ANALISIS PERFOMANSI ALGORITMA BEAM PADA PENCARIAN RUTE TERPENDEK DI KOTA TASIKMALAYA

INFORMATIK TEORITIK



SIDANG SKRIPSI
Oleh :
Aldy Mercyano Iqbal
10108065

Penguji 1:
Galih Hermawan ,S.Kom., M.T.

Penguji 2:
Adam Mukharil Bachtiar, S.Kom.

Penguji 3:
Tati Harihayati M., S.T., M.T.

LATAR BELAKANG





Penelitian sebelumnya menggunakan algoritma *ant colony* namun dirasa LAMBAT pencariannya.



PERUMUSAN MASALAH



Menganalisis Performansi
Algortima Pencarian Beam
Dalam Mencari Rute Terpendek
di Kota Tasikmalaya.

TUJUAN



Mengetahui Perfomansi

Algoritma Pencarian Beam Dalam Menentukan Rute Terpendek di Kota Tasikmalaya.

BATASAN MASALAH

ALGORITMA

BEAM

PERHITUNGAN KOMPLEKSITAS

KEBUTUHAN WAKTU

NOTASI ASIMPTOTIK

BIG O

APLIKASI

SIMULASI

MASUKAN

TITIK AWAL DAN AKHIR

MEDIA

ANDROID



Algoritma Pencarian Beam

PENGERTIAN ALGORITMA

1 2 3 4 5

Metode heuristik untuk menyelesaikan masalah optimasi kombinatorial

Optimasi algoritma *best-first search* untuk mengurangi kebutuhan memori.

ANALISIS KASUS

1 2 3 4 5

- Pencarian rute terpendek.
- Kelemahan algoritma pada penelitian sebelumnya dalam menentukan rute terpendek di kota Tasikmalaya.

ANALISIS STRUKTUR DATA

1 2 3 4 5

Graph : struktur data yang tersusun atas simpul-simpul atau titik-titik (Simpul) yang terhubung satu sama lain dengan garis-garis (*edge*).

JENIS	REPRESENTASI
→ BERARAH → TIDAK BERARAH	→ ADJACENCY LIST → ADJACENCY MATRIX

ANALISIS STRATEGI (1)

1 2 3 4 5

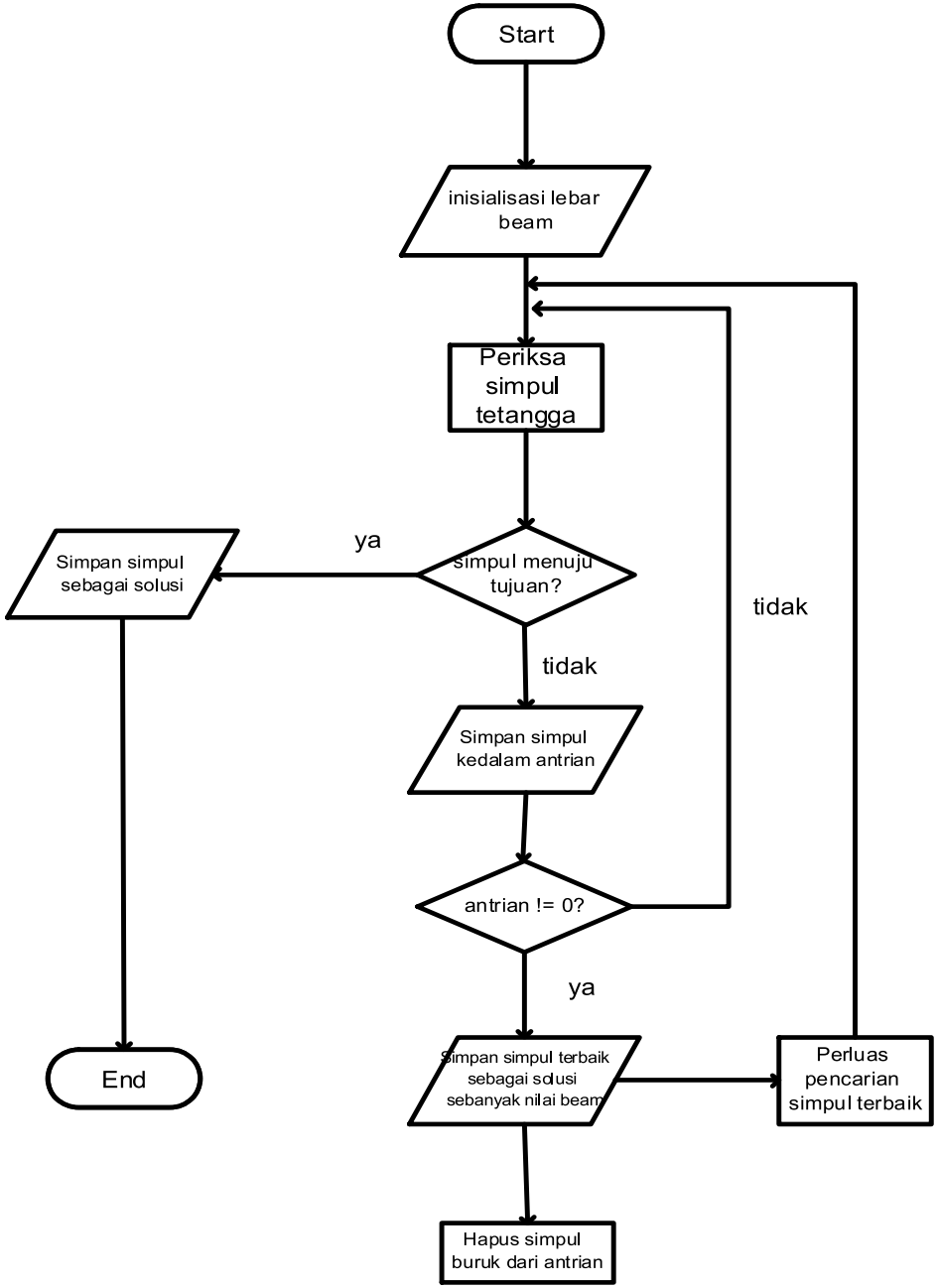
Algoritma pencarian beam merupakan optimasi dari algoritma *best-first search.*

ANALISIS STRATEGI (2)

1 2 3 4 5

1. Asumsikan senarai untuk simpul yang sudah dibangkitkan namun berlum diuji sebagai list OPEN dan senarai untuk simpul yang telah diuji sebagai CLOSE.
2. Masukan simpul awal pada list OPEN.
3. Kerjakan langkah-langkah berikut hingga tujuan ditemukan atau antrian OPEN sudah kosong:
    a) Ambil simpul terbaik dari list OPEN.
    b) Bangkitkan semua suksesor dari simpul yang ada di list OPEN.
    c) Untuk setiap suksesor kerjakan:
        - Jika simpul tersebut belum pernah dibangkitkan, periksa simpul tersebut dan masukkan ke OPEN;
        - Jika simpul tersebut sudah pernah dibangkitkan, ubah parent jika lintasan baru lebih menjanjikan. Hapus simpul tersebut dari antrian OPEN.

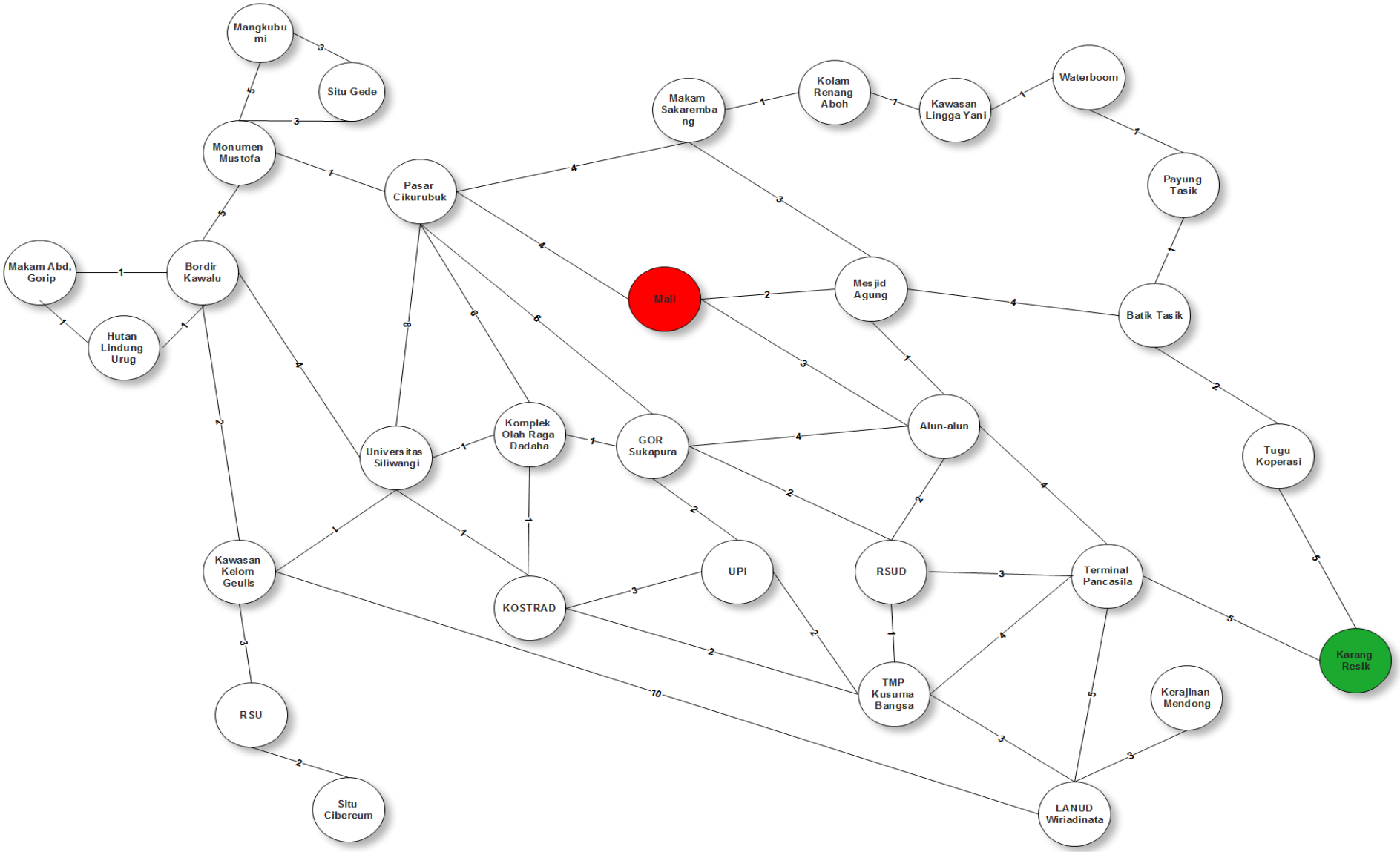
FLOWCHART ALGORITMA PENCARIAN BEAM


Start
inisialisasi lebar beam
Periksa simpul tetangga
simpul menuju tujuan?
ya
Simpan simpul sebagai solusi
End
tidak
Simpan simpul kedalam antrian
antrian != 0?
tidak
ya
Simpan simpul terbaik sebagai solusi sebanyak nilai beam
Perluas pencarian simpul terbaik
Hapus simpul buruk dari antrian



CONTOH KASUS

PETA TASIKMALAYA



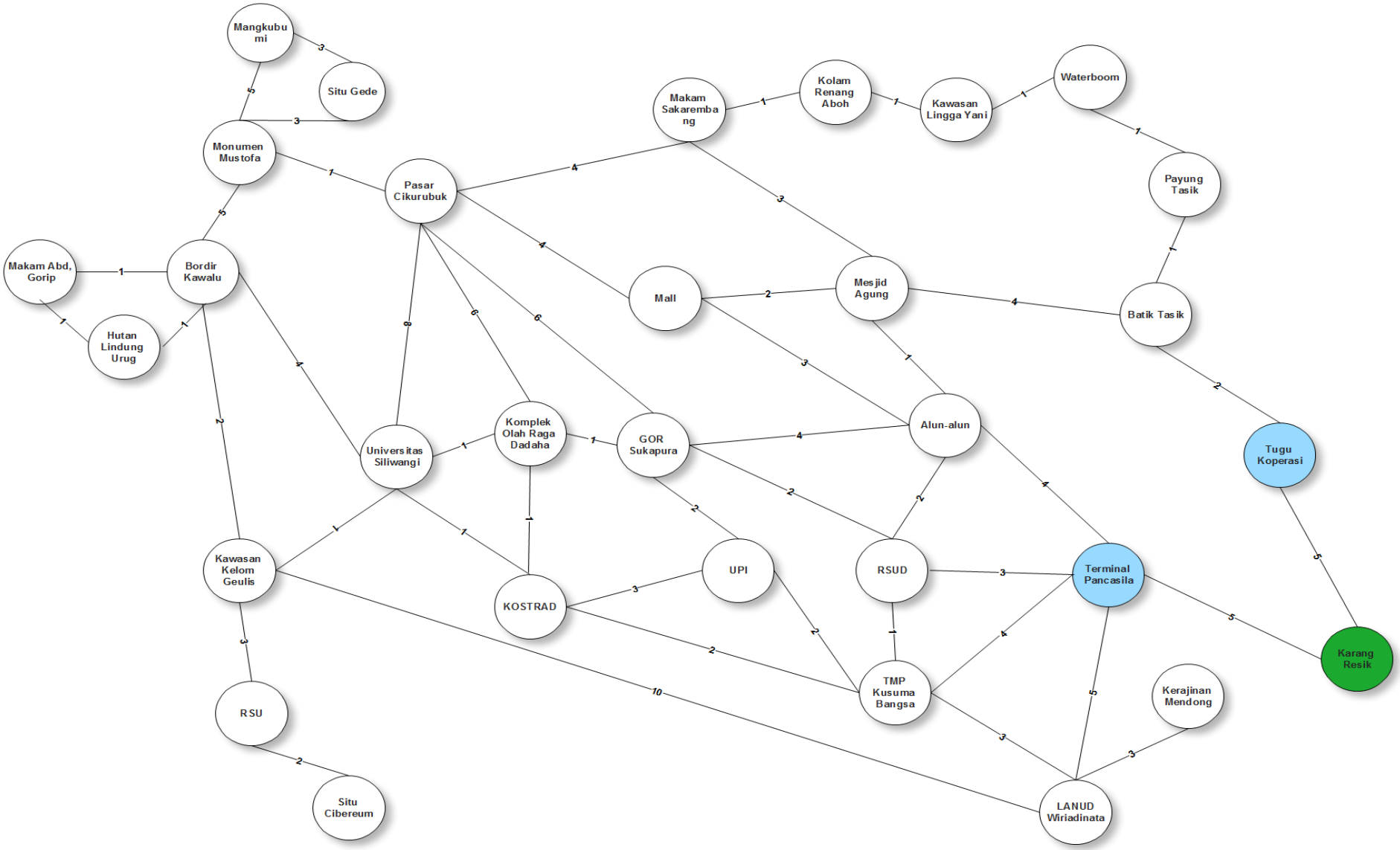
KONDISI AWAL

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik	(Karang Resik)

KONDISI PENELUSURAN PERTAMA

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik	(Karang Resik)
	(Karang Resik, Terminal Pancasila), (Karang Resik, Tugu Koperasi)

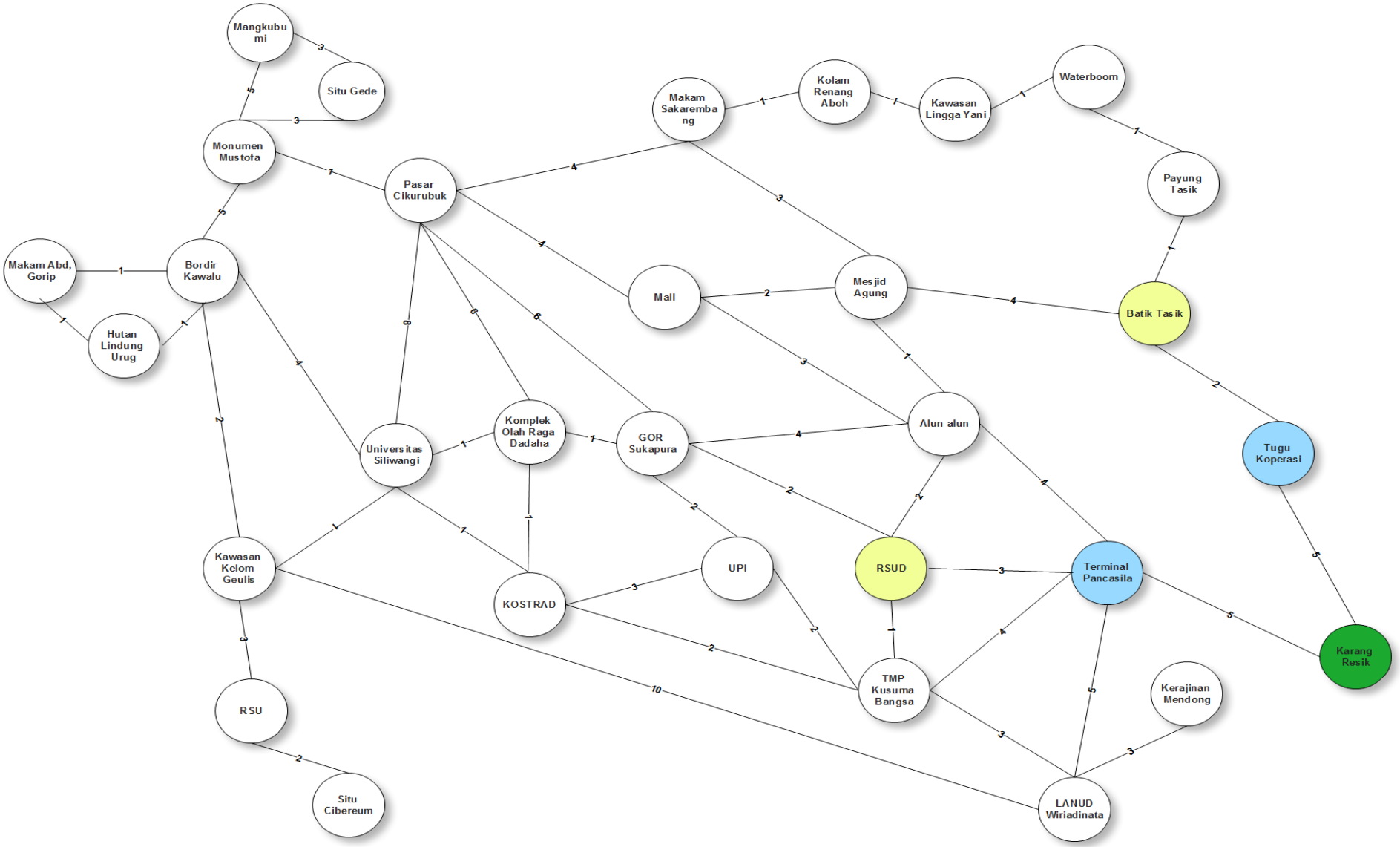
PENELUSURAN PERTAMA



KONDISI PENELUSURAN KE DUA

Simpul Yang Diperiksa	Urutan Simpul
Karang Resik dan Tugu Koperasi	(Karang Resik, Terminal Pancasila), (Karang Resik, Tugu Koperasi)
	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)

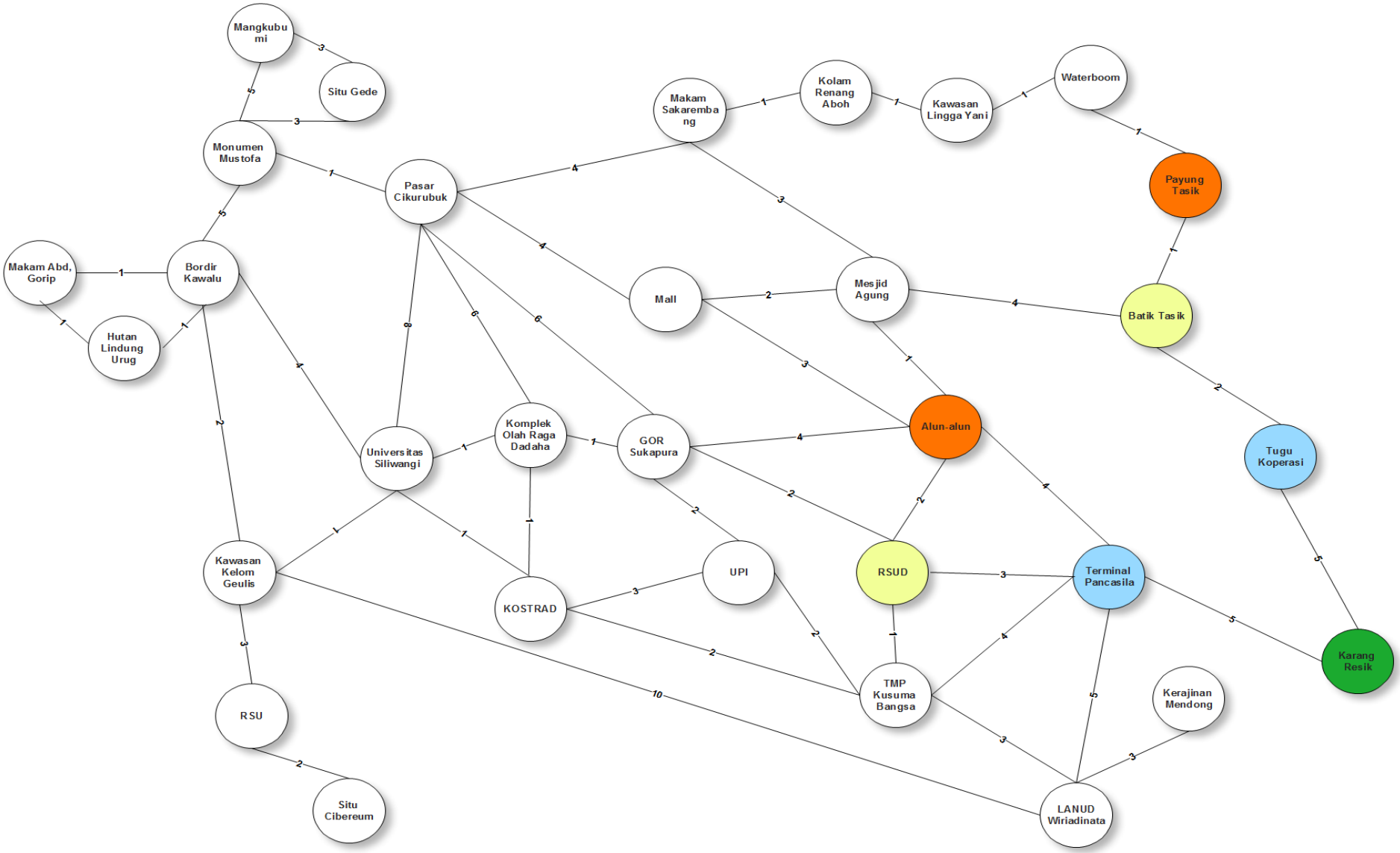
PENELUSURAN KE DUA



KONDISI PENELUSURAN KE TIGA

Simpul Yang Diperiksa	Urutan Simpul
RSUD dan Batik Tasik	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)
	(Karang Resik, Terminal Pancasila, RSUD, Alun-Alun), (Karang Resik, Tugu Koperasi, Batik Tasik, Payung Tasik)

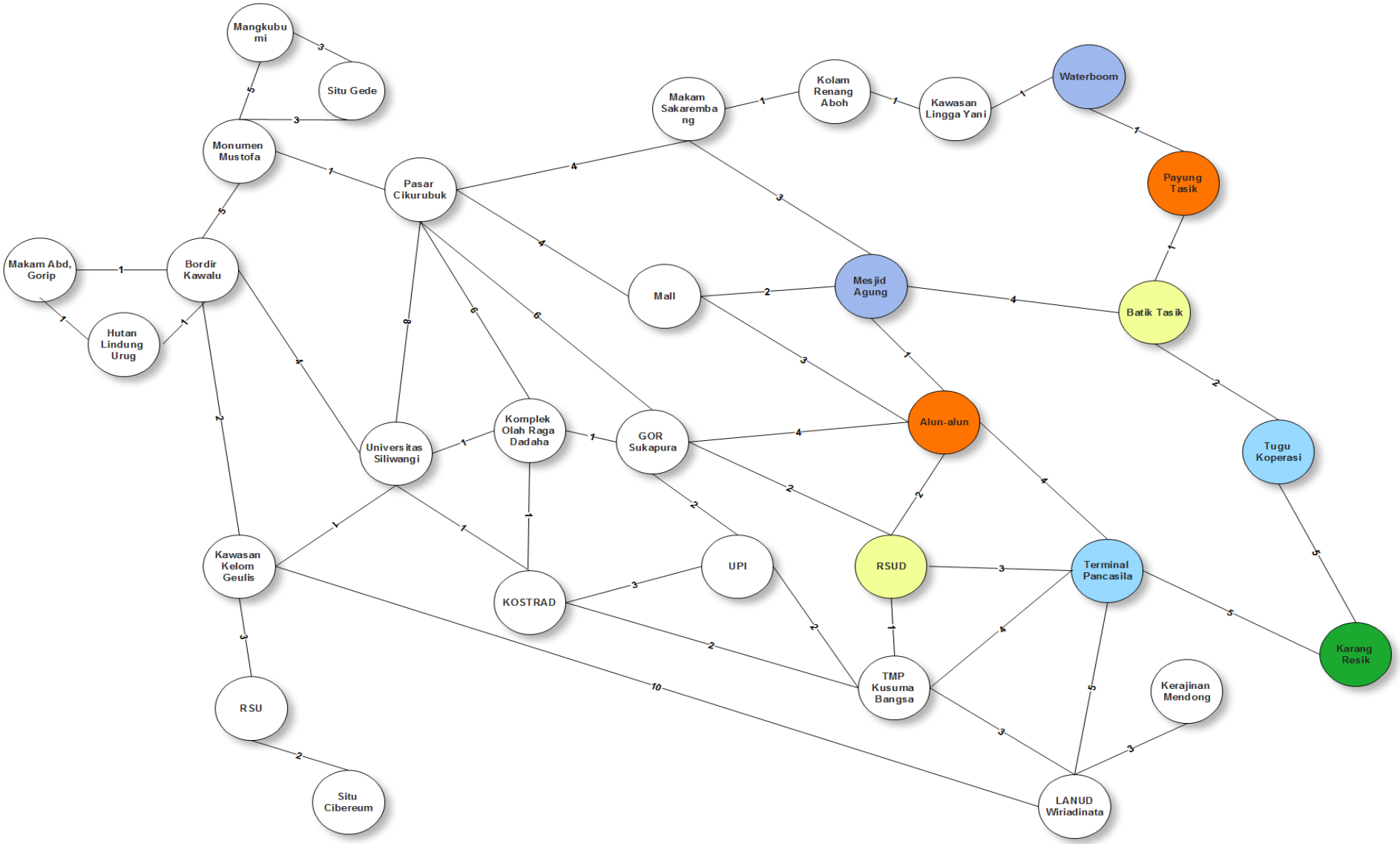
PENELUSURAN KE TIGA



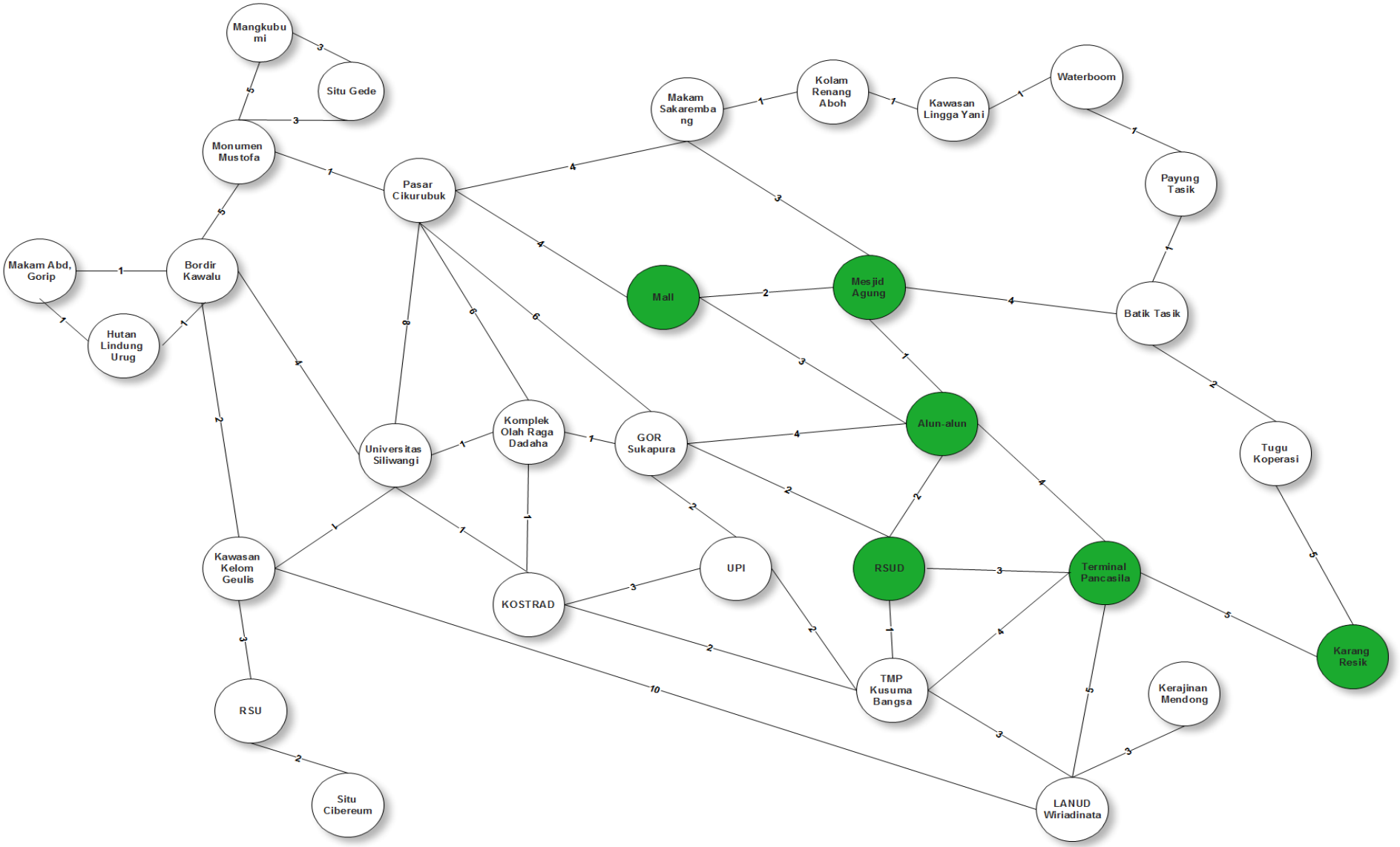
KONDISI PENELUSURAN KE EMPAT

Simpul Yang Diperiksa	Urutan Simpul
Alun-alun dan Payung Tasik	(Karang Resik, Terminal Pancasila, RSUD), (Karang Resik, Tugu Koperasi, Batik Tasik)
	(Karang Resik, Terminal Pancasila, RSUD, Alun-Alun, Mesjid Agung), (Karang Resik, Tugu Koperasi, Batik Tasik, Payung Tasik, Waterboom)

PENELUSURAN KE EMPAT



HASIL PENCARIAN





PENGUJIAN

RENCANA PENGUJIAN

Pengujian 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

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

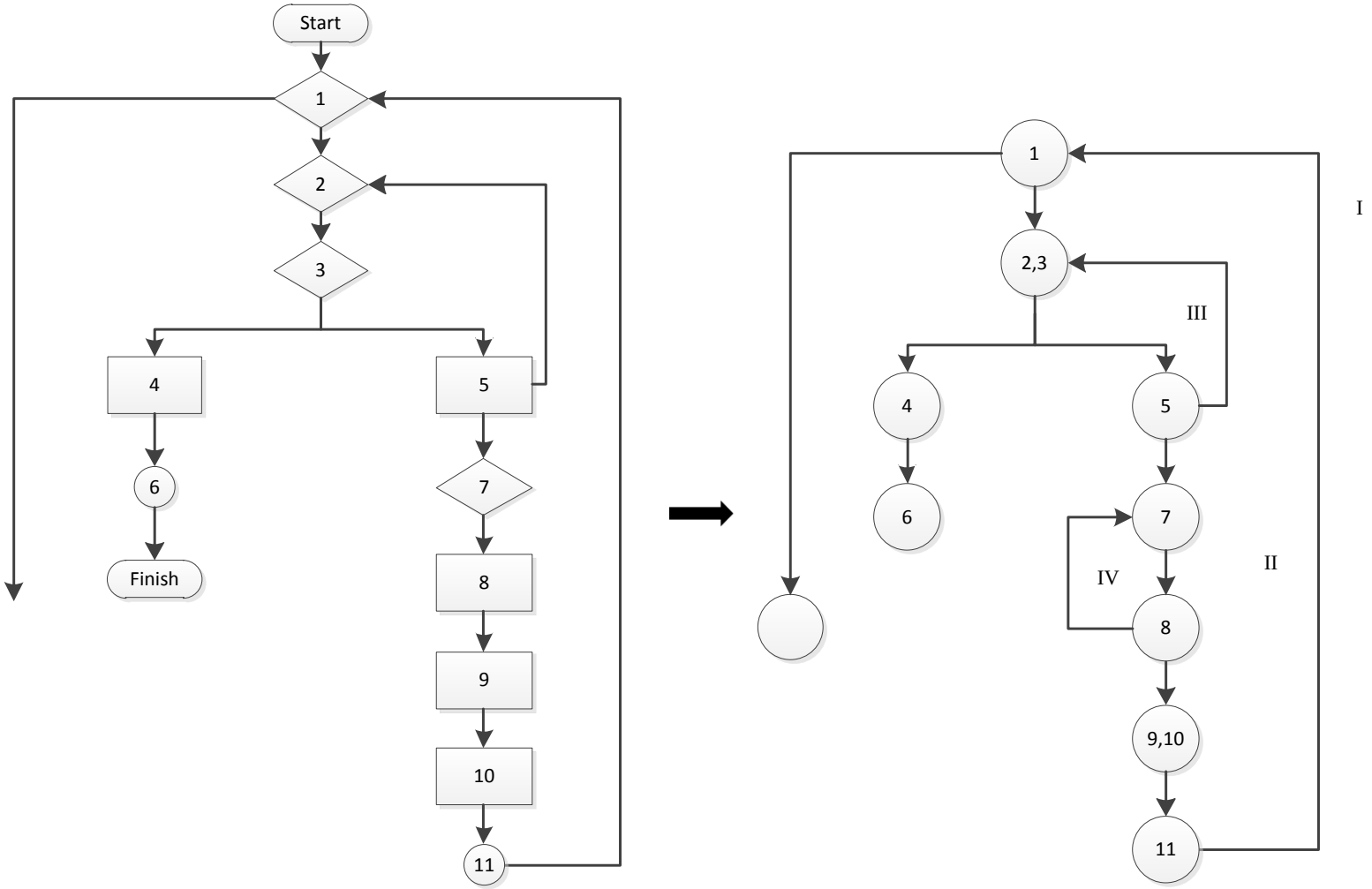
PENGUJIAN *WHITE BOX*

1 2 3 4 5

1	**while(solusi != 0) do**
2	for sejumlah simpul dengan ukuran lebar_beam do
3	if(simpul = sousi) then
4	return solusi ← {simpul};
5	antrian ← {simpul};
6	endif
	endfor
7	while((antrian != 0) and (lebar_beam > solusi)) do
8	simpul ← simpul dalam antrian dengan nilai h terkecil;
9	antrian ← hapus nilai yang tidak terpakai;
10	solusi ← {simpul};
11	endwhile
12	Endwhile

PENGUJIAN *WHITE BOX*

1 2 3 4 5



PENGUJIAN *WHITE BOX*

1 2 3 4 5

Cyclometic Complexity
V(G) = E - N + 2
V(G) = 12 – 10 + 2
V(G) = 4

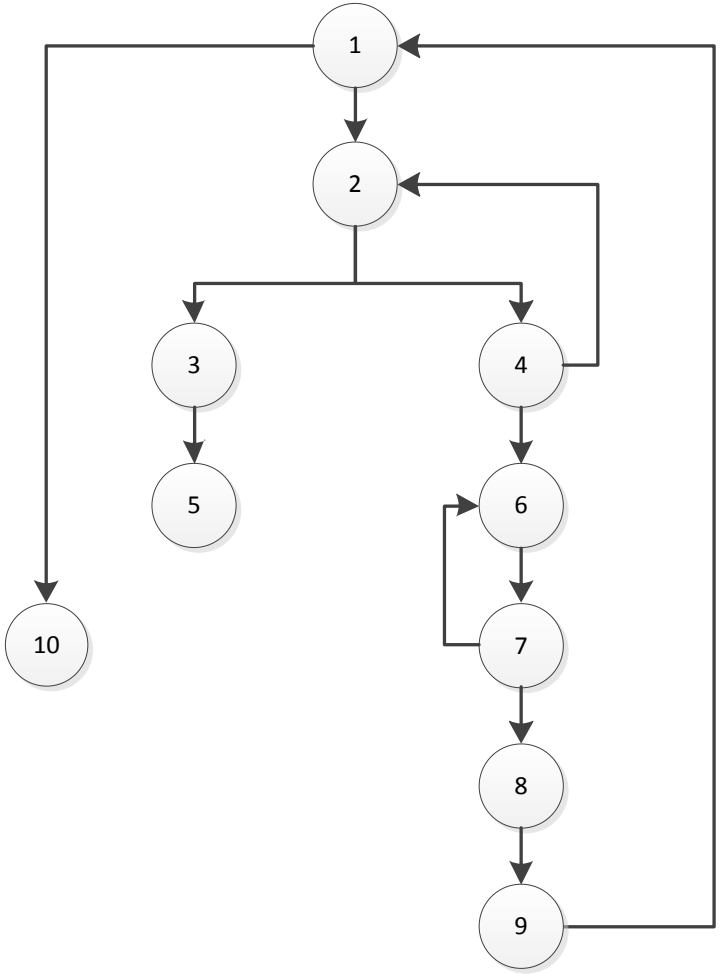
Independent Path
Path 1: 1-2-3-5-7-8-9-10-11-1-12
Path 2: 1-2-3-5-7-8-9-10-11-1
Path 3: 1-2-3-5-2-3
Path 4: 1-2-3-5-7-8-7

Hitung Jumlah region
Jumlah Region = 3 + 1
Jumlah Region = 4

Hitung jumlah predicate Node
$V(G) = 3 + 1 = 3$
$V(G) = 4$

PENGUJIAN *WHITE BOX*

1 2 3 4 5



PENGUJIAN *WHITE BOX*

1 2 3 4 5

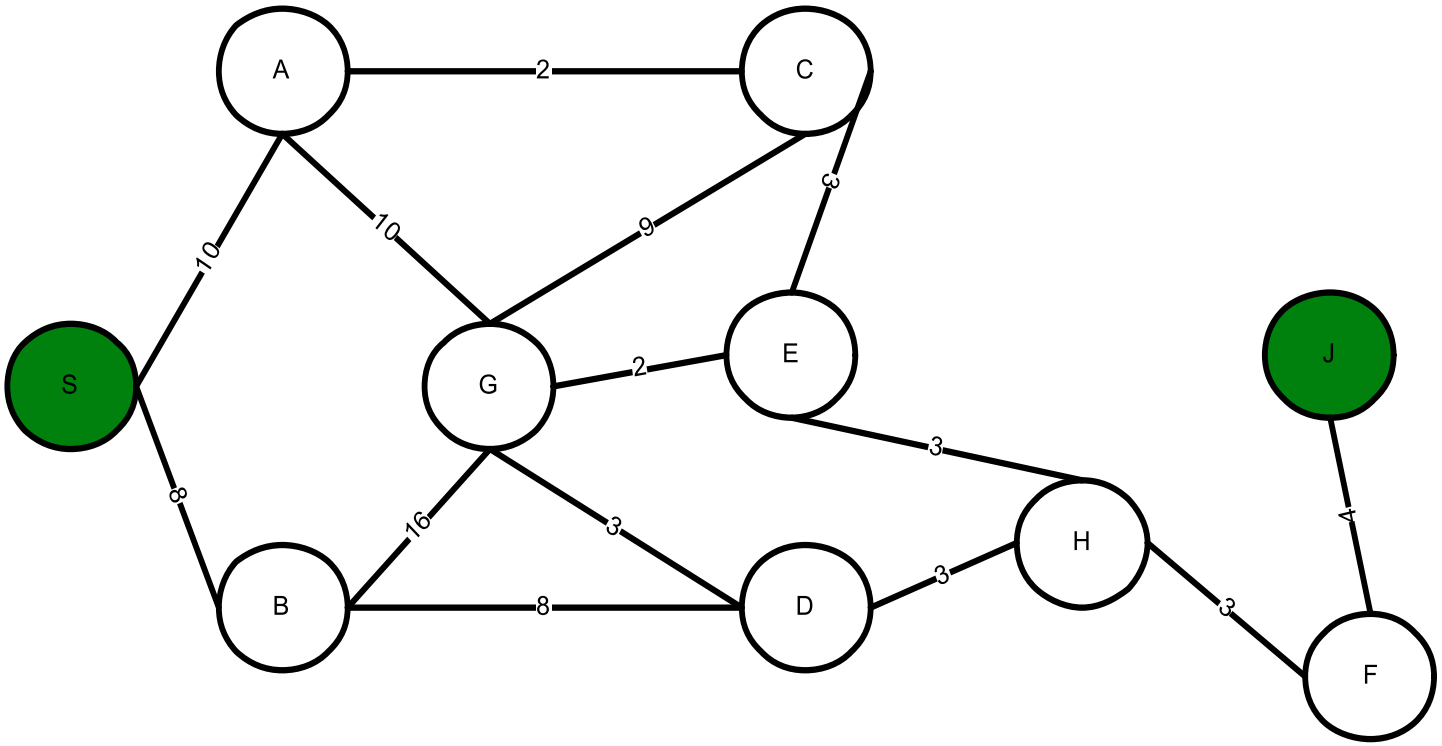
Node	1	2	3	4	5	6	7	8	9	10	Jumlah
1		1									0
2			1	1							1
3					1						0
4		1				1					1
5											0
6							1				0
7						1		1			1
8									1		0
9	1										0
10											0
Jumlah											3

$$V(g) = x + 1$$
$$V(g) = 3 + 1$$
$$V(G) = 4$$



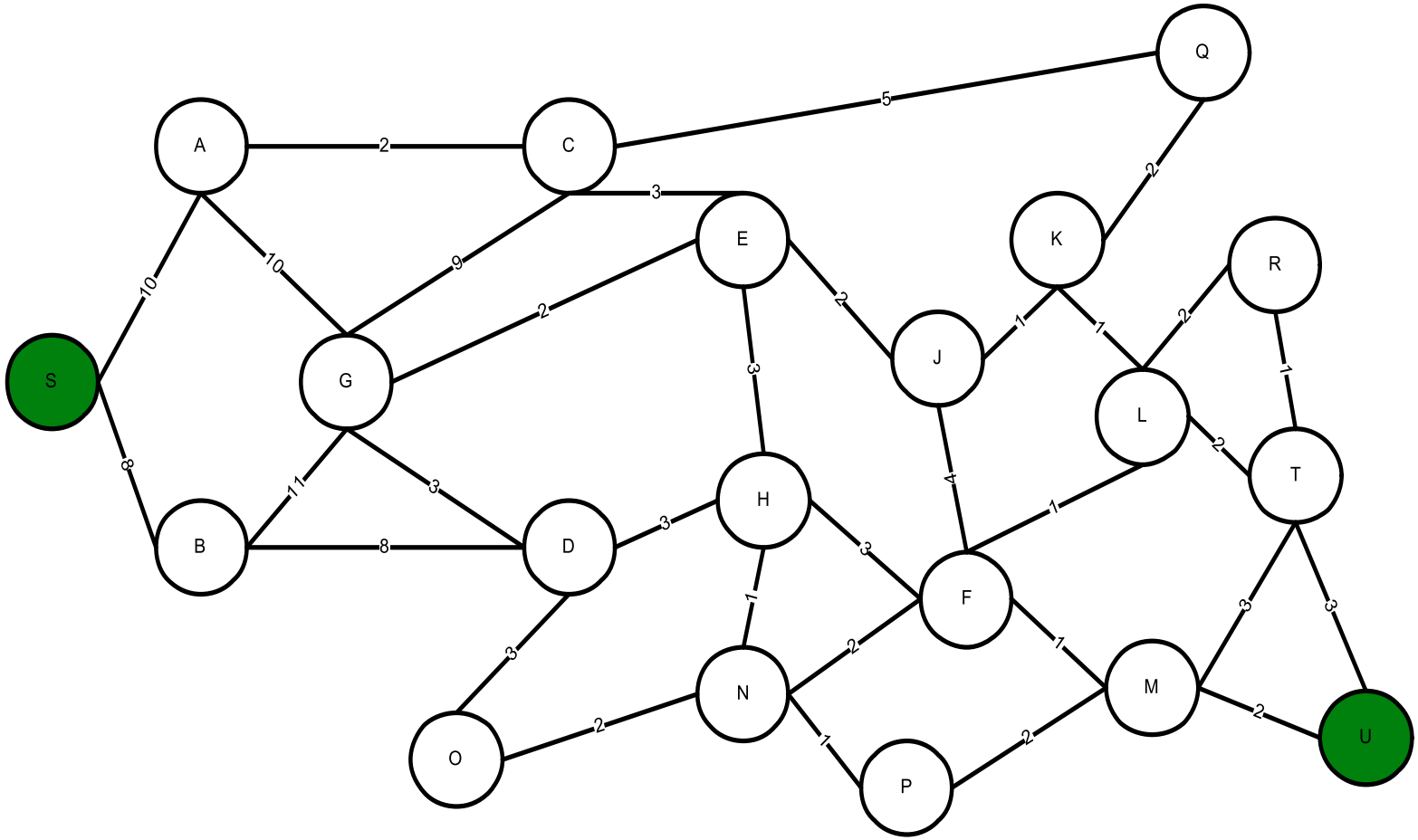
PENGUJIAN PERFORMANSI

SEPULUH SIMPUL



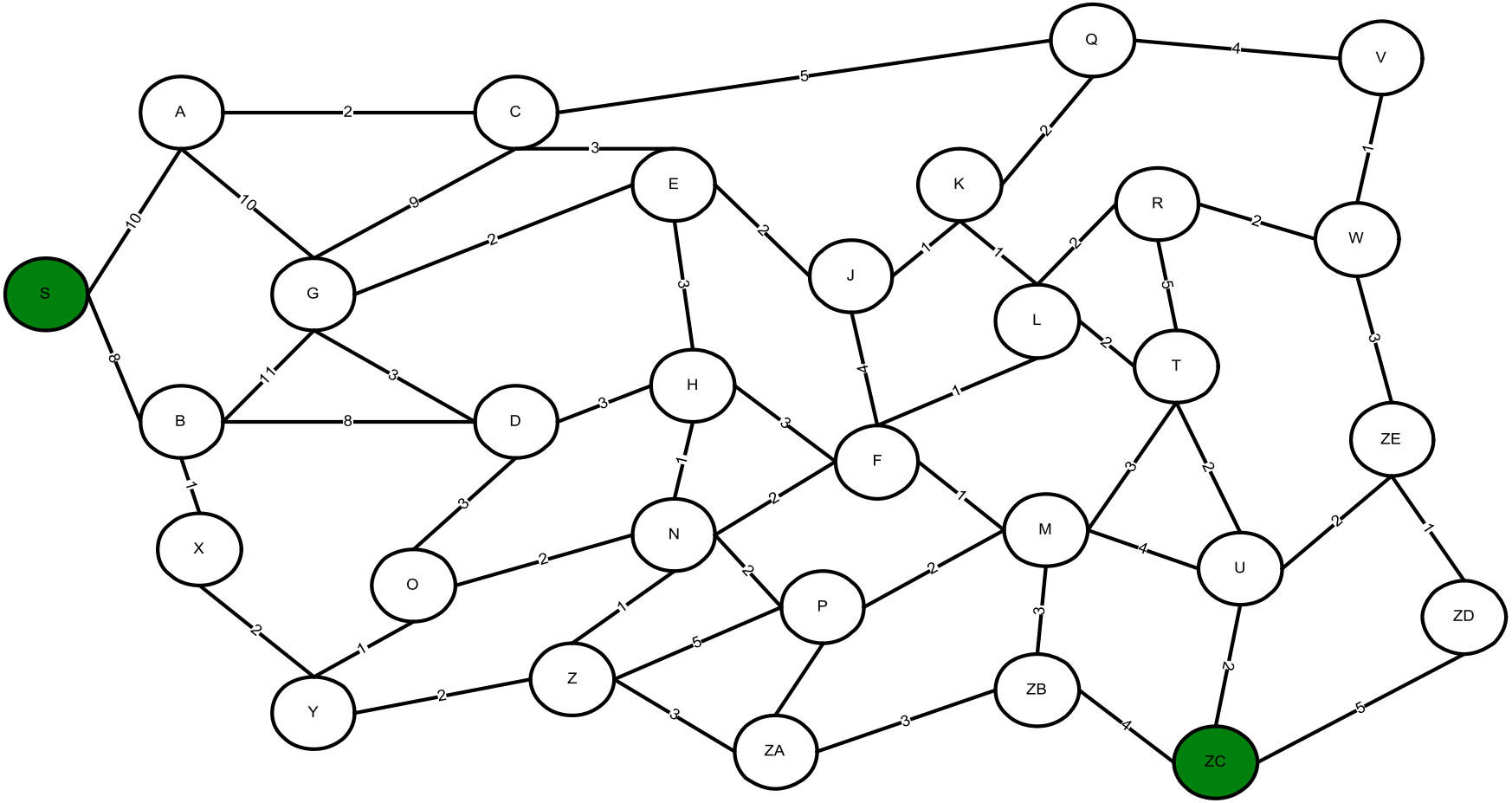
No	Lebar Beam	Simpul Tujuan ditemukan
1	1	Tidak
2	2	Ya
3	3	Ya
4	4	Ya
5	5	Ya
6	6	Ya
7	7	Ya
8	8	Ya
9	9	Ya
10	10	Ya

DUA PULUH SIMPUL



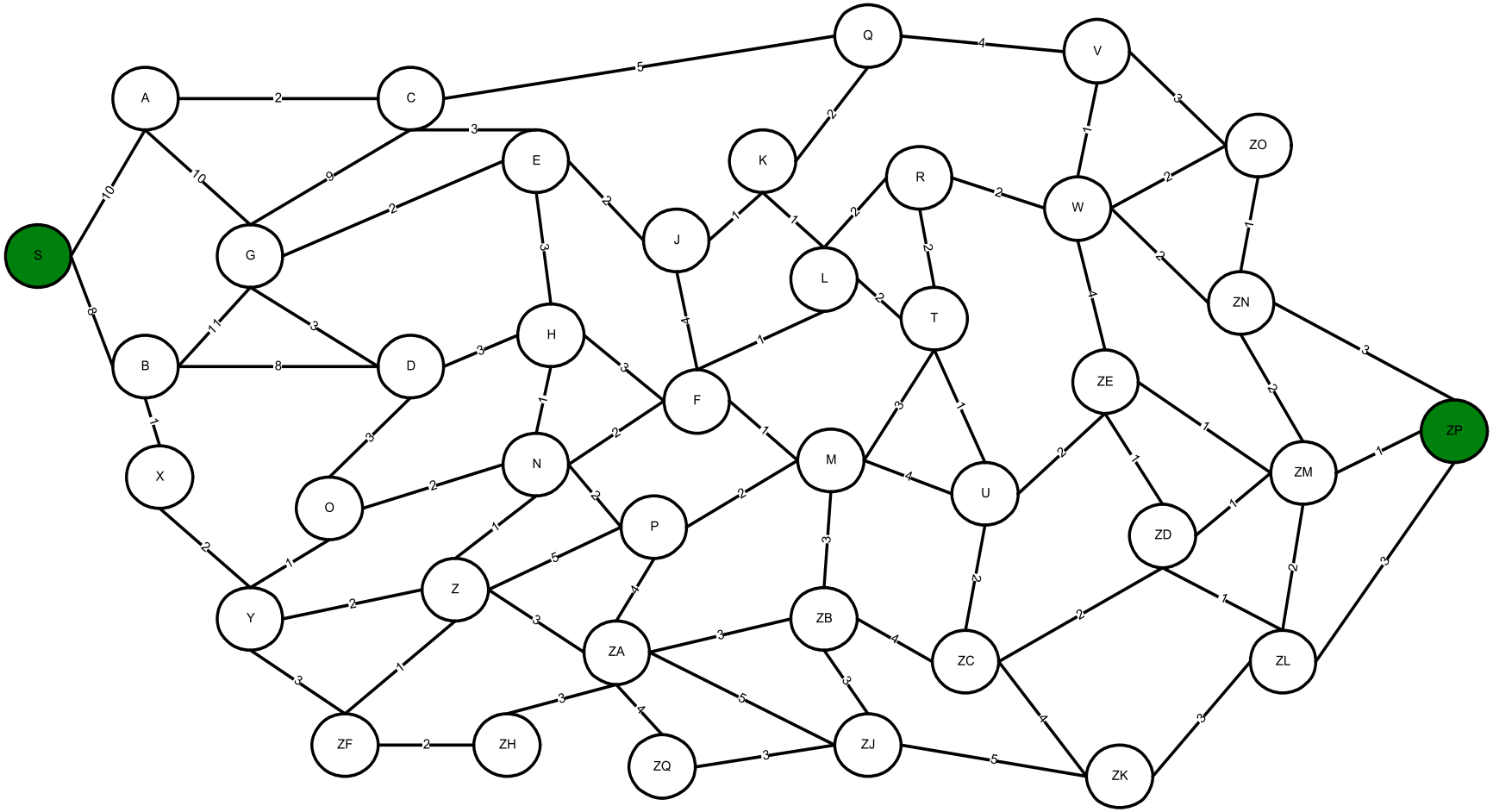
No	Lebar Beam	Simpul Tujuan ditemukan
1	1	Tidak
2	2	Ya
3	3	Ya
4	4	Ya
5	5	Ya
6	6	Ya
7	7	Ya
8	8	Ya
9	9	Ya
10	10	Ya

TIGA PULUH SIMPUL



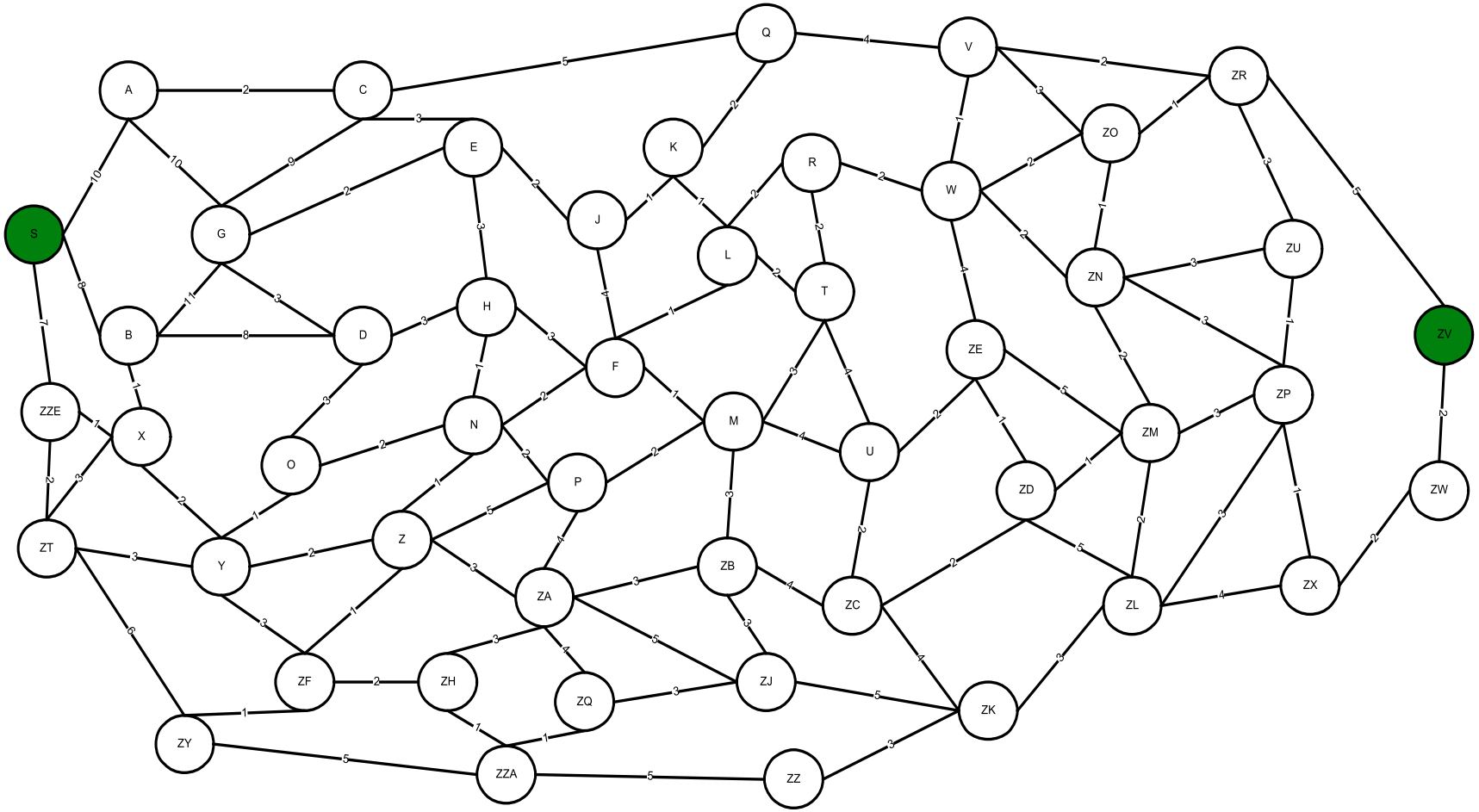
No	Lebar Beam	Simpul Tujuan ditemukan
1	1	Tidak
2	2	Ya
3	3	Ya
4	4	Ya
5	5	Ya
6	6	Ya
7	7	Ya
8	8	Ya
9	9	Ya
10	10	Ya

EMPAT PULUH SIMPUL



No	Lebar Beam	Simpul Tujuan ditemukan
1	1	Tidak
2	2	Tidak
3	3	Ya
4	4	Ya
5	5	Ya
6	6	Ya
7	7	Ya
8	8	Ya
9	9	Ya
10	10	Ya

LIMA PULUH SIMPUL



No	Lebar Beam	Simpul Tujuan ditemukan
1	1	Tidak
2	2	Ya
3	3	Ya
4	4	Ya
5	5	Ya
6	6	Ya
7	7	Ya
8	8	Ya
9	9	Ya
10	10	Ya

APLIKASI SIMULASI ANDROID

1 2

No	Lebar Beam	Jumlah 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 jalur hijau	Bank Mega	Ya
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

No	Lebar Beam	Jumlah Awal	Lokasi Tujuan	Rute ditemukan
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



KESIMPULAN

KESIMPULAN

- Algoritma pencarian beam dapat menemukan hasil pencarian rute tergantung dengan nilai heuristik dan lebar beamnya.
- Algoritma pencarian beam memiliki kemungkinan tidak menemukan solusi atau tujuan yang disebabkan oleh pemangkasan simpul yang tidak digunakan.
- Semakin besar lebar beam, maka kemungkinan solusi ditemukan semakin besar pula.
- Pada aplikasi simulasi di android, lokasi akan selalu ditemukan karena lokasi awal dan lokasi tujuan langsung berhubungan atau bertetangga.

SARAN

Aplikasi simulasi android lebih dikembangkan lagi dengan memperhatikan simpul-simpul yang terdapat pada *google map* untuk mencapai suatu lokasi yang ingin di tuju di kota Tasikmalaya karena dengan memperhatikan simpul-simpul yang terdapat pada *google map*, memungkinkan hasil pencarian dengan solusi yang berbeda, baik itu ditemukan atau tidak dengan menggunakan parameter yang berbeda-beda.

THANK YOU!!!

