BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.1 Implementasi Simulasi

Tahap implementasi sistem merupakan tahap pembuatan perangkat lunak, yang merupakan tahap lanjutan dari tahap perancangan sistem, mengimplementasikan Tahap yang dilakukan untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dimengerti oleh komputer serta penerapan perangkat lunak pada keadaan yang sebenarnya.

IV.1.1 Implementasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk membangun perangkat lunak simulasi pencarian rute terpendek di kota Tasikmalaya adalah sebagai berikut:

Tabel IV-1 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Core i5 2,5Ghz
2	RAM	4 Gb
3	Harddisk	500 Gb
4	VGA	2 Gb
5	Monitor	1336x768

IV.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membangun perangkat lunak pencarian rute terpendek di kota Tasikmalaya menggunakan algoritma pencarian beam adalah sebagai berikut:

Tabel IV-2 Spesifikasi Implementasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Windows 7	Sistem Operasi
2	Eclipse IDE	Tool Pembangun Perangkat Lunak (Android)
3	Macromedia Dreamweaver	Tool Pembangun Perangkat Lunak (*Web Service*)
4	Star UML	Pemodelan Sistem (Android)
5	Power Designer	Permodelan Sistem (*Web Service*)

IV.1.3 Implementasi Basis Data

Tabel IV-3 Implementasi Data

No	Nama Tabel	Nama File
1	Admin	admin.sql
2	Lokasi	lokasi.sql

Tabel IV-4 Query Admin

```
1  CREATE TABLE IF NOT EXISTS `admin` (
2      `username` varchar(50) NOT NULL,
3      `password` varchar(50) NOT NULL,
4      PRIMARY KEY (`username`)
5      ) ENGINE=MyISAM DEFAULT
       CHARSET=latin1;
6      PRIMARY KEY (`username`)
7  ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Tabel IV-5 Query Lokasi

```
1  CREATE TABLE IF NOT EXISTS `lokasi` (
2        `id` int(11) NOT NULL AUTO_INCREMENT,
3        `nama_lokasi` varchar(100) DEFAULT NULL,
4        `latitude` double NOT NULL,
5        `longitude` double NOT NULL,
6        PRIMARY KEY (`id`),
7        UNIQUE KEY `nama_titik` (`nama_lokasi`)
8  ) ENGINE=MyISAM  DEFAULT CHARSET=latin1
   AUTO_INCREMENT=122 ;
```

IV.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan untuk setiap tampilan program yang dibangun dalam bentuk file code. Pada aplikasi simulasi ini, terdapat dua aplikasi yang dibangun yaitu aplikasi android dan aplikasi web service.

1. Aplikasi Simulasi Pada Android

Tabel IV-6 Implementasi Antarmuka Aplikasi Simulasi Android

No	Form Antarmuka	Proses	Nama File	Keterangan
1	Form Halaman Utama	Memilih menu yang tersedia	ActivityUtama.java	Tampilan untuk memilih pilihan antara pencarian rute atau melakukan update data lokasi
2	From Pencarian Lokasi	Memasukan data lokasi dan data tujuan	Pencarian Rute.java	Tampilan untuk mengisi lokasi awal dan lokasi tujuan
3	Form Pengaturan Parameter	Memasukan atau mengubah parameter beam	Pengaturan.java	Tampilan untuk mengatur besar nilai parameter beam
4	Form Hasil Pencarian	Hasil pencarian lokasi	ActivityResult.java	Tampilan hasil pencarian rute terpendek

Gambar dari implementasi antarmuka dapat dilihat pada lampiran A.

2. Antarmuka *Web Service*

Tabel IV-7 Implementasi Antarmuka Web Service

No	Form Antarmuka	Proses	Nama File	Keterangan
1	Form Login	Halaman isian bila untuk masuk ke dalam sistem web service	login.php	Tampilan halaman utama dan merupakan halaman login admin
2	From Halaman Utama	Memilih menu yang tersedia	index.php	Tampilan menu awal web service
3	Form Tambah Data Lokasi	Memasukan data lokasi	tambah_lokasi.php	Tampilan untuk menambah data lokasi
4	Form Ubah Data Lokasi	Mengubah data Lokasi	ubah_lokasi.php	Tampilan untuk mengubah data lokasi

IV.2 Pengujian Simulasi

Pengujian merupakan hal terpenting yang bertujuan untuk menemukan kesalahan–kesalahan atau kekurangan–kekurangan pada perangkat lunak yang akan diuji. Pengujian dilakukan untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pada tahap pengujian simulasi perangkat lunak ini terdiri dari beberapa bagian yaitu :

1. Rencana Pengujian
2. Pengujian *White Box*
3. Pengujian *Black Box*
4. Kesimpulan Pengujian

IV.2.1 Rencana Pengujian

Rencana pengujian perangkat lunak ini dibagi menjadi 2 bagian yaitu pengujian *black box* dan pengujian *white box*. Pengujian *white box* dilakukan untuk menguji kesalahan logik dan asumsi yang tidak tepat pada kemungkinan eksekusi. Pengembangan kode-kode program selalu memungkinkan terjadinya alur program yang tidak tereksekusi dan kesalahan *typography* yang sulit ditemukan. Sedangkan Pengujian *black box* berfokus pada pengujian semua perintah-perintah dan kondisi yang dieksekusi oleh perangkat lunak dan pengujian ini dilakukan oleh pembangun aplikasi.

Tabel IV-8 Rencana Pengujian Aplikasi Simulasi Android

No	Item Uji	Detail Pengujian	Jenis Uji
1	Algoritma Pencarian Beam	Uji algoritma	*White box*
2	Pencarian Lokasi	Input lokasi awal lokasi tujuan	*Black Box*
3	Parameter	Input parameter algoritma	*Black Box*

Tabel IV-9 Rencana Pengujian Web Service

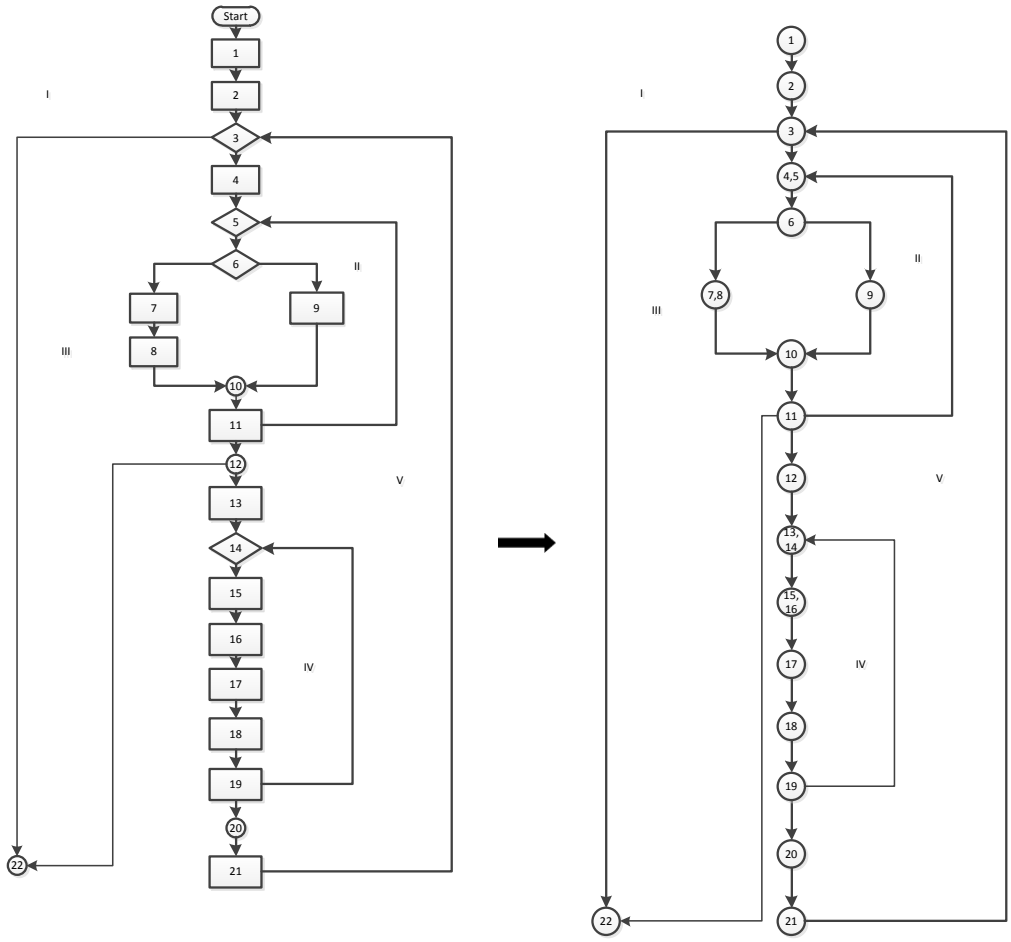
No	Item Uji	Detail Pengujian	Jenis Uji
1	Login	Pengujian validasi data login	*Black box*
2	Tambah Data Lokasi	Pengujian validasi penambahan data lokasi	*Black Box*

IV.2.2 Pengujian White Box

Pengujian *white box* bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak apakah berjalan dengan baik atau tidak. Pengujian *white box* akan digunakan pada algoritma pencarian beam, untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis. Adapun tahapan yang ada pada pengujuan *white box* yaitu:

1. Pengubahan pseucode menjadi *flowchart* kemudian menjadi *flowgraph.*

No	Algoritma Pencarian Beam
1	i ← 0
2	input beam
3	**while**(solusi[i] != null) **do**
4	j ← 0
5	**for** j = 0 to j < beam **do**
6	**if**(simpul = solusi[j]) then
7	solusi[j] ← simpul
8	j← beam-1
	else
9	antrian[j] ← simpul
10	endif
11	j ← j+1
12	**endfor**
13	j←0
14	**while** (antrian[j] != null) **do**
15	compare(h)
16	simpul ← h
17	delete(antrian[j])
18	solusi[j] ← simpul
19	j ← j+1
20	**endwhile**
21	i ←i+1
22	**endwhile**



Gambar IV-1 Pengubahan Flowchart Menjadi Flowgraph Algoritma Pencarian Beam

1. Cyclometic Complexity

   V(G) = E - N + 2

   V(G) = 21 – 18 + 2

   V(G) = 4

2. Independent Path

   Path 1: 1-2-3-22

   Path 2: 1-2-3-4-5-6-9-10-11-12-22

   Path 3: 1-2-3-4-5-6-7-8-11-12-22

   Path 4: 1-2-3-4-5-6-9-10-11-12-13-14-15-16-17-18-19-20-21-1-2-3-22

   Path 5: 1-2-3-4-5-6-9-10-11-12-13-14-15-16-17-18-19-20-21-1-2-3-22

3. Hitung Jumlah region

   Jumlah Region = Jumlah Kurva Tertutup + 1 (Region Terluar)

   Jumlah Region = 4 + 1

   Jumlah Region = 5

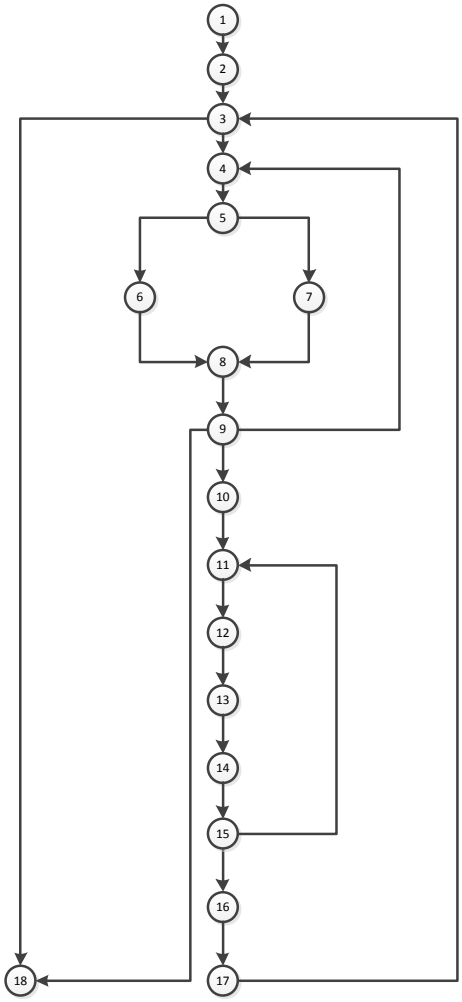
4. Hitung jumlah predicate Node

   $V(G) = Jumlahpredicate\ node + 1$

   $V(G) = 4 + 1 = 5$

   $V(G) = 5$

5. Penomoran ulang flowgraph



Gambar IV-2 Penomoran Ulang Flow Graph Penomoran Ulang Graph

6. Graph matriks

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Jumlah
1		1																	0
2			1																0
3				1														1	1
4					1														0
5						1	1												1
6								1											0
7								1											0
8									1										0
9										1								1	1
10											1								0
11												1							0
12													1						0
13														1					0
14															1				0
15											1					1			1
16																	1		0
17																		1	0
18			1																0
Jumlah																			4

$V(g) = x + 1$

$V(g) = 4 + 1$

$V(G) = 5$

7. Kesimpulan Pengujian White Box

Berdasarkan pengujian pada setiap metode, dihasilkan nilai Cyclomatic Complexity yang sama yaitu 5. Maka dapat disimpulkan bahwa pengujian *white*

box pada algoritma pencarian beam berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.3 Pengujian Black Box

Berdasarkan rencana pengujian yang telah disusun maka dapat dilakukan pengujian black box yang disajikan pada tabel berikut.

1. Pengujian *Black Box* Aplikasi Android

Tabel IV-10 Pengujian Pencarian Lokasi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Masukan data lokasi awal dan lokasi tujuan dan setelah itu klik tombol cari.	Menampilkan rute terpendek pada peta	Dapat menampilkan rute terpendek pada peta	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Lokasi awal dan lokasi tujuan dikosongkan.	Muncul pesan kesalahan data lokasi harus di isi.	Menampilkan pesan kesalahan data lokasi harus di isi.	[✓] Diterima [ ] Ditolak

Tabel IV-11 Pengujian Parameter

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Nilai parameter (besar nilai beam).	Menyimpan data parameter	Nilai parameter disimpan	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			

Data Masukan	*Yang Diharapkan*	*Pengamatan*	*Kesimpulan*
Nilai parameter dikosongkan.	Muncul pesan kesalahan parameter harus diisi.	Menampilkan pesan kesalahan data parameter harus di isi.	[✓] Diterima [ ] Ditolak

2. **Pengujian *Black Box* Web Service**

Tabel IV-12 Pengujian Login Web Service

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Masukan *username* dan *password* setelah itu klik tombol login.	Masuk ke dalam sistem *web service*	Dapat masuk kedalam sistem *web service*	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Username dan password kosong atau salah.	Muncul pesan kesalahan data harus diisi.	Menampilkan pesan kesalahan data harus di isi.	[✓] Diterima [ ] Ditolak

Tabel IV-13 Pengujian Tambah Data Lokasi

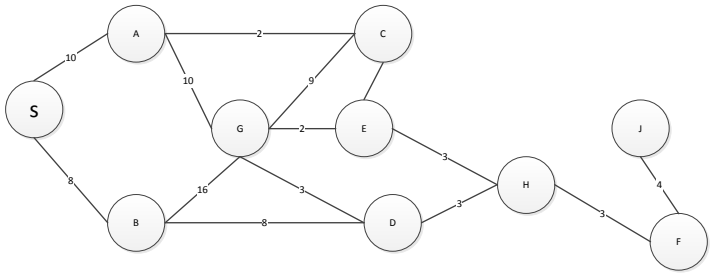
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Data lokasi	Menyimpan data lokasi	Data lokasi disimpan	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Kosong)			
Data Masukan	***Yang Diharapkan***	***Pengamatan***	***Kesimpulan***
Data lokasi dikosongkan.	Muncul pesan kesalahan data	Menampilkan pesan kesalahan	[✓] Diterima

	lokasi harus di isi.	data lokasi harus di isi.	[ ] Ditolak

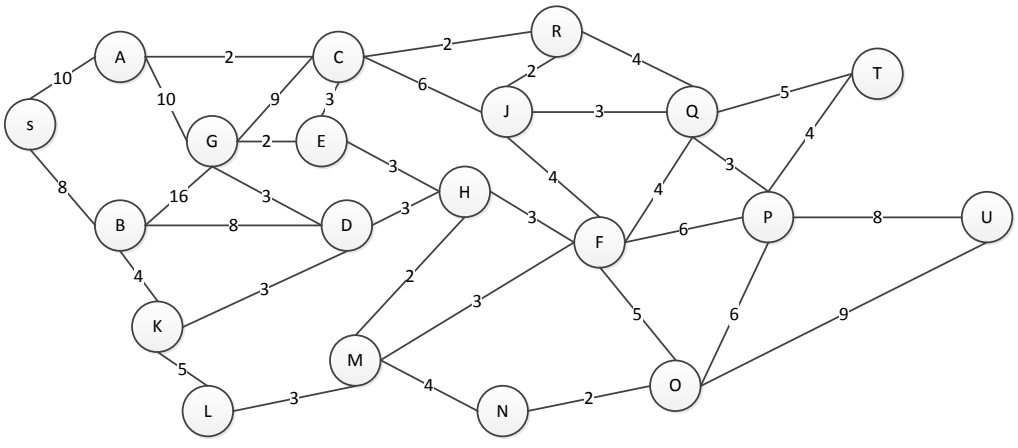
V.2.4 Pengujian Performansi

Pada pengujian performansi algoritma ini mendapatkan besar nilai beam yang berbeda. Pengujian dimulai dari simpul sebanyak sepuluh hingga tiga puluh simpul dengan nilai beam yang dimulai dari satu sampai lima. Berikut adalah hasil pengujian algoritma pencarian beam.

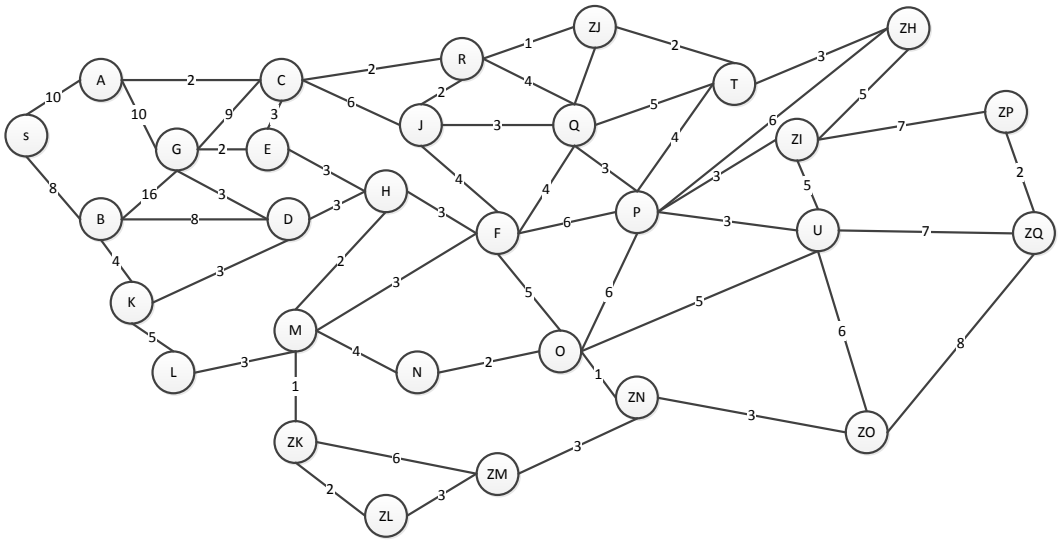
Parameter nilai beam yang diujikan hingga lima dan jumlah penelusuran simpul yang di ujikan hingga tiga puluh simpul. Berikut adalah bentuk graph yang akan digunakan dalam menguji algoritma pencarian beam:



Gambar IV-3 Graph 10 Simpul



Gambar IV-4 Graph 20 Simpul



Gambar IV-5 Graph 30 Simpul

Tabel IV-14 Tabel Pengujian Algoritma Pencarian Beam Berdasarkan Besar Nilai Beam

No	Nilai Beam	Jumlah Simpul	Simpul		Simpul Tujuan ditemukan
			Awal	Tujuan	
1	1	10	S	J	Tidak
2	1	10	S	F	Tidak
3	1	10	S	H	Tidak
4	1	20	S	U	Tidak
5	1	20	S	T	Tidak
6	1	20	S	O	Tidak
7	1	30	S	ZP	Tidak
8	1	30	S	ZQ	Tidak
9	1	30	S	ZO	Tidak
10	2	10	S	J	Ya
11	2	10	S	F	Ya
12	2	10	S	H	Ya
13	2	20	S	U	Ya
14	2	20	S	T	Ya
15	2	20	S	O	Ya
16	2	30	S	ZP	Tidak
17	2	30	S	ZQ	Tidak
18	2	30	S	ZO	Ya
19	3	10	S	J	Ya
20	3	10	S	F	Ya
21	3	10	S	H	Ya
22	3	20	S	U	Ya

No	Nilai Beam	Jumlah Simpul	Simpul		Simpul Tujuan ditemukan
			Awal	Tujuan	
23	3	20	S	T	Ya
24	3	20	S	O	Ya
25	3	30	S	ZP	Tidak
26	3	30	S	ZQ	Tidak
27	3	30	S	ZO	Ya
28	4	10	S	J	Ya
29	4	10	S	F	Ya
30	4	10	S	H	Ya
31	4	20	S	U	Tidak
32	4	20	S	T	Tidak
33	4	20	S	O	Ya
34	4	30	S	ZP	Ya
35	4	30	S	ZQ	Ya
36	4	30	S	ZO	Ya
37	5	10	S	J	Ya
38	5	10	S	F	Ya
39	5	10	S	H	Ya
40	5	20	S	U	Ya
41	5	20	S	T	Ya
42	5	20	S	O	Ya
43	5	30	S	ZP	Ya
44	5	30	S	ZQ	Ya
45	5	30	S	ZO	Ya

Pada Tabel IV-14 Tabel Pengujian Algoritma Pencarian Beam Berdasarkan Besar Nilai Beam, dapat dilihat bahwa dengan nilai beam 1 atau nilai yang terkecil, lokasi tujuan tidak ditemukan meskipun tujuan berbeda-beda. Ketika nilai besar beam di ubah menjadi lebih besar, solusi tujuan ditemukan meskipun dengan lebar beam yang semakin besar, tidak menjamin tujuan ditemukan. Hal ini dapat disebabkan oleh pemangkasan simpul ketika proses penelusuran simpul. Perubahan simpul tujuan juga dapat mempengaruhi apakah tujuan ditemukan atau tidak.

Tabel IV-15 Tabel Pengujian Algoritma Pencarian Beam Berdasarkan Simpul Tujuan

No	Nilai Beam	Jumlah Simpul	Simpul		Simpul Tujuan ditemukan
			Awal	Tujuan	
1	1	10	S	J	Tidak
2	1	20	S	U	Tidak
3	1	30	S	ZQ	Tidak
4	2	10	S	J	Ya

No	Nilai Beam	Jumlah Simpul	Simpul		Simpul Tujuan ditemukan
			Awal	Tujuan	
5	2	20	S	U	Ya
6	2	30	S	ZQ	Tidak
7	3	10	S	J	Ya
8	3	20	S	U	Ya
9	3	30	S	ZQ	Tidak
10	4	10	S	J	Ya
11	4	20	S	U	Tidak
12	4	30	S	ZQ	Ya
13	5	10	S	J	Ya
14	5	20	S	U	Ya
15	5	30	S	ZQ	Ya

Pada Tabel IV-15 Tabel Pengujian Algoritma Pencarian Beam Berdasarkan Simpul Tujuan, dapat dilihat bahwa pencarian beam menghasilkan pencarian yang konsisten ketika dibandingkan dengan pengujian berdasarkan besar nilai beam.

Tabel IV-16 Pengujian Pada Aplikasi Simulasi Android

No	Nilai Beam	Lokasi Awal	Lokasi Tujuan	Rute ditemukan
1	1	Mesjid Agung	Alfamart Galunggung	Ya
2	1	Yogya Toserba	PLN Tasikmalaya	Ya
3	1	Stasiun Tasikmalaya	TMP Kusuma Bangsa	Ya
4	1	Asia Plaza	Pasar Cikurubuk	Ya
5	1	Rumah makan jalur hijau	Bank Mega	Ya
6	2	Mesjid Agung	Alfamart Galunggung	Ya
7	2	Yogya Toserba	PLN Tasikmalaya	Ya
8	2	Stasiun Tasikmalaya	TMP Kusuma Bangsa	Ya
9	2	Asia Plaza	Pasar Cikurubuk	Ya
10	2	Rumah makan jalur hijau	Bank Mega	Ya
11	3	Mesjid Agung	Alfamart Galunggung	Ya
12	3	Yogya Toserba	PLN Tasikmalaya	Ya
13	3	Stasiun Tasikmalaya	TMP Kusuma Bangsa	Ya
14	3	Asia Plaza	Pasar Cikurubuk	Ya
15	3	Rumah makan	Bank Mega	Ya

No	Nilai Beam	Lokasi Awal	Lokasi Tujuan	Rute ditemukan
		jalur hijau		
16	4	Mesjid Agung	Alfamart Galunggung	Ya
17	4	Yogya Toserba	PLN Tasikmalaya	Ya
18	4	Stasiun Tasikmalaya	TMP Kusuma Bangsa	Ya
19	4	Asia Plaza	Pasar Cikurubuk	Ya
20	4	Rumah makan jalur hijau	Bank Mega	Ya
21	5	Mesjid Agung	Alfamart Galunggung	Ya
22	5	Yogya Toserba	PLN Tasikmalaya	Ya
23	5	Stasiun Tasikmalaya	TMP Kusuma Bangsa	Ya
24	5	Asia Plaza	Pasar Cikurubuk	Ya
25	5	Rumah makan jalur hijau	Bank Mega	Ya

Pada Tabel IV-16 Pengujian Pada Aplikasi Simulasi Android, dapat dilihat bahwa algoritma pencarian beam dapat membantu dalam menemukan rute terpendek di kota Tasikmalaya, karena setelah diuji dengan lokasi awal dan lokasi tujuan yang berbeda, rute dapat ditemukan dengan algoritma pencarian beam.

IV.2.5 Kesimpulan Pengujian

Berdasarkan hasil pengujian, dapat diambil kesimpulan bahwa besar nilai beam pada algoritma pencarian beam berpengaruh dalam menemukan tujuan atau solusi meskipun nilai lebar beam tidak menjamin solusi ditemukan atau tidak. Pencarian memungkinkan tidak menemukan solusi yang dikarenakan pemangkasan simpul. Dengan simpul tujuan yang berbeda, dapat dihasilkan hasil pencarian yang berbeda. Sedangkan pada aplikasi android, algoritma pencarian beam membantu dalam mencari rute terpendek dengan nilai parameter yang berbeda.

.