3.MN7,MN10,
6	Jurusan	IPS, Akuntansi

## d. Spesifikasi Lowongan C (LC)

No	Kriteria	Value
1	Gaji	2.000.000 – 4.500.000
2	Usia	18-30 Tahun
3	Pendidikan	S1
4	Skill	SK1,SK2,SK4
5	Minat	MN2,MN3.MN10
6	Jurusan	Sistem Informasi (SI)

Dari data-data pelamar dan 3 lowongan yang tersedia, yang dilakukan pertama kali adalah menormalisasikan tiap kriteria yang ada.

### 1. Normalisasi Gaji (NG)

Normalisasi gaji dilakukan dengan menghitung berdasarkan batas atas dan bawahnya seperti ini

$$NG = (\text{Value Estimasi Gaji Pelamar} - \text{Batas Bawah Gaji}) / (\text{Batas Atas Gaji} - \text{Batas Bawah Gaji})$$

$$NG(P,LA) = (4.000.000 - 2.000.000) / (10.000.000 - 2.000.000) \\ = 0.25$$

Dengan rumus yang sama didapatkan nilai NG(P,LB) dan NG(P,LC) seperti pada tabel 4.2 berikut

**Tabel 4. 1 Nilai Normalisasi Gaji Pelamar Terhadap Lowongan Kerja**

Normalisasi Gaji	Nilai
NG(P,LA)	0.25
NG(P,LB)	1.5
NG(P,LC)	0.8

### 2. Normalisasi Usia (NU)

Normalisasi usia dilakukan dengan menghitung berdasarkan batas atas dan bawahnya seperti ini

$$NU = (\text{Value Usia Pelamar} - \text{Batas Bawah Usia}) / (\text{Batas Atas Usia} - \text{Batas Bawah Usia})$$

$$NU(P,LA) = (25 - 20) / (30 - 20) = 0.5$$

Dengan rumus yang sama didapatkan nilai NU(P,LB) dan NU(P,LC) seperti pada tabel 4.3 berikut

**Tabel 4. 2 Nilai Normalisasi Usia Pelamar terhadap lowongan kerja**

Normalisasi Usia	Nilai
NU(P,LA)	0.5
NU(P,LB)	0.5833
NU(P,LC)	0.5833

### 3. Normalisasi Minat (NM)

Normalisasi minat dilakukan dengan menghitung berapa jumlah minat pelamar yang diakui oleh lowongan kerja

$NM = (\text{Skill Pelamar yang ada di lowongan} - \text{nilai batas bawah}) / (\text{nilai batas atas} - \text{nilai batas bawah})$

$$NM(P,LA) = (2-1)/(3-1) = 0.5$$

Keterangan

Minat P yang ada pada LA adalah MN2 dan MN3 sehingga minat pelamar yang dihitung adalah 2

Dengan rumus dan analisa yang sama, maka didapatkan nilai untuk NM(P,LB) dan NM(P,LC) seperti pada tabel 4.4 berikut ini

**Tabel 4. 3 Nilai Normalisasi Minat Pelamar terhadap lowongan kerja**

Normalisasi Minat	Nilai
NM(P,LA)	0.5
NM(P,LB)	0
NM(P,LC)	0.5

### 4. Normalisasi Skill (NS)

Normalisasi skill dilakukan dengan menghitung berapa jumlah skill pelamar yang diakui oleh lowongan kerja

$NS = (\text{Skill Pelamar yang ada di lowongan} - \text{nilai batas bawah}) / (\text{nilai batas atas} - \text{nilai batas bawah})$

$$NS(P,LA) = (3-1)/(5-1) = 0.5$$

Keterangan

Skill P yang ada pada LA adalah SK1,SK2 dan SK3 sehingga skill pelamar yang dihitung adalah 3

Dengan rumus dan analisa yang sama, maka didapatkan nilai untuk NS(P,LB) dan NS(P,LC) seperti pada tabel 4.5 berikut ini

**Tabel 4. 4 Nilai Normalisasi Skill Pelamar terhadap lowongan kerja**

Normalisasi Minat	Nilai
NS(P,LA)	0.5
NS(P,LB)	0
NS(P,LC)	0.5

5. Normalisasi Pendidikan (NP)

Normalisasi pendidikan dilihat dari point jenjang pendidikan yang telah dirancang sebelumnya, dapat dilihat pada tabel 4.6 berikut ini

**Tabel 4. 5 Poin jenjang pendidikan**

Jenjang Pendidikan	Poin
SMK/SMA sederajat	1
Sarjana (S1)	2
Magister (S2)	3
Doktoral (S3)	4

Dengan melihat tabel 2.6 maka dapat ditentukan poin untuk normalisasi jenjang pendidikan (NP) dengan rumus seperti berikut

$NP = (\text{Nilai Pendidikan Pelamar} - \text{nilai pendidikan batas bawah}) / (\text{Nilai pendidikan batas atas} - \text{nilai pendidikan batas bawah})$

$$NP(P,LA) = (2-0)/(2-0)=1$$

Dengan rumus dan analisa yang sama, maka didapatkan nilai untuk NP(P,LB) dan NP(P,LC) seperti pada tabel 4.7 berikut ini



**Tabel 4. 6 Nilai Normalisasi Pendidikan Pelamar terhadap lowongan kerja**

Normalisasi Minat	Nilai
NP(P,LA)	1
NP(P,LB)	2
NP(P,LC)	1

#### 6. Normalisasi Jurusan (NJ)

Untuk normalisasi jurusan, apabila jurusan yang ditempuh pelamar sesuai dengan lowongan maka nilainya adalah 1 dan jika tidak sesuai maka nilainya adalah 0. Hasil normalisasi NJ ditampilkan pada tabel 4.8 berikut ini.

**Tabel 4. 7 Nilai Normalisasi Jurusan Pelamar terhadap lowongan kerja**

Normalisasi Minat	Nilai
NJ(P,LA)	1
NJ(P,LB)	0
NJ(P,LC)	1

#### 4.1.3.3 Perhitungan Promethee

Dari normalisasi yang telah dilakukan pada tahap sebelumnya maka akan dapat dibuat sebuah matrik yang memudahkan dalam menghitung promethee lebih jauh. Matrik hasil normalisasi dapat dilihat pada tabel 4.9 berikut ini

**Tabel 4. 8 Matrik Hasil Normalisasi**

Kriteria	LA	LB	LC
GJ	0.25	1.5	0.8
US	0.5	0.5833	0.5833
MN	0.5	0	0.5
SK	0.5	0	0.5

PD	1	2	1
JU	1	0	1

Dari matrik diatas selanjutnya adalah menghitung nilai preferensi berpasangan tiap kriteria yang ada

1. Preferensi Gaji (PG) (Min)

- a.  $PG(LA, LB) = \text{Nilai Gaji LB} - \text{Nilai Gaji LA} = 1.5 - 0.25 = 1.25$
- b.  $PG(LA, LC) = \text{Nilai Gaji LC} - \text{Nilai Gaji LA} = 0.55$
- c.  $PG(LB, LA) = -1.25$
- d.  $PG(LB, LC) = -0.7$
- e.  $PG(LC, LA) = -0.55$
- f.  $PG(LC, LB) = 0.7$

2. Preferensi Usia (PU) (Min)

- a.  $PU(LA, LB) = \text{Nilai Usia LB} - \text{Nilai Usia LA} = 0.5833 - 0.5 = 0.0833$
- b.  $PU(LA, LC) = 0.0833$
- c.  $PU(LB, LA) = -0.0833$
- d.  $PU(LB, LC) = 0$
- e.  $PU(LC, LA) = -0.0833$
- f.  $PU(LC, LB) = 0$

3. Preferensi Minat (PM) (Maks)

- a.  $PM(LA, LB) = \text{Nilai Minat LA} - \text{Nilai Minat LB} = 0.5 - 0 = 0.5$
- b.  $PM(LA, LC) = 0$
- c.  $PM(LB, LA) = -0.5$
- d.  $PM(LB, LC) = -0.5$
- e.  $PM(LC, LA) = 0$
- f.  $PM(LC, LB) = 0.5$

4. Preferensi Skill (PS) (Maks)

- a.  $PS(LA, LB) = \text{Nilai Skill LA} - \text{Nilai Skill LB} = 0.5 - 0 = 0.5$

- b.  $PS(LA,LC) = 0$
  - c.  $PS(LB,LA) = -0.5$
  - d.  $PS(LB,LC) = -0.5$
  - e.  $PS(LC,LA) = 0$
  - f.  $PS(LC,LB) = 0.5$
5. Preferensi Pendidikan (PP) (Maks)
- a.  $PP(LA,LB) = \text{Nilai Pendidikan LA} - \text{Nilai Pendidikan LB} = 1 - 2 = -1$
  - b.  $PP(LA,LC) = 0$
  - c.  $PP(LB,LA) = 1$
  - d.  $PP(LB,LC) = 1$
  - e.  $PP(LC,LA) = 0$
  - f.  $PP(LC,LB) = -1$
6. Preferensi Jurusan (PJ) (Maks)
- a.  $PJ(LA,LB) = \text{Nilai Jurusan LA} - \text{Nilai Jurusan LB} = 1 - 0 = 1$
  - b.  $PJ(LA,LC) = 0$
  - c.  $PJ(LB,LA) = -1$
  - d.  $PJ(LB,LC) = -1$
  - e.  $PJ(LC,LA) = 0$
  - f.  $PJ(LC,LB) = 1$

Dari perhitungan preferensi berpasangan tersebut selanjutnya akan dijumlahkan untuk semua kriterianya menjadi total preferensi (TP)

$$TP(LA,LB) = PG(LA,LB) + PU(LA,LB) + PM(LA,LB) + PS(LA,LB) + PP(LA,LB) + PJ(LA,LB) = 1.25 + 0.00833 + 0.5 + 0.5 + (-1) + 1 = 2,3333$$

$$TP(LA,LC) = 0.6333$$

$$TP(LB,LA) = -2.3333$$

$$TP(LB,LC) = -1.7$$

$$TP(LC,LA) = -0.6333$$

$$TP(LC,LB) = 1.7$$

Dari total preferensi yang sudah dihitung tersebut lalu dibuatlah sebuah matrik total preferensi untuk lowongan untuk mempermudah untuk memvisualisasikan hasil perhitungan. Matrik total preferensi dapat dilihat pada tabel 4.10 berikut ini

**Tabel 4. 9 Matriks Total Preferensi Tiap Lowongan**

TP	LA	LB	LC
LA	0	2.3333	0.6333
LB	-2.3333	0	-1.7
LC	-0.6333	1.7	0

#### 4.1.3.4 Perangkingan Lowongan

Untuk membuat perangkingan dari kasus diatas maka yang harus dilakukan adalah mencari nilai *Entering Flow (EF)*, *Leaving Flow* dan *Net Flow* dari matrik total preferensi ditabel 2.10 tersebut

1. *Entering Flow* LA =  $1/(\text{total lowongan} - 1) * (\text{TP(LA,LA)} + \text{TP(LB,LA)} + \text{TP(LC,LA)}) = 1/(3-1) * (0 + (-2.3333) + (-0.6333)) = -1.4833$
2. *Entering Flow* LB = 2.0167
3. *Entering Flow* LC = -0.5333
4. *Leaving Flow* LA =  $1/(\text{total lowongan} - 1) * (\text{TP(LA,LA)} + \text{TP(LA,LB)} + \text{TP(LA,LC)}) = 1/(3-1) * (0 + 2.3333 + 0.6333) = 1.4833$
5. *Leaving Flow* LB = -2.0167
6. *Leaving Flow* LC = 0.5333

$$7. \text{ Net Flow LA} = \text{Leaving Flow LA} - \text{Entering Flow LA} = 1.4833 - (-1.4833) = 2.966$$

$$8. \text{ Net Flow LB} = -4.033$$

$$9. \text{ Net Flow LC} = 1.0677$$

Dari perhitungan *Net Flow* LA, LB dan LC didapatkan bahwa LA memiliki nilai yang terbesar, maka sistem akan mengurutkan rekomendasi lowongan kepada pelamar dari yang *Net Flow* terbesar ke *Net Flow* terkecil yaitu LA, LC, LB.

Untuk mencari persentase kecocokan, digunakanlah normalisasi nilai *Net Flow* seperti berikut :

Nilai Terendah: -4.033

Nilai Tertinggi: 2.966

Rentang Nilai:  $2.966 - (-4.033) = 6.999$

Persentase LA =  $2.996 - (-4.033) / 6.999 * 100\% = 100\%$

Persentase LB =  $-4.033 - (-4.033) / 6.999 * 100\% = 0\%$

Persentase LC =  $1,0677 - (-4,033) / 6,999 * 100\% = 72,9 \%$

#### **4.1.4 Pengkodean**

Setelah membuat rancangan database, langkah selanjutnya adalah melakukan tahapan pengkodean untuk membangun Aplikasi Kesesuaian Lowongan Kerja. Pada tahap pengkodean ini penulis menggunakan bahasa pemrograman PHP dan dengan bantuan *template Admin LTE* untuk mempercepat proses pengkodean. Berikut ini tampilan sistem yang sudah penulis buat

##### **4.1.4.1 Halaman Register**

Halaman awal merupakan halaman yang digunakan untuk pelamar baru mendaftar agar bisa memakai aplikasi lowongan kerja. Halaman ini dapat dilihat pada gambar 4.1 berikut ini

**Gambar 4. 1 Halaman Register**

Pelamar baru mengisi form register secara lengkap. Textbox nama lengkap diisi dengan nama lengkap pelamar, textbox nik diisi dengan NIK pelamar 16 digit dan nantinya NIK ini akan digunakan untuk namauser saat login, email yang diinputkan adalah email aktif, password yang dimasukkan harus sama antara password dan kolom retype password. Setelah semua inputan terisi, pelamar dapat menekan tombol register untuk menyimpan data. Saat proses registrasi ini dilakukan, maka status pelamar adalah belum terverifikasi, dan akan diubah statusnya menjadi verified oleh admin. Algoritma dapat dilihat pada tabel 4.11 dibawah

**Tabel 4. 10 Algoritma Simpan Registrasi**

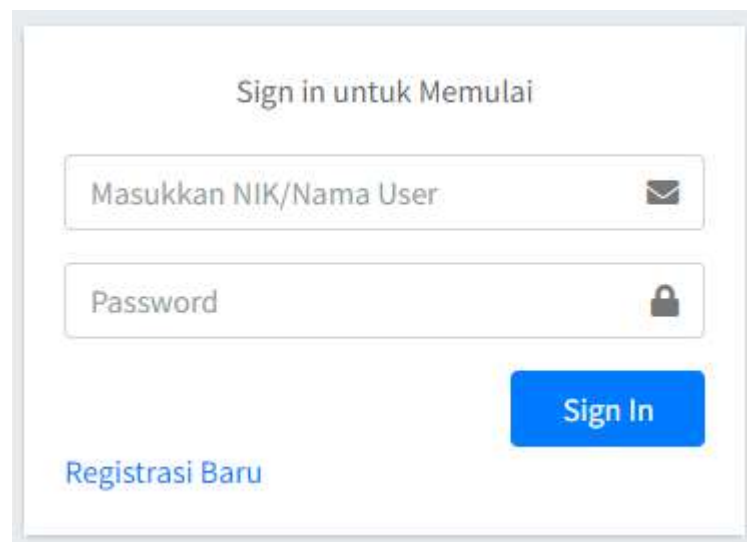
Keterangan		Code
Algoritma Registrasi	Simpan	<ol style="list-style-type: none"> <li>1. Mulai fungsi simpan.</li> <li>2. Ambil input NIK, Email, Nama Lengkap, Password, dan Ulangi Password.</li> <li>3. Lakukan pencarian di database untuk mencari jumlah baris dengan NIK yang sama.</li> </ol>

	<ol style="list-style-type: none"><li>4. Lakukan pencarian di database untuk mencari jumlah baris dengan Email yang sama.</li><li>5. Lakukan pemeriksaan apakah NIK sudah terdaftar. Jika ya, tampilkan pesan "NIK sudah Terdaftar !!" dan redirect ke halaman register.</li><li>6. Lakukan pemeriksaan apakah Email sudah terdaftar. Jika ya, tampilkan pesan "Email sudah Terdaftar !!" dan redirect ke halaman register.</li><li>7. Lakukan pemeriksaan apakah password sama dengan ulangi password. Jika tidak, tampilkan pesan "Password Tidak Match !!" dan redirect ke halaman register.</li><li>8. Jika semua pemeriksaan di atas berhasil, siapkan data untuk disimpan:<ul style="list-style-type: none"><li>• Email</li><li>• NIK</li><li>• Nama Lengkap</li><li>• Password yang telah di-hash</li><li>• Status Pelamar</li></ul></li><li>9. Mulai transaksi database.</li><li>10. Simpan data yang telah disiapkan ke dalam tabel 'tb_pelamar'.</li><li>11. Jika transaksi berhasil:<ul style="list-style-type: none"><li>• Commit transaksi.</li><li>• Tampilkan pesan "Data Berhasil disimpan !!".</li><li>• Redirect ke halaman login.</li></ul></li></ol>
--	--

	<p>12. Jika transaksi gagal:</p> <ul style="list-style-type: none"> <li>• Rollback transaksi.</li> <li>• Tampilkan pesan "Data Gagal disimpan !!".</li> <li>• Redirect ke halaman register.</li> </ul>
--	--

#### 4.1.4.2 Halaman Login

Halaman ini digunakan oleh admin dan juga pelamar yang akan menggunakan aplikasi lowongan kerja. Gambar halaman ini dapat dilihat pada gambar 4.2 berikut ini



**Gambar 4. 2 Halaman Login**

Pelamar memasukkan nik dan password yang benar untuk bisa login kedalam aplikasi. Nantinya akan diarahkan ke halaman dashboard apabila berhasil login. Algoritma login dapat dilihat pada tabel 4.12 berikut ini



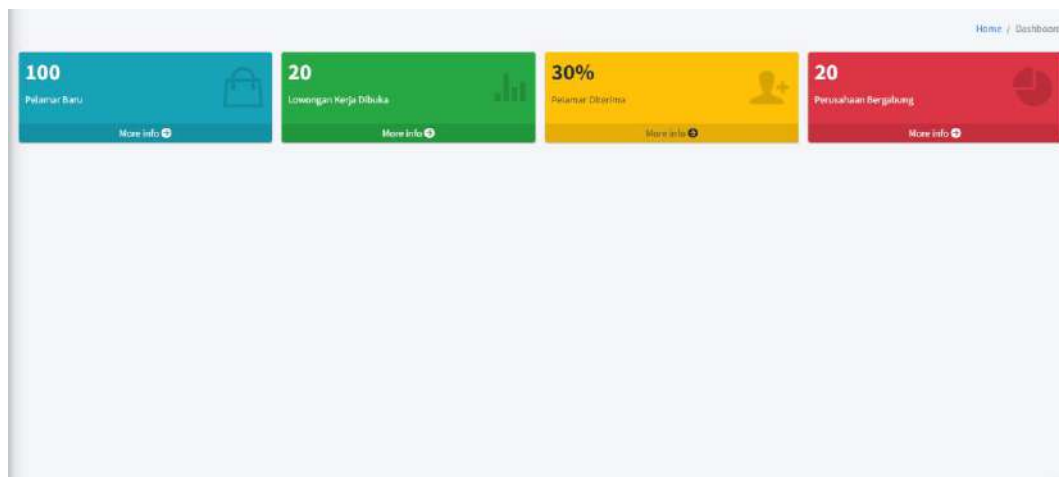
**Tabel 4. 11 Algoritma Login**

Keterangan	Code
Proses Login	<ol style="list-style-type: none"> <li>1. Mulai fungsi signin.</li> <li>2. Ambil input NIK dan Password dari pengguna.</li> <li>3. Lakukan pencarian di database untuk mencari baris dengan NIK dan Password yang sesuai di tabel 'tb_user' untuk login admin.</li> <li>4. Lakukan pencarian di database untuk mencari baris dengan NIK dan Password yang sesuai di tabel 'tb_pelamar' untuk login peserta.</li> <li>5. Jika jumlah baris hasil pencarian untuk login admin lebih dari atau sama dengan 1, maka: <ul style="list-style-type: none"> <li>• Ambil data dari hasil query cekLoginAdmin.</li> <li>• Siapkan data untuk disimpan dalam session, termasuk 'logged_in' dengan nilai 'sukses', 'is_login' dengan nilai 'true', nama admin, peran admin, dan id admin.</li> <li>• Set session dengan data tersebut.</li> <li>• Tampilkan pesan "Login Sukses !!".</li> <li>• Redirect ke halaman dashboard.</li> </ul> </li> <li>6. Jika jumlah baris hasil pencarian untuk login peserta lebih dari atau sama dengan 1, maka: <ul style="list-style-type: none"> <li>• Ambil data dari hasil query cekLogin.</li> <li>• Siapkan data untuk disimpan dalam session, termasuk 'logged_in' dengan nilai 'sukses', 'is_login' dengan nilai 'true', nama peserta, peran peserta (diasumsikan sebagai 'Peserta'), dan id peserta.</li> <li>• Set session dengan data tersebut.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>• Tampilkan pesan "Login Sukses !!".</li> <li>• Redirect ke halaman dashboard.</li> </ul> <p>7. Jika tidak ada hasil pencarian yang sesuai, tampilkan pesan "NIK Atau Password Salah !!" dan redirect ke halaman login.</p>
--	--

#### 4.1.4.3 Halaman Dashboard

Halaman ini menampilkan informasi tentang summary aplikasi, antara lain jumlah pelamar, jumlah lowongan dibuka, informasi jumlah perusahaan yang bergabung dan informasi persentase pelamar yang diterima. Gambar halaman dashboard dapat dilihat pada gambar 4.3 berikut ini



**Gambar 4. 3 Halaman Dashboard**

#### 4.1.4.4 Halaman Data Diri Pelamar

Halaman ini digunakan oleh pelamar untuk mengedit profil dirinya, dan juga estimasi gaji yang diinginkannya. Gambar data diri pelamar dapat dilihat pada gambar 4.4 berikut ini

**Gambar 4. 4 Halaman Data Diri Pelamar**

Pelamar dapat mengubah data diri ini setelah statusnya diubah menjadi verified oleh admin. Biodata ini wajib dilengkapi oleh pelamar, karena informasi dari halaman ini akan menjadi dasar perhitungan nilai dengan metode promethee. Algoritma terkait halaman ini dapat dilihat pada tabel 4.13 berikut ini

**Tabel 4. 12 Algoritma Halaman Data Diri**

Keterangan	Code
Proses Upload Dokumen	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>upload_data</b>.</li> <li>2. Siapkan variabel <b>data</b> untuk menyimpan hasil upload.</li> <li>3. Tentukan jenis file yang diizinkan (<b>allowed_types</b>), lokasi penyimpanan upload (<b>upload_path</b>), dan batasan ukuran file (<b>max_size</b>).</li> <li>4. Inisialisasi konfigurasi upload dengan nilai-nilai yang telah ditentukan.</li> <li>5. Lakukan proses upload dengan konfigurasi yang telah ditentukan.</li> <li>6. Jika proses upload file gagal:</li> </ol>

	<ul style="list-style-type: none"> <li>• Simpan pesan error upload dalam variabel <code>data['error ' + <code>namaFile</code>]</code>.</li> <li>• Set <code>data['pesan']</code> dengan pesan yang menyatakan bahwa proses upload gagal beserta pesan error upload.</li> </ul> <p>7. Jika proses upload berhasil:</p> <ul style="list-style-type: none"> <li>• Simpan data dari hasil upload dalam variabel <code>hasil</code>.</li> <li>• Set <code>data['upload_data']</code> dengan data dari hasil upload.</li> <li>• Set <code>data['pesan']</code> dengan pesan 'sukses'.</li> <li>• Set <code>data['nama_file']</code> dengan nama file dari hasil upload.</li> </ul> <p>8. Kembalikan variabel <code>data</code> yang berisi hasil upload.</p>
Proses update data	<ol style="list-style-type: none"> <li>1. Mulai fungsi <code>`ubah_data`</code>.</li> <li>2. Ambil input nomor HP, jenis kelamin, lama kerja, gaji, tanggal lahir, id, skill, jenjang pendidikan, dan jurusan.</li> <li>3. Ambil input minat dan cek apakah input skill dan minat tidak kosong serta merupakan array: <ul style="list-style-type: none"> <li>- Jika iya, gabungkan nilai-nilai skill dan minat menjadi satu string.</li> </ul> </li> <li>4. Cek apakah file foto diunggah: <ul style="list-style-type: none"> <li>- Jika tidak, siapkan data untuk disimpan dalam variabel <code>`data_simpan`</code>.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>- Mulai transaksi database.</li> <li>- Tentukan kondisi WHERE untuk update data berdasarkan id.</li> <li>- Lakukan update data_simpan ke dalam tabel 'tb_pelamar' sesuai dengan id.</li> <li>- Jika transaksi berhasil, commit transaksi, tampilkan pesan "Data Berhasil Diubah !", dan redirect ke halaman pelamar.</li> <li>- Jika transaksi gagal, rollback transaksi, tampilkan pesan "Data Gagal Diubah !", dan redirect ke halaman pelamar.</li> </ul> <p>5. Jika file foto diunggah:</p> <ul style="list-style-type: none"> <li>- Lakukan upload foto menggunakan fungsi `upload_data`.</li> <li>- Jika upload foto berhasil, siapkan data untuk disimpan dalam variabel `data_simpan` dengan tambahan nama file hasil upload.</li> <li>- Mulai transaksi database.</li> <li>- Tentukan kondisi WHERE untuk update data berdasarkan id.</li> <li>- Lakukan update data_simpan ke dalam tabel 'tb_pelamar' sesuai dengan id.</li> <li>- Jika transaksi berhasil, commit transaksi, tampilkan pesan "Data Berhasil Diubah !", dan redirect ke halaman pelamar.</li> <li>- Jika transaksi gagal, rollback transaksi, tampilkan pesan "Data Gagal Diubah !", dan redirect ke halaman pelamar.</li> <li>- Jika upload foto gagal, tampilkan pesan "Foto Gagal Diupload !" beserta pesan error upload, dan redirect ke halaman pelamar.</li> </ul>
--	--

#### 4.1.4.5 Halaman Pendidikan Pelamar

Halaman ini digunakan pelamar untuk menambahkan atau mengubah data pendidikan yang telah ditempuh oleh pelamar. Gambar halaman pendidikan pelamar dapat dilihat pada gambar 4.5 berikut ini

No	Jenjang Pendidikan	Nama Institusi	Tahun Masuk	Tahun Selesai	Jurusan	File	Aksi
1.	Sarjana (S1)	IB Darmajaya	2019	2024	S1 Sistem Informasi	Download	Hapus

**Gambar 4. 5 Halaman Pendidikan Pelamar**

Pelamar menambahkan semua riwayat pendidikan yang telah ditempuh olehnya, dengan halaman ini, HRD akan bisa melihat riwayat pendidikan meskipun hard copynya tidak dikirimkan. Algoritma pada halaman pendidikan pelamar dapat dilihat pada tabel 2.14 berikut ini

**Tabel 4. 13 Algoritma Halaman Riwayat Pendidikan**

Keterangan	Code
Simpan riwayat pendidikan	<ol style="list-style-type: none"> <li>1. Mulai fungsi `tambah_data`.</li> <li>2. Ambil input id_pelamar, jenjang pendidikan, nama sekolah, tahun mulai, tahun selesai, dan id jurusan.</li> <li>3. Periksa apakah file pendidikan diunggah: <ul style="list-style-type: none"> <li>- Jika tidak, tampilkan pesan "File Upload Kosong !" dan redirect ke halaman pendidikan.</li> </ul> </li> <li>4. Jika file pendidikan diunggah:</li> </ol>

	<ul style="list-style-type: none"> <li>- Lakukan upload file pendidikan menggunakan fungsi `upload_data`.</li> <li>- Jika upload file berhasil: <ul style="list-style-type: none"> <li>- Siapkan data untuk disimpan dalam variabel `data_simpan` dengan tambahan nama file dari hasil upload.</li> <li>- Mulai transaksi database.</li> <li>- Simpan data_simpan ke dalam tabel 'tb_pendidikan'.</li> <li>- Jika transaksi berhasil, commit transaksi, tampilkan pesan "Data Berhasil Disimpan !", dan redirect ke halaman pendidikan.</li> <li>- Jika transaksi gagal, rollback transaksi, tampilkan pesan "Data Gagal Disimpan !", dan redirect ke halaman pendidikan.</li> <li>- Jika upload file gagal, tampilkan pesan "Foto Gagal Diupload !" beserta pesan error upload, dan redirect ke halaman pendidikan.</li> </ul> </li> </ul>
--	---

#### **4.1.4.6 Halaman Rekomendasi Lowongan Kerja**

Halaman ini nantinya akan menampilkan 5 rekomendasi lowongan kerja yang sesuai dengan kondisi pelamar dan juga akan menampilkan semua lowongan yang dibuka hingga waktu tertentu. Di halaman inilah proses promethee dilakukan. Gambar halaman rekomendasi lowongan kerja dapat dilihat pada halaman 4.6 berikut ini

The screenshot displays a web application interface for job recommendations. It is titled "Data Rekomendasi Pekerjaan 3 Terbaik" and includes a navigation bar with "Home" and "Rekomendasi Pekerjaan". The main content area is divided into three columns for job categories: "Jaringan Komputer", "Full Stack Developer", and "Akuntansi". Each category has a representative image and a "View Detail" button. On the right side, there are several data tables showing preference matrices and scores for different job types (PNS, ASN, PNS) across various skill categories.

**Gambar 4. 6 Halaman Rekomendasi Lowongan Kerja**

Pelamar dapat melakukan apply pada lowongan kerja yang tersedia. Jika rekomendasi lowongan kerja dirasa tidak memuaskan atau pelamar ingin memilih lowongan lain maka hal tersebut bisa dilakukan. Algoritma pada halaman rekomendasi dapat dilihat pada tabel 4.15 berikut ini

**Tabel 4. 14 Algoritma Halaman Rekomendasi Lowongan**

Keterangan	Code
Normalisasi skill	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>normalisasiSkillDiakui</b> dengan parameter \$id_skill_pelamar dan \$id_skill_lowongan.</li> <li>2. Hitung nilai maksimum (\$max) sebagai jumlah elemen dalam array \$id_skill_lowongan.</li> <li>3. Tentukan elemen-elemen yang sama antara \$id_skill_pelamar dan \$id_skill_lowongan dan simpan dalam variabel \$anggota_sama.</li> <li>4. Hitung jumlah elemen dalam \$anggota_sama dan simpan dalam variabel \$jumlah_anggota_sama.</li> </ol>



	<p>5. Jika nilai <math>\\$max</math> sama dengan 1, maka:</p> <ul style="list-style-type: none"> <li>• Kembalikan nilai 0 karena tidak ada keterampilan yang diakui.</li> </ul> <p>6. Selain itu:</p> <ul style="list-style-type: none"> <li>• Kembalikan nilai dari rumus normalisasi, yaitu <math>(\\$jumlah\_anggota\_sama - 1)</math> dibagi oleh <math>(\\$max - 1)</math>, yang menunjukkan seberapa banyak keterampilan yang diakui oleh lowongan pekerjaan dibandingkan dengan total keterampilan yang dibutuhkan.</li> </ul>
Normalisasi jurusan	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>normalisasiJurusanDiakui</b> dengan parameter <math>\\$id\_jurusan\_pelamar</math> dan <math>\\$id\_jurusan\_lowongan</math>.</li> <li>2. Tentukan elemen-elemen yang sama antara <math>\\$id\_jurusan\_pelamar</math> dan <math>\\$id\_jurusan\_lowongan</math> dan simpan dalam variabel <math>\\$anggota\_sama</math>.</li> <li>3. Hitung jumlah elemen dalam <math>\\$anggota\_sama</math> dan simpan dalam variabel <math>\\$jumlah\_anggota\_sama</math>.</li> <li>4. Kembalikan nilai <math>\\$jumlah\_anggota\_sama</math>.</li> </ol>
Normalisasi minat	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>normalisasiMinatDiakui</b> dengan parameter <math>\\$id\_minat\_pelamar</math> dan <math>\\$id\_minat\_lowongan</math>.</li> <li>2. Hitung nilai maksimum (<math>\\$max</math>) sebagai jumlah elemen dalam array <math>\\$id\_minat\_lowongan</math>.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Tentukan elemen-elemen yang sama antara \$id_minat_pelamar dan \$id_minat_lowongan dan simpan dalam variabel \$anggota_sama.</li> <li>4. Hitung jumlah elemen dalam \$anggota_sama dan simpan dalam variabel \$jumlah_anggota_sama.</li> <li>5. Jika nilai \$max sama dengan 1, maka: <ul style="list-style-type: none"> <li>• Kembalikan nilai 0 karena tidak ada minat yang diakui.</li> </ul> </li> <li>6. Selain itu: <ul style="list-style-type: none"> <li>• Kembalikan nilai dari rumus normalisasi, yaitu <math>(\\$jumlah\_anggota\_sama - 1)</math> dibagi oleh <math>(\\$max - 1)</math>, yang menunjukkan seberapa banyak minat yang diakui oleh lowongan pekerjaan dibandingkan dengan total minat yang dibutuhkan.</li> </ul> </li> </ol>
Normalisasi umur	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>normalisasiUmur</b> dengan parameter \$nilai, \$min, dan \$max.</li> <li>2. Periksa apakah nilai maksimum (\$max) sama dengan nilai minimum (\$min): <ul style="list-style-type: none"> <li>• Jika iya, kembalikan nilai 0 karena rentang umur tidak dapat dihitung.</li> </ul> </li> <li>3. Selain itu, kembalikan nilai dari rumus normalisasi umur, yaitu <math>(\\$nilai - \\$min)</math> dibagi oleh <math>(\\$max - \\$min)</math>, yang menunjukkan seberapa jauh nilai umur berada di dalam rentang dari nilai minimum hingga nilai maksimum.</li> </ol>

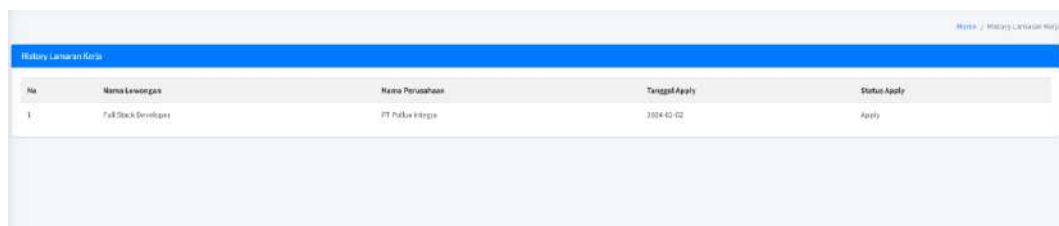
<p>Hitung Preferensi berpasangan</p>	<ol style="list-style-type: none"> <li>1. Mulai fungsi <b>hitungPreferensi</b> dengan parameter \$lowongan.</li> <li>2. Inisialisasi dictionary kosong untuk menyimpan preferensi.</li> <li>3. Untuk setiap lowongan A dalam daftar lowongan:             <ol style="list-style-type: none"> <li>a. Untuk setiap lowongan B dalam daftar lowongan:                 <ul style="list-style-type: none"> <li>- Periksa apakah nama_lowongan_A tidak sama dengan nama_lowongan_B:                     <ul style="list-style-type: none"> <li>- Jika iya, hitung preferensi relatif untuk setiap kriteria:                         <ul style="list-style-type: none"> <li>- Preferensi umur (min): umur_B - umur_A</li> <li>- Preferensi gaji (min): gaji_B - gaji_A</li> <li>- Preferensi minat (maks): minat_A - minat_B</li> <li>- Preferensi skill (maks): skill_A - skill_B</li> <li>- Preferensi jurusan (maks): jurusan_A - jurusan_B</li> <li>- Preferensi pendidikan (maks): pendidikan_A - pendidikan_B</li> </ul> </li> <li>- Jumlahkan preferensi relatif dari setiap kriteria untuk mendapatkan total preferensi_A_B.</li> <li>- Simpan total_preferensi_A_B untuk pasangan lowongan dalam dictionary preferensi dengan kunci "\$nama_lowongan_A - \$nama_lowongan_B".</li> <li>- Jika nama_lowongan_A sama dengan nama_lowongan_B, beri nilai 0 pada preferensi untuk pasangan lowongan yang sama.</li> </ul> </li> </ul> </li> </ol> </li> <li>4. Kembalikan dictionary preferensi yang berisi nilai preferensi relatif untuk setiap pasangan lowongan.</li> </ol>

<p>Proses rekomendasi lowongan kerja</p>	<ol style="list-style-type: none"> <li>1. Mendapatkan data lowongan yang masih dibuka dari database.</li> <li>2. Mendapatkan data pelamar berdasarkan ID pelamar yang terdapat dalam session pengguna.</li> <li>3. Mendapatkan kriteria pelamar seperti skill, minat, gaji, pendidikan, umur, dan jurusan.</li> <li>4. Membagi string skill, minat, dan jurusan pelamar menjadi array.</li> <li>5. Membuat array untuk menyimpan data kriteria pelamar.</li> <li>6. Inisialisasi array kosong untuk menyimpan data preferensi lowongan.</li> <li>7. Looping untuk setiap lowongan: a. Mendapatkan kriteria lowongan seperti skill, minat, jurusan, gaji, usia minimum, usia maksimum, dan pendidikan. b. Membagi string skill, minat, dan jurusan lowongan menjadi array. c. Normalisasi kriteria lowongan berdasarkan kriteria pelamar. d. Menyimpan data preferensi lowongan.</li> <li>8. Menghitung preferensi lowongan dengan memanggil fungsi hitungPreferensi.</li> <li>9. Mendapatkan nama-nama lowongan.</li> <li>10. Membuat matriks preferensi lowongan.</li> <li>11. Menghitung total jumlah lowongan.</li> <li>12. Menghitung matriks entering flow dan leaving flow.</li> <li>13. Mengurutkan matriks net flow dari yang terbesar ke yang terkecil.</li> </ol>

14. Membuat list rekomendasi berdasarkan matriks net flow.

#### 4.1.4.7 Halaman Histori Lamaran

Halaman ini digunakan untuk menampilkan lamaran kerja yang telah dilamar oleh pelamar, baik itu diterima kerja maupun ditolak. Gambar histori lamaran dapat dilihat pada gambar 4.7 berikut ini



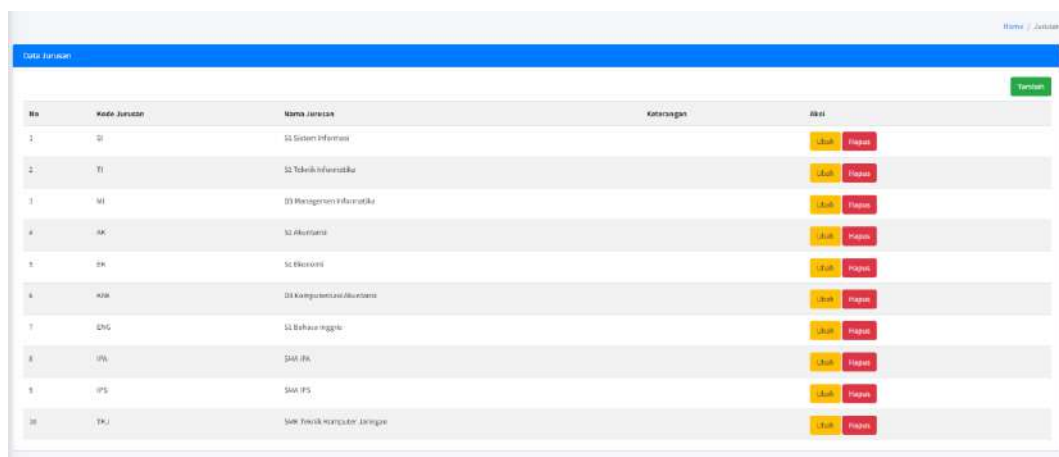
No	Nama Lowongan	Nama Perusahaan	Tanggal Apply	Status Apply
1	Full Stack Developer	PT Pallas Integrate	2024-02-02	Apply

**Gambar 4. 7 Halaman Histori Lamaran**

Dari halaman ini, pelamar dapat melihat status dari lamaran yang dilakukan, apabila ada perubahan status maka diharapkan pelamar dapat segera melakukan follow up.

#### 4.1.4.8 Halaman Jurusan

Halaman ini digunakan oleh admin untuk mengelola data jurusan yang umum ada di Indonesia. Gambar halaman jurusan dapat dilihat pada gambar 4.8 berikut ini



No	Kode Jurusan	Nama Jurusan	Keterangan	Aksi
1	SI	SI Sistem Informatika		Lamar Hapus
2	TI	SI Teknik Informatika		Lamar Hapus
3	MI	DI Manajemen Informatika		Lamar Hapus
4	IK	SI Informatika		Lamar Hapus
5	IK	SI Elektronika		Lamar Hapus
6	IK	DI Komputerisasi Akuntansi		Lamar Hapus
7	ENG	SI Bahasa Inggris		Lamar Hapus
8	IPA	SM IPA		Lamar Hapus
9	IPS	SM IPS		Lamar Hapus
10	TKJ	SM Teknik Komputer Jaringan		Lamar Hapus

**Gambar 4. 8 Halaman Jurusan**

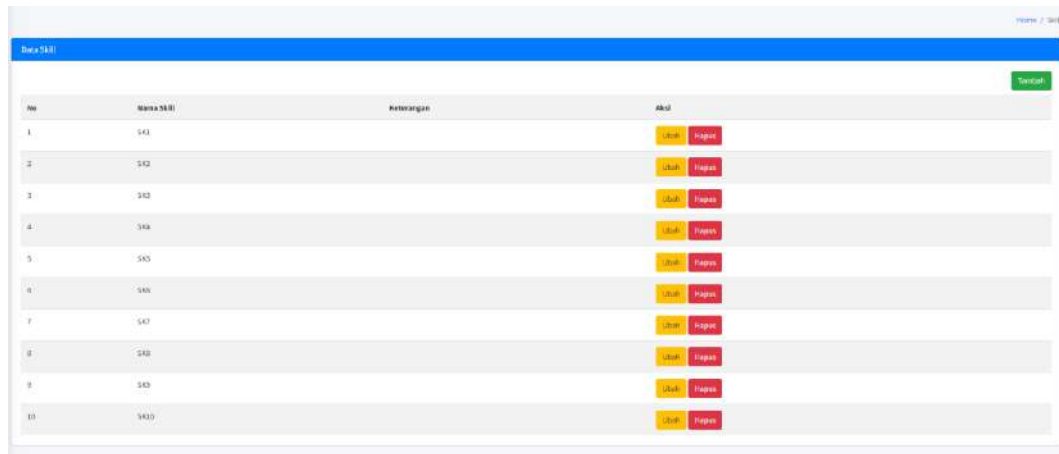
Algoritma pada halaman jurusan dapat dilihat pada tabel 4.16 berikut ini

**Tabel 4. 15 Algoritma Halaman Jurusan**

Keterangan	Code
Proses tambah data	<ol style="list-style-type: none"> <li>1. Mendapatkan data dari formulir, yaitu kode jurusan, nama jurusan, dan keterangan.</li> <li>2. Menyiapkan data yang akan disimpan ke dalam database dalam bentuk array asosiatif dengan kunci sebagai nama kolom di tabel dan nilai sebagai data yang diambil dari formulir.</li> <li>3. Memulai transaksi dengan database untuk memastikan operasi penambahan data jurusan dilakukan secara atomik.</li> <li>4. Melakukan operasi penyimpanan data ke dalam tabel <code>tb_jurusan</code>.</li> <li>5. Memeriksa status transaksi, jika gagal (FALSE), maka melakukan rollback transaksi untuk membatalkan operasi yang sudah dilakukan dan menampilkan pesan kesalahan.</li> <li>6. Jika berhasil, melakukan commit transaksi untuk menyimpan perubahan ke dalam database secara permanen dan menampilkan pesan keberhasilan.</li> </ol>

#### 4.1.4.9 Halaman Skill

Halaman ini digunakan oleh admin untuk mengelola data skill yang sesuai dengan lowongan kerja. Gambar halaman dapat dilihat pada gambar 4.9 berikut ini



No	Nama Skill	Keterangan	Aksi
1	SK1		Tambah Hapus
2	SK2		Tambah Hapus
3	SK3		Tambah Hapus
4	SK4		Tambah Hapus
5	SK5		Tambah Hapus
6	SK6		Tambah Hapus
7	SK7		Tambah Hapus
8	SK8		Tambah Hapus
9	SK9		Tambah Hapus
10	SK10		Tambah Hapus

**Gambar 4. 9 Halaman Skill**

Algoritma pada halaman skill dapat dilihat pada tabel 4.17 berikut ini

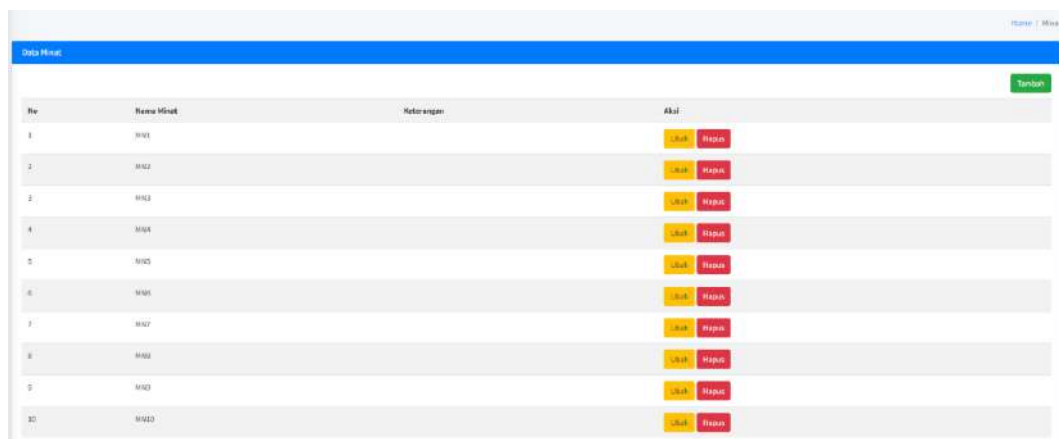
**Tabel 4. 16 Algoritma Halaman Skill**

Keterangan	Code
Proses Tambah Data	<ol style="list-style-type: none"> <li>1. Mendapatkan data dari formulir, yaitu nama skill dan keterangan.</li> <li>2. Menyiapkan data yang akan disimpan ke dalam database dalam bentuk array asosiatif dengan kunci sebagai nama kolom di tabel dan nilai sebagai data yang diambil dari formulir.</li> <li>3. Memulai transaksi dengan database untuk memastikan operasi penambahan data skill dilakukan secara atomik.</li> <li>4. Melakukan operasi penyimpanan data ke dalam tabel <code>tb_skill</code>.</li> <li>5. Memeriksa status transaksi, jika gagal (FALSE), maka melakukan rollback transaksi</li> </ol>

	<p>untuk membatalkan operasi yang sudah dilakukan dan menampilkan pesan kesalahan.</p> <p>6. Jika berhasil, melakukan commit transaksi untuk menyimpan perubahan ke dalam database secara permanen dan menampilkan pesan keberhasilan.</p>
--	--

#### 4.1.4.10 Halaman Minat

Halaman ini digunakan oleh admin untuk mengelola data minat yang sesuai dengan lowongan kerja secara umum. Gambar halaman minat dapat dilihat pada gambar 4.10 berikut ini



No	Nama Minat	Keterangan	Aksi
1	MI01		<a href="#">Edit</a> <a href="#">Hapus</a>
2	MI02		<a href="#">Edit</a> <a href="#">Hapus</a>
3	MI03		<a href="#">Edit</a> <a href="#">Hapus</a>
4	MI04		<a href="#">Edit</a> <a href="#">Hapus</a>
5	MI05		<a href="#">Edit</a> <a href="#">Hapus</a>
6	MI06		<a href="#">Edit</a> <a href="#">Hapus</a>
7	MI07		<a href="#">Edit</a> <a href="#">Hapus</a>
8	MI08		<a href="#">Edit</a> <a href="#">Hapus</a>
9	MI09		<a href="#">Edit</a> <a href="#">Hapus</a>
10	MI10		<a href="#">Edit</a> <a href="#">Hapus</a>

**Gambar 4. 10 Halaman Minat**

Algoritma pada halaman minat dapat dilihat pada tabel 4.18 berikut ini



Tabel 4. 17 Algoritma Halaman Minat

Keterangan	Code
Proses Tambah Data	<ol style="list-style-type: none"> <li>1. Mendapatkan data dari formulir, yaitu nama minat dan keterangan.</li> <li>2. Menyiapkan data yang akan disimpan ke dalam database dalam bentuk array asosiatif dengan kunci sebagai nama kolom di tabel dan nilai sebagai data yang diambil dari formulir.</li> <li>3. Memulai transaksi dengan database untuk memastikan operasi penambahan data minat dilakukan secara atomik.</li> <li>4. Melakukan operasi penyimpanan data ke dalam tabel <b>tb_minat</b>.</li> <li>5. Memeriksa status transaksi, jika gagal (FALSE), maka melakukan rollback transaksi untuk membatalkan operasi yang sudah dilakukan dan menampilkan pesan kesalahan.</li> <li>6. Jika berhasil, melakukan commit transaksi untuk menyimpan perubahan ke dalam database secara permanen dan menampilkan pesan keberhasilan.</li> </ol>

#### 4.1.4.11 Halaman Perusahaan

Halaman ini digunakan oleh admin untuk mengelola data perusahaan yang nantinya akan memberikan lowongan kerja. Gambah halaman perusahaan dapat dilihat pada gambar 4.11 berikut ini

No	Jenis perusahaan	Nama perusahaan	Keterangan	No Telp	Alamat	Aksi
1	Software House	PT Pudu Integri		08190605222	Jakarta	<span>Tambah</span> <span>Hapus</span>
2	Transportasi	Tiki		877986	Lampung	<span>Tambah</span> <span>Hapus</span>
3	Hassanah	Hanang		88333	Lampung	<span>Tambah</span> <span>Hapus</span>
4	Revisi	Saraga		2222		<span>Tambah</span> <span>Hapus</span>

**Gambar 4. 11 Halaman Perusahaan**

Algoritma pada halaman perusahaan dapat dilihat pada tabel 4.19 berikut ini

**Tabel 4. 18 Algoritma Halaman Perusahaan**

Keterangan	Code
Proses Tambah Data	<ol style="list-style-type: none"> <li>1. Mendapatkan data dari formulir, yaitu jenis perusahaan, nama perusahaan, keterangan, alamat, dan nomor telepon.</li> <li>2. Menyiapkan data yang akan disimpan ke dalam database dalam bentuk array asosiatif dengan kunci sebagai nama kolom di tabel dan nilai sebagai data yang diambil dari formulir.</li> <li>3. Memulai transaksi dengan database untuk memastikan operasi penambahan data perusahaan dilakukan secara atomik.</li> <li>4. Melakukan operasi penyimpanan data ke dalam tabel <code>tb_perusahaan</code>.</li> </ol>

	<ol style="list-style-type: none"> <li>5. Memeriksa status transaksi, jika gagal (FALSE), maka melakukan rollback transaksi untuk membatalkan operasi yang sudah dilakukan dan menampilkan pesan kesalahan.</li> <li>6. Jika berhasil, melakukan commit transaksi untuk menyimpan perubahan ke dalam database secara permanen dan menampilkan pesan keberhasilan.</li> </ol>
--	--

#### 4.1.4.12 Halaman Tipe Pekerjaan

Halaman ini digunakan oleh admin untuk mengelola tipe pekerjaan yang nantinya akan mengklasifikasikan lowongan pekerjaan. Gambar halaman tipe pekerjaan dapat dilihat pada gambar 4.12 berikut ini

No	Nama Pekerjaan	Jurusan	Minat	Skill	Aksi
1	programmer	SI Sistem Informasi, SI Teknik Informatika, SI Manajemen Informatika	WEB, MOBILE	SKL, SKL, SKL	Tambah
2	akuntansi	SI Sistem Informasi, SI Akuntansi, SI Ekonomi, SI Akuntansi Keuangan	WEB, MOBILE	SKL, SKL, SKL, SKL	Tambah
3	Jaringan Komputer	SI Teknik Informatika, DIII Teknik Komputer Jaringan	WEB, MOBILE	SKL, SKL, SKL	Tambah

**Gambar 4. 12 Halaman Tipe Pekerjaan**

Algoritma pada halaman tipe pekerjaan dapat dilihat pada tabel 4.20 berikut ini

**Tabel 4. 19 Algoritma Halaman Tipe Pekerjaan**

Keterangan	Code
Proses Tambah Data	<ol style="list-style-type: none"> <li>1. Mendapatkan data dari formulir, yaitu skill, jurusan, minat, nama pekerjaan, dan keterangan.</li> </ol>

2. Mengecek dan menggabungkan data skill, jurusan, dan minat jika tersedia, dengan cara mengubah array menjadi satu string yang dipisahkan oleh koma (,).
3. Menyiapkan data yang akan disimpan ke dalam database dalam bentuk array asosiatif dengan kunci sebagai nama kolom di tabel dan nilai sebagai data yang diambil dari formulir.
4. Memulai transaksi dengan database untuk memastikan operasi penambahan data pekerjaan dilakukan secara atomik.
5. Melakukan operasi penyimpanan data ke dalam tabel **tb\_pekerjaan**.
6. Memeriksa status transaksi, jika gagal (FALSE), maka melakukan rollback transaksi untuk membatalkan operasi yang sudah dilakukan dan menampilkan pesan kesalahan.
7. Jika berhasil, melakukan commit transaksi untuk menyimpan perubahan ke dalam database secara permanen dan menampilkan pesan keberhasilan.

#### 4.1.4.13 Halaman Lowongan Kerja

Halaman ini digunakan admin untuk mengelola data lowongan kerja yang dibuka sesuai periodenya. Gambar halaman lowongan kerja dapat dilihat pada gambar 4.13 berikut ini

No	Nama Perusahaan	Nama Lowongan	Jenis Pekerjaan	Jenjang Pendidikan	Image	Aksi
1	PT Pulu Intgra	Full Stack Developer	programmer	Sarjana (S1)		<a href="#">Edit</a> <a href="#">Hapus</a>
2	THI	Marketing	Marketing	Diploma		<a href="#">Edit</a> <a href="#">Hapus</a>
3	Saritaga	IT Support Komputer	IT Support Komputer	SMA/SMK Setara		<a href="#">Edit</a> <a href="#">Hapus</a>

**Gambar 4. 13 Halaman Lowongan Kerja**

Algoritma pada halaman jurusan dapat dilihat pada tabel 4.21 berikut ini

**Tabel 4. 20 Algoritma Halaman Lowongan Kerja**

Keterangan	Code
Proses Tambah Data	<ol style="list-style-type: none"> <li>1. Mengecek apakah file foto lowongan ada.</li> <li>2. Jika file foto lowongan tidak kosong, lanjutkan ke langkah 3; jika kosong, lanjutkan ke langkah 7.</li> <li>3. Mengupload file foto lowongan.</li> <li>4. Jika pesan upload adalah "sukses", lanjutkan ke langkah 5; jika tidak, lanjutkan ke langkah 6.</li> <li>5. Menyiapkan data yang akan disimpan ke database:               <ol style="list-style-type: none"> <li>a. Ambil data nama lowongan dari input form.</li> <li>b. Ambil data id perusahaan dari input form.</li> <li>c. Ambil data id pekerjaan dari input form.</li> <li>d. Ambil data jenjang pendidikan dari input form.</li> <li>e. Ambil data gaji minimum dari input form.</li> <li>f. Ambil data gaji maksimum</li> </ol> </li> </ol>

	<p>dari input form. g. Ambil data usia maksimum dari input form. h. Ambil data usia minimum dari input form. i. Ambil data status lowongan dari input form. j. Ambil data keterangan dari input form. k. Ambil data tanggal buka dari input form. l. Ambil data tanggal tutup dari input form. m. Ambil nama file foto lowongan dari hasil upload.</p> <p>6. Tampilkan pesan gagal dengan informasi error upload.</p> <p>7. Tampilkan pesan gagal karena file foto lowongan kosong.</p>
--	---

#### 4.1.4.14 Halaman Lamaran Kerja

Halaman ini digunakan oleh admin untuk mengelola data pelamar yang telah melamar pada perusahaan dan mengubah status dari pelamar sesuai dengan keputusan dari pemberi kerja. Gambar halaman lamaran kerja dapat dilihat dari gambar 4.14 berikut ini

No	Nama Perusahaan	Nama Lowongan	Jenis Pekerjaan	Jenjang Pendidikan	Image	Aksi
1.	Sinergi	Jaringan Komputer	Jaringan Komputer	SMA/SMK Setara		Lamar Pelamar
2.	TRE	Akuntansi	Akuntansi	Diploma		Lamar Pelamar
3.	PT Pelita Integri	Full Stack Developer	programmer	Setara D3		Lamar Pelamar

**Gambar 4. 14 Halaman Lamaran Kerja**

Algoritma pada halaman lamaran kerja dapat dilihat pada tabel 4.22 berikut ini

Tabel 4. 21 Algoritma Halaman Lamaran Kerja

Keterangan	Code
Proses Update Status Lolow	<ol style="list-style-type: none"> <li>1. Ambil data yang diterima dari form: a. Ambil id dari input form. b. Ambil id status lolos dari input form.</li> <li>2. Siapkan data yang akan diupdate: a. Buat objek <code>data_simpan</code> yang berisi: - 'id_status_lolos': id_status_lolos yang diambil dari input form.</li> <li>3. Mulai transaksi database.</li> <li>4. Lakukan update pada tabel <code>tb_apply_job</code>: a. Cari data berdasarkan id. b. Lakukan update data_simpan pada tabel <code>tb_apply_job</code> dengan kondisi where.</li> <li>5. Periksa status transaksi: a. Jika status transaksi adalah FALSE: - Rollback transaksi. - Tampilkan pesan gagal. b. Selain itu: - Commit transaksi. - Tampilkan pesan sukses.</li> </ol>

## 4.2 Pembahasan

### 4.2.1 Pengujian dengan *Blackbox Testing*

Blackbox testing merupakan pengujian yang dapat dilakukan dengan pengamatan, pada hasil eksekusi melalui beberapa data uji dan memeriksa fungsional yang terdapat pada sistem. Pengujian blackbox dilakukan dengan cara yang relative bertentangan dengan kebutuhan dan memastikan sistem dapat menangani semua masukan yang tidak sesuai, sehingga user hanya bisa memasukkan data yang benar ke dalam sistem. Pengujian ini berusaha menemukan kesalahan pada sistem kesesuaian lowongan kerja pada IIB DARMAJAYA diantaranya :

- a. Fungsi-fungsi yang tidak benar atau hilang terdapat pada sistem.
- b. Kesalahan dalam perancangan sistem dan kinerja sistem.
- c. Kesalahan dalam struktur data atau akses database eksternal dalam sistem

#### **4.2.1.1 Perangkat Pengujian**

Adapun perangkat pengujian yang digunakan pada pengujian sistem ini adalah sebagai berikut:

- a. Pengujian offline dengan localhost phpMyAdmin
- b. Menggunakan web browser Google Chrome
- c. Device laptop Asus VivoBook Intel Core i5
- d. Operating system Windows 10 dengan resolusi layar 1366 x 768 pixels.

#### **4.2.1.2 Material Pengujian**

Materi yang diuji dalam pengujian aplikasi kesesuaian lowongan kerja pada IIB DARMAJAYA adalah sebagai berikut:

- a. Kebutuhan fungsional yang terkait dengan tiap menu dari aplikasi kesesuaian lowongan pekerjaan.
- b. Tampilan *user interface* yang terkait dalam interaksi *user* dengan sistem.
- c. Kompatibilitas sistem terhadap sistem operasi yang digunakan.

#### **4.2.1.3 Hasil Uji Sistem**

Adapun pengujian web kesesuaian lowongan pekerjaan dapat dilihat padatablel pengujian dibawah ini adalah sebagai berikut:



Tabel 4. 22 Hasil Pengujian Sistem (Level Admin)

Skenario Pengujian	Kasus Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Halaman <i>Login</i>	Memasukkan <i>username</i> dan <i>password</i>	Admin berhasil melakukan <i>login</i>	Sesuai	Normal
<i>Dashboard</i> Admin	Masuk ke tampilan utama ( <i>dashboard</i> )	<i>Dashboard</i> muncul tanpa ada <i>error</i>	Sesuai	Normal
Skenario Pengujian	Kasus Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Menu Jurusan	Menguji menambahkan data, mengedit data, menghapus data jurusan	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Skill	Menguji menambahkan data, mengedit data, menghapus data skill	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Minat	Menguji menambahkan data, mengedit data, menghapus data minat	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Status Lolos	Menguji menambahkan data, mengedit data, menghapus data status lolos	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu status peserta	Menguji menambahkan data, mengedit data, menghapus data status peserta	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal

Menu Perusahaan	Menguji menambahkan data, mengedit data, menghapus data perusahaan	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Pelamar	Menguji, mengedit data, view pelamar dan download file pendidikan	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Skenario Pengujian	Kasus Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Menu lowongan kerja	Menguji menambahkan data, mengedit data, menghapus data lowongan kerja	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Lamaran Kerja	Menguji lihat pelamar, mengedit data, melihat profil pelamar	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal

**Tabel 4. 23 Hasil Pengujian Sistem (Level Pelamar Kerja)**

Skenario Pengujian	Kasus Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Halaman Register	menguji proses penambahan data registrasi	Semua tombol dapat diakses dan dapat memberikan	Sesuai	Normal

		respon dengan baik		
Halaman <i>Login</i>	Memasukkan <i>username</i> dan <i>password</i>	Pelamar berhasil melakukan <i>login</i>	Sesuai	Normal
<i>Dashboard</i> Admin	Masuk ke tampilan utama ( <i>dashboard</i> )	<i>Dashboard</i> muncul tanpa ada <i>error</i>	Sesuai	Normal
Skenario Pengujian	Kasus Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
Menu Profil Pelamar	Menguji mengedit data	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Pendidikan	Menguji menambahkan data, mengedit data, menghapus data pendidikan	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu Rekomendasi Lamaran	Menguji menambahkan data, lihat detail pekerjaan dan melamar pekerjaan	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal
Menu historilamaran	Menguji tampilan list pekerjaan yang sudah dilamar	Semua tombol dapat diakses dan dapat memberikan respon dengan baik	Sesuai	Normal

#### 4.2.2 Kelebihan Sistem

Kelebihan dari sistem kesesuaian lowongan pekerjaan pada IIB DARMAJAYA dengan *promethee* ini antara lain :

- a. Penyesuaian Lebih Mudah: Dalam konteks prediksi kesesuaian lowongan kerja, kemungkinan ada perluasan atau perubahan dalam algoritma *Promethee*. Penambahan kriteria dapat menambah kolom data untuk mencari lebih jauh kesesuaiannya
- b. Pencegahan kesalahan lamar pekerjaan: Dengan rekomendasi yang tepat, para pencari kerja akan menghemat waktu saat memilih list lowongan kerja yang diinginkan.
- c. Efisiensi waktu: Dengan merencanakan lowongan yang ingin dilamar berdasarkan prioritas, dapat mengalokasikan waktu dengan lebih efisien.

#### 4.2.3 Kekurangan Sistem

- a. Konsistensi Data: Dalam sistem kesesuaian lowongan pekerjaan, akurasi prediksi sangat tergantung pada kualitas data lowongan dan data diri pelamar. Prototipe awal mungkin memiliki keterbatasan dalam menggambarkan data yang sesuai, yang dapat mempengaruhi akurasi hasil.
- b. Kesalahan Prediksi: Tidak ada metode prediksi yang sepenuhnya akurat. Terkadang, algoritma bisa memberikan hasil yang salah, yang dapat mengakibatkan perhitungan yang tidak perlu atau lowongan yang tidak sesuai.

Jika implementasinya dilakukan dengan hati-hati dan dikelola dengan baik, algoritma *promethee* dapat memberikan manfaat yang signifikan dalam menentukan kesesuaian lowongan pekerjaan dengan pelamar kerja.