

**SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF  
TERDEKAT DI KECAMATAN SUMBERSARI MENGGUNAKAN  
ALGORITMA DIJKSTRA**

**LAPORAN AKHIR**



Oleh  
**Selvi Sugiarto**  
**E31151598**

**PROGRAM STUDI MANAJEMEN INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
2018**

**SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF  
TERDEKAT DI KECAMATAN SUMBERSARI MENGGUNAKAN  
ALGORITMA DIJKSTRA**

**LAPORAN AKHIR**



Sebagai salah satu syarat untuk memperoleh gelar Ahli Madya ( A.Md )  
di Program Studi Manajemen Informatika  
Jurusan Teknologi Informasi

Oleh  
**Selvi Sugiarto**  
**E31151598**

**PROGRAM STUDI MANAJEMEN INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI JEMBER  
2018**


KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI JEMBER

SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF TERDEKAT  
DI KECAMATAN SUMBERSARI MENGGUNAKAN ALGORITMA  
DJKSTRA

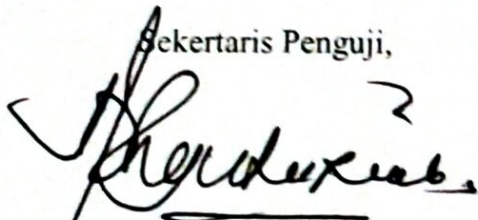
Selvi Sugiarto (E31151598)  
Telah Diuji pada Tanggal 9 Juli 2018  
Telah Dinyatakan Memenuhi Syarat

HALAMAN PENGESAHAN

Ketua Penguji,

  
Hendra Yufit Riskiawan, S.Kom, M.Cs  
NIP. 19830203 200604 1 003

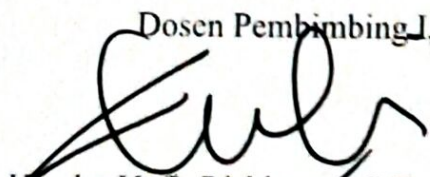
Sekretaris Penguji,

  
Wahyu Kurnia Dewanto, S.Kom, MT  
NIP. 19710408 200112 1 003

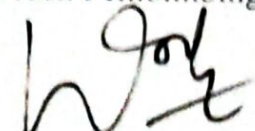
Anggota Penguji,

  
Taufiq Rizaldi, S.ST, MT  
NIK.19890329 201503 1 001

Dosen Pembimbing I,

  
Hendra Yufit Riskiawan, S.Kom, M.Cs  
NIP. 19830203 200604 1 003

Dosen Pembimbing II,

  
I Putu Dody Lesmana, ST, MT  
NIP. 19790921 200501 1 001

Menyetujui,

Ketua Jurusan Teknologi Informasi,

  
Wahyu Kurnia Dewanto, S.Kom, MT  
NIP. 19710408 200112 1 003

## **SURAT PERNYATAAN**

Saya yang bertandatangan di bawah ini:

Nama : Selvi Sugiarto

NIM : E31151598

Menyatakan dengan sebenar–benarnya bahwa segala pernyataan dalam laporan akhir saya yang berjudul “SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF TERDEKAT DI KECAMATAN SUMBERSARI MENGGUNAKAN ALGORITMA DIJKSTRA” merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam daftar pustaka di bagian akhir Laporan Akhir ini.

Jember, 17 Juli 2018

Selvi Sugiarto  
NIM. E3115158

## **HALAMAN PERSEMBAHAN**

Dengan mengucapkan syukur Alhamdulillah, karya sederhana ini teruntuk orang-orang terkasih:

1. Papa dan Mama tersayang, sebagai penyemangat dan penasehat yang telah membimbing saya menjadi orang yang lebih baik, selalu mendoakan saya. Terimakasih juga untuk semangatnya yang tidak membiarkan saya untuk menyerah dan menemani saya hingga larut malam dalam menyelesaikan tugas akhir ini.
2. Bapak Hendra Yufit Riskiawan, S.Kom, M.Cs dan Bapak I Putu Dody Lesmana, ST, MT terimakasih untuk bimbingan dan semangatnya selama penyelesaian tugas akhir ini.
3. Bapak WahyuKurniaDewanto,S.Kom, MT yang sudah motivasi dan selalu memberi semangat kepada saya
4. Bintang Cahyono alias Spiderman yang sudah menjadi sarana penunjang saya dalam mengerjakan tugas akhir dan memberi harapan baru ketika saya mulai menyerah.
5. Teman-teman yang tidak dapat saya sebutkan semua karena telah menemani dan saling berbagi, mau direpotkan oleh saya sehingga saya dapat menyelesaikan tugas akhir ini.

----- TerimaKasih -----

Almamaterku Tercinta

## **HALAMAN MOTTO**

"Tuhan Memberikan Apa Yang Kamu Butuhkan, Bukan Apa Yang Kamu  
Inginkan."

"Lakukan Segalanya Dengan Santai."

# **Sistem Informasi Penentuan Jalur Alternatif Terdekat Di Kecamatan Sumbersari Menggunakan Algoritma Dijkstra**

**Selvi Sugiarto**

Program Studi Manajemen Informatika

Jurusan Teknologi Informasi

## **ABSTRAK**

Sistem informasi penentuan jalur alternatif terdekat merupakan kegiatan input, proses dan output akan data jalan yang ada di Kecamatan Sumbersari seperti data nama jalan, data jarak dan data tingkat kemacetan setiap harinya. Sistem ini digunakan untuk menemukan jalur alternatif terdekat untuk menghemat waktu tempuh dan terhindar dari kemacetan. Kecamatan Sumbersari merupakan salah satu kecamatan terpadat yang ada di kota Jember, dengan luas wilayah 35.32 km<sup>2</sup> yang terdiri dari 7 kelurahan dan 33 lingkungan. Kepadatan jumlah penduduk di Kecamatan Sumbersari semakin diperparah dengan para pendatang yang berniat untuk bekerja ataupun belajar di wilayah ini. Banyak para pelajar dari luar kota bahkan dari luar pulau. Melihat permasalahan ini dibutuhkan sebuah sistem informasi yang dapat membantu para pengguna jalan menemukan jalur alternatif terdekat dan terhindar dari kemacetan menggunakan algoritma dijkstra yang diterapkan di jalan sekitar Kecamatan Sumbersari dengan menggunakan Netbeans 8.2.

Kata Kunci : *Sistem informasi, Algoritma Dijkstra, Netbeans 8.2*

**Sistem Informasi Penentuan Jalur Alternatif Terdekat Di Kecamatan  
Sumbersari Menggunakan Algoritma Dijkstra**

**Selvi Sugiarto**

Program Studi Manajemen Informatika

Jurusan Teknologi Informasi

**ABSTRACT**

The information system of the determination of the nearest alternative route is the input, process and output activity of the existing road data in Sumbersari Subdistrict such as road name, distance data and data of traffic jam everyday. This system is used to find the nearest alternative path to save the travel time and avoid congestion. Sumbersari District is one of the densest districts in Jember city, with an area of 35.32 km<sup>2</sup> which consists of 7 grass and 33 neighborhoods. Population density in Kecamatan Sumbersari has been exacerbated by migrants intending to work or study in the region. Many students from outside the city even from outside the island. Seeing this problem requires an information system that can help road users find the nearest alternative path and avoid congestion using the dijkstra algorithm applied on the road around Sumbersari Sub-district by using Netbeans 8.2.

**Keyword :** *Information System, Algorithm Dijkstra, Netbeans 8.2*



## **RINGKASAN**

**SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF TERDEKAT DI KECAMATAN SUMBERSARI MENGGUNAKAN ALGORITMA DIJKSTRA**, Selvi Sugiarto, NIM E31151598, Tahun 2018, Teknologi Informasi, Politeknik Negeri Jember, Bapak Hendra Yufit Riskiawan, S.Kom, M.Cs (pembimbing I) dan Bapak I Putu Dody Lesmana, ST, MT ( Pembimbing II )

Dalam kehidupan sehari-hari sering kali seseorang melakukan perjalanan dari suatu daerah ke daerah lain. Dan sangat lazim sekali apabila memilih jarak terdekat dari dua daerah tersebut untuk dilalui, karena dapat menghemat waktu, tenaga serta bahan bakar tentunya. Kesulitan menentukan jarak terpendek timbul karena terdapat banyak jalur yang ada pada tiap daerah karena dalam kenyataan dari daerah A ke daerah Z tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui . Algoritma Dijkstra merupakan algoritma yang dapat digunakan untuk menemukan jalur terpendek.

Kecamatan Sumbersari merupakan salah satu kecamatan terpadat yang ada di kota Jember, dengan luas wilayah 35.32 km<sup>2</sup> yang terdiri dari 7 kelurahan dan 33 lingkungan. Kepadatan jumlah penduduk di Kecamatan Sumbersari semakin diperparah dengan para pendatang yang berniat untuk bekerja ataupun belajar di wilayah ini. Hal ini semakin memperparah tingkat kemacetan di Kecamatan sumbersari. Ditambah lagi dengan banyaknya pedagang kaki lima yang berjualan di tengah-tengah aktivitas masyarakat, pedagang banyak menempati badan jalan semakin sempit untuk kendaraan bermotor. Melihat permasalahan ini maka dibuatlah suatu sistem informasi yang dapat membantu para pengguna jalan untuk menemukan jalur alternatif terdekat dan menghemat waktu tempuh.

## **PRAKATA**

Puji syukur penulis panjatkan kehadiran Allah S.W.T. atas berkat, rahmat dan karunia-Nya, maka penulisan karya tulis ilmiah berjudul “Sistem Informasi Penentuan Jalur Alternatif Terdekat Di Kecamatan Summersar Menggunakan Algoritma Dijkstra” dapat diselesaikan dengan baik.

Tulisan ini adalah laporan hasil penelitian yang dilaksanakan mulai tanggal 7 oktober sampai dengan 22 februari 2017 bertempat di Politeknik Negeri Jember, sebagai salah satu syarat untuk memperoleh gelar Ahli Madya (AMD) di Program Studi Manajemen Informatika Jurusan Teknologi Informasi.

Penulis menyampaikan penghargaan dan ucapan terimakasih yang sebesar-besarnya sebagai berikut:

1. Direktur Politeknik Negeri Jember.
2. Wahyu Kurina Dewanto, S.Kom, M.T, selaku Ketua Jurusan Teknologi Informasi.
3. Dwi Putro Sawro S, S.Kom, M.Kom, Ketua Program Studi Manajemen Informatika.
4. Hendra Yufit Riskiawan, S.Kom, M.Cs, selaku Pembimbing I.
5. I Putu Dody Lesmana, ST, MT, selaku Pembimbing II.
6. Rekan-rekan dan semua pihak yang telah ikut membantu dalam pelaksanaan penelitian dan penulisan laporan ini.

Laporan Karya Tulis Ilmiah ini masih kurang sempurna, mengharapkan kritik dan saran yang sifatnya membangun perbaikan dimasa mendatang. Semoga tulisan ini bermanfaat.

Jember, 17 Juli 2018

Selvi Sugiarto

E31151598



**PERNYATAAN PERSETUJUAN PUBLIKASI  
KARYA ILMIAH UNTUK KEPENTINGAN  
AKADEMIS**

Yang bertandatangan di bawah ini, saya:

**Nama : Selvi Sugiarto**  
**NIM : E31151598**  
**Program Studi : Manajemen Informatika**  
**Jurusan : Teknologi Informasi**

Demi pengembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada UPT. Perpustakaan Politeknik Negeri Jember, Hak Bebas Royalti Non Eksklusif (Non-Exclusive Royalty Free Right) atas Karya Ilmiah berupa Laporan Tugas Akhir yang berjudul:

**”SISTEM INFORMASI PENENTUAN JALUR ALTERNATIF  
TERDEKAT DI KECAMATAN SUMBERSARI MENGGUNAKAN  
ALGORITMA DIJKSTRA”**

Dengan Hak Bebas Royalti Non-Eksklusif ini UPT. Perpustakaan Politeknik Negeri Jember berhak menyimpan, mengalihkan media atau format, mengelola dalam bentuk Pangkalan Data (DataBase), mendistribusikan karya dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta.

Saya bersedia untuk menanggung secara pribadi tanpa melibatkan pihak Politeknik Negeri Jember, segala bentuk tuntutan hukum yang timbul atas Pelanggaran Hak Cipta dalam Karya Ilmiah ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

**Dibuat di : Jember**  
**Pada Tanggal : 17 Juli 2018**  
**Yang Menyatakan,**

**Nama : Selvi Sugiarto**  
**NIM : E31151598**

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL .....</b>	<b>ii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iii</b>
<b>SURAT PERNYATAAN .....</b>	<b>iv</b>
<b>HALAMAN PERSEMBAHAN .....</b>	<b>v</b>
<b>HALAMAN MOTTO .....</b>	<b>vi</b>
<b>ABSTRAK .....</b>	<b>vii</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>RINGKASAN .....</b>	<b>ix</b>
<b>PRAKATA .....</b>	<b>x</b>
<b>PERNYATAAN PERSETUJUAN .....</b>	<b>xi</b>
<b>DAFTAR ISI .....</b>	<b>xii</b>
<b>DAFTAR TABEL .....</b>	<b>xv</b>
<b>DAFTAR GAMBAR .....</b>	<b>xvi</b>
<b>DAFTAR KODE PROGRAM .....</b>	<b>xvii</b>
<b>BAB 1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
<b>BAB 2. TINJAUAN PUSTAKA .....</b>	<b>4</b>
2.1 Definisi Sistem Informasi .....	4
2.1.1 Sistem .....	4
2.1.2 Informasi .....	4
2.1.3 Sistem Informasi .....	4
2.2 Definisi Jalur Alternatif .....	5
2.2.1 Jalur .....	5

2.2.2	Alternatif .....	5
2.2.3	Jalur Alternatif .....	5
2.3	Teori Graf .....	5
2.3.1	Jenis-jenis Graf .....	6
2.4	Algoritma Dijkstra .....	8
2.4.1	Implementasi Dijkstra .....	8
2.5	UML .....	9
2.5.1	Use Case Diagram .....	10
2.5.2	Class Diagram .....	11
2.5.3	State Chart .....	13
2.6	Klasifikasi Jalan .....	14
2.6.1	Jalan Arteri .....	14
2.6.2	Jalan Kolektor .....	15
2.6.3	Jalan Lokal .....	16
2.6.4	Jalan Lingkungan .....	16
2.7	Penelitian yang Mendahului .....	17
2.7.1	Penggunaan Algoritma Dijkstra Untuk Pencarian Jalur Tercepat dan Jalur Terpendek .....	17
2.7.2	Perancangan Sistem Informasi Geografis Penentuan Jalur Jalan Optimum Menggunakan Metode Dijkstra Kota Yogyakarta Berbasis Web.....	17
2.8	State Of The Art .....	18
<b>BAB 3.</b>	<b>METODE KEGIATAN.....</b>	<b>20</b>
3.1	Tempat dan Waktu .....	20
3.2	Bahan dan Alat.....	20
3.2.1	Bahan .....	20
3.2.2	Alat .....	20
3.3	Metode Kegiatan .....	21

<b>BAB 4. HASIL DAN PEMBAHASAN.....</b>	<b>23</b>
4.1 Communication .....	23
4.2 Quick Plan .....	27
4.3 Modelling Quick Design .....	27
4.3.1 Definisi Aktor .....	28
4.3.2 Definisi Use Case .....	28
4.3.3 Skenario Use Case .....	29
4.4 Pembuatan Aplikasi .....	33
4.4.1 Iterasi 1 .....	33
4.4.2 Iterasi 2 .....	36
4.4.3 Iterasi 3 .....	39
4.5 Perhitungan Algoritma Dijkstra .....	46
<b>BAB 5. KESIMPULAN DAN SARAN .....</b>	<b>50</b>
5.1 Kesimpulan .....	50
5.2 Saran.....	50
<b>DAFTAR PUSTAKA .....</b>	<b>51</b>
<b>LAMPIRAN .....</b>	<b>52</b>

## DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-simbol Use Case Diagram.....	10
Tabel 2.2 Simbol-simbol Class Diagram .....	11
Tabel 2.3 Simbol-simbol Statechart Diagram .....	14
Tabel 2.4 State Of The Art.....	18
Tabel 4.1 Data Jalan .....	23
Tabel 4.2 Kondisi Lalu Lintas.....	26
Tabel 4.3 Definisi Aktor .....	28
Tabel 4.4 Definisi Use Case .....	28
Tabel 4.5 Skenario Use Case Melihat Peta Kecamatan Sumbersari .....	29
Tabel 4.6 Skenario Use Case Melihat Graf Kecamatan Sumbersari .....	30
Tabel 4.7 Skenario Use Case Menentukan Titik Awal Keberangkatan .....	30
Tabel 4.8 Skenario Use Case Menentukan Titik Akhir Tujuan .....	30
Tabel 4.9 Skenario Use Case Mendapat Nilai Jarak .....	31
Tabel 4.10 Skenario Use Case Algoritma Dijkstra .....	31
Tabel 4.11 Skenario Use Case Menentukan Titik 1 Kemacetan .....	31
Tabel 4.12 Skenario Use Case Menentukan Titik 2 Kemacetan .....	32
Tabel 4.13 Skenario Use Case Mendapat Tingkat Kemacetan .....	32
Tabel 4.14 Skenario Use Case Mendapatkan Informasi Jalur Terpendek .....	32
Tabel 4.15 Pengujian Black Box Melihat Peta Kecamatan Sumbersari .....	36
Tabel 4.16 Pengujian Black Box Melihat Graf Kecamatan Sumbersari .....	39
Tabel 4.17 Pengujian Black Box Informasi Jalur Alternatif Terdekat .....	45
Tabel 4.18 Matriks .....	47
Tabel 4.19 Perhitungan Manual Lintasan Terpendek .....	47
Tabel 4.20 Hasil Perhitungan Manual .....	49

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Graf Sederhana .....	6
Gambar 2.2 Graf Ganda .....	7
Gambar 2.3 Graf Semu .....	7
Gambar 2.4 Graf Tidak Berarah .....	7
Gambar 2.5 Graf Berarah .....	8
Gambar 2.6 Algoritma Dijkstra .....	9
Gambar 2.7 Contoh Use Case Diagram pendaftaran internet banking BCA.	10
Gambar 2.2 Contoh Class Diagram transaksi pembelian barang.....	12
Gambar 2.3 Contoh Statechart Diagram .....	13
Gambar 3.1 Metode Pengembangan Prototype Menurut Pressman .....	21
Gambar 4.1 Usecase Diagram .....	27
Gambar 4.2 Melihat Peta Kecamatan Sumbersari .....	34
Gambar 4.3 Design Mockup Melihat Peta Kecamatan Sumbersari .....	34
Gambar 4.4 Form Peta Kecamatan Sumbersari .....	35
Gambar 4.5 Melihat Graf Kecamatan Sumbersari .....	37
Gambar 4.6 Design Mockup Melihat graf Kecamatan Sumbersari .....	37
Gambar 4.7 Form Graf Kecamatan Sumbersari .....	38
Gambar 4.8 Informasi Alternatif Jalur Terdekat .....	40
Gambar 4.9 Design Mockup Informasi Alternatif Jalur Terdekat .....	41
Gambar 4.10 Form Informasi Alternatif Jalur Terdekat .....	44
Gambar 4.11 Graf Kecamatan Sumbersari .....	46



## DAFTAR KODE PROGRAM

	Halaman
Kode Program 4.1 Melihat Peta Kecamatan Summersari .....	34
Kode Program 4.2 Melihat Graf Kecamatan Summersari .....	37
Kode Program 4.3 Informasi Jalur Alternatif Terdekat .....	41

## **BAB 1. PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Dalam kehidupan sehari-hari sering kali seseorang melakukan perjalanan dari suatu daerah ke daerah lain. Dan sangat lazim sekali apabila memilih jarak terdekat dari dua daerah tersebut untuk dilalui, karena dapat menghemat waktu, tenaga serta bahan bakar tentunya. Kesulitan menentukan jarak terpendek timbul karena terdapat banyak jalur yang ada pada tiap daerah karena dalam kenyataan dari daerah A ke daerah Z tidak hanya memiliki satu jalur saja, banyak sekali jalur yang dapat dilalui .

Algoritma Dijkstra merupakan algoritma yang dapat digunakan untuk menemukan jalur terpendek, algoritma ini memiliki beberapa kelebihan dibanding algoritma lain, yaitu cepat dan optimal, serta cara perhitungan yang secara bertahap dapat memberikan hasil yang akurat. Jadi algoritma ini sangat cocok untuk diterapkan dalam pencarian jalur terdekat.

Kemacetan adalah situasi atau keadaan tersendatnya atau bahkan terhentinya lalu lintas yang disebabkan oleh banyaknya jumlah kendaraan melebihi kapasitas jalan. Kemacetan banyak terjadi di kota-kota besar, terutama yang tidak mempunyai transportasi publik yang baik atau memadai ataupun juga tidak seimbangnya kebutuhan jalan dengan kepadatan penduduk, misalnya kota Jember.

Kecamatan Sumbersari merupakan salah satu kecamatan terpadat yang ada di kota Jember, dengan luas wilayah 35.32 km<sup>2</sup> yang terdiri dari 7 kelurahan dan 33 lingkungan. Kepadatan jumlah penduduk di Kecamatan Sumbersari semakin diperparah dengan para pendatang yang berniat untuk bekerja ataupun belajar di wilayah ini. Banyak para pelajar dari luar kota bahkan dari luar pulau, bermigrasi ke Kota Jember karena tingkat pendidikan kota Jember cukup bagus. Transportasi merupakan salah satu akses yang dapat digunakan untuk bepergian, banyak masyarakat Jember memilih menggunakan alat transportasi pribadi, dll. Hal ini semakin memperparah tingkat kemacetan di Kecamatan sumbersari. Ditambah lagi dengan banyaknya pedagang kaki lima yang berjualan di tengah-tengah

aktivitas masyarakat, pedagang banyak menempati badan jalan semakin sempit untuk kendaraan bermotor.

Penurunan kondisi jalan raya juga menjadi salah satu penyebab kemacetan yang merupakan dampak dari kemampuan pemeliharaan dan rehabilitasi jalan yang terbatas, laju perubahan jalan yang lebih lambat dari laju kerusakan jalan, perubahan volume lalu lintas maupun intensitas bahan yang terus meningkat.

Banyak hal yang dapat ditimbulkan dari kemacetan, antara lain dapat meningkatkan waktu tempuh, biaya transportasi, gangguan yang serius bagi gangguan produk, penurunan produktivitas kerja, dan pemanfaatan energi kerja yang sia-sia. Kemacetan juga memberikan dampak yang serius bagi penurunan tingkat kesehatan. Dengan melihat kondisi tersebut, maka penulis mengambil judul:

**“ Sistem Informasi Penentuan Jalur Alternatif Terdekat di Kecamatan Sumbersari Menggunakan Algoritma Dijkstra ”**

## **1.2 Perumusan Masalah**

Dari latar belakang tersebut, rumusan masalah yang perlu dikaji lebih dalam adalah bagaimana menentukan jalur terpendek untuk menuju tempat tujuan agar dapat menghemat waktu dan terhindar dari kemacetan menggunakan Algoritma Dijkstra.

## **1.3 Batasan Masalah**

Agar penulisan tugas akhir ini tidak menyimpang dan mengambang dari tujuan yang semula direncanakan sehingga mempermudah mendapatkan data dan informasi yang diperlukan, maka penulis menetapkan batasan-batasan sebagai berikut:

- a. Jalur yang menjadi objek adalah jalan kolektor primer yang dapat dilalui kendaraan roda 4.
- b. Sistem ini hanya mencari dan memberikan solusi untuk jalur yang berada di wilayah administratif Kecamatan Sumbersari.
- c. Metode yang digunakan Algoritma Dijkstra.

#### **1.4 Tujuan**

Tujuan dari Tugas Akhir ini adalah membuat sebuah program yang dapat membantu para pengguna jalan untuk menempuh jalur terdekat yang lebih menghemat waktu tempuh di jalur yang berada di jalan sekitar wilayah administratif Kecamatan Summersari.

#### **1.5 Manfaat**

Adapun manfaat dari Tugas Akhir ini antara lain sebagai berikut:

- a. Bagi penulis, berharap penelitian ini akan mampu menambah wawasan serta lebih memahami tentang Algoritma Dijkstra.
- b. Bagi almamater, penelitian ini dapat menambah referensi yang ada dan dapat digunakan oleh semua pihak yang membutuhkan. Penelitian ini, juga diharapkan dapat memberikan sumbangan pemikiran terutama dalam ilmu Algoritma Dijkstra.
- c. Bagi pembaca, penelitian ini diharapkan memberikan sumbangan kepustakaan yang merupakan informasi tambahan yang berguna bagi pembaca dan dapat memberikan sumbangan pemikiran bagi pihak-pihak yang mempunyai permasalahan yang sama atau ingin mengadakan penelitian lebih lanjut.

## **BAB 2. TINJAUAN PUSTAKA**

### **2.1 Definisi Sistem Informasi**

#### **2.1.1 Sistem**

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem (Hutahaean, 2014).

#### **2.1.2 Informasi**

Informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi berguna untuk mengambil keputusan karena informasi menurunkan ketidakpastian (atau meningkatkan pengetahuan). Informasi menjadi penting karena berdasarkan informasi itu para pengelola dapat mengetahui kondisi objektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara-cara tertentu (Muhlishudin dan Oktafianto, 2016).

#### **2.1.3 Sistem Informasi**

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan (Hutahaean, 2014).

Informasi dalam lingkup sistem informasi memiliki beberapa ciri, yaitu:

- a. Baru, informasi yang didapat sama sekali baru dan segar bagi penerima.
- b. Tambahan, informasi dapat memperbaharui atau memberikan tambahan pada informasi yang telah ada.
- c. Korektif, informasi dapat menjadi suatu koreksi atas informasi sebelumnya.
- d. Penegas, informasi dapat mempertegas informasi yang telah ada.

## 2.2 Definisi Jalur Alternatif

### 2.2.1 Jalur

Jalur lalu lintas adalah keseluruhan bagian perkerasan jalan yang diperuntukkan untuk lalu lintas kendaraan. Jalur lalu lintas terdiri dari beberapa lajur (lane) kendaraan.

### 2.2.2 Alternatif

Alternatif adalah satu dari dua atau lebih cara untuk mencapai tujuan atau akhir yang sama. Alternatif tidak harus menjadi pengganti dekat untuk pilihan pertama (alternatif lain), atau harus memecahkan masalah dengan cara tertentu.

### 2.2.3 Jalur Alternatif

Jalur Alternatif adalah satu dari dua atau lebih jalur yang ada, kemudian dipilih sebagai jalur kendaraan yang akan dilewati dengan pilihan yang terdiri dari beberapa rumusan yang dapat dijadikan sebuah solusi untuk memilih jalur kendaraan.

## 2.3 Teori Graf

Dalam matematika dan ilmu komputer, teori graf adalah cabang kajian yang mempelajari sifat-sifat graf. Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (vertex atau node) yang terhubung oleh sisi (edge) atau busur (arc). Biasanya graf digambarkan sebagai kumpulan titik-titik (melambangkan simpul). Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama. Sisi yang demikian dinamakan gelang (loop).

Sebuah struktur graf bisa dikembangkan dengan memberi bobot pada tiap sisi. Graf berbobot dapat digunakan untuk melambangkan banyak konsep berbeda. Suatu graph  $G$  dapat dinyatakan sebagai:

$$G = \langle V, E \rangle$$

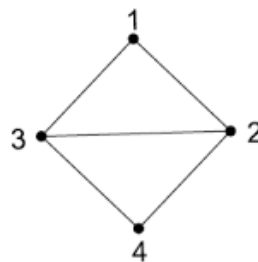
Graph  $G$  terdiri atas himpunan  $V$  yang berisikan simpul pada graf tersebut dan himpunan dari  $E$  yang berisi sisi pada graf tersebut. Himpunan  $E$  dinyatakan sebagai pasangan dari simpul yang ada dalam  $V$ .

### 2.3.1 Jenis-Jenis Graf

Graf dapat dikelompokkan berdasarkan ada tidaknya edgenya yang paralel atau loop, jumlah vertexnya, berdasarkan ada tidaknya arah pada edgenya, ada tidaknya bobot pada edgenya, atau ada tidaknya hubungan dengan graf lain. Berikut ini adalah jenis graf berdasarkan ada tidaknya edge yang paralel atau loop.

#### a. Graf Sederhana

Graf Sederhana adalah graf yang tidak mempunyai edge ganda dan atau loop, loop adalah edge yang menghubungkan sebuah vertex dengan dirinya sendiri. Berikut adalah contoh graf sederhana :

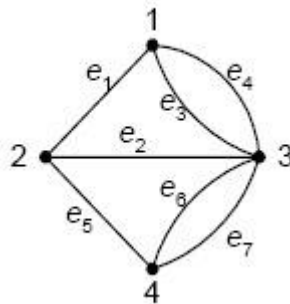


Gambar 2.1 Graf Sederhana

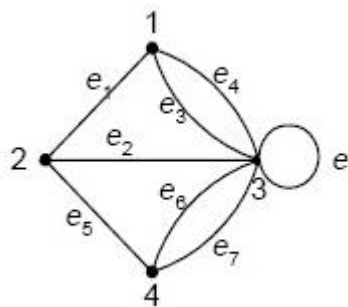
#### b. Graf Tak-Sederhana

Graf Tak-Sederhana adalah graf yang tidak memiliki edges ganda dan atau loop. Graf tak-sederhana dapat dibagi menjadi dua, yaitu:

- 1) Graf Ganda (multigraph), adalah graf yang mengandung edge ganda. Sisi ganda yang menghubungkan sepasang vertexs biasa lebih dari satu buah.
- 2) Graf semu (pseudograph), adalah graf yang mempunyai loop, terasuk juga graf yang mempunya edge dan loop ganda karena itu graf semu lebih umum daripada graf ganda, karena graf semu edgenya dapat terhubung dengan dirinya sendiri.



Gambar 2.2 Graf Ganda

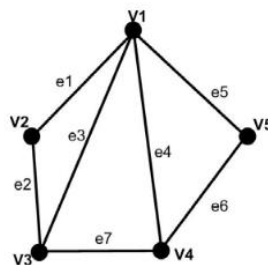


Gambar 2.3 Graf Semu

Selain ada tidaknya edge yang paralel atau loop, graf dapat juga dikelompokkan berdasarkan orientasi arah atau panah.

a. Graf Tak-Berarah (undirect-graph)

Graf tak berarah adalah graf yang edgenya tidak mempunyaorientasi arah atau panah. Pada graf ini, urutan pasangan verteks yang dihubungkan oleh edge tidak diperhatikan. Jadi  $(v_j, v_k) = (v_k, v_j)$  adalah edge yang sama.

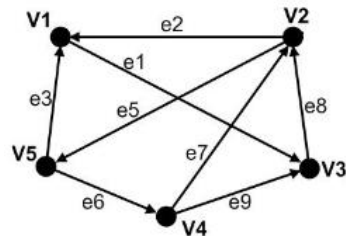


Gambar 2.4 Graf Tidak Berarah



b. Graf Berarah (direct graph atau digraph)

Graf berarah adalah graf yang disetiap edgenya memiliki orietasi arah atau panah. Pada graf berarah  $(v_j, v_k) \neq (v_k, v_j)$ .



Gambar 2.5 Graf Berarah

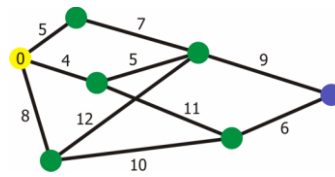
## 2.4 Algoritma Dijkstra

Algoritma ini ditemukan oleh Edsger Wybe Dijkstra dan dipublikasikan pada tahun 1959 pada sebuah jurnal Numerische Mathematik yang berjudul “A Note on Two Problems in Connexion with Graps”. Algoritma Dijkstra adalah salah satu algoritma untuk memecahkan masalah “single source shortest path”. Algoritma ini sering digambarkan sebagai algoritma greedy (tamak).

Dijkstra merupakan salah satu varian bentuk algoritma populer dalam pemecahan persoalan terkait masalah optimasi pencarian lintasan terpendek sebuah lintasan yang mempunyai panjang minimum dari verteks  $a$  ke  $z$  dalam graph berbobot, bobot tersebut adalah bilangan tersebut adalah bobot positif jadi tidak dapat dilalui oleh node negatif. Namun jika terjadi demikian, maka penyalasain yang diberikan infinity (tak hingga). Pada Algoritma Dijkstra, node digunakan karena algoritma Dijkstra menggunakan graph berarah untuk penentuan rute lintasan terpendek.

### 2.4.1 Implementasi Dijkstra

Algoritma ini bertujuan menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Pada algoritma dijkstra pemecahan masalah diperuntukkan untuk sebuah Graph  $G = (G, E)$  yang berbobot non negatif. Diasumsikan  $w(i, j) \geq 0$  untuk masing-masing edge  $(i, j) \in E$ .



Gambar 2.6 Algoritma Dijkstra

Pertama-tama tentukan titik mana yang akan menjadi node awal, lalu beri bobot jarak pada node pertama ke node terdekat satu persatu, Dijkstra akan melakukan pengembangan pencarian dari satu titik ke titik lain dan ketitik selanjutnya tahap demi tahap inilah urutan logika dari Algoritma Dijkstra:

- Beri nilai bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada node awal dan nilai tak hingga terhadap node lain (belum terisi).
- Set semua node “Belum Terjamah” dan set node awal sebagai “Node Keberangkatan”.
- Dari No Keberangkatan, pertimbangkan node tatangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke node C berjarak 2, maka jarak ke C melewati B menjadi  $6+2=8$ . Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
- Saat kita selesai mempertimbangkan setiap jarak terhadap node tetangga, tandai node yang telah terjamah sebagai “Node Terjamah”. Node terjamah tidak akan pernah dicek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
- Set “Node Belum Terjamah” dengan jarak terkecil (dari node keberangkatan) sebagai “Node Keberangkatan” selanjutnya dan lanjutkan kembali ke step 3.

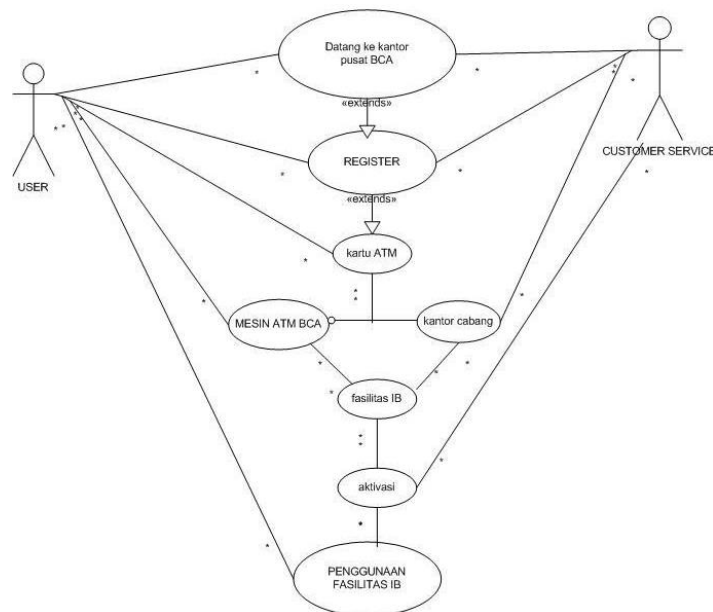
## 2.5 UML

Menurut S, Rosa A dan M. Salahuddin (2016:9) pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisai bahasa pemodelan untuk membangun sebuah perangkat lunak yang menggunakan teknik

pemrograman berorientasi objek, yaitu Unified Modeling Language (UML). Teknik UML muncul karena adanya kebutuhan pemodelan visual untuk memperinci, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan sebuah bahasa visual yang diciptakan sebagai pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.


### 2.5.1 Use Case Diagram

Menurut S, Rosa A dan M. Salahuddin (2016:9) mengungkapkan “use case atau diagram use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan di buat”.

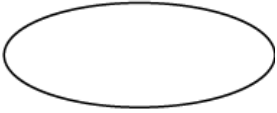
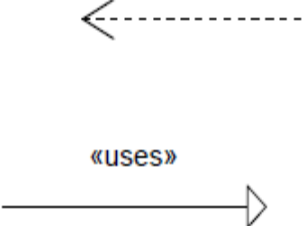




Gambar 2.7 Contoh Use Case Diagram pendaftaran internet banking BCA

Tabel 2.1 Simbol-simbol Use Case Diagram ( Rosa A. S – M. Shalahuddin, 2016)

Simbol	Nama Simbol	Keterangan
	Aktor	Orang, proses atau sistem lain yang berinteraksidengan sistem informasi yang dibuat. Jadi, meskipun simbol dari aktor adalah gambar orang, tetapi belum tentu aktor merupakan orang. Biasanya dinyatakan menggunakan kata benda diawal nama aktor.

Tabel 2.1 Simbol-simbol Use Case Diagram ( Rosa A. S – M. Shalahuddin, 2016) (Lanjutan)

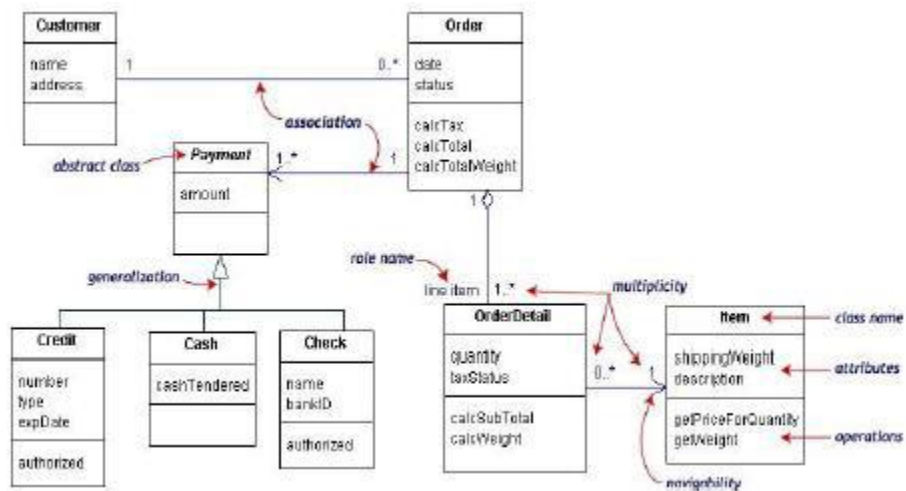
Simbol	Nama Simbol	Keterangan
	Use Case	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja awal frase nama use case.
	Include dan Uses	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini. Include berarti use case yang ditambahkan akan selalu dipanggil saat use case tambahan di jalankan.
	Asosiasi/ Association	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
	Generalisasi	Hubungan generalisasi dan spesialisasi antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

### 2.5.2 Class Diagram

Menurut S, Rosa A dan M. Salahuddin (2016:9) mengungkapkan “Diagram class atau class diagram menggambarkan struktur sistem dari segi

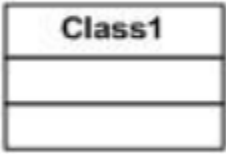
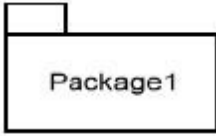
pedefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Berikut contoh class diagram.

Diagram kelas dibuat agar pembuat program membuat kelas sesuai rancangan didalam didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

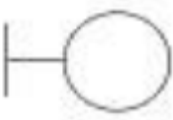

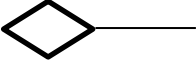
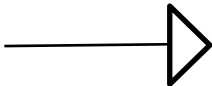


Gambar 2.2 Contoh Class Diagram transaksi pembelian barang.

Tabel 2.2 Simbol-simbol class diagram ( Rosa A. S – M. Shalahuddin, 2016)

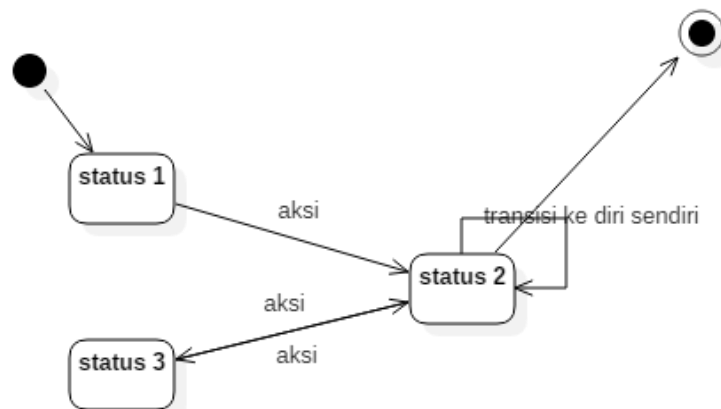
Simbol	Nama Simbol	Deskripsi
	Class/ Kelas	Menggambarkan sesuatu yang mengkapsulkan informasi di kelas, menampung nama kelas, atribut dan method.
	Package	Digunakan untuk mengelompokkan kelas-kelas yang mempunyai persamaan.
1, 1...*, 0...1	Asosiasi	Asosiasi yang menghubungkan class dengan class multiplicity.

Tabel 2.2 Simbol-simbol class diagram ( Rosa A. S – M. Shalahuddin, 2016) (Lanjutan)

Simbol	Nama Simbol	Deskripsi
	Boundary Class	Menggambarkan Class yang menjadi antar muka aktor dengan sistem.
 Nama_interface	Antarmuka / Interface	Sama dengan konsep interface dalam pemrograman berorientasi objek.
	Aggregation	Menggambarkan suatu class terdiri dari class lain atau suatu class adalah bagian dari class lain.
	Generalization	Merupakan sebuah taxonomic relationship antara class yang lebih umum dengan class yang lebih khusus.



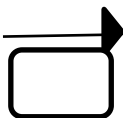
### 2.5.3 Statechart Diagram

Menurut S, Rosa A dan M. Salahuddin (2016:9), statechart diagram atau dalam Bahasa Indonesia disebut dengan diagram mesin status atau sering disebut juga diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.



Gambar 2.3 Contoh Statechart Diagram

Tabel 2.3 Simbol-simbol statechart diagram (Rosa A. S – M. Shalahuddin, 2016)

Simbol	Nama Simbol	Keterangan
	Start / Satatus Awal	State atau initial state adalah keadaan awal pada saat sistem mulai hidup.
	End / Satatus Akhir	End atau final state adalah keadaan akhir dari dasar hidup suatu sistem.
	Event State	State atau status adalah keadaan sistem pada waktu tertentu, state spat berubah jika ada event teterntu yang meniru perubahan tersebut.

## 2.6 Klasifikasi Jalan

Jalan adalah seluruh bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukan bagi lalu lintas umum, yang berada pada permukaan tanah, di atas permukaan tanah, di bawah permukaan tanah dan/atau air serta di atas permukaan air, kecuali jalan rel dan jalan kabel (UU No. 22 Tahun 2009). Jalan dapat diklasifikasikan berdasarkan fungsi, kelas dan statusnya. Dan apabila merujuk pada peraturan Peraturan Pemerintah UU No. 22 Tahun 2009, maka klasifikasi jalan berdasarkan fungsinya dapat dibedakan menjadi :

### 2.6.1 Jalan Arteri

Jalan arteri adalah jalan umum yang melayani angkutan utama dengan ciri-ciri perjalanan jarak jauh, kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi secara efisien.

#### 2.6.1.1 Jalan arteri primer

Jalan arteri primer sebagaimana dimaksud adalah jalan yang menghubungkan secara berdaya guna antara pusat kegiatan nasional atau antara pusat kegiatan nasional dengan pusat kegiatan wilayah. Sistem jaringan jalan primer disusun berdasarkan rencana tata ruang dan pelayanan distribusi barang dan jasa untuk pengembangan semua wilayah di tingkat nasional, dengan menghubungkan semua simpul jasa distribusi yang berwujud pusat-pusat kegiatan. Jalan arteri primer mempunyai kapasitas yang lebih besar dari volume lalu lintas rata-rata. Pada jalan arteri primer lalu lintas jarak jauh tidak boleh

terganggu oleh lalu lintas ulang alik, lalu lintas lokal, dan kegiatan lokal. Jalan arteri primer didesain berdasarkan kecepatan rencana paling rendah 60 (enam puluh) kilometer per jam dengan lebar badan jalan paling sedikit 11 (sebelas) meter.

#### 2.6.1.2 Jalan arteri sekunder

Jalan arteri sekunder adalah jalan yang menghubungkan kawasan primer dengan kawasan sekunder kesatu, kawasan sekunder kesatu dengan kawasan sekunder kesatu, atau kawasan sekunder kesatu dengan kawasan sekunder kedua. Jalan arteri sekunder melayani angkutan utama dengan ciri perjalanan jarak jauh kecepatan rata-rata tinggi, dan jumlah jalan masuk dibatasi, dengan peranan pelayanan jasa distribusi untuk masyarakat dalam kota. Didaerah perkotaan disebut sebagai jalan protokol. Jalan arteri sekunder didesain berdasarkan kecepatan rencana paling rendah 30 (tiga puluh) kilometer per jam dengan lebar badan jalan paling sedikit 11 (sebelas) meter.

#### 2.6.2 Jalan Kolektor

Jalan kolektor adalah jalan umum yang melayani angkutan pengumpul atau pembagi dengan ciri perjalanan jarak sedang, kecepatan rata-rata sedang dan jumlah jalan dibatasi.

##### 2.6.2.1 Jalan kolektor primer

Jalan kolektor primer sebagaimana dimaksud adalah jalan yang menghubungkan secara berdaya guna antara pusat kegiatan nasional dengan pusat kegiatan lokal, antarpusat kegiatan wilayah, atau antara pusat kegiatan wilayah dengan pusat kegiatan lokal. Jalan kolektor primer yang memasuki kawasan perkotaan dan/atau kawasan pengembangan perkotaan tidak boleh terputus. Jalan kolektor primer didesain berdasarkan kecepatan rencana paling rendah 40 (empat puluh) kilometer per jam dengan lebar badan jalan paling sedikit 9 (sembilan) meter.

##### 2.6.2.2 Jalan kolektor sekunder

Jalan kolektor sekunder adalah jalan yang menghubungkan kawasan sekunder kedua dengan kawasan sekunder kedua atau kawasan sekunder kedua dengan kawasan sekunder ketiga. Jalan kolektor sekunder adalah jalan yang



melayani angkutan pengumpulan atau pembagi dengan ciri-ciri perjalanan jarak sedang, kecepatan rata-rata sedang, dan jumlah jalan masuk dibatasi, dengan peranan pelayanan jasa distribusi untuk masyarakat di dalam kota. Jalan kolektor sekunder didesain berdasarkan kecepatan rencana paling rendah 20 (dua puluh) kilometer per jam dengan lebar badan jalan paling sedikit 9 (sembilan) meter.

### 2.6.3 Jalan Lokal

Merupakan jalan umum yang berfungsi melayani angkutan setempat dengan ciri perjalanan jarak dekat, kecepatan rata-rata rendah, dan jumlah jalan masuk tidak dibatasi.

#### 2.6.3.1 Jalan lokal primer

Jalan lokal primer adalah jalan yang menghubungkan secara berdaya guna pusat kegiatan nasional dengan pusat kegiatan lingkungan, pusat kegiatan wilayah dengan pusat kegiatan lingkungan, antarpusat kegiatan lokal, atau pusat kegiatan lokal dengan pusat kegiatan lingkungan, serta antarpusat kegiatan lingkungan. Jalan lokal primer didesain berdasarkan kecepatan rencana paling rendah 20 (dua puluh) kilometer per jam dengan lebar badan jalan paling sedikit 7,5 (tujuh koma lima) meter.

#### 2.6.3.2 Jalan lokal sekunder

Jalan lokal sekunder adalah jalan yang menghubungkan kawasan sekunder kesatu dengan perumahan, kawasan sekunder kedua dengan perumahan, kawasan sekunder ketiga dan seterusnya sampai ke perumahan. Jalan lokal sekunder didesain berdasarkan kecepatan rencana paling rendah 10 (sepuluh) kilometer per jam dengan lebar badan jalan paling sedikit 7,5 (tujuh koma lima) meter.

### 2.6.4 Jalan Lingkungan

Merupakan jalan umum yang berfungsi melayani angkutan lingkungan dengan ciri perjalanan jarak pendek, dan kecepatan rata-rata rendah.

#### 2.6.4.1 Jalan lingkungan primer

Jalan lingkungan primer sebagaimana dimaksud adalah jalan yang menghubungkan antarpusat kegiatan di dalam kawasan perdesaan dan jalan di dalam lingkungan kawasan perdesaan. Jalan lingkungan primer didesain berdasarkan kecepatan rencana paling rendah 15 (lima belas) kilometer per jam

dengan lebar badan jalan paling sedikit 6,5 (enam koma lima) meter. 4.2 Jalan lingkungan sekunder

Jalan lingkungan sekunder sebagaimana dimaksud adalah jalan yang menghubungkan antarpersil dalam kawasan perkotaan. Jalan lingkungan sekunder didesain berdasarkan kecepatan rencana paling rendah 10 (sepuluh) kilometer per jam dengan lebar badan jalan paling sedikit 6,5 (enam koma lima) meter. Jalan lingkungan sekunder yang tidak diperuntukkan bagi kendaraan bermotor beroda 3 (tiga) atau lebih harus mempunyai lebar badan jalan paling sedikit 3,5 (tiga koma lima) meter.

## **2.7 Penelitian yang Mendahului**

2.6.1 Imron Fauzi, 2011 pernah meneliti “Penggunaan Algoritma Dijkstra Untuk Pencarian Jalur Tercepat dan Jalur Terpendek (Studi Kasus Pada Jalan Raya antara Wilayah Blok M dan Kota)” hasil dari penelitian ini menjelaskan bahwa di Jakarta kemacetan selalu terjadi setiap hari kerja, khususnya jalan raya antara blok M dan kota. Banyak pengguna jalan tidak mengetahui jalur lain yang dapat diakses untuk menghindari kemacetan dan menemukan jalur terpendek untuk menghemat waktu. Pada pengembangan sistem ini penulis menggunakan metode spiral model yaitu metode pengembangan evolusioner yang menyatukan sifat iterasi dari prototyping dengan kontrol dan aspek sistematis dari model sekuensial atau waterfall model. Solusi dari penelitian ini adalah pembuatan sebuah aplikasi berbasis web yang bisa memberikan jalur alternatif terdekat dengan menggunakan algoritma Dijkstra.

2.6.2 Muhammad Andri, 2010 pernah meneliti “Perancangan Sistem Informasi Geografis Penentuan Jalur Jalan Optimum Menggunakan Metode Dijkstra Kota Yogyakarta Berbasis Web” hasil dari penelitian ini menjelaskan bahwa kota Yogyakarta memiliki banyak predikat sebagai kota pelajar, kota kota budaya dan kota pariwisata. Dengan banyaknya predikat tersebut, penyampaian informasi masih sangat terbatas, hal ini menjadi salah satu kendala dalam penyampaian informasi dibidang lalu lintas dalam menemukan jalan optimum. Pada pengembangan sistem ini, penulis menggunakan metode Dijkstra yaitu pencarian

rute terdekat dengan membandingkan satu atau lebih rute yang ada. Solusi yang didapat dari penelitian ini adalah pembuatan sebuah aplikasi berbasis web yang bisa memberikan jalur alternatif terdekat.

Berdasarkan hal diatas, masalah yang sering timbul adalah kurangnya informasi untuk para pengguna jalan untuk jalur lain yang dapat dilalui. Oleh sebab itu perlu adanya suatu sistem informasi yang dapat membantu para pengguna jalan untuk mendapatkan informasi tentang jalur terdekat dan menghindari kemacetan.

## 2.8 State Of The Art

Berdasarkan isi dari kedua karya tulis di atas maka Tugas Akhir yang berjudul “Sistem Informasi Penentuan Jalur Alternatif Terdekat di Kota Jember” ini memiliki perbandingan sebagai berikut :

Tabel 2.4 State Of The Art

No	Aspek Analisa	Penulis		
		Selvi Sugiarto	Imron Fauzi	Muhammad Andri
1	Judul	Sistem Informasi Penentuan Jalur Alternatif Terdekat di Kota Jember Menggunakan Algoritma Dijkstra	Penggunaan Algoritma Dijkstra Untuk Pencarian Jalur Tercepat Dan Jalur Terpendek (Studi Kasus Pada Jalan Raya Antara Wilayah Blok M Dan Kota	Perancangan Sistem Informasi Geografis Penentuan Jalur Jalan Optimum Menggunakan Metode Dijkstra Kota Yogyakarta Berbasis Web
2	Topik	Sistem Informasi Geografis	Sistem Informasi Geografis	Sistem Informasi Geografis
3	Objek	Jalan sekitar Perguruan Tinggi	Jalan Raya Antara Wilayah	Kota Yogyakarta

Tabel 2.4 State Of The Art (Lanjutan)

No.	Aspek	Penulis		
	Analaisa	Selvi Sugiarto	Imron Fauzi	Muhammad Andri
4	Bahasa Pemrograman	Java	PHP, MySQL	PHP, MySQL
5	Metode	Prototype	Spiral Model	Metode Dijkstra
6	Tahun	2017	2011	2010
7	Fitur	- Peta Kecamatan Sumbersari - Graf Kecamatan Sumbersari - Alternatif Jalur Terdekat	-Login Admin -Update, Delete, Search Wilayah -Grafik Penggunaan Algoritma Dijkstra Oleh User	-Input Lokasi Pengguna -Output Jarak Ke Tempat Tujuan

## **BAB 3. METODE KEGIATAN**

### **3.1 Tempat dan Waktu**

Penelitian tugas akhir berjudul “Sistem Informasi Penentuan Jalur Alternatif Terdekat di Kota Jember Menggunakan Algoritma Dijkstra” dilaksanakan selama 6 bulan, mulai dari September 2017 – Maret 2018 yang bertempat di Politeknik Negeri Jember.

### **3.2 Bahan dan Alat**

#### **3.2.1 Bahan**

Bahan yang digunakan di dalam penelitian ini meliputi data pengukuran jalan yang berada di wilayah administratif Kecamatan Sumbersari, Jember dan data pengamatan jalur untuk menentukan tingkat kemacetan setiap harinya.

#### **3.2.2 Alat**

Alat yang dibutuhkan dalam proses pembuatan aplikasi ini ada dua jenis perangkat, yaitu perangkat keras dan perangkat lunak seperti yang diuraikan sebagai berikut.

##### **a. Perangkat Keras**

Perangkat keras yang digunakan dalam pembuatan sistem informasi ini adalah satu unit laptop dengan spesifikasi sebagai berikut :

- 1) Laptop Leovo G41
- 2) Processor AMD A8-7410 APU with AMD RADEON R5 Graphics 2.20 GHz
- 3) RAM 4 GB

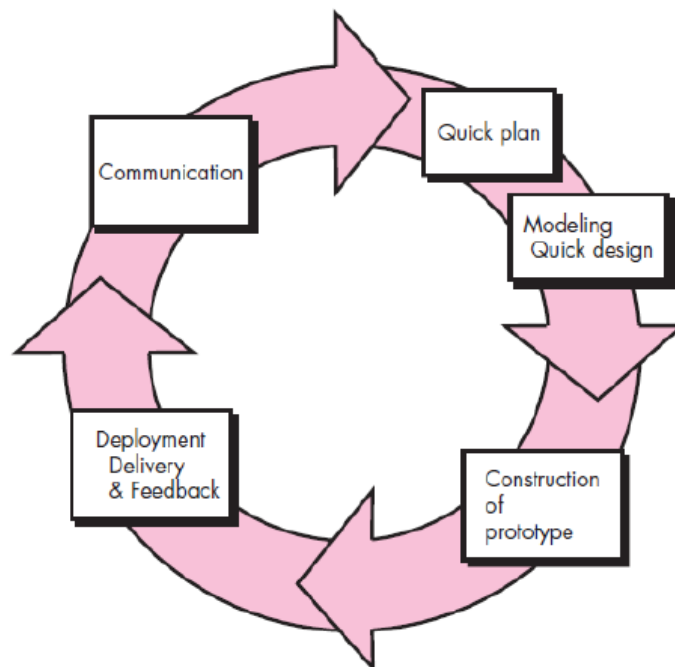
##### **b. Perangkat Lunak**

Perangkat lunak yang digunakan dalam pembuatan sistem informasi ini adalah sebagai berikut :

- 1) Sistem Operasi Windows 8
- 2) Microsoft Word 2010
- 3) NetBeans IDE 8.2
- 4) Paint

### 3.3 Metode Kegiatan

Metode yang digunakan untuk Sistem Informasi ini yaitu Model Prototype. Model prototype dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Dengan begitu, komunikasi antara pengembang dan pelanggan akan terjalin dengan baik, sehingga memperkecil kesalahan pembuatan software.



Gambar 3.1 Metode Pengembangan *Prototype* Menurut Pressman

Penjelasan tahapan - tahapan pada gambar 3.1 adalah sebagai berikut :

a. Communication

Tahap cummunication (komunikasi) adalah tahap mengumpulkan data-data kebutuhan yang harus dipenuhi dengan melihat beberapa literatur seperti buku, karya tulis ilmiah, serta website. Data awal tersebut dikumpulkan dalam bentuk dokumen untuk menyusun desain perangkat lunak pada tahap berikutnya.

b. Quick Plan

Quick Plan yaitu membuat design atau rancangan sederhana berupa Usecase dan Class Diagram yang akan mendefinisikan tentang alur dari proses yang akan dikerjakan.

c. Modeling Quick Design

Modeling Quick Design adalah tahap membangun mockup dari aplikasi yang akan dijadikan sebagai tampilan dari aplikasi yang akan dikembangkan beserta susunan database yang akan digunakan.

d. Construction of Prototype

Construction of Prototype adalah tahapan pengkodean sistem yang telah dibangun berdasarkan quick plan dan mockup yang telah direncanakan pada tahap sebelumnya. Mula-mula pengembang akan membuat layout yang sesuai dengan mockup dan membangun database yang telah ditentukan. Kemudian semua itu akan dibangun menjadi sebuah aplikasi prototype yang akan diserahkan kepada pengguna untuk uji coba.

e. Deployment Delivery and Feedback

Pada tahap ini dilakukan pengujian sistem terhadap pengguna yang nantinya akan memberikan feedback atau timbal balik bagi pengembang agar dapat mengembangkan aplikasinya menjadi lebih baik. Jika pengguna memberi feedback dengan saran atau maka pengembang dapat menggunakan saran untuk melakukan proses dari tahap pertama namun pada iterasi berbeda.

## BAB 4. HASIL DAN PEMBAHASAN

### 4.1 Communication

Tahap ini adalah tahap awal yang dilakukan dalam membangun Sistem Informasi Alternatif Jalur Terekat di Kecamatan Sumbersari Menggunakan Algoritma Dijkstra. Pada tahap ini dilakukan survei pengumpulan data nama jalan dan jarak dari setiap jalan yang di dapat dari pengklasifikasian kelas jalan kolektor primer. Titik yang menjadi node diletakkan pada setiap pertigaan dan perempatan jalan yang terdapat di Kecamatan Sumbersari. Untuk jarak digunakan skala 1 : 1000. Data yang didapatkan terdapat pada Tabel 4.1

Tabel 4.1 Data Jalan

Node	Nama Node (Titik)	Titik yang di Tuju	Jarak
1	Pertigaan Jalan Tapak Siring	2	500 m
		7	1000 m
2	Pertigaan Jalan Srganan	1	500 m
		3	1700 m
3	Pertigaan Jalan Kranjan	2	1700 m
		4	700 m
4	Pertigaan Jalan Parang Tritis	3	700 m
		4	650 m
5	Pertigaan Jalan Parang Tritis 2	4	650 m
		6	450 m
6	Pertigaan Jalan Koptu Berlian	5	450 m
		12	1700 m
7	Perempatan Jalan Mangandaran	1	1000 m
		8	1200 m
8	Pertigaan Jalan Tawang Mangu	7	1200 m
		9	1100 m
9	Pertigaan Jalan Danau Toba	8	1100 m
		10	850 m



Tabel 4.1 Data Jalan (Lanjutan)

Node	Nama Node (Titik)	Titik yang di Tuju	Jarak
10	Perempatan Jalan Mastrip	11	550 m
		9	850 m
		11	600 m
		14	1200 m
11	Pertigaan Jalan Batu Raden	9	550 m
		10	600 m
		13	1200 m
12	Pertigaan Jalan Kaliurang	6	1700 m
		13	1200 m
		27	1500 m
		11	1200 m
13	Perempatan Jalan Tidar/Kaliurang	12	1200 m
		15	350 m
		24	2400 m
		10	1200 m
14	Perempatan Jalan Kalimantan/DPR	16	900 m
		22	500 m
		13	350 m
		16	450 m
15	Pertigaan Jalan Riau	17	250 m
		14	900 m
		15	450 m
16	Perempatan Jalan Belitung	17	140 m
		18	450 m
		15	250 m
17	Pertigaan Jalan Jawa	16	140 m
		18	350 m

Tabel 4.1 Data Jalan (Lanjutan)

Node	Nama Node (Titik)	Titik yang di Tuju	Jarak
18	Pertigaan Jalan Halmahera	16	450 m
		17	350 m
		19	650 m
19	Pertigaan Jalan Gunung Batu	18	650 m
		20	400 m
		21	1100 m
20	Pertigaan Jalan Karimata	19	400 m
		21	700 m
		23	750 m
21	Pertigaan Jalan Letjen Panjaitan	19	1100 m
		20	700 m
		22	500 m
22	Pertigaan Jalan Letjen Suprpto	30	700 m
		14	1400 m
		21	500 m
23	Pertigaan Jalan Sriwijaya	29	550 m
		20	750 m
		24	900 m
24	Perempatan Jalan Pierre Tendean	25	1200 m
		13	2400 m
		23	900 m
25	Pertigaan Jalan Letjen Sutoyo	27	1800 m
		26	1500 m
		23	1200 m
		26	1500 m
		31	2100 m

Tabel 4.1 Data Jalan (Lanjutan)

Node	Nama Node (Titik)	Titik yang di Tuju	Jarak
26	Pertigaa Jalan Ajisaka	24	1500 m
		25	1500 m
		28	2000 m
27	Pertigaan Jalan Sritanjung	12	1500 m
		24	1800 m
		28	2000 m
28	Perempatan Jalan Yos Sudarso	26	2000 m
		27	2000 m
		32	7300 m
29	Pertigaan Jalan Kahuripan	22	750 m
		30	550 m
		31	600 m
30	Pertigaan Jalan Pajajaran	21	700 m
		29	550 m
31	Pertigaan Jalan Basuki Rahmat	25	2100 m
		29	600 m
32	Pertigaan Jalan Walter Monginsidi	28	7300 m

Untuk tingkat kemacetan, setiap titik yang di set tingkat kemacetannya akan ditambahkan bobotnya dengan tingkatan kemacetan. Penentuan tingkat kemacetan diambil dari pengamatan setiap harinya, dapat dilihat pada tabel berikut :

Tabel. 4.2 Kondisi Lalu Lintas

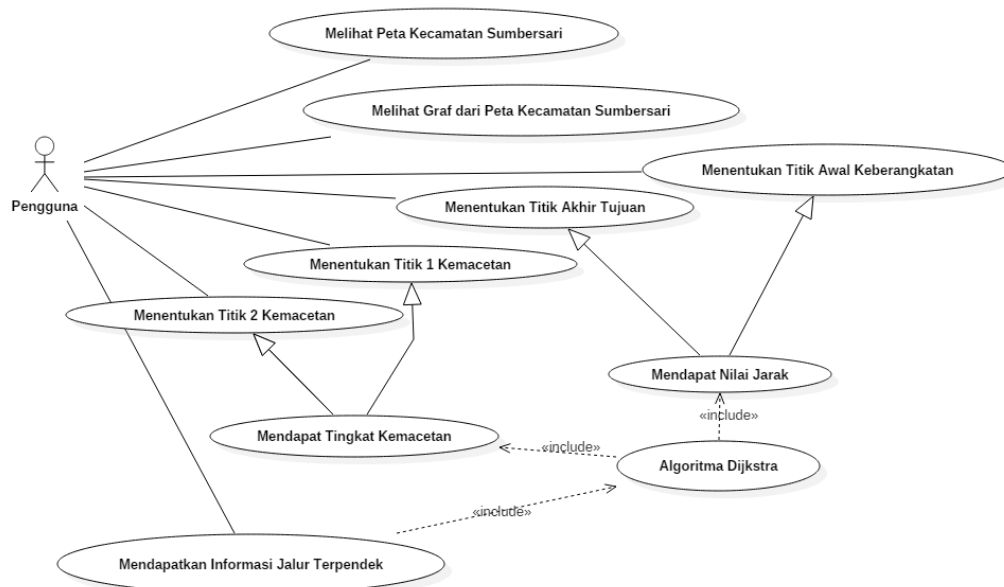
KONDISI LALU LINTAS	TINGKATAN KEMACETAN
NORMAL	0
SEDANG	1000
MACET	2000

## 4.2 Quick Plan

Tahap selanjutnya adalah tahap Quick Plan yaitu tahap dimana developer mulai memikirkan perencanaan alur proses pembangunan prototype secara cepat. Developer akan menentukan dan menjelaskan input dan output dari prototype yang akan dibangun. Pada tahap ini developer akan menyesuaikan apakah analisisnya sudah sama dengan apa yang diinginkan oleh pelanggan. Tahap ini juga merupakan tahap lanjutan dari Communication, karena tahap ini semua data harus sudah lengkap untuk menentukan input outputnya secara benar agar dapat dilanjutkan ke tahap desain.

## 4.3 Modeling Quick Design

Tahap Modeling Quick Design yaitu membuat model untuk mempresentasikan perangkat lunak yang dapat dipahami oleh pengguna, sehingga pengguna mengerti bagaimana alur kerja perangkat lunak yang akan dibuat. Pengguna dapat melihat rancangan alur kerja sistem yang akan dimodelkan menggunakan pemodelan UML. Untuk mengetahui jalannya proses didalam sistem informasi dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Usecase Diagram

#### 4.3.1 Definisi Aktor

Pada tabel 4.3 merupakan pendefinisian aktor kepada usecase sistem yang dikembangkan oleh pengembang :

Tabel 4.3 Definisi Aktor

No	Aktor	Deskripsi
1.	Pengguna	Orang yang menggunakan sistem untuk melihat peta, graf dan mencari jalur alternatif tercepat.

#### 4.3.2 Definisi Usecase

Pada tabel 4.4 ini merupakan pendefinisian usecase pada sistem informasi penentuan jalur alternatif terdekat di kecamatan Summersari :

Tabel 4.4 Definisi Usecase

No.	Usecase	Deskripsi
1.	Melihat Peta Kecamatan Summersari	Merupakan suatu proses untuk melihat peta Kecamatan Summersari yang dapat dilakukan pengguna.
2.	Melihat Graf Kecamatan Summersari	Merupakan suatu proses untuk melihat graf Kecamatan Summersari yang dapat dilakukan pengguna.
3.	Menentukan titik awal keberangkatan	Merupakan suatu proses untuk menentukan titik awal keberangkatan pengguna.
4.	Menentukan titik akhir tujuan	Merupakan suatu proses untuk menentukan titik akhir tujuan pengguna.
5.	Mendapat Nilai Jarak	Merupakan suatu proses untuk mendapatkan nilai dari titik awal keberangkatan menuju titik akhir tujuan.
6.	Algoritma Dijkstra	Merupakan suatu proses perhitungan sistem untuk mendapatkan alternatif jalur terdekat dari titik awal

Tabel 4.4 Definisi Usecase (Lanjutan)

No.	Usecase	Deskripsi
		keberangkatan menuju titik akhir tujuan
7.	Menentukan titik 1 kemacetan	Merupakan suatu proses untuk menentukan titik awal kemacetan pengguna.
8.	Menentukan titik 2 kemacetan	Merupakan suatu proses untuk menentukan titik akhir kemacetan pengguna
9.	Mendapat tingkat kemacetan	Merupakan suatu proses untuk mendapatkan informasi tingkat kemacetan dari titik 1 ke titik 2.
10.	Mendapatkan Informasi Jalur Terpendek	Merupakan suatu proses untuk mendapatkan informasi jalur terpendek yang dapat ditempuh pengguna.

#### 4.3.3 Skenario Usecase

Berikut ini adalah skenario jalannya masing-masing usecase yang telah didefinisikan sebelumnya :

Nama usecase : Melihat Peta Kecamatan Summersari

Skenario usecase : Skenario usecase terdapat pada tabel 4.5

Tabel 4.5 Skenario usecase Melihat Peta Kecamatan Summersari

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menekan button Peta	2. Menampilkan peta Kecamatan Summersari

Nama usecase : Melihat Graf Kecamatan Summersari

Skenario usecase : Skenario usecase terdapat pada tabel 4.6

Tabel 4.6 Skenario usecase Melihat Graf Kecamatan Summersari

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menekan button Graf	2. Menampilkan graf Kecamatan Summersari

Nama usecase : Menentukan titik awal keberangkatan

Skenario usecase : Skenario usecase terdapat pada tabel 4.7

Tabel 4.7 Skenario usecase menentukan titik awal keberangkatan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menentukan titik awal keberangkatan	2. Menyimpan titik awal keberangkatan

Nama usecase : Menentukan titik akhir tujuan

Skenario usecase : Skenario usecase terdapat pada tabel 4.8

Tabel 4.8 Skenario Usecase menentukan titik akhir tujuan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menentukan titik akhir tujuan	2. Menyimpan titik akhir tujuan

Nama usecase : Mendapat nilai jarak

Skenario usecase : Skenario usecase terdapat pada tabel 4.9

Tabel 4.9 Skenario usecase mendapat nilai jarak

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	1. Menyimpan nilai jarak dari penentuan titik awal keberangkatan dan titik akhir tujuan.

Nama usecase : Algoritma dijkstra

Skenario usecase : Skenario usecase terdapat pada tabel 4.10

Tabel 4.10 Skenario usecase algoritma dijkstra

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	1. Melakukan perhitungan untuk mendapatkan jalur alternatif terdekat.

Nama usecase : Menentukan titik 1 kemacetan

Skenario usecase : Skenario usecase terdapat pada tabel 4.11

Tabel 4.11 Skenario usecase menentukan titik 1 kemacetan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menentukan titik 1 kemacetan	2. Menyimpan titik 1 kemacetan

Nama usecase : Menentukan titik 2 kemacetan

Skenario usecase : Skenario usecase terdapat pada tabel 4.12



Tabel 4.12 Skenario usecase menentukan titik 2 kemacetan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menentukan titik 2 kemacetan	2. Menyimpan titik 2 kemacetan

Nama usecase : Mendapat tingkat kemacetan

Skenario usecase : Skenario usecase terdapat pada tabel 4.13

Tabel 4.13 Skenario usecase mendapat tingkat kemacetan

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
	1. Menyimpan nilai tingkat kemacetan dari penentuan titik 1 kemacetan dan titik 2 kemacetan.

Nama usecase : Mendapatkan informasi jalur terpendek

Skenario usecase : Skenario usecase terdapat pada tabel 4.14

Tabel 4.14 Skenario usecase mendapatkan informasi jalur terpendek

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
1. Menentukan titik awal keberangkatan.	2. Menyimpan titik awal keberangkatan.
3. Menentukan titik akhir tujuan.	4. Menyimpan titik akhir tujuan.
5. Menentukan titik 1 kemacetan.	6. Menyimpan titik 1 kemacetan.

Tabel 4.14 Skenario usecase mendapatkan informasi jalur terpendek (lanjutan)

Aksi Aktor	Reaksi Sistem
<b>Skenario Normal</b>	
7. Menentukan titik 2 kemacetan.	
	8. Menyimpan titik 2 kemacetan.
9. Mendapatkan informasi jalur terpendek.	10. Perhitungan algoritma dijkstra oleh sistem.

#### 4.4 Pembuatan Aplikasi

Setelah data dan design sudah terkumpul, pada tahap ini dilakukan pengkodean dari sistem informasi penentuan alternatif jalur terdekat di Kecamatan Summersari. Pada tahap ini perancangan yang dibuat ialah dari sisi pengguna yang diantaranya akan dijelaskan pada iterasi 1 menampilkan peta Kecamatan Summersari, pada iterasi 2 menampilkan graf Kecamatan Summersari, pada iterasi 3 informasi jalur terdekat.

##### 4.4.1 Iterasi 1

Pada iterasi 1, dilakukan pembuatan menampilkan peta Kecamatan Summersari. Berikut adalah tahapan prototype pada pembuatan menampilkan peta Kecamatan Summersari.

##### a. Communication

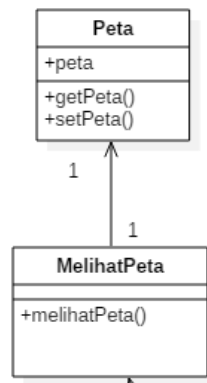
Pada tahap communication, usecase yang digunakan adalah proses yang dapat dilakukan dari sisi pengguna. Usecase yang dapat dilakukan adalah menampilkan peta Kecamatan Summersari.

##### b. Quick Plan

Pada tahap quick plan, macam-macam usecase yang digunakan adalah :

- 1) Menampilkan peta Kecamatan Summersari : Skenario usecase mengacu pada tabel 4.5.

Class Diagram yang digunakan mengacu pada gambar 4.2



Gambar 4.2 Melihat Peta Kecamatan Summersari

#### c. Quick Design

Pada tahap ini membangun mockup. Design mockup yang dibangun dapat dilihat pada gambar 4.3.



Gambar 4.3 Design mockup melihat peta Kecamatan Summersari

#### d. Construction of prototype

Pada tahap ini yaitu mengimplementasikan code sesuai design yang telah dibangun.

```

1. private void btnPetaActionPerformed(java.awt.event.ActionEvent evt) {
2.     PetaDialog petadialog = new PetaDialog(this,true);
3.     petadialog.setParent1(this);
4.     petadialog.setVisible(true);
5. }
  
```

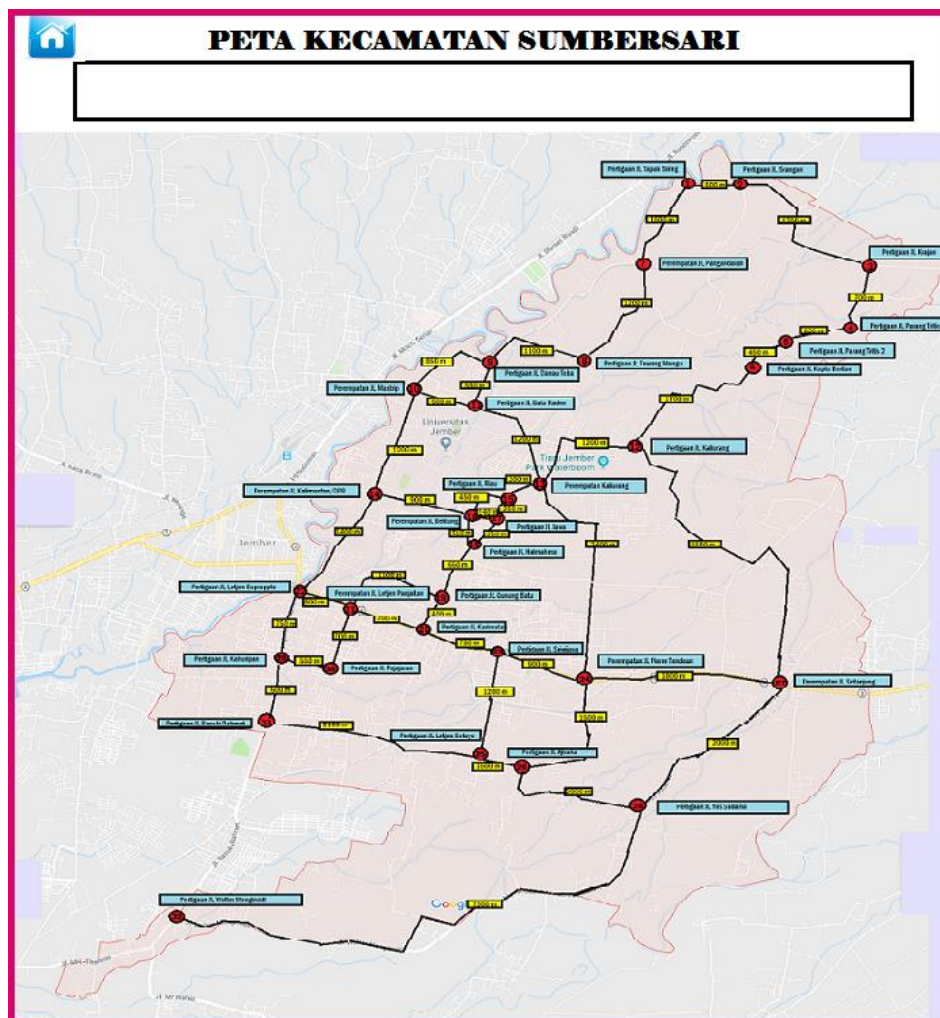
Kode Program 4.1 Melihat Peta Kecamatan Summersari

Fungsi `void btnPetaActionPerformed(java.awt.event.ActionEvent evt)` diatas adalah untuk menampilkan peta Kecamatan Sumbersari yang ada pada form `PetaDialog`.

### e. Deployment Delivery and Feedback

Pada tahap ini merupakan tahapan pengujian pada iterasi melihat peta Kecamatan Summersari, apabila pengguna tidak sesuai dengan hasil yang di harapkan maka pengembang memperbaiki sistem tersebut sehingga aplikasi atau sistem yang dibuat sesuai dengan yang di inginkan oleh pengguna pengujian ini menggunakan black box pada gambar 4.4 adalah Deployment Delivery dan pada tabel 4.15 adalah pengujian black box melihat peta Kecamatan Summersari.


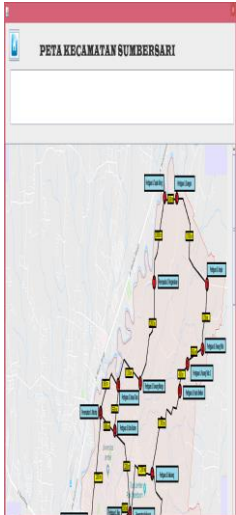
### 1) Deployment delivery



Gambar 4.4 Form Peta Kecamatan Summersari

## 2) Feedback

Tabel 4.15 Pengujian black box melihat peta Kecamatan Summersari

No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Menampil- kan peta kecamatan Summersari dengan mengklik button peta pada main form.		Sistem sukses menampilkan peta Kecamatan Subersari		Valid

## 4.4.2 Iterasi 2

Pada iterasi 2, dilakukan pembuatan menampilkan graf Kecamatan Summersari. Berikut adalah tahapan prototype pada pembuatan menampilkan peta Kecamatan Summersari.

## a. Communication

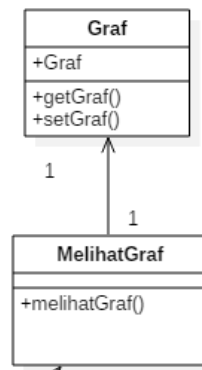
Pada tahap cummunication, usecase yang digunakan adalah proses yang dapat dilakukan dari sisi pengguna. Usecase yang dapat dilakukan adalah menampilkan graf Kecamatan Summersari.

## b. Quick Plan

Pada tahap quick plan, macam-macam usecase yang digunakan adalah :

- 1) Menampilkan graf Kecamatan Summersari : Skenario usecase mengacu pada tabel 4.7.

Class Diagram yang digunakan mengacu pada gambar 4.5



Gambar 4.5 Melihat Graf Kecamatan Sumbersari

#### c. Quick Design

Pada tahap ini membangun mockup. Design mockup yang dibangun dapat dilihat pada gambar 4.6.



Gambar 4.6 Design mockup melihat graf Kecamatan Sumbersari

#### d. Construction of prototype

Pada tahap ini yaitu mengimplementasikan code sesuai design yang telah dibangun.

1. `private void btnGrafActionPerformed(java.awt.event.ActionEvent evt) {`
2.     `GrafDialog grafdialog = new GrafDialog(this,true);`
3.     `grafdialog.setParent(this);`
4.     `grafdialog.setVisible(true);`
5.     `}`

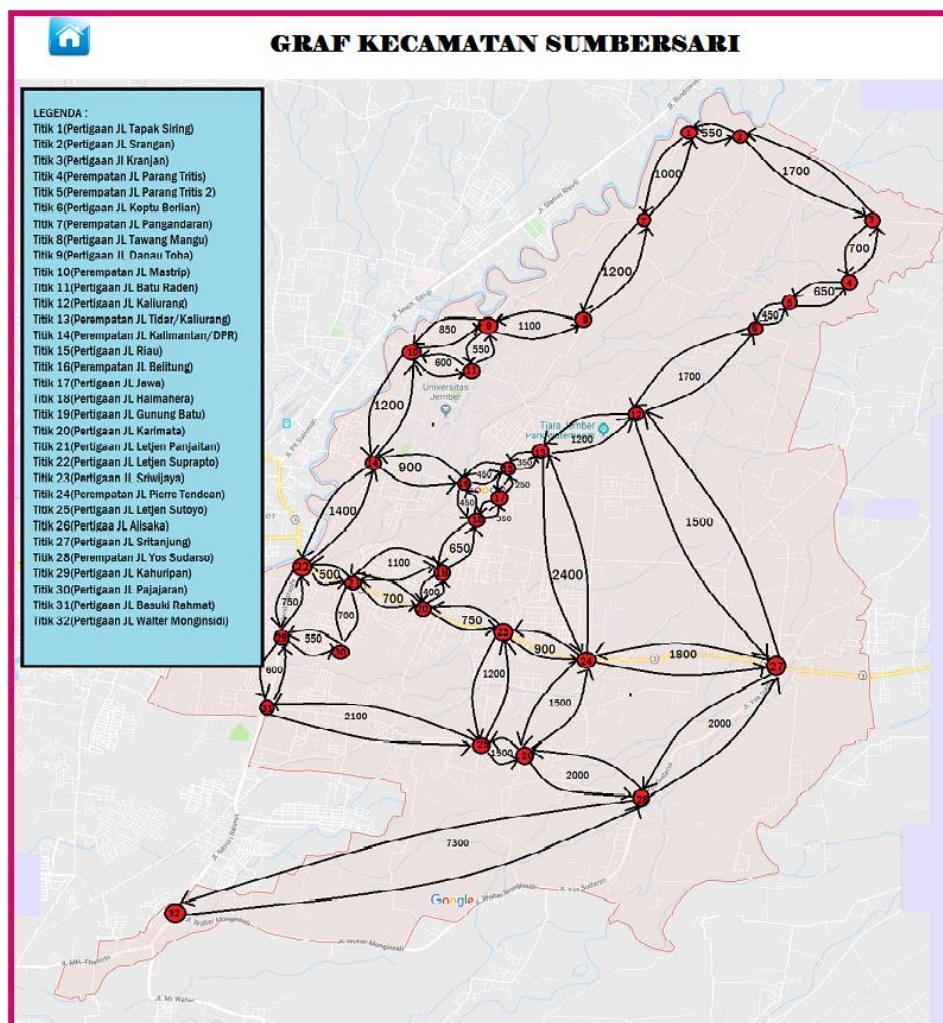
Kode Program 4.2 Melihat Graf Kecamatan Sumbersari

Fungsi `void btnGrafActionPerformed(java.awt.event.ActionEvent evt)` diatas adalah untuk menampilkan graf Kecamatan Summersari yang ada pada form GrafDialog.

e. Deployment Delivery and Feedback

Pada tahap ini merupakan tahapan pengujian pada iterasi melihat peta Kecamatan Summersari, apabila pengguna tidak sesuai dengan hasil yang di harapkan maka pengembang memperbaiki sistem tersebut sehingga aplikasi atau sistem yang dibuat sesuai dengan yang di inginkan oleh pengguna pengujian ini menggunakan black box pada gambar 4.7 adalah Deployment Delivery dan pada tabel 4.16 adalah pengujian black box melihat peta Kecamatan Summersari.

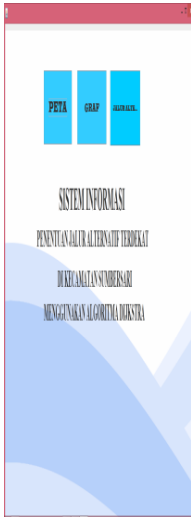
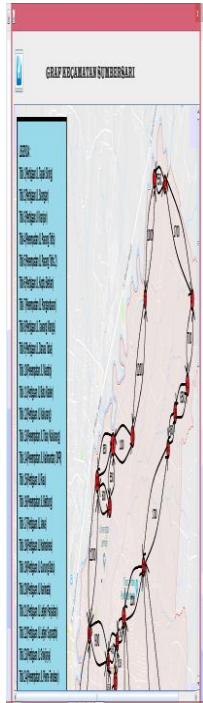
1) Deployment delivery



Gambar 4.7 Form graf Kecamatan Summersari

## 2) Feedback

Tabel 4.16 Pengujian black box melihat graf Kecamatan Summersari

No.	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Menampilkan graf kecamatan Summersari dengan mengklik button graf pada main form.		Sistem sukses menampilkan graf Kecamatan Subersari		Valid

## 4.4.3 Iterasi 3

Pada iterasi 3, dilakukan pembuatan menampilkan graf Kecamatan Summersari. Berikut adalah tahapan prototype pada pembuatan alternatif jalur terdekat yang dapat dilalui di Kecamatan Summersari.

## a. Communication

Pada tahap cummunication, usecase yang digunakan adalah proses yang dapat dilakukan dari sisi pengguna. Usecase yang dapat dilakukan adalah menentukan titik awal keberangkatan, menentukan titik akhir tujuan, algoritma dijkstra, mendapat nilai jarak, menentukan titik 1 kemacetan, menentukan titik 2 kemacetan, mendapat tingkat kemacetan, dan mendapatkan informasi jalur terpendek.

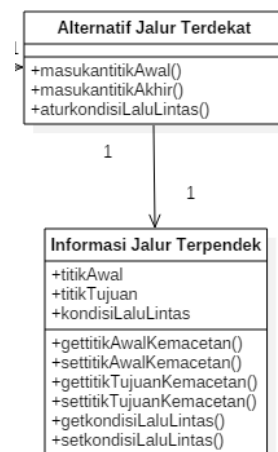
## b. Quick Plan

Pada tahap quick plan, macam-macam usecase yang digunakan adalah :



- 1) Menentukan titik awal keberangkatan : Skenario usecase mengacu pada tabel 4.7.
- 2) Menentukan titik akhir tujuan : Skenario usecase mengacu pada tabel 4.8.
- 3) Mendapat nilai jarak : Skenario usecase mengacu pada tabel 4.9.
- 4) Algoritma dijkstra : Skenario usecase mengacu pada tabel 4.10.
- 5) Menentukan titik 1 kemacetan : Skenario usecase mengacu pada tabel 4.11.
- 6) Menentukan titik 2 kemacetan : Skenario usecase mengacu pada tabel 4.12.
- 7) Mendapat tingkat kemacetan : Skenario usecase mengacu pada tabel 4.13.
- 8) Mendapatkan informasi jalur terpendek : Skenario usecase mengacu pada tabel 4.14.

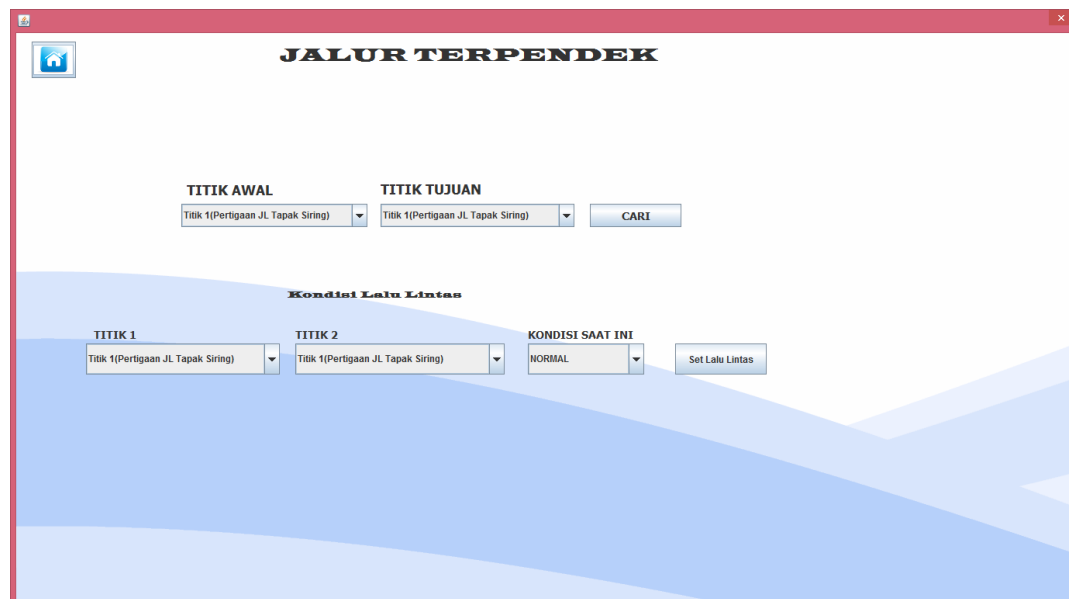
Class Diagram yang digunakan mengacu pada gambar 4.8



Gambar 4.8 Informasi Alternatif Jalur Terdekat

### c. Quick Design

Pada tahap ini membangun mockup. Design mockup yang dibangun dapat dilihat pada gambar 4.9.



Gambar 4.9 Design Mockup Informasi Alternatif Jalur Terdekat

d. Construction of prototype

Pada tahap ini yaitu mengimplementasikan code sesuai design yang telah dibangun.

- 1) `private void cariActionPerformed(java.awt.event.ActionEvent evt) {`
- 2)     `int NODE_AWAL = cbAwal.getSelectedIndex();`
- 3)     `int NODE_TUJUAN = cbTujuan.getSelectedIndex();`
- 4)     `if(NODE_AWAL != NODE_TUJUAN) {`
- 5)         `String lintasan = djs.getShortestPath(NODE_AWAL,`  
`NODE_TUJUAN);`
- 6)         `String[] isi = {"Titik 1(Pertigaan JL Tapak Siring)","Titik 2(Pertigaan`  
`JL Sangan)","Titik 3(Pertigaan Jl Kranjan)",`
- 7)         `"Titik 4(Perempatan JL Parang Tritis)","Titik 5(Perempatan JL Parang Tritis`  
`2)",`
- 8)         `"Titik 6(Pertigaan JL Koptu Berlian)","Titik 7(Perempatan JL`  
`Pangandaran)","Titik 8(Pertigaan JL Tawang Mangu)",`
- 9)         `"Titik 9(Pertigaan JL Danau Toba)","Titik 10(Perempatan JL Mastrip)","Titik`  
`11(Pertigaan JL Batu Raden)","Titik 12(Pertigaan JL Kaliurang)",`

```

10) "Titik 13(Perempatan JL Tidar/Kaliurang)", "Titik 14(Perempatan JL
    Kalimantan/DPR)", "Titik 15(Pertigaan JL Riau)", "Titik 16(Perempatan JL
    Belitung)",
11) "Titik 17(Pertigaan JL Jawa)", "Titik 18(Pertigaan JL Halmahera)", "Titik
    19(Pertigaan JL Gunung Batu)", "Titik 20(Pertigaan JL Karimata)",
12) "Titik 21(Pertigaan JL Letjen Panjaitan)", "Titik 22(Pertigaan JL Letjen
    Suprpto)", "Titik 23(Pertigaan JL Sriwijaya)", "Titik 24(Perempatan JL Pierre
    Tendean)",
13) "Titik 25(Pertigaan JL Letjen Sutoyo)", "Titik 26(Pertigaa JL Ajisaka)", "Titik
    27(Pertigaan JL Sritanjung)", "Titik 28(Perempatan JL Yos Sudarso)",
14) "Titik 29(Pertigaan JL Kahuripan)", "Titik 30(Pertigaan JL Pajajaran)", "Titik
    31(Pertigaan JL Basuki Rahmat)", "Titik 32(Pertigaan JL Walter
    Monginsidi)"};
15)         String[] tempHasil = lintasan.split(",");
16)         String tampilan = isi[Integer.parseInt(tempHasil[0])-1];
17)         for(int i = 1; i < tempHasil.length; i++){
18)             tampilan += ", " + isi[Integer.parseInt(tempHasil[i])-1];
19)         }
20)         tampilan = "Lintasan terpendek dari " + cbAwal.getSelectedItem() + " ke
    " + cbTujuan.getSelectedItem() + " adalah " + tampilan;
21)         MainForm.hasilDijkstra = tampilan;
22)         JFrame frame = (JFrame) SwingUtilities.getWindowAncestor(this);
23)         PetaDialog petadialog = new PetaDialog(frame, true);
24)         petadialog.setLocationRelativeTo(null);
25)         petadialog.setAlternative(this);
26)         petadialog.setParent(this);
27)         petadialog.setVisible(true);
28) } else {
29)         JOptionPane.showMessageDialog(null, "Titik awal keberangkatan
    tidak boleh sama dengan titik tujuan");
30)     }

```

```

31) }
32) private void cbLLAwalActionPerformed(java.awt.event.ActionEvent evt) {
33)     Int kondisi=djs.getKondisiLaluLintasSaatIni
        (cbLLAwal.getSelectedIndex(), cbLLTujuan.getSelectedIndex());
34)     cbLL.setEnabled(true);
35)     if (kondisi == 0) {
36)         JOptionPane.showMessageDialog(null, "Tidak ada jalur");
37)         cbLL.setSelectedIndex(0);
38)         cbLL.setEnabled(false);
39)     } else {
40)         cbLL.setSelectedIndex(kondisi-1);
41)     }
42) }
43) private void cbLLTujuanActionPerformed(java.awt.event.ActionEvent evt)
    {
44)     cbLLAwalActionPerformed(evt);
45) }
46) private void btnLLActionPerformed(java.awt.event.ActionEvent evt) {
47)     djs.setKondisiLaluLintasSaatIni(cbLLAwal.getSelectedIndex(),
        cbLLTujuan.getSelectedIndex(), cbLL.getSelectedIndex());
48)     JOptionPane.showMessageDialog(null, "Perubahan disimpan");
49) }

```

Kode Program 4.3 Informasi Jalur Alternatif Terdekat

Fungsi void cariActionPerformed(java.awt.event.ActionEvent evt) diatas adalah untuk mencari jalur alterntif terdekat dengan menentukan titik awal keberangkatan dan titik akhir tujuan.

Fungsi void cbLLAwalActionPerformed(java.awt.event.ActionEvent evt) diatas adalah untuk menentukan titik 1 kemacetan.

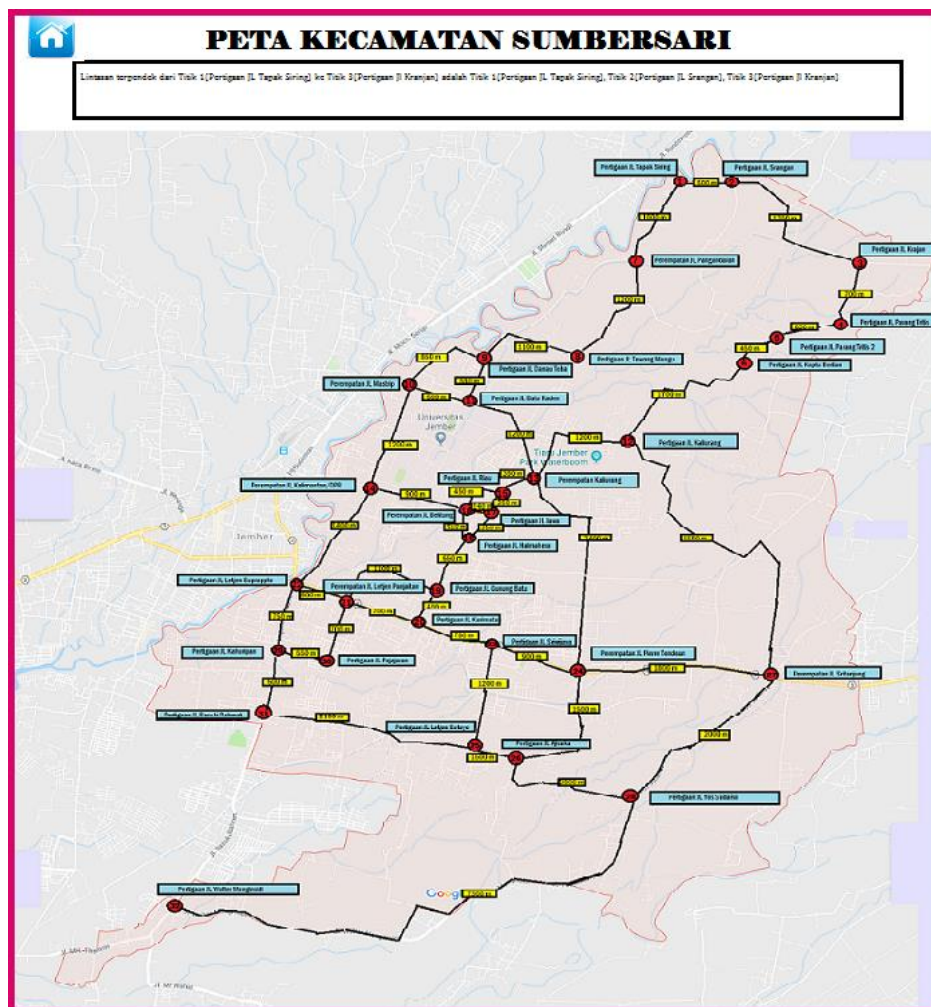
Fungsi void cbLLTujuanActionPerformed(java.awt.event.ActionEvent evt) diatas adalah untuk menentukan titik 2 kemcetan.

Fungsi void btnLLActionPerformed(java.awt.event.ActionEvent evt) diatas adalah untuk menyimpan tingkat kemacetan yang terjadi pada titik1 dan titik 2 yang sudah di tentukan sebelumnya.

e. Deployment Delivery and Feedback

Pada tahap ini merupakan tahapan pengujian pada iterasi melihat peta Kecamatan Summersari, apabila pengguna tidak sesuai dengan hasil yang di harapkan maka pengembang memperbaiki sistem tersebut sehingga aplikasi atau sistem yang dibuat sesuai dengan yang di inginkan oleh pengguna pengujian ini menggunakan black box pada gambar 4.10 adalah Deployment Delivery dan pada tabel 4.17 adalah pengujian black box alternatif jalur terdekat.

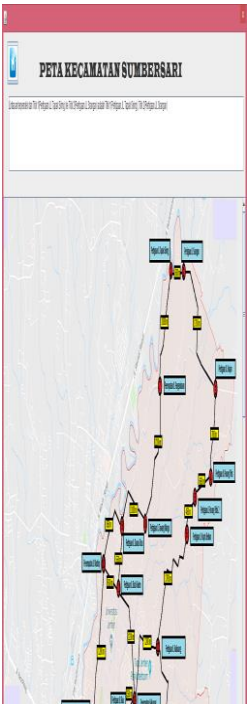

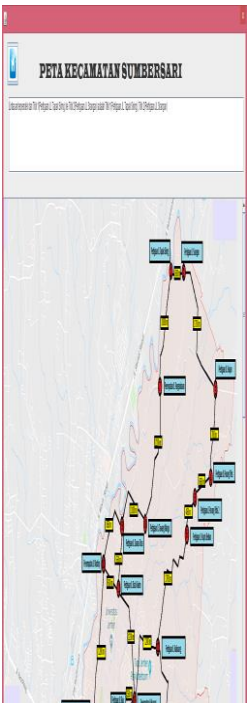
3) Deployment delivery



Gambar 4.10 Form Informasi Alternatif Jalur Terdekat

## 4) Feedback

Tabel 4.17 Pengujian black box informasi alternatif jalur terdekat

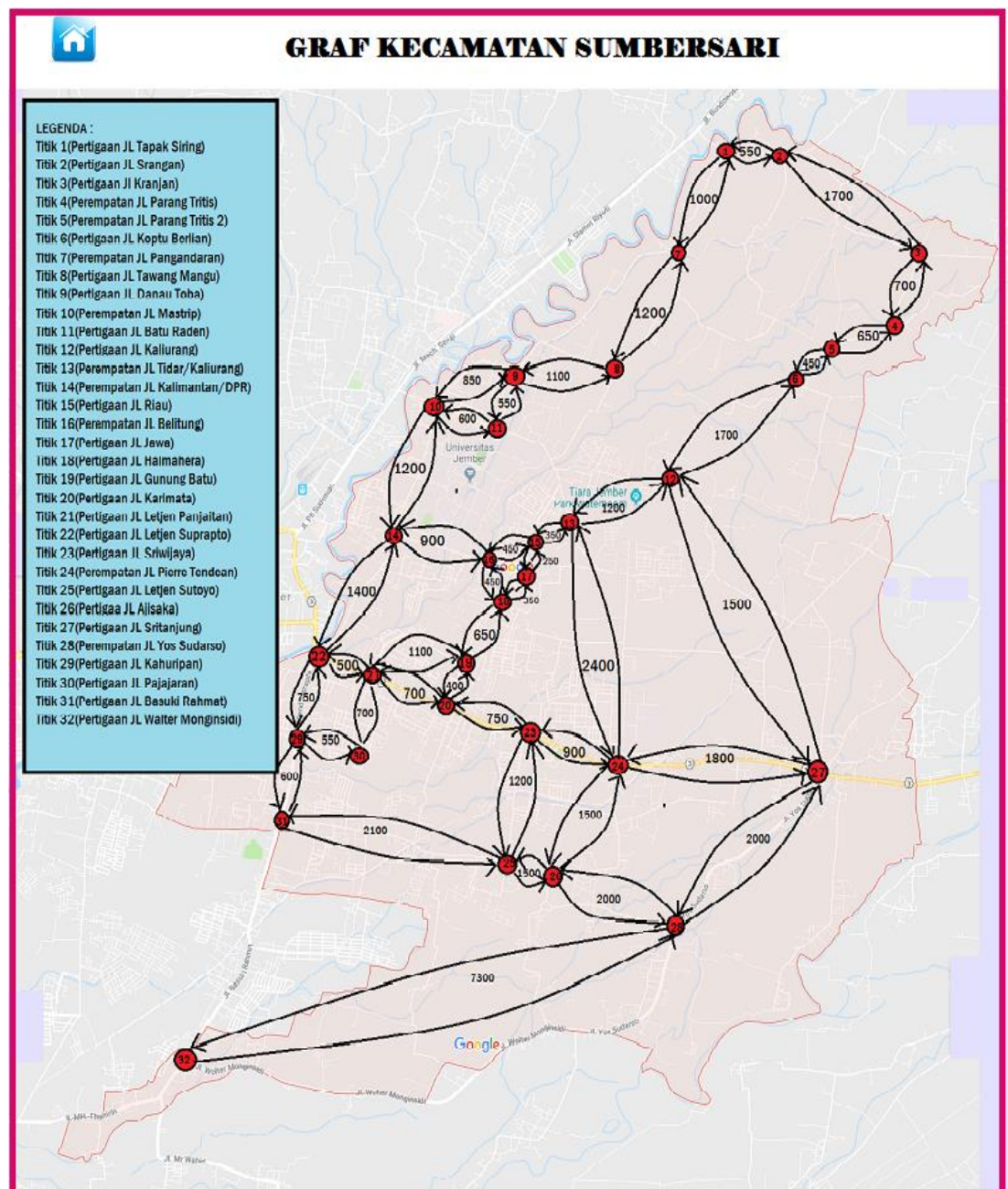
No	Skenario Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Menampilkan jalur alternatif terdekat yang dapat dilalui dengan menentukan titik awal keberangkatan dan titik akhir tujuan.	<div> <div>TITIK AWAL</div> <div>TITIK TUJUAN</div> <div> <div>Titik 1/Perempatan PP...</div> <div>Titik 4/Perempatan J...</div> <div>CARI</div> </div> </div>	Sistem sukses menampilkan jalur alternatif terdekat yang dapat dilalui.		Valid
2.	Menampilkan jalur alternatif terdekat yang dapat dilalui setelah mengatur tingkat kemacetan.		Sistem sukses menampilkan jalur alternatif terdekat yang dapat dilalui.		Valid

#### 4.5 Perhitungan Algoritma Dijkstra

Perhitungan manual menggunakan Algoritma Dijkstra dapat dilihat dibawah ini :

##### a. Graf Kecamatan Summersari

Jarak yang terdapat pada graf kecamatan Summersari digunakan sebagai bobot untuk menghitung jalur terpendek.



Gambar 4.11 Graf Kecamatan Summersari



### b. Matriks

Tahapan pertama memulai perhitungan menggunakan algoritma dijkstra yaitu membuat matriks dari setiap bobot yang ada pada peta Kecamatan Summersari. Untuk matriks yang berisi angka 0, merupakan titik yang bertujuan pada dirinya sendiri(tetap di titik awal), untuk matriks yang berisi jarak merupakan titik awal menuju titik selanjutnya yang mempunyai jarak, sedangkan matriks yang berisi simbol infinity ( $\infty$ ) merupakan titik yang tidak terhubung secara langsung antara titik keberangkatan dengan titik tujuan. Dibawah ini merupakan matriks kecamatan Summersari.

Tabel 4.18 Matriks

	1	2	3	4	5	6	7	8
1	0	550	$\infty$	$\infty$	$\infty$	$\infty$	1000	$\infty$
2	550	0	900	1800	$\infty$	$\infty$	$\infty$	$\infty$
3	$\infty$	900	0	1100	2300	$\infty$	$\infty$	$\infty$
4	$\infty$	1800	1100	0	550	$\infty$	$\infty$	1100
5	$\infty$	$\infty$	2300	550	0	450	$\infty$	$\infty$
6	$\infty$	$\infty$	$\infty$	$\infty$	450	0	$\infty$	$\infty$
7	1000	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	400
8	$\infty$	$\infty$	$\infty$	1100	$\infty$	$\infty$	400	0

### c. Perhitungan Lintasan Terpendek

1) Simpul awal a = 1, tujuan kesemua simpul.

Tabel 4.19 Perhitungan Manual Lintasan Terpendek Kecamatan Summersari

Lela- ran	Simpul yang dipilih	Linta- san	S(NODE)								D (JARAK)							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Inisi al	-	-	0	0	0	0	0	0	0	0	$\infty$	550	$\infty$	$\infty$	$\infty$	$\infty$	1000	$\infty$
1	1	1	1	0	0	0	0	0	0	0	$\infty$	550 (1,2)	$\infty$	$\infty$	$\infty$	$\infty$	1000 (1,7)	$\infty$
2	2	1,2	1	1	0	0	0	0	0	0	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	$\infty$	$\infty$	1000 (1,7)	$\infty$
3	7	1,7	1	1	0	0	0	0	1	0	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	$\infty$	$\infty$	1000 (1,7)	1400 (1,7,8)
4	8	1,7, 8	1	1	0	0	0	0	1	1	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	$\infty$	$\infty$	1000 (1,7)	1400 (1,7,8)
5	3	1,2,3	1	1	1	0	0	0	1	1	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	3750 (1,2,3, 5)	$\infty$	1000 (1,7)	1400 (1,7,8)



Tabel 4.19 Perhitungan Manual Lintasan Terpendek Kecamatan Sumpster (Lanjutan)

Lelaran	Simpul yang dipilih	Lintasan	S(NODE)								D (JARAK)							
			1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
6	4	1,2,4	1	1	1	1	0	0	1	1	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	2900 (1,2,4,5)	$\infty$	1000 (1,7)	1400 (1,7,8)
7	5	1,2,4,5	1	1	1	1	1	0	1	1	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	2900 (1,2,4,5)	3350 (1,2,4,5,6)	1000 (1,7)	1400 (1,7,8)
8	6	1,2,4,5,6	1	1	1	1	1	1	1	1	$\infty$	550 (1,2)	1450 (1,2,3)	2350 (1,2,4)	2900 (1,2,4,5)	3350 (1,2,4,5,6)	1000 (1,7)	1400 (1,7,8)

Lelaran digunakan sebagai array perulangan untuk melakukan perhitungan dalam algoritma dijkstra. Lelaran ke 1 artinya perhitungan pertama yang dimulai, selalu dimulai dari titik awal keberangkatan. Simpul yang dipilih adalah simpul yang di pilih untuk melakukan perhitungan, simpul yang pertama dipilih adalah simpul awal keberangkatan. Lintasan adalah jalur yang dapat dilalui dari titi awal ke titik akhir. S(NODE) adalah penanda titik titik yang sudah terjamah dengan angka 1 dan titik yang belum tertambah dengan angka 0. D(JARAK) adalah bobot antara titik awal dan titik tujuan yang di dapat dari tabel matriks. Lelaran ke 1, merupakan titik awal keberangkatan yaitu titik 1, lintasan yang terpilih tetap pada titik 1, karena tujuan pada titik 1. Karena, titik satu sudah terjamah, maka di tandai dengan angka 1 dan titik yang belum terjamah ditandai dengan angka 0. Untuk D(JARAK) dilihat dari matriks, pada node 1 karena titik awal maka di set simbol  $\infty$  (infinity), untuk titik berikutnya yaitu node 2 terisi bobot 550 yang didapat dari jarak titik 1 ke titik 2, kemudian node 3 terisi simbol  $\infty$  (infinity) karena titik 1 tidak mengarah langsung ke titik 3. Kemudian pada node 7 terdapat bobot 1000 yang didapat dari jarak titik 1 ke titik 7. Untuk lelaran yang ke dua di dapat dari perbandingan node yang sudah terisi bobot, disini ada dua node yaitu node 2 dan node 7, ambil node yang bobotnya bernilai lebih kecil untuk dijadikan lelaran yang berikutnya.

a. Hasil Lintasan Terpendek

Tabel 4.20 Hasil Perhitungan Manual Lintasan Terpendek Kecamatan Summersari

Awal	Tujuan	Lintasan Terpendek	Titik yang dilalui
1	2	550 (1,2)	1
1	3	1450 (1,2,3)	2
1	4	2350 (1,2,4)	2
1	5	2900 (1,2,4,5)	2,4
1	6	3350 (1,2,4,5,6)	2,4,5
1	7	1000 (1,7)	1
1	8	1400 (1,7,8)	7

Pada kolom awal, merupakan titik awal keberangkatan, disini dimulai dari titik 1. Kolom tujuan adalah kolom tujuan dari titik keberangkatan yang menuju kesemua titik yang ada pada graf. Kemudian kolom lintasan terpendek adalah hasil dari setiap titik yang akan dituju dengan jarak terpendek. Kemudian kolom titik yang dilalui adalah kumpulan titik-titik yang dilalui dari titik awal keberangkatan ke titik tutjuan.

## **BAB 5. KESIMPULAN DAN SARAN**

### **5.1 Kesimpulan**

Berdasarkan hasil yang dibahas dari Sistem Informasi Penentuan Jalur Alternatif Terdekat di Kecamatan Sumbersari, didapatkan kesimpulan sebagai berikut :

- a. Sistem informasi ini dapat membantu pengguna untuk mengetahui informasi jalur alternatif terdekat yang dapat dilalui sekitar Kecamatan Sumbersari dan menghemat waktu tempuh.
- b. Algoritma dijkstra berhasil diterapkan dalam penentuan rute terpendek di Kecamatan Sumbersari.
- c. Algoritma dijkstra merupakan suatu metode untuk menentukan rute terpendek dengan memilih suatu titik koordinat awal dan koordinat akhir untuk menentukan suatu rute lokasi.
- d. Untuk melakukan perhitungan dijkstra, tahapan pertama yaitu membuat tabel matriks untuk semua node, digunakan untuk meletakkan jarak yang sudah terlihat pada peta. Tahapan kedua yaitu melakukan perhitungan dari awal node keberangkatan, kemudian membandingkan jarak antar node, ambil node yang nilainya terkecil untuk dijadikan titik terjamah selanjutnya. Untuk node yang sudah terjamah, jika dimatriks terdapat jarak, maka tidak usah dibandingkan jaraknya.

### **5.2 Saran**

Berdasarkan kesimpulan yang telah dikemukakan. Dapat diajukan beberapa saran untuk pengembangan lebih lanjut :

- a. Membuat desain lebih bersifat user interface.
- b. Menambahkan fitur – fitur yang lain agar lebih nyaman digunakan.
- c. Memberi penanda pada peta untuk rute yang dilewati setelah perhitungan dilakukan.
- d. Menambahkan fitur yang dapat terhubung langsung dengan Google Map.

## DAFTAR PUSTAKA

- Andri, Muhammad. 2010. “Perancangan Sistem Informasi Geografis Penentuan Jalur Jalan Optimum Menggunakan Metode Dijkstra Kota Yogyakarta Berbasis Web”. Repository Amikom. [http://repository.amikom.ac.id/files/Publikasi06.11.1142\\_.pdf](http://repository.amikom.ac.id/files/Publikasi06.11.1142_.pdf). [12 Maret 2017]
- B, Franda. 2015. Bab 2. Landasan Teori. Eprints. <http://eprints.polsri.ac.id/1547/3/BAB%20II.pdf>. 2015. [12 Maret 2017]
- E-Journal UAJY. Tanpa Tahun. *Bab 2. Landasan Teori*. E-Journal UAJY. <http://e-journal.uajy.ac.id/5123/3/2TS13156.pdf>. [12 Maret 2017]
- Fauzi, Imron. 2011. “Penggunaan Algoritma Dijkstra Dalam Pencarian Rute Tercepat Dan Rute Terpendek (Studi Kasus Pada Jalan Raya Antara Wilayah Blok M Dan Kota)”. Repository Uinjt. <http://repository.uinjt.ac.id/dspace/bitstream/123456789/2696/1/IMRO20FAUZI-FST.pdf>. [12 Maret 2017]
- Hutahean, J. 2014. “Konsep Sistem Informasi”. *Of Deepublish, Online-Edition*. <https://books.google.co.id>. [30 Maret 2017]
- Repository USU. Tanpa Tahun. Bab 2. Landasan Teori. Repository USU. <http://repository.usu.ac.id/bitstream/123456789/48624/3/Chapter%20II.pdf>. [12 Maret 2017]
- S, Rosa A. dan M. Shalahuddin. 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika. [12 Mei 2018]
- Telkom University. Tanpa Tahun. *Shortest Path Algorithm (Dijkstra, Belman-Ford)*. Telkom University. <http://leanna.staff.telkomuniversity.ac.id/files/2015/09/4.SHORTEST-PATH-ALGORITHM.pdf>. [12 Maret 2017]