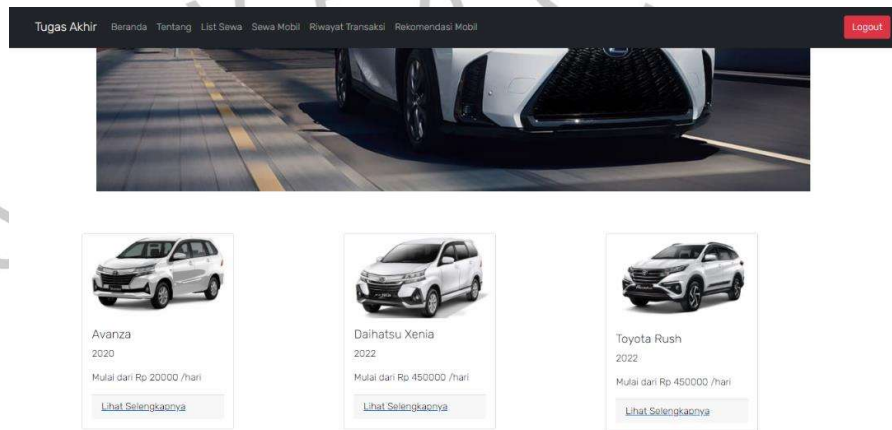


BAB V

HASIL DAN PEMBAHASAN

5.1 Hasil dan Uraian Perancangan Tampilan

Perancangan aplikasi sewa mobil berbasis web untuk pengguna telah dirancang dan berikut hasil perancangannya..



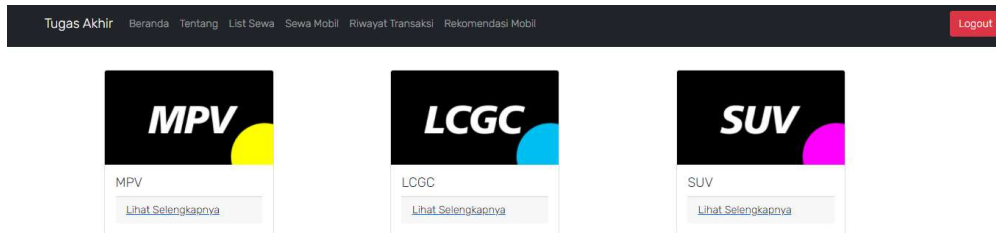
Gambar 5. 1 Tampilan Beranda

Halaman beranda menampilkan *slide* gambar dan rangkuman daftar kendaraan beserta foto mobil harga sewa dan tombol untuk melihat detail mobil yang dipilih. Pada halaman beranda juga terdapat *navigation* menu yang dimana setiap menu tersebut mengarah ke halaman sesuai dengan yang dipilih oleh *user*.



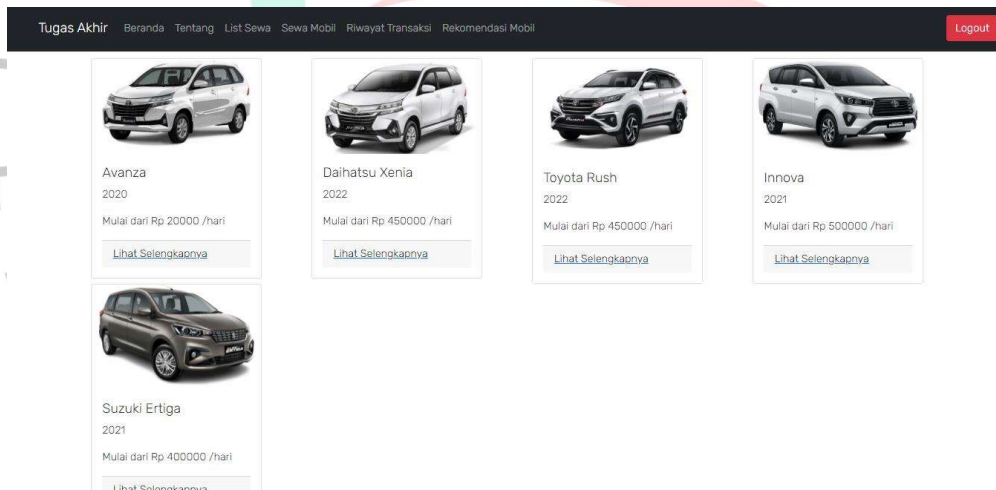
Gambar 5. 2 Tampilan List Sewa

Pada tampilan list sewa berisi data dari transaksi sewa yang dilakukan *user* dengan menampilkan gambar mobil, mobil, harga sewa, durasi sewa, tanggal sewa, total harga sewa, tombol hapus transaksi dan tombol untuk melakukan proses *Checkout*.



Gambar 5. 3 Tampilan Sewa Mobil

Pada tampilan sewa mobil berisi daftar kategori yang dapat dipilih oleh penyewa, halaman sewa mobil menampilkan informasi berupa nama kategori, gambar dan tombol untuk melihat detail dari kategori



Gambar 5. 4 Tampilan List Mobil

Pada tampilan *list* mobil berisi daftar mobil yang disewakan yang dapat dipilih oleh penyewa, halaman *list* sewa mobil menampilkan informasi berupa nama mobil, gambar, harga sewa dan tahun mobil dan tombol untuk melihat detail dari mobil.

Tugas Akhir Beranda Tentang List Sewa Sewa Mobil Riwayat Transaksi Rekomendasi Mobil Logout

Data Penyewa

Nama Depan

Nama Belakang

Email Address

Nomor Telepon

Alamat

Provinsi

Kota

[Place Order](#)

Mobil	Harga	Durasi	Total
Avanza	20000	1 hari	20000
Total Keseluruhan			20000

Gambar 5. 5 Tampilan *Checkout Detail*

Halaman *Checkout detail* berisi form yang akan diisi oleh penyewa mulai dari nama depan, nama belakang, *email address*, nomor telepon, alamat, provinsi dan kota. Selain itu juga terdapat tabel yang berisi data dari mobil yaitu nama mobil, harga, durasi, dan total harga.

Rental Mobil Rekomendasi Mobil List Sewa Sewa Mobil Riwayat Transaksi Tentang Logout

Rekomendasi Mobil

Harga Sewa :

300.000 - 400.000

Pilih

300.000 - 400.000

400.000 - 500.000

Fitur :

Select

AC, AUDIO

AC, AUDIO, HEAD UNIT, USB PORT

AC, AUDIO, HEAD UNIT, USB PORT, DASH CAM, REAR CAM

Konsumsi BBM :

14-15 km/liter

Pilih

14-15 km/liter

16-17 km/liter


Kapasitas :

4

Pilih

4

7



Agya

300000 / hari

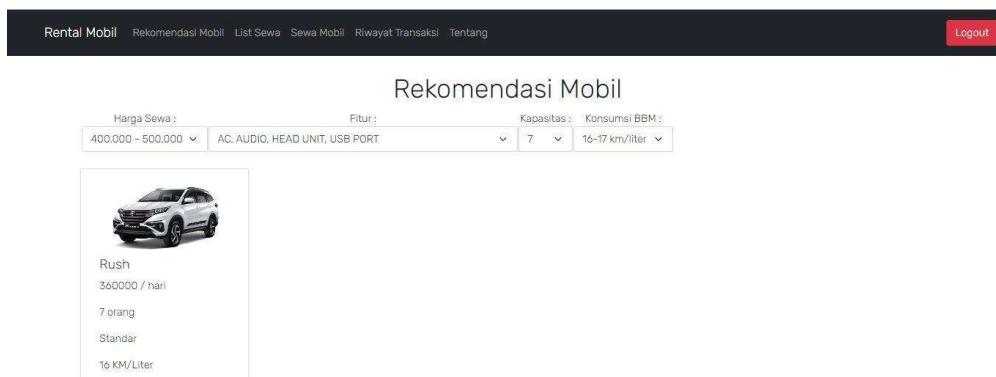
4 orang

Rendah

14 KM/Liter

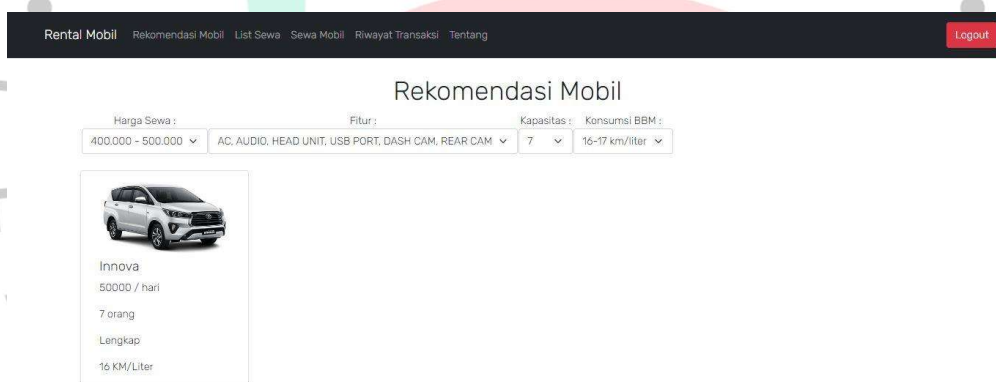
Gambar 5. 6 Tampilan Rekomendasi Mobil

Tampilan rekomendasi mobil merupakan tampilan dari halaman rekomendasi mobil yang menggunakan perhitungan *Analytical Hierarchy Process* atau *AHP*. Perhitungan tersebut melalui tahap *input* harga sewa, fitur, kapasitas dan konsumsi bahan bakar. Dari perhitungan tersebut didapatkan hasil urutan dari nilai terbaik yang akan muncul pada bagian *Card*.



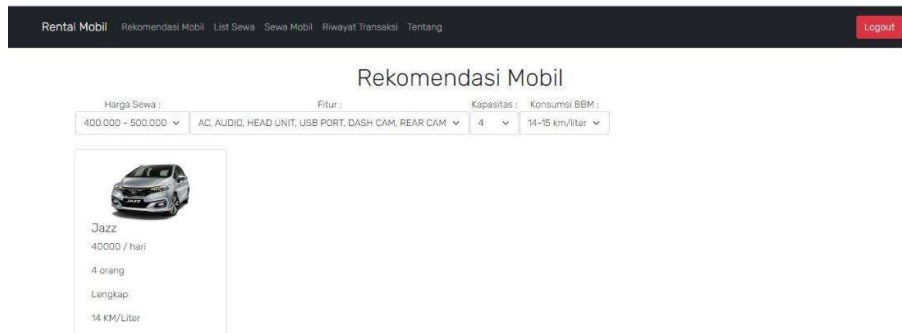
Gambar 5. 7 Rekomendasi Mobil 1

Pada gambar 5.7 di atas merupakan rekomendasi mobil yang muncul pada saat pengguna memilih kriteria harga 400.000 – 500.000, fitur *AC, AUDIO, HEAD UNIT, USB PORT*, kapasitas 7 orang dan konsumsi bahan bakar 16-17 km/liter.



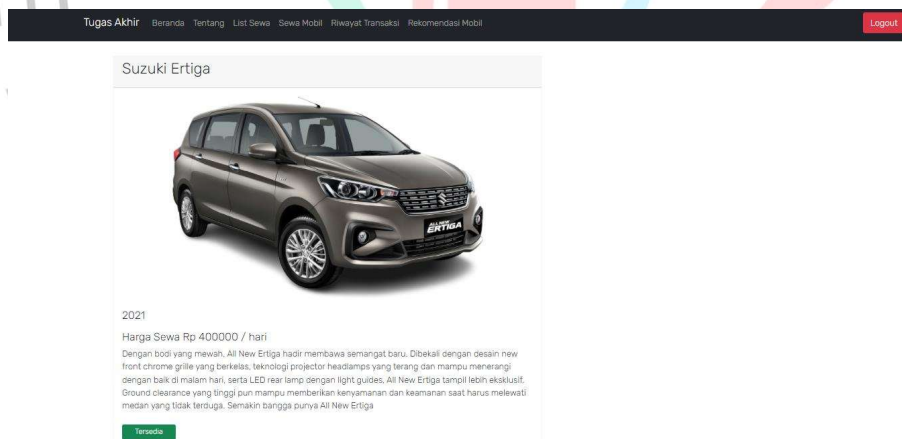
Gambar 5. 8 Rekomendasi Mobil 2

Pada gambar 5.8 di atas merupakan rekomendasi mobil yang muncul pada saat pengguna memilih kriteria harga 400.000 – 500.000, fitur *AC, AUDIO, HEAD UNIT, USB PORT, DASH CAM, REAR CAM* kapasitas 7 orang dan konsumsi bahan bakar 16-17 km/liter.



Gambar 5. 9 Rekomendasi Mobil 3

Pada gambar 5.9 di atas merupakan rekomendasi mobil yang muncul pada saat pengguna memilih kriteria harga 400.000 – 500.000, fitur *AC, AUDIO, HEAD UNIT, USB PORT, DASH CAM, REAR CAM* kapasitas 4 orang dan konsumsi bahan bakar 14 km/liter.

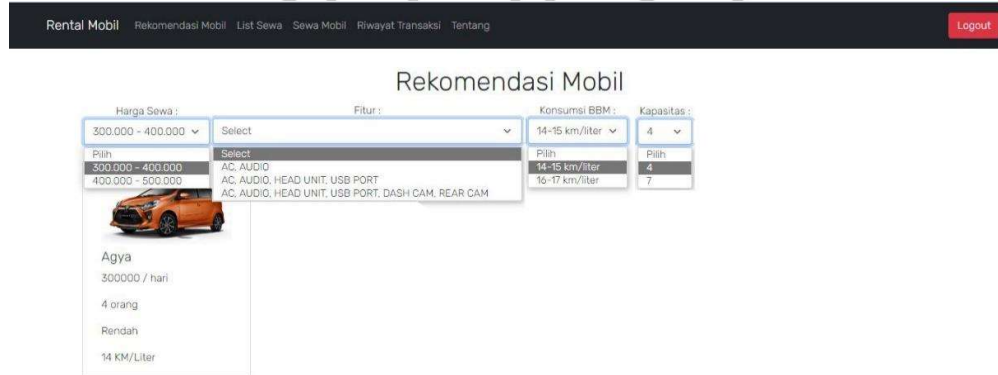


Gambar 5. 10 Tampilan Detail Mobil

Halaman detail mobil berisi informasi detail dari mobil seperti nama mobil, tahun mobil, deskripsi dari mobil, harga sewa, ketersediaan unit, *field* untuk mengisi tanggal sewa dan tombol sewa sekarang yang berfungsi untuk melakukan proses sewa mobil yang dipilih.

5.2 Hasil Perhitungan Perhitungan Algoritma *Analytic Hierarchy Process (AHP)*

Algoritma *AHP* telah diterapkan pada aplikasi sewa mobil berbasis *web* untuk rekomendasi mobil yang sesuai dengan yang dipilih oleh penyewa. Perhitungan algoritma *AHP* menggunakan rumus yang telah ditentukan dengan menggunakan lima buah data untuk perhitungan algoritma.



Gambar 5. 7 Kriteria Perhitungan *AHP*

Tabel 5.1 Data Mobil

Mobil	Fitur	Harga	Kapasitas	Konsumsi Bahan Bakar
Avanza	AC, AUDIO, HEAD UNIT, USB PORT	350.000	7	14 liter/km
Xenia	AC, AUDIO	330.000	7	16 liter/km
Rush	AC, AUDIO, HEAD UNIT, USB PORT, DASH CAM, REAR CAM	430.000	7	14 liter/km
Ertiga	AC, AUDIO, HEAD UNIT, USB PORT	420.000	7	14 liter/km
Innova	S	500.000	7	16 liter/km

Beberapa data pada tabel 5.1 akan di tentukan relasi antar yang digunakan, ada 4 yang digunakan dalam kasus tersebut, yaitu harga, kapasitas pengunjung, fasilitas dan luas ruangan. Dapat diasumsikan dari kasus tersebut bahwa harga 1.6 kali lebih penting daripada fasilitas, fasilitas 2.4 kali lebih penting daripada

kapasitas pengunjung, kapasitas pengunjung 1.7 kali lebih penting dari pada luas ruangan. Setelah menentukan relasi antar , buat matriks dari relasi antar tersebut.

Pada data pada tabel 5.2 tersebut akan melalui tahap perhitungan untuk menentukan relasi antar yang digunakan. Terdapat 4 yang digunakan yaitu fitur, harga, kapasitas dan konsumsi bahan bakar. Dari data tersebut diamsusikan bahwa harga 5 kali lebih penting dari kapasitas, harga 2 kali lebih penting dari *bbm*, harga 3 kali lebih penting dari fitur, kapasitas 4 kali lebih penting dari *bbm*, kapasitas 7 kali lebih penting dari fitur, dan *bbm* 5 kali lebih penting daripada fitur.

Tabel 5.2 Matriks Relasi

Matriks Relasi				
	Harga	Kapasitas	BBM	Fitur
Harga	1	5	2	3
Kapasitas	0,2	1	4	7
BBM	0,5	0,25	1	5
Fitur	0,33333333	0,142857143	0,2	1

Setelah melalui perhitungan untuk mendapatkan matriks dari perhitungan relasi antar , berlanjut pada tahap normalisasi yang bertujuan untuk mencari rata-rata bobot per. Berikut hasil dari perhitungan normalisasi .

Tabel 5.3 Normalisasi Data

Normalisasi Data				
	Harga	Kapasitas	BBM	Fitur
Harga	0,491803279	0,782122905	0,277778	0,1875
Kapasitas	0,098360656	0,156424581	0,555556	0,4375
BBM	0,245901639	0,039106145	0,138889	0,3125
Fitur	0,163934426	0,022346369	0,027778	0,0625

Selanjutnya mencari rata-rata dari setiap , untuk mencari rata-rata menggunakan cara menjumlahkan setiap barisnya lalu dibagi dengan jumlah yaitu 4 . Berikut pada tabel 5.4 hasil dari rata-rata setiap kriteria.

Tabel 5.4 Rata-rata

Rata-rata
0,43480099
0,311960198
0,184099168
0,069139643

Setelah mendapatkan rata-rata setiap kriteria, selanjutnya melakukan normalisasi atau pembulatan dari setiap nilai rata-rata. Hasil dari normalisasi rata-rata tersebut merupakan nilai akhir yang mewakili bobot dari setiap kriteria. Berikut hasil dari normalisasi dari nilai rata-rata pada tabel 5.3.

Tabel 5.5 Bobot

Bobot
0,4
0,3
0,2
0,1

Tahap selanjutnya mencari bobot dari setiap alternatif. Proses pertama yaitu mencari bobot untuk harga dari setiap alternatif. Berikut hasil perhitungan dari pencarian bobot harga yang disajikan pada tabel 5.6.

Tabel 5.6 Alternatif Harga

Alternatif Harga					
Harga	Avanza	Rush	Ertiga	Innova	Xenia
Avanza	1	0,33333333	0,5	0,2	2
Rush	3	1	2	0,5	4
Ertiga	2	0,5	1	0,33333333	3
Innova	5	2	3	1	5
Xenia	0,5	0,25	0,333333	0,2	1
Jumlah	11,5	4,08333333	6,8333333	2,23333333	15

Dari tabel 5.5 di atas didapatkan jumlah dari setiap perbandingan alternatif pada harga. Jumlah nilai perbandingan tersebut digunakan untuk menghitung bobot harga dari setiap alternatif.

Tabel 5.7 Jumlah Nilai Perbandingan Per dan Bobot Harga

Jumlah	Bobot Harga
4,03333333	0,1017234
10,5	0,2648172
6,83333333	0,1723413
16	0,4035309
2,28333333	0,0575872
39,65	1

Pada tabel 5.6 di atas terdapat kolom bobot harga yang didapatkan dari jumlah perbandingan per alternatif dibagi dengan jumlah keseluruhan dari nilai perbandingan setiap alternatif terhadap harga. Contohnya seperti jumlah perbandingan untuk alternatif avanza yaitu 0,1017324 dibagi dengan 39,65 yang merupakan jumlah keseluruhan dari nilai perbandingan setiap alternatif terhadap harga.

Tabel 5.8 Alternatif Kapasitas

Kapasitas	Alternatif Kapasitas				
	Avanza	Rush	Ertiga	Innova	Xenia
Avanza	1	1	1	1	1
Rush	1	1	1	1	1
Ertiga	1	1	1	1	1
Innova	1	1	1	1	1
Xenia	1	1	1	1	1
Jumlah	5	5	5	5	5

Tabel 5.9 Jumlah Nilai Perbandingan Per Alternatif dan Bobot Kapasitas

Jumlah	Bobot Kapasitas
5	0,2
5	0,2
5	0,2
5	0,2
5	0,2
25	1

Pada tabel 5.7 dan 5.8 menggunakan proses perhitungan yang sama dengan tabel 5.5 dan 5.6 yang dimana bertujuan untuk mencari bobot kapasitas dari setiap alternatifnya.

Tabel 5.10 Alternatif BBM

Alternatif BBM					
BBM	Avanza	Rush	Ertiga	Innova	Xenia
Avanza	1	3	0,333333	2	0,5
Rush	0,333333	1	0,2	0,5	0,25
Ertiga	3	5	1	4	2
Innova	2	2	0,25	1	0,33333
Xenia	2	4	0,5	3	1
Jumlah	8,333333	15	2,283333	10,5	4,08333

Tabel 5.11 Jumlah Nilai Perbandingan Per dan Bobot BBM

Jumlah	Bobot BBM
6,83333333	0,1699834
2,28333333	0,0567993
15	0,3731343
5,58333333	0,1388889
10,5	0,261194
40,2	1

Pada tabel 5.9 dan 5.10 menggunakan proses perhitungan yang sama dengan tabel 5.5 dan 5.6 yang dimana bertujuan untuk mencari bobot kapasitas dari setiap alternatifnya.

Tabel 5.12 Alternatif Fitur

Alternatif Fitur					
Fitur	Avanza	Rush	Ertiga	Innova	Xenia
Avanza	1	0,5	2	0,333333	3
Rush	2	1	3	0,5	4
Ertiga	0,5	0,33333333	1	0,25	2
Innova	3	2	4	1	5
Xenia	0,333333	0,25	0,5	0,2	1
	6,833333	4,08333333	10,5	2,283333	15

Tabel 5.13 Jumlah Nilai Perbandingan Bobot Fitur

Jumlah	Bobot Fitur
6,83333333	0,1765719
10,5	0,2713178
4,08333333	0,1055125
15	0,3875969
2,28333333	0,0590009
38,7	1

Tabel 5.9 dan 5.10 menggunakan proses perhitungan yang sama dengan tabel 5.5 dan 5.6 yang dimana bertujuan untuk mencari bobot kapasitas dari setiap alternatifnya. Tahap selanjutnya setelah semua alternatif telah mendapatkan bobot setiap nya maka proses selanjutnya yaitu melakukan perangkungan dengan cara melakukan perkalian matriks antara setiap nilai bobot per alternatif dengan bobot keseluruhan . Berikut proses perkalian matriks tersebut.

Tabel 5.14 Perkalian Matriks Keseluruhan

Perkalian Matriks Keseluruhan				
Avanza	0,101723413	0,2	0,169983416	0,1765719
Rush	0,26481715	0,2	0,056799337	0,2713178
Ertiga	0,17234132	0,2	0,373134328	0,1055125
Innova	0,403530895	0,2	0,138888889	0,3875969
Xenia	0,057587222	0,2	0,26119403	0,0590009

Setiap nilai bobot alternatif pada tabel 5.13 di atas dilakukan perkalian tabel bobot yang telah didapatkan dari proses normalisasi rata-rata pada tahap sebelumnya. Berikut tabel 5.15 yang berisi bobot dari setiap kriteria .

Tabel 5.15 Bobot

Bobot
0,4
0,3
0,2
0,1

Berikut contoh dari perhitungan perkalian matriks keseluruhan. Pada alternatif *avanza* mempunyai nilai bobot harga 0,101723413, bobot kapasitas 0,2, bobot *bbm* 0,169983416 dan bobot fitur 0,1765719. Setiap bobot tersebut dilakukan

perkalian berpasangan dengan setiap bobot , seperti contohnya bobot harga 0,101723413 dikalikan dengan bobot 0,4, tahap perhitungan tersebut berlaku bagi setiap alternatif tersebut dengan konsep baris dikali dengan kolom.

Setelah melalui tahap perkalian matriks keseluruhan didapatkan nilai ranking dari setiap alternatif. Berikut tabel yang berisi nilai akhir dari perhitungan ranking perkalian matriks keseluruhan.

Tabel 5.16 *Ranking*

<i>Ranking</i>	
Avanza	0,152343241
Rush	0,20441851
Ertiga	0,214114643
Innova	0,287949826
Xenia	0,141173781
Jumlah	1

Dari tabel tersebut disimpulkan bahwa *Innova* menempati peringkat pertama, *Ertiga* kedua, *Rush* keempat, *Avanza* kelima dan *Xenia* menempati urutan terakhir. Setelah melakukan perankingan beralih ke tahap mencari lambda (λ) Max. Untuk rumus lambda (λ) Max yaitu (jumlah 1 / rata-rata 1) + (jumlah 2 / rata-rata 2) + (jumlah 3 / rata-rata 3) + (jumlah 4 / rata-rata 4) / 4 (total). Sebagai contoh seperti berikut $(1,739203961/0,43480099) + (1,247840792/0,311960198) + (0,736396673/0,184099168) + (0,276558573/0,069139643) = 4,0000000022999 + 4,0096463022508 + 4,00000000543185 + 4,00000001446348 = 16,00964632 / 4 = 4,002411581$. Dari perhitungan tersebut didapatkan nilai *lambda max* yaitu **4,002411581**. Setelah mendapatkan *lambda max* selanjutnya mencari *CR* atau *Consistency Ratio*, *IR*, dan *Consistency Index*. Berikut perhitungan dari *CR*, *IR* dan *CI*.

$$CR = (\text{Lambda Max} - \text{Jumlah}) : (\text{Jumlah} - 1)$$

$$CR = (4,002411581 - 4) : (4 - 1) = 0,00080386$$

Tabel 5.17 Skala Index

Skala Index										
n	1	2	3	4	5	6	7	8	9	10
IR	0	0	0,58	0,9	1,12	1,2	1,32	1,41	1,45	1,49

IR = n ke 4 = 0,9 (sesuai dengan skala index yang dimana jumlah 4 memiliki IR 0,9)

CI = CR : IR

CI = 0,00080386 : 0,9 = **0,000893**

Dari hasil perhitungan *Consistency Index* didapatkan nilai *CI* sebesar 0,000893 yang diperoleh dari perhitungan nilai *Consistency Ratio* dibagi dengan nilai *Index Ratio*.

• Berikutnya perankingan alternatif dari setiap yaitu harga, kapasitas, fitur dan konsumsi bbm.

Tabel 5.18 Peringkat harga

Harga	
avanza	0,040689365
rush	0,10592686
ertiga	0,068936528
innova	0,161412358
xenia	0,023034889

Tabel 5.19 Peringkat kapasitas

Kapasitas	
avanza	0,06
rush	0,06
ertiga	0,06
innova	0,06
xenia	0,06

Tabel 5.20 Peringkat bbm

BBM	
avanza	0,0339967
rush	0,0113599
ertiga	0,0746269
innova	0,0277778
xenia	0,0522388

Tabel 5.21 Peringkat fitur

Fitur	
avanza	0,0176572
rush	0,0271318
ertiga	0,0105512
innova	0,0387597
xenia	0,0059001

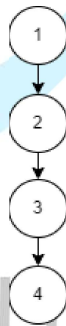
5.3 Hasil Pembahasan

Pengujian terhadap rancangan sistem yang dibangun oleh peneliti harus dilakukan untuk mengetahui apakah fungsionalitas dari aplikasi berjalan sesuai dengan rancangan atau tidak. Hasil dari pengujian menggunakan metode *black box* dan *white box* adalah sebagai berikut :

5.3.1 White Box Testing

Tabel 5.22 Whitebox Testing Algoritma

No	Hasil yang Diharapkan	Source Code	Hasil Pengujian
1	Pada saat user mengakses halaman rekomendasi mobil, user memilih nilai untuk setiap dan user menekan tombol submit	<pre> const [vector, setVector] = useState(getWeights()); (1) const resultRanked = () => { const resultRank = resultMobil(mobil); let list = []; for (let i = 0; i < 8; i++) { list[i] = resultRank[0][i] * vector['ev'][0] + resultRank[1][i] * vector['ev'][1] + resultRank[2][i] * vector['ev'][2] + resultRank[3][i] * vector['ev'][3]; } (2) const handleSubmit = (e) => { e.preventDefault(); setVector(getWeights(matriks)); (3) }; useEffect(() => { setRanked(resultRanked()); console.log(ranked); }, [vector]); (4) </pre>	Algoritma berhasil dijalankan dan rekomendasi mobil muncul



Gambar 5.11 Cyclomatic Complexity Menu

Pada gambar 5.11 merupakan *Cyclomatic Complexity* pada *flowgraph navigation* menu pada aplikasi sewa mobil. Berikut penjelasan dari *Cyclomatic Complexity* di atas.

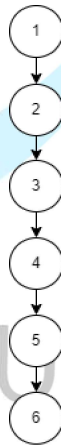
Jalur = 1,2,3,4

$$V(G) = 3 \text{ Edges} - 4 \text{ Node} + 2 = 1$$

Hasil tersebut sesuai dengan jumlah jalur yang dapat digunakan untuk menyelesaikan masalah.

Tabel 5.23 *Whitebox Testing Navigation Bar*

No	Hasil yang Diharapkan	Source Code	Hasil Pengujian
1	Pada saat user mengakses rekomendasi mobil	<code><Nav.Link href="/kriteria">Rekomendasi Mobil</Nav.Link></code> (1)	Berhasil menampilkan halaman rekomendasi mobil
2	Pada saat user mengakses list sewa	<code><Nav.Link href="/detail-sewa">List Sewa</Nav.Link></code> (2)	Berhasil menampilkan halaman list sewa
3	Pada saat user mengakses sewa mobil	<code><Nav.Link href="/sewa-mobil">Sewa Mobil</Nav.Link></code> (3)	Berhasil menampilkan halaman sewa mobil
4	Pada saat user mengakses riwayat transaksi	<code><Nav.Link href="/riwayat-transaksi">Riwayat Transaksi</Nav.Link></code> (4)	Berhasil menampilkan halaman riwayat transaksi
5	Pada saat user mengakses tentang	<code><Nav.Link href="/tentang">Tentang</Nav.Link></code> (5)	Berhasil menampilkan halaman tentang
6	Pada saat user mengakses login/logout	<code>{AuthButtons}</code> (6)	Berhasil menampilkan halaman login/logout



Gambar 5.12 *Cyclomatic Complexity Algoritma*

Pada gambar 5.12 di atas merupakan *Cyclomatic Complexity* pada *flowgraph algoritma* pada aplikasi sewa mobil. Berikut penjelasan dari *Cyclomatic Complexity* di atas.

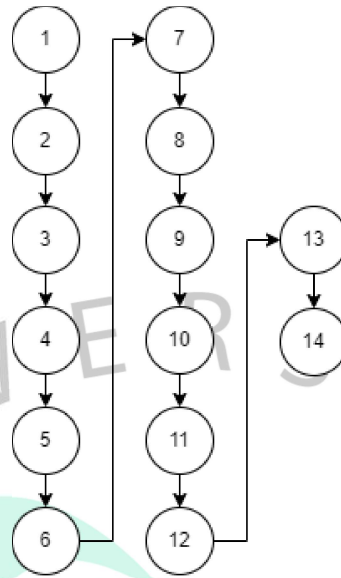
Jalur = 1,2,3,4,5,6

$$V(G) = 5 \text{ Edges} - 6 \text{ Node} + 2 = 1$$

Hasil tersebut sesuai dengan jumlah jalur yang dapat digunakan untuk menyelesaikan masalah.

Tabel 5.24 *Whitebox Testing Detail Mobil*

No	Hasil Yang Diharapkan	Source Code	Hasil Pengujian
1.	Pada saat user mengakses list mobil	<pre>showMobilList = mobil.map((item, idx) => { (1) return (<div className="col" key={idx}> (2) <Container> (3) <Row className="g-4"> (4) <Col> (5) <Card style={{ width: '16rem'}}> (6) <Card.Img variant="top" src={`http://127.0.0.1:8000/\${item.image}`} className="card-img-top embed- responsive-item"/> (7) <Card.Body> (8) <Card.Title>{item.nama}</Card.Title> (9) <Card.Text> (10) {item.tahunmobil} </Card.Text> <Card.Text> (11) Mulai dari Rp {item.hargasewa_asli} /hari </Card.Text> <Card.Footer> (12) <Link to={`/mobil/\${item.category.slug}/\${item.sl ug}`}> (13) Lihat Selengkapnya (14) </Link> </Card.Footer> </Card.Body> </Card></pre>	Berhasil menampilkan halaman list mobil



Gambar 5.13 Cyclomatic Complexity List Mobil

Pada gambar 5.13 di atas merupakan *Cyclomatic Complexity* pada *flowgraph list mobil* pada aplikasi sewa mobil. Berikut penjelasan dari *Cyclomatic Complexity* di atas.

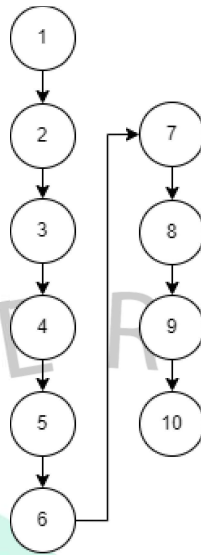
Jalur = 1,2,3,4,5,6,7,8,9,10,11,12,13,14

$$V(G) = 15 \text{ Edges} - 14 \text{ Node} + 2 = 1$$

Hasil tersebut sesuai dengan jumlah jalur yang dapat digunakan untuk menyelesaikan masalah.

Tabel 5.25 Whitebox Testing List Sewa

No	Hasil Yang Diharapkan	Source Code	Hasil Pengujian
1.	Pada saat user mengakses halaman list sewa	<pre> <tr> <td width="10%"> (1) </td> <td> {item.mobil.nama} (2) </td> <td width="15%" className="text-center">Rp {item.mobil.hargasewa_asli}</td> (3) <td width="15%"> <div className="input-group"> <button onClick={() => handleDecrement(item.id)} type="button" className="input-group-text"></button> (4) <div className="form-control text-center">{item.mobil_qty} / Hari</div> (5) <button type="button" onClick={() => handleIncrement(item.id)} className="input- group-text">+</button> (6) </div> </td> <td width="15%" className="text-center">{item.tgl_sewa}</td> (7) <td width="15%" className="text-center">Rp {item.mobil.hargasewa_asli * item.mobil_qty}</td> (8) <td width="10%"> <button type="button" onClick={ (e) => deleteSewaItem(e, item.id) } className="btn btn-danger btn- sm">Hapus</button> (9) </td> <td> <Link to="/checkout" className="btn btn-success btn- sm">Checkout</Link> (10) </td> </tr> </pre>	Berhasil menampilkan halaman list sewa



Gambar 5.14 Cyclomatic Complexity Menu

Pada gambar 5.14 merupakan *Cyclomatic Complexity* pada *flowgraph list sewa* pada aplikasi sewa mobil. Berikut penjelasan dari *Cyclomatic Complexity* di atas.

Jalur = 1,2,3,4,5,6,7,8,9,10

$$V(G) = 11 \text{ Edges} - 10 \text{ Node} + 2 = 1$$

Hasil tersebut sesuai dengan jumlah jalur yang dapat digunakan untuk menyelesaikan masalah

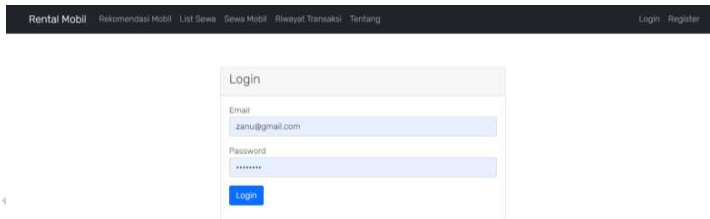
5.3.2 Black Box Testing

Black box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil input dan output dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik.

Tabel 5.26 Blackbox Testing

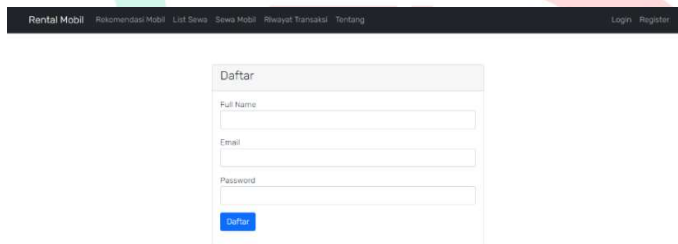
No	Skenario Pengujian	Hasil Pengujian
1	User mengakses halaman login	Berhasil menampilkan halaman login

Dokumentasi



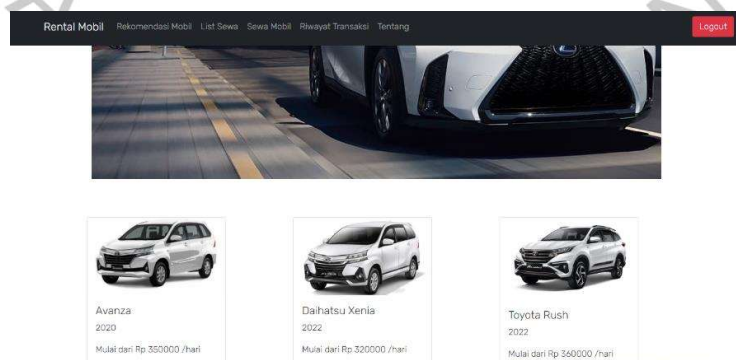
2	User mengakses halaman registrasi	Berhasil menampilkan halaman registrasi
---	-----------------------------------	-----------------------------------------

Dokumentasi



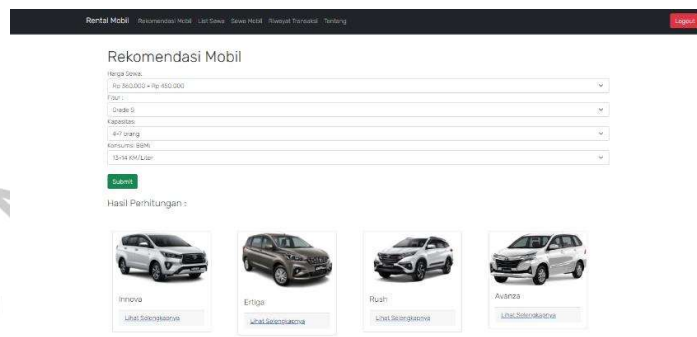
3	User mengakses halaman beranda	Berhasil menampilkan halaman beranda
---	--------------------------------	--------------------------------------

Dokumentasi



4	User mengakses halaman rekomendasi mobil	Berhasil menampilkan halaman rekomendasi mobil
---	------------------------------------------	------------------------------------------------

Dokumentasi



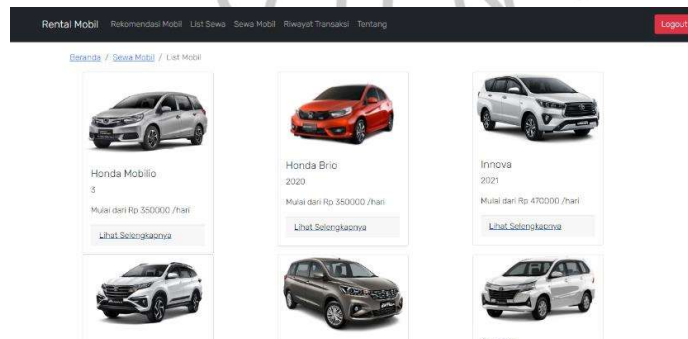
5	User mengakses halaman list sewa	Berhasil menampilkan halaman list sewa
---	----------------------------------	----------------------------------------

Dokumentasi



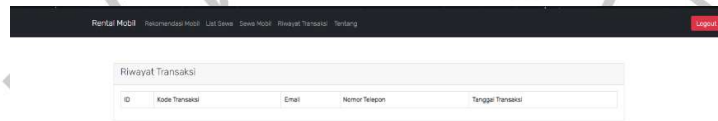
6	User mengakses halaman sewa mobil	Berhasil menampilkan halaman sewa mobil
---	-----------------------------------	-----------------------------------------

Dokumentasi



9 User mengakses halaman riwayat transaksi Berhasil menampilkan halaman riwayat transaksi

Dokumentasi



10 User mengakses halaman detail sewa Berhasil menampilkan halaman detail sewa

Dokumentasi

