

BAB 3 METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Dalam penelitian “Sistem Diagnosa Penyakit Diabetes Melitus Dan Rekomendasi Makanan Menggunakan Metode Certainty Factor” memiliki beberapa tahapan untuk memenuhi metodologi penelitian dan perancangan sistem ini dapat di implementasikan secara tersusun. Adapun beberapa tahapan tersebut adalah:

1. Identifikasi Masalah

Dalam tahapan ini, proses yang dilakukan berupa mendefinisikan sistem yang sedang dibangun beserta tujuannya, perumusan masalah dari penelitian, mencari sumber data yang digunakan dalam penelitian, serta melakukan analisis kebutuhan yang diperlukan dalam penelitian ini.

2. Studi Literatur

Pada tahapan ini, dilakukan pencarian informasi terkait topik yang diambil dan metode yang digunakan. Pencarian informasi ini dapat dilakukan secara *online* maupun *offline*. Jika tahapan ini dilakukan secara *offline*, maka langkah yang dapat diambil adalah dengan melakukan wawancara dan mendapatkan informasi melalui buku ataupun jurnal terkait. Sedangkan, jika pencarian informasi secara *online* maka dapat mendapatkan informasi melalui jurnal *online*

3. Akuisisi Pengetahuan

Pada tahapan ini, akuisisi pengetahuan dilakukan dengan cara melakukan wawancara secara *online* bersama dengan pakar atau dokter ahli pada bidang terkait untuk mendapatkan data gejala dan hipotesa untuk mendiagnosa penyakit Diabetes. Serta melakukan wawancara secara langsung bersama dengan ahli gizi untuk mendapatkan data menu makanan selama 7 hari (*Meal Plan* dan panduan makan bagi penderita penyakit Diabetes).

4. Representasi Pengetahuan

Tahapan ini merupakan tahapan selanjutnya setelah diperolehnya akuisisi pengetahuan. Pada tahapan ini, dilakukan pembuatan *rules* sesuai dengan

metodologi yang digunakan berdasarkan pengetahuan yang telah didapatkan sebelum dilakukannya implementasi kedalam sistem.

5. Perancangan sistem

Pada tahapan ini, terdapat beberapa proses yang dibutuhkan. Adapun proses tersebut adalah *Data flow diagram*, *Use case diagram*, *Structure table*, *flowchart*, *Mockup*, *sitemap* serta rancangan *User interface (UI)*,

6. Implementasi

Pada tahapan ini, merupakan tahap dimana mulai dilakukan implementasi perhitungan dengan menggunakan metode *Certainty Factor* untuk melakukan diagnosa terhadap penyakit Diabetes dan memberikan rekomendasi makanan bagi penderita penyakit Diabetes. Hasil perhitungan dari metode ini digunakan sebagai acuan untuk melakukan diagnosa terhadap penyakit yang sedang diuji.

7. Pengujian Sistem dan Evaluasi

Pada tahapan ini, dilakukan pengujian terhadap sistem yang sudah dilakukan pada tahapan-tahapan sebelumnya. Kemudian, tahapan ini memiliki tujuan untuk melihat kesesuaian hasil dan keakuratan dari implementasi perhitungan dengan menggunakan metode *Certainty Factor*. Tahap pengujian sistem ini dilakukan oleh pengguna sistem dan melakukan evaluasi pengukuran tingkat kepuasan serta kelayakan terhadap pengguna sistem pakar diagnosa ini melalui pengisian formulir *Ease of User Satisfaction (USE) Quistionare*.

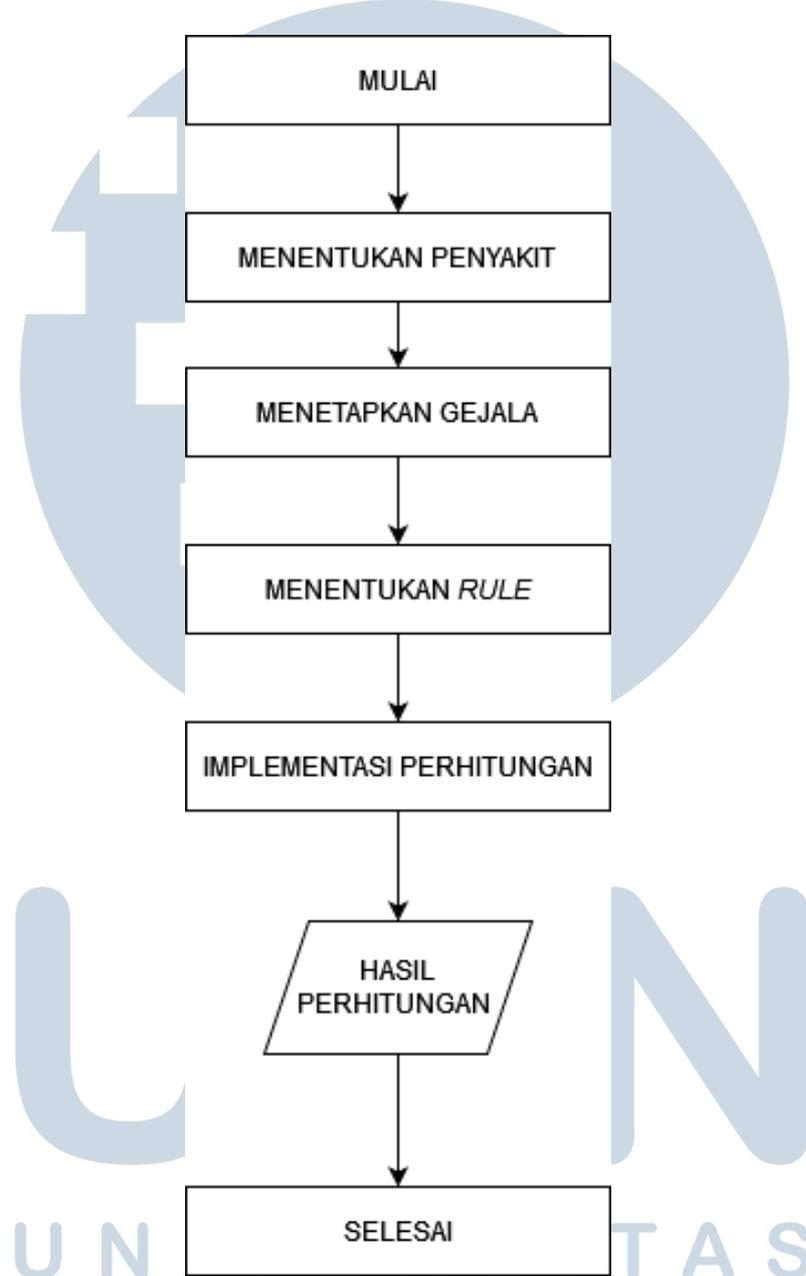
3.2 Perancangan Sistem

Pada bagian ini membahas perancangan sistem dengan mengimplementasikan perhitungan menggunakan metode *certainty factor* yang terdiri dari membuat *flowchart*, *design*, implementasi pemrograman, struktur *database*, dan tabel data.

3.2.1 Struktur Certainty Factor

Struktur dari *Certainty Factor* yang telah diimplementasikan dalam sistem diawali dengan menentukan gejala dan penyakit yang ingin di diagnosa, kemudian menentukan *rule* yang telah didapatkan dari pakar yang telah

diwawancarai. Kemudian dilakukan perhitungan sehingga memperoleh hasil, Alur yang menggambarkan langkah-langkah tersebut dapat dilihat pada gambar 3.1.



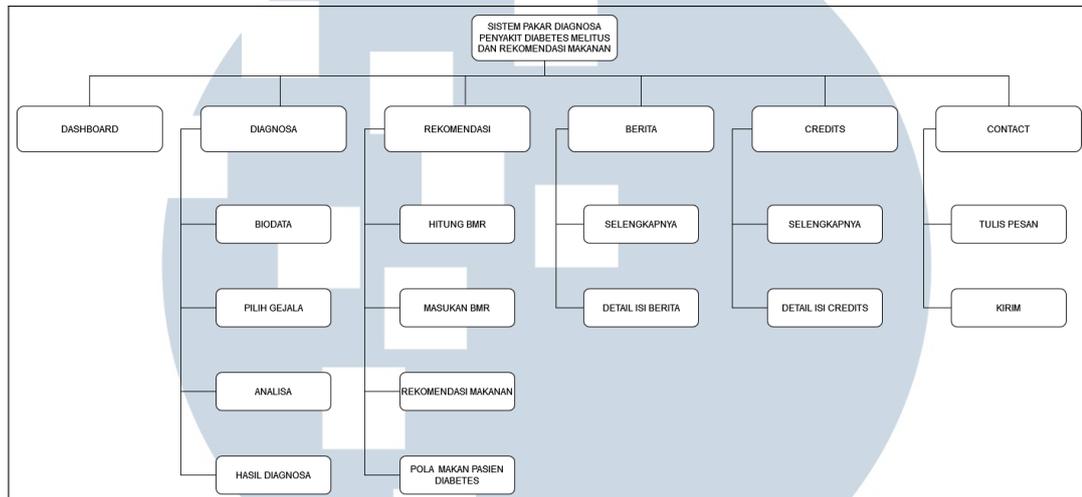
Gambar 3.1. Alur dari *Certainty Factor*.

3.2.2 Sitemap

Sitemap adalah sebuah diagram yang dibentuk sebagai gambaran untuk memetakan halaman-halaman yang berhubungan pada *website* sistem pakar diagnosa penyakit Diabetes Melitus.

Dalam bagian ini *Sitemap* terdiri dari 2 bagian yaitu :

A *Sitemap User*



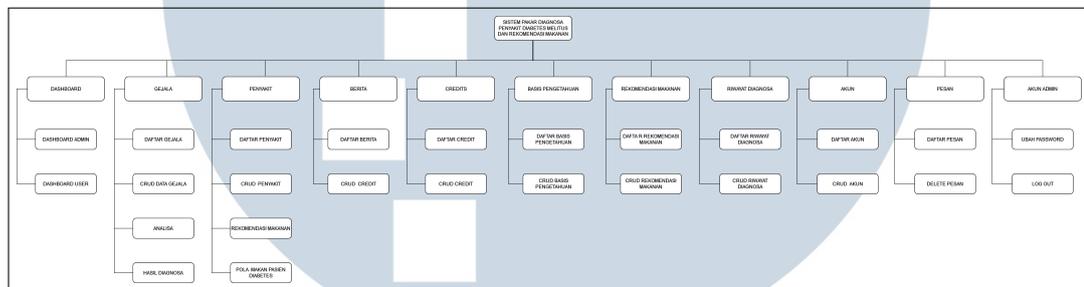
Gambar 3.2. *Sitemap User*

Pada gambar di 3.2 merupakan penjelasan dari *sitemap admin* yang menjelaskan secara singkat mengenai alur kerja dari sisi *user* yang telah dibuat. *Website* dimulai dengan menyambut user pada halaman dashboard, yang kemudian terdapat 6 (enam) jenis menu, yaitu : Menu Dashboard yang didalamnya berisikan 1 (satu) *Hero Section* yang didalamnya terdapat tombol diagnosa yang akan mengarahkan pengguna ke halaman diagnosa dan tombol rekomendasi yang akan mengarahkan pengguna ke halaman rekomendasi. Menu Diagnosa yang dimulai dengan mengisi biodata, kemudian akan lanjut ke halaman pemilihan gejala yang didalamnya terdapat 10 (sepuluh) gejala yang digunakan untuk mendiagnosa penyakit Diabetes Melitus menggunakan algoritma *Certainty Factor*.

Setelah itu pengguna akan dapat melihat hasil presentase perhitungan, deskripsi penyakit dan solusi dari penyakit tersebut. Menu Rekomendasi berisikan calculator perhitungan *Basal Metabolic Rate (BMR)* yang akan digunakan untuk menentukan rekomendasi makanan yang akan diberikan, kemudian terdapat kolom rekomendasi makanan yang akan digunakan untuk memunculkan hasil rekomendasi makanan ketika telah memasukan hasil *Basal Metabolic Rate (BMR)* yang telah dihitung sebelumnya.

Pada bagian bawah halaman juga terdapat pola makan bagi penderita Diabetes Melitus. Menu Berita berisikan tentang berita mengenai pengetahuan terhadap penyakit Diabetes Melitus. Menu *Credits* berisikan tentang informasi dari dokter spesialis ataupun pakar yang berpartisipasi dalam perancangan *website* sistem pakar diagnosa penyakit Diabetes Melitus dan Rekomendasi makanan serta Menu Contact yang berfungsi untuk memeberikan saran ataupun menghubungi *admin*.

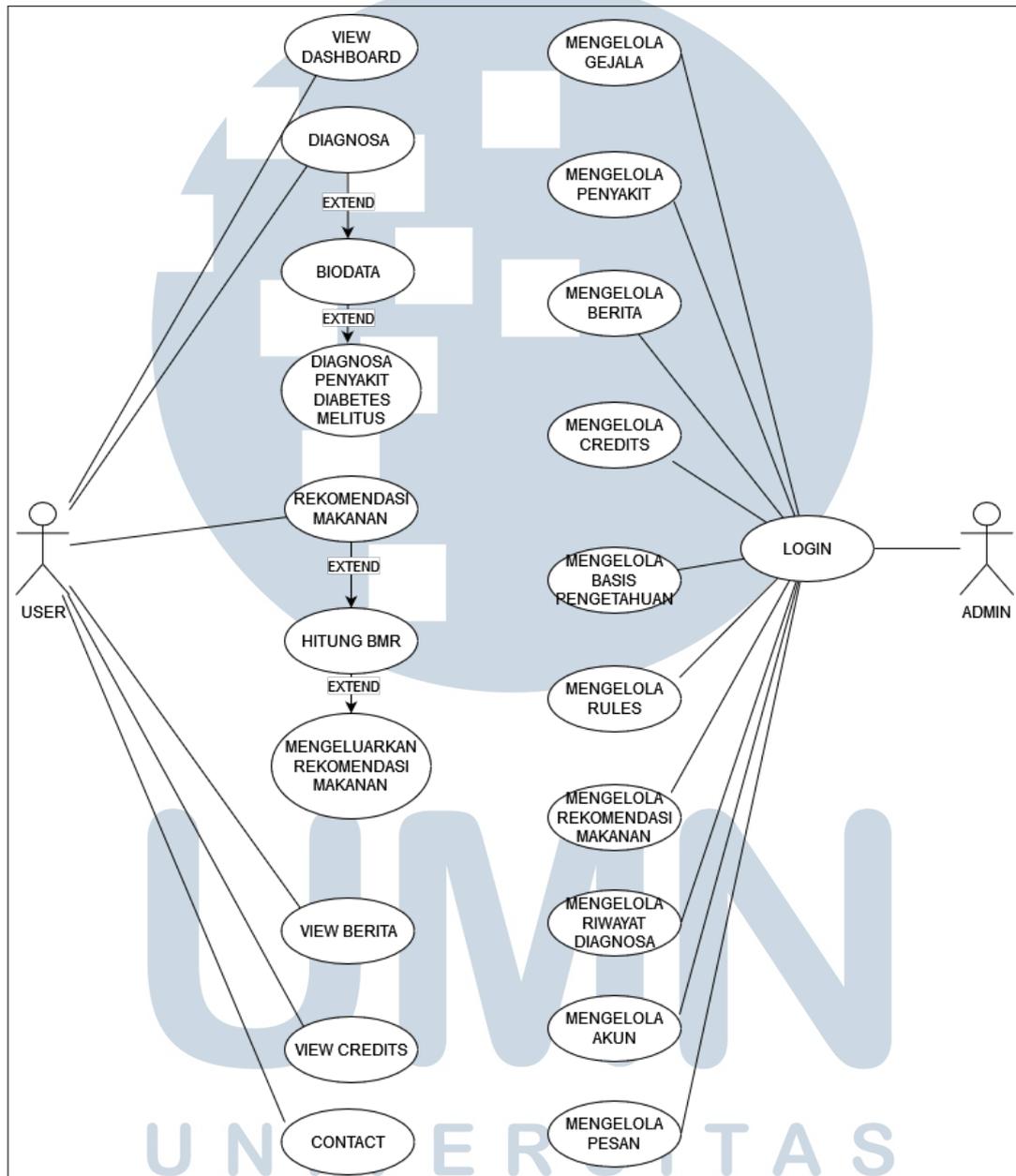
B Sitemap Admin



Gambar 3.3. Sitemap Admin

Pada gambar di 3.3 merupakan penjelasan dari *sitemap admin* yang menjelaskan secara singkat mengenai alur kerja dari sisi *admin* yang telah dibuat. Pada halaman *admin* terdapat 10 (sepuluh) menu yang masing-masing menu merupakan isi konten dari halama *user*. Masing-masing menu memiliki fungsi *Create*, *Edit* dan *Delete* untuk dapat mengubah data yang terdapat didalamnya.

3.2.3 Use Case Diagram



Gambar 3.4. Use Case Diagram

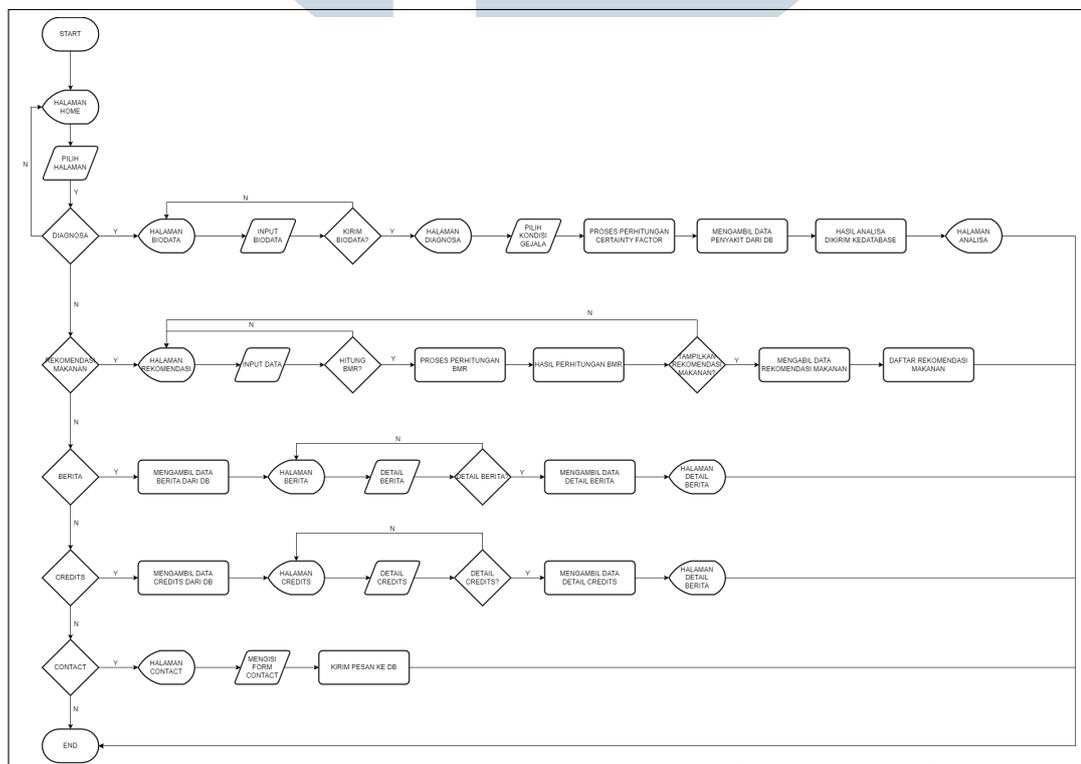
Gambar 3.4 merupakan *Use Case Diagram* dari sistem yang telah dibangun yang didalamnya terdapat *actor admin* dan *actor user*.

3.2.4 Flowchart

Flowchart merupakan diagram yang terdiri dari bentuk-bentuk geometris seperti kotak, oval, panah, dan berbagai simbol lainnya yang mewakili tindakan, keputusan, input/output, atau titik awal/akhir dari suatu proses. Setiap bentuk ini terhubung oleh panah yang menunjukkan aliran urutan langkah-langkah yang diikuti dalam proses tersebut. Berikut ini adalah beberapa bagian *flowchart* yang menggambarkan urutan langkah-langkah dalam penggunaan *Website* "Sistem Pakar Diagnosa Penyakit Diabetes Melitus Dan Rekomendasi Makanan Menggunakan Metode Certainty Factor":

A Flowchart User

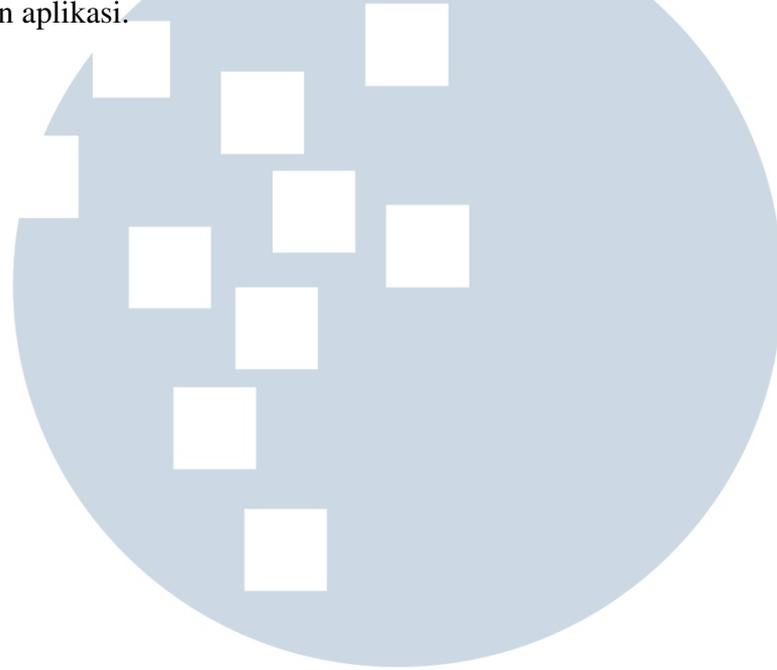
Berikut dibawah ini merupakan *flowchart* yang menggambarkan urutan langkah-langkah jalannya aplikasi dari sisi *user* dari awal sampai akhir penggunaan aplikasi.



Gambar 3.5. Flowchart User

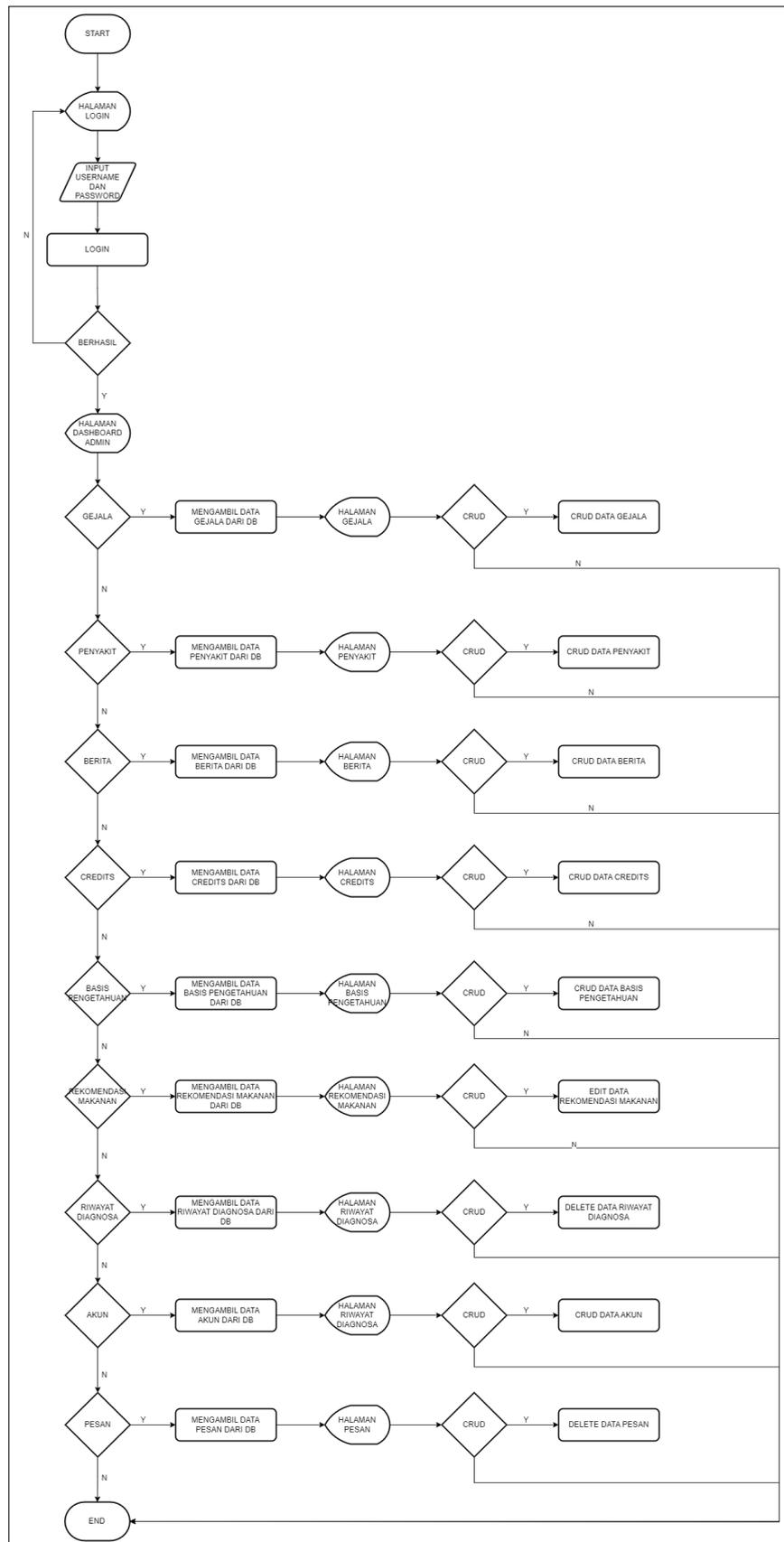
B Flowchart Admin

Berikut dibawah ini merupakan *flowchart* yang menggambarkan urutan langkah-langkah jalannya aplikasi dari sisi *Admin* dari awal sampai akhir penggunaan aplikasi.



UMMN

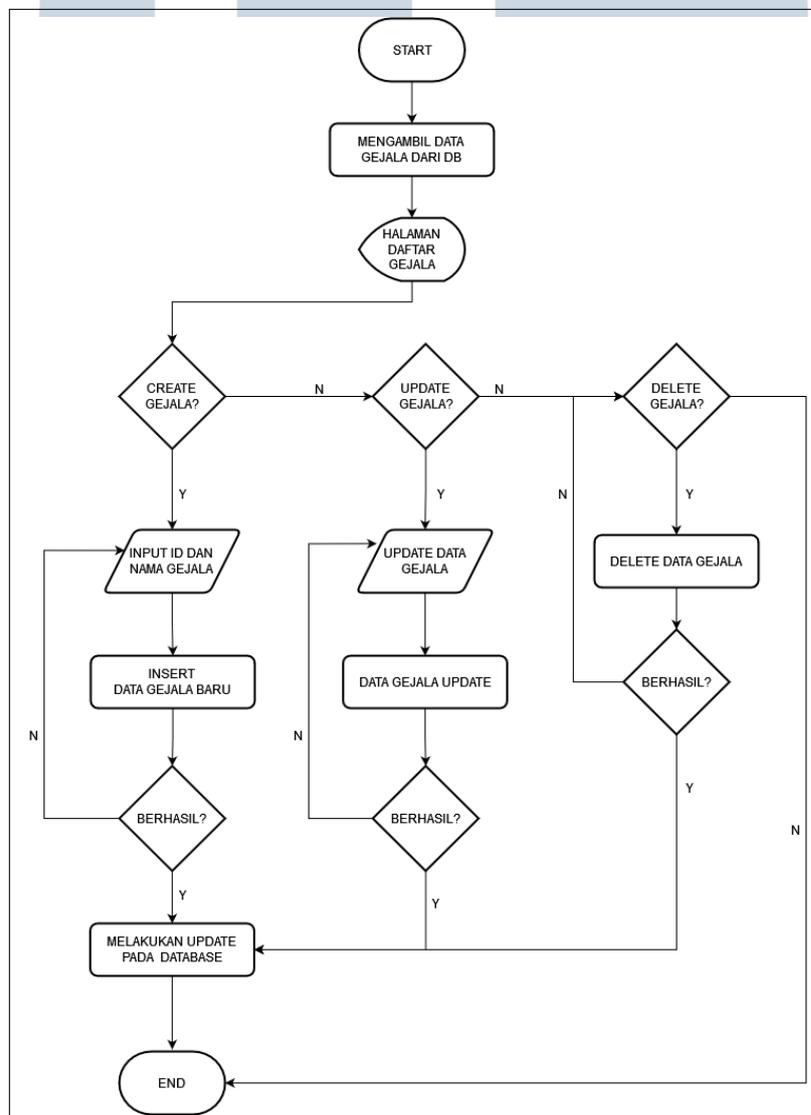
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Flowchart Admin

Gambar 3.6 merupakan gambar *flowchart admin* secara keseluruhan yang didalamnya terdapat sebuah *sidebar* yaitu, gejala, penyakit, berita, basis pengetahuan, rekomendasi makanan, riwayat diagnosa, akun dan pesan. Admin dapat melakukan perubahan terhadap isi data yang terdapat pada masing-masing menu. Berikut dibawah ini merupakan *flowchart* dari masing-masing menu yang terdapat pada *sidebar*.

C Flowchart Gejala (Admin)

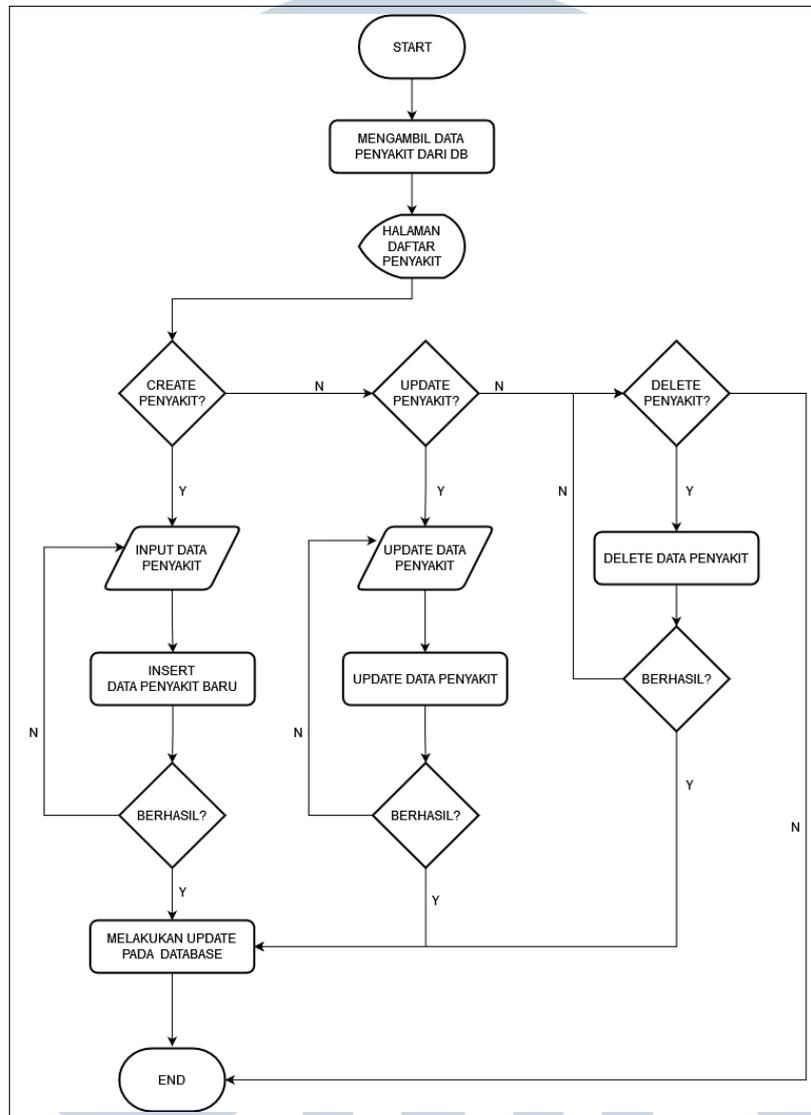


Gambar 3.7. *Flowchart Gejala Admin*

Gambar 3.7 merupakan urutan proses pada halaman gejala yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap gejala seperti *Create, Read, Update, Delete (CRUD)* data gejala pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar gejala melalui proses *Read* pada tabel gejala yang terdapat pada database. Admin dapat menambahkan data gejala dengan menginput data gejala yang ingin ditambahkan, kemudian data gejala yang ingin ditambahkan akan dimasukkan ke dalam tabel gejala pada database. Jika admin ingin melakukan *Update* terhadap data gejala, maka admin dapat mengubah data gejala yang lama dengan data gejala yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data gejala yang ada, admin dapat langsung memilihnya dan data gejala akan langsung di hapus pada tabel gejala di database.



D Flowchart Penyakit Admin

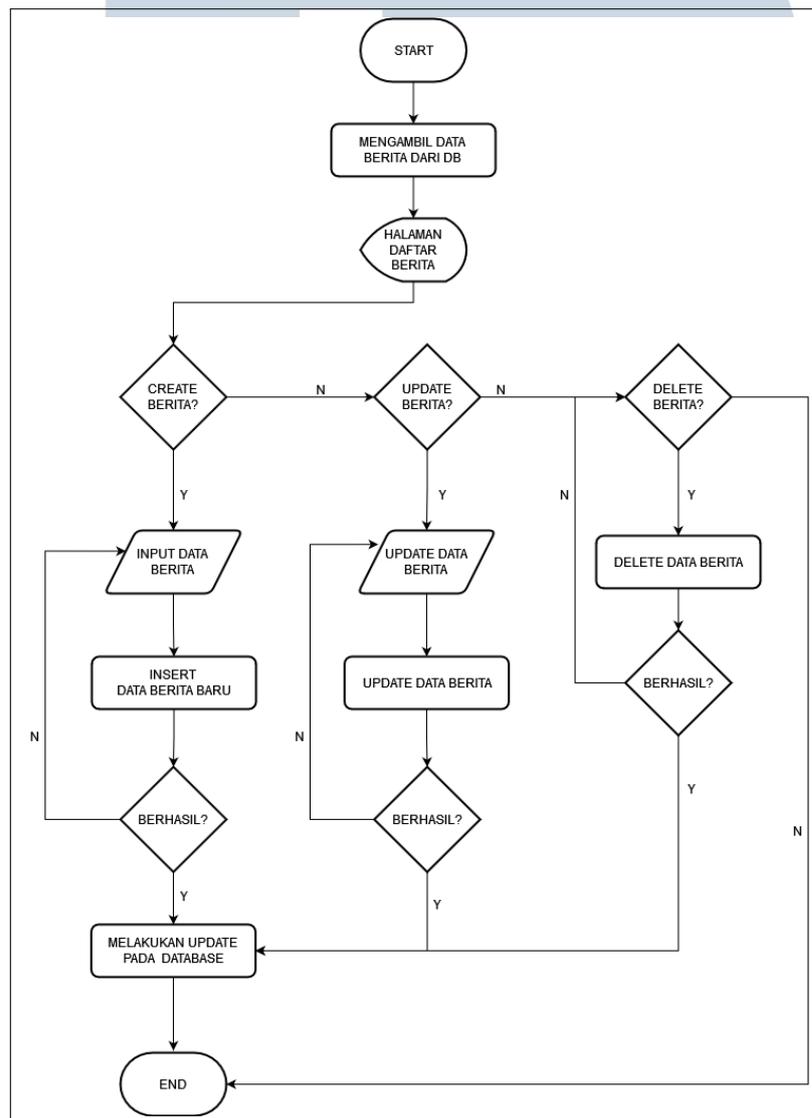


Gambar 3.8. Flowchart Penyakit Admin

Gambar 3.8 merupakan urutan proses pada halaman penyakit yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap penyakit seperti *Create*, *Read*, *Update*, *Delete* (*CRUD*) data penyakit pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar penyakit melalui proses *Read* pada tabel penyakit yang terdapat pada database. Admin dapat menambahkan data penyakit dengan menginput data penyakit yang ingin ditambahkan, kemudian data penyakit yang ingin ditambahkan akan dimasukkan ke dalam tabel penyakit pada database. Jika admin ingin melakukan *Update* terhadap

data penyakit, maka admin dapat mengubah data penyakit yang lama dengan data penyakit yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data penyakit yang ada, admin dapat langsung memilihnya dan data penyakit akan langsung di hapus pada tabel penyakit di database.

E Flowchart Berita Admin



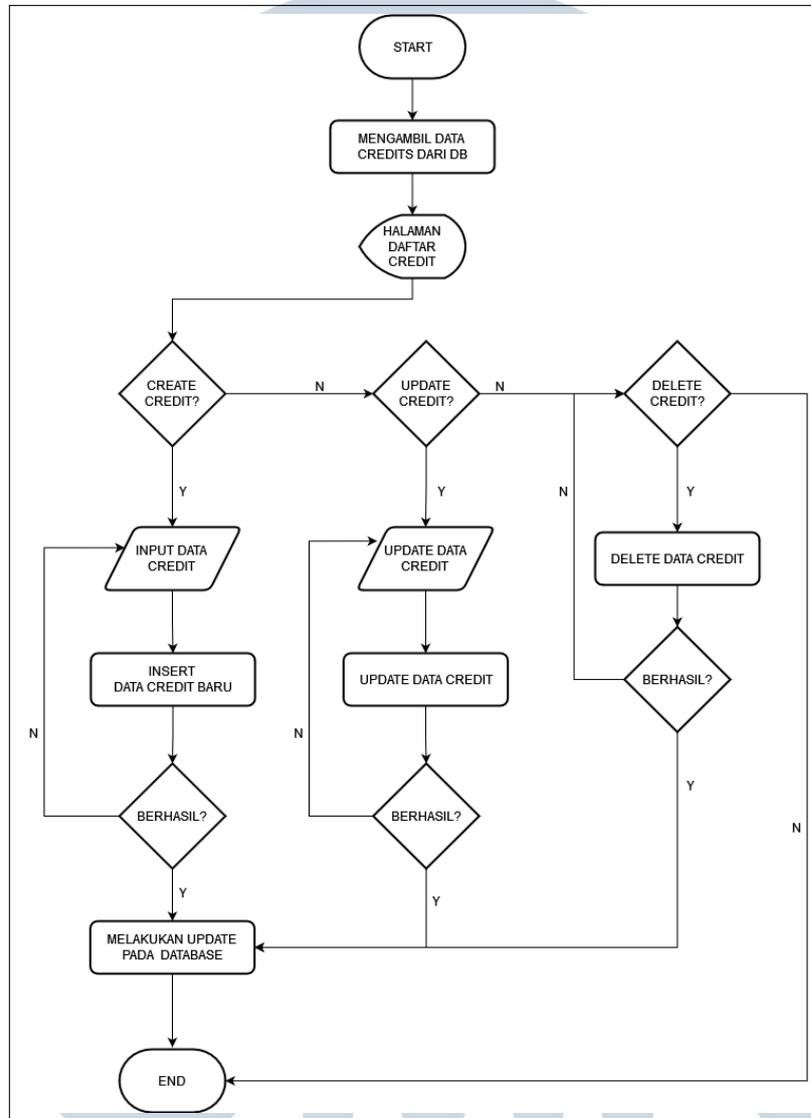
Gambar 3.9. Flowchart Berita Admin

Gambar 3.9 merupakan urutan proses pada halaman berita yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap berita seperti *Create*, *Read*, *Update*, *Delete* (*CRUD*) data berita pada sistem.

Pada halaman ini admin dapat melihat tabel yang berisikan daftar berita melalui proses *Read* pada tabel pengetahuan yang terdapat pada database. Admin dapat menambahkan data berita dengan menginput data berita yang ingin ditambahkan, kemudian data berita yang ingin ditambahkan akan dimasukkan ke dalam tabel pengetahuan pada database. Jika admin ingin melakukan *Update* terhadap data berita, maka admin dapat mengubah data berita yang lama dengan data berita yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data berita yang ada, admin dapat langsung memilihnya dan data berita akan langsung di hapus pada tabel berita di database.



F Flowchart Credit Admin

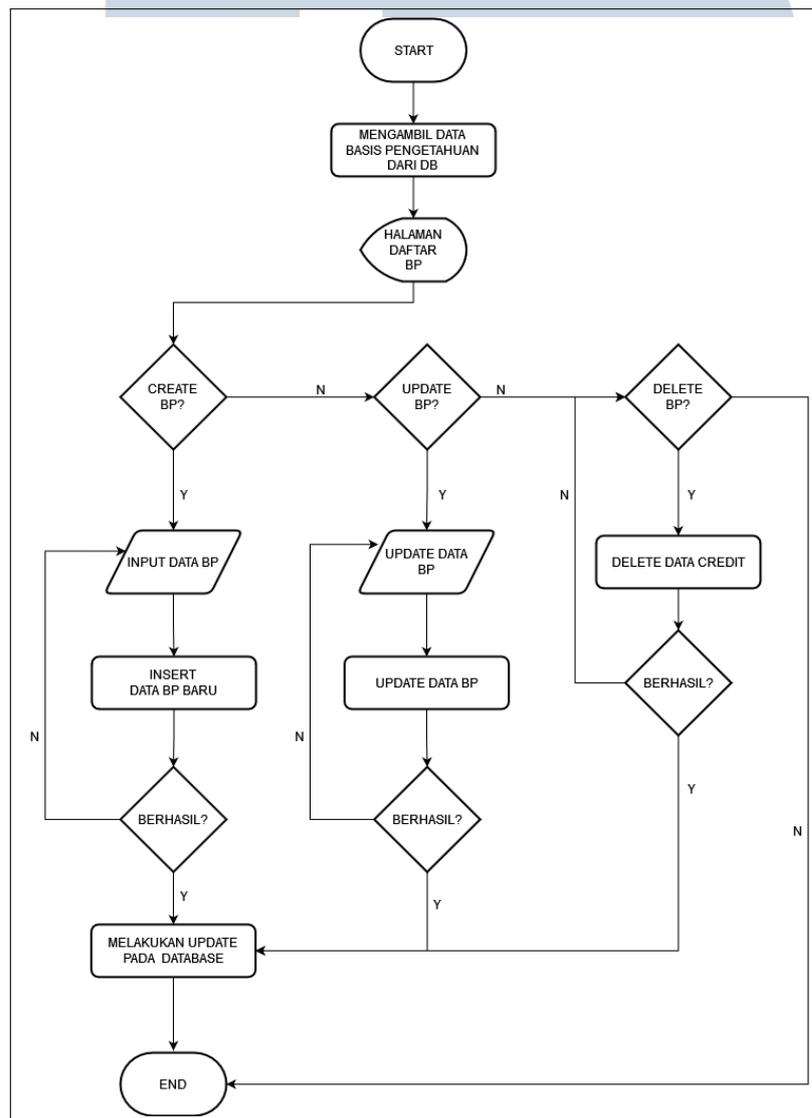


Gambar 3.10. Flowchart Credit Admin

Gambar 3.10 merupakan urutan proses pada halaman credit yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap credit seperti *Create*, *Read*, *Update*, *Delete* (*CRUD*) data credit pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar credit melalui proses *Read* pada tabel credit yang terdapat pada database. Admin dapat menambahkan data credit dengan menginput data credit yang ingin ditambahkan, kemudian data credit yang ingin ditambahkan akan dimasukkan ke dalam tabel credit pada database. Jika admin ingin melakukan *Update* terhadap data credit, maka admin

dapat mengubah data credit yang lama dengan data berita yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data credit yang ada, admin dapat langsung memilihnya dan data credit akan langsung di hapus pada tabel credit di database.

G Flowchart Basis Pengetahuan Admin



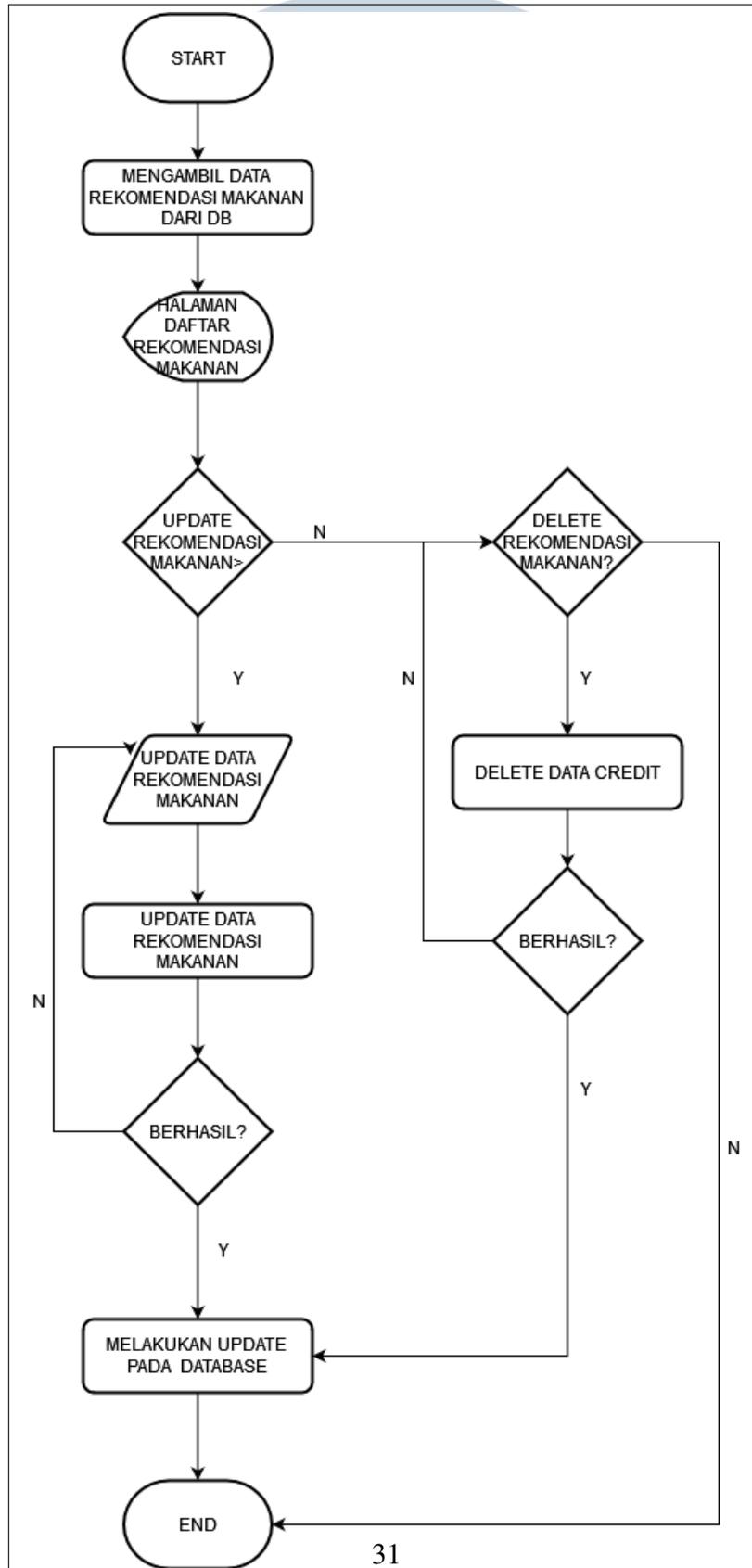
Gambar 3.11. Flowchart Basis Pengetahuan Admin

Gambar 3.11 merupakan urutan proses pada halaman basis pengetahuan yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap basis pengetahuan seperti *Create*, *Read*, *Update*, *Delete*

(*CRUD*) data basis pengetahuan pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar basis pengetahuan melalui proses *Read* pada tabel basis pengetahuan yang terdapat pada database. Admin dapat menambahkan data basis pengetahuan dengan menginput data basis pengetahuan yang ingin ditambahkan, kemudian data basis pengetahuan yang ingin ditambahkan akan dimasukkan ke dalam tabel basis pengetahuan pada database. Jika admin ingin melakukan *Update* terhadap data basis pengetahuan, maka admin dapat mengubah data basis pengetahuan yang lama dengan data basis pengetahuan yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data basis pengetahuan yang ada, admin dapat langsung memilihnya dan data basis pengetahuan akan langsung di hapus pada tabel basis pengetahuan di database.



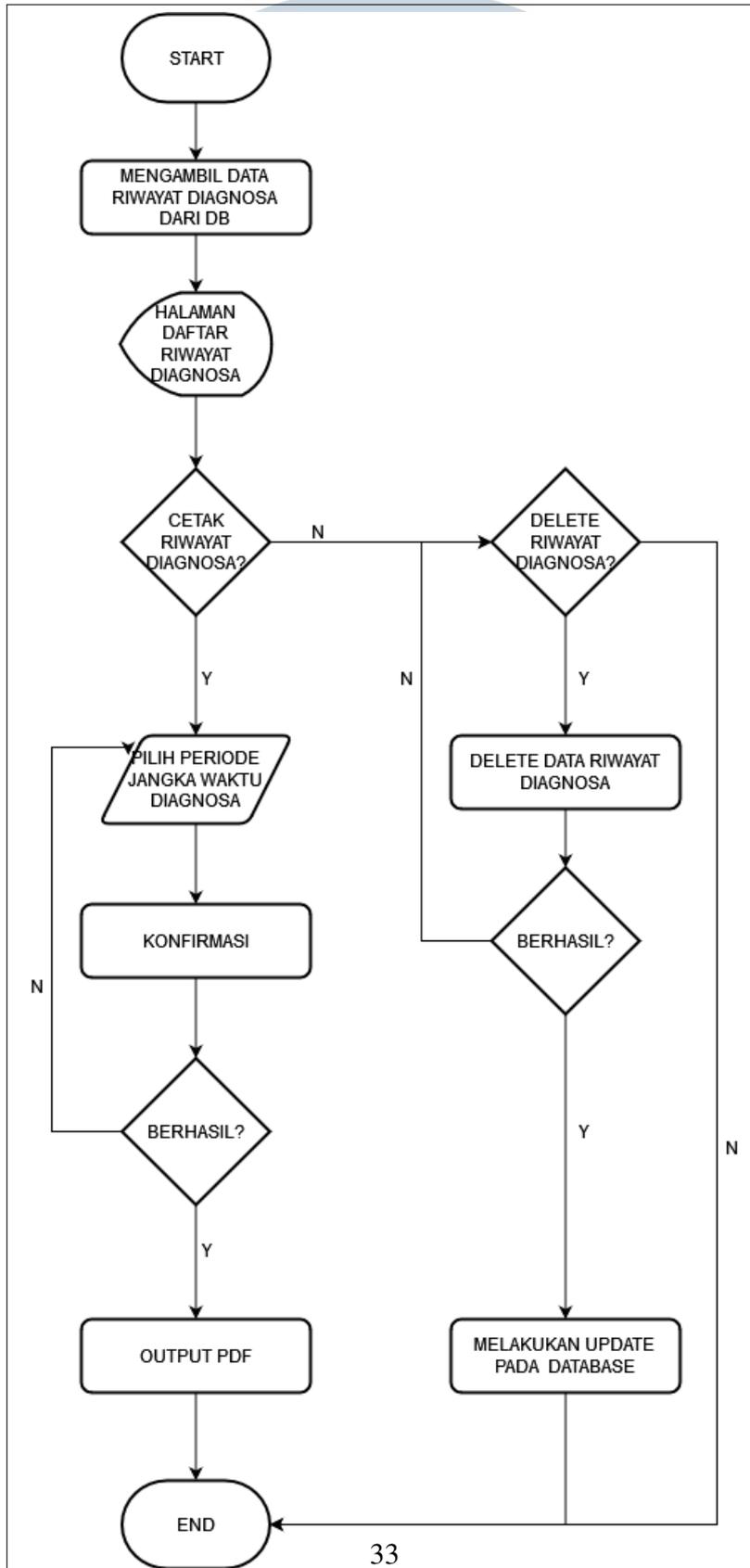
H Flowchart Rekomendasi Makanan Admin



Gambar 3.12 merupakan urutan proses pada halaman rekomendasi makanan yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap basis pengetahuan seperti *Read, Update, Delete (RUD)* data credit pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar rekomendasi makanan melalui proses *Read* pada tabel *Food Recommendation* yang terdapat pada database. Jika admin ingin melakukan *Update* terhadap data rekomendasi makanan, maka admin dapat mengubah data rekomendasi makanan yang lama dengan rekomendasi makanan yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data rekomendasi makanan yang ada, admin dapat langsung memilihnya dan data rekomendasi makanan akan langsung di hapus pada tabel *food recommendation* di database.



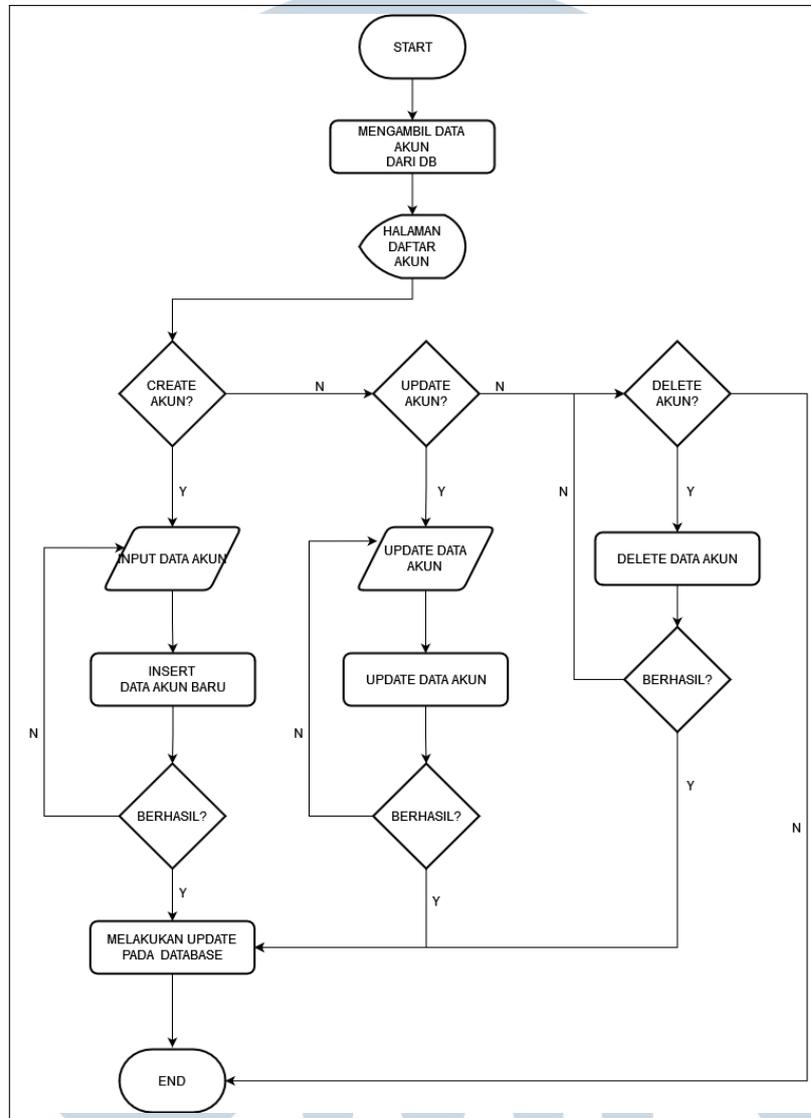
I Flowchart Riwayat DiagnosaAdmin



Gambar 3.13 merupakan urutan proses pada halaman riwayat diagnosa yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap basis pengetahuan seperti mencetak hasil diagnosa sesuai dengan periode waktu yang ditentukan dan melakukan *Delete* terhadap data riwayat diagnosa pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar riwayat diagnosa melalui proses *Read* pada tabel diagnosa yang terdapat pada database. Jika admin ingin melakukan *Delete* pada data riwayat diagnosa, maka admin dapat langsung memilihnya dan menghapus data tersebut yang nantinya akan diupdate pada database.



J Flowchart Akun Admin

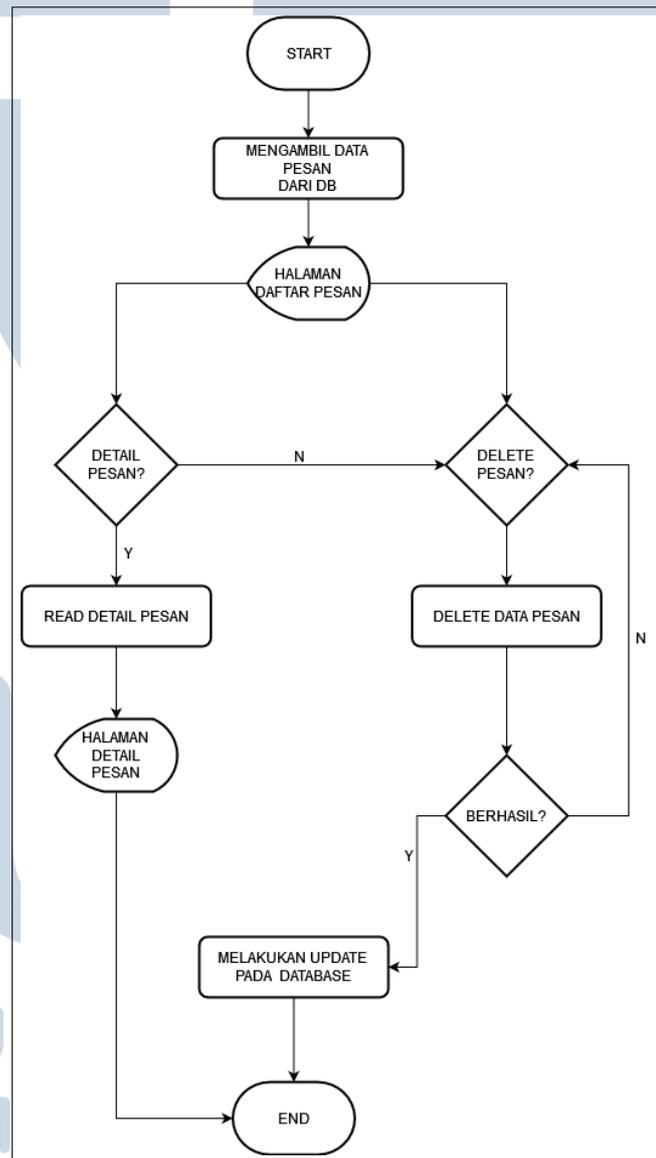


Gambar 3.14. Flowchart Akun Admin

Gambar 3.14 merupakan urutan proses pada halaman akun yang dapat dilakukan oleh admin dalam sistem. Admin dapat melakukan manajemen terhadap akun seperti *Create*, *Read*, *Update*, *Delete* (CRUD) data akun pada sistem. Pada halaman ini admin dapat melihat table yang berisikan daftar akun melalui proses *Read* pada tabel *users* yang terdapat pada database. Admin dapat menambahkan data akun dengan menginput data akun yang ingin ditambahkan, kemudian data akun yang ingin ditambahkan akan dimasukkan ke dalam tabel *users* pada database. Jika admin ingin melakukan *Update* terhadap data akun, maka admin dapat

mengubah data akun yang lama dengan data akun yang baru. Setelah itu admin juga dapat melakukan *Delete* pada data akun yang ada, admin dapat langsung memilihnya dan data akun akan langsung di hapus pada tabel *users* di database.

K Flowchart Pesan Admin

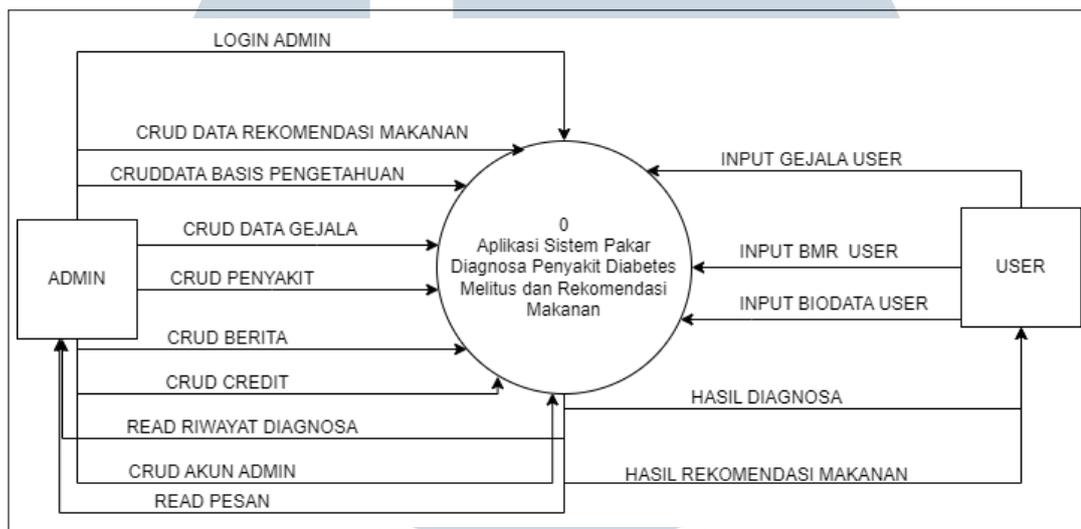


Gambar 3.15. Flowchart Pesan Admin

Gambar 3.15 merupakan urutan proses pada halaman pesan yang dapat dilakukan oleh admin dalam sistem. Pada halaman ini admin dapat melihat tabel yang berisikan data pesan melalui proses *Read* pada tabel pesan yang terdapat pada

database. Admin dapat melihat isi detail dari isi pesan tersebut dan admin juga dapat menghapus pesan tersebut yang nantinya akan dihapus oleh sistem sesuai dengan data yang dipilih oleh admin.

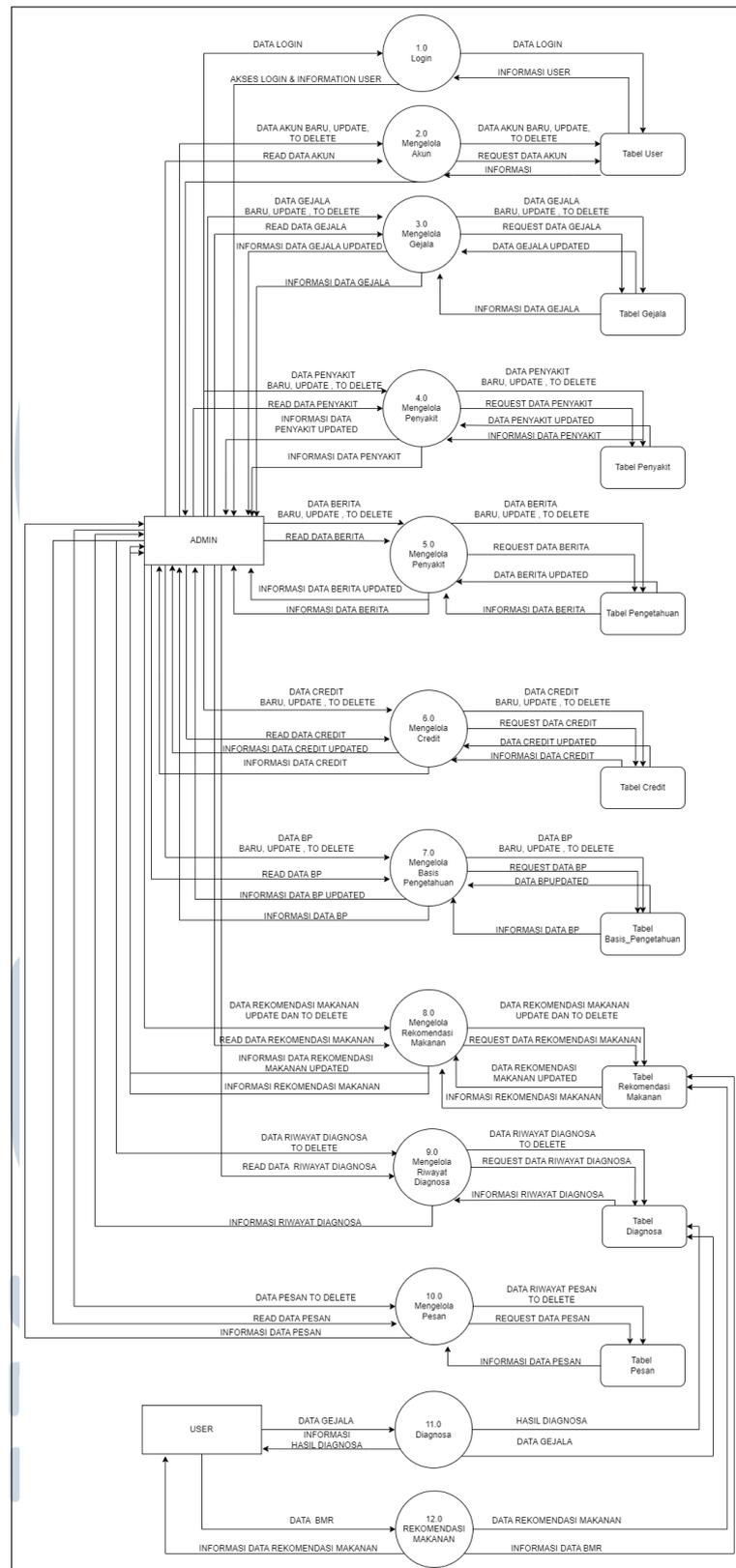
3.2.5 Data Flow Diagram



Gambar 3.16. Data Flow Diagram Level 0

Gambar 3.16 merupakan *data flow diagram level 0* dari sistem yang telah dibangun. Pada diagram ini terdapat sisi admin dan juga pengguna. Pengguna dapat langsung mengakses ke sistem pakar tanpa melakukan *login* terlebih dahulu, pengguna dapat melakukan diagnosa terhadap penyakit Diabetes Melitus dengan memilih gejala-gejala yang telah diberikan oleh pakar pada sistem dan dapat mendapatkan rekomendasi makanan berdasarkan dengan *BMR* yang telah dihitung. Selain itu pengguna juga dapat melihat informasi penyakit, berita, *credits* dan *contact*.

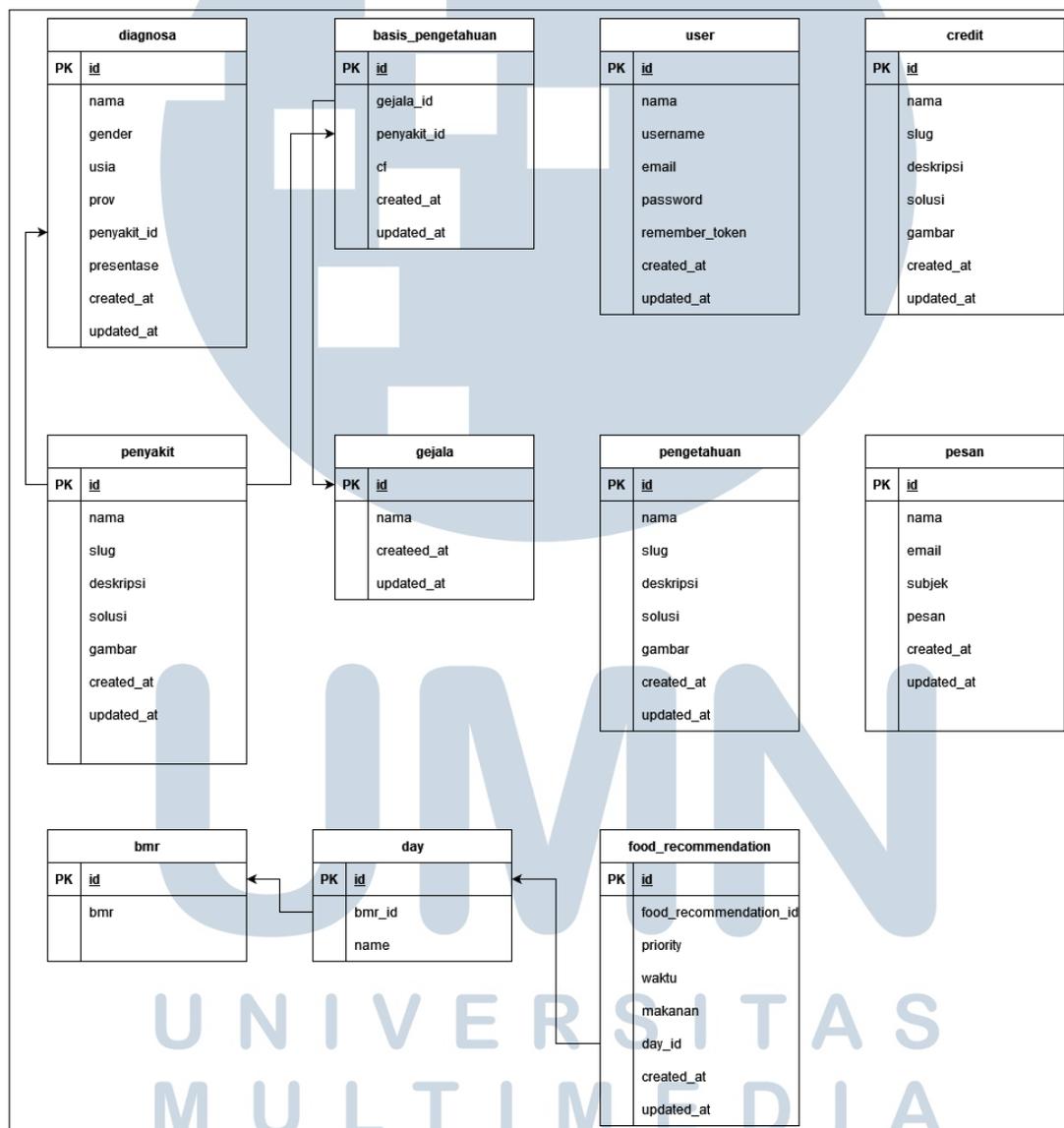
Kemudian selanjutnya adalah sisi admin. Untuk masuk kedalam halaman admin, admin harus melakukan *login* terlebih dahulu. Jika login yang dilakukan telah berhasil maka admin langsung ditunjukkan kehalaman *dashboard* admin, kemudian terdapat *sidebar* yang berisikan menu-menu yang dapat diatur oleh admin seperti gejala, penyakit, berita, credit, basis pengetahuan, rekomendasi makanan, riwayat diagnosa, akun dan pesan. Pada masing-masing menu terdapat kolom aksi yang dapat digunakan admin untuk melakukan *Create, Read, Update, Delete* pada data.



Gambar 3.17. Data Flow Diagram Level 1

Gambar 3.17 merupakan *Data Flow Diagram Level 1* yang merupakan lanjutan dari *Diagram level 0* yang berisikan tentang penjelasan aliran data yang terjadi pada sistem pakar yang telah dibangun.

3.2.6 Table Relationship Diagram



Gambar 3.18. *Entity Relationship Diagram (ERD)*

Gambar 3.18 merupakan hubungan relasi antara setiap tabel yang digunakan dalam *database*.

3.2.7 Struktur Tabel

Berikut dibawah ini merupakan 11 (sebelas) buah struktur tabel yang terdapat pada *Entity Relationship Diagram (ERD)* yang digunakan aplikasi sistem diagnosa penyakit Diabetes Melitus dan Rekomendasi Makanan menggunakan metode certainty factory yang dapat kita lihat pada gambar 3.16 yaitu, tabel diagnosa, penyakit, basis pengetahuan, gejala, pengetahuan, user, credit, pesan, bmr, day dan *food recommendation*.

A Tabel Diagnosa

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key, Auto Increment Menyimpan seluruh id diagnosa
2	nama penderita	varchar	100	Unique Menyimpan nama dari user
3	gender	varchar	100	Unique Menyimpan jenis kelamin dari user
4	usia	varchar	20	Menyimpan usia dari user
5	provinsi	longtext		Unique Menyimpan provinsi dari user
6	penyakit_id	char	4	Menyimpan id penyakit
7	presentase	double	8,2	Menyimpan data presentasi
8	created_at	timestamp		Menyimpan tanggal dibuat
9	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.1. Struktur Tabel Diagnosa

Tabel 3.1 merupakan struktur dari tabel diagnosa. Tabel ini berfungsi untuk menyimpan data-data dari hasil diagnosa yang telah dilakukan oleh *user* dalam menganalisa gejala yang dialami untuk menentukan presentase penyakit Diabetes Melitus. Tabel ini juga berfungsi untuk menampilkan daftar data hasil diagnosa yang telah dilakukan oleh *user* pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel diagnosa :

- ID merupakan Primary Key dari tabel diagnosa yang tipe datanya bigint dengan panjang data sebanyak 20 (dua puluh) dan memiliki Extra AUTO INCREMENT. Kolom ini berfungsi untuk menyimpan seluruh id diagnosa.
- Nama Penderita merupakan nama dari *user* atau pasien yang telah melakukan diagnosa, tipe datanya varchar dan merupakan dengan panjang data sebanyak

100 (seratus). Kolom ini berfungsi untuk menyimpan nama dari *user* atau pasien.

- Gender merupakan jenis kelamin dari *user* atau pasien yang telah melakukan diagnosa, tipe datanya *varchar* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan jenis kelamin dari *user* atau pasien.
- Usia merupakan usia dari *user* atau pasien yang telah melakukan diagnosa, tipe datanya *integer* dengan panjang data sebanyak 11 (sebelas). Kolom ini berfungsi untuk menyimpan usia dari *user* atau pasien.
- Provinsi merupakan provinsi dari *user* atau pasien yang telah melakukan diagnosa, tipe datanya *varchar* dan merupakan *Unique Key* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan usia dari *user* atau pasien.
- Penyakit Id merupakan *foreign key* yang menunjuk pada tabel penyakit. Kolom ini berfungsi untuk menyimpan data penyakit yang sudah ada.
- Presentase merupakan Presentase dari hasil diagnosa yang telah dilakukan *user* atau pasien. Presentase memiliki tipe data *double* agar dapat menyimpan nilai numerik dengan titik desimal yang memiliki panjang data 8 digit dengan 2 digit dibelakang koma. Kolom ini berfungsi untuk menyimpan data presentase dari hasil diagnosa yang telah dilakukan.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya diagnosa dan memiliki tipe data *timestamp*.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya diagnosa dan memiliki tipe data *timestamp*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

B Tabel Penyakit

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	char	4	Primary Key, Menyimpan seluruh id penyakit
2	nama	varchar	255	Unique Menyimpan nama dari penyakit
3	slug	varchar	255	Menyimpan link dari penyakit
4	deskripsi	longtext		Unique Menyimpan deskripsi dari penyakit
5	solusi	longtext		Unique Menyimpan solusi dari penyakit
6	gambar	varchar	255	Unique Menyimpan gambar dari penyakit
7	created_at	timestamp		Menyimpan tanggal dibuat
8	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.2. Struktur Tabel Penyakit

Tabel 3.2 merupakan struktur dari tabel penyakit. Tabel ini berfungsi untuk menyimpan data-data penyakit. Tabel ini juga berfungsi untuk menampilkan daftar data penyakit pada halaman admin serta ditampilkan pada halaman diagnosa setelah selesai melakukan analisa. Berikut dibawah ini merupakan data yang disimpan oleh tabel penyakit :

- ID merupakan Primary Key dari tabel penyakit yang tipe datanya char dengan panjang data sebanyak 4 (empat). Kolom ini berfungsi untuk menyimpan seluruh id penyakit.
- Nama merupakan nama dari penyakit yang telah ditambahkan, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama penyakit.
- slug merupakan slug dari penyakit yang tipe datanya varchar dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan link penyakit.
- Deskripsi merupakan deskripsi dari penyakit yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan deskripsi penyakit.

- Solusi merupakan solusi dari penyakit yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan solusi penyakit.
- Gambar merupakan gambar dari penyakit yang telah ditambahkan. Tipe datanya *varchar* dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan gambar dari penyakit.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya penyakit dan memiliki tipe data *timestamp*.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya penyakit dan memiliki tipe data *timestamp*.

C Tabel Basis Pengetahuan

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key, Auto Increment Menyimpan seluruh id dari basis pengetahuan
2	gejala_id	char	4	Unique Menyimpan id dari gejala
3	penyakit_id	char	4	Unique Menyimpan id dari penyakit
4	CF	double	8,2	Menyimpan nilai keyakinan dari CF
5	created_at	timestamp		Menyimpan tanggal dibuat
6	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.3. Struktur Tabel Basis Pengetahuan

Tabel 3.3 merupakan struktur dari tabel basis pengetahuan. Tabel ini berfungsi untuk menyimpan data-data basis pengetahuan seperti data nilai CF yang telah diberikan oleh pakar atau dokter spesialis. Tabel ini juga berfungsi untuk menampilkan daftar data basis pengetahuan yang pada halaman admin serta ditampilkan pada halaman diagnosa pada saat user ingin melakukan diagnosa. Berikut dibawah ini merupakan data yang disimpan oleh tabel basis pengetahuan :

- ID merupakan Primary Key dari tabel basis pengetahuan yang tipe datanya *bigint* dengan panjang data sebanyak 20 (dua puluh) dan memiliki Extra

AUTO INCREMENT. Kolom ini berfungsi untuk menyimpan seluruh id dari basis pengetahuan.

- Gejala id merupakan id gejala yang datanya diambil pada pada tabel gejala, tipe datanya char dan merupakan *Unique Key* dengan panjang data sebanyak 4 (empat). Kolom ini berfungsi untuk menyimpan id gejala.
- Penyakit id merupakan id penyakit yang datanya diambil pada tabel penyakit, tipe datanya char dan merupakan *Unique Key* dengan panjang data sebanyak 4 (empat). Kolom ini berfungsi untuk menyimpan id penyakit.
- CF merupakan nilai keyakinan CF yang telah diberikan oleh seorang pakar atau dokter spesialis. Tipe datanya *double* agar dapat menyimpan nilai numerik dengan titik desimal yang memiliki panjang data 8 digit dengan 2 digit dibelakang koma. Kolom ini berfungsi untuk menyimpan data nilai keyakinan CF.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya basis pengetahuan dan memiliki tipe data timestamp.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya basis pengetahuan dan memiliki tipe data timestamp.

D Tabel Gejala

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	char	4	Primary Key, Unique Menyimpan seluruh id dari gejala
2	nama	varchar	255	Unique Menyimpan id dari gejala
3	created_at	timestamp		Menyimpan tanggal dibuat
4	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.4. Struktur Tabel Basis Pengetahuan

Tabel 3.4 merupakan struktur dari tabel gejala. Tabel ini berfungsi untuk menyimpan data-data gejala. Tabel ini juga berfungsi untuk menampilkan daftar data gejala pada halaman admin serta ditampilkan pada halaman diagnosa pada saat ingin melakukan diagnosa. Berikut dibawah ini merupakan data yang disimpan oleh tabel gejala :

- ID merupakan Primary Key dari tabel gejala yang tipe datanya char dengan panjang data sebanyak 4 (empat) dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan seluruh id dari gejala.
- Nama merupakan nama dari gejala yang telah ditambahkan, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama gejala.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya gejala dan memiliki tipe data timestamp.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya gejala dan memiliki tipe data timestamp.

E Tabel Pesan

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key, Auto Increment Menyimpan seluruh id dari pesan
2	nama	varchar	100	Unique Menyimpan nama dari pengirim pesan
3	email	varchar	100	Unique Menyimpan email pengirim pesan
4	subjek	varchar	100	Unique Menyimpan subjek pengirim pesan
5	pesan	longtext		Unique Menyimpan isi pesan
6	created_at	timestamp		Menyimpan tanggal dibuat
7	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.5. Struktur Tabel Pesan

Tabel 3.5 merupakan struktur dari tabel pesan. Tabel ini berfungsi untuk menyimpan data-data pesan yang telah dikirim oleh *user*. Tabel ini juga berfungsi untuk menampilkan daftar data pesan pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel pesan :

- ID merupakan Primary Key dari tabel pesan yang tipe datanya bigint dengan panjang data sebanyak 20 (dua puluh) dan memiliki Extra AUTO INCREMENT. Kolom ini berfungsi untuk menyimpan seluruh id pesan.

- Nama merupakan nama dari pengirim pesan, tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan nama pengirim pesan.
- Email merupakan email dari pengirim pesan, tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan email pengirim pesan.
- Subjek merupakan subjek yang ditulis oleh pengirim pesan, tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan subjek yang ditulis oleh pengirim pesan.
- Pesan merupakan isi pesan dari pengirim pesan, tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan isi pesan yang telah dikirim oleh pengirim pesan.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya pesan dan memiliki tipe data timestamp.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya pesan dan memiliki tipe data timestamp.



F Tabel Credit

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	char	4	Primary Key Menyimpan seluruh id dari credit
2	nama	varchar	255	Unique Menyimpan nama dari credit
3	slug	varchar	255	Menyimpan link data credit
4	deskripsi	longtext		Unique Menyimpan deskripsi dari credit
5	solusi	longtext		Unique Menyimpan solusi dari credit
6	gambar	varchar	255	Unique Menyimpan gambar dari credit
7	created_at	timestamp		Menyimpan tanggal dibuat
8	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.6. Struktur Tabel Credit

Tabel 3.6 merupakan struktur dari tabel credit. Tabel ini berfungsi untuk menyimpan data-data credit. Tabel ini juga berfungsi untuk menampilkan daftar data credit pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel credit :

- ID merupakan Primary Key dari tabel credit yang tipe datanya char dengan panjang data sebanyak 4 (empat). Kolom ini berfungsi untuk menyimpan seluruh id credit.
- Nama merupakan nama dari data credit yang telah ditambahkan, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama data credits.
- slug merupakan slug dari credit yang tipe datanya varchar dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan link data credit.
- Deskripsi merupakan deskripsi dari data credit yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan deskripsi dari data credit.

- Solusi merupakan solusi dari data credit yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan solusi dari data credit.
- Gambar merupakan gambar dari data credit yang telah ditambahkan. Tipe datanya *varchar* dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan gambar dari data credit.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya credit dan memiliki tipe data *timestamp*.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya credit dan memiliki tipe data *timestamp*.

G Tabel Pengetahuan

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	char	4	Primary Key Menyimpan seluruh id dari pengetahuan
2	nama	varchar	255	Unique Menyimpan nama dari pengetahuan
3	slug	varchar	255	Menyimpan link data pengetahuan
4	deskripsi	longtext		Unique Menyimpan deskripsi dari pengetahuan
5	solusi	longtext		Unique Menyimpan solusi dari pengetahuan
6	gambar	varchar	255	Unique Menyimpan gambar dari pengetahuan
7	created_at	timestamp		Menyimpan tanggal dibuat
8	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.7. Struktur Tabel Pengetahuan

Tabel 3.7 merupakan struktur dari tabel pengetahuan. Tabel ini berfungsi untuk menyimpan data-data pengetahuan. Tabel ini juga berfungsi untuk menampilkan daftar data pengetahuan pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel pengetahuan :

- ID merupakan Primary Key dari tabel pengetahuan yang tipe datanya char dengan panjang data sebanyak 4 (empat). Kolom ini berfungsi untuk menyimpan seluruh id pengetahuan.
- Nama merupakan nama dari data pengetahuan yang telah ditambahkan, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama data pengetahuan.
- slug merupakan slug dari pengetahuan yang tipe datanya varchar dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan link data pengetahuan.
- Deskripsi merupakan deskripsi dari data pengetahuan yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan deskripsi dari data pengetahuan.
- Solusi merupakan solusi dari data pengetahuan yang telah ditambahkan. Tipe datanya *longtext* dan merupakan *Unique Key*. Kolom ini berfungsi untuk menyimpan solusi dari data pengetahuan.
- Gambar merupakan gambar dari data pengetahuan yang telah ditambahkan. Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan gambar dari data pengetahuan.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya pengetahuan dan memiliki tipe data timestamp.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya pengetahuan dan memiliki tipe data timestamp.

H Tabel *Users*

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	bigint	20	Primary Key, Auto Increment Menyimpan seluruh id dari user

2	nama	varchar	255	Unique Menyimpan nama dari user
3	username	varchar	255	Menyimpan username user
4	email	varchar	255	Unique Menyimpan email dari user
5	password	varchar	255	Unique Menyimpan password dari user
6	remember_token	varchar	100	Unique Menyimpan token
7	created_at	timestamp		Menyimpan tanggal dibuat
8	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.8. Struktur Tabel Users

Tabel 3.8 merupakan struktur dari tabel *users*. Tabel ini berfungsi untuk menyimpan data-data *users*. Tabel ini juga berfungsi untuk menampilkan daftar data *users* pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel *users* :

- ID merupakan Primary Key dari tabel *users* yang tipe datanya bigint dengan panjang data sebanyak 20 (dua puluh) dan memiliki Extra AUTO INCREMENT. Kolom ini berfungsi untuk menyimpan seluruh id *users*.
- Nama merupakan nama dari *users* yang telah ditambahkan, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama *users*.
- Username merupakan username yang telah dibuat oleh *users*, Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan username dari *users*.
- Email merupakan email dari *users*, tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan email dari *users*.
- Password merupakan password yang telah dibuat *users* dalam pembuatan akun. Tipe datanya varchar dan merupakan *Unique Key* dengan panjang

data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan password dari *users*.

- Remember Token merupakan sebuah token yang diperlukan untuk mengingat riwayat login *users*. Tipe datanya varchar dan merupakan *Unique Key* dengan panjang data sebanyak 100 (seratus). Kolom ini berfungsi untuk menyimpan token dari *users*.
- Created at merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya *users* dan memiliki tipe data timestamp.
- Updated at merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya *users* dan memiliki tipe data timestamp.

I Tabel *Food Recommendation*

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	int	11	Primary Key, Auto Increment Menyimpan seluruh id food_recommendation
2	food_recommendation_id	int	11	Unique Menyimpan nama dari user
3	priority	int	11	Menyimpan priority
4	waktu	varchar	255	Unique Menyimpan waktu rekomendasi makanan
5	makanan	longtext		Unique Menyimpan makanan
6	day_id	int	11	Unique Menyimpan id hari
7	created_at	timestamp		Menyimpan tanggal dibuat
8	updated_at	timestamp		Menyimpan tanggal diupdate

Tabel 3.9. Struktur Tabel Food Recommendation

Tabel 3.9 merupakan struktur dari tabel *food recommendation*. Tabel ini berfungsi untuk menyimpan data-data makanan yang akan direkomendasikan.

Tabel ini juga berfungsi untuk menampilkan daftar data makanan pada halaman admin. Berikut dibawah ini merupakan data yang disimpan oleh tabel *food recommendation* :

- ID merupakan Primary Key dari tabel *food recommendation* yang tipe datanya int dengan panjang data sebanyak 11 (sebelas) dan memiliki Extra AUTO INCREMENT. Kolom ini berfungsi untuk menyimpan seluruh id *food recommendation*.
- *food recommendation id* merupakan id yang digunakan untuk menghubungkannya kepada tabel *bmr* sehingga dapat menyesuaikan *bmr* dengan makanan yang akan direkomendasikan, tipe datanya int dengan panjang data sebanyak 11 (sebelas). Kolom ini berfungsi untuk menyimpan seluruh *food recommendation id*.
- Priority merupakan priority terhadap makanan yang akan direkomendasikan, tipe datanya int dengan panjang data sebanyak 11 (sebelas). Kolom ini berfungsi untuk menyimpan priority dari rekomendasi makanan.
- Waktu merupakan waktu yang akan ditampilkan pada rekomendasi makanan berdasarkan prioritynya, tipe datanya varchar dengan panjang data sebanyak 255 (dua ratus lima puluh lima). Kolom ini berfungsi untuk menyimpan nama waktu yang akan digunakan.
- Makanan merupakan daftar makanan yang akan direkomendasikan sesuai dengan hasil perhitungan *bmr* yang ada, tipe datanya longtext agar dapat memberikan rekomendasi makanan yang cukup banyak.
- *day id* merupakan id dari hari yang akan digunakan dalam rekomendasi makanan, tipe datanya adalah int dengan panjang data sebanyak 11 (sebelas).
- *Created at* merupakan kolom yang berfungsi untuk menyimpan tanggal dibuatnya *food recommendation* dan memiliki tipe data timestamp.
- *pdated at* merupakan kolom yang berfungsi untuk menyimpan tanggal diupdatenya *food recommendation* dan memiliki tipe data timestamp.

J Tabel BMR

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	int	11	Primary Key Menyimpan seluruh id BMR
2	bmr	int	11	Menyimpan data BMR

Tabel 3.10. Struktur Tabel BMR

Tabel 3.10 merupakan struktur dari tabel bmr. Tabel ini berfungsi untuk menyimpan data bmr yang akan digunakan dalam rekomendasi makanan, tabel ini hanya menyimpan 2 (dua) buah data yaitu, id dengan tipe data int yang panjang datanya sebanyak 11 (sebelas) dan bmr dengan tipe data int yang panjang datanya sebanyak 11 (sebelas).

K Tabel Day

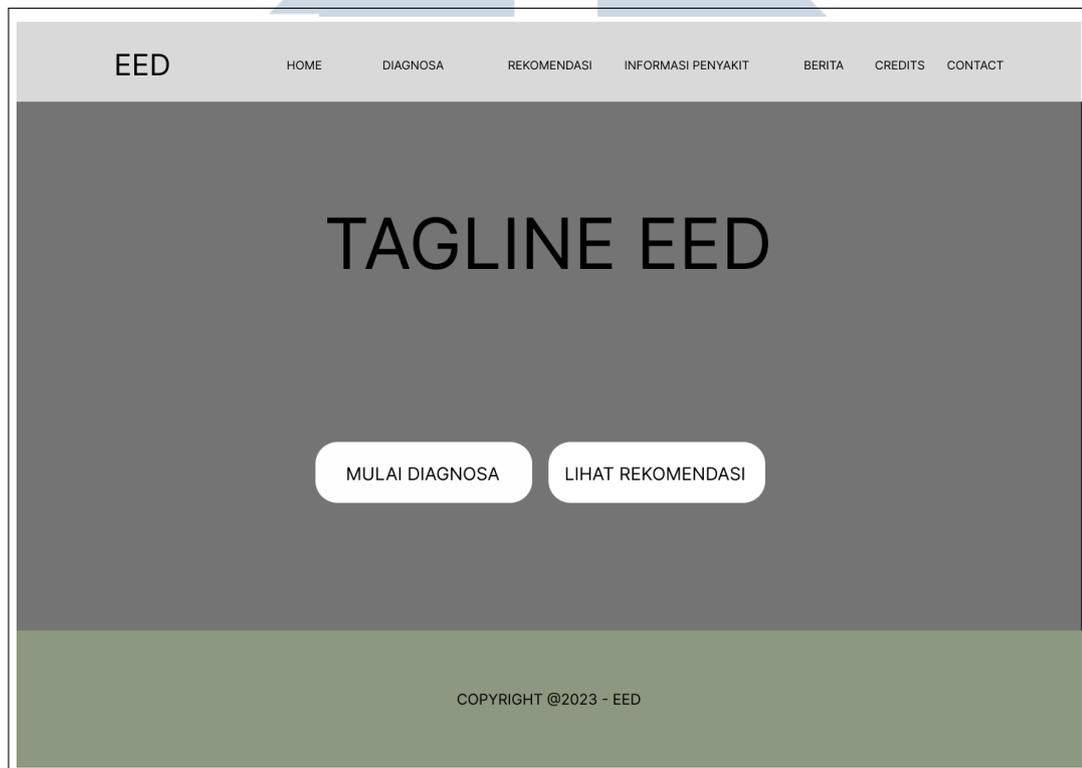
No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	int	11	Primary Key Menyimpan seluruh id day
2	bmr_id	int	11	Menyimpan data BMR
3	name	varchar	255	Menyimpan nama hari

Tabel 3.11. Struktur Tabel Day

Tabel 3.11 merupakan struktur dari tabel *day*. tabel ini berfungsi untuk menyimpan data hari yang akan digunakan dalam rekomendasi makanan, tabel ini hanya menyimpan 3 (tiga) buah data yaitu, id dengan tipe data int yang panjang datanya sebanyak 11 (sebelas), bmr_id dengan tipe data int yang panjang datanya sebanyak 11 (sebelas) dan *name* dengan tipe data varchar yang panjang datanya sebanyak 255 (dua ratus lima puluh lima).

3.2.8 Low Fidelity Mockup User

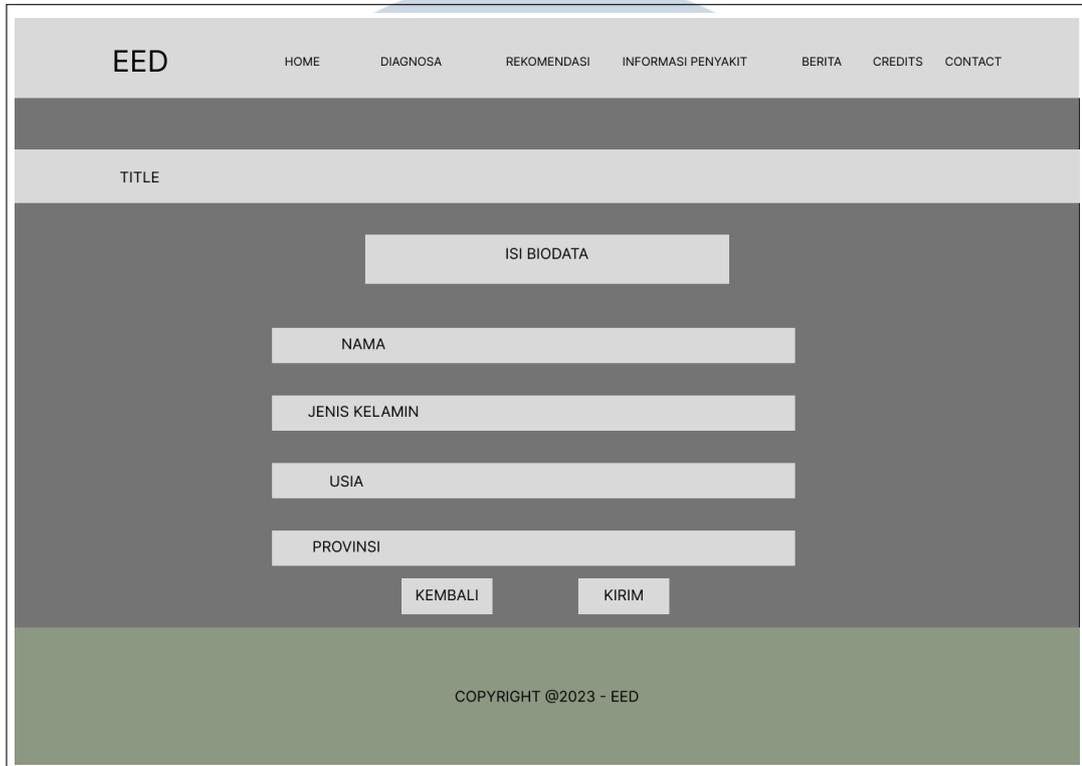
A Low Fidelity Mockup Halaman Home



Gambar 3.19. Low Fidelity Mockup Halaman Home

Gambar 3.19 merupakan *Mockup* dari halaman home. Dalam bagian ini terdapat 1 buah *Hero Section* yang berisikan tagline dan terdapat 2 tombol yang bertujuan untuk membantu *user* agar dapat langsung berpindah ke halaman yang diinginkan. Tombol yang terdapat pada *Hero Section* yaitu, tombol diagnosa yang akan langsung memindahkan user ke halaman diagnosa dan tombol rekomendasi yang akan langsung memindahkan user ke halaman rekomendasi untuk mendapatkan rekomendasi makanan. Selain itu, pada bagian atas halaman terdapat sebuah *Navigation Bar* yang berfungsi untuk menavigasi *user* ke halaman yang diinginkannya.

B *Low Fidelity Mockup* Halaman Diagnosa



The mockup shows a web page for a diagnostic system. At the top is a navigation bar with the logo 'EED' and menu items: HOME, DIAGNOSA, REKOMENDASI, INFORMASI PENYAKIT, BERITA, CREDITS, and CONTACT. Below the navigation bar is a header area with the word 'TITLE'. The main content area is a dark grey rectangle containing a form titled 'ISI BIODATA'. The form has four input fields: 'NAMA', 'JENIS KELAMIN', 'USIA', and 'PROVINSI'. Below these fields are two buttons: 'KEMBALI' and 'KIRIM'. At the bottom of the page is a green footer area with the text 'COPYRIGHT @2023 - EED'.

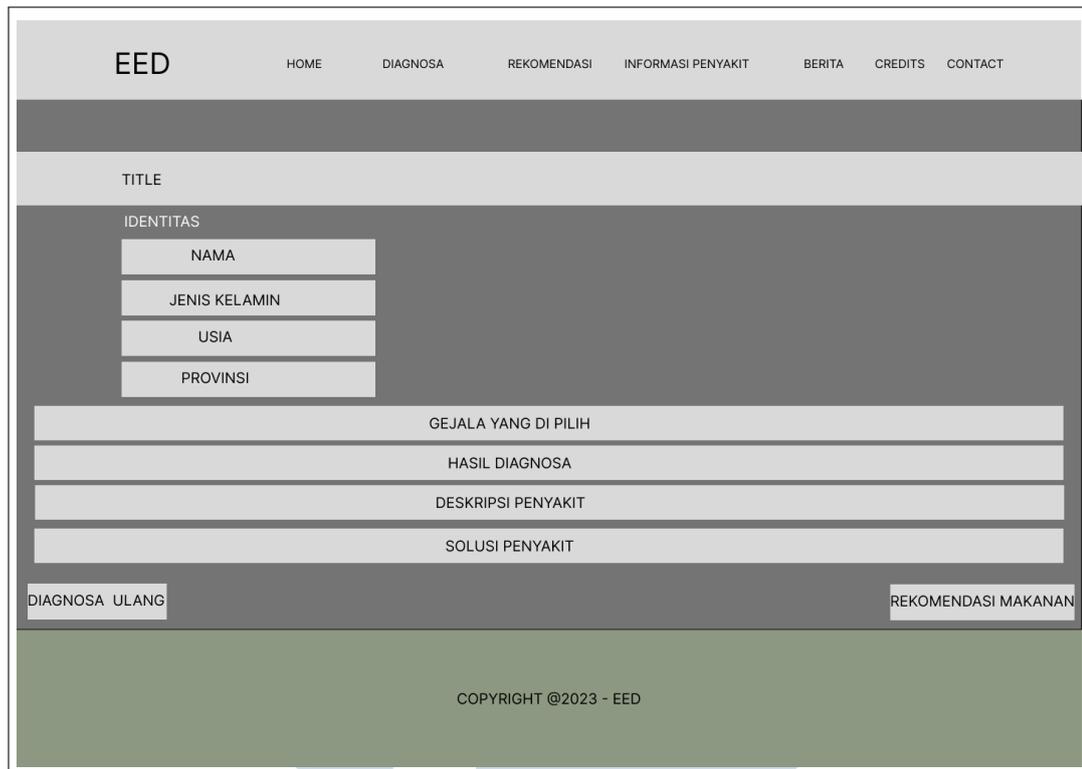
Gambar 3.20. *Low Fidelity Mockup* Halaman Diagnosa

Gambar 3.20 merupakan *Mockup* dari halaman diagnosa. Dalam bagian ini, terdapat kolom biodata yang akan diisi oleh *user* sebelum melakukan diagnosa. Biodata yang akan diisi terdiri dari nama, jenis kelamin, usia, dan provinsi.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.21. *Low Fidelity Mockup* Halaman Diagnosa Lanjutan

Gambar 3.21 merupakan *Mockup* dari halaman diagnosa ketika sudah mengisi biodata pada halaman sebelumnya. Dalam bagian ini, terdapat biodata yang telah diisi sebelumnya serta dengan daftar gejala yang akan dipilih oleh *user* sesuai dengan kondisi yang dirasakan.

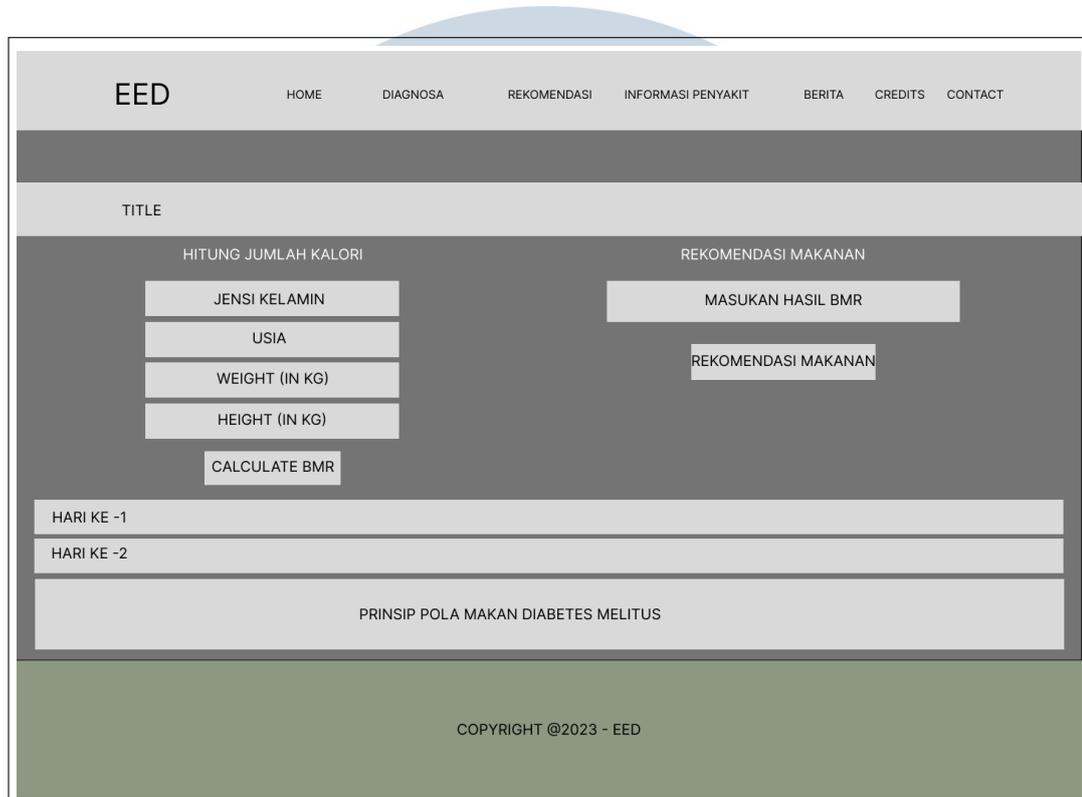


Gambar 3.22. *Low Fidelity Mockup* Halaman Diagnosa Lanjutan

Gambar 3.22 merupakan *Mockup* dari halaman diagnosa ketika sudah selesai mengisi gejala sesuai dengan kondisi *user* pada halaman sebelumnya. Dalam bagian ini, terdapat gejala yang telah dipilih, hasil diagnosa, deskripsi penyakit dan solusi penyakit. Pada bagian bawah halaman terdapat 2 tombol yakni, tombol diagnosa ulang untuk melakukan diagnosa ulang pada halaman pemilihan gejala dan tombol rekomendasi makanan yang berfungsi untuk memindahkan *user* langsung ke halaman rekomendasi makanan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

C Low Fidelity Mockup Halaman Rekomendasi

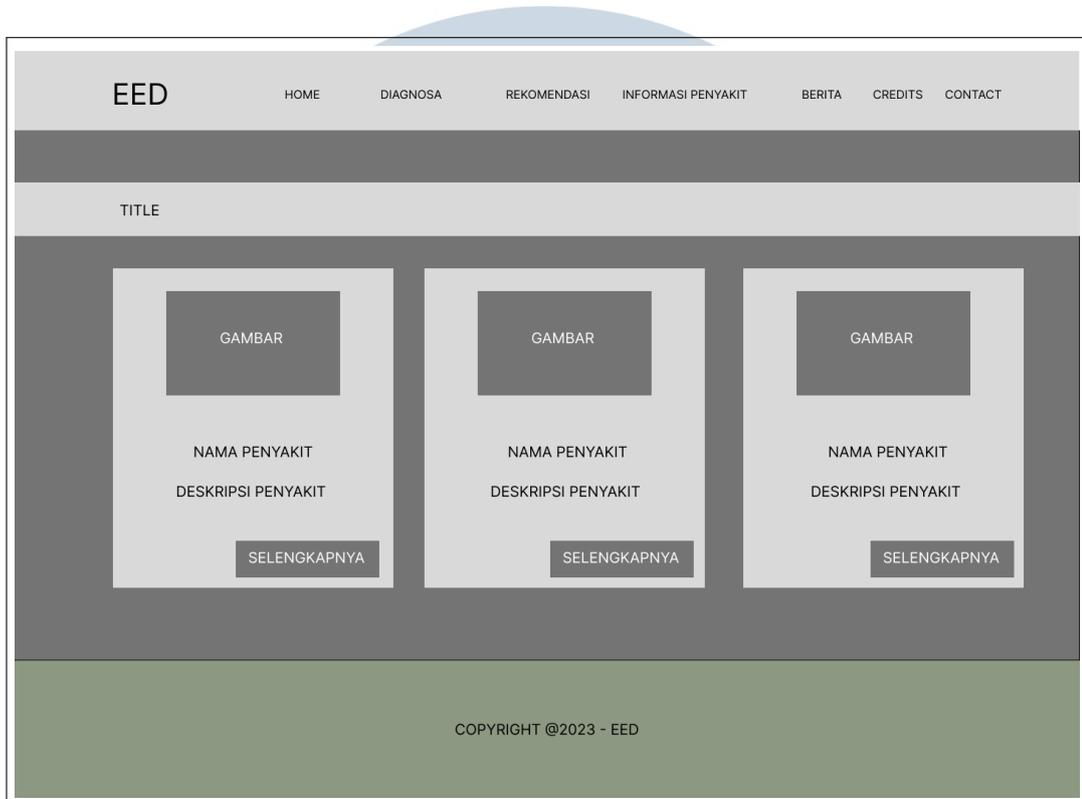


Gambar 3.23. Low Fidelity Mockup Halaman Rekomendasi

Gambar 3.23 merupakan *Mockup* dari halaman rekomendasi. Dalam bagian ini, terdapat kalkulator perhitungan *Basal Metabolic Rate (BMR)* untuk mendapatkan rekomendasi makanan serta terdapat prinsip pola makan Diabetes Melitus pada bagian bawah halaman.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

D *Low Fidelity Mockup* Halaman Informasi Penyakit

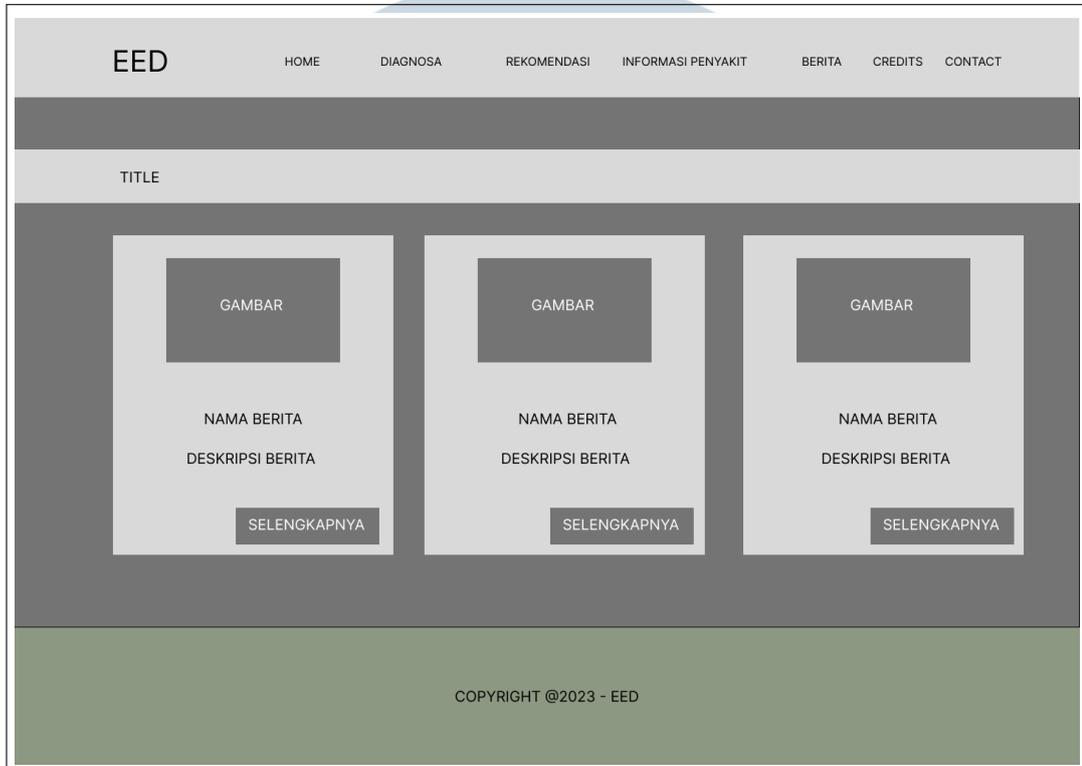


Gambar 3.24. *Low Fidelity Mockup* Halaman Informasi Penyakit

Gambar 3.24 merupakan *Mockup* dari halaman informasi penyakit. Dalam bagian ini, terdapat informasi dari penyakit yang telah diinput oleh admin.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

E Low Fidelity Mockup Halaman Berita

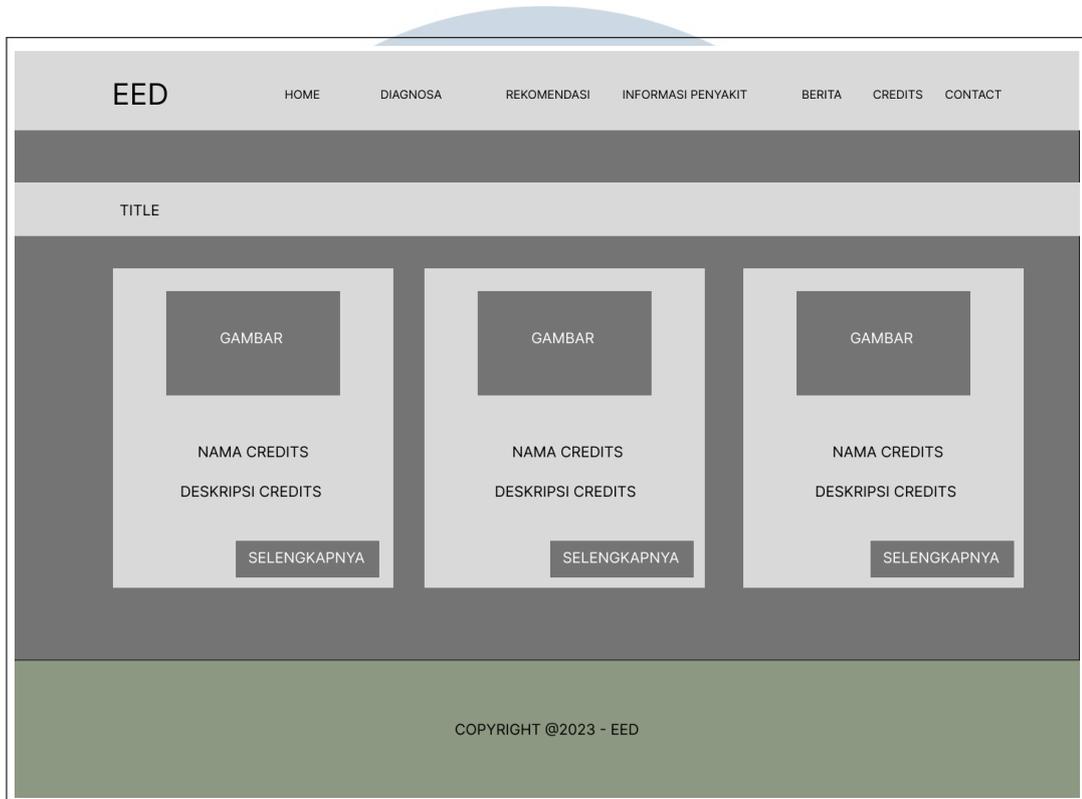


Gambar 3.25. Low Fidelity Mockup Halaman Berita

Gambar 3.25 merupakan *Mockup* dari halaman berita. Dalam bagian ini, terdapat informasi dari berita yang telah diinput oleh admin.

UWMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

F *Low Fidelity Mockup* Halaman Credits

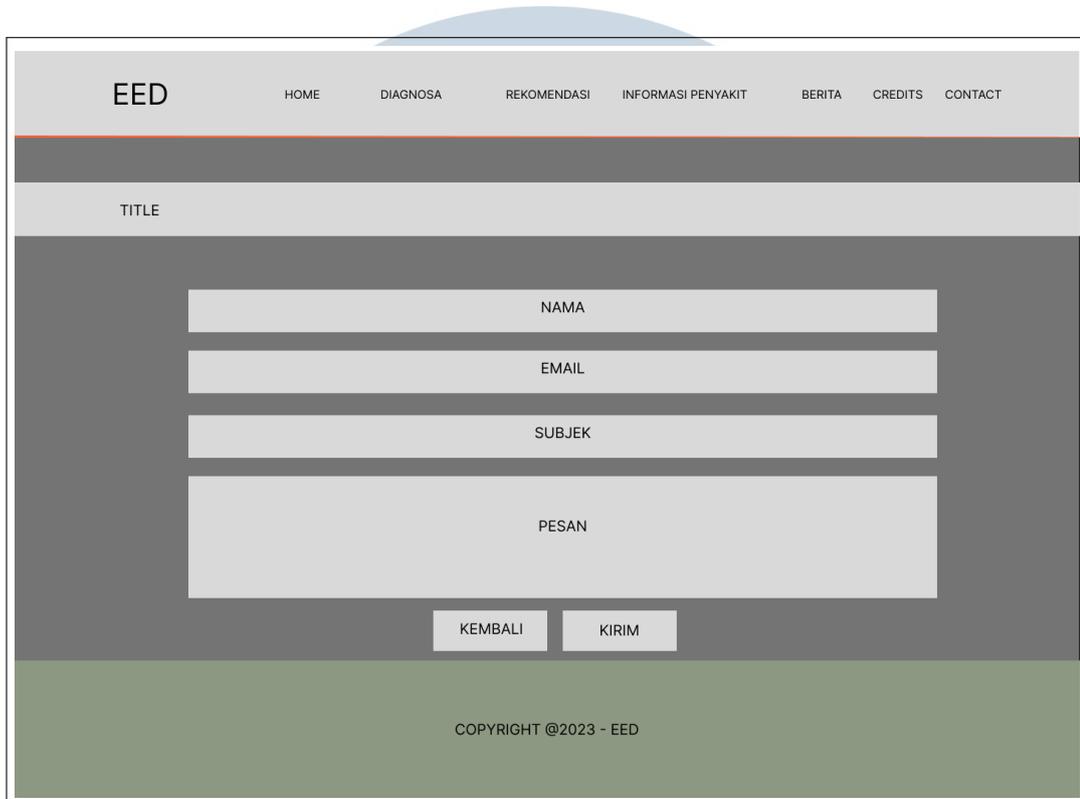


Gambar 3.26. *Low Fidelity Mockup* Halaman Credits

Gambar 3.26 merupakan *Mockup* dari halaman credits. Dalam bagian ini, terdapat informasi dari credits yang telah diinput oleh admin.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

G *Low Fidelity Mockup* Halaman Contact



The image shows a low-fidelity mockup of a contact page. At the top, there is a navigation bar with the logo 'EED' on the left and menu items: HOME, DIAGNOSA, REKOMENDASI, INFORMASI PENYAKIT, BERITA, CREDITS, and CONTACT. Below the navigation bar is a dark grey header area with the word 'TITLE' in white. The main content area is a dark grey rectangle containing a contact form. The form consists of four input fields: 'NAMA', 'EMAIL', 'SUBJEK', and 'PESAN'. Below the 'PESAN' field are two buttons: 'KEMBALI' and 'KIRIM'. At the bottom of the page is a green footer area with the text 'COPYRIGHT @2023 - EED'.

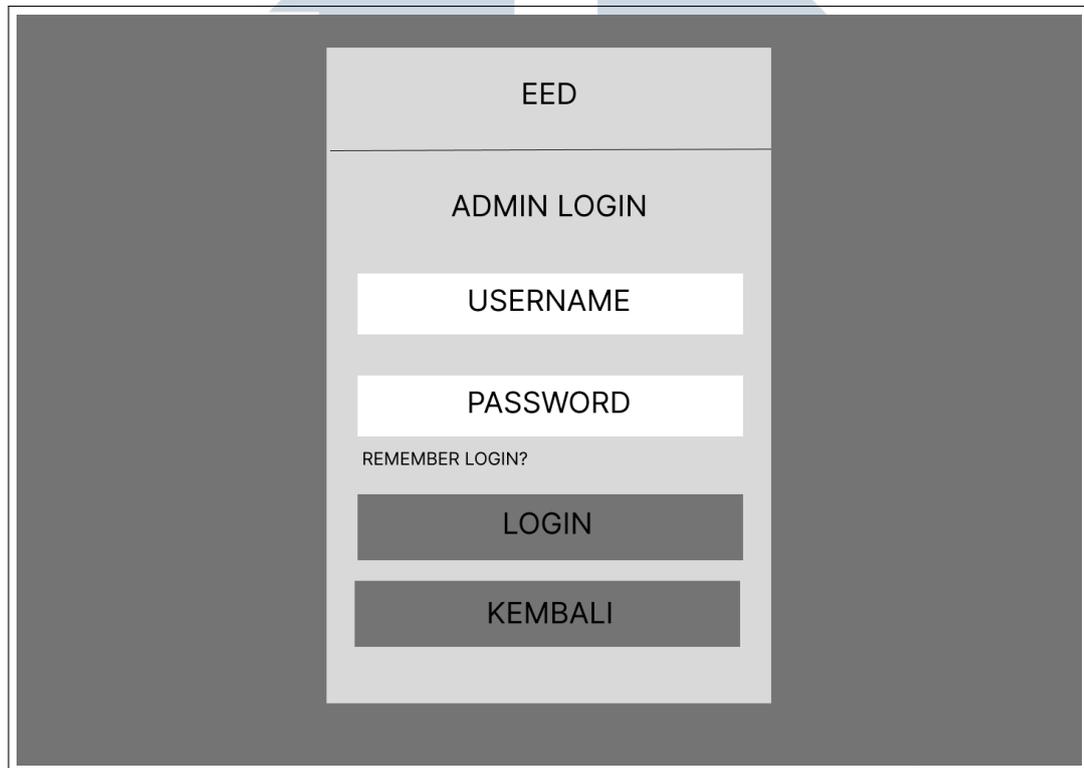
Gambar 3.27. *Low Fidelity Mockup* Halaman Contact

Gambar 3.27 merupakan *Mockup* dari halaman contact. Dalam bagian ini, terdapat form yang dapat diisi oleh *user* jika ingin menghubungi admin.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.9 Low Fidelity Mockup Admin

A Low Fidelity Mockup Halaman Login



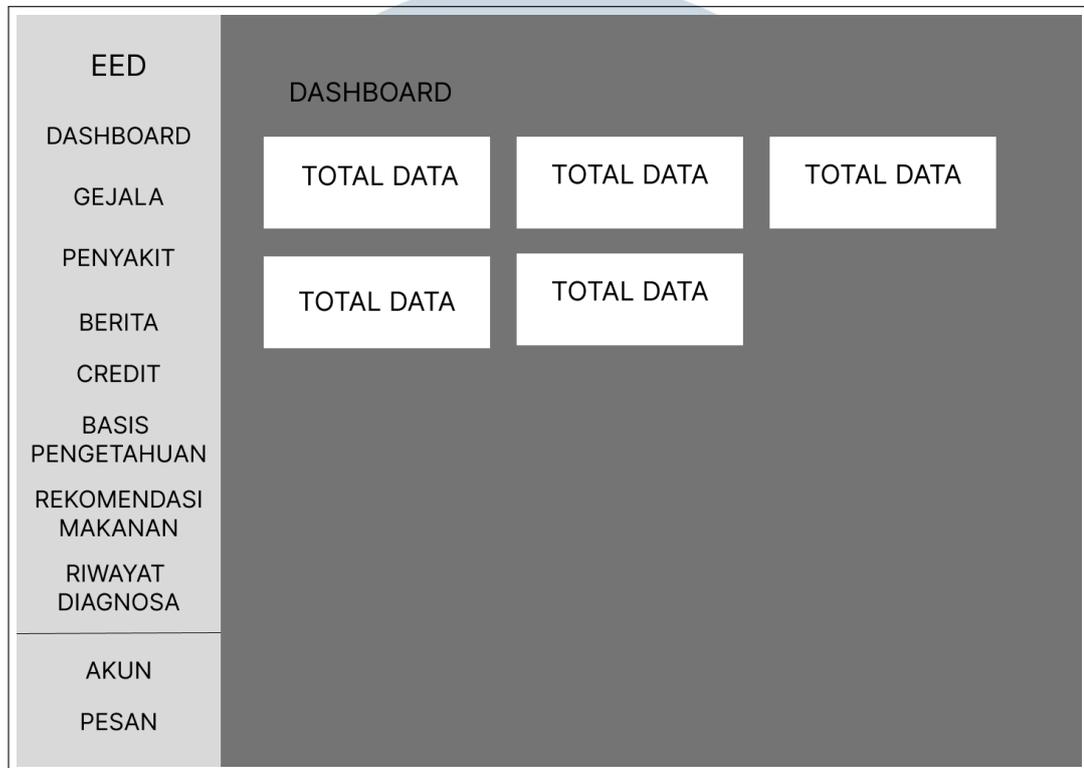
The image shows a low-fidelity mockup of an admin login page. It features a central white form box on a dark gray background. At the top of the form is the text 'EED'. Below that is the title 'ADMIN LOGIN'. The form contains two input fields: 'USERNAME' and 'PASSWORD'. Below these fields is a checkbox labeled 'REMEMBER LOGIN?'. At the bottom of the form are two buttons: 'LOGIN' and 'KEMBALI'.

Gambar 3.28. Low Fidelity Mockup Halaman Login

Gambar 3.28 merupakan *Mockup* dari halaman login admin. Dalam bagian ini, admin dapat menuliskan username serta password yang dimilikinya untuk melanjutkan ke halaman dashboard admin.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

B Low Fidelity Mockup Halaman Dashboard Admin

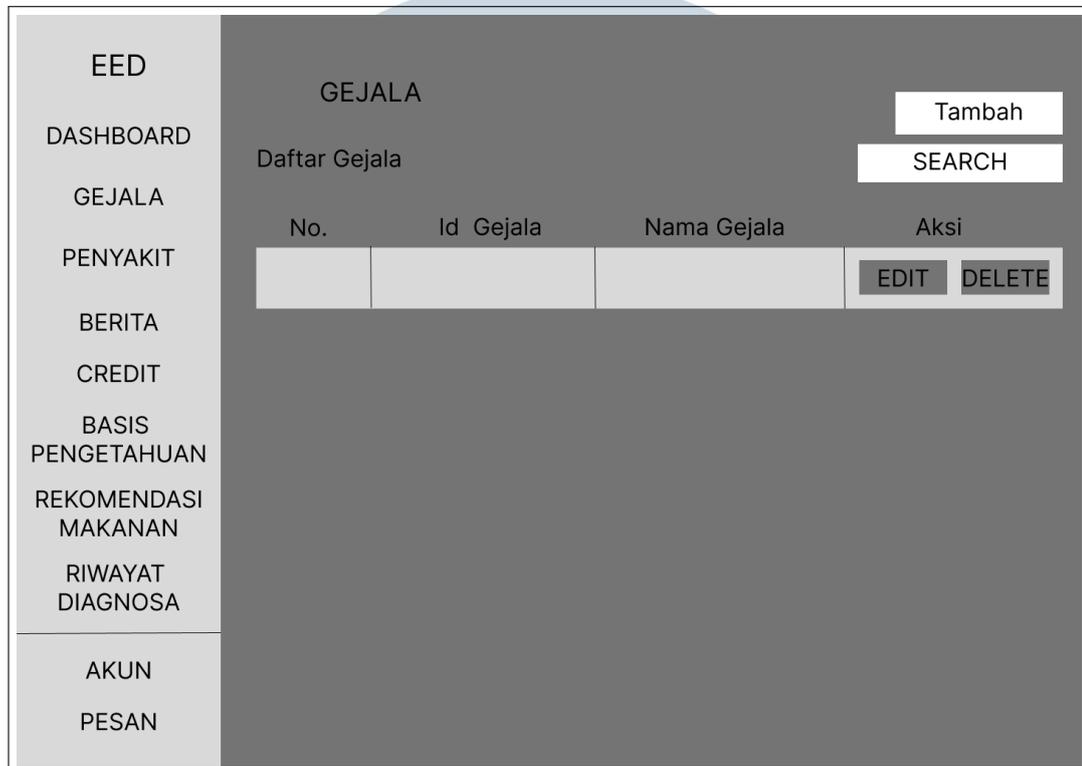


Gambar 3.29. Low Fidelity Mockup Halaman Dashboard Admin

Gambar 3.29 merupakan *Mockup* dari halaman *Dashboard* admin. Dalam bagian ini, terdapat informasi dari total jumlah data dari gejala, penyakit, makanan, diagnosa, dan pesan serta pada bagian kiri terdapat *sidebar* yang dapat menavigasikan admin jika ingin ke menu lainnya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

C Low Fieldity Mockup Halaman Gejala Admin

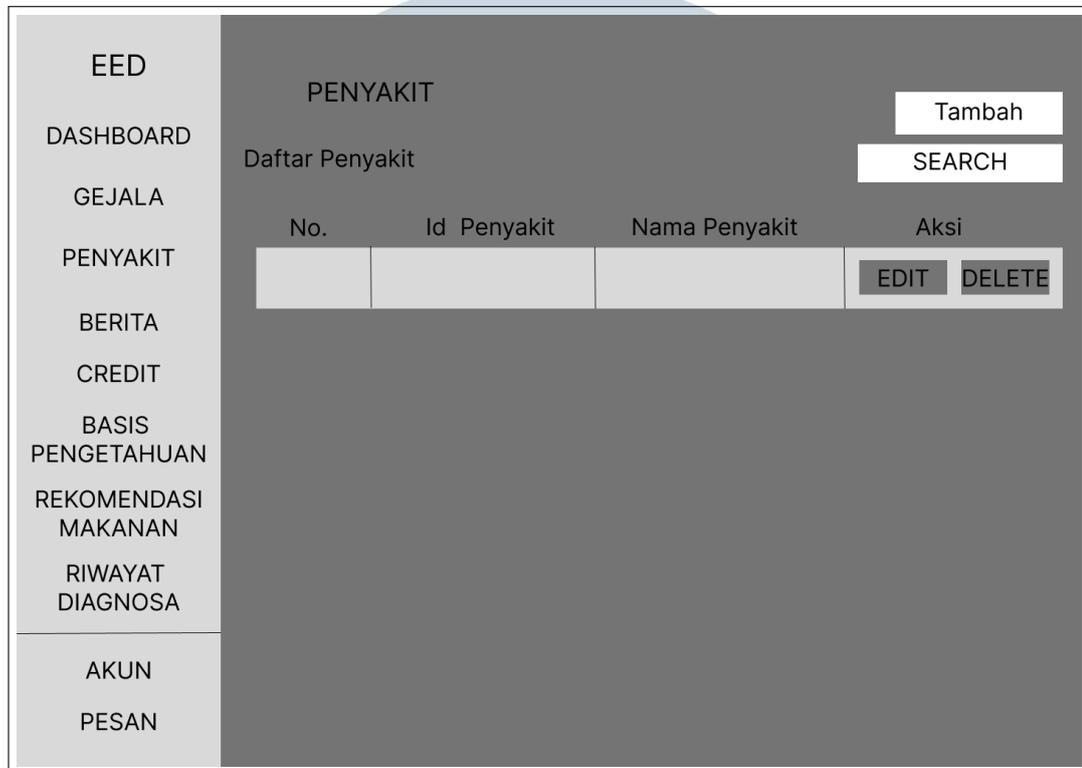


Gambar 3.30. Low Fieldity Mockup Halaman Gejala Admin

Gambar 3.30 merupakan *Mockup* dari halaman gejala admin. Dalam bagian ini, terdapat daftar dari gejala yang sudah diinput oleh admin. Pada bagian atas kanan halaman terdapat tombol *create* yang dapat digunakan untuk menambahkan data gejala serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data gejala yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

D Low Fidelity Mockup Halaman Penyakit Admin

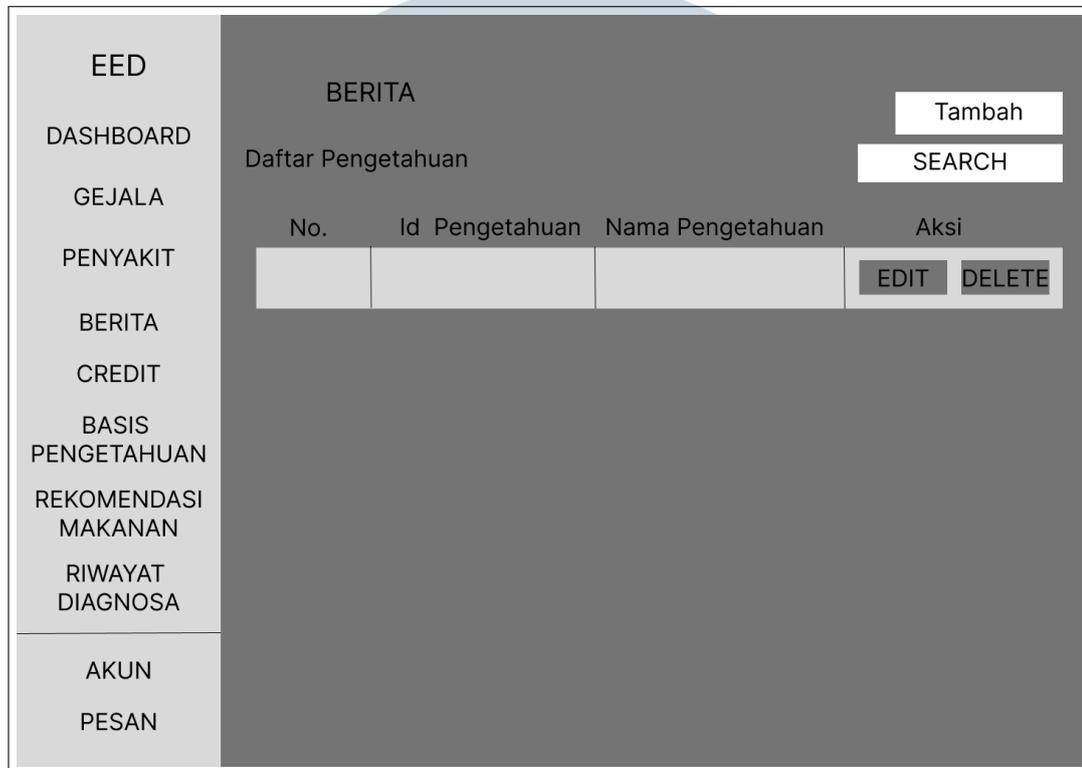


Gambar 3.31. Low Fidelity Mockup Halaman Penyakit Admin

Gambar 3.31 merupakan *Mockup* dari halaman penyakit admin. Dalam bagian ini, terdapat daftar dari penyakit yang sudah diinput oleh admin. Pada bagian atas kanan halaman terdapat tombol *create* yang dapat digunakan untuk menambahkan data penyakit serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data penyakit yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

E Low Fidelity Mockup Halaman Berita Admin

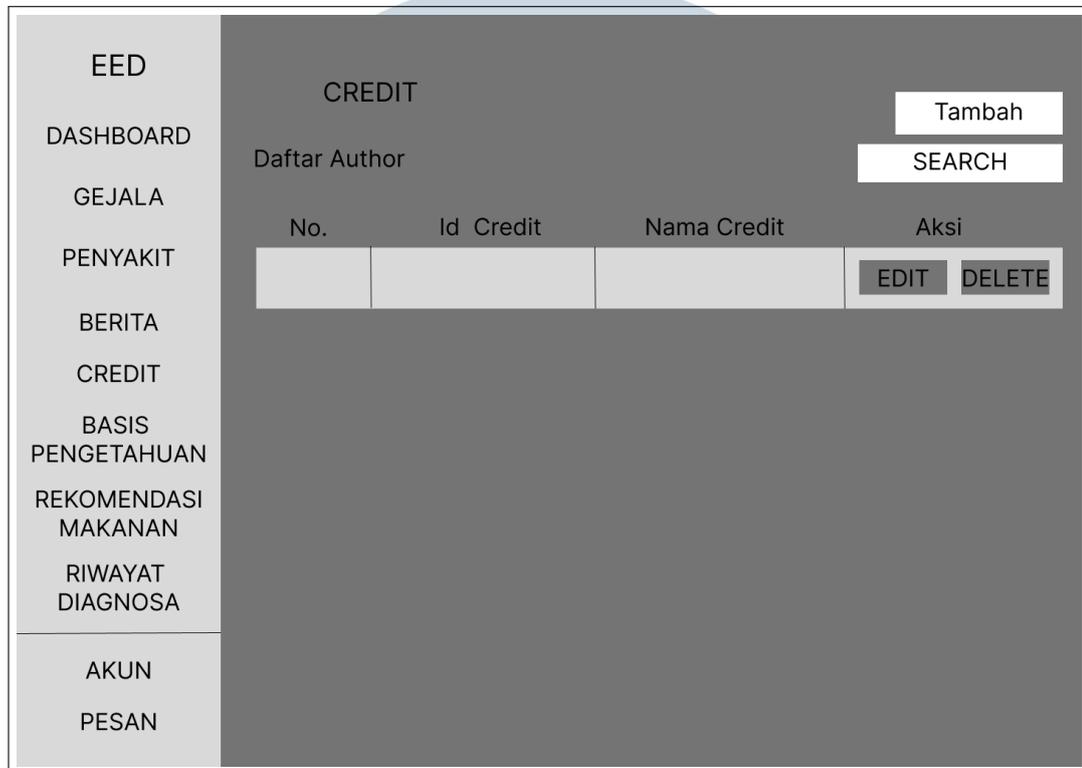


Gambar 3.32. Low Fidelity Mockup Halaman Berita Admin

Gambar 3.32 merupakan *Mockup* dari halaman berita admin. Dalam bagian ini, terdapat daftar dari berita yang sudah diinput oleh admin. Pada bagian atas kanan halaman terdapat tombol *create* yang dapat digunakan untuk menambahkan data berita serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data berita yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

F Low Fidelity Mockup Halaman Credit Admin

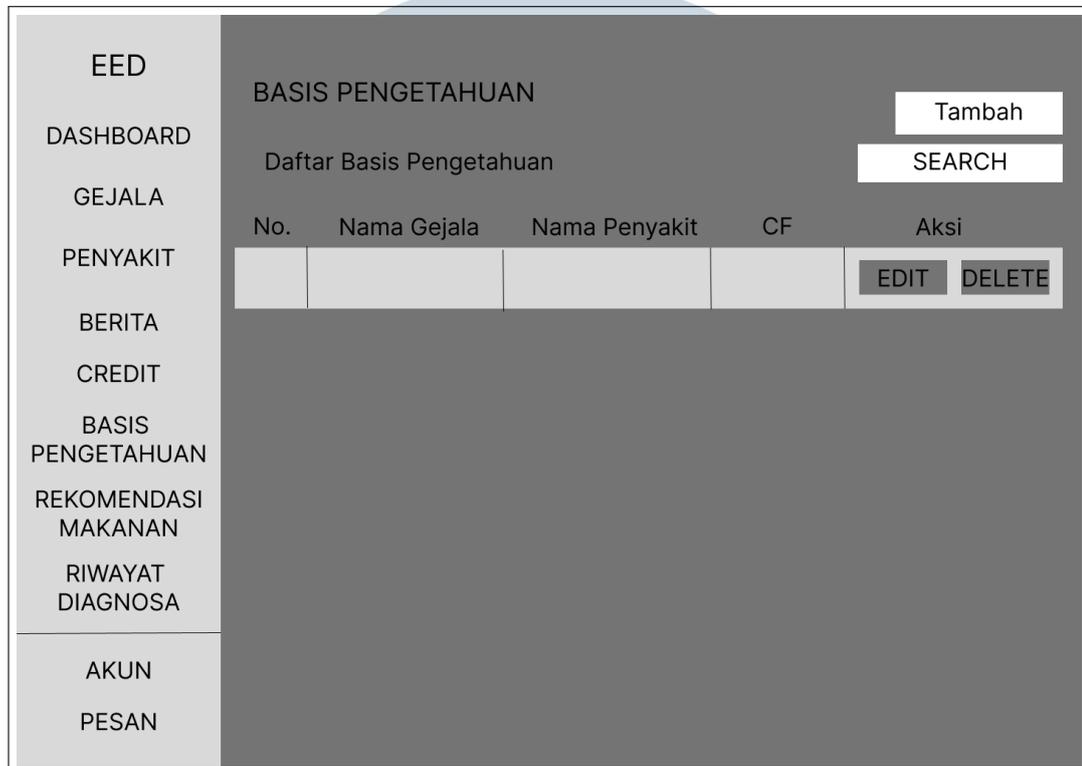


Gambar 3.33. Low Fidelity Mockup Halaman Credit Admin

Gambar 3.33 merupakan *Mockup* dari halaman *credit* admin. Dalam bagian ini, terdapat daftar dari *credit* yang sudah diinput oleh admin. Pada bagian atas kanan halaman terdapat tombol *create* yang dapat digunakan untuk menambahkan data *credit* serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data *credit* yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

G *Low Fidelity Mockup* Halaman Basis Pengetahuan Admin

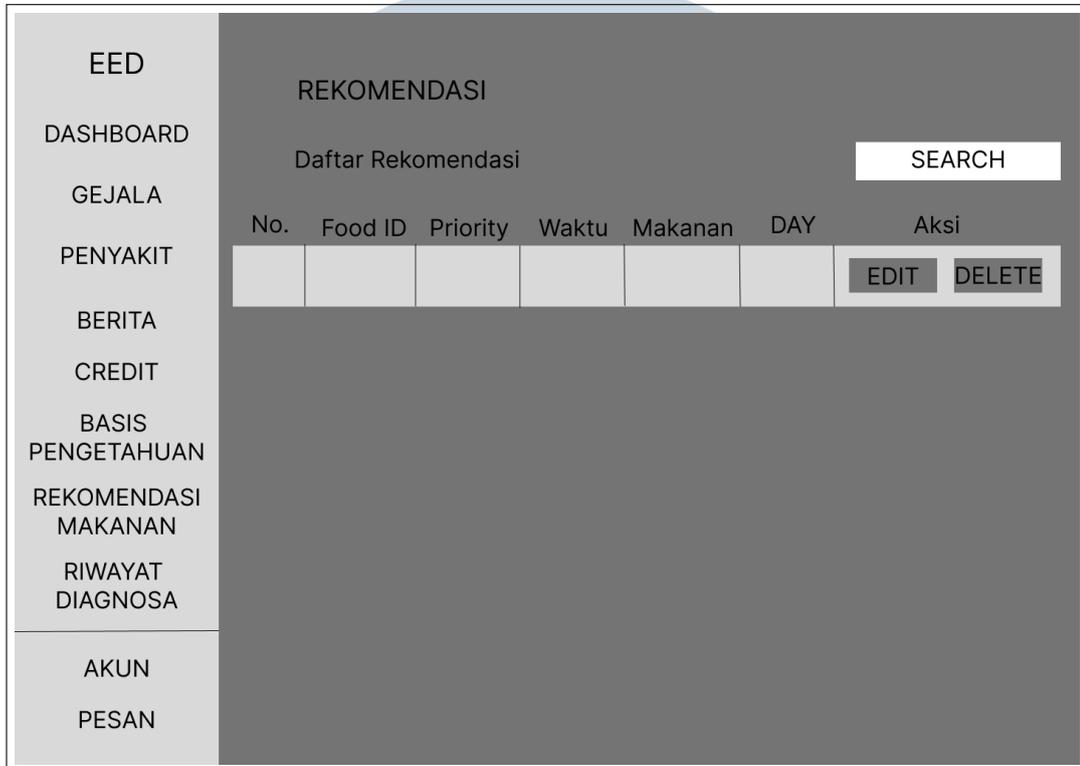


Gambar 3.34. *Low Fidelity Mockup* Halaman Basis Pengetahuan Admin

Gambar 3.34 merupakan *Mockup* dari halaman basis pengetahuan admin. Dalam bagian ini, terdapat daftar dari basis pengetahuan yang sudah diinput oleh admin. Pada bagian atas kanan halaman terdapat tombol *create* yang dapat digunakan untuk menambahkan data basis pengetahuan serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data basis pengetahuan yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

H Low Fidelity Mockup Halaman Rekomendasi Makanan Admin

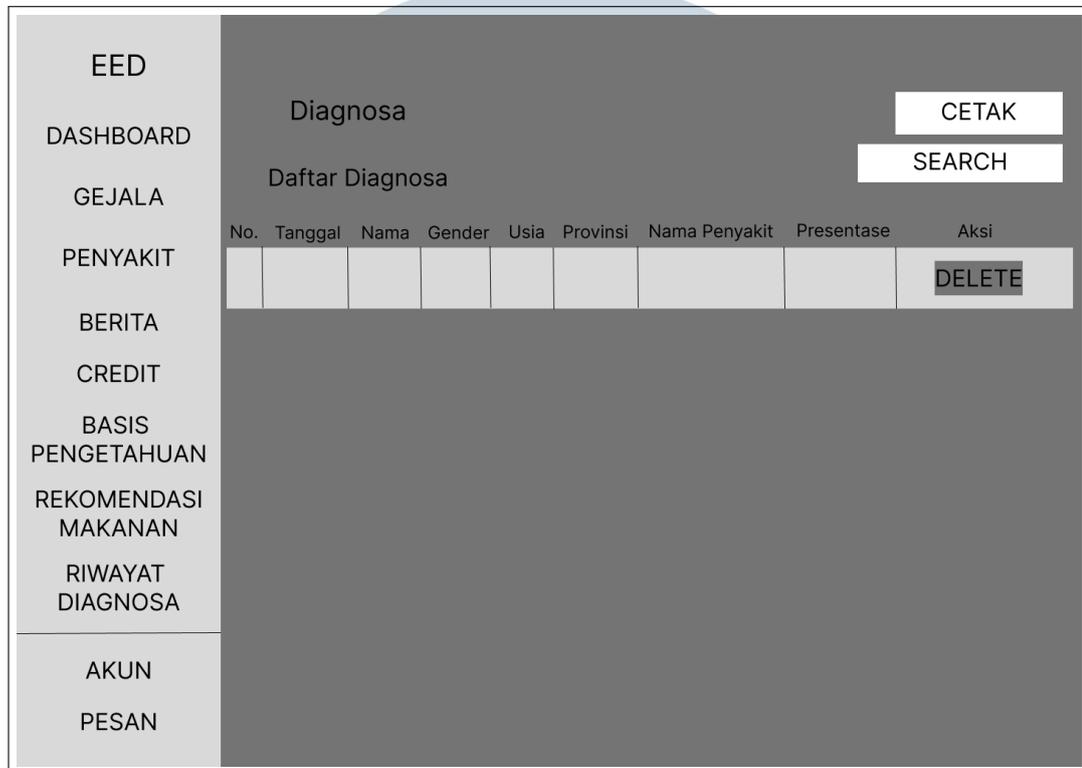


Gambar 3.35. Low Fidelity Mockup Halaman Rekomendasi Makanan Admin

Gambar 3.35 merupakan *Mockup* dari halaman rekomendasi makanan admin. Dalam bagian ini, terdapat daftar dari rekomendasi makanan yang sudah diinput oleh admin. Terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data rekomendasi makanan yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

I Low Fidelity Mockup Halaman Riwayat Diagnosa Admin

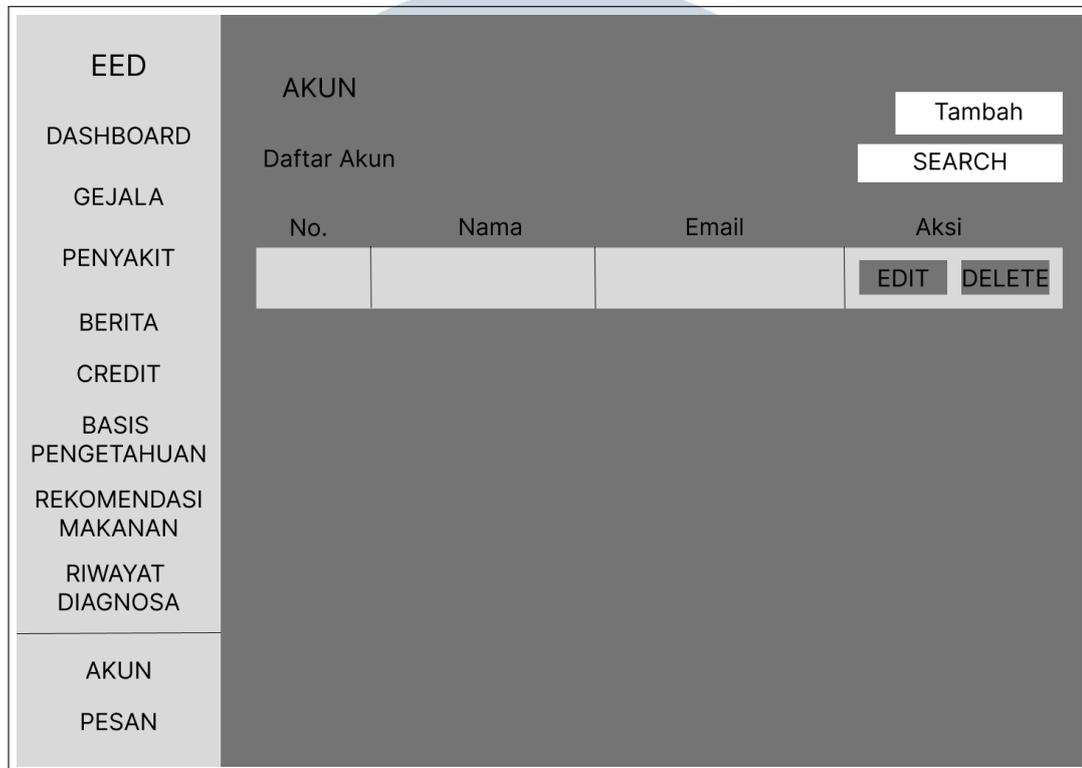


Gambar 3.36. Low Fidelity Mockup Halaman Riwayat Diagnosa Admin

Gambar 3.36 merupakan *Mockup* dari halaman riwayat diagnosa admin. Dalam bagian ini, terdapat daftar dari rekomendasi makanan yang sudah diinput oleh admin. Pada bagian kanan atas terdapat sebuah tombol cetak yang dapat digunakan untuk mencetak riwayat hasil diagnosa serta terdapat hanya satu tombol pada kolom aksi yaitu, *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data riwayat diagnosa yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

J Low Fidelity Mockup Halaman Riwayat Diagnosa Admin

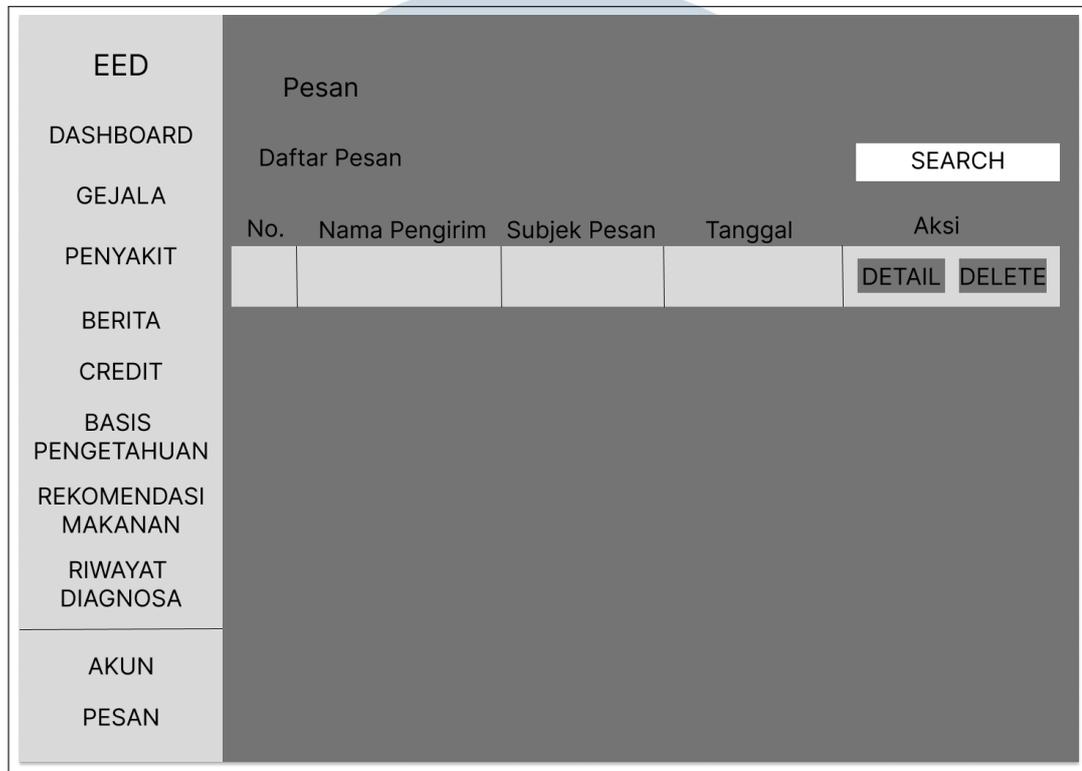


Gambar 3.37. Low Fidelity Mockup Halaman Akun Admin

Gambar 3.37 merupakan *Mockup* dari halaman akun admin. Dalam bagian ini, terdapat daftar dari akun yang sudah dibuat oleh admin. Pada bagian kanan atas terdapat sebuah tombol *create* yang dapat digunakan untuk membuat akun baru serta terdapat 2 tombol pada kolom aksi yaitu, *edit* untuk melakukan perubahan pada data dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data akun yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

K Low Fidelity Mockup Halaman Pesan Admin



Gambar 3.38. Low Fidelity Mockup Halaman Pesan Admin

Gambar 3.38 merupakan *Mockup* dari halaman pesan admin. Dalam bagian ini, terdapat daftar dari pesan yang telah dikirim oleh *user*. Terdapat 2 tombol pada kolom aksi yaitu, *detail* untuk melihat isi detail dari pesan yang telah dikirim dan *delete* untuk menghapus data. Selain itu juga terdapat kolom search untuk mencari data pesan yang ada.

UNIVERSITAS
MULTIMEDIA
NUSANTARA