

SKRIPSI

**PEMBANGUNAN PERANGKAT LUNAK MEDIA
PEMBELAJARAN INTERAKTIF UNTUK ALGORITMA
DIJKSTRA**



Fedrian Hermana

NPM: 2014730008

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

UNDERGRADUATE THESIS

**SOFTWARE DEVELOPMENT FOR DIJKSTRA'S
ALGORITHM INTERACTIVE LEARNING MEDIA**



Fedrian Hermana

NPM: 2014730008

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



LEMBAR PENGESAHAN

PEMBANGUNAN PERANGKAT LUNAK MEDIA PEMBELAJARAN INTERAKTIF UNTUK ALGORITMA DIJKSTRA

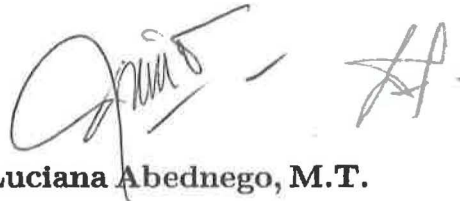
Fedrian Hermana

NPM: 2014730008

Bandung, 6 Desember 2018

Menyetujui,

Pembimbing



Luciana Abednego, M.T.

Ketua Tim Penguji



Mariskha Tri Adithia, P.D.Eng

Anggota Tim Penguji



Pascal Alfadian, M.Comp.

Mengetahui,

Ketua Program Studi



Mariskha Tri Adithia, P.D.Eng



PERNYATAAN

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

PEMBANGUNAN PERANGKAT LUNAK MEDIA PEMBELAJARAN INTERAKTIF UNTUK ALGORITMA DIJKSTRA

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 6 Desember 2018



Fedrian Hermana
NPM: 2014730008

ABSTRAK

Algoritma adalah urutan langkah berhingga untuk memecahkan masalah logika atau matematika. Memahami cara suatu modul algoritma untuk menjalankan aksinya merupakan kemampuan/kecakapan penting yang harus dipahami dan dikuasai karena proses tersebut sangat dibutuhkan pada saat menganalisa, membangun dan mendokumentasikan perangkat lunak. Masalah yang dihadapi saat ini adalah mempelajari sebuah algoritma bukanlah hal yang mudah. Masalah tersebut muncul karena biasanya suatu algoritma disajikan ke dalam bentuk notasi matematika atau kalimat-kalimat logika yang sulit untuk dimengerti oleh orang yang akan menggunakannya.

Pada skripsi ini dibangun sebuah perangkat lunak yang menampilkan visualisasi algoritma. Algoritma yang dipilih untuk divisualisasi adalah Algoritma Dijkstra. Perangkat lunak yang dibuat memodelkan masalah dalam memilih rute terpendek ketika mereka melakukan perpindahan dari suatu tempat ke tempat lain dengan bantuan konsep graf. Dengan menggunakan perangkat lunak tersebut, pengguna dapat belajar bagaimana cara Algoritma Dijkstra bekerja. Perangkat lunak dirancang sedemikian rupa agar penggunaannya dapat dengan mudah mempelajari algoritma tersebut dan mereka dapat menggunakan algoritma tersebut sebagai alat bantu mereka kedepannya dalam mencari rute terpendek.

Berdasarkan pengujian yang telah dilakukan, penggunaan visualisasi sebuah algoritma dapat membantu orang dalam mempelajari sebuah algoritma baru. Pengujian terhadap responden yang mempelajari Algoritma Dijkstra tanpa menggunakan perangkat lunak rata-rata melakukan kesalahan 1,933 kali sedangkan responden yang mempelajari Algoritma Dijkstra dengan bantuan perangkat lunak rata-rata melakukan kesalahan 0,533 kali. Dengan adanya gambar-gambar dan animasi dapat menarik perhatian lebih orang sehingga proses pembelajaran dapat lebih mudah.

Kata-kata kunci: Algoritma, Visualisasi, Graf, Algoritma Dijkstra, Rute Terpendek, Metode Pembelajaran

ABSTRACT

Algorithm is a finite steps for solving logical or mathematical problems. Comprehend in how an algorithm works are important for software analyzing, building, and documenting. Learning how an algorithm works are difficult. The problem occurs because of the explanation for an algorithm usually comes in mathematical or logical notation that hard to learn.

In this undergraduate thesis, a visualization for an algorithm has been made. The software made visualize Dijkstra's Algorithm. The software made visualize the shortest path problem in a graph model. The software helps a user learn how Dijkstra's Algorithm works. Hopefully, the software will help a user to understand Dijkstra's Algorithm easily. Understanding Dijkstra's Algorithm help a user to solve the shortest path problem in real life.

Based on the software testing, a visualization of an algorithm can help a user to learn how an algorithm works. Respondents who learn how Dijkstra's Algorithm works without visualization made 1,933 mistakes on average. On the other hand, respondents who learn how Dijkstra's Algorithm works with visualization only made 0,533 mistakes on average. The visualization contains pictures and animations that increase the attention of respondents and make them easily learn Dijkstra's Algorithm.

Keywords: Algorithm, Visualization, Graph, Dijkstra's Algorithm, Shortest Path, Learning Method

*Dipersembahkan kepada Tuhan Yesus Kristus, keluarga tercinta,
semua teman-teman yang sudah mendukung, semua orang yang
terlibat dalam pembuatan skripsi ini dan diri sendiri*

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas berkat, penyertaan dan perlindungan yang telah diberikan kepada penulis sehingga dapat menyelesaikan skripsi ini yang berjudul Pembangunan Perangkat Lunak Media Pembelajaran Interaktif untuk Algoritma Dijkstra. Penulis juga berterima kasih kepada pihak-pihak yang telah membantu serta memberikan dukungan kepada penulis dalam menyelesaikan skripsi ini. Penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Elias Hermana dan Ibu Titin Sunarti yang selalu memberikan dukungan selama proses pengerjaan skripsi ini.
2. Kakak penulis, Handani Hermana yang selalu memberikan semangat untuk menyelesaikan skripsi ini.
3. Ibu Luciana Abednego dan Bapak Husnul Hakim selaku dosen pembimbing saat penyusunan skripsi.
4. Ibu Mariskha Tri Adithia dan Bapak Pascal Alfadian selaku dosen penguji yang telah memberikan saran dan masukan untuk memperbaiki penyusunan skripsi ini.
5. Kaum Muda dan Remaja Gereja Sidang Pantekosta di Indonesia selaku responden pengujian perangkat lunak yang telah membantu saat proses pengujian perangkat lunak.
6. Andreas Novian, Agina Rinda, Daniel Ferdinan, Kresna Dwi Cahyo, dan Reza Reynaldi selaku responden pengujian yang telah memberikan saran dan penulian terhadap perangkat lunak yang dibangun.
7. Daniel Ferdinan, Stillmen Vallian, Marchella Metta, Kelvin Tandika, Stanley Hanes, Agus Saputra, Ivan, Kevin Jonathan, dan Kresna Dwi Cahyo selaku teman kuliah yang selalu mendukung dan memberikan semangat dalam menyusun skripsi ini.
8. Cechillia Revie, Reina Marchelline, Louisiana Chanata, Laura Chandra, Nelson, Peter, Reinaldo, Rinaldi, Mario Salomo, Leonardo, Alfonsus Evan dan Elvin Hermanawan selaku teman yang selalu memberikan semangat dan dorongan kepada penulis.
9. Keluarga Bapak Tan Yanto Tantra yang sudah memberikan tempat tinggal dan selalu memberikan dukungan kepada penulis.
10. Pihak-pihak lain yang telah membantu penulisan skripsi ini, yang terus memberikan doa dan semangat kepada penulis.

Akhir kata, penulis berharap agar skripsi ini dapat bermanfaat bagi pembaca yang hendak melakukan penelitian dan pengembangan yang terkait dengan skripsi ini.

Bandung, Desember 2018

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 Graf	5
2.1.1 Definisi Graf	5
2.1.2 Terminologi Graf	6
2.1.3 Representasi Graf	10
2.2 Algoritma Dijkstra	13
2.2.1 <i>Pseudocode</i> Algoritma <i>Dijkstra</i>	13
2.3 Bahasa Pemrograman Javascript	14
2.3.1 Struktur Javascript	15
2.4 SVG	16
2.5 CANVAS	18
2.6 Media Pengajaran	19
2.6.1 Visualisasi Algoritma	20
2.7 Desain Pengujian	20
2.7.1 <i>Between Subjects</i>	21
2.7.2 <i>Within Subjects</i>	21
3 ANALISIS	23
3.1 Analisis Masalah	23
3.1.1 Analisis Algoritma Dijkstra	23
3.2 Analisis Perangkat Lunak	32
3.2.1 Kebutuhan Antarmuka	32
3.2.2 Kebutuhan Fungsional	32
3.2.3 Pemodelan Diagram <i>Use Case</i>	32
3.2.4 Pemilihan Metode Pengujian	34
4 PERANCANGAN	35

4.1	Perancangan Perangkat Lunak	35
4.1.1	Flowchart Perangkat Lunak	35
4.1.2	Kebutuhan Masukan	36
4.1.3	Kebutuhan Keluaran	37
4.1.4	Perancangan Modul	38
4.1.5	Pemilihan Graf	39
4.1.6	Perancangan Algoritma	43
4.2	Perancangan Antarmuka	43
5	IMPLEMENTASI DAN PENGUJIAN PERANGKAT LUNAK	47
5.1	Hasil Implementasi	47
5.1.1	Implementasi Perangkat Lunak	47
5.1.2	Implementasi Antarmuka Perangkat Lunak	47
5.2	Pengujian Perangkat Lunak	53
5.2.1	Pengujian Fungsional	54
5.2.2	Kesimpulan Pengujian	65
6	KESIMPULAN DAN SARAN	67
6.1	Kesimpulan	67
6.2	Saran	67
	DAFTAR REFERENSI	69
	A KODE PROGRAM	71
	B DOKUMENTASI PENGUJIAN	123
	C HASIL PENGUJIAN	127

DAFTAR GAMBAR

2.1	Graf tidak berarah(<i>Undirected Graph</i>)	5
2.2	Graf berarah(<i>Directed Graph</i>)	6
2.3	Ketetangaan dalam sebuah graf	6
2.4	Simpul terpencil(<i>Isolated vertex</i>)	7
2.5	Graf Kosong	7
2.6	<i>Degree</i> pada graf tidak berarah	8
2.7	Graf dengan gelang	8
2.8	Graf tidak berarah(<i>Undirected Graph</i>)	8
2.9	Graf tidak terhubung	9
2.10	(<i>Subgraph</i>)	9
2.11	Graf berbobot (<i>Weighted graph</i>)	10
2.12	Pohon (<i>Tree</i>)	10
2.13	<i>Adjacency matrix</i> pada graf tidak berbobot	11
2.14	<i>Adjacency matrix</i> pada graf berbobot	11
2.15	Graf tidak berarah(<i>Undirected Graph</i>)	12
2.16	Graf berbobot untuk representasi <i>linked list</i>	12
2.17	Representasi graf pada Gambar menggunakan metode <i>linked list</i>	13
2.18	Input graf G	14
2.19	Hasil Dijkstra dengan $(G, 1)$	14
2.20	Vektor v yang menghubungkan titik A dan B	16
2.21	Contoh pembuatan graf menggunakan <i>Canvas HTML5</i>	18
2.22	Hasil penggunaan <i>between subjects</i> terhadap 6 orang partisipan	21
2.23	Hasil penggunaan <i>within subjects</i> terhadap 3 orang partisipan	21
3.1	Graf berbobot untuk penghitungan Algoritma Dijkstra	24
3.2	Langkah 1 Algoritma Dijkstra	24
3.3	Langkah 2 Algoritma Dijkstra	24
3.4	Langkah 3 Algoritma Dijkstra	25
3.5	Langkah 4 Algoritma Dijkstra	25
3.6	Langkah 5 Algoritma Dijkstra	25
3.7	Langkah 6 Algoritma Dijkstra	26
3.8	Langkah 7 Algoritma Dijkstra	26
3.9	Graf berarah dengan bobot untuk penghitungan Algoritma Dijkstra	27
3.10	Langkah 1 Algoritma Dijkstra pada graf berarah	27
3.11	Langkah 2 Algoritma Dijkstra pada graf berarah	28
3.12	Langkah 3 Algoritma Dijkstra pada graf berarah	28
3.13	Langkah 4 Algoritma Dijkstra pada graf berarah	29
3.14	Langkah 5 Algoritma Dijkstra pada graf berarah	29
3.15	<i>Flowchart</i> Algoritma Dijkstra	31
3.16	Diagram <i>usecase</i> untuk perangkat lunak	33
4.1	<i>Flowchart</i> untuk perangkat lunak visualisasi Algoritma Dijkstra	35
4.2	Representasi graf pada perangkat lunak	36

4.3	Rancangan desain struktur modul kode program untuk perangkat lunak	38
4.4	Rancangan Graf Tidak Berarah 1	39
4.5	Rancangan Graf Tidak Berarah 2	39
4.6	Rancangan Graf Tidak Berarah 3	40
4.7	Rancangan Graf Tidak Berarah 4	40
4.8	Rancangan Graf Tidak Berarah 5	40
4.9	Rancangan Graf Tidak Berarah 6	41
4.10	Rancangan Graf Tidak Berarah 7	41
4.11	Rancangan Graf Berarah 1	42
4.12	Rancangan Graf Berarah 2	42
4.13	Rancangan Graf Berarah 3	43
4.14	Rancangan antarmuka <i>modal box</i> 1	44
4.15	Rancangan antarmuka <i>modal box</i> 2	44
4.16	Rancangan antarmuka halaman utama	45
5.1	Implementasi <i>modal box</i> pertama	47
5.2	Implementasi <i>modal box</i> kedua sebagai katalog graf	48
5.3	Implementasi halaman utama dari perangkat lunak	49
5.4	Hasil implementasi graf 1 untuk perangkat lunak	50
5.5	Hasil implementasi graf 2 untuk perangkat lunak	51
5.6	Hasil implementasi graf 3 untuk perangkat lunak	51
5.7	Hasil implementasi graf 4 untuk perangkat lunak	51
5.8	Hasil implementasi graf 5 untuk perangkat lunak	52
5.9	Hasil implementasi graf 6 untuk perangkat lunak	52
5.10	Hasil implementasi graf 7 untuk perangkat lunak	52
5.11	Hasil implementasi graf 8 untuk perangkat lunak	53
5.12	Hasil implementasi graf 9 untuk perangkat lunak	53
5.13	Hasil implementasi graf 10 untuk perangkat lunak	53
5.14	Tampilan visualisasi Algoritma Dijkstra berbasis <i>web</i>	54
5.15	Pengujian tombol pada perangkat lunak	55
5.16	Pengujian tombol 2 pada perangkat lunak	55
5.17	Pengujian katalog graf pada perangkat lunak	56
5.18	Pengujian batas-batas masukan pada perangkat lunak	56
5.19	Pengujian Algoritma Dijkstra menggunakan graf 1 pada perangkat lunak	57
5.20	Pengujian Algoritma Dijkstra menggunakan graf 2 pada perangkat lunak	58
5.21	Pengujian Algoritma Dijkstra menggunakan graf 3 pada perangkat lunak	58
5.22	Pengujian Algoritma Dijkstra menggunakan graf 4 pada perangkat lunak	59
5.23	Pengujian Algoritma Dijkstra menggunakan graf 5 pada perangkat lunak	59
5.24	Pengujian Algoritma Dijkstra menggunakan graf 6 pada perangkat lunak	60
5.25	Pengujian Algoritma Dijkstra menggunakan graf 7 pada perangkat lunak	60
5.26	Pengujian Algoritma Dijkstra menggunakan graf 8 pada perangkat lunak	61
5.27	Pengujian Algoritma Dijkstra menggunakan graf 9 pada perangkat lunak	61
5.28	Pengujian Algoritma Dijkstra menggunakan graf 10 pada perangkat lunak	62
5.29	Rentang usia responden pengujian	63
5.30	Perbandingan jumlah kesalahan pada pengujian	64
B.1	Dokumentasi responden saat mencoba perangkat lunak	123
B.2	Dokumentasi pengujian perangkat lunak terhadap responden	124
B.3	Dokumentasi pengujian perangkat lunak oleh Kresna Dwi Cahyo,S.T.	124
B.4	Dokumentasi pengujian perangkat lunak oleh Andreas Novian,S.T.	125
B.5	Dokumentasi pengujian perangkat lunak oleh Agina Rinda dan Reza Reynaldi	125

DAFTAR TABEL

2.1	Representasi <i>adjacency list</i>	12
3.1	Tabel hasil akhir perhitungan Algoritma Dijkstra pada graf tidak berarah	26
3.2	Tabel proses perhitungan Algoritma Dijkstra pada graf tidak berarah	27
3.3	Tabel hasil akhir perhitungan Algoritma Dijkstra pada graf berarah	30
3.4	Tabel proses perhitungan Algoritma Dijkstra pada graf berarah	30
4.1	Matriks bobot untuk graf pada Gambar ??	37
5.1	Tabel hasil perhitungan manual Algoritma Dijkstra dari graf Gambar 5.19	57
5.2	Tabel hasil pengujian Algoritma Dijkstra pada kelompok 1	63
5.3	Tabel hasil pengujian Algoritma Dijkstra pada kelompok 2	64
5.4	Tabel hasil penilaian perangkat lunak Algoritma Dijkstra	65
5.5	Saran atas perangkat lunak Algoritma Dijkstra menurut responden	65
C.1	Tabel respon dari responden saat pengujian eksperimental	127

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Algoritma adalah urutan langkah berhingga untuk memecahkan masalah logika atau matematika. Memahami cara suatu modul algoritma dalam menjalankan aksinya merupakan kemampuan/kecakapan penting yang harus dipahami dan dikuasai karena proses tersebut sangat dibutuhkan pada saat menganalisis, membangun dan mendokumentasikan perangkat lunak [1]. Suatu algoritma memiliki peran yang sangat penting dalam kehidupan suatu program. Dengan algoritma yang sesuai, suatu program dapat berjalan dengan tepat dan dapat mencapai target pekerjaan yang akan dicapai oleh program tersebut. Masalah yang dihadapi saat ini adalah mempelajari sebuah algoritma bukanlah hal yang mudah. Masalah tersebut muncul karena biasanya suatu algoritma disajikan ke dalam bentuk notasi matematika atau kalimat-kalimat logika yang sulit untuk dimengerti oleh orang yang akan menggunakannya.

Secara umum, struktur data dan algoritma adalah masalah inti dari Pendidikan Informatika [2]. Ketika seseorang tidak memahami sebuah algoritma dan struktur data dari pekerjaan yang sedang dikerjakan, tujuan dari pekerjaan yang sedang dilakukan akan sulit untuk mencapai target. Pada awalnya, orang mempelajari algoritma dengan cara membaca secara langsung atau memodelkan algoritma tersebut menjadi sebuah *flow chart*. Cara belajar semacam itu akan memakan waktu yang cukup lama. Selain itu, tidak semua orang dapat dengan mudah mengerti alur dari algoritma tersebut. Hal tersebut menyebabkan proses dalam menyelesaikan sebuah pekerjaan akan menjadi lebih lama. Oleh karena itu, diperlukan sebuah media yang dapat membantu orang untuk mempermudah proses belajar sebuah algoritma.

Semakin berkembangnya teknologi yang ada, media penunjang yang digunakan sebagai alat bantu pembelajaran semakin beragam. Salah satu media yang saat ini digunakan adalah dengan memvisualisasikan ilmu/algoritma tersebut ke dalam bentuk visual yang dinamis. Memvisualisasikan algoritma ke dalam bentuk visual dapat menarik minat belajar dan mempermudah orang dalam mempelajari algoritma.

Dalam skripsi ini akan dibangun sebuah perangkat lunak yang akan menampilkan visualisasi algoritma. Algoritma yang dipilih untuk divisualisasikan adalah Algoritma Dijkstra. Perangkat lunak yang dibuat akan memodelkan masalah dalam memilih rute terpendek ketika mereka melakukan perpindahan dari suatu tempat ke tempat lain. Masalah pemilihan rute terpendek akan diwakilkan menggunakan teori graf. Graf yang akan digunakan pada perangkat lunak adalah graf berarah dan graf tidak berarah tanpa ada *loop* dan simpul terpencil/*isolated vertex*.

Pada pembangunan perangkat lunak ini, pengguna diharapkan dapat belajar bagaimana cara Algoritma Dijkstra bekerja. Perangkat lunak yang dibangun akan dirancang sedemikian rupa agar penggunaanya dapat dengan mudah mempelajari algoritma tersebut dan mereka dapat menggunakan algoritma tersebut sebagai alat bantu mereka kedepannya dalam mencari rute terpendek. Pengujian perangkat lunak ini akan menggunakan metode *between subjects* yang dilakukan untuk mengetahui performa dan pencapaian tujuan pembangunan perangkat lunak ini. Pengujian yang dilakukan akan melibatkan responden secara langsung agar mendapatkan hasil yang maksimal.

1.2 Rumusan Masalah

Pada Skripsi ini terdapat beberapa masalah yang akan menjadi fokus utama, yaitu:

1. Bagaimana cara Algoritma Dijkstra untuk mencari rute terpendek ?
2. Bagaimana cara mengimplementasikan Algoritma Dijkstra dalam bentuk perangkat lunak sehingga dapat membantu pengguna untuk mempelajari algoritma tersebut ?

1.3 Tujuan

Skripsi ini bertujuan untuk :

1. Mempelajari cara kerja dari Algoritma Dijkstra
2. Membangun perangkat lunak untuk membantu pengguna untuk mempelajari Algoritma Dijkstra

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah pengguna tidak dapat membuat model graf baru

1.5 Metodologi

Metodologi penelitian yang akan digunakan meliputi langkah-langkah sebagai berikut :

1. Melakukan studi literatur mengenai Algoritma Dijkstra
2. Melakukan studi literatur mengenai desain pengujian *within subjects* dan *between subjects*
3. Mempelajari metode pembelajaran masa kini
4. Mempelajari bahasa pemrograman Javascript
5. Mempelajari SVG dan *canvas* pada HTML
6. Menentukan contoh graf yang dapat mewakili semua jenis graf yang ada
7. Melakukan analisis kebutuhan perangkat lunak
8. Membuat perancangan perangkat lunak
9. Mengimplementasikan hasil rancangan perangkat lunak yang sudah dibuat
10. Melakukan pengujian dan eksperimen yang melibatkan responden
11. Menulis dokumen skripsi

1.6 Sistematika Pembahasan

Sistematika pembahasan dibagi ke dalam beberapa bab yang akan dijelaskan sebagai berikut :

1. Bab 1 - Pendahuluan
Bab 1 berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

2. Bab 2 - Dasar Teori
Bab 2 berisi tentang teori-teori tentang Graf, bahasa pemrograman Javascript , *canvas* dan SVG pada HTML, Algoritma Dijkstra, metode pembelajaran dan cara pengujian.
3. Bab 3 - Analisis Perangkat Lunak
Bab 3 berisi tentang analisis masalah, analisis kebutuhan perangkat lunak, contoh kasus pencarian rute terpendek dari sebuah graf dengan Algoritma Dijkstra beserta *flowchart*-nya dan pembahasan diagram *use case* untuk perangkat lunak.
4. Bab 4 - Perancangan Perangkat Lunak
Bab 4 berisi tentang perancangan perangkat lunak, perancangan antarmuka, perancangan graf yang akan digunakan dan perancangan algoritma.
5. Bab 5 - Implementasi dan Pengujian perangkat lunak
Bab 5 berisi tentang hasil implementasi perangkat lunak dan pengujian perangkat lunak.
6. Bab 6 - Kesimpulan dan Saran
Bab 6 berisi tentang kesimpulan akhir yang diperoleh serta saran-saran yang diperlukan untuk pengembangan yang lebih lanjut.