

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Pada penelitian ini memiliki tahapan perancangan *website* dengan menggunakan enam tahapan utama, yaitu tahapan yang dimulai dari studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian serta perbaikan sistem, dan diakhiri dengan evaluasi sistem. Uraian dari tiap tahapan akan dijelaskan sebagai berikut.

#### 1. Studi Literatur

Studi literatur merupakan tahap awal yang dilakukan pada penelitian ini. Pada tahapan ini dilakukan pencarian hasil penelitian terdahulu seperti jurnal, untuk mendukung teori-teori pada penelitian. Sedangkan sumber lain seperti buku elektronik maupun artikel, digunakan untuk menambah informasi serta memahami lebih lanjut mengenai topik penelitian yang dibahas. Teori terkait Sistem Pendukung Keputusan dan metode *Simple Additive Weighting* (SAW) didapatkan pada tahapan ini.

#### 2. Pengumpulan Data

Setelah mendapatkan teori dasar, maka dibutuhkan pengumpulan data. Pada tahapan ini dilakukan wawancara dengan narasumber, yaitu karyawan pada beberapa toko yang menjual motor listrik maupun sepeda listrik. Wawancara tersebut bertujuan untuk menentukan kriteria yang digunakan pada penelitian. Kriteria tersebut merupakan pertimbangan atau preferensi para konsumen ketika membeli suatu motor atau sepeda listrik. Selain melakukan wawancara, dilakukan juga pengumpulan data terkait produk motor listrik dan sepeda listrik dari brosur yang terdapat pada toko serta sumber lainnya seperti situs web resmi dari tiap merek. Data yang diambil meliputi harga produk, *power* dinamo, kecepatan maksimum, jarak tempuh, kapasitas baterai, daya angkut, sistem rem, warna yang tersedia, serta gambar produk.

#### 3. Perancangan Sistem

Pada tahapan ini dilakukan perancangan *website* yang dibangun. Tahap ini meliputi pembuatan *wireframe*, *flowchart*, serta struktur *database*.

#### 4. Implementasi Sistem

Pada tahapan ini dilakukan implementasi terhadap sistem pendukung keputusan yang telah dirancang sebelumnya. Pengembangan sistem ini juga menerapkan metode *Simple Additive Weighting* berdasarkan pada kriteria yang telah ditentukan yang sesuai dengan topik penelitian.

#### 5. Pengujian & Perbaikan Sistem

Setelah sistem berhasil diimplementasi dan dikembangkan, diperlukan pengujian seperti pengecekan alur sistem guna menemukan *bug* atau *error*. Kemudian jika ditemukan kesalahan atau kekurangan, maka dapat segera diperbaiki dan diuji kembali. Sedangkan jika sistem telah berjalan dengan baik dan lancar, maka pada tahapan ini juga dapat dilakukan pengujian seperti membandingkan hasil perhitungan sistem dengan perhitungan manual.

#### 6. Evaluasi

Pada tahapan ini dilakukan penyebaran kuesioner kepada masyarakat umum, khususnya yang tertarik dengan motor atau sepeda listrik. Kuesioner yang menggunakan metode *End User Computing Satisfaction* (EUCS) bertujuan untuk mengetahui tingkat kepuasan pengguna setelah menggunakan *website*. Kemudian data tersebut diolah serta ditarik kesimpulannya untuk menunjukkan apakah sistem bernilai baik atau tidak.

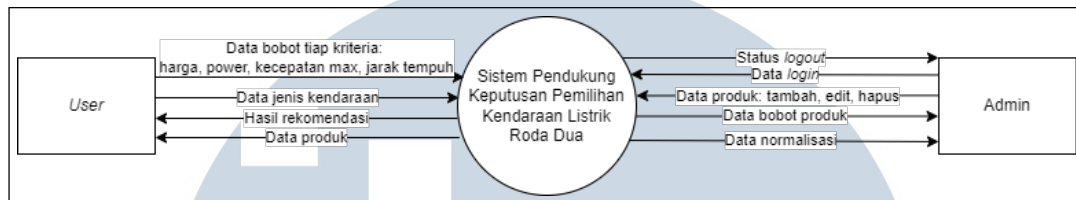
### 3.2 Perancangan Sistem

Pada bagian ini, perancangan sistem pendukung keputusan pemilihan kendaraan listrik roda dua dapat dijelaskan mulai dari diagram alir aplikasi, diagram relasi antar tabel, struktur basis data, diagram alir data, hingga perancangan desain antarmuka aplikasi.

#### 3.2.1 Data Flow Diagram

*Data-Flow Diagram (DFD)* adalah representasi grafis dari aliran data melalui sebuah sistem. DFD menunjukkan jenis-jenis data yang dimasukkan ke sistem dan yang dihasilkan dari suatu sistem. Diagram alir data pada sistem ini terdiri atas DFD level 0 atau yang sering disebut dengan *context diagram* serta DFD level 1.

Berikut ini adalah *context diagram* dari situs Torselis.

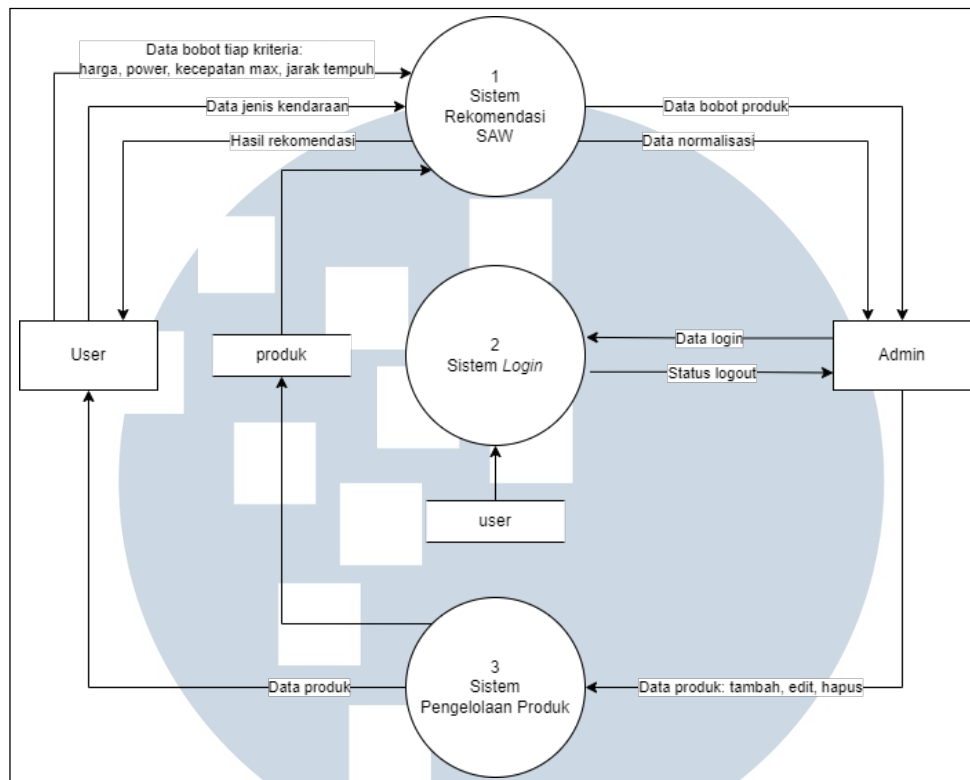


Gambar 3.1. *Context Diagram* situs Torselis

*Data flow diagram* level nol pada sistem ini terdapat pada Gambar 3.1. Diagram tersebut menjelaskan tentang dua tipe entitas yang terdapat pada sistem, yaitu *user* dan *admin*. Entitas *user* memiliki peran sebagai pengguna sistem pendukung keputusan pemilihan kendaraan listrik roda dua ini. *User* bisa mendapatkan hasil rekomendasi kendaraan dengan memberikan *input* berupa jenis kendaraan, yaitu motor listrik atau sepeda listrik, dan bobot dari setiap kriteria. Kriteria-kriteria tersebut antara lain adalah harga, *power*, kecepatan maksimum, dan jarak tempuh kendaraan. Hasil yang diberikan dari sistem adalah hasil rekomendasi kendaraan listrik roda dua yang sesuai pilihan *user* berdasarkan perhitungan dengan metode *Simple Additive Weighting*, dimana data normalisasi yang digunakan dalam proses perhitungan tersebut, dapat dilihat oleh *admin*.

Sedangkan entitas *admin* memiliki peran sebagai pengguna yang dapat mengatur seluruh sistem pendukung keputusan ini. *Admin* dapat menambahkan, mengedit, dan menghapus data pada sistem. Data tersebut diantaranya adalah daftar produk, yang juga dapat dilihat oleh entitas *user*. Untuk dapat melakukan pengaturan data, *admin* harus masuk ke dalam sistem dengan melakukan *login*.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



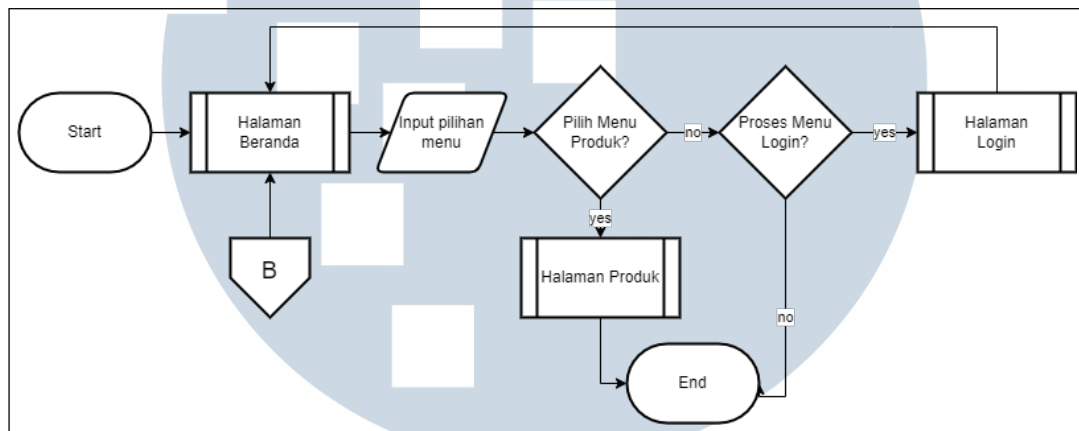
Gambar 3.2. DFD Level 1 situs Torselis

Gambar 3.2 menggambarkan diagram alur data level satu pada sistem. Diagram tersebut menjelaskan beberapa proses yang ada dalam sistem pendukung keputusan ini. Proses-proses tersebut antara lain sebagai berikut :

1. Sistem Rekomendasi SAW adalah proses pengolahan data bobot tiap kriteria beserta data jenis kendaraan yang telah *diinput* oleh *user*. Proses ini terhubung langsung dengan *database*, serta menampilkan hasil rekomendasi yang didapat dari perhitungan menggunakan metode SAW. Dari proses perhitungan tersebut, didapat pula data bobot produk dan data normalisasi yang bisa dilihat oleh admin.
2. Sistem *Login* adalah proses masuk ke dalam sistem yang dilakukan oleh admin untuk mengakses halaman Daftar Produk Motor Listrik dan proses keluar dari sistem oleh admin. Proses ini terhubung dengan *database user* untuk melakukan proses autentikasi.
3. Sistem Pengelolaan Produk adalah proses pengelolaan data produk oleh admin. Admin dapat menambahkan, mengedit, dan menghapus data pada daftar produk yang terhubung langsung dengan *database produk*.

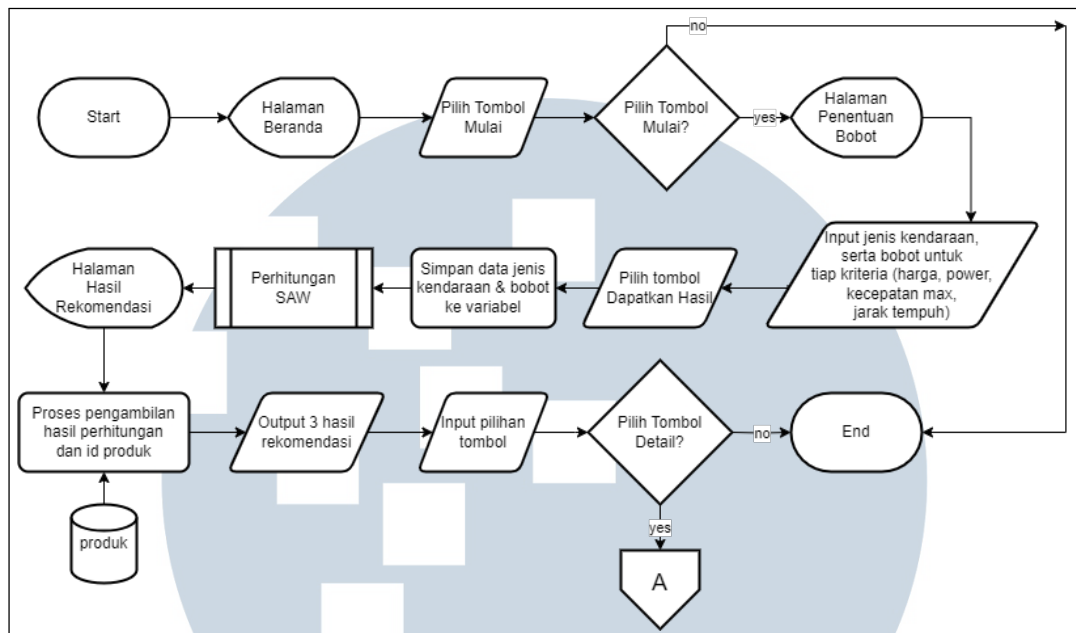
### 3.2.2 Flowchart

*Flowchart* adalah sebuah diagram yang menggambarkan alur proses atau logika suatu sistem, yang digunakan untuk menganalisis, merepresentasikan dan mendokumentasikan proses bisnis, sistem komputer dan *software*. Alur logika pada sistem ini dibahas mulai dari alur situs secara menyeluruh, dilanjutkan dengan alur proses dari menu yang ada, lalu proses masuk ke halaman admin, hingga alur perhitungan metode *Simple Additive Weighting*.



Gambar 3.3. *Flowchart* situs Torselis

Pada Gambar 3.3 menunjukkan alur proses sistem pendukung keputusan secara menyeluruh. Proses dimulai dari proses Halaman Beranda yang kemudian dijelaskan pada Gambar 3.4. Proses itu juga merupakan alur lanjutan dari tahapan yang terdapat pada Gambar 3.6. Proses dilanjutkan dengan *user* memasukkan pilihan menu, yaitu menu Produk atau menu Login. Jika pilihan merupakan menu Produk, maka proses dapat dilanjutkan ke dalam proses Halaman Produk yang dijabarkan pada Gambar 3.5. Kemudian saat proses Halaman Produk telah selesai, maka alur proses berhenti. Sedangkan jika pilihan tertuju pada menu *Login*, maka proses dapat dilanjutkan ke dalam proses Halaman *Login* yang dijelaskan pada Gambar 3.6. Kemudian proses berlanjut ke proses Halaman Beranda saat telah selesai. Saat *user* memutuskan untuk tidak memilih menu apapun, maka alur proses sistem ini berhenti.

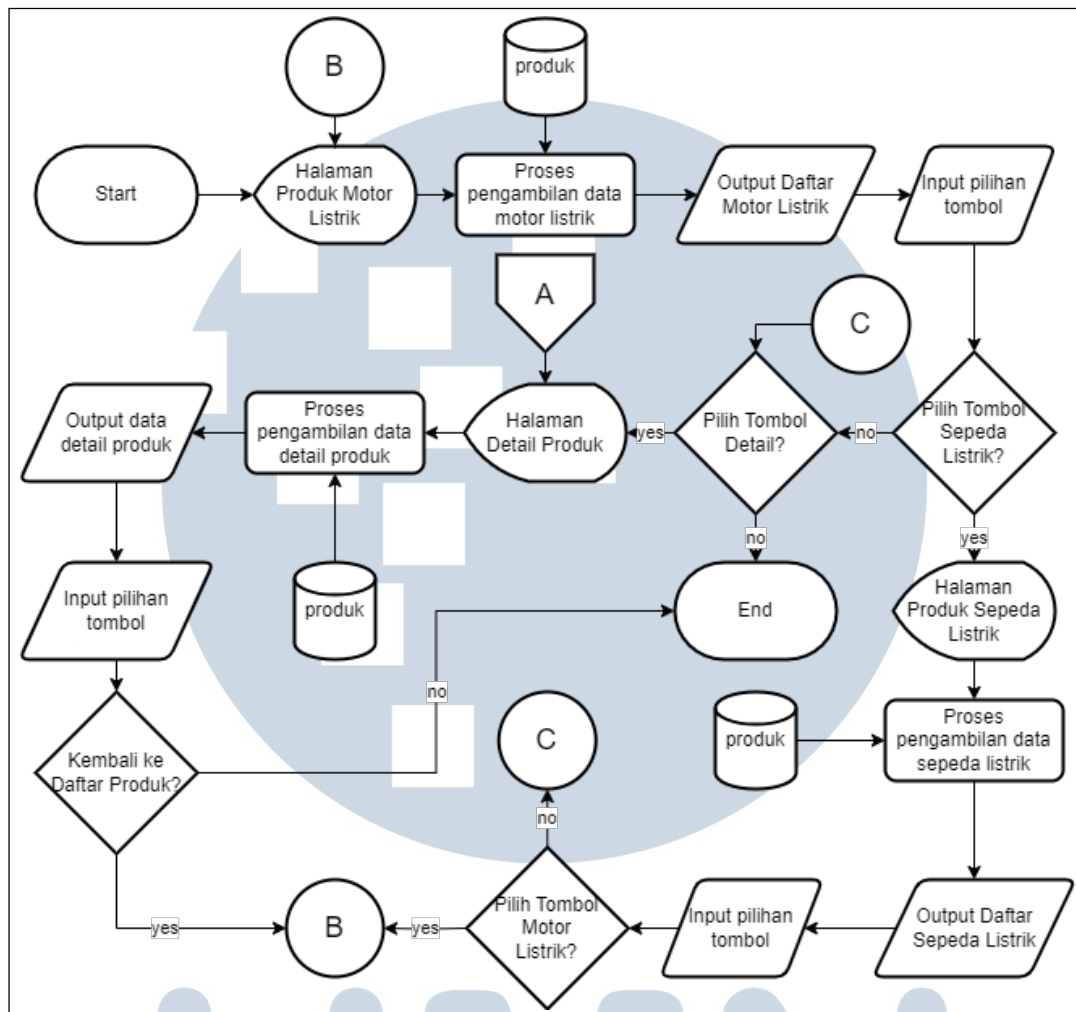


Gambar 3.4. Flowchart halaman Beranda

Gambar 3.4 merupakan rincian dari alur proses Halaman Beranda yang terdapat pada Gambar 3.3. Proses dimulai dengan tampilan halaman Beranda yang meminta *user* untuk memasukkan pilihan pada tombol Mulai. Jika *user* tidak memilih apapun, maka proses berhenti. Sedangkan jika tombol dipilih, maka tampilan berpindah ke halaman Penentuan Bobot. Pada halaman tersebut *user* memasukkan data jenis kendaraan serta bobot untuk masing-masing kriteria yang tersedia. Kriteria tersebut diantaranya adalah kriteria harga, *power*, kecepatan maksimum, dan jarak tempuh. Kemudian *user* dapat memilih tombol Dapatkan Hasil.

Setelah itu, tombol Dapatkan Hasil menyimpan data bobot dan jenis kendaraan yang telah dimasukkan sebelumnya ke dalam variabel. Variabel tersebut kemudian digunakan pada proses selanjutnya, yaitu proses Perhitungan SAW yang dijelaskan pada Gambar 3.11. Saat proses perhitungan selesai, tampilan berpindah ke halaman Hasil Rekomendasi. Pada halaman tersebut dilakukan proses pengambilan variabel hasil perhitungan serta pengambilan data produk, khususnya *id*, dari *database*.

Saat semua data yang dibutuhkan telah didapat, maka tiga hasil rekomendasi muncul. Pada halaman Hasil Rekomendasi, *user* dapat menentukan pilihan. Jika tombol Detail dipilih, maka proses berpindah ke halaman lain yang terdapat pada Gambar 3.5. Sedangkan jika tidak, maka proses selesai.



Gambar 3.5. Flowchart halaman Produk

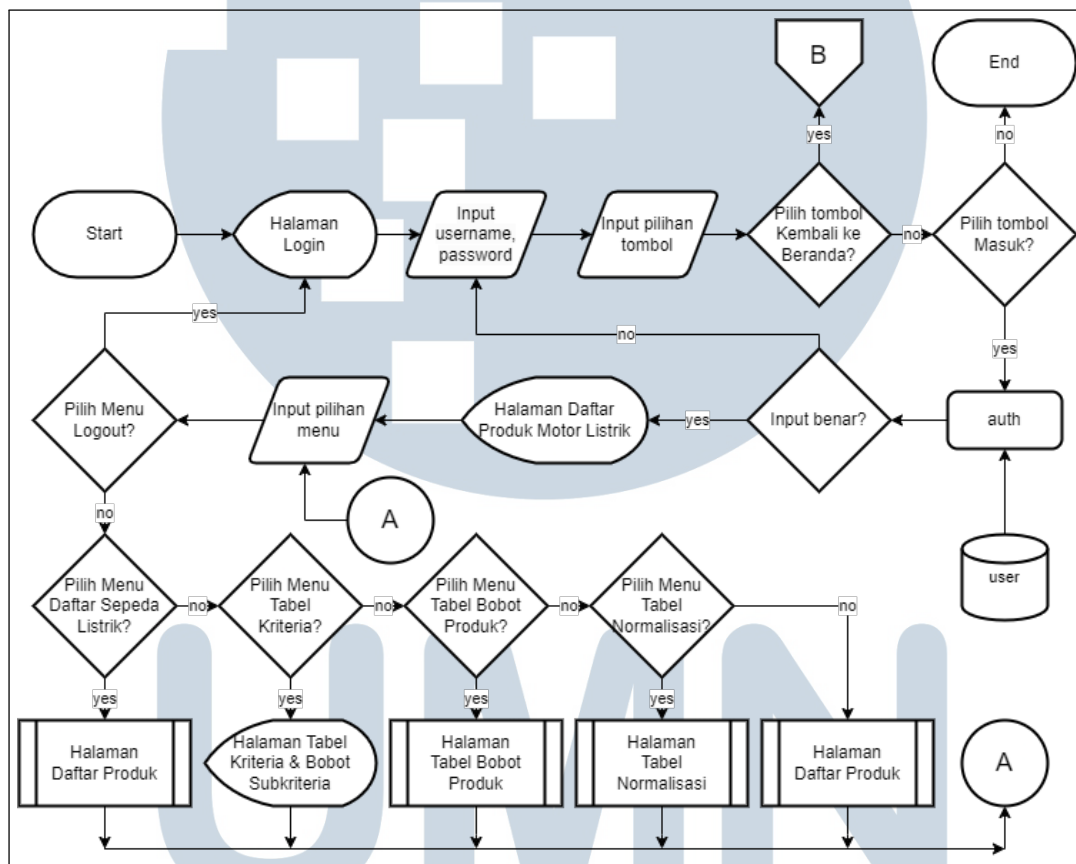
Gambar 3.5 menunjukkan rincian alur proses halaman Produk yang telah dibahas sebelumnya pada Gambar 3.3. Alur dimulai dengan tampilan halaman Produk yang berisikan daftar motor listrik, dilanjutkan dengan proses pengambilan data dari *database*. Saat data telah didapat, maka daftar motor listrik ditampilkan.

Kemudian alur berlanjut dengan *user* memberikan pilihan. Jika tombol Sepeda Listrik dipilih, maka halaman berpindah ke halaman Produk yang berisikan sepeda listrik. Sedangkan jika tidak dan *user* memilih tombol Detail, maka halaman berpindah ke halaman Detail Produk. Saat tidak ada tombol yang dipilih, maka proses berakhir.

Kembali ke bagian halaman Produk Sepeda Listrik, proses pengambilan data dari *database* dilakukan untuk mengeluarkan daftar sepeda listrik. Sama seperti halaman Produk Motor Listrik, terdapat dua pilihan tombol, yaitu tombol yang balik

mengarah ke halaman Produk Motor Listrik, serta tombol Detail.

Pada saat tombol Detail dipilih, halaman berpindah ke halaman Detail Produk. Halaman tersebut juga merupakan lanjutan dari alur proses yang terdapat pada Gambar 3.4. Proses pengambilan data detail produk dilakukan dengan mengambil data dari *database*, yang kemudian ditampilkan bersama dengan tombol Kembali ke Daftar Produk. Jika tombol dipilih, maka alur kembali ke halaman Produk Motor Listrik. Jika tidak, maka alur proses berakhir.



Gambar 3.6. Flowchart halaman Login

Gambar 3.6 menunjukkan rincian alur proses halaman *Login* yang telah dibahas sebelumnya pada Gambar 3.3. Alur proses dimulai dari ditampilkannya halaman *Login* yang dapat diberi masukan data *username* dan *password*. Jika setelah itu *user* memilih tombol Kembali ke Beranda, maka proses berpindah ke alur yang terdapat pada Gambar 3.3. Tetapi jika tombol Masuk dipilih, maka alur berlanjut ke proses autentikasi data. Sedangkan jika tidak ada tombol yang dipilih, maka alur proses berakhir.

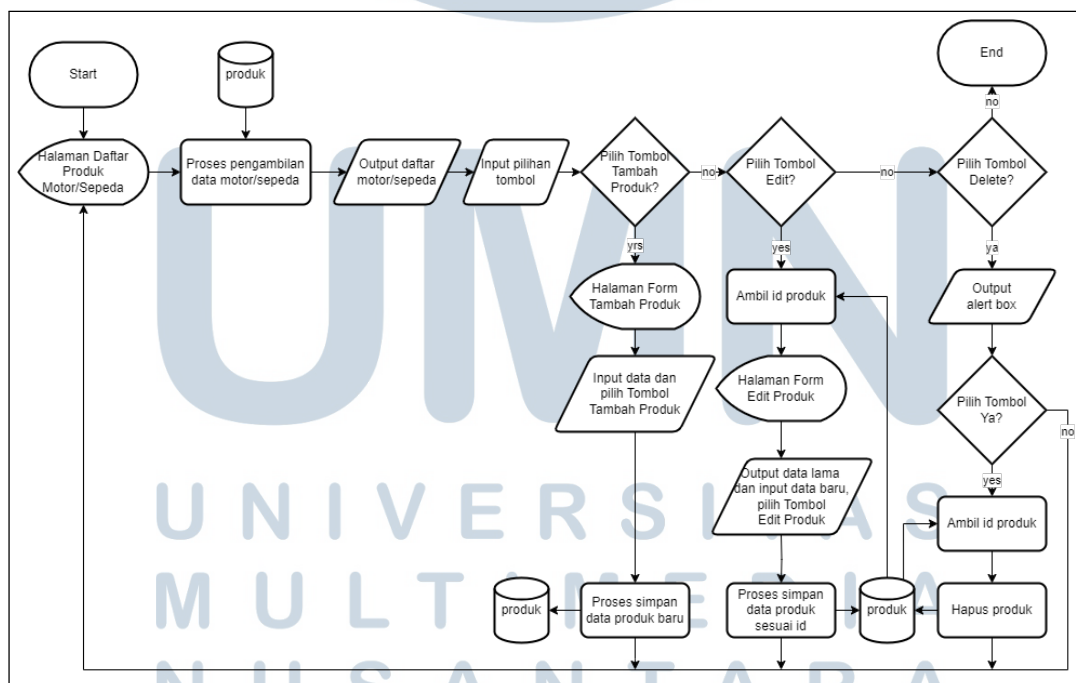
Proses autentikasi dilakukan dengan mengambil data dari *database user*.



Jika data benar, maka alur berpindah ke tampilan halaman Daftar Produk Motor Listrik. Sedangkan jika tidak, maka alur kembali berpindah ke proses *input username* dan *password*. Pada halaman Daftar Produk, terdapat beberapa pilihan menu navigasi, diantaranya menu Daftar Motor Listrik, Daftar Sepeda Listrik, Tabel Kriteria & Bobot Subkriteria, Tabel Bobot Produk, dan menu Tabel Normalisasi.

Jika menu *Logout* dilihat, alur kembali ke tampilan halaman *Login*. Kemudian jika menu Produk Sepeda Listrik dipilih, maka alur berpindah ke proses halaman Daftar Produk Sepeda Listrik yang dijabarkan pada Gambar 3.7. Sedangkan jika menu Tabel Kriteria yang dipilih, maka halaman berpindah ke tampilan halaman Tabel Kriteria & Bobot Subkriteria. Kemudian jika *user* memilih menu Tabel Bobot Produk, maka alur berlanjut ke proses yang terdapat pada Gambar 3.8. Sedangkan jika menu Tabel Normalisasi yang dipilih, maka alur berlanjut ke proses yang dijelaskan pada Gambar 3.9.

Saat *user* tidak memilih menu apapun, maka alur dilanjutkan ke proses halaman Daftar Produk. Proses tersebut sama dengan proses yang muncul ketika menu Produk Sepeda Listrik dipilih. Detail prosesnya juga dijelaskan pada Gambar 3.7. Selanjutnya proses berlanjut dengan kembali memasukkan pilihan menu.



Gambar 3.7. Flowchart halaman Daftar Produk

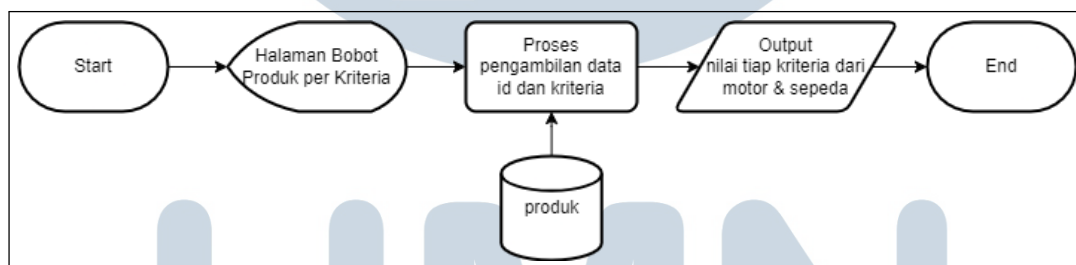
Gambar 3.7 merupakan alur proses yang telah disinggung sebelumnya pada Gambar 3.6. Alur dimulai dengan tampilan halaman Daftar Produk yang memuat

motor atau sepeda listrik. Kemudian dilakukan proses pengambilan data dari *database* untuk menampilkan daftar produk yang sesuai. Pada halaman tersebut terdapat beberapa tombol pilihan, diantaranya tombol Tambah Produk, Edit, dan *Delete* produk.

Saat tombol Tambah Produk dipilih, maka tampilan halaman Form Tambah Produk dimunculkan. Disana dapat dilakukan masukan data serta memilih tombol Tambah Produk. Data yang dimasukkan kemudian disimpan ke dalam *database*.

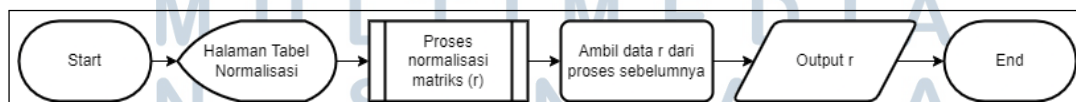
Selanjutnya saat *user* memilih tombol Edit produk, dilakukan proses pengambilan *id* produk. Kemudian tampilan berpindah ke halaman Form Edit Produk, dimana dimunculkan data produk yang lama berdasarkan *id* yang telah diambil. *User* dapat memasukkan data produk yang baru dan memilih tombol Edit Produk. Setelah itu data baru tersebut disimpan ke dalam *database*.

Sedangkan saat tombol *Delete* produk dipilih, muncul *alert box* berisi tombol Ya dan Tidak. Jika tombol Ya dipilih, maka data *id* diambil dari *database* dan produk dihapus juga dari *database*. Semua proses sebelumnya kemudian kembali ke alur tampilan halaman Daftar Produk.



Gambar 3.8. *Flowchart* halaman Tabel Bobot Produk

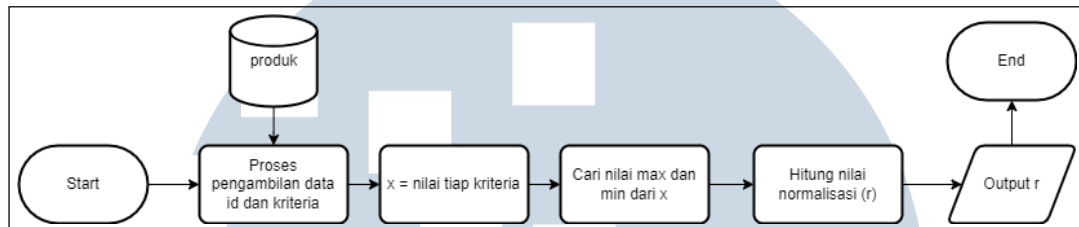
Gambar 3.8 merupakan penjelasan dari alur proses yang terdapat pada Gambar 3.6. Alur ini dimulai dengan tampilan halaman Bobot Produk per Kriteria. Pada halaman tersebut, dilakukan proses pengambilan data *id* dan bobot produk setiap kriteria dari *database*. Selanjutnya nilai yang sudah diambil sebelumnya ditampilkan berdasar kelompok motor & sepeda listrik.



Gambar 3.9. *Flowchart* halaman Tabel Normalisasi

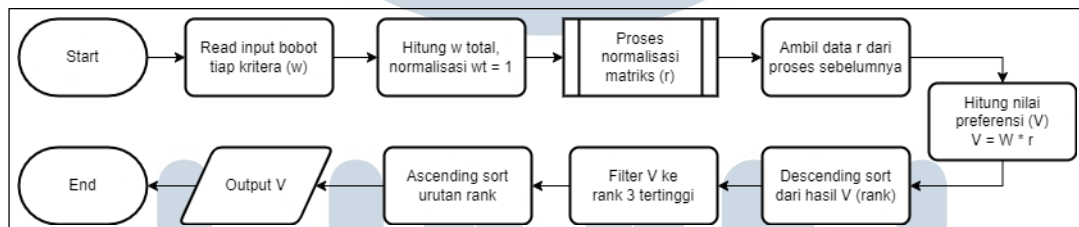
Gambar 3.9 yaitu alur halaman Tabel Normalisasi yang merupakan kelanjutan dari proses yang terdapat pada Gambar 3.6. Alur ini dimulai dengan

munculnya tampilan halaman Tabel Normalisasi. Dilanjutkan dengan proses normalisasi matriks yang dijabarkan pada Gambar 3.10. Setelah proses selesai, maka nilai normalisasi ( $r$ ) diambil dan ditampilkan, lalu alur ini berhenti.



Gambar 3.10. Flowchart proses normalisasi matriks

Seperti yang dibahas pada Gambar 3.9, proses normalisasi matriks dibahas pada Gambar 3.10. Alur ini dimulai dengan proses pengambilan data *id* dan nilai setiap kriteria dari *database*. Kemudian data tersebut disimpan ke dalam variabel. Lalu alur dilanjutkan dengan mencari nilai maksimum dan minimum dari nilai tiap kriteria. Nilai tersebut digunakan dalam perhitungan nilai normalisasi. Selanjutnya alur ini berhenti setelah nilai  $r$  dikeluarkan.



Gambar 3.11. Flowchart Perhitungan SAW

Alur proses yang terakhir ditampilkan pada Gambar 3.11. Alur ini dimulai dengan mengambil data bobot kriteria ( $w$ ) yang telah didapat dari proses sebelumnya pada Gambar 3.4 dan mencari nilai total dari  $w$  serta menormalisasikannya menjadi nilai 1. Kemudian proses normalisasi matriks kembali dilakukan seperti pada Gambar 3.10.

### 3.2.3 Database Schema

*Database schema* adalah sebuah rancangan *database* yang menunjukkan hubungan satu data dengan yang lain yang terdapat pada sistem. *Database* yang digunakan dalam pengerjaan sistem ini adalah phpMyAdmin. Terdapat dua tabel

yaitu produk dan *user*. Skema tersebut dapat dilihat pada Gambar 3.12, dimana antar tabel tersebut tidak terdapat hubungan satu sama lain.



Gambar 3.12. Database Schema situs Torselis

### 3.2.4 Table Structure

Struktur tabel *database* pada perancangan sistem Torselis dapat mulai dijabarkan dari Tabel 3.1.

#### A Tabel Data Admin

Tabel 3.1 merupakan tabel *user*, yaitu tabel yang memiliki fungsi untuk proses autentikasi data *login* dan masuk ke dalam halaman admin.

Tabel 3.1. Tabel admin

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	int	11	<i>id</i> admin
<i>username</i>	varchar	100	nama admin
<i>password</i>	varchar	100	kata sandi

## B Tabel Data Produk

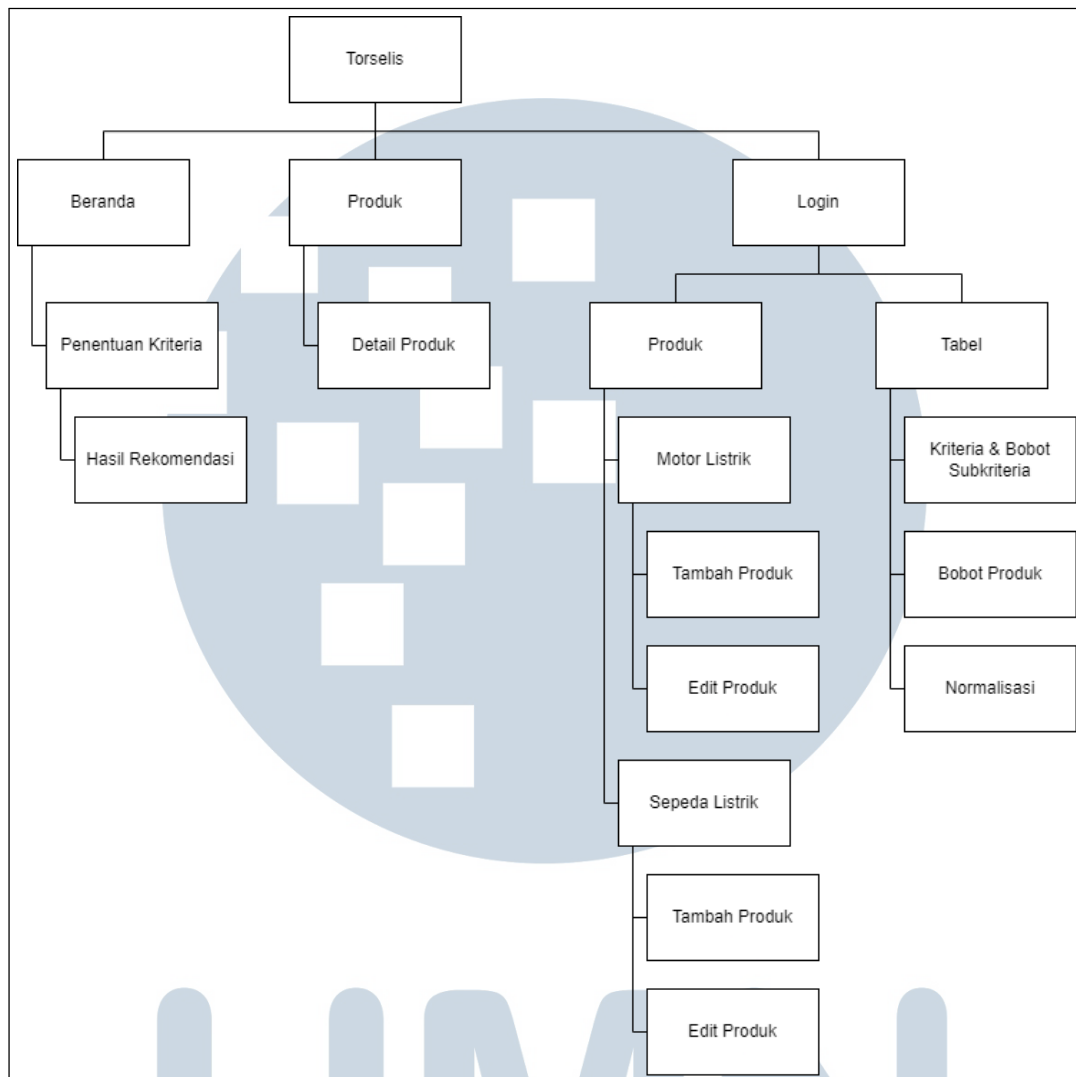
Tabel 3.2 merupakan tabel produk, yaitu tabel yang berfungsi untuk menyimpan informasi terkait produk yang ada pada situs Torselis. Informasi tersebut dapat berasal dari *form* tambah produk maupun edit produk. Untuk kolom *id*, *slug*, *created\_at*, *updated\_at*, dan *deleted\_at* memiliki data yang ditulis oleh sistem.

Tabel 3.2. Tabel produk

Nama Kolom	Tipe Data	Panjang	Keterangan
<i>id</i>	int	11	<i>id</i> kendaraan
<i>jenis_kendaraan</i>	int	11	kode kendaraan
<i>nama</i>	varchar	255	nama kendaraan
<i>slug</i>	varchar	255	nama yang dipisahkan dengan <i>hyphen</i>
<i>harga</i>	int	11	harga kendaraan dalam rupiah
<i>power</i>	int	11	<i>power</i> kendaraan dalam watt
<i>kecepatan_max</i>	int	11	kecepatan kendaraan dalam km/h
<i>jarak_tempuh</i>	int	11	jarak tempuh kendaraan dalam kilometer
<i>baterai</i>	varchar	50	kapasitas baterai kendaraan
<i>daya_angkut</i>	int	11	daya angkut kendaraan dalam kilogram
<i>warna</i>	varchar	255	warna kendaraan
<i>sistem_rem</i>	varchar	255	<i>brake system</i> kendaraan
<i>gambar</i>	varchar	255	foto kendaraan
<i>created_at</i>	datetime		tanggal pembuatan data produk
<i>updated_at</i>	datetime		tanggal pembaharuan data produk
<i>deleted_at</i>	datetime		tanggal penghapusan data produk

### 3.2.5 Sitemap

*Sitemap* adalah peta yang mencakup semua halaman dalam website atau navigasi utama untuk mengetahui halaman yang terdapat dalam web. *Sitemap* berfungsi memudahkan proses untuk menemukan *landing page*, keterkaitan satu halaman dengan halaman lain, dan relevansi setiap halaman [25]. *Sitemap* pada sistem dapat dilihat pada Gambar 3.13.

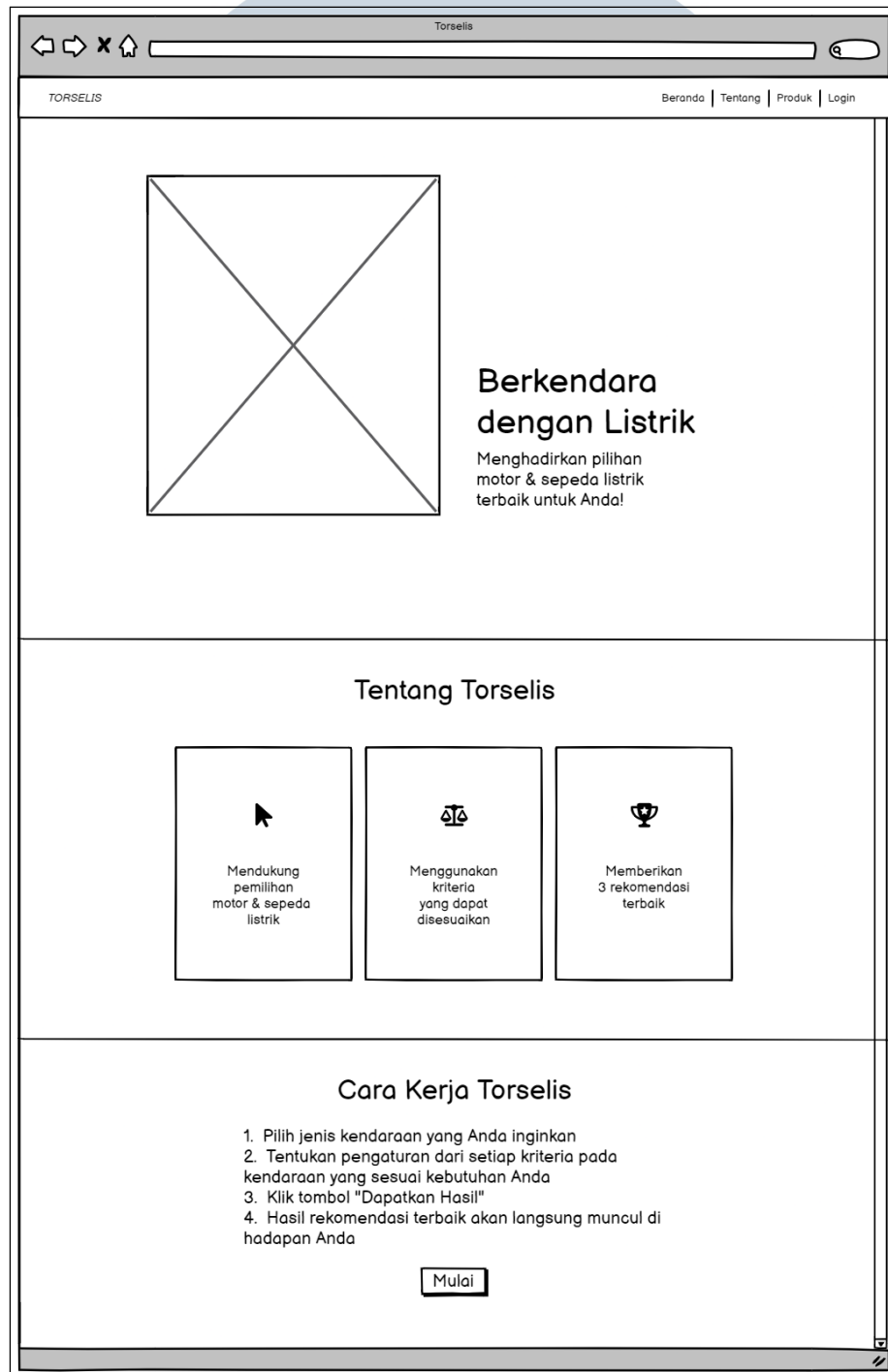


Gambar 3.13. Sitemap situs Torselis

### 3.2.6 Perancangan Desain Antarmuka

Pada tahapan ini ditampilkan gambaran kasar dari *website* yang dibangun. Gambar 3.14 sampai Gambar 3.23 hanya sebatas *wireframe* yang bertujuan untuk mengatur tata letak elemen, sehingga dapat memberikan visualisasi singkat sebelum tahap implementasi dimulai. *Wireframe* tersebut dibuat menggunakan aplikasi Balsamiq. Terdapat beberapa bagian desain antarmuka pada *website* ini, yaitu desain halaman Beranda, Penentuan Kriteria, Hasil Rekomendasi, Daftar Produk, Detail Produk, *Login* Admin, Daftar Produk (pada halaman admin), Formulir Tambah Produk, Tabel Kriteria & Bobot Subkriteria, dan yang terakhir halaman Bobot Produk.

Perancangan desain antarmuka halaman Beranda dapat dilihat pada Gambar 3.14 berikut ini.



Gambar 3.14. Desain antarmuka halaman Beranda

Gambar 3.14 merupakan desain antarmuka halaman Beranda yang berisi berbagai informasi singkat mengenai *website*, salah satunya tentang bagaimana cara mendapatkan rekomendasi kendaraan listrik pada situs tersebut. Halaman ini merupakan halaman yang diakses pertama kali oleh pengunjung. Pada halaman ini terdapat papan navigasi yang berfungsi untuk mengakses halaman dan fitur lain yang terdapat pada sistem. Tombol Mulai yang terdapat pada bagian bawah halaman berfungsi untuk berpindah ke halaman Penentuan Kriteria yang digambarkan pada Gambar 3.15.

TORSSELIS Beranda | Tentang | Produk | Login

1. Pilih Jenis Kendaraan

Motor Listrik  Sepeda Listrik

2. Tentukan Pengaturan Tiap Kriteria

Harga

Power

Kecepatan Max

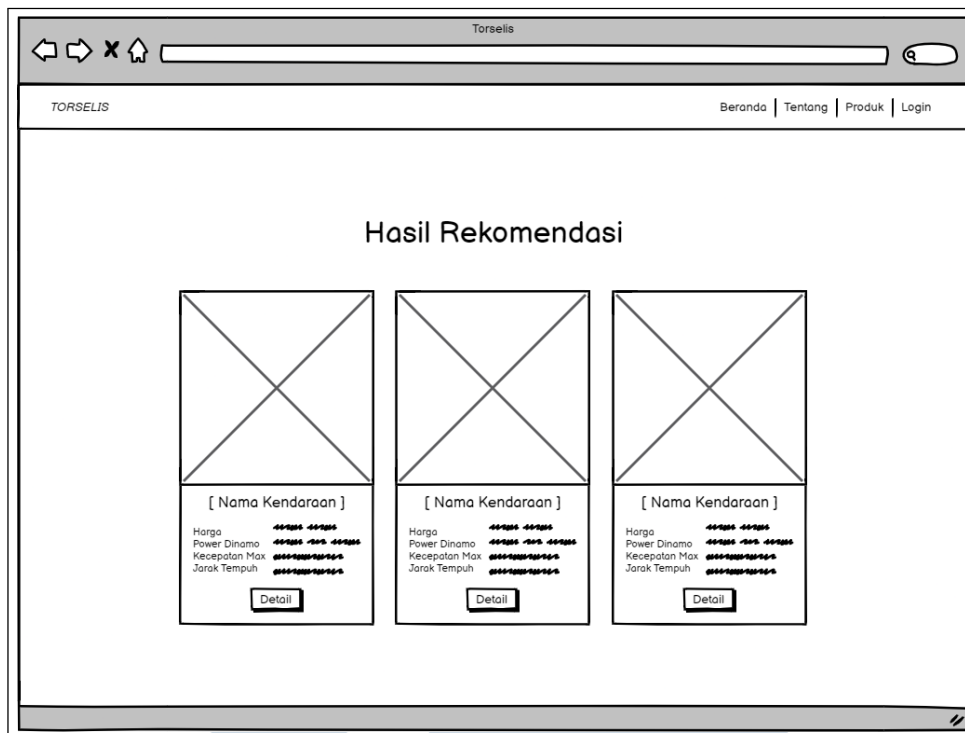
Jarak Tempuh

3. Klik Tombol

Gambar 3.15. Desain antarmuka halaman Penentuan Kriteria

Setelah menekan tombol Mulai tampilan berpindah ke halaman Penentuan Kriteria seperti pada Gambar 3.15. Pada halaman ini terdapat tiga tahapan dalam mendapatkan hasil rekomendasi kendaraan listrik. Yang pertama yaitu memilih jenis kendaraan pada *radio button*, lalu dilanjutkan dengan menentukan pengaturan tiap kriteria. Kriteria-kriteria tersebut memiliki kolom *input* berupa *dropdown*. Tahap yang terakhir yaitu menekan tombol Dapatkan Hasil yang mengarah ke halaman Hasil Rekomendasi seperti pada Gambar 3.16.

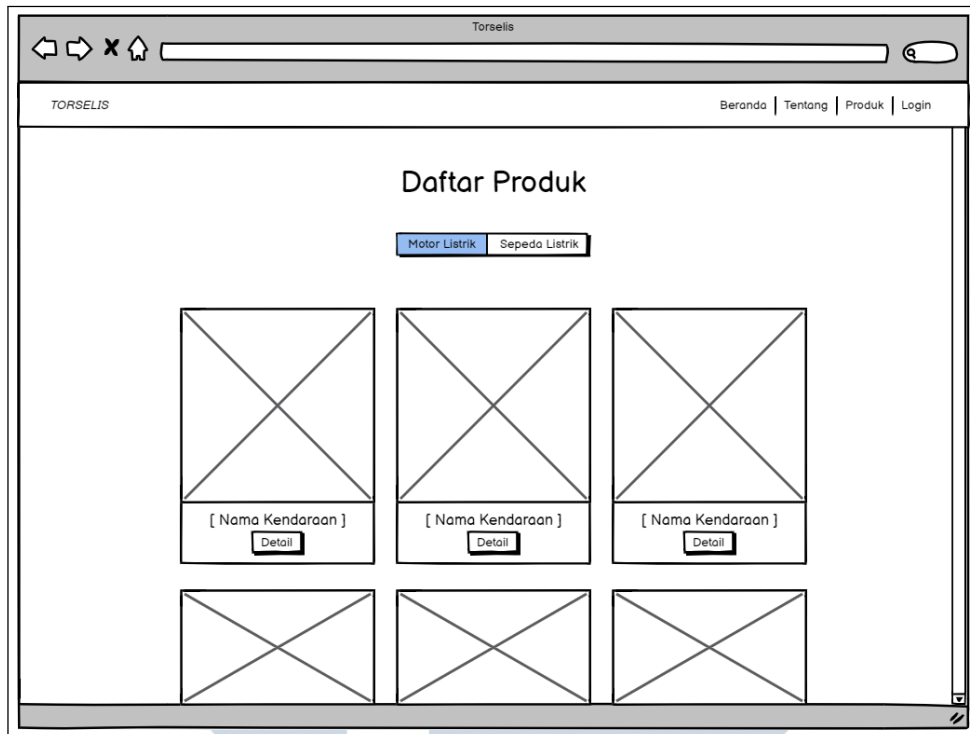




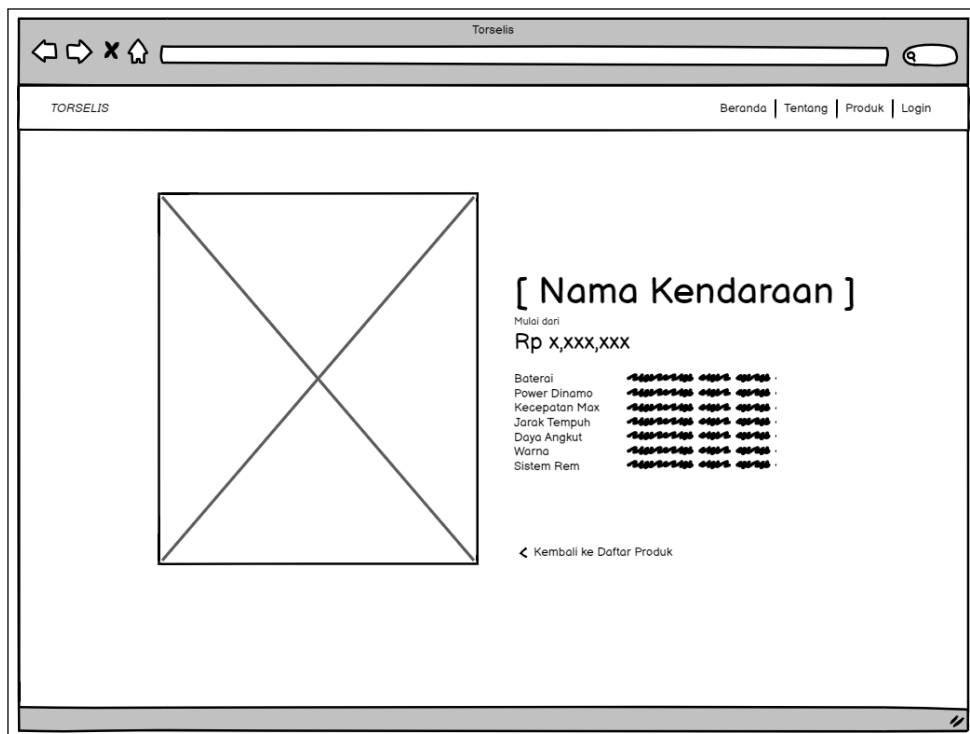
Gambar 3.16. Desain antarmuka halaman Hasil Rekomendasi

Gambar 3.16 merupakan kelanjutan dari halaman Penentuan Kriteria. Halaman ini dirancang untuk menampilkan tiga hasil kendaraan listrik terbaik setelah melalui proses perhitungan menggunakan metode *Simple Additive Weighting*. Informasi yang ditampilkan hanya sebatas gambar, nama kendaraan, harga, *power*, kecepatan maksimum, jarak tempuh, serta tombol Detail.

Kemudian Gambar 3.17 merupakan desain antarmuka halaman Produk yang berisi daftar produk kendaraan listrik roda dua, yaitu motor listrik dan sepeda listrik. Di bawah judul halaman, terdapat tombol yang berfungsi untuk berpindah halaman sesuai jenis produk yang dipilih. Informasi yang tersedia pada masing-masing *card* produk yaitu sebatas nama kendaraan serta tombol Detail yang mengarah ke Gambar 3.18.

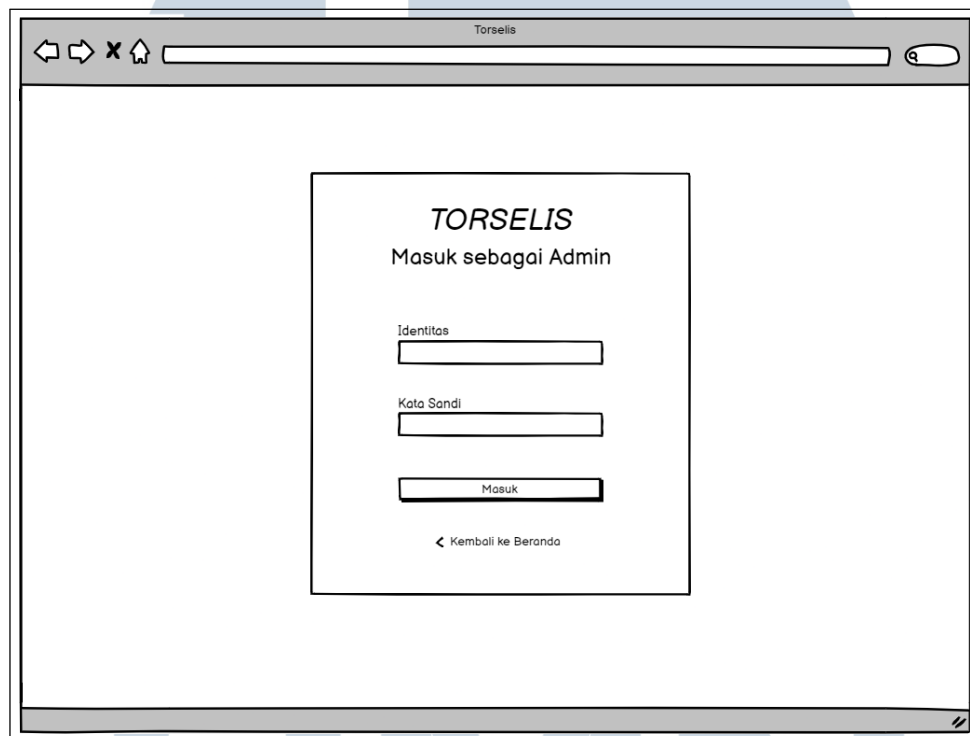


Gambar 3.17. Desain antarmuka halaman Daftar Produk



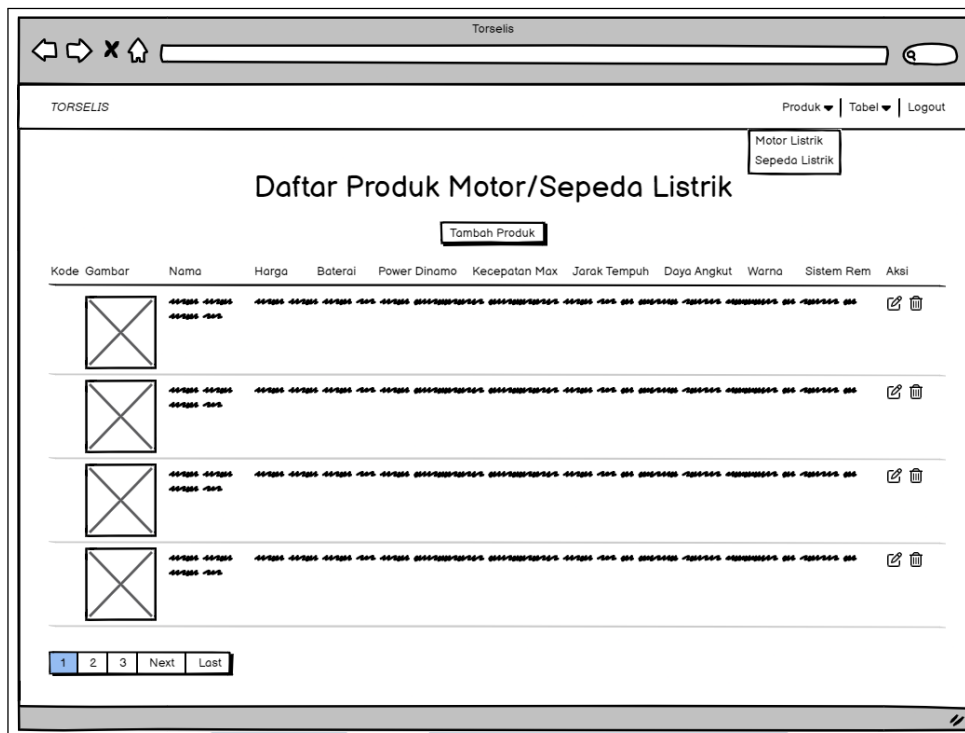
Gambar 3.18. Desain antarmuka halaman Detail Produk

Gambar 3.18 merupakan desain tampilan ketika pengguna ingin melihat informasi mendetail terkait produk. Terdapat data yang ditampilkan berupa nama kendaraan, harga, kapasitas baterai, *power* dinamo, kecepatan maksimum, jarak tempuh, daya angkut, warna, serta sistem rem. Di bawah data tersebut, terdapat tombol yang mengarah balik ke halaman Daftar Produk yaitu halaman yang terdapat pada Gambar 3.17.



Gambar 3.19. Desain antarmuka halaman *Login Admin*

Gambar 3.19 merupakan desain antarmuka halaman *Login* yang berisi *form input* identitas serta kata sandi. Halaman ini tidak memiliki menu navigasi. Adapun saat tombol *Masuk* dipilih dan proses autentikasi berhasil, maka halaman berpindah ke halaman *Daftar Produk* yang terdapat pada Gambar 3.20.



Gambar 3.20. Desain antarmuka halaman Daftar Produk

Gambar 3.20 merupakan desain antarmuka halaman Daftar Produk, yaitu halaman yang pertama kali diakses saat memasuki halaman admin. Desain antarmuka tersebut berlaku untuk daftar produk motor listrik maupun sepeda listrik. Pada halaman admin, terdapat menu navigasi yang mengarah ke beberapa halaman, diantaranya menu Produk dan menu Tabel. Adapun isi dari halaman tersebut adalah judul halaman, tombol Tambah Produk yang mengarah ke halaman pada Gambar 3.21, serta tabel yang berisi informasi lengkap mengenai produk yang tersimpan dalam sistem. Salah satu kolomnya, yaitu kolom Aksi, memiliki tombol untuk mengedit informasi produk maupun menghapusnya. Kemudian di bawah tabel tersebut terdapat tombol halaman yang berfungsi sebagai *pagination* agar data produk yang ditampilkan tidak terlalu banyak.

The screenshot shows a web browser window with the title 'Torselis'. The page content includes a navigation bar with 'TORSELIS' on the left and 'Produk ▼ | Tabel ▼ | Logout' on the right. The main heading is 'Form Tambah/Edit Produk'. The form fields are arranged in two columns:

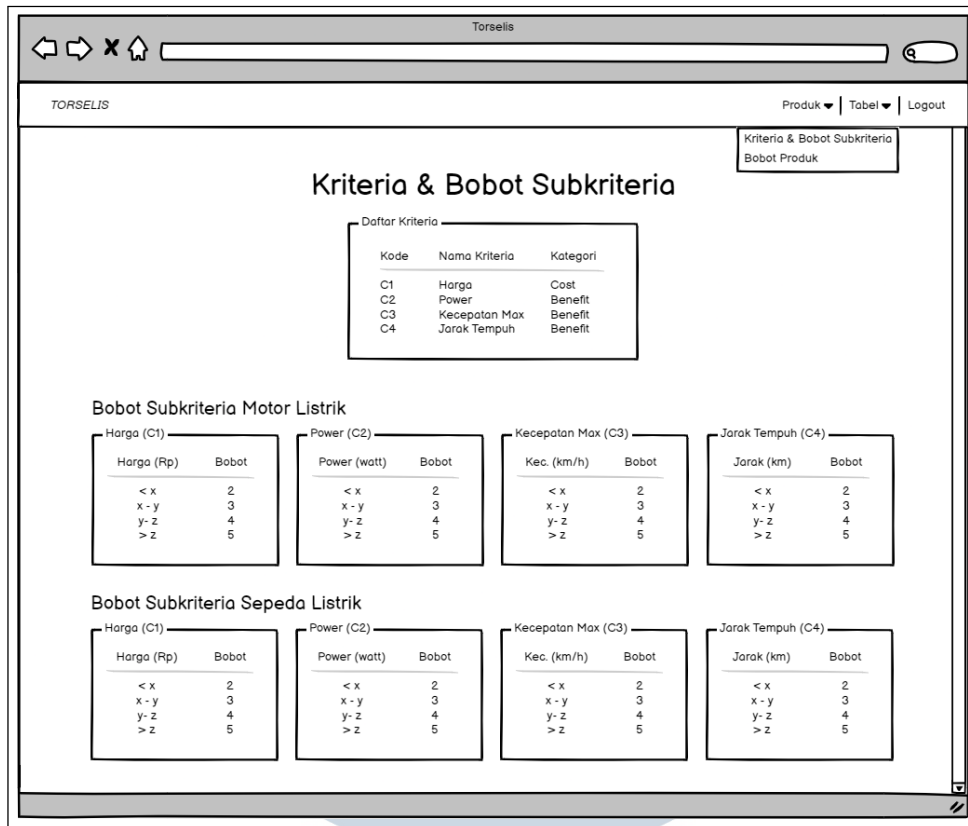
- Column 1: Nama, Harga (Rp) (containing '1650000'), Power (watt), Kecepatan Max (km/h), Jarak Tempuh (km).
- Column 2: Baterai (containing '60 V 20 Ah'), Daya Angkut (kg), Warna (containing 'hitam, biru'), Sistem Rem (containing 'Drum/Disc brake'), Gambar (with 'Choose File' and 'No file chosen...' buttons).

A 'Tambah Produk' button is centered below the form fields.

Gambar 3.21. Desain antarmuka halaman Tambah Produk

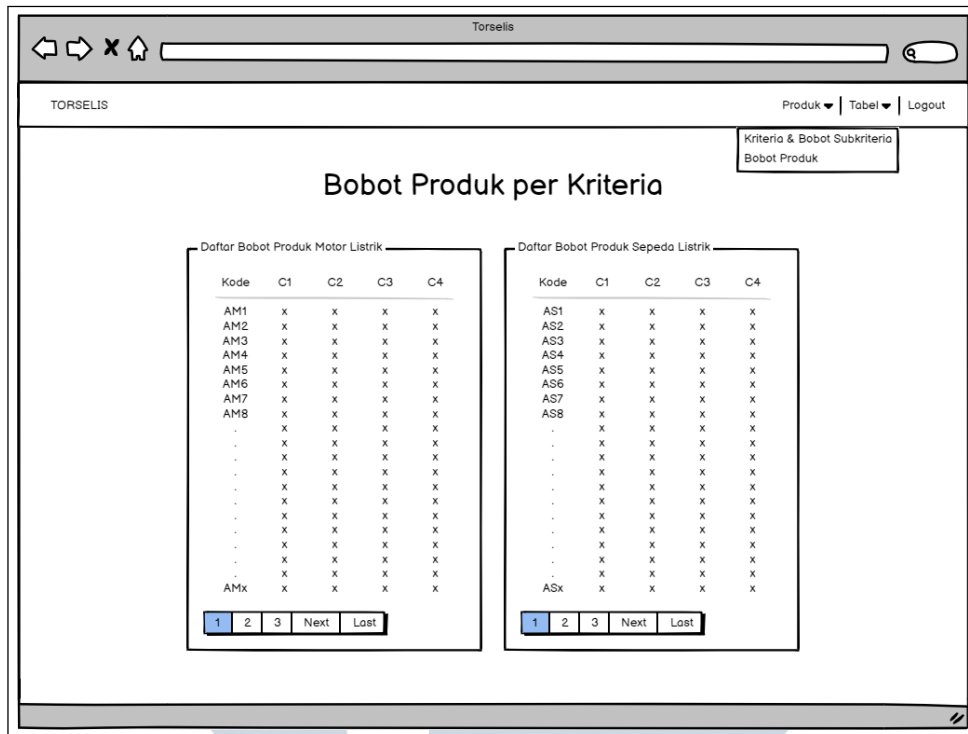
Gambar 3.21 merupakan desain antarmuka halaman Tambah Produk yang berisi sebuah *form*. Kolom *input* tersebut dapat diisi oleh pengguna serta terdapat bagian untuk mengunggah gambar. Kemudian di bawah kolom *input* tersebut terdapat tombol Tambah Produk yang berfungsi untuk menyimpan produk yang baru saja ditambahkan dan tampilan berpindah ke halaman Daftar Produk seperti pada Gambar 3.20. Desain antarmuka halaman ini berlaku juga untuk halaman Edit Produk.

U M M N  
 UNIVERSITAS  
 MULTIMEDIA  
 NUSANTARA



Gambar 3.22. Desain antarmuka halaman Kriteria & Bobot Subkriteria

Gambar 3.22 merupakan desain antarmuka halaman Kriteria & Bobot Subkriteria. Halaman ini hanya terbatas pada tampilan informasi mengenai daftar kriteria yang terdapat pada sistem, kategori atribut dari setiap kriteria, serta bobot subkriteria yang dijadikan acuan pembobotan pada halaman Penentuan Kriteria, yang terdapat pada Gambar 3.15.



Gambar 3.23. Desain antarmuka halaman Bobot Produk

Gambar 3.23 merupakan desain antarmuka halaman Bobot Produk yang berfungsi untuk menampilkan daftar bobot atau nilai alternatif dari seluruh motor maupun sepeda listrik. Di bawah setiap bagian daftar, terdapat tombol yang berguna sebagai *pagination* agar informasi yang ditampilkan tidak terlalu banyak. Desain antarmuka ini juga berlaku untuk halaman Tabel Normalisasi. Tata letak dan bagiannya sama, hanya isinya saja yang berbeda karena melalui proses perhitungan yang berbeda pula.

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

## BAB 4 HASIL DAN DISKUSI

### 4.1 Spesifikasi Sistem

Dalam pembangunan sistem, bahasa pemrograman yang digunakan dalam membuat aplikasi adalah PHP dengan CodeIgniter 4 sebagai *framework* pendukung, serta MySQL sebagai basis data. Kemudian dalam prosesnya tersebut, digunakan perangkat pendukung berupa *hardware* dan *software*. *Hardware* yang digunakan adalah laptop dengan spesifikasi

1. Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz
2. RAM 4.00 GB
3. GPU NVIDIA GeForce 920MX

Sedangkan spesifikasi *software* yang digunakan dalam pengembangan sistem ini adalah sebagai berikut

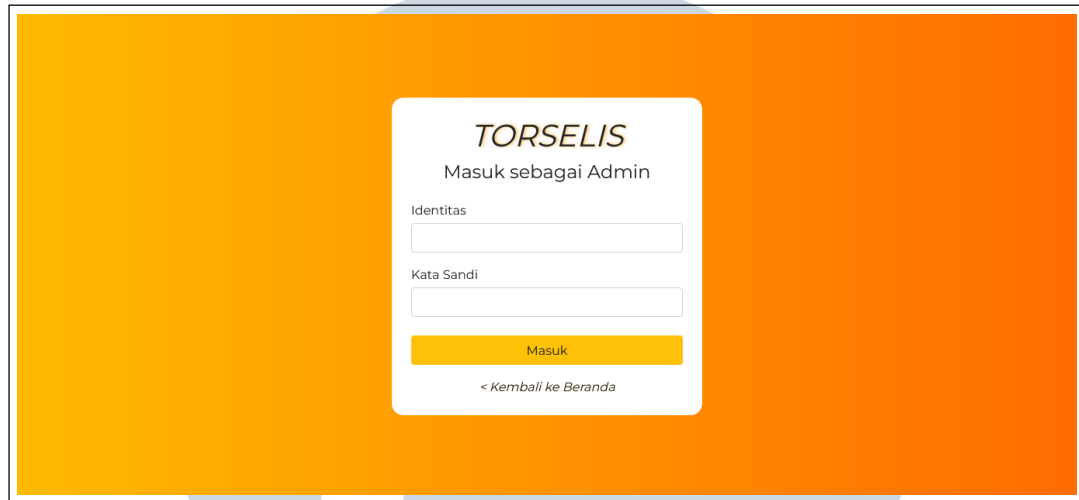
1. Sistem Operasi Windows
2. Google Chrome
3. Visual Studio Code
4. XAMPP v3.3.0
5. phpMyAdmin v5.1.1
6. Balsamiq

### 4.2 Hasil Implementasi Desain Antarmuka

Desain antarmuka pada Subbab 3.2.6 yang telah diimplementasikan dapat dilihat pada Gambar 4.1 hingga Gambar 4.11. Gambar-gambar tersebut merupakan hasil akhir dari *website* yang telah dibangun. Pada sistem ini terdapat dua peran pengguna, yaitu *user* biasa serta admin. *User* biasa hanya memiliki akses pada halaman Beranda, halaman Produk beserta detailnya, halaman Penentuan Kriteria, serta halaman Hasil Rekomendasi. Sedangkan admin memiliki seluruh akses pada sistem.



#### 4.2.1 Halaman Login



**TORSELIS**  
Masuk sebagai Admin

Identitas

Kata Sandi

Masuk

[< Kembali ke Beranda](#)

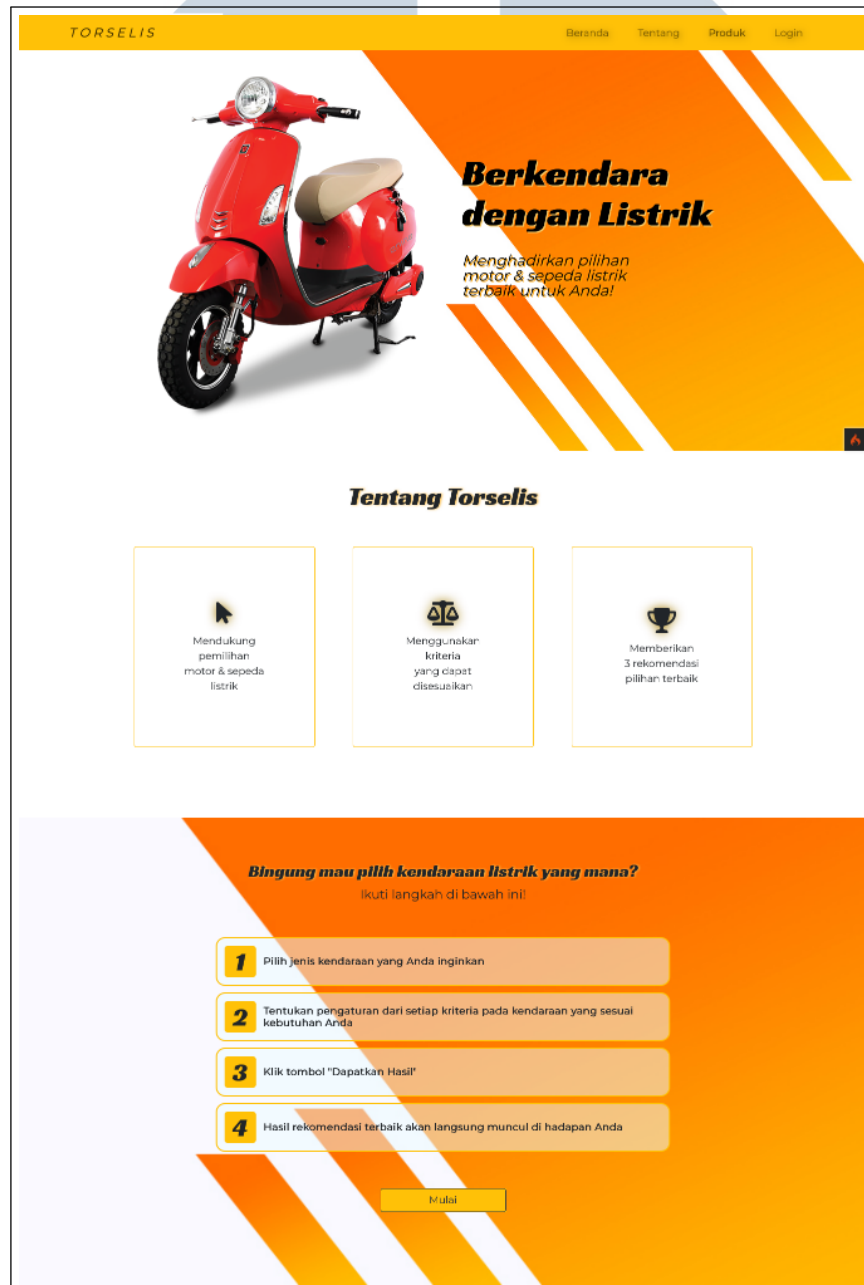
Gambar 4.1. Antarmuka halaman Login

Gambar 4.1 merupakan antarmuka halaman *Login* yang dapat diakses seluruh pengguna tetapi hanya dapat berfungsi untuk admin. Halaman ini merupakan tampilan yang muncul saat pengguna menekan pilihan yang terdapat pada menu navigasi di halaman utama. Pada halaman ini admin hanya perlu mengisi kolom *input* identitas dan kata sandi untuk masuk ke dalam halaman admin dengan menekan tombol *Masuk*. Tombol yang mengarah kembali ke halaman *Beranda* pun tersedia di bawah tombol *Masuk*. Saat identitas dan kata sandi benar dan telah terdaftar pada sistem, maka tampilan dapat berpindah ke halaman *Daftar Produk* seperti yang terlihat pada Gambar 4.7.

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

## 4.2.2 Antarmuka Sistem pada Sisi User

### A Halaman Beranda



Gambar 4.2. Antarmuka halaman Beranda

Gambar 4.2 merupakan halaman yang pertama kali diakses saat pengguna mengunjungi situs Torselis. Terdapat beberapa informasi terkait situs tersebut serta cara kerja sistem dalam memberikan rekomendasi kendaraan listrik. Tombol Mulai yang terdapat pada *section* terakhir merupakan tombol yang mengarah ke halaman Penentuan Kriteria seperti pada Gambar 4.3.

## B Halaman Penentuan Kriteria

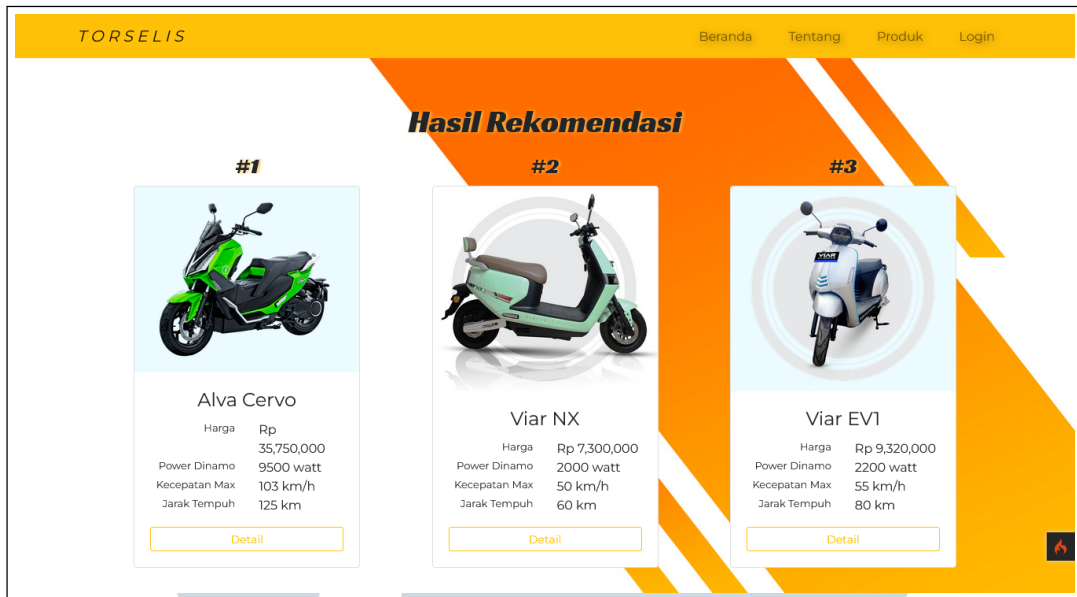
Gambar 4.3 merupakan antarmuka halaman Penentuan Kriteria yang berisi *form*. Pengguna dapat memilih jenis kendaraan serta pengaturan untuk setiap kriteria yang dijadikan acuan perhitungan sistem. Saat tombol Dapatkan Hasil dipilih, maka seluruh *input* disimpan ke dalam variabel yang dapat mengeluarkan hasil rekomendasi pada halaman selanjutnya seperti pada Gambar 4.4.

The screenshot shows the 'Penentuan Kriteria' (Criteria Selection) page on the Torselis website. The page features a yellow header with the 'TORSELIS' logo and navigation links for 'Beranda', 'Tentang', 'Produk', and 'Login'. The main content area is white with a large orange and yellow diagonal graphic. It contains three sections: 1. 'Pilih Jenis Kendaraan' (Choose Vehicle Type) with two radio buttons: 'Motor Listrik' (unselected) and 'Sepeda Listrik' (selected). 2. 'Tentukan Pengaturan Tiap Kriteria' (Determine Settings for Each Criterion) with four dropdown menus: 'Harga' (Price), 'Power', 'Kecepatan Max' (Maximum Speed), and 'Jarak Tempuh' (Range), each with 'Pilih kriteria' (Choose criteria) as the selected option. 3. '3. Klik Tombol' (Click Button) with a yellow 'Dapatkan Hasil' (Get Results) button.

Gambar 4.3. Antarmuka halaman Penentuan Kriteria

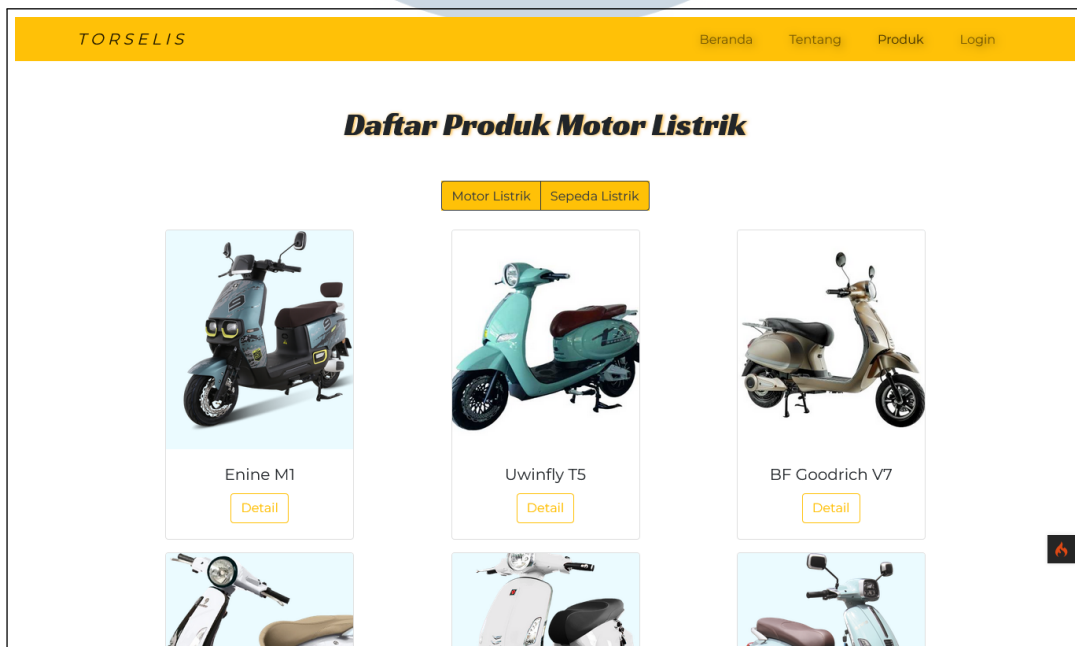
## C Halaman Hasil Rekomendasi

Pada Gambar 4.4 muncul tiga hasil kendaraan yang paling sesuai dengan pengaturan kriteria yang dipilih oleh pengguna. Hasil tersebut telah diurutkan dari peringkat tertinggi hingga peringkat ketiga. Informasi yang terdapat pada halaman ini terbatas pada gambar, nama kendaraan, harga, *power*, kecepatan maksimum, serta jarak tempuh. Jika pengguna ingin melihat informasi produk secara lebih lengkap, dapat diakses tombol Detail yang mengarah ke halaman Detail Produk seperti yang dapat dilihat pada Gambar 4.6.



Gambar 4.4. Antarmuka halaman Hasil Rekomendasi

## D Halaman Produk

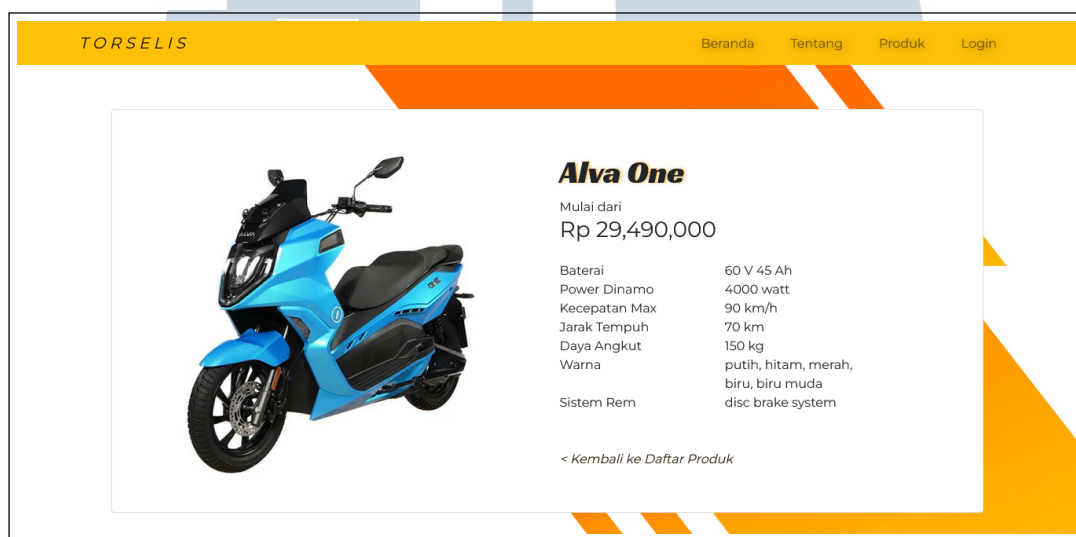


Gambar 4.5. Antarmuka halaman Produk

Gambar 4.5 merupakan antarmuka yang muncul ketika pengguna memilih menu Produk pada menu navigasi. Seluruh produk yang terdaftar pada sistem ditampilkan pada halaman ini berdasarkan jenis kendaraannya. Informasi yang

ditampilkan hanya terdapat gambar, nama kendaraan, serta tombol Detail. Untuk mengubah jenis kendaraan, pengguna dapat memilih tombol yang terdapat di bawah judul halaman. Sedangkan untuk melihat informasi lengkap dari produk, pengguna dapat menekan tombol Detail yang mengarah ke halaman Detail Produk seperti pada Gambar 4.6.

## E Halaman Detail Produk



Gambar 4.6. Antarmuka halaman Detail Produk

Gambar 4.6 merupakan antarmuka halaman Detail Produk yang dapat diakses dengan menekan tombol Detail pada halaman Daftar Produk maupun halaman Hasil Rekomendasi. Pada halaman ini ditampilkan gambar, nama, harga, dan informasi lengkap dari produk tersebut. Selain itu terdapat tombol untuk kembali ke halaman Daftar Produk yang terletak di bawah informasi produk.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 4.2.3 Antarmuka Sistem pada Sisi Admin

#### A Halaman Daftar Produk

Gambar 4.7 merupakan halaman Daftar Produk, khususnya produk sepeda listrik, yang dapat diakses setelah admin berhasil masuk ke dalam sistem. Hasil implementasi halaman pada produk motor listrik dan sepeda listrik memiliki tampilan yang sama. Pada halaman ini terdapat tombol Tambah Produk yang mengarah ke halaman Tambah Produk seperti pada Gambar 4.8, serta terdapat tabel yang berisi seluruh informasi mengenai produk yang terdaftar pada sistem. Karena informasi yang ada terlalu banyak, maka isi dari tabel tersebut dapat digeser saat admin ingin melihat keseluruhan datanya, serta terdapat tombol edit dan *delete* pada ujung kanan tabel. Adapun di bawah tabel terdapat tombol angka yang bertujuan sebagai penanda halaman yang bertujuan untuk membatasi jumlah produk yang ditampilkan.

No	Kode	Gambar	Nama	Harga	Baterai	Power Dinamo	Kecepatan Max	Jarak Tempuh	Daya Angkut	Warna
1	AM2		Uwinfly Kitty	8,990,000	48 V 20 Ah	500	46	40	150	merah, kuning, pin biru, hijau
2	AM4		Goda 140	3,599,000	48 V 12 Ah	500	40	40	130	Kuning, putih, hitam biru, hijau
3	AM6		Lisgo Tuniu	4,000,000	48 V 12 Ah	500	35	65	200	Pink, hijau, biru
4	AM24		Ofero Stareer Max	7,499,000	48 V 20 Ah	600	46	85	180	Cokelat, hijau
5	AM26		Enine SE-8S	2,999,000	48 V 12 Ah	500	35	40	150	hijau muda, coklat hijau tua, biru muda biru tua

Gambar 4.7. Antarmuka halaman Daftar Produk

## B Halaman Tambah Produk

Gambar 4.8 merupakan antarmuka halaman Tambah Produk yang dapat diakses setelah menekan tombol Tambah Produk pada halaman Daftar Produk. Halaman ini berisi *form* yang terdiri beberapa kolom *input* seperti nama, spesifikasi kendaraan, hingga gambar kendaraan. Pada bagian akhir *form* terdapat tombol Tambah Produk yang berfungsi untuk menyimpan data ke dalam *database* kemudian memindahkan tampilan ke halaman Daftar Produk. Halaman Tambah Produk ini memiliki tata letak yang sama dengan halaman Edit Produk. Perbedaannya hanya terletak pada judul halaman serta kolom *input* yang sudah terisi dengan data yang tersedia.

The screenshot shows a web form titled "Form Tambah Produk" on a yellow and orange background. The form is organized into two columns of input fields. The left column includes: "Nama" (text input), "Harga (Rp)" (text input with a currency symbol), "Power (watt)" (text input with a unit symbol), "Kecepatan Max (km/h)" (text input with a unit symbol), "Jarak Tempuh (km)" (text input with a unit symbol), and "Preview" (text input). The right column includes: "Baterai" (text input), "Daya Angkut (kg)" (text input with a unit symbol), "Warna" (text input), "Sistem Rem" (text input), "Jenis Kendaraan" (dropdown menu showing "Motor Listrik"), and "Gambar" (image upload area with a "Browse..." button and a "No file ...lected." message). A yellow "Tambah Produk" button is positioned at the bottom center of the form. The top navigation bar features the "TORSELIS" logo on the left and "Produk", "Tabel", and "Logout" links on the right.

Gambar 4.8. Antarmuka halaman Tambah Produk

## C Halaman Tabel Kriteria & Bobot Subkriteria

Gambar 4.9 merupakan antarmuka halaman Tabel Kriteria dan Bobot Subkriteria yang berisi beberapa tabel. Tabel yang pertama merupakan daftar kriteria yang digunakan pada sistem, sedangkan kelompok tabel yang kedua dan ketiga merupakan daftar bobot subkriteria dari setiap produk, yaitu motor listrik

dan sepeda listrik, yang digunakan dalam perhitungan menggunakan metode *Simple Additive Weighting*. Seluruh tabel tersebut merupakan tabel statis, dimana data yang ditampilkan hanya berupa tampilan dan tidak terhubung ke dalam *database*.

**Kriteria & Bobot Subkriteria**

Kode	Nama Kriteria	Kategori
C1	Harga	Cost
C2	Power	Benefit
C3	Kecepatan Max	Benefit
C4	Jarak Tempuh	Benefit

**Bobot Subkriteria Motor Listrik**

Harga (C1)	Bobot	Power (C2)	Bobot	Kecepatan Max (C3)	Bobot	Jarak Tempuh (C4)	Bobot
Murah	4	Kecil	1	Rendah	1	Dekat	1
Sedang	3	Sedang	2	Biasa	2	Sedang	2
Cukup Mahal	2	Besar	3	Cukup Tinggi	3	Cukup Jauh	3
Mahal	1	Sangat Besar	4	Tinggi	4	Jauh	4

Gambar 4.9. Antarmuka halaman Tabel Kriteria & Bobot Subkriteria

#### D Halaman Tabel Bobot Produk

**Bobot Produk per Kriteria**

Nilai Alternatif Motor Listrik				
Kode	C1	C2	C3	C4
AM1	15990000	1000	50	70
AM3	12899000	2000	75	65
AM5	11500000	1000	45	55
AM7	12299000	2000	65	75
AM8	12999000	2000	65	75
AM9	12499000	1200	60	70
AM10	17999000	2000	75	70
AM11	21999000	2000	75	70
AM12	21970000	5000	70	50
AM13	27980000	3000	70	60

Nilai Alternatif Sepeda Listrik				
Kode	C1	C2	C3	C4
AS2	8990000	500	46	40
AS4	3599000	500	40	40
AS6	4000000	500	35	65
AS24	7499000	600	46	85
AS26	2999000	500	35	40
AS27	5500000	800	55	40
AS28	5699000	500	55	65
AS29	6999000	600	46	85
AS30	8799000	1200	55	80
AS40	4699000	500	35	65

Gambar 4.10. Antarmuka halaman Bobot Produk



Gambar 4.10 merupakan antarmuka halaman Bobot Produk yang berisi nilai dari setiap alternatif pada setiap kriteria. Nilai yang ditampilkan hanya berupa kode beserta keempat kriteria yang digunakan pada sistem, yaitu harga (C1), *power* (C2), kecepatan maksimum (C3), dan jarak tempuh (C4). Seluruh nilai tersebut diambil dari *database* untuk merepresentasikan nilai rating kecocokan yang merupakan tahapan pada perhitungan menggunakan metode *Simple Additive Weighting*. Agar jumlah data yang ditampilkan tidak terlalu banyak, maka di bawah tabel terdapat tombol berupa angka untuk membatasi data.

### E Halaman Tabel Normalisasi

Gambar 4.11 merupakan halaman Tabel Normalisasi yang berisi hasil perhitungan dari normalisasi nilai yang terdapat pada Gambar 4.10. Seluruh nilai tersebut merupakan nilai olahan dari *database*, dimana nilai dapat berubah saat ada produk baru yang ditambahkan maupun diedit. Tabel pada halaman tersebut bertujuan untuk merepresentasikan nilai normalisasi yang merupakan salah satu tahap dalam perhitungan menggunakan metode *Simple Additive Weighting*.

Nilai Normalisasi Motor Listrik				
Kode	C1	C2	C3	C4
AM1	0.4565	0.1053	0.4854	0.5385
AM3	0.5659	0.2105	0.7282	0.5000
AM5	0.6348	0.1053	0.4369	0.4231
AM7	0.5935	0.2105	0.6311	0.5769
AM8	0.5616	0.2105	0.6311	0.5769
AM9	0.5840	0.1263	0.5825	0.5385
AM10	0.4056	0.2105	0.7282	0.5385
AM11	0.3318	0.2105	0.7282	0.5385
AM12	0.3323	0.5263	0.6796	0.3846
AM13	0.2609	0.3158	0.6796	0.4615
AM14	0.3480	0.3158	0.6796	0.4615

Nilai Normalisasi Sepeda Listrik				
Kode	C1	C2	C3	C4
AS2	0.3336	0.4167	0.8364	0.4706
AS4	0.8333	0.4167	0.7273	0.4706
AS6	0.7498	0.4167	0.6364	0.7647
AS24	0.3999	0.5000	0.8364	1.0000
AS26	1.0000	0.4167	0.6364	0.4706
AS27	0.5453	0.6667	1.0000	0.4706
AS28	0.5262	0.4167	1.0000	0.7647
AS29	0.4285	0.5000	0.8364	1.0000
AS30	0.3408	1.0000	1.0000	0.9412
AS40	0.6382	0.4167	0.6364	0.7647
AS41	0.6249	0.4167	0.9091	0.4706

Gambar 4.11. Antarmuka halaman Tabel Normalisasi

### 4.3 Hasil Implementasi Perhitungan dengan Metode SAW

Potongan kode yang digunakan dalam mengembangkan sistem terdiri atas beberapa *function*, yaitu *function* `__construct`, `getWeight`, `indexNormalisasi`, `getPreferensi`, serta `getRank`. *Function* `__construct` merupakan *function* yang dibuat untuk memanggil model produk, dengan alasan model tersebut sering digunakan pada setiap *function*, sehingga model tersebut tidak perlu dipanggil secara berulang. Potongan kode dari *function* tersebut dapat dilihat pada Kode 4.1.

```
1 public function __construct() {  
2     $this->produkModel = new ProdukModel();  
3 }
```

Kode 4.1: Potongan kode *Function* `__construct`

Saat tombol Dapatkan Hasil pada halaman Penentuan Kriteria yang terdapat pada Gambar 4.3 dipilih, maka dipanggil *function* `getWeight` seperti yang terlihat pada Kode 4.2. Nilai dari jenis kendaraan yang dipilih kemudian tersimpan pada variabel *type*, sedangkan nilai dari setiap kriteria disimpan sebagai bobot pada variabel `w1` hingga `w4`. Kemudian nilai dari setiap bobot dijumlahkan untuk dijadikan pembagi saat melakukan konversi bobot dan didapatkan nilai bobot yang baru.

Selanjutnya *function* `getPreferensi` yang terdapat pada Kode 4.6 dipanggil dengan mengirimkan variabel dari jenis kendaraan serta bobot setiap kriteria pada parameternya, dimana hasil dari *function* tersebut berupa *array* dan disimpan ke dalam variabel `$V`. Setelah itu nilai pada variabel `V` dijadikan parameter saat memanggil *function* `getRank` seperti yang terdapat pada Kode 4.7. Nilai yang dikembalikan oleh *function* `getRank` kemudian disimpan ke dalam variabel `Vr`.

Untuk mengambil nilai dari variabel `Vr` dengan peringkat tertinggi, yaitu peringkat satu hingga tiga, maka dibutuhkan pencarian data dalam *array* menggunakan *foreach*. Saat data dengan peringkat satu hingga tiga ditemukan, maka data tersebut disimpan dalam variabel `Vr_result` yang berupa *array*. Karena peringkat dalam *array* `Vr_result` masih dalam urutan acak, maka *usort* digunakan untuk mengurutkan *array* menggunakan fungsi perbandingan. Hasil akhir dari variabel `Vr_result` kemudian disimpan dalam variabel data yang digunakan untuk menampilkan hasil rekomendasi pada halaman Hasil Rekomendasi seperti pada Gambar 4.4.

```
1 public function getWeight() {  
2     // get product type & weight
```

```

3     $type = $this->request->getVar('type');
4     $w1 = $this->request->getVar('w_c1');
5     $w2 = $this->request->getVar('w_c2');
6     $w3 = $this->request->getVar('w_c3');
7     $w4 = $this->request->getVar('w_c4');
8
9     // convert weight
10    $sum_w = $w1 + $w2 + $w3 + $w4;
11    $w1 = $w1 / $sum_w;
12    $w2 = $w2 / $sum_w;
13    $w3 = $w3 / $sum_w;
14    $w4 = $w4 / $sum_w;
15
16    $V = $this->getPreferensi($type, $w1, $w2, $w3, $w4);
17    $Vr = $this->getRank($V);
18
19    // filter Vr to third highest rank
20    $Vr_result = [];
21    foreach ($Vr as $Vr) {
22        if (in_array($Vr['rank'], [1, 2, 3]))
23            $Vr_result[] = $Vr;
24    }
25
26    // sort result based on rank
27    usort($Vr_result, function ($a, $b) {
28        return $a['rank'] - $b['rank'];
29    });
30
31    $data = [
32        'title' => 'Hasil Rekomendasi | Torselis',
33        'preferensi' => $Vr_result
34    ];
35    return view('home/result', $data);
36 }

```

Kode 4.2: Potongan kode *Function* getWeight

*Function* indexNormalisasi pada Kode 4.3 berfungsi untuk memanggil *function* getNormalisasiMolis dan getNormalisasiSelis yang terdapat pada Kode 4.4 dan Kode 4.5, serta menyimpan hasil dari *function* tersebut pada variabel data untuk ditampilkan pada halaman Tabel Normalisasi yang terdapat pada Gambar 4.11.

```

1 public function indexNormalisasi(){
2     $molis = $this->getNormalisasiMolis();

```

```

3     $selis = $this->getNormalisasiSelis();
4
5     $data = [
6         'title' => 'Normalisasi | Torselis',
7         'molis' => $molis,
8         'selis' => $selis
9     ];
10    return view('admin/saw_table/normalisasi', $data);
11 }

```

Kode 4.3: Potongan kode *Function* indexNormalisasi

*Function* getNormalisasiMolis pada Kode 4.4 di bawah ini memiliki fungsi untuk menormalisasi nilai dari setiap kriteria dari jenis kendaraan motor listrik. *Function* tersebut awalnya memanggil model *database* dari produk motor listrik yang disimpan pada variabel *molis*. Kemudian data yang didapat dilakukan pencarian nilai tertinggi ataupun terendah dari setiap kolom kriteria. Karena kriteria harga memiliki atribut *cost*, maka cukup nilai minimum yang dicari pada kolom harga. Sedangkan kriteria *power*, kecepatan maksimum, dan jarak tempuh cukup dicari nilai maksimumnya karena memiliki atribut *benefit*. kemudian seluruh nilai minimum maupun maksimum tersebut disimpan ke dalam variabel.

Tahap selanjutnya yaitu mencari nilai *r* atau nilai normalisasi. Berdasarkan Rumus 2.1, saat atribut berupa *cost*, maka nilai minimum dari kriteria tersebut dibagi dengan nilai yang sedang dihitung. Oleh karena itu, pada baris ke-15 Kode 4.4 diterapkan rumus tersebut dengan kriteria harga sebagai kolomnya, kemudian disimpan ke dalam variabel *array* *r\_m1*. Sedangkan untuk kriteria dengan atribut *benefit* memiliki perhitungan yaitu nilai yang sedang dihitung dibagi dengan nilai maksimumnya. Seluruh nilai yang telah dinormalisasi kemudian disimpan ke dalam variabel *r* untuk dikembalikan ke *function* yang memanggilnya, yaitu *function* indexNormalisasi pada Kode 4.3 serta *function* getPreferensi pada Kode 4.6.

```

1 public function getNormalisasiMolis(){
2     // get products
3     $molis = $this->produkModel->getMolis();
4
5     // get min max
6     // cost-min: harga, benefit-max: power, kec, jarak
7     $min_c1 = min(array_column($molis, 'harga'));
8     $max_c2 = max(array_column($molis, 'power'));
9     $max_c3 = max(array_column($molis, 'kecepatan_max'));
10    $max_c4 = max(array_column($molis, 'jarak_tempuh'));

```

```

11
12 // count r
13 $r = [];
14 foreach ($molis as $m) {
15     $m['r_m1'] = $min_c1 / $m['harga'];
16     $m['r_m2'] = $m['power'] / $max_c2;
17     $m['r_m3'] = $m['kecepatan_max'] / $max_c3;
18     $m['r_m4'] = $m['jarak_tempuh'] / $max_c4;
19     $r[] = $m;
20 }
21 return $r;
22 }

```

Kode 4.4: Potongan kode *Function* getNormalisasiMolis

Hampir sama dengan Kode 4.4, *function* getNormalisasiSelis pada Kode 4.5 memiliki perhitungan yang sama. Perbedaan dari keduanya hanya terletak pada model *database* yang dipanggil, yaitu getSelis. Kemudian *function* ini juga mengembalikan nilai r kepada *function* yang memanggilnya, yaitu *function* indexNormalisasi pada Kode 4.3 serta *function* getPreferensi pada Kode 4.6.

```

1 public function getNormalisasiSelis(){
2     // get products
3     $selis = $this->produkModel->getSelis();
4
5     // get min max
6     // cost-min: harga, benefit-max: power, kec, jarak
7     $min_c1 = min(array_column($selis, 'harga'));
8     $max_c2 = max(array_column($selis, 'power'));
9     $max_c3 = max(array_column($selis, 'kecepatan_max'));
10    $max_c4 = max(array_column($selis, 'jarak_tempuh'));
11
12    // count r
13    $r = [];
14    foreach ($selis as $s) {
15        $s['r_s1'] = $min_c1 / $s['harga'];
16        $s['r_s2'] = $s['power'] / $max_c2;
17        $s['r_s3'] = $s['kecepatan_max'] / $max_c3;
18        $s['r_s4'] = $s['jarak_tempuh'] / $max_c4;
19        $r[] = $s;
20    }
21    return $r;
22 }

```

Kode 4.5: Potongan kode *Function* getNormalisasiSelis

*Function* `getPreferensi` pada Kode 4.6 merupakan *function* yang dipanggil oleh *function* `getWeight` pada Kode 4.2 dan memiliki fungsi untuk menghitung nilai preferensi atau nilai *V*. *Function* tersebut mendapatkan kiriman variabel *type*, *w1*, *w2*, *w3*, dan *w4* yang dapat digunakan dalam perhitungan mencari nilai *V*. Pertama-tama kode memeriksa apakah nilai dari variabel *type* atau jenis kendaraan bernilai satu atau dua. Jika satu, yang berarti kendaraan tersebut merupakan motor listrik, maka *function* `getPreferensi` memanggil *function* `getNormalisasiMolis` untuk mendapatkan nilai *r* yang kemudian disimpan ke dalam variabel *r*. Sedangkan jika jenis kendaraan bernilai 2, yang berarti sepeda listrik, maka *function* `getNormalisasi` yang dipanggil.

Nilai *r* yang dikembalikan setelah memanggil kedua *function* kemudian disimpan per kolom *array*-nya ke dalam variabel *r1* hingga *r4*. Selanjutnya nilai *r* tersebut digunakan untuk menghitung nilai *V*, yaitu dengan melakukan perkalian bobot dengan nilai *r*, atau variabel *w1* hingga *w4* dengan variabel *r1* hingga *r4*. Seluruh perkalian tersebut kemudian disimpan ke dalam variabel *r* pada *V1* hingga *V4* untuk dijumlahkan dan didapatkan nilai *V* total atau variabel *Vt*. Setiap proses *looping* tersebut, nilai pada variabel *r* disimpan ke dalam *array* dengan variabel *V*. Saat proses selesai, maka nilai pada variabel *V* dikembalikan kepada *function* yang memanggilnya.

```

1 public function getPreferensi($type, $w1, $w2, $w3, $w4){
2     // get r and set new var name
3     if ($type == '1') {
4         $r = $this->getNormalisasiMolis();
5         $r1 = array_column($r, 'r_m1');
6         $r2 = array_column($r, 'r_m2');
7         $r3 = array_column($r, 'r_m3');
8         $r4 = array_column($r, 'r_m4');
9     } elseif ($type == '2') {
10        $r = $this->getNormalisasiSelis();
11        $r1 = array_column($r, 'r_s1');
12        $r2 = array_column($r, 'r_s2');
13        $r3 = array_column($r, 'r_s3');
14        $r4 = array_column($r, 'r_s4');
15    }
16
17    // count v = sum(w*r)
18    $V = [];
19    $i = 0;
20    foreach ($r as $r) {

```

```

21     $r['V1'] = $w1 * $r1[$i];
22     $r['V2'] = $w2 * $r2[$i];
23     $r['V3'] = $w3 * $r3[$i];
24     $r['V4'] = $w4 * $r4[$i];
25     $r['Vt'] = $r['V1'] + $r['V2'] + $r['V3'] + $r['V4'];
26     $V[] = $r;
27     $i++;
28 }
29 return $V;
30 }

```

Kode 4.6: Potongan kode *Function* getPreferensi

Kode 4.7 merupakan potongan kode dari *function* getRank yang dipanggil oleh *function* getWeight pada Kode 4.2. *Function* ini mendapatkan kiriman variabel V untuk dilakukan pemberian peringkat. Nilai pada kolom Vt dalam *array* V disimpan ke dalam variabel Vt dan digunakan pada fungsi *arsort* yang berfungsi mengurutkan nilai dalam *array* secara *descending* atau dari nilai terbesar.

Kemudian nilai pada variabel Vt digunakan pada fungsi *array\_search*, dengan tujuan untuk mencari sebuah nilai yang ada pada *array* dengan variabel vt, dan memunculkan *output* berupa *key* yang memiliki nilai yang di cari pada variabel Vt. *Key* tersebut digunakan sebagai acuan untuk pemberian peringkat pada kolom rank pada variabel V. Sehingga saat proses selesai, nilai pada variabel V yang telah mengandung kolom rank disimpan ke dalam variabel Vr untuk dikembalikan.

```

1 public function getRank($V){
2     $Vt = array_column($V, 'Vt');
3     arsort($Vt);
4
5     $Vr = [];
6     $rank = 1;
7     foreach ($Vt as $vt) {
8         $i = array_search($vt, $V);
9         $V[$i]['rank'] = $rank++;
10        $Vr = $V;
11    }
12    return $Vr;
13 }

```

Kode 4.7: Potongan kode *Function* getRank

#### 4.4 Hasil Uji Skenario Sistem terhadap Perhitungan Manual

Untuk menguji ketepatan hasil perhitungan yang dilakukan oleh sistem, diperlukan perhitungan secara manual. Langkah-langkah perhitungan dengan menggunakan metode *Simple Additive Weighting* (SAW) sudah dijelaskan pada Subbab 2.3. Pembahasan dan perhitungan manual terkait penerapan metode SAW pada sistem dapat dilihat dari penjelasan berikut ini.

##### 1. Menentukan Kriteria ( $C_i$ )

Penentuan kriteria yang dijadikan acuan dalam pendukung keputusan pemilihan kendaraan listrik roda dua dilakukan dengan wawancara kepada beberapa penjual motor dan sepeda listrik. Dari hasil wawancara tersebut, didapatkan empat kriteria yang dijadikan pertimbangan pemilihan kendaraan listrik roda dua. Kriteria-kriteria tersebut antara lain harga, *power*, kecepatan maksimum, dan jarak tempuh.

Tabel 4.1. Daftar kriteria beserta atributnya

Kriteria (C)	Keterangan	Atribut
C1	Harga	<i>Cost</i>
C2	<i>Power</i>	<i>Benefit</i>
C3	Kecepatan maksimum	<i>Benefit</i>
C4	Jarak tempuh	<i>Benefit</i>

Penentuan atribut *cost* atau *benefit* pada Tabel 4.1 menggunakan asumsi masyarakat secara umum, dimana *cost* semakin kecil nilainya semakin bagus, sedangkan *benefit* artinya semakin besar nilainya semakin bagus. Oleh karena itu, karena kriteria harga semakin murah semakin bagus, maka termasuk atribut *cost*, sedangkan untuk kriteria *power*, kecepatan maksimum, dan jarak tempuh termasuk ke dalam atribut *benefit*.

##### 2. Menentukan Bobot Tiap Kriteria ( $w_j$ )

Dalam menentukan bobot dari setiap kriteria, digunakan subkriteria yang terbagi ke dalam skala 1 sampai dengan 4. Daftar subkriteria tersebut dapat dilihat pada Tabel 4.2.



Tabel 4.2. Daftar subkriteria beserta bobotnya

Kriteria	Subkriteria	Bobot
C1	Murah	4
	Sedang	3
	Cukup Mahal	2
	Mahal	1
C2	Kecil	1
	Sedang	2
	Besar	3
	Sangat Besar	4
C3	Lambat	1
	Biasa	2
	Cukup Cepat	3
	Cepat	4
C4	Rendah	1
	Sedang	2
	Cukup Tinggi	3
	Tinggi	4

Proses pembobotan ini dilakukan oleh pengguna sistem secara langsung, sehingga bobot tiap kriteria yang dihasilkan bersifat dinamis sesuai kebutuhan pengguna. Dengan kata lain setiap pengguna memiliki prioritas yang berbeda terhadap kriteria dalam memilih kendaraan listrik roda dua, sehingga nilai yang dihasilkan pun berbeda dari setiap pengguna. Tabel 4.3 menunjukkan skenario pemberian bobot yang digunakan dalam perhitungan manual ini.

Tabel 4.3. Contoh nilai bobot

Kriteria (C)	Keterangan	Subkriteria	Bobot	KB
C1	Harga	Murah	4	0,33
C2	Power	Sedang	2	0,17
C3	Kecepatan maksimum	Cukup cepat	3	0,25
C4	Jarak tempuh	Cukup tinggi	3	0,25
<b>Total</b>			12	1,00

Skenario bobot yang digunakan perlu dikonversi agar total bobot menjadi

sama dengan satu ( $\Sigma w = 1$ ) seperti pada nilai KB (Konversi Bobot) pada Tabel 4.3. Hal ini dilakukan agar nilai bobot pada masing-masing kriteria tidak memiliki kesenjangan bobot yang terlalu tinggi. Perhitungan dari nilai KB dilakukan dengan membagi bobot tiap kriteria dengan total bobot dari semua kriteria. Nilai tersebut kemudian dijadikan nilai  $w_j$  untuk digunakan pada perhitungan selanjutnya.

### 3. Menentukan Nilai *Rating* Kecocokan

Pada tahap ini dilakukan proses pengumpulan data kendaraan listrik roda dua. Sampel yang digunakan pada perhitungan manual ini menggunakan data kendaraan sepeda listrik. Karena seluruh data yang terdapat pada Tabel 4.4 sudah berupa angka, maka tidak perlu adanya perubahan nilai dan dapat digunakan sebagai nilai *rating* kecocokan.

Tabel 4.4. Sampel data sepeda listrik pada situs Torselis

Alternatif	Nama produk	C1 (Rp)	C2 (watt)	C3 (km/h)	C4 (km)
A1	Enine SE-8S	2.999.000	500	35	40
A2	Uwinfly D8P	5.500.000	800	55	40
A3	Enine NE-2	5.699.000	500	55	65
A4	Ofero Stareer Pro	6.999.000	600	46	85
A5	Ofero Stareer 2 Pro	8.799.000	1200	55	80

### 4. Membuat Matriks Keputusan

Kemudian pada tahap ini, nilai yang terdapat pada Tabel 4.4 dibuat ke dalam bentuk matriks pada Persamaan 4.1 sebagai berikut.

$$x_{ij} = \begin{bmatrix} 2999000 & 500 & 35 & 40 \\ 5500000 & 800 & 55 & 40 \\ 5699000 & 500 & 55 & 65 \\ 6999000 & 600 & 46 & 85 \\ 8799000 & 1200 & 55 & 80 \end{bmatrix} \quad (4.1)$$

### 5. Melakukan Normalisasi Matriks ( $r_{ij}$ )

Sebelum melakukan proses normalisasi matriks, atribut dari setiap kriteria, yaitu *cost* atau *benefit*, harus ditentukan terlebih dahulu. Daftar atribut

tersebut dapat dilihat pada Tabel 4.1. Berikut merupakan perhitungan nilai normalisasi dengan menggunakan Rumus 2.1 yang terbagi ke dalam kriteria.

(a) Kriteria Harga (C1)

$$\min\{x_{i1}\} = \min\{2999000, 5500000, 5699000, 6999000, 8799000\}$$

$$\min\{x_{i1}\} = 2999000$$

$$r_{11} = \frac{2999000}{2999000} = 1$$

$$r_{21} = \frac{2999000}{5500000} = 0,5452727273$$

$$r_{31} = \frac{2999000}{5699000} = 0,5262326724$$

$$r_{41} = \frac{2999000}{6999000} = 0,4284897843$$

$$r_{51} = \frac{2999000}{8799000} = 0,3408341857$$

(b) Kriteria Power (C2)

$$\max\{x_{i2}\} = \max\{500, 800, 500, 600, 1200\} = 1200$$

$$r_{12} = \frac{500}{1200} = 0,4166666667$$

$$r_{22} = \frac{800}{1200} = 0,6666666667$$

$$r_{32} = \frac{500}{1200} = 0,4166666667$$

$$r_{42} = \frac{600}{1200} = 0,5$$

$$r_{52} = \frac{1200}{1200} = 1$$

(c) Kriteria Kecepatan Maksimum (C3)

$$\max\{x_{i3}\} = \max\{35, 55, 55, 46, 55\} = 55$$

$$r_{13} = \frac{35}{55} = 0,6363636364$$

$$r_{23} = \frac{55}{55} = 1$$

$$r_{33} = \frac{55}{55} = 1$$

$$r_{43} = \frac{46}{55} = 0,8363636364$$

$$r_{53} = \frac{55}{55} = 1$$

(d) Kriteria Jarak Tempuh (C4)

$$\max\{x_{i4}\} = \max\{40, 40, 65, 85, 80\} = 85$$

$$r_{14} = \frac{40}{85} = 0,4705882353$$

$$r_{24} = \frac{40}{85} = 0,4705882353$$

$$r_{34} = \frac{65}{85} = 0,7647058824$$

$$r_{44} = \frac{85}{85} = 1$$

$$r_{54} = \frac{80}{85} = 0,9411764706$$

Setelah melakukan perhitungan nilai normalisasi dari masing-masing alternatif pada setiap kriteria, didapatkan matriks normalisasi ( $r_{ij}$ ) seperti pada Persamaan 4.2 berikut.

$$r_{ij} = \begin{bmatrix} 1 & 0,417 & 0,636 & 0,471 \\ 0,545 & 0,667 & 1 & 0,471 \\ 0,526 & 0,417 & 1 & 0,765 \\ 0,428 & 0,5 & 0,836 & 1 \\ 0,341 & 1 & 1 & 0,941 \end{bmatrix} \quad (4.2)$$

## 6. Menentukan *Ranking*

Pada tahap yang terakhir ini, dilakukan proses penjumlahan dari hasil perkalian matriks pada Persamaan 4.2 dengan bobot masing-masing kriteria ( $w_j$ ) yang terdapat pada kolom KB di Tabel 4.3. Perhitungan tersebut dilakukan berdasarkan Rumus 2.2 yang kemudian diberikan peringkat dari nilai tertinggi. Alternatif dengan peringkat pertama merupakan rekomendasi terbaik dalam menentukan pilihan. Berikut perhitungan peringkat dari setiap alternatif.

$$V_1 = (0,33 \times 1) + (0,17 \times 0,417) + (0,25 \times 0,636) + (0,25 \times 0,471) \\ = 0,333 + 0,069 + 0,159 + 0,118 = 0,6795157457$$

$$V_2 = (0,33 \times 0,545) + (0,17 \times 0,667) + (0,25 \times 1) + (0,25 \times 0,471) \\ = 0,182 + 0,111 + 0,250 + 0,118 = 0,6605157457$$

$$V_3 = (0,33 \times 0,526) + (0,17 \times 0,417) + (0,25 \times 1) + (0,25 \times 0,765) \\ = 0,175 + 0,069 + 0,250 + 0,191 = 0,6860318058$$

$$V_4 = (0,33 \times 0,428) + (0,17 \times 0,5) + (0,25 \times 0,836) + (0,25 \times 1) \\ = 0,143 + 0,083 + 0,209 + 0,250 = 0,6852541705$$

$$V_5 = (0,33 \times 0,341) + (0,17 \times 1) + (0,25 \times 1) + (0,25 \times 0,941) \\ = 0,114 + 0,167 + 0,250 + 0,235 = 0,7655721795$$

Hasil dari perhitungan sebelumnya dapat dirangkum ke dalam bentuk matriks pada Persamaan 4.3 berikut ini.

$$V_{ij} = \begin{bmatrix} 0,333 & 0,069 & 0,159 & 0,118 \\ 0,182 & 0,111 & 0,250 & 0,118 \\ 0,175 & 0,069 & 0,250 & 0,191 \\ 0,143 & 0,083 & 0,209 & 0,250 \\ 0,114 & 0,167 & 0,250 & 0,235 \end{bmatrix} \quad V_i = \begin{bmatrix} 0,680 \\ 0,661 \\ 0,686 \\ 0,685 \\ 0,766 \end{bmatrix} \quad (4.3)$$

Selanjutnya nilai  $V_i$  dari setiap alternatif diberi peringkat seperti pada Tabel 4.5. Hasil menunjukkan bahwa A5 memiliki nilai tertinggi diantara alternatif lainnya, yaitu 0,766. Dari penentuan peringkat pada alternatif sepeda listrik, Ofero Stareer 2 Pro lah yang menjadi alternatif terbaik yang sesuai dengan skenario bobot kriteria.

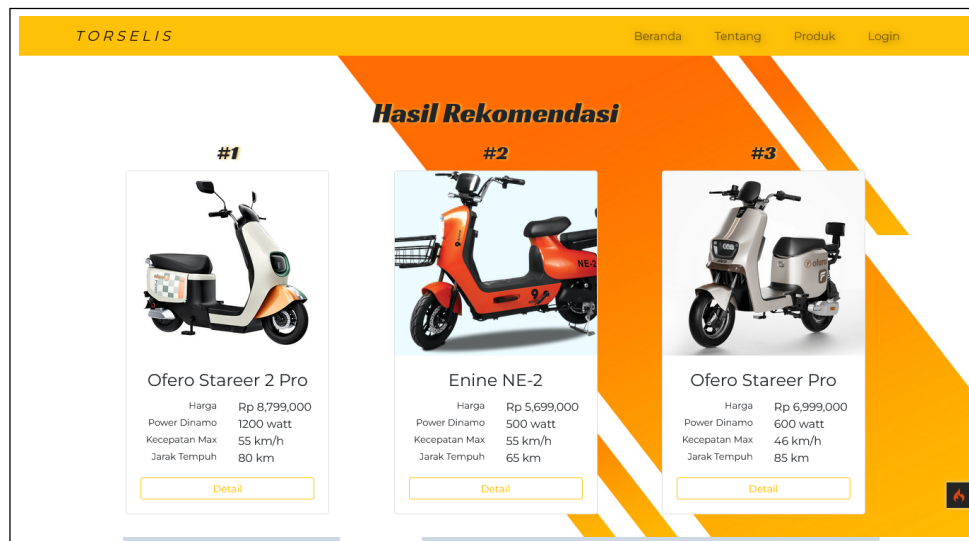
Tabel 4.5. Hasil peringkat dari sampel data sepeda listrik pada situs Torselis

Alternatif	Nama produk	$V_i$	Rank
A5	Ofero Stareer 2 Pro	0,766	1
A3	Enine NE-2	0,686	2
A4	Ofero Stareer Pro	0,685	3
A1	Enine SE-8S	0,680	4
A2	Uwinfly D8P	0,661	5

The screenshot shows the Torselis website interface. At the top, there is a navigation bar with the logo 'TORSSELIS' and links for 'Beranda', 'Tentang', 'Produk', and 'Login'. The main content area is titled '1. Pilih Jenis Kendaraan' and features two radio buttons: 'Motor Listrik' (unselected) and 'Sepeda Listrik' (selected). Below this, section '2. Tentukan Pengaturan Tiap Kriteria' contains four dropdown menus: 'Harga' (Murah), 'Power' (Sedang), 'Kecepatan Max' (Cukup Cepat), and 'Jarak Tempuh' (Cukup Tinggi). At the bottom, section '3. Klik Tombol' includes a yellow button labeled 'Dapatkan Hasil'.

Gambar 4.12. Antarmuka saat pemberian bobot pada situs Torselis

Gambar 4.13 merupakan antarmuka dari hasil yang diberikan oleh sistem. Hasil tersebut didapat dari pemberian bobot yang sama dengan perhitungan manual seperti yang terlihat pada Gambar 4.12. Kedua hasil, yaitu dari perhitungan manual dan dari *output* sistem, memiliki peringkat dari pilihan alternatif yang sama. Ofero Stareer 2 Pro sebagai peringkat 1, Enine NE-2 sebagai peringkat 2, dan Ofero Stareer Pro sebagai peringkat 3. Sehingga dari uji skenario di atas, sistem telah memberikan hasil yang tepat.



Gambar 4.13. Antarmuka hasil rekomendasi sepeda listrik pada situs Torselis

#### 4.5 Hasil Evaluasi Sistem

Evaluasi pada sistem dilakukan dengan menyebarkan *link website* untuk diuji beserta dengan *link Google Form* yang berisi kuesioner untuk diisi oleh responden. Target responden tersebut merupakan pengguna yang tergabung dalam komunitas motor atau sepeda listrik maupun pengguna yang memiliki ketertarikan terhadap otomotif dan kendaraan listrik. Kuesioner terdiri dari sepuluh pertanyaan dengan landasan EUCS (*End User Computing Satisfaction*) yang terdapat pada Gambar 2.2. Kemudian untuk menilai setiap pertanyaan tersebut, digunakan skala Likert yang terdapat pada Tabel 2.1.

Terdapat lima dimensi yang dibahas berdasarkan pertanyaan yang telah dibuat, masing-masing dimensi memiliki dua pertanyaan. Dimensi tersebut diantaranya adalah dimensi *content*, *accuracy*, *format*, *timeliness*, dan *ease of use*. Jumlah responden yang didapat selama proses evaluasi yaitu 71 orang. Berikut merupakan hasil rekapitulasi kuesioner yang telah diisi oleh pengguna yang dapat dilihat pada Tabel 4.6.

Tabel 4.6. Tabel rekapitulasi kuesioner

No. Pertanyaan	Dimensi	STS	TS	N	S	SS
1	<i>Content</i>	2	1	14	23	31
2	<i>Content</i>	0	2	14	37	18
3	<i>Accuracy</i>	0	3	12	36	20
4	<i>Accuracy</i>	1	3	18	29	20
5	<i>Format</i>	0	1	11	33	26
6	<i>Format</i>	0	1	9	46	15
7	<i>Timeliness</i>	0	3	7	24	37
8	<i>Timeliness</i>	0	3	13	33	22
9	<i>Ease of Use</i>	0	4	7	32	28
10	<i>Ease of Use</i>	0	2	7	31	31

#### 4.5.1 Dimensi Isi (Content)

Pada dimensi *content*, terdapat dua pertanyaan yang diberikan kepada pengguna. Pertanyaan tersebut diantaranya adalah:

1. Apakah konten informasi pada situs sudah memudahkan Anda dalam menentukan pilihan kendaraan listrik roda dua?
2. Apakah situs telah memberikan informasi data yang cukup dan tepat mengenai motor dan sepeda listrik?

Tabel 4.7. Tabel perhitungan hasil kuesioner pada dimensi isi (*content*)

No. Pertanyaan	Dimensi	STS	TS	N	S	SS	TS	SR	I
1	<i>Content</i>	2	1	14	23	31	307	4,3239	SP
2	<i>Content</i>	0	2	14	37	18	298	4,1972	P

Berdasarkan jawaban responden yang telah diolah pada Tabel 4.7, dilakukan perhitungan pada dimensi *content* menggunakan Rumus 2.4 untuk mendapatkan nilai TS dan SR. Hasil tersebut kemudian dicocokkan dengan interpretasi berdasarkan interval yang terdapat pada Tabel 2.2.

$$SR_1 = \frac{(31 \times 5) + (23 \times 4) + (14 \times 3) + (1 \times 2) + (2 \times 1)}{71} = \frac{307}{71} = 4,323943662$$

$$SR_2 = \frac{(18 \times 5) + (37 \times 4) + (14 \times 3) + (2 \times 2) + (0 \times 1)}{71} = \frac{298}{71} = 4,197183099$$

Hasil perhitungan nilai SR pada dimensi *content* yang pertama ditemukan memiliki tingkat kepuasan pengguna dengan interpretasi "Sangat Puas", sedangkan untuk pertanyaan nomor dua memiliki tingkat kepuasan pengguna dengan interpretasi "Puas".

#### 4.5.2 Dimensi Keakuratan (Accuracy)

Pada dimensi *accuracy*, terdapat dua pertanyaan yang diberikan kepada pengguna. Pertanyaan tersebut diantaranya adalah:

1. Apakah situs telah menyediakan hasil rekomendasi yang akurat sesuai dengan tingkatan kriteria yang Anda pilih?
2. Apakah Anda puas dengan keakuratan hasil rekomendasi yang diberikan oleh situs?

Tabel 4.8. Tabel perhitungan hasil kuesioner pada dimensi keakuratan (*accuracy*)

No. Pertanyaan	Dimensi	STS	TS	N	S	SS	TS	SR	I
3	<i>Accuracy</i>	0	3	12	36	20	298	4,1972	P
4	<i>Accuracy</i>	1	3	18	29	20	295	4,1549	P

Berdasarkan jawaban responden yang telah diolah pada Tabel 4.8, dilakukan perhitungan pada dimensi *accuracy* menggunakan Rumus 2.4 untuk mendapatkan nilai TS dan SR. Hasil tersebut kemudian dicocokkan dengan interpretasi berdasarkan interval yang terdapat pada Tabel 2.2.

$$SR_3 = \frac{(20 \times 5) + (36 \times 4) + (12 \times 3) + (3 \times 2) + (0 \times 1)}{71} = \frac{298}{71} = 4,197183099$$

$$SR_4 = \frac{(20 \times 5) + (29 \times 4) + (18 \times 3) + (3 \times 2) + (1 \times 1)}{71} = \frac{295}{71} = 4,154929577$$

Hasil perhitungan nilai SR pada dimensi *accuracy* yang pertama ditemukan memiliki tingkat kepuasan pengguna dengan interpretasi "Puas", dan untuk pertanyaan nomor empat memiliki tingkat kepuasan pengguna dengan interpretasi yang sama, yaitu "Puas".



### 4.5.3 Dimensi Bentuk (Format)

Pada dimensi *format*, terdapat dua pertanyaan yang diberikan kepada pengguna. Pertanyaan tersebut diantaranya adalah:

1. Apakah menurut Anda proses dari penentuan tingkatan kriteria hingga pemberian hasil rekomendasi yang disajikan oleh situs memiliki bentuk yang menarik?
2. Apakah tampilan dari informasi yang diberikan oleh situs sudah jelas dan memiliki nilai estetika?

Tabel 4.9. Tabel perhitungan hasil kuesioner pada dimensi bentuk (*format*)

No. Pertanyaan	Dimensi	STS	TS	N	S	SS	TS	SR	I
5	<i>Format</i>	0	1	11	33	26	308	4,3380	SP
6	<i>Format</i>	0	1	9	46	15	297	4,1831	P

Berdasarkan jawaban responden yang telah diolah pada Tabel 4.9, dilakukan perhitungan pada dimensi *format* menggunakan Rumus 2.4 untuk mendapatkan nilai TS dan SR. Hasil tersebut kemudian dicocokkan dengan interpretasi berdasarkan interval yang terdapat pada Tabel 2.2.

$$SR_5 = \frac{(26 \times 5) + (33 \times 4) + (11 \times 3) + (1 \times 2) + (0 \times 1)}{71} = \frac{308}{71} = 4,338028169$$

$$SR_6 = \frac{(15 \times 5) + (46 \times 4) + (9 \times 3) + (1 \times 2) + (0 \times 1)}{71} = \frac{297}{71} = 4,183098592$$

Hasil perhitungan nilai SR pada dimensi *format* yang pertama ditemukan memiliki tingkat kepuasan pengguna dengan interpretasi "Sangat Puas", sedangkan untuk pertanyaan nomor enam memiliki tingkat kepuasan pengguna dengan interpretasi "Puas".

### 4.5.4 Dimensi Ketepatan Waktu (Timeliness)

Pada dimensi *timeliness*, terdapat dua pertanyaan yang diberikan kepada pengguna. Pertanyaan tersebut diantaranya adalah:

1. Apakah Anda mendapatkan hasil rekomendasi dalam waktu yang cepat?

2. Apakah situs menyediakan informasi yang *up-to-date*?

Tabel 4.10. Tabel perhitungan hasil kuesioner pada dimensi ketepatan waktu (*timeliness*)

No. Pertanyaan	Dimensi	STS	TS	N	S	SS	TS	SR	I
7	<i>Timeliness</i>	0	3	7	24	37	315	4,4366	SP
8	<i>Timeliness</i>	0	3	13	33	22	300	4,2254	SP

Berdasarkan jawaban responden yang telah diolah pada Tabel 4.10, dilakukan perhitungan pada dimensi *timeliness* menggunakan Rumus 2.4 untuk mendapatkan nilai TS dan SR. Hasil tersebut kemudian dicocokkan dengan interpretasi berdasarkan interval yang terdapat pada Tabel 2.2.

$$SR_7 = \frac{(37 \times 5) + (24 \times 4) + (7 \times 3) + (3 \times 2) + (0 \times 1)}{71} = \frac{315}{71} = 4,436619718$$

$$SR_8 = \frac{(22 \times 5) + (33 \times 4) + (13 \times 3) + (3 \times 2) + (0 \times 1)}{71} = \frac{300}{71} = 4,225352113$$

Hasil perhitungan nilai SR pada dimensi *timeliness* yang pertama ditemukan memiliki tingkat kepuasan pengguna dengan interpretasi "Sangat Puas", dan untuk pertanyaan nomor delapan memiliki tingkat kepuasan pengguna dengan interpretasi yang sama, yaitu "Sangat Puas".

#### 4.5.5 Dimensi Kemudahan (Ease of Use)

Pada dimensi *ease of use*, terdapat dua pertanyaan yang diberikan kepada pengguna. Pertanyaan tersebut diantaranya adalah:

1. Apakah proses saat memasukkan data atau mencari informasi pada situs sudah ramah pengguna (*user friendly*)?
2. Apakah situs Torselis mudah digunakan?

Tabel 4.11. Tabel perhitungan hasil kuesioner pada dimensi kemudahan (*ease of use*)

No. Pertanyaan	Dimensi	STS	TS	N	S	SS	TS	SR	I
9	<i>Ease of Use</i>	0	4	7	32	28	304	4,2817	SP
10	<i>Ease of Use</i>	0	2	7	31	31	311	4,3803	SP

Berdasarkan jawaban responden yang telah diolah pada Tabel 4.11, dilakukan perhitungan pada dimensi *ease of use* menggunakan Rumus 2.4 untuk mendapatkan nilai TS dan SR. Hasil tersebut kemudian dicocokkan dengan interpretasi berdasarkan interval yang terdapat pada Tabel 2.2.

$$SR_9 = \frac{(28 \times 5) + (32 \times 4) + (7 \times 3) + (4 \times 2) + (0 \times 1)}{71} = \frac{304}{71} = 4,281690141$$

$$SR_{10} = \frac{(31 \times 5) + (31 \times 4) + (7 \times 3) + (2 \times 2) + (0 \times 1)}{71} = \frac{311}{71} = 4,38028169$$

Hasil perhitungan nilai SR pada dimensi *ease of use* yang pertama ditemukan memiliki tingkat kepuasan pengguna dengan interpretasi "Sangat Puas", dan untuk pertanyaan nomor sepuluh memiliki tingkat kepuasan pengguna dengan interpretasi yang sama, yaitu "Sangat Puas".

#### 4.5.6 Hasil Akhir Perhitungan tiap Dimensi

Setelah perhitungan nilai SR dilakukan pada setiap pertanyaan, maka dilakukan perhitungan skor akhir atau yang disebut nilai SA. Pada Tabel 4.12 telah terangkum nilai SR dari seluruh pertanyaan.

Tabel 4.12. Tabel nilai SR dari seluruh pertanyaan

No. Pertanyaan	Dimensi	SR	I
1	<i>Content</i>	4,3239	SP
2	<i>Content</i>	4,1972	P
3	<i>Accuracy</i>	4,1972	P
4	<i>Accuracy</i>	4,1549	P
5	<i>Format</i>	4,3380	SP
6	<i>Format</i>	4,1831	P
7	<i>Timeliness</i>	4,4366	SP
8	<i>Timeliness</i>	4,2254	SP
9	<i>Ease of Use</i>	4,2817	SP
10	<i>Ease of Use</i>	4,3803	SP

Berdasarkan Tabel 4.12, ditemukan bahwa nilai SR pada pertanyaan nomor tujuh dengan dimensi *timeliness* memiliki nilai tertinggi, yaitu 4,4366 dengan interpretasi "Sangat Puas". Adapun nilai paling rendah yaitu sebesar 4,1549,

ditemukan pada pertanyaan nomor empat dengan dimensi *accuracy*. Meskipun demikian, interval dari nilai SR tersebut masih masuk ke dalam interpretasi "Puas". Kedua hal tersebut menandakan bahwa sistem telah memberikan hasil rekomendasi dalam waktu yang cepat, tetapi keakuratan dari hasil tersebut masih belum sangat memuaskan.

Untuk mendapatkan nilai tingkat kepuasan pengguna secara keseluruhan, maka perlu dihitung skor akhir (SA) menggunakan Rumus 2.5. Perhitungan nilai SA tersebut adalah sebagai berikut.

$$SA = \frac{4.3239 + 4.1972 + 4.1972 + 4.1549 + 4.3380}{10} + \frac{4.1831 + 4.4366 + 4.2254 + 4.2817 + 4.3803}{10} = 4,2718$$

Setelah perhitungan skor akhir dilakukan, ditemukan bahwa secara keseluruhan sistem memiliki tingkat kepuasan pengguna sebesar 4,2718 yaitu dengan interpretasi "Sangat Puas". Sehingga dapat disimpulkan bahwa sistem pendukung keputusan yang dikembangkan telah memenuhi kebutuhan pengguna.

