

## BAB III

### METODE DAN PERANCANGAN APLIKASI

#### 3.1 Metode Penelitian

Metode penelitian yang dilakukan dalam penelitian sebagai berikut

1. Studi Literatur

Studi literatur merupakan tahapan dimana peneliti melakukan pembelajaran dari buku, jurnal, artikel, maupun referensi lain, yang tersedia secara *online* maupun *offline*, yang berhubungan dengan penelitian yang dilakukan.

2. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan sistem, meliputi fitur-fitur yang perlu disediakan, seperti *software*, dan data-data mahasiswa yang sudah dikonsultasi.

3. Desain Sistem

Pada tahap desain sistem ini dilakukan perancangan *user interface*, desain modul-modul aplikasi, dan desain konten.

4. Pemrograman Sistem

Pada tahap ini, dilakukan pemrograman sistem yang dibutuhkan sesuai rancangan, dan berjalan sesuai rancangan yang telah dibuat.

5. Testing

Pada tahap ini, dilakukan *testing* atau percobaan terhadap aplikasi yang telah dirancang untuk memeriksa apakah ada *error*, kekurangan, atau ada yang tidak sesuai dengan rancangan.

6. Implementasi

Pada tahap ini, dilakukan implementasi sistem yang sudah dibuat peneliti kepada pengguna.

#### 7. Evaluasi

Pada tahap ini, dilakukan evaluasi terhadap pengguna sistem. Berdasarkan hasil evaluasi dilakukan analisa untuk ditarik kesimpulan dari penelitian.

#### 8. Penulisan Laporan

Pada tahap ini, dilakukan penulisan laporan untuk menjabarkan penelitian yang dilakukan.

### 3.2 Rancangan Aplikasi

Berikut ini adalah penjelasan secara garis besar tentang alur proses aplikasi. Dalam penelitian ini, aplikasi SPK dibuat dengan bahasa C# dan digunakan untuk mendukung keputusan beasiswa di UMN dengan mengimplementasikan metode *profile matching*. Gambar 3.1 menampilkan garis besar alur aplikasi yang telah dibuat. Dimulai dengan *input* data mahasiswa, lalu mengecek data terakhir atau bukan. Jika sudah selesai menambahkan data, sistem akan melakukan pemetaan *gap* kompetensi. Lalu dilakukan proses pembobotan nilai *gap* berdasarkan pemetaan *gap* kompetensi sebelumnya. Sistem akan menghitung tabel bobot berdasarkan hasil nilai *gap*. Pengelompokkan *core factor* dan *secondary factor* pada setiap sub-aspek seperti IPK, tanggungan orang tua, dan lainnya. Setelah dikelompokkan, akan dihitung nilai total untuk setiap aspeknya. Berdasarkan hasil yang diperoleh, akan digunakan untuk menghitung ranking. Ranking semakin besar, maka semakin besar mahasiswa tersebut mendapat kesempatan untuk memperoleh beasiswa.



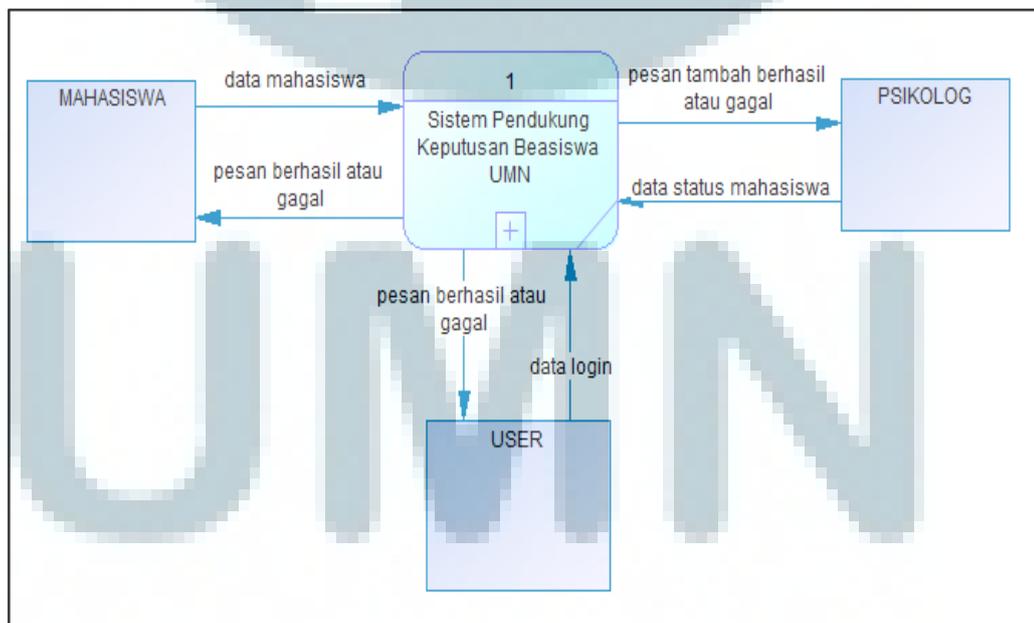
Gambar 3.1 Diagram Alir Aplikasi

### 3.3 Data Flow Diagram (DFD)

Menurut Kristanto (2003), *Data Flow Diagram (DFD)* adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluaran dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

#### 3.3.1 Context Diagram

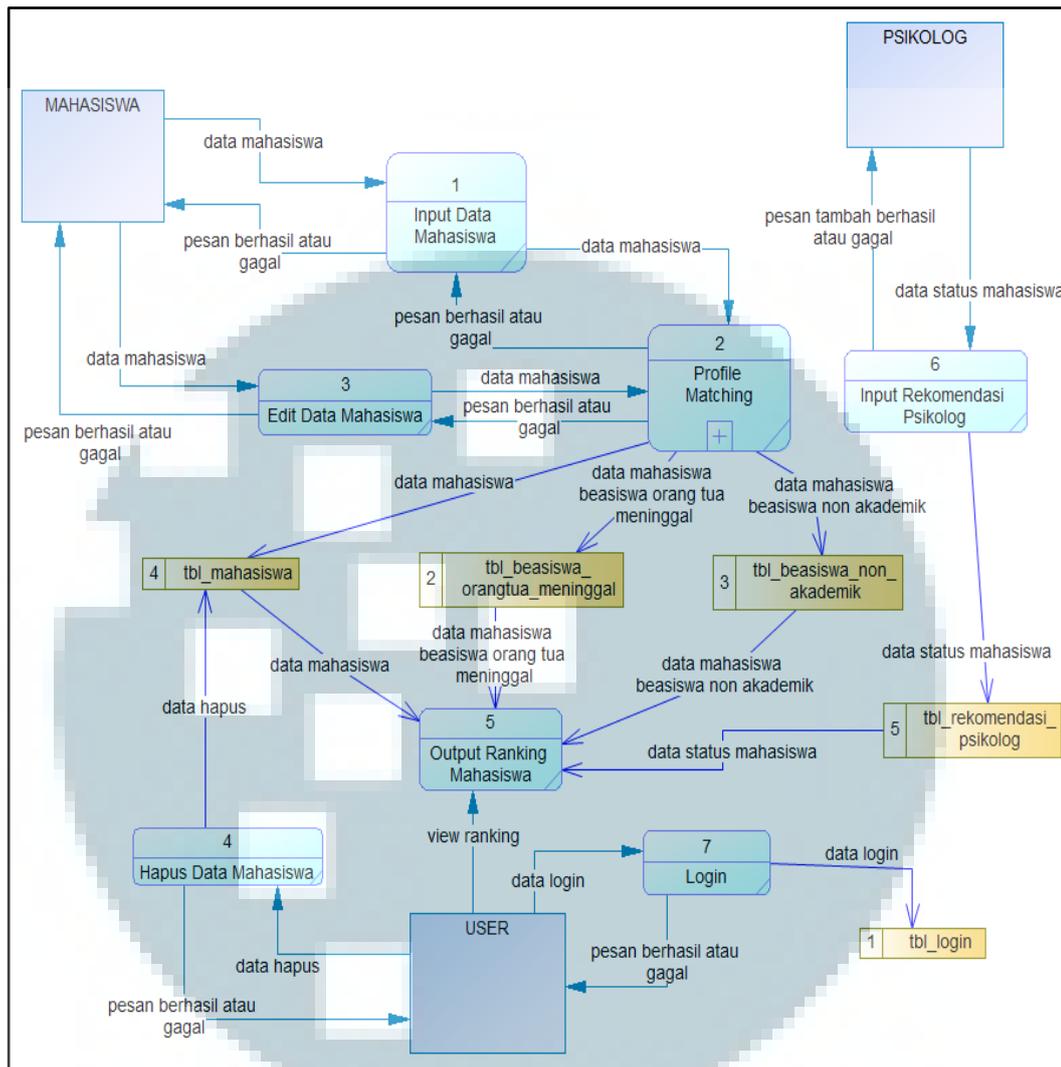
Pada *Context Diagram* digambarkan bahwa aplikasi menerima data mahasiswa, data status mahasiswa, dan data login. Terdapat tiga entitas, yaitu mahasiswa, psikolog, dan user. Data tersebut diproses oleh aplikasi Sistem Pendukung Keputusan Beasiswa UMN. Proses yang telah dilakukan aplikasi tersebut mengembalikan pesan tambah berhasil atau gagal kepada psikolog, pesan berhasil atau gagal kepada mahasiswa dan user. Ilustrasi dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Context Diagram* Aplikasi

### 3.3.2 DFD Level 1

Pada DFD Level 1 digambarkan bahwa data yang dimasukkan pengguna dilanjutkan ke dalam tujuh proses. Terdapat tiga entitas (User, Mahasiswa, dan Psikolog), tujuh proses (Login, Input Data Mahasiswa, Profile Matching, Edit Data Mahasiswa, Hapus Data Mahasiswa, Output Ranking Mahasiswa, Input Rekomendasi Psikolog), dan lima tabel (tbl\_login, tbl\_basiswa\_non\_akademik, tbl\_basiswa\_orangtua\_meninggal, tbl\_mahasiswa, dan tbl\_rekomendasi\_psikolog). Data login ke proses Login yang terhubung dengan *database* MySQL untuk mengecek pada tbl\_login, lalu akan mengembalikan pesan berhasil atau gagal. Data mahasiswa ke proses Input Data Mahasiswa, lalu dilanjutkan ke proses Profile Matching. Data mahasiswa ke proses Edit Data Mahasiswa, lalu dilanjutkan ke proses Profile Matching. Data hapus ke proses Hapus Data Mahasiswa, dimana proses tersebut akan menghapus data di dalam tbl\_mahasiswa. Data view ranking ke proses Output Ranking Mahasiswa, dimana proses tersebut menampilkan data dari tbl\_basiswa\_orangtua\_meninggal dan tbl\_basiswa\_non\_akademik. Data status mahasiswa ke proses Input Rekomendasi Psikolog, dimana data tersebut akan disimpan ke dalam tbl\_rekomendasi\_psikolog dan akan ditampilkan pada saat menampilkan ranking mahasiswa. Ilustrasi dapat dilihat pada Gambar 3.3.

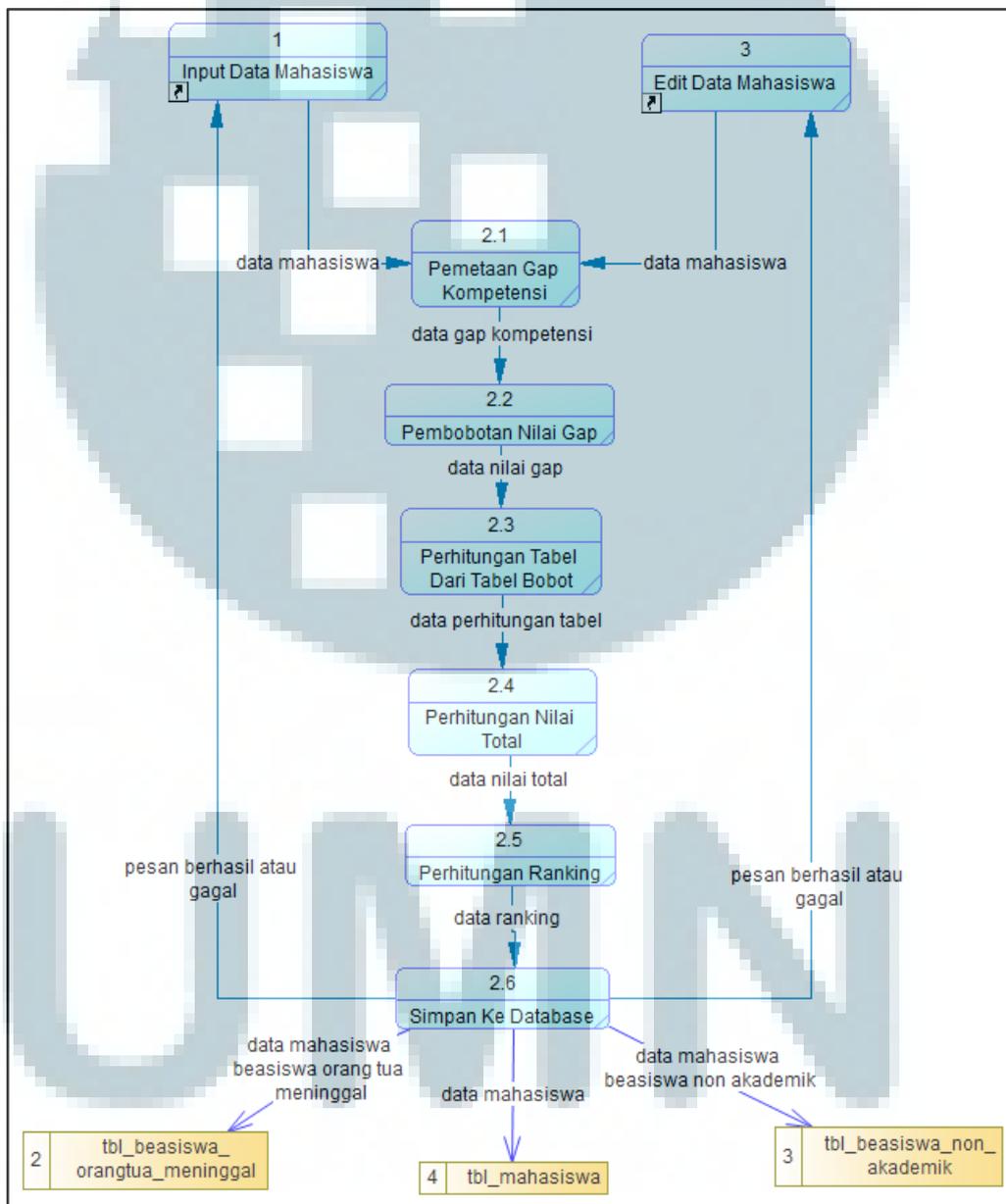


Gambar 3.3 DFD Level 1 Aplikasi

### 3.3.3 DFD Level 2 Proses Profile Matching

Pada DFD Level 2 proses Profile Matching, terdapat delapan proses (Input Data Mahasiswa, Edit Data Mahasiswa, Pemetaan Gap Kompetensi, Pembobotan Nilai Gap, Perhitungan Tabel dari Tabel Bobot, Perhitungan Nilai Total, Perhitungan Ranking, dan Simpan ke Database), tiga tabel (tbl\_basiswa\_orangtua\_meninggal, tbl\_basiswa\_non\_akademik, dan tbl\_mahasiswa). Pada DFD ini digambarkan masukan dari pengguna yaitu data mahasiswa diproses oleh Pemetaan Gap Kompetensi. Proses berikutnya adalah

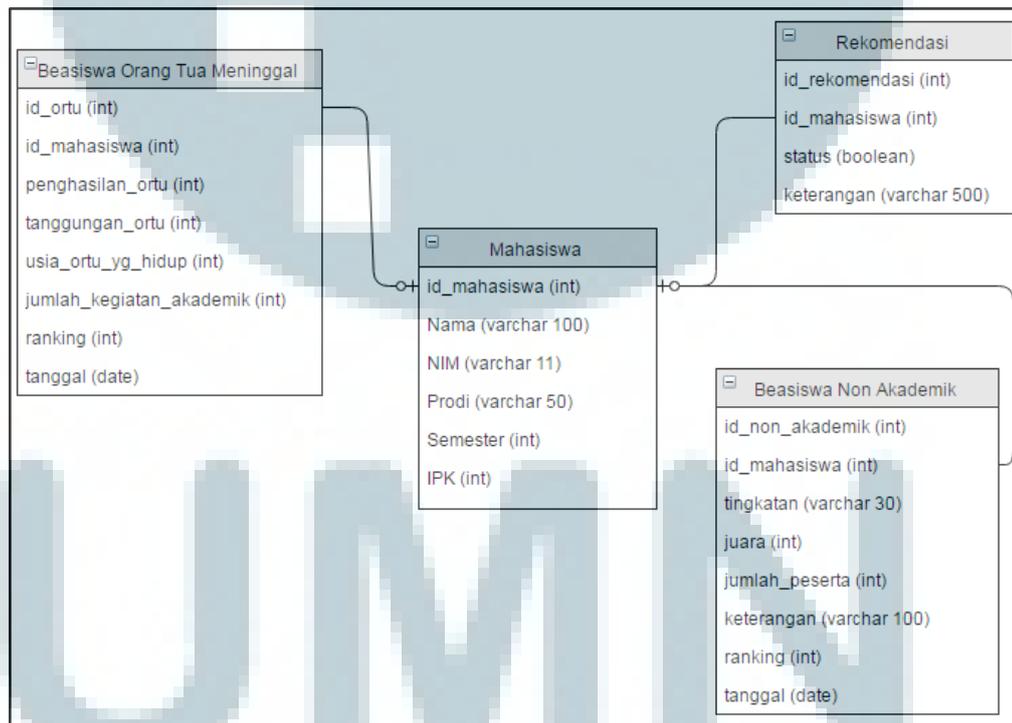
Pembobotan Nilai Gap, hasil dari pembobotan tersebut akan diproses ke Perhitungan Tabel dari Tabel Bobot. Setelah perhitungan sudah didapat, diproses ke Perhitungan Nilai Total. Hasil dari perhitungan nilai total akan diproses ke Perhitungan Ranking. Hasil akhir dari perhitungan ranking akan disimpan ke dalam *database* yang menggunakan MySQL. Ilustrasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 DFD Level 2 Proses *Profile Matching*

### 3.4 Entity Relationship Diagram

Untuk menjelaskan hubungan antar objek data dalam *database* dapat digambarkan dengan *Entity Relationship Diagram* (ERD). Terdapat empat buah tabel yang memiliki hubungan yakni Mahasiswa, Rekomendasi, Beasiswa Orang Tua Meninggal, dan Beasiswa Non Akademik. Pada tabel Mahasiswa terdapat atribut *primary key* *id\_mahasiswa* yang terdapat juga di tabel beasiswa orang tua meninggal, rekomendasi, dan beasiswa non akademik sebagai *foreign key*. Hubungan antara tabel mahasiswa dengan yang lainnya adalah 1 to 0. Mahasiswa bisa punya satu atau tidak punya beasiswa orang tua meninggal atau beasiswa non akademik atau rekomendasi. Hubungan antar entitas dan atributnya dapat dilihat pada Gambar 3.5.



Gambar 3.5 ERD Sistem Pendukung Keputusan Beasiswa UMN

### 3.5 Struktur Tabel

Dalam pembuatan sistem pendukung keputusan beasiswa UMN terdapat lima tabel, dimana empat diantaranya adalah tabel yang memiliki hubungan dengan tabel lainnya dan tabel terakhir tidak memiliki hubungan dengan tabel lainnya. Kelima tabel tersebut adalah:

1. Nama Tabel : mahasiswa  
Fungsi : menyimpan data mahasiswa  
*Primary Key* : id\_mahasiswa  
*Foreign Key* : -

Tabel 3.1 Tabel mahasiswa

No.	Nama Kolom	Tipe Data	Panjang	Keterangan
1.	id_mahasiswa	Int		Id unik mahasiswa, <i>auto increment</i>
2.	nama	Varchar	50	Nama mahasiswa
3.	NIM	Varchar	11	NIM mahasiswa
4.	prodi	Varchar	50	Prodi mahasiswa
5.	semester	Int		Semester mahasiswa
6.	IPK	Double		IPK mahasiswa

2. Nama Tabel : login  
Fungsi : menyimpan data untuk *login*  
*Primary Key* : -  
*Foreign Key* : -

Tabel 3.2 Tabel login

No.	Nama Kolom	Tipe Data	Panjang	Keterangan
1.	id_ID	Int		Id unik ID, <i>auto increment</i>
2.	ID	Varchar	100	ID pengguna
3.	password	Varchar	100	Password pengguna

3. Nama Tabel : beasiswa\_orangtua\_meninggal
- Fungsi : menyimpan data mahasiswa yang mendaftar beasiswa orang tua meninggal
- Primary Key* : id\_orstu
- Foreign Key* : id\_mahasiswa

Tabel 3.3 Tabel beasiswa\_orangtua\_meninggal

No.	Nama Kolom	Tipe Data	Panjang	Keterangan
1.	id_orstu	Int		Id unik beasiswa orang tua meninggal, <i>auto increment</i>
2.	id_mahasiswa	Int		Id unik mahasiswa, <i>auto increment</i>
3.	penghasilan_orstu	Int		Penghasilan orang tua mahasiswa
4.	tanggungan_orstu	Int		Tanggungan orang tua mahasiswa
5.	usia_orstu	Int		Usia orang tua mahasiswa yang masih hidup
6.	jumlah_kegiatan	Int		Jumlah kegiatan akademik mahasiswa
7.	ranking	Double		Hasil ranking dari <i>Profile Matching</i> untuk beasiswa orang tua meninggal
8.	tanggal	Date		Tanggal saat data ditambahkan

4. Nama Tabel : beasiswa\_non\_akademik
- Fungsi : menyimpan data mahasiswa yang mendaftar beasiswa non akademik
- Primary Key* : id\_non\_akademik
- Foreign Key* : id\_mahasiswa

Tabel 3.4 Tabel beasiswa\_non\_akademik

No.	Nama Kolom	Tipe Data	Panjang	Keterangan
1.	id_non_akademik	Int		Id unik beasiswa non akademik, <i>auto increment</i>
2.	id_mahasiswa	Int		Id unik mahasiswa, <i>auto increment</i>
3.	tingkatan	varchar	30	Tingkatan pertandingan
4.	juara	Int		Juara yang dicapai di pertandingan
5.	jumlah_peserta	Int		Jumlah peserta yang mengikuti pertandingan
6.	keterangan	varchar	50	Keterangan pertandingan
7.	ranking	Double		Hasil ranking dari <i>Profile Matching</i> untuk beasiswa non akademik
8.	tanggal	Date		Tanggal saat data ditambahkan

5. NamaTabel : rekomendasi\_psikolog

Fungsi : menyimpan data tambahan dari psikolog untuk beasiswa orang tua meninggal

*Primary Key* : id\_rekomend

*Foreign Key* : id\_mahasiswa

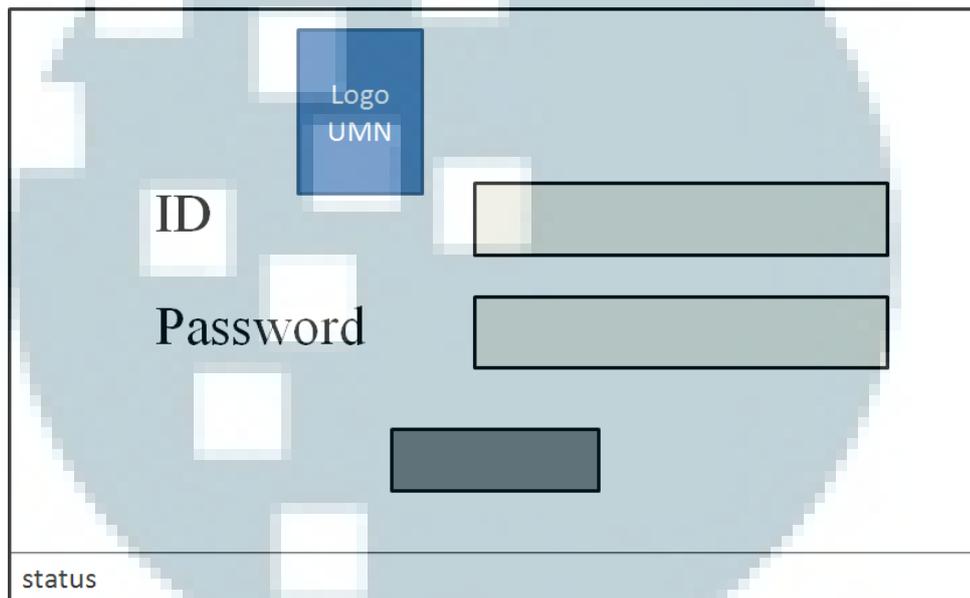
Tabel 3.5 Tabel rekomendasi\_psikolog

No.	Nama Kolom	Tipe Data	Panjang	Keterangan
1.	id_rekomend	Int		Id unik rekomendasi psikolog, <i>auto increment</i>
2.	id_mahasiswa	Int		Id unik mahasiswa, <i>auto increment</i>
3.	status	Tinyint		Status diterima atau ditolak dari Psikolog
4.	keterangan	Varchar	500	Ket. dari Psikolog

### 3.6 Rancangan Tampilan Antarmuka Aplikasi

Dalam pembuatan sistem pendukung keputusan ini, dibuat suatu desain standar untuk tampilan antarmuka secara garis besar. Berikut adalah gambaran desain tampilan antarmuka yang telah dirancang oleh peneliti.

Saat aplikasi dijalankan, maka halaman *login* akan ditampilkan. Halaman *login* dapat dilihat pada Gambar 3.6.



Gambar 3.6 Rancangan Tampilan Halaman *Login*

Halaman *login* memiliki dua *field* dan satu tombol, *field* tersebut adalah *ID* dan *password*. Sedangkan tombol yang ada adalah tombol untuk melakukan proses *login* setelah kedua *field* sudah terisi. Setelah *user login*, *user* akan dibawa ke tampilan menu utama aplikasi. Halaman utama dapat dilihat pada Gambar 3.7.

Menu 1	Menu 2	Menu 3	
Sub Menu			

Gambar 3.7 Rancangan Tampilan Halaman Utama

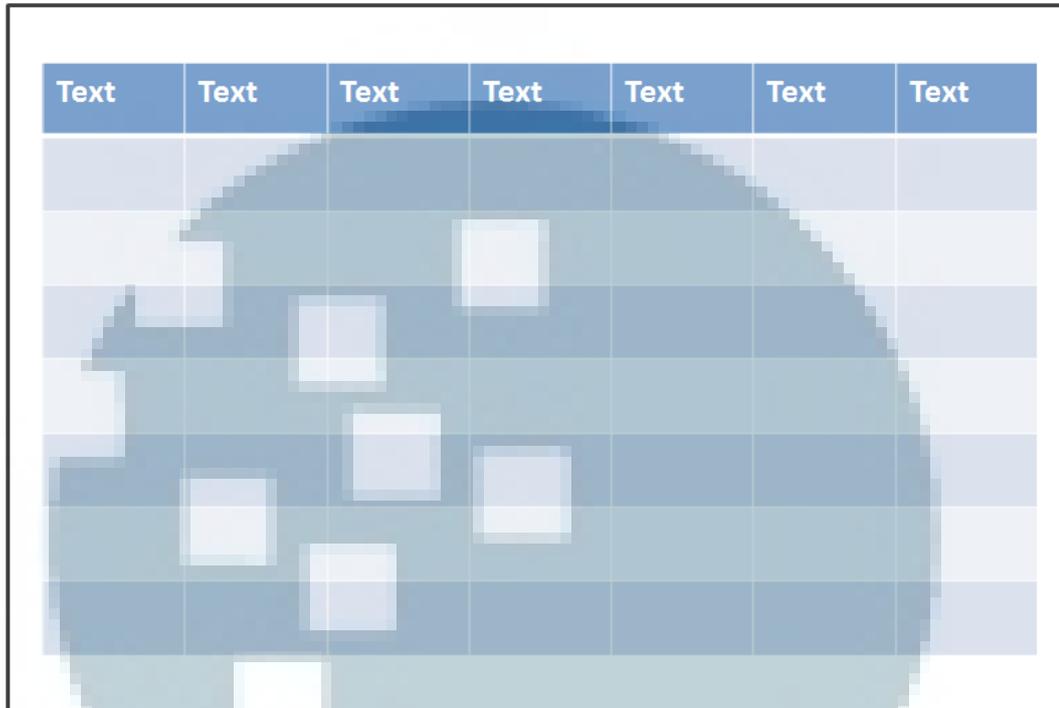
Halaman utama memiliki tiga *menu*, dimana setiap *menu* terdapat *sub-menu* yang digunakan untuk mengakses halaman *input*, *edit*, *delete*, *view*, *help*, dan *about*. Halaman *input* dapat dilihat pada Gambar 3.8.

UMMN

Gambar 3.8 Rancangan Tampilan Halaman *Input*

Pada halaman *input*, terdapat dua bagian utama. Bagian atas yaitu untuk memasukkan data mahasiswa yang meliputi nama, nomor induk mahasiswa (NIM), prodi, semester, dan indeks prestasi kumulatif (IPK). Bagian bawah yaitu memasukkan data beasiswa orang tua meninggal atau beasiswa non akademik,

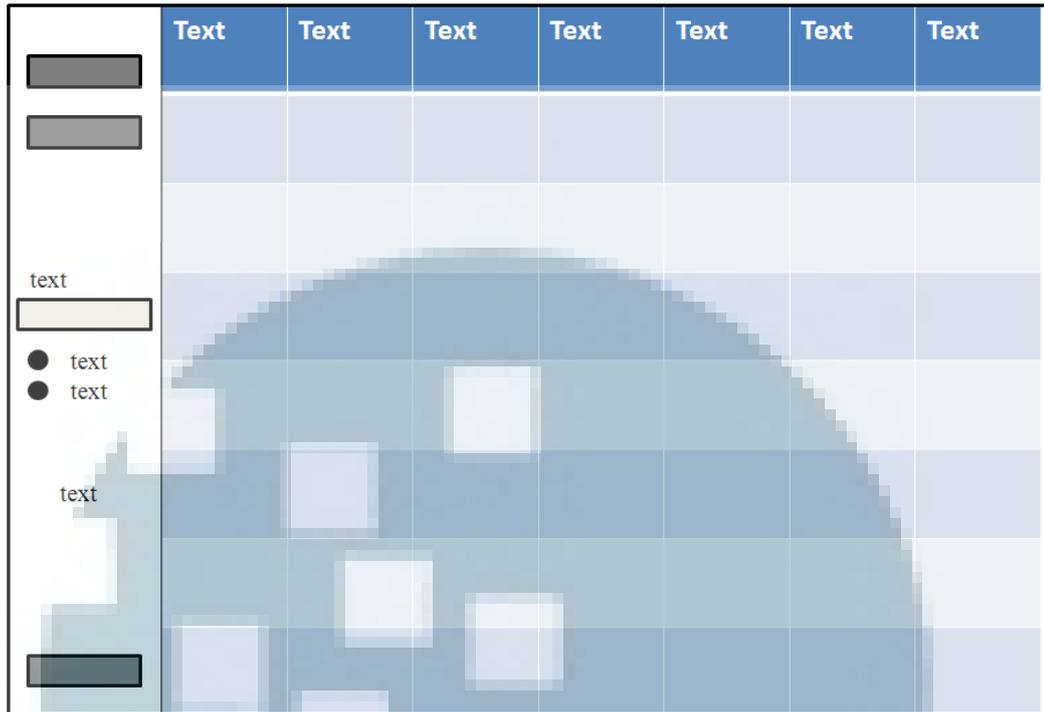
dan terdapat beberapa *field* untuk setiap beasiswa. Terdapat juga dua tombol, yaitu tombol untuk melakukan penambahan data dan tombol untuk membatalkan.



Gambar 3.9 Rancangan Tampilan Halaman *Edit*

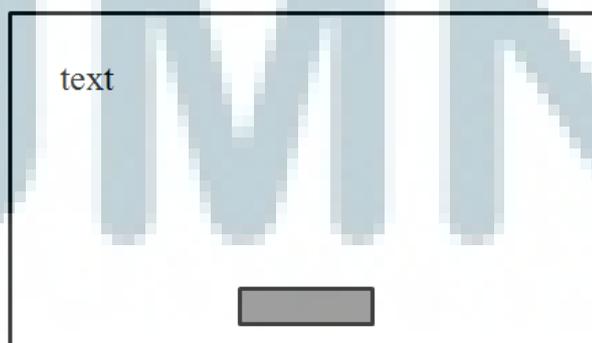
Pada halaman *edit* yang dapat dilihat pada Gambar 3.9, terdapat sebuah *listview* yang menampilkan data mahasiswa yang tersimpan di *database*. Ketika pengguna melakukan klik dua kali pada salah satu barisan, maka akan muncul halaman *Edit Data* yang tampilannya serupa dengan halaman *input*. *Listview* akan langsung memperbarui data yang sudah diubah.





Gambar 3.11 Rancangan Tampilan Halaman *View*

Pada halaman *View* yang dapat dilihat pada Gambar 3.11, terdapat *listview* yang menampilkan data mahasiswa yang tersimpan di *database* termasuk hasil dari metode yang diimplementasikan. Pada bagian kiri terdapat tiga tombol, dua tombol di atas adalah tombol untuk memilih beasiswa yang ingin ditampilkan. Tombol yang di bawah adalah tombol untuk mencetak data yang ada pada *listview*. Terdapat juga *field* yang digunakan untuk fitur pencarian atau *search* berdasarkan nama mahasiswa atau NIM mahasiswa yang tersimpan di *database*.



Gambar 3.12 Rancangan Tampilan Halaman *About*

Pada halaman *About* yang dapat dilihat pada Gambar 3.12, terdapat teks yang menuliskan tentang aplikasi yang telah dibuat. Terdapat juga satu tombol untuk menutup halaman tersebut.

