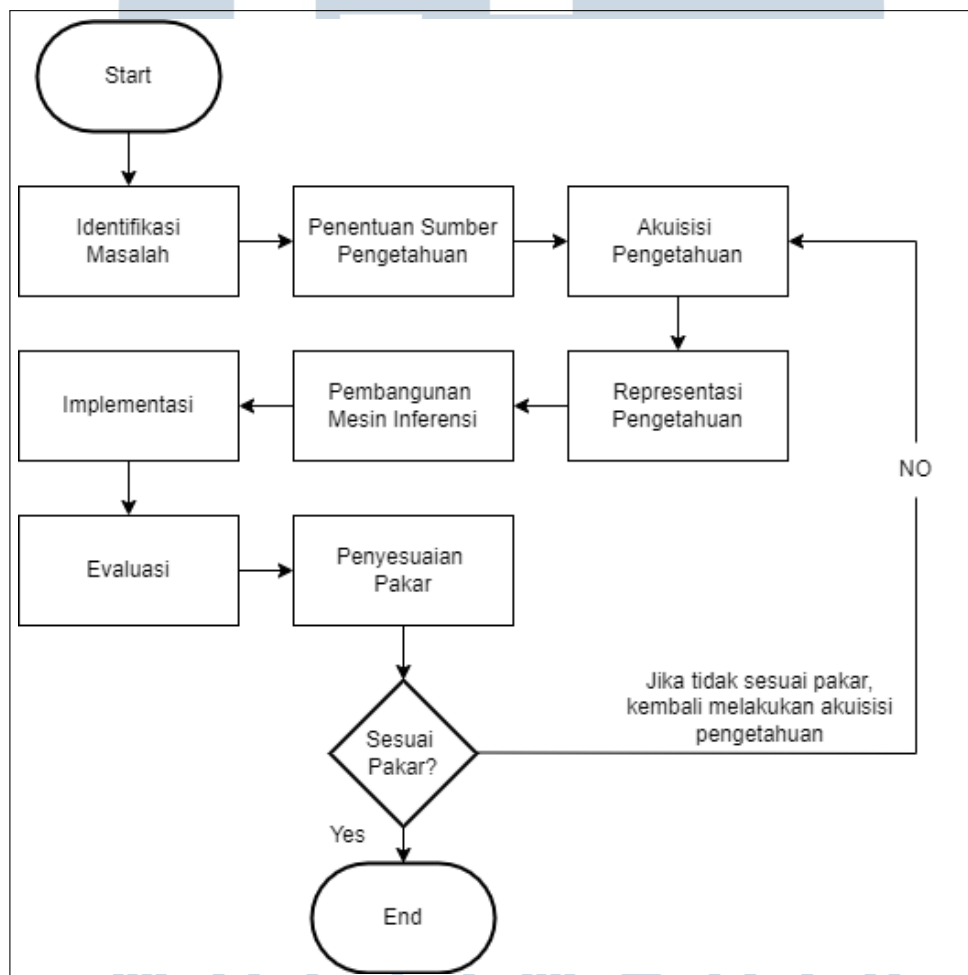


BAB 3 METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Dalam merancang sistem pakar untuk *diet* dan *bulking*, diperlukan beberapa tahap penelitian terlebih dahulu. Tujuannya adalah agar hasil yang diharapkan bisa menjadi solusi untuk masalah yang dihadapi dalam studi ini secara terstruktur, efektif, dan efisien.



Gambar 3.1. Tahapan penelitian

Berikut adalah penjelasan tentang diagram di atas sesuai dengan langkah-langkahnya, yaitu:

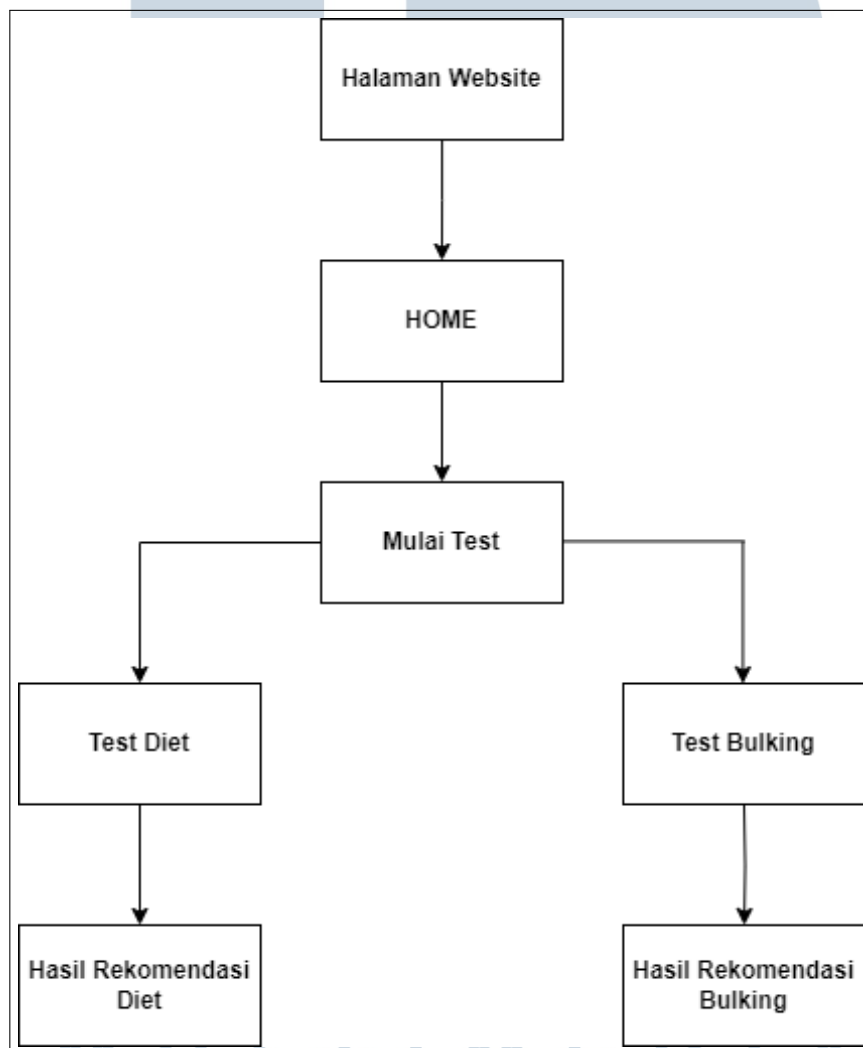
1. Identifikasi masalah
Langkah pertama di mana masalah spesifik yang ingin diselesaikan oleh sistem pakar diidentifikasi.
2. Penentuan sumber pengetahuan
Mengidentifikasi sumber pengetahuan yang relevan untuk masalah yang dihadapi, seperti ahli di bidang terkait atau literatur.
3. Akuisisi pengetahuan
Mengumpulkan pengetahuan dari sumber yang telah ditentukan untuk membangun basis pengetahuan sistem.
4. Representasi pengetahuan
Menyusun pengetahuan yang telah dikumpulkan dalam format yang dapat diproses oleh sistem, seperti aturan atau basis data.
5. Pembangunan mesin inferensi
Mengembangkan mesin inferensi yang akan memproses pengetahuan dan memberikan solusi berdasarkan *input* yang diterima.
6. Implementasi
Menerapkan sistem pakar yang telah dikembangkan ke bentuk nyata untuk digunakan oleh *user*.
7. Evaluasi
Menilai kinerja sistem pakar dan efektivitasnya dalam menyelesaikan masalah yang diidentifikasi sebelumnya.
8. Proses penyesuaian pakar
Proses evaluasi untuk menentukan apakah hasil yang diberikan oleh sistem pakar sesuai dengan pengetahuan dan pengalaman pakar manusia, jika 'ya', sistem pakar telah selesai dikembangkan, dan jika 'tidak', melakukan akuisisi pengetahuan kembali.

Diagram ini memberikan gambaran tentang langkah-langkah yang terlibat dalam pengembangan sistem pakar, serta alur logis dari satu langkah ke langkah lainnya.

3.2 Perancangan Sistem

3.2.1 Sitemap

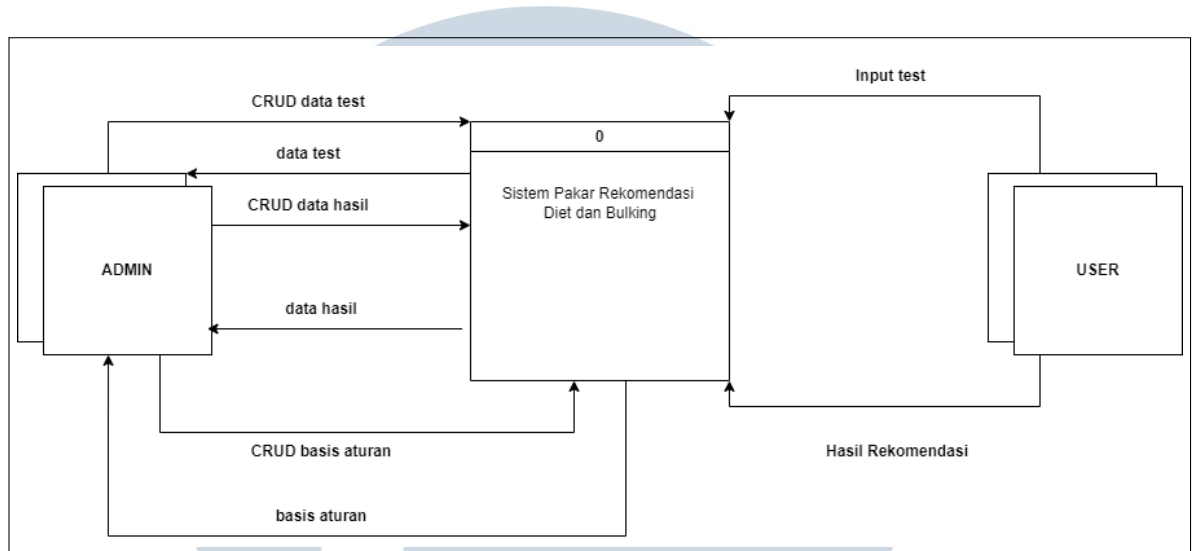
Sitemap berfungsi untuk menguraikan struktur halaman atau navigasi dalam aplikasi *web* untuk rekomendasi *diet* dan *bulking*. *Sitemap* akan memberi visualisasi hubungan antara halaman utama, submenu, dan fitur-fitur penting dalam aplikasi, sehingga memudahkan *user* dalam menggunakan aplikasi.



Gambar 3.2. Sitemap Sistem

User dapat melakukan *test* untuk kebutuhan tubuh mereka dengan cara *checklist* terhadap kebutuhan yang dibutuhkan. Hal ini menunjukkan bahwa *user* dapat mengakses menu "Mulai Test!". Setelah *user* mengisi semua *test*, hasil rekomendasi akan diberikan sesuai dengan hasil *test user*

3.2.2 Data Flow Diagram



Gambar 3.3. DFD Level 0

DFD *level 0* adalah representasi sederhana dari sistem pakar diagnosa rekomendasi *diet* dan *bulking*. Berikut akan dijelaskan mengenai hal yang terdapat pada DFD *level 0*:

1. Admin

- Admin dapat memantau dan mengelola aplikasi.
- Hanya admin yang dapat melakukan operasi CRUD.
- CRUD: Admin dapat melakukan operasi CRUD untuk data *diet*, *bulking*, dan basis aturan (*rule*) dalam sistem diagnosa.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

2. Sistem Pakar Rekomendasi *diet* dan *bulking*

- Sistem pakar rekomendasi *diet* dan *bulking* menerima input *user*.
- Sistem menerima input *test* yang dilakukan *user* pada saat melakukan *test* untuk menghasilkan rekomendasi *diet* atau *bulking*
- Sistem melakukan proses data yang telah diinput oleh *user* untuk menampilkan hasil rekomendasi.

3. *User*

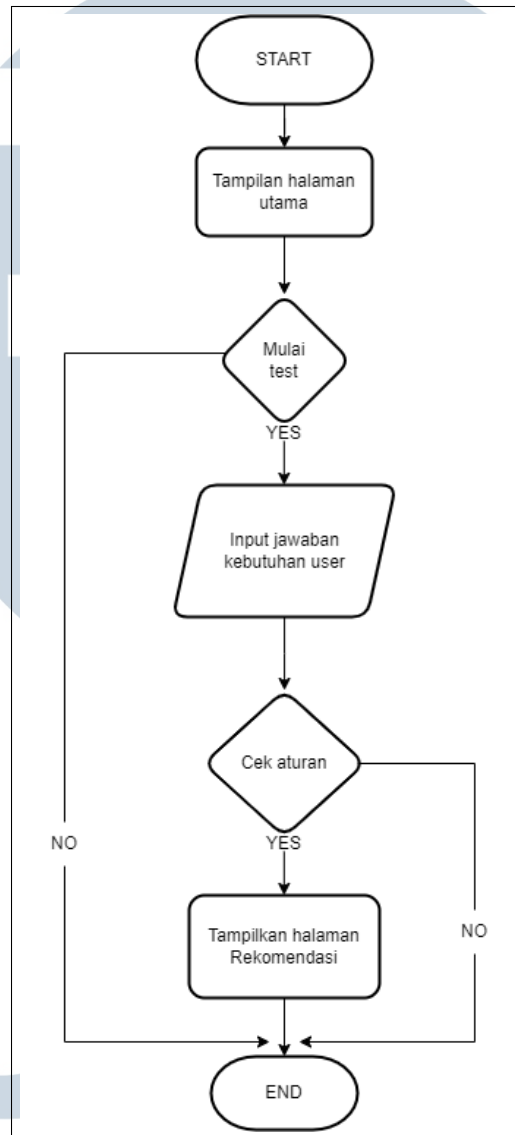
- *User* dapat melakukan *test* dan mengisi kebutuhan yang dibutuhkan oleh tubuh *user*.
- *User* menerima hasil dari *test* yang dilakukan sesuai jawaban *user* saat melakukan *test*.

Diagram DFD *level 0* ini menunjukkan interaksi antara sistem pakar rekomendasi, *user*, dan admin. Tugas admin adalah mengelola data dan aturan (*rule*) pada sistem pakar rekomendasi, untuk tugas *user* adalah menggunakan aplikasi sistem pakar rekomendasi untuk melakukan *test* yang nanti akan diberikan hasil rekomendasi.



3.2.3 Flowchart

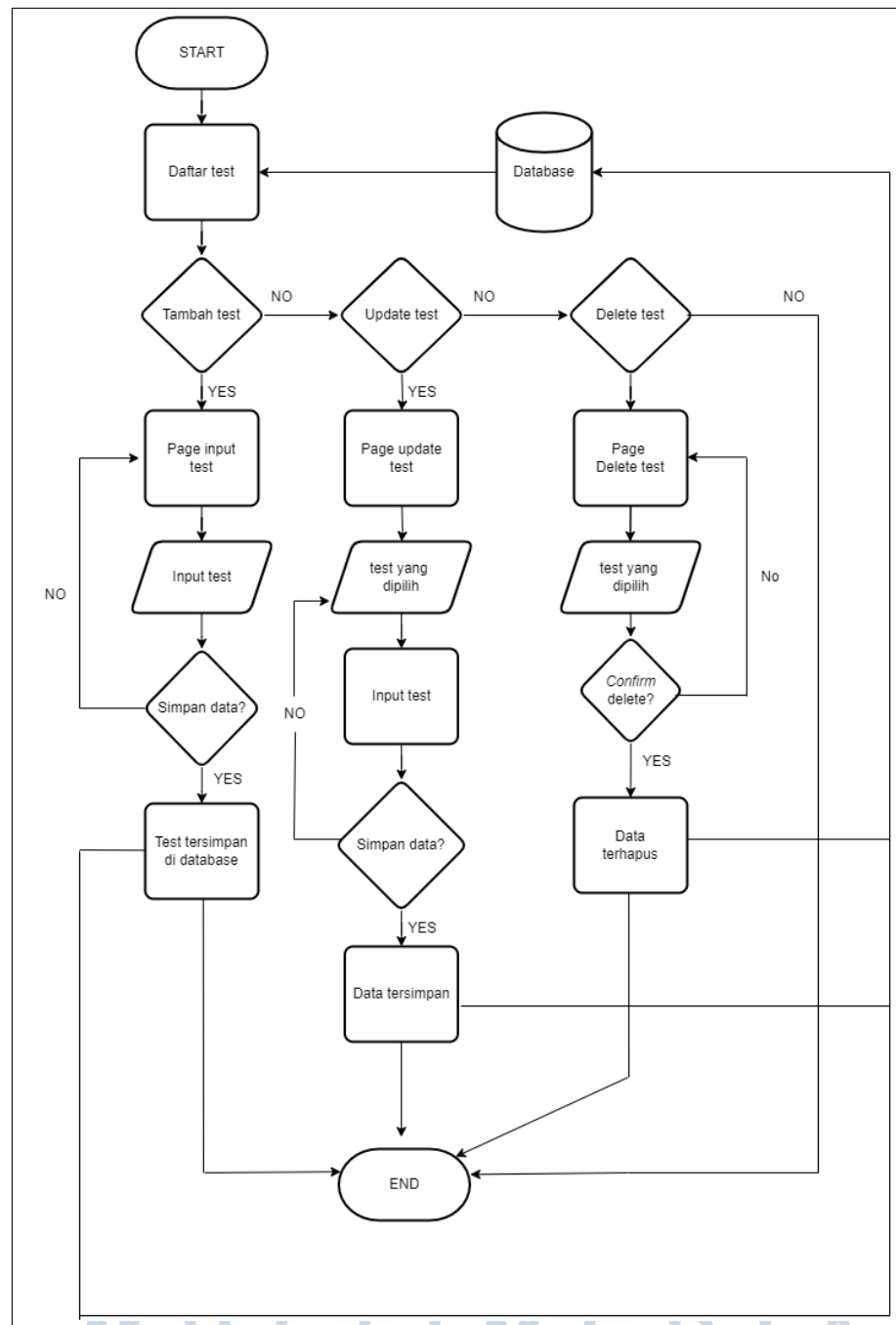
1. Flowchart Alur Test



Gambar 3.4. Flowchart Alur Test

Flowchart alur *test* menggambarkan alur penggunaan *website* sistem pakar rekomendasi *diet* dan *bulking*. Pertama, dimulai dengan menampilkan halaman utama pada *website*. Kedua, *user* memulai *test* dengan menekan tombol "Mulai Test". Ketiga, *user* menginput jawaban sesuai kebutuhan *user*, apakah *user* ingin *diet* atau ingin melakukan *bulking*. Terakhir, setelah *user* selesai melakukan *test*, *website* akan menampilkan halaman rekomendasi sesuai dengan hasil input jawaban dari *user*.

2. Flowchart test



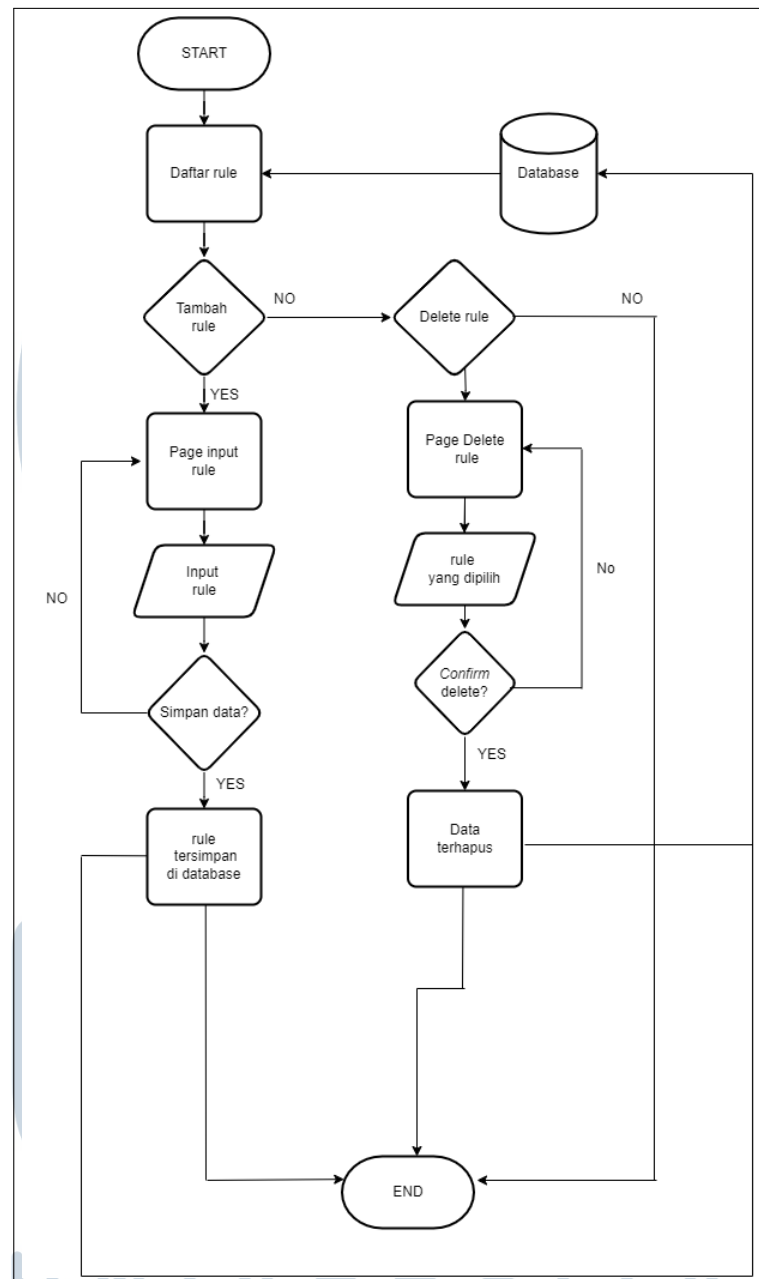
Gambar 3.5. Flowchart test

Flowchart test menjelaskan proses pengelolaan data *test*. Pada bagian proses data *test*, terdapat tiga pilihan, yaitu menambah *test*, menghapus *test*, dan memperbarui *test*.

- Jika menambah *test*, harus mengisi data *test* yang ingin ditambah, dan setelah selesai menginput data *test*, data *test* akan tersimpan di *database* yang nanti akan muncul pada saat *user* melakukan *test*.
- Jika menghapus *test*, *test* akan dihapus dari database dan *test* tersebut sudah tidak ada saat *user* melakukan *test*.
- Jika memperbarui *test*, harus memilih *test* yang ingin diubah. Setelah selesai mengedit data *test*, *save*, dan *test* yang diubah akan tersimpan di *database*.



3. Flowchart Rule



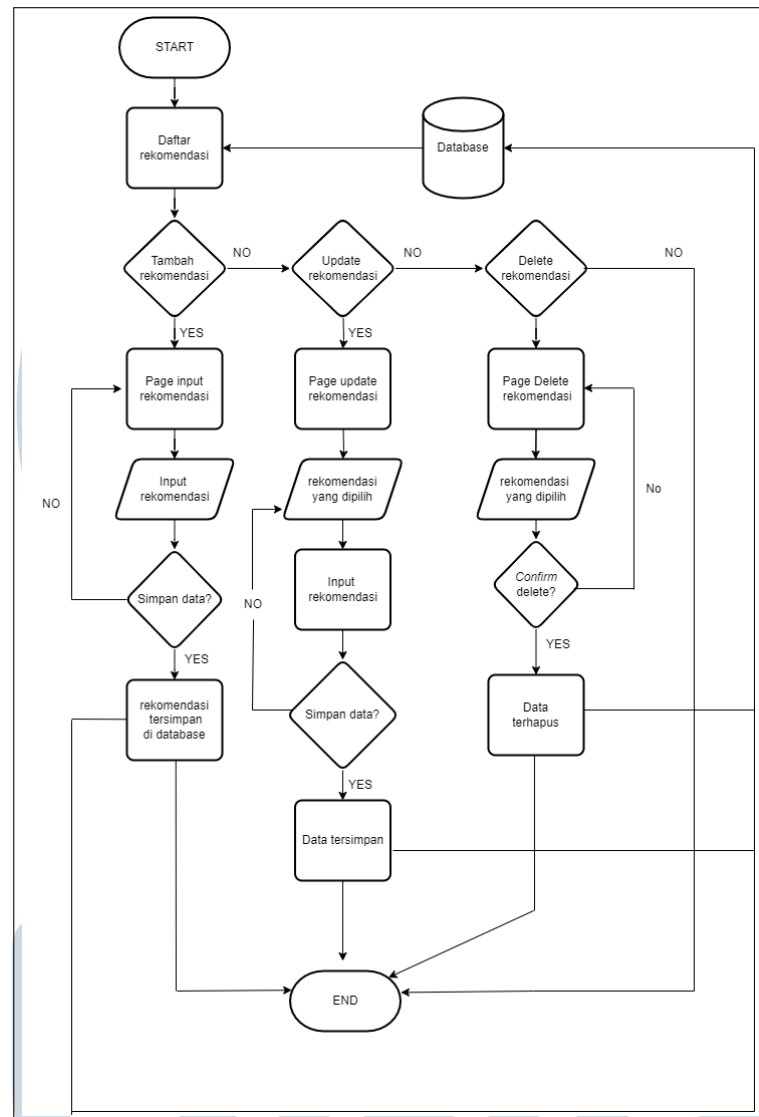
Gambar 3.6. Flowchart Rule

Flowchart rule atau *flowchart* aturan menjelaskan proses pengelolaan data *rule*. *Rule* atau aturan merupakan instruksi logis untuk mengolah informasi yang ada dan menuju pada suatu kesimpulan atau rekomendasi. Pada bagian proses data *rule*, terdapat dua pilihan, yaitu menambah *rule* dan menghapus *rule*.

- Jika menambah *rule*, harus mengisi data, dan setelah selesai menginput data *rule*, data *rule* akan tersimpan di *database*.
- Jika menghapus *rule*, harus memilih *rule* yang ingin dihapus dari *database*. Setelah selesai menghapus *rule*, instruksi pada *rule* yang dihapus tidak akan bisa digunakan untuk memberikan rekomendasi.



4. Flowchart Rekomendasi

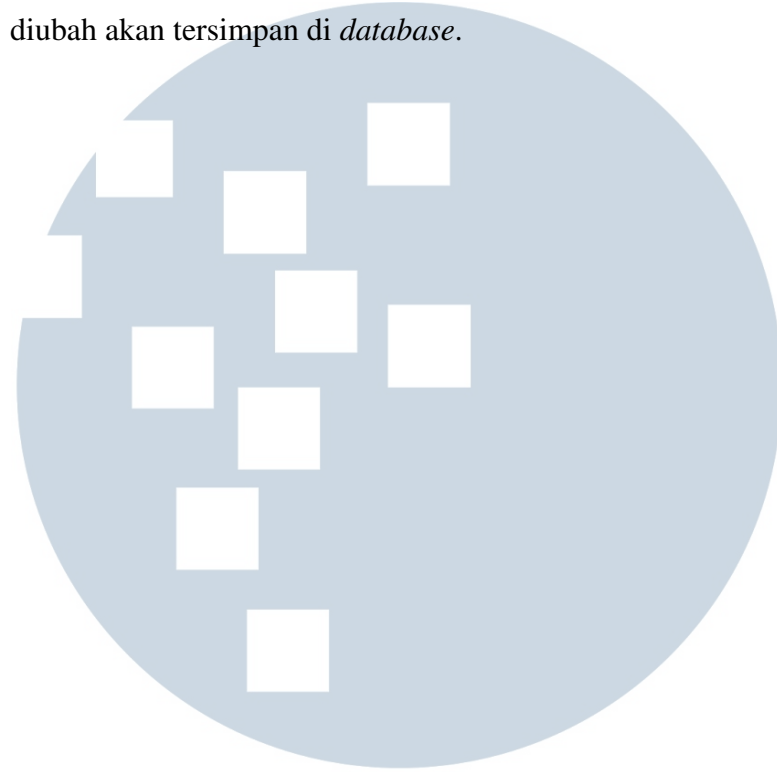


Gambar 3.7. Flowchart rekomendasi

Flowchart rekomendasi menjelaskan proses pengelolaan data rekomendasi. Pada bagian proses data rekomendasi, terdapat tiga pilihan, yaitu menambah rekomendasi, menghapus rekomendasi, dan memperbarui rekomendasi.

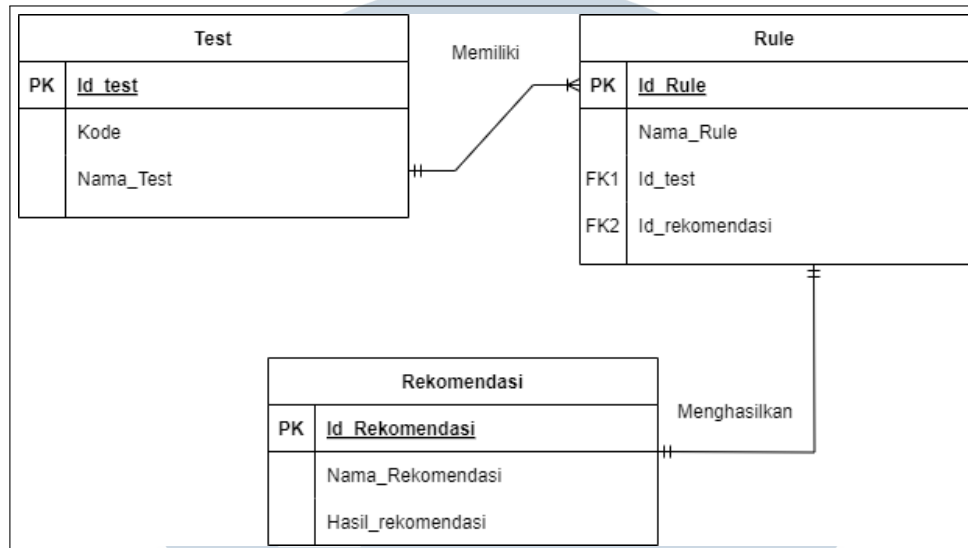
- Jika ingin menambah rekomendasi, harus mengisi data rekomendasi, dan setelah selesai menginput data rekomendasi, data rekomendasi baru akan tersimpan di *database*.
- Jika menghapus rekomendasi, rekomendasi akan dihapus dari database dan tidak akan bisa ditampilkan pada halaman rekomendasi.

- Jika memperbarui rekomendasi, rekomendasi yang ingin diubah. Setelah selesai mengedit data rekomendasi, *save*, dan rekomendasi yang diubah akan tersimpan di *database*.



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.4 Entity Relationship Diagram



Gambar 3.8. ERD

Entity Relationship Diagram (ERD) menggambarkan hubungan antara tiga entitas, *Test*, *Rule*, dan *Rekomendasi*. Entitas *test* memiliki atribut "Id_test" sebagai *primary key* yang mengidentifikasi setiap *test* secara unik, serta atribut tambahan seperti *Kode* dan *Nama_Test*. Entitas *rule* juga memiliki *primary key*, "Id_Rule", yang diikuti oleh atribut "Nama_Rule", "Id_test" sebagai *foreign key* yang merujuk ke entitas *test*, dan "Id_rekomendasi" sebagai *foreign key* yang merujuk ke entitas *rekomendasi*. Hubungan antara *test* dan *rule* adalah *One-to-Many* (1:N), yang berarti satu *test* dapat memiliki banyak aturan yang diterapkan padanya. Entitas *rekomendasi* memiliki atribut "Id_rekomendasi" sebagai *primary key*, serta "Nama_Rekomendasi" dan "Hasil_rekomendasi". Hubungan antara *rule* dan *rekomendasi* adalah *One-to-One*, jadi setiap aturan akan menghasilkan satu rekomendasi tertentu, dan setiap rekomendasi hanya terkait dengan satu aturan.

Diagram ERD ini menggambarkan bagaimana data mengenai *test* dan rekomendasi saling berhubungan melalui aturan untuk membentuk sistem rekomendasi *diet* dan *bulking*. Setiap *test* diberi *id* dan nama, dan dihubungkan dengan rekomendasi melalui aturan yang mengatur hasil rekomendasi sesuai dengan jawaban dari *user* saat melakukan *test*. *Entity Relationship Diagram* memberikan penjelasan singkat tentang bagaimana *test* dan rekomendasi dihubungkan melalui aturan atau *rule* dan dapat membentuk sistem pakar rekomendasi *diet* dan *bulking*.

3.2.5 Skema Database

Skema *database* menjelaskan tentang desain skema basis data yang digunakan dalam sistem rekomendasi yang berisi tabel yang diperlukan dan hubungan antar tabel.

1. Tabel Aturan

- (a) Tabel aturan berisi tentang basis pengetahuan untuk menghubungkan jawaban dari *test* yang di isi oleh *user* untuk menghasilkan rekomendasi.
- (b) Dua hal penting yang terdapat dalam tabel aturan, yaitu:

Tabel 3.1. Tabel Struktur Database dietbulking_aturan

No.	Nama Kolom	Tipe Data	Keterangan
1	id_test	bigint (20)	Primary Key dari tabel <i>test</i>
2	id_rekomendasi	bigint (20)	Primary Key dari tabel rekomendasi

- **id_test** merupakan daftar *test* yang berhubungan dengan hasil rekomendasi.
- **id_rekomendasi** merupakan nama rekomendasi yang terdapat dalam aturan.

2. Tabel Test

Tabel 3.2. Struktur tabel dietbulking_test

Kolom	Tipe Data	Keterangan
id	bigint(20)	Primary Key
nama	varchar(255)	Nama test

- (a) Tabel ini digunakan untuk menyimpan seluruh data tentang *test* dalam sistem rekomendasi.
- (b) 'nama' dalam kolom tabel adalah nama *test* yang digunakan dalam pertanyaan untuk mengetahui kebutuhan *user*.

3. Tabel Rekomendasi

- (a) Tabel rekomendasi menyimpan tentang data rekomendasi hasil dari *test* yang telah dilakukan oleh *user*.

Tabel 3.3. Struktur tabel dietbulking_rekomendasi

Kolom	Tipe Data	Keterangan
id	bigint(20)	Primary Key
nama	varchar(255)	Nama rekomendasi
hasil_rekomendasi	text	Hasil rekomendasi berdasarkan hasil test

(b) ada tiga kolom pada tabel rekomendasi, yaitu:

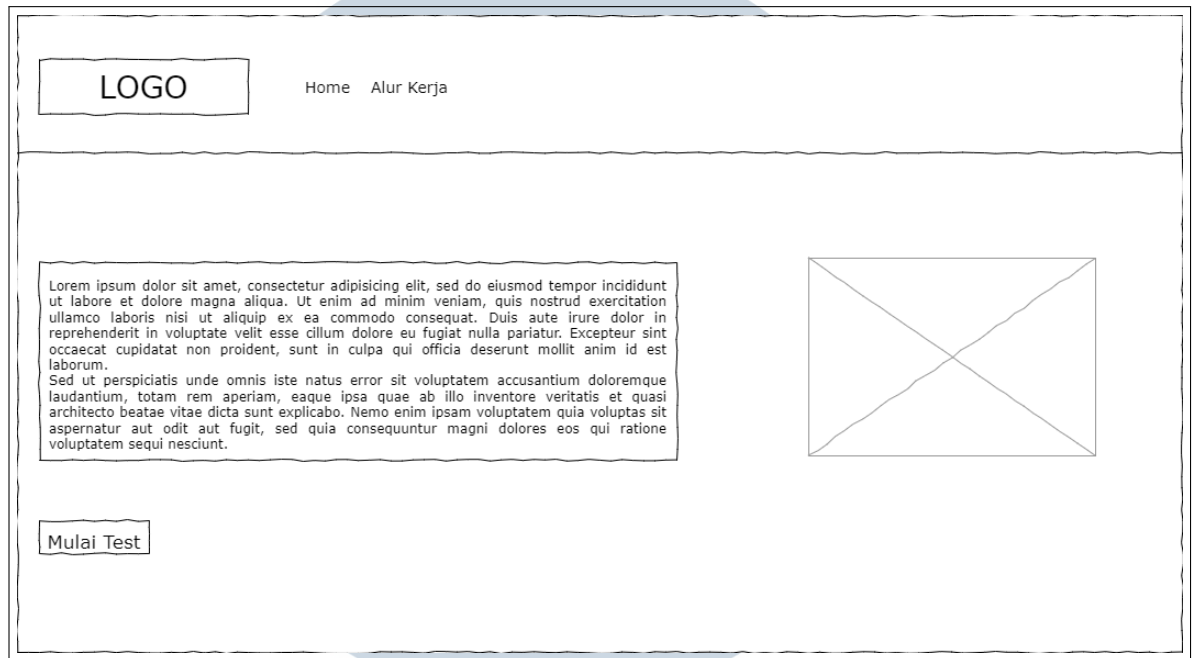
- 'id': berisi data *primary key*
- 'nama': merupakan nama rekomendasi
- "hasil_rekomendasi": Menyimpan hasil rekomendasi yang sesuai dengan jawaban *test user*

Skema *database* dibuat untuk mendukung sistem pakar rekomendasi dengan menyimpan informasi tentang *test*, hasil rekomendasi, dan aturan atau *rule* yang menghubungkan antara hasil *test* dan rekomendasi.



3.3 Wireframe

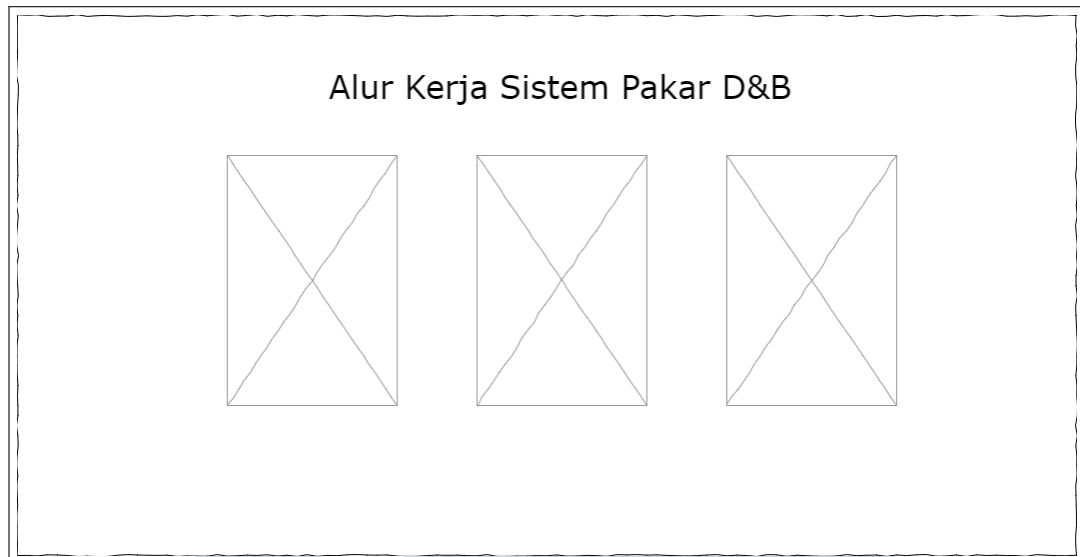
1. Halaman Utama



Gambar 3.9. Halaman Utama

Desain sederhana *wireframe* ini menampilkan halaman utama *website* yang dirancang untuk membantu pengguna memenuhi kebutuhan *diet* atau *bulking* yang ingin dijalankan. Terdapat logo di pojok kiri atas, yang berfungsi sebagai identitas *website*. Di sebelah kanan logo, terdapat menu navigasi yang mencakup tautan ke halaman utama atau *home* dan navigasi untuk mengarahkan ke bagian alur kerja. Di tengah halaman terdapat deskripsi singkat terkait *website diet* dan *bulking*. Di sebelah kanan deskripsi, terdapat gambar yang bermakna "Gizi Seimbang". Terakhir, di bawah deskripsi ada tombol navigasi untuk mengarahkan *user* untuk melakukan *test* [14].

2. Halaman Alur Kerja



Gambar 3.10. Halaman Alur Kerja

Pada *wireframe* halaman alur kerja ini, menampilkan gambar yang di bawahnya terdapat deskripsi bagaimana alur kerja dari *website* sistem rekomendasi *diet* dan *bulking* ini.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

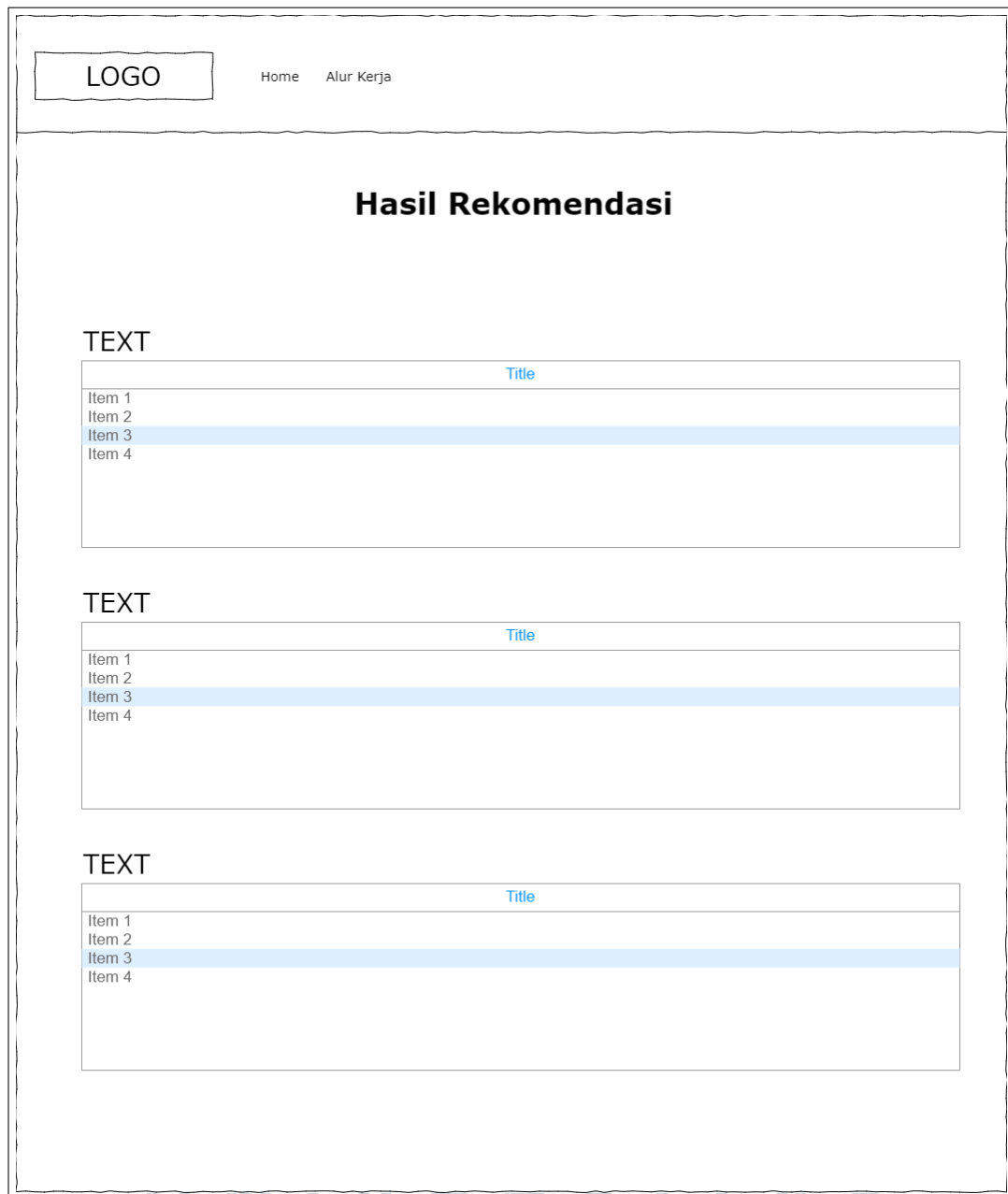
3. Halaman Test

The wireframe shows a rectangular page layout. At the top left is a box labeled 'LOGO'. To its right are two links: 'Home' and 'Alur Kerja'. Below this is a large section titled 'Q&A'. This section contains seven identical blocks, each labeled 'Section 1' through 'Section 7'. Each block has three radio buttons labeled 'O Y O N'. At the bottom center of the page is a button labeled 'Continue'.

Gambar 3.11. Halaman Test

Pada *wireframe* halaman *test*, dikumpulkan beberapa informasi yang disajikan menjadi beberapa pertanyaan untuk *user* jawab sesuai dengan kebutuhan *user*. Di bagian kiri atas, terdapat logo yang menjadi identitas *website*. Terdapat tombol navigasi 'Home' dan tombol navigasi "Alur Kerja" untuk kembali ke halaman utama di samping logo. Pertanyaan yang ada pada halaman *test* harus diisi semua untuk membantu *website* dalam memberikan rekomendasi yang sesuai dengan kebutuhan *user*.

4. Halaman Hasil Rekomendasi



Gambar 3.12. Halaman Hasil Rekomendasi

Pada halaman hasil rekomendasi pada *website* ini dirancang untuk menampilkan hasil rekomendasi dari jawaban *user* saat melakukan *test*. Di bagian kiri atas, terdapat logo yang menjadi identitas *website*. Terdapat tombol navigasi 'Home' dan tombol navigasi "Alur Kerja" untuk kembali ke halaman utama di samping logo. Di tengah halaman, terdapat *table* yang

menampilkan hasil rekomendasi dari jawaban yang *user input*. Desain pada halaman hasil rekomendasi ini memastikan *user* mendapatkan semua jawaban yang diperlukan dengan jelas setelah melakukan *test* selesai. Halaman hasil rekomendasi ini dibuat dengan fokus pada kejelasan dan kemudahan akses, sehingga *user* dapat mengetahui kebutuhan yang mereka harus lakukan dan *user* mendapat informasi apa saja yang harus mereka lakukan selama *diet* atau *bulking*.

