

SKRIPSI

**PENCARIAN JALUR TERPENDEK MENGGUNAKAN
ALGORITMA DIJKSTRA PADA LINGKUNGAN YANG
BERSIFAT FUZZY**



Erlangga Laimena

NPM: 2013730021

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2018**

UNDERGRADUATE THESIS

**THE SHORTEST PATH USING THE DIJKSTRA IN FUZZY
ENVIRONMENT**



Erlangga Laimena

NPM: 2013730021

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2018**


LEMBAR PENGESAHAN



**PENCARIAN JALUR TERPENDEK MENGGUNAKAN
ALGORITMA DIJKSTRA PADA LINGKUNGAN YANG
BERSIFAT FUZZY**

Erlangga Laimena

NPM: 2013730021


Bandung, 17 Mei 2018

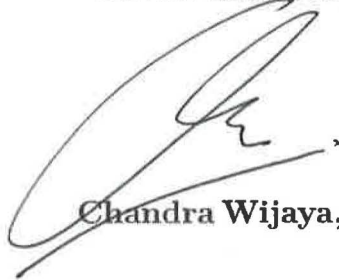
Menyetujui,

Pembimbing



Mariskha Tri Adithia, P.D.Eng

Ketua Tim Penguji



Chandra Wijaya, M.T.

Anggota Tim Penguji



Dr.rer.nat. Cecilia Esti Nugraheni

Mengetahui,

Ketua Program Studi



Mariskha Tri Adithia, P.D.Eng



PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENCARIAN JALUR TERPENDEK MENGGUNAKAN ALGORITMA DIJKSTRA PADA LINGKUNGAN YANG BERSIFAT FUZZY

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 17 Mei 2018



Erlangga Laimena
NPM: 2013730021

ABSTRAK

Saat ini kegiatan transportasi sudah merupakan hal yang biasa dilakukan orang-orang banyak. Biasanya dalam melakukan kegiatan transportasi, orang-orang memilih rute yang menghasilkan jarak tempuh yang minimal karena masalah waktu. Namun seringkali pada praktiknya, dalam perjalanan sering ditemukan hambatan. Beberapa faktor yang menghambat sebuah perjalanan biasanya karena ruas jalan yang padat, bencana alam, kerusakan jalan dan lainnya. Karena faktor-faktor inilah setiap perjalanan tidak pasti menghasilkan waktu tempuh minimum meskipun dipilih jalur dengan total jarak minimum. Maka dibutuhkan teknik untuk memilih jalur yang menghasilkan total waktu minimum.

Salah satu algoritma yang populer digunakan untuk menyelesaikan masalah pencarian jalur terpendek adalah Algoritma *Dijkstra*. Algoritma *Dijkstra* bekerja dengan menghitung penjumlahan antar jalur lalu dipilih jalur dengan total penjumlahan antar jalur yang paling minimum. Algoritma *Dijkstra* biasa digunakan pada *weighted graph* berarah maupun *weighted graph* tidak berarah. Syarat utama agar sebuah *graph* dapat dilakukan penelusuran menggunakan Algoritma *Dijkstra* adalah *weight* untuk setiap *edge* harus bernilai positif.

Fuzzy merupakan sebuah logika yang memiliki kemampuan untuk merepresentasikan sebuah masalah yang bersifat samar. Dalam merepresentasikan kondisi yang samar, *fuzzy* menggunakan *membership value* pada rentang 0 sampai 1 ($[0, 1]$). Apabila sebuah pernyataan bernilai 0 maka pernyataan itu sangat salah, sebaliknya apabila bernilai 1 maka pernyataan tersebut sangat benar. Apabila *membership value* untuk sebuah pernyataan memiliki nilai pada rentang $[0, 1]$, maka nilai kebenaran dari pernyataan tersebut memiliki kebenaran sebesar nilai *membership value* yang didapat. Pada masalah transportasi diatas, didapatkan variabel yang samar adalah "waktu". Waktu tempuh sebuah rute akan direpresentasikan menggunakan *fuzzy number* dengan metode *Triangular Fuzzy Number*.

Pada skripsi ini akan dibuat perangkat lunak yang memiliki kemampuan untuk menyelesaikan masalah pencarian jalur terpendek menggunakan Algoritma *Dijkstra* pada lingkungan yang bersifat fuzzy. Untuk jalur terpendek pada skripsi ini didefinisikan rute yang menghasilkan total waktu paling minimum. Agar *triangular fuzzy number* dapat diimplementasikan pada perangkat lunak, dibutuhkan proses *defuzzification* untuk mengubah *triangular fuzzy number* menjadi bilangan riil (\mathbf{R}).

Berdasarkan hasil pengujian yang dilakukan, perangkat lunak yang sudah dibangun dapat menjalankan Algoritma *Dijkstra* untuk kasus yang bersifat *Fuzzy*.

Kata-kata kunci: fuzzy, algoritma, dijkstra, triangular

ABSTRACT

The current activities of transport already is wont to do people a lot. Usually in the transport activities, the people choose the routes that generate minimal mileage due to problems of time. But often in practice, in the course often found obstacles. Some of the factors that impede a trip is usually due to the hectic roads, natural disasters, damage to roads and other. It is these factors because each trip is not definitely produce minimum travel time though selected lines with a total minimum distance. Then it takes technique to choose a path that yields a total minimum time.

One of the popular algorithms used to solve the problem of searching shortest path Algorithm is Dijkstra. Dijkstra's algorithm works by calculating the summation of last line between selected lines with a total sum of between the lines of the most minimum. Dijkstra algorithm used commonly on weighted graph or weighted towards graph is not directional. The main requirement for a graph can do a search using Dijkstra's algorithm is a weight to each edge must be a positive value.

Fuzzy is a logic that has the ability to represent a the problem is vague. In a vague condition represents, fuzzy use membership value in the range 0 to 1 ($[0, 1]$). When a statement is worth 0 then the statement was very wrong, otherwise, once worth 1 then the statement correct. If the membership value for a statement have values in the range $[0, 1]$, then the truth value of the statement have the truth of the value of membership value obtained. On the problem of transport above, obtained a vague variable is "time". It takes a route will be represented using fuzzy number with method of Triangular Fuzzy Number.

This thesis will be made in software that have kemampuan to complete the shortest path search problem using Dijkstra's Algorithm in environments that are fuzzy. For the shortest paths on this route defined thesis that yields a total most of the time to a minimum. In order for triangular fuzzy number can be implemented on a device software, it takes defuzzification to change the process of triangular fuzzy number be real numbers (\mathbb{R}).

Based on the results of testing done, already built software that can Dijkstra algorithm is run for the case that is Fuzzy.

Keywords: fuzzy, algorithm, dijkstra, triangular

Penulis mempersembahkan skripsi ini untuk kedua Orang Tua

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, atas limpahan rahmat dan hidayahnya penulis dapat menyelesaikan skripsi yang berjudul "Pencarian jalur terpendek menggunakan algoritma dijkstra pada lingkungan yang bersifat fuzzy".

Dalam kesempatan ini penulis mengucapkan terimakasih yang sedalam-dalamnya kepada Yth :

1. Orang tua atas doa dan semangat yang diberikan.
2. Bapak Husnul selaku dosen pembimbing yang telah membimbing sampai skripsi ini selesai.
3. Teman-teman yang sudah memberikan dukungan moral dan doanya.

Penulis menyadari bahwa dalam penyusunan laporan ini jauh dari sempurna, baik dari segi penyusunan, bahasan, ataupun penulisannya. Oleh karena itu penulis mengharapkan kritik dan saran yang sifatnya membangun, khususnya dari dosen dibidang yang terkait guna menjadi acuan dalam bekal pengalaman bagi penulis untuk lebih baik di masa yang akan datang.

Bandung, Mei 2018

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Penulisan	3
2 LANDASAN TEORI	5
2.1 Fuzzy	5
2.1.1 Propositional Logic [1]	5
2.1.2 Crisp Sets [1]	7
2.1.3 Fuzzy Logic [1]	7
2.1.4 Fuzzy Sets [1]	8
2.1.5 Membership Function	9
2.1.6 Alpha Cuts [1]	10
2.1.7 Fuzzy Number [1]	11
2.1.8 Defuzzification [2]	12
2.2 Graph	13
2.2.1 Jenis-jenis Graph	13
2.2.2 Incident dan Degree [3]	13
2.2.3 Representasi Graph	14
2.2.4 Konektivitas Graph	16
2.3 Dijkstra [3]	20
2.3.1 Pseudocode Dijkstra	22
3 ANALISIS	23
3.1 Fuzzy Number untuk Merepresentasikan Waktu Tempuh	23
3.1.1 Implementasi Triangular Fuzzy Number Terhadap Algoritma Dijkstra	23
3.1.2 Flow Chart Dijkstra	25
3.2 Analisis Perangkat Lunak	25
3.2.1 Flow Chart Perangkat Lunak	26
3.2.2 Use Case dan Skenario	27
4 PERANCANGAN	31

4.1	Kebutuhan Masukan dan Keluaran	31
4.2	Rancangan Antar Muka	32
4.3	Diagram Kelas Rinci	34
4.4	Rincian Atribut dan Method	36
4.4.1	Kelas FuzzyNumber	36
4.4.2	Kelas Place	37
4.4.3	Kelas Graph	38
4.4.4	Tester	41
4.4.5	Kelas GUIFile	41
4.4.6	Kelas GUIStart	41
4.4.7	Kelas GUIHasil	41
4.5	Sequence Diagram	42
5	IMPLEMENTASI DAN PENGUJIAN	45
5.1	Implementasi Antarmuka	45
5.2	Pengujian Fungsional	46
5.2.1	Pengujian Manual	46
5.2.2	Pengujian Dengan Menggunakan Perangkat Lunak	62
5.2.3	Kesimpulan Pengujian Fungsional	70
5.3	Pengujian Eksperimental	70
6	KESIMPULAN DAN SARAN	73
6.1	Kesimpulan	73
6.2	saran	73
	DAFTAR REFERENSI	75
	A KODE PROGRAM	77
	B PENELUSURAN DIJKSTRA TEST CASE 2	107
	C PENELUSURAN DIJKSTRA TEST CASE 3	109
	D PENELUSURAN DIJKSTRA TEST CASE 4	115
	E PENGUJIAN EKSPERIMENTAL GRAPH DENGAN VERTEX 10	119
	F PENGUJIAN EKSPERIMENTAL GRAPH DENGAN VERTEX 20	125
	G PENGUJIAN EKSPERIMENTAL GRAPH DENGAN VERTEX 30	139
	H PENGUJIAN EKSPERIMENTAL GRAPH DENGAN VERTEX 40	155
	I PENGUJIAN EKSPERIMENTAL GRAPH DENGAN VERTEX 50	173

DAFTAR GAMBAR

2.1	Kurva Triangular	9
2.2	Kurva Trapezoidal	10
2.3	Fuzzy Set dengan α -cut	10
2.4	Fuzzy Number untuk kondisi \bar{N}	11
2.5	Jenis-jenis Graph	13
2.6	Simple Graph	14
2.7	Disconnected Graph	17
2.8	Graph dengan Euler Path	17
2.9	Graph dengan Euler Circuit	18
2.10	Graph dengan Hamilton Path	19
2.11	Graph dengan Hamilton Circuit	19
2.12	Weighted Graph	20
3.1	Weighted Graph dengan Fuzzy Number	24
3.2	Flowchart Implementasi Triangular Fuzzy Number Terhadap Algoritma Dijkstra	25
3.3	Diagram Kelas	26
3.4	Flow Chart Cara Kerja Perangkat Lunak	27
3.5	Use Case yang merepresentasikan hak dari User	28
4.1	Contoh file masukan untuk perangkat lunak	31
4.2	Rancangan Antarmuka File Input	32
4.3	Rancangan Antarmuka Start	33
4.4	Rancangan Antarmuka Hasil	33
4.5	Diagram Kelas Final	35
4.6	Kelas Fuzzy Number	36
4.7	Kelas Place	37
4.8	Kelas Graph	38
4.9	Kelas Tester	41
4.10	Kelas GUIFile	41
4.11	Kelas GUIStart	41
4.12	Kelas GUIHasil	41
4.13	Sequence Diagram Perangkat Lunak	42
5.1	Antarmuka Pertama	45
5.2	Antarmuka Kedua	46
5.3	Antarmuka Ketiga	46
5.4	Graph Test Case Pertama dengan Triangular Fuzzy Number	48
5.5	Graph Test Case 1 Hasil Deffuzification	49
5.6	Graph Test Case Kedua dengan Triangular Fuzzy Number	51
5.7	Graph Test Case 2 Hasil Deffuzification	53
5.8	Graph Test Case Ketiga dengan Triangular Fuzzy Number	56
5.9	Graph Test Case 3 Hasil Deffuzification	58
5.10	Graph Test Case Keempat dengan Triangular Fuzzy Number	60

5.11	Graph Test Case 4 Hasil Deffuzification	62
5.12	Tampilan antarmuka untuk memasukan input	63
5.13	Tampilan antarmuka untuk memilih start	63
5.14	Tampilan antarmuka untuk menampilkan hasil penelusuran	64
5.15	Tampilan antarmuka Graph yang telah dilakukan penelusuran Dijkstra	65
5.16	Hasil 1	66
5.17	Hasil 2	66
5.18	Hasil 3	67
5.19	Hasil 4	67
5.20	Hasil 5	68
5.21	Kurva Perbandingan jumlah vertex dengan rata-rata waktu	71
B.1	Lampiran Penelusuran Dijkstra Test Case 2 bagian 1	107
B.2	Lampiran Penelusuran Dijkstra Test Case 2 bagian 2	107
C.1	Lampiran Penelusuran Dijkstra Test Case 3 bagian 1	109
C.2	Lampiran Penelusuran Dijkstra Test Case 3 bagian 2	110
C.3	Lampiran Penelusuran Dijkstra Test Case 3 bagian 3	111
C.4	Lampiran Penelusuran Dijkstra Test Case 3 bagian 4	112
C.5	Lampiran Penelusuran Dijkstra Test Case 3 bagian 5	113
C.6	Lampiran Penelusuran Dijkstra Test Case 3 bagian 6	114
D.1	Lampiran Penelusuran Dijkstra Test Case 4 bagian 1	115
D.2	Lampiran Penelusuran Dijkstra Test Case 4 bagian 2	116
D.3	Lampiran Penelusuran Dijkstra Test Case 4 bagian 3	117
D.4	Lampiran Penelusuran Dijkstra Test Case 4 bagian 4	117
D.5	Lampiran Penelusuran Dijkstra Test Case 4 bagian 5	118

DAFTAR TABEL

2.1	Tabel Kebenaran Negasi (\neg)	5
2.2	Tabel Kebenaran Konjungsi (\wedge)	6
2.3	Table Kebenaran Disjungsi (\vee)	6
2.4	Tabel Kebenaran Implikasi (\rightarrow)	6
2.5	Tabel Kebenaran Ekuivalen (\leftrightarrow)	6
2.6	Operasi Himpunan	7
2.7	Fuzzy Matrix	8
2.8	Adjacency List untuk Graph sederhana	15
2.9	Tabel Operasi Dijkstra	20
3.1	Tabel informasi Edge, Fuzzy Number dan nilai Graded Mean Value	24
3.2	Tabel Operasi Dijkstra 2	24
5.1	Tabel Deffuzification Graph Gambar	48
5.2	Hasil Penelusuran Dijkstra	49
5.3	Tabel Deffuzification Graph Gambar	52
5.4	Tabel Deffuzification Graph Gambar	57
5.5	Tabel Deffuzification Graph Gambar	61
5.6	Perbandingan hasil penelusuran Dijkstra untuk Test Case 1	68
5.7	Perbandingan hasil penelusuran Dijkstra untuk Test Case 2	68
5.8	Perbandingan hasil penelusuran Dijkstra untuk Test Case 3	69
5.9	Perbandingan hasil penelusuran Dijkstra untuk Test Case 4	70
E.1	Graph 1	119
E.2	Graph 2	119
E.3	Graph 3	120
E.4	Graph 4	120
E.5	Graph 5	120
E.6	Graph 6	121
E.7	Graph 7	121
E.8	Graph 8	121
E.9	Graph 9	122
E.10	Graph 10	122
E.11	Graph 11	122
E.12	Graph 12	123
E.13	Graph 13	123
E.14	Graph 14	123
E.15	Graph 15	124
E.16	Running Time Masing-Masing graph	124
F.1	Graph 1	125
F.2	Graph 2	126
F.3	Graph 3	127

F.4	Graph 4	128
F.5	Graph 5	128
F.6	Graph 6	129
F.7	Graph 7	130
F.8	Graph 8	131
F.9	Graph 9	132
F.10	Graph 10	133
F.11	Graph 11	134
F.12	Graph 12	135
F.13	Graph 13	136
F.14	Graph 14	137
F.15	Graph 15	137
F.16	Running Time Masing-Masing graph	138
G.1	Graph 1	139
G.2	Graph 2	140
G.3	Graph 3	141
G.4	Graph 4	142
G.5	Graph 5	143
G.6	Graph 6	144
G.7	Graph 7	145
G.8	Graph 8	146
G.9	Graph 9	147
G.10	Graph 10	148
G.11	Graph 11	149
G.12	Graph 12	150
G.13	Graph 13	151
G.14	Graph 14	152
G.15	Graph 15	153
G.16	Running Time Masing-Masing graph	154
H.1	Graph 1	156
H.2	Graph 2	157
H.3	Graph 3	158
H.4	Graph 4	159
H.5	Graph 5	160
H.6	Graph 6	161
H.7	Graph 7	162
H.8	Graph 8	163
H.9	Graph 9	164
H.10	Graph 10	165
H.11	Graph 11	166
H.12	Graph 12	167
H.13	Graph 13	168
H.14	Graph 14	169
H.15	Graph 15	170
H.16	Running Time Masing-Masing graph	171
I.1	Graph 1	174
I.2	Graph 2	175
I.3	Graph 3	176
I.4	Graph 4	177

I.5	Graph 5	178
I.6	Graph 6	179
I.7	Graph 7	181
I.8	Graph 8	182
I.9	Graph 9	183
I.10	Graph 10	184
I.11	Graph 11	185
I.12	Graph 12	186
I.13	Graph 13	188
I.14	Graph 14	189
I.15	Graph 15	190
I.16	Running Time Masing-Masing graph	190

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam melakukan aktifitas bepergian dari suatu lokasi menuju lokasi tujuan, biasanya terdapat banyak alternatif jalan yang dapat digunakan menuju lokasi tujuan. Kebanyakan orang biasanya cenderung memilih jalur yang jarak tempuhnya kecil karena semua alternatif jalan yang ada pasti akan mengarah kepada satu tujuan dan juga dengan memilih jarak tempuh yang kecil akan menghasilkan waktu tempuh yang kecil pula. Untuk pencarian jalur terpendek, sudah banyak ditemukan algoritma untuk menyelesaikan permasalahan tersebut. Beberapa contoh algoritma yang digunakan untuk menyelesaikan jalur terpendek antara lain algoritma *prim*, algoritma *mst kruskal*, algoritma *dijkstra* dan lainnya. Pada skripsi ini akan digunakan algoritma *dijkstra* untuk menyelesaikan permasalahan pencarian jalur terpendek. Sedikit pembahasan mengenai algoritma *dijkstra*, algoritma ini bekerja dengan masukan bilangan riil positif dan pemilihan jalurnya dengan memperhatikan penjumlahan antara jarak sekarang dengan jarak yang sudah ditempuh.

Di Indonesia khususnya di Kota Bandung setiap harinya pasti terjadi aktifitas transportasi yang dilakukan oleh masyarakat Kota Bandung maupun masyarakat luar Kota Bandung untuk melakukan aktifitas baik itu untuk bekerja maupun wisata. Karena Kota Bandung merupakan salah satu kota yang memiliki objek wisata terkenal dan juga sekarang sudah banyak masyarakat Kota Bandung yang memiliki kendaraan pribadi dan hampir setiap hari digunakan bepergian di dalam Kota Bandung, sering terjadi kemacetan di ruas-ruas jalan tertentu. Kemacetan inilah salah satu faktor terhambatnya kelancaran dalam bertransportasi di Kota Bandung. Beberapa faktor lainnya yang mempengaruhi terhambatnya sebuah transportasi selain kemacetan adalah terjadinya kecelakaan, bencana alam, kerusakan jalan dan lainnya. Apabila faktor-faktor tadi diimplementasikan pada saat sedang melakukan pencarian rute terpendek menggunakan algoritma *dijkstra*, maka sudah tidak efektif lagi apabila diambil penjumlahan jarak tempuh terpendek karena belum tentu jarak tempuh terpendek menghasilkan waktu tempuh yang cepat pula. Karena faktor-faktor penghambat transportasi tadi, maka pada skripsi ini akan digunakan waktu untuk menentukan pencarian jalur terpendek. Dalam merepresentasikan ketidakpastian waktu tempuh yang di dapat dari pemilihan sebuah jalur alternatif akan di gunakan *fuzzy number*. *fuzzy number* terdiri dari batasan-batasan yang menghasilkan *fuzzy set*. Karena algoritma *dijkstra* membutuhkan bilangan riil positif sedangkan *fuzzy number* menghasilkan batasan-batasan *fuzzy set*, maka perlu dilakukan proses modifikasi dari *fuzzy number* menjadi sebuah bilangan riil positif.

1.2 Rumusan Masalah

Rumusan masalah yang di dapatkan dari penjabaran masalah di atas adalah sebagai berikut :

1. Bagaimana caranya merepresentasikan waktu dalam pencarian rute terpendek dengan *fuzzy number*?
2. Bagaimana caranya menggunakan *fuzzy number* dalam Algoritma *Dijkstra* sehingga algoritma dapat digunakan untuk pencarian rute terpendek?

3. Bagaimana cara membangun perangkat lunak untuk pencarian rute terpendek menggunakan Algoritma *Dijkstra* dalam lingkungan yang bersifat *fuzzy*?

1.3 Tujuan

Tujuan dibuatnya skripsi ini adalah untuk:

1. Mempelajari cara merepresentasikan waktu dalam pencarian rute terpendek dengan *fuzzy number*.
2. Mempelajari cara menggunakan *fuzzy number* dalam Algoritma *Dijkstra* untuk pencarian rute terpendek.
3. Membangun perangkat lunak pencarian rute terpendek menggunakan Algoritma *Dijkstra* pada lingkungan yang bersifat *fuzzy*.
4. Melakukan pengujian pengaruh jumlah *vertex* terhadap lama waktu untuk menyelesaikan pencarian jalur terpendek menggunakan algoritma *dijkstra*.

1.4 Batasan Masalah

1. Kendaraan
Diasumsikan *user* selalu menggunakan kendaraan mobil.
2. Wilayah
Wilayah yang digunakan adalah wilayah Kota Bandung saja.
3. Jalan
Jalan yang digunakan hanya jalan di dalam Kota Bandung saja tidak termasuk jalan tol.
4. Faktor yang diperhitungkan
Faktor yang diperhitungkan dalam pencarian jalur terpendek adalah waktu tempuh yang dimiliki oleh sebuah jalur.

1.5 Metodologi Penelitian

Urutan metodologi yang dilakukan oleh penulis secara umum yaitu :

1. Studi Literatur
Pada tahap ini, akan dilakukan pencarian dan juga mempelajari bahan-bahan yang berkaitan dengan masalah yang dihadapi seperti *fuzzy sets*, *graph* dan algoritma *dijkstra*
2. Analisis Perangkat Lunak
Pada tahap ini sudah mulai melakukan analisis hal-hal apa saja yang dibutuhkan untuk membangun perangkat lunak untuk menyelesaikan masalah yang sedang diteliti.
3. Perancangan Perangkat Lunak
Pada tahap ini dilakukan pembuatan perangkat lunak berdasarkan kebutuhan yang diperoleh dari hasil analisis yang sudah dilakukan pada tahap sebelumnya.
4. Pengujian
Pada tahap ini dilakukan eksperimen menggunakan data riil untuk menguji perangkat lunak.

5. Kesimpulan

Pada tahap ini akan ditarik kesimpulan dari hasil implementasi algoritma *dijkstra* pada lingkungan yang bersifat *Fuzzy*.

1.6 Sistematika Penulisan

1. Bab 1 Pendahuluan

Pada bab 1 berisi latar belakang masalah, rumusan masalah, tujuan, batasan masalah, metodologi penulisan dan sistematika penulisan.

2. Bab 2 Dasar Teori

Pada bab 2 berisi pembahasan mengenai *Fuzzy Sets*, *Graph*, algoritma *Dijkstra* dan cara implementasi *Fuzzy Number* terhadap algoritma *Dijkstra*.

3. Bab 3 Analisis

Pada bab 3 berisi analisis mengenai kebutuhan apa saja yang dibutuhkan untuk membangun perangkat lunak.

4. Bab 4 Perancangan Perangkat Lunak

Pada bab 4 berisi informasi seputar perangkat lunak yang telah dibangun seperti data dan proses yang terjadi.

5. Bab 5 Implementasi dan Pengujian

Pada bab 5 berisi pengujian yang dilakukan terhadap perangkat lunak menggunakan data yang sudah ditentukan.

6. Bab 6 Kesimpulan

Pada bab 6 berisi kesimpulan dan hasil yang dicapai oleh penulis dalam melakukan pembangunan perangkat lunak ini.

