



**ANALISIS METODE HAVERSINE DALAM MENENTUKAN
PENCARIAN RUTE TERDEKAT PADA GRAPH**

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH:

NAMA : SUTOYO
NPM : 1624371084
PROGRAM STUDI : SISTEM KOMPUTER

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

LEMBAR PENGESAHAN

**ANALISIS METODE HAVERSINE DALAM MENENTUKAN
PENCARIAN RUTE TERDEKAT PADA GRAPH**

Disusun Oleh:

NAMA : SUTOYO
NPM : 1624371084
PROGRAM STUDI : SISTEM KOMPUTER

**Skripsi Telah Disetujui oleh Dosen Pembimbing Skripsi
Pada Tanggal 05 November 2019**

Dosen Pembimbing I



Andysah P. Siahaan, S.Kom., M.Kom., Ph.D.

Dosen Pembimbing II



Zulham Sitorus, S.Kom., M.Kom.

Mengetahui:

Dekan Fakultas Sains dan Teknologi



Sri Standi Indira, S.T., M.Sc.

Ketua Program Studi Sistem Komputer



Eko Hariyanto, S.Kom., M.Kom.



UNIVERSITAS PEMBANGUNAN PANCA BUDI FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km 4,5 Medan Fax. 061-8458077 PO.BOX : 1099 MEDAN

PROGRAM STUDI TEKNIK ELEKTRO	(TERAKREDITASI)
PROGRAM STUDI TEKNIK ARSITEKTUR	(TERAKREDITASI)
PROGRAM STUDI SISTEM KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI TEKNIK KOMPUTER	(TERAKREDITASI)
PROGRAM STUDI AGROTEKNOLOGI	(TERAKREDITASI)
PROGRAM STUDI PETERNAKAN	(TERAKREDITASI)

PERMOHONAN MENGAJUKAN JUDUL SKRIPSI

yang bertanda tangan di bawah ini :

Nama Lengkap	: SUTOYO
Tanggal/Tgl. Lahir	: SIKARA KARA 1 / 12 November 1995
Nomor Pokok Mahasiswa	: 1624371084
Program Studi	: Sistem Komputer
Spesialisasi	: Sistem Kendali Komputer
SKS Kredit yang telah dicapai	: 133 SKS, IPK 3.17

yang ini mengajukan judul skripsi sesuai dengan bidang ilmu, dengan judul:

Judul SKRIPSI	Persetujuan
Teknik Penajaman Citra Digital Dengan Menggunakan Metode Contrast Stretching	<input type="checkbox"/>
Implementasi Penyembunyian Dan Penyandian Pesan Pada Citra Menggunakan Algoritma Affine Cipher	<input checked="" type="checkbox"/>
Analisis Metode Haversine Dalam Menentukan Pencarian Rute Terdekat Pada Graph	<input checked="" type="checkbox"/> 18/10/2018

yang disetujui oleh Kepala Program Studi diberikan tanda



Rektor I,
[Signature]
(Ir. Bhakti Alamisyah, M.T., Ph.D.)

Medan, 18 Oktober 2018
Pemohon,
[Signature]
(Sutoyo)

Nomor :
Tanggal :
Disahkan oleh :
Dekan
[Signature]
(Sri Shindi Indira, S.T., M.Sc.)

Tanggal :
Disetujui oleh :
Dosen Pembimbing I :
[Signature]
(Anisya Purnama Siala)

Tanggal : 10 Oktober 2018
Disetujui oleh :
Kep. Prodi Sistem Komputer
[Signature]
(MUHAMMAD IQBAL, S.Kom., M.Kom.)

Tanggal : 19-10-18
Disetujui oleh :
Dosen Pembimbing II :
[Signature]
(Sutedjo Sitangir)



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Pembimbing I : Andy Sah Putra Utama Sahaan, S.kom, M.com
 Pembimbing II : Zulham Sitorus, S.kom, M.com
 Mahasiswa : SUTOYO
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1624371084
 Tingkat Pendidikan : Strata Satu (S1)
 Tugas Akhir/Skripsi : Analisis Metode Heuristik Dalam Menentukan Pencarian Rute Terdekat pada Graph

WAKTU	PEMBAHASAN MATERI	PARAF	KETERANGAN
1/12	Rum Bs I		
1/12	Rum Bs II		
1/12	Rum Bs II, III		
1/1	Rum Bs III, Revisi Judul		
1/2	Rum Bs III, IV		
3	Rum Bs IV, V		
3	Acc Semm		
0	Acc Sidang		
11	Acc Jilid		

Medan, 27 November 2018

Diketahui/Ditetujui oleh :
 Dekan,



Sri Shindi Indira, S.T., M.Sc.



UNIVERSITA'S PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
 website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
 Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
 Fakultas : SAINS & TEKNOLOGI
 Pembimbing I : Andysah Putera Utama Siahaan, S.kom, M.com
 Pembimbing II : Zulham Sitorus, S.kom, M.com
 Mahasiswa : SUTOYO
 Jurusan/Program Studi : Sistem Komputer
 Nomor Pokok Mahasiswa : 1624371084
 Program Pendidikan : Strata Satu (S1)
 Tugas Akhir/Skripsi : Analisis Metode Haversine Dalam Menentukan pencarian rute Terdekat pada Graph

NO	PEMBAHASAN MATERI	PARAF	KETERANGAN
19.	Analisa dan perulia sctk konsep ts di muncy di prosedur apa masalah. $\bar{L}, \bar{U}, \bar{S}$		
19.	Des. layout ke TV-2 Bant Ceeli Antu.		
19.	Course & prosedur apa masalah.		
19.	Analisa dan aplikasi di senil dan. upkenti		
19.	Revisi, leplapi berlus.		
19.	Des. untkl melakukan Senre hasil.		

Medan, 27 November 2018
 Diketahui/Disetujui oleh :
 Dekan,



Sri Shindi Indira, S.T., M.Sc.



UNIVERSITAS PEMBANGUNAN PANCA BUDI
FAKULTAS SAINS & TEKNOLOGI

Jl. Jend. Gatot Subroto Km. 4,5 Telp (061) 8455571
website : www.pancabudi.ac.id email: unpab@pancabudi.ac.id
Medan - Indonesia

Universitas : Universitas Pembangunan Panca Budi
Fakultas : SAINS & TEKNOLOGI
Pembimbing I : Andysah Putera Utama Sahaan S.kom, M.kom
Pembimbing II : Zulham Sitarus S.kom, M.kom
Mahasiswa : SUTOYO
Jurusan/Program Studi : Sistem Komputer
No Pokok Mahasiswa : 1624371084
Tingkat Pendidikan : Strata Satu (S1)
Tugas Akhir/Skripsi : Analisis metode Haversine Dalam Menentukan Pencarian rute Terdekat Pada Graph

NO	PEMBAHASAN MATERI	PARAF	KETERANGAN
10 - 19	Apa sedang mija lujan		
20 - 19	Ace jid lra. Nepsi		

Medan, 22 Oktober 2019
Diketahui/Disetujui oleh :
Dekan



Sri Shindi Indira, S.T., M.Sc.

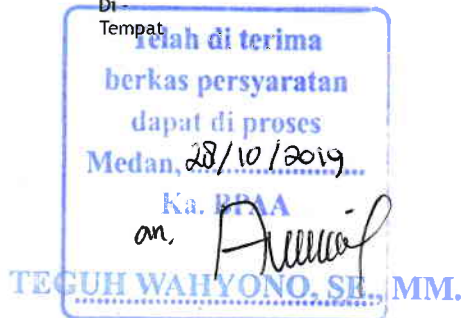
Telah Diperiksa oleh LPMU dengan Plagiarisme...51...%

FM-BPAA-2012-041

Hal : Permohonan Meja Hijau



Medan, 23 Oktober 2019
Kepada Yth : Bapak/Ibu Dekan
Fakultas SAINS & TEKNOLOGI
UNPAB Medan



Dengan hormat, saya yang bertanda tangan di bawah ini :

Nama : SUTOYO
Tempat/Tgl. Lahir : SIKARA-KARA 1 / 12 NOVEMBER 1995
Nama Orang Tua : SUWITO
N. P. M : 1624371084
Fakultas : SAINS & TEKNOLOGI
Program Studi : Sistem Komputer
No. HP : 082165771575
Alamat : Jl. Keluarga Gg Jadi

Datang bermohon kepada Bapak/Ibu untuk dapat diterima mengikuti Ujian Meja Hijau dengan judul Analisis Metode Haversine Dalam Menentukan Pencarian Rute Terdekat Pada Graph, Selanjutnya saya menyatakan :

- 1. Melampirkan KKM yang telah disahkan oleh Ka. Prodi dan Dekan
- 2. Tidak akan menuntut ujian perbaikan nilai mata kuliah untuk perbaikan indek prestasi (IP), dan mohon diterbitkan ijazahnya setelah lulus ujian meja hijau.
- 3. Telah tercap keterangan bebas pustaka
- 4. Terlampir surat keterangan bebas laboratorium
- 5. Terlampir pas photo untuk ijazah ukuran 4x6 = 5 lembar dan 3x4 = 5 lembar Hitam Putih
- 6. Terlampir foto copy STTB SLTA dilegalisir 1 (satu) lembar dan bagi mahasiswa yang lanjutan D3 ke S1 lampirkan ijazah dan transkripnya sebanyak 1 lembar.
- 7. Terlampir pelunasan kwintasi pembayaran uang kuliah berjalan dan wisuda sebanyak 1 lembar
- 8. Skripsi sudah dijilid lux 2 examplar (1 untuk perpustakaan, 1 untuk mahasiswa) dan jilid kertas jeruk 5 examplar untuk penguji (bentuk dan warna penjilidan diserahkan berdasarkan ketentuan fakultas yang berlaku) dan lembar persetujuan sudah di tandatangi dosen pembimbing, prodi dan dekan
- 9. Soft Copy Skripsi disimpan di CD sebanyak 2 disc (Sesuai dengan Judul Skripsinya)
- 10. Terlampir surat keterangan BKKOL (pada saat pengambilan ijazah)
- 11. Setelah menyelesaikan persyaratan point-point diatas berkas di masukan kedalam MAP
- 12. Bersedia melunaskan biaya-biaya uang dibebankan untuk memproses pelaksanaan ujian dimaksud, dengan perincian sbb :

1. [102] Ujian Meja Hijau	: Rp.	750.000
2. [170] Administrasi Wisuda	: Rp.	1.500.000
3. [202] Bebas Pustaka	: Rp.	100.000
4. [221] Bebas LAB	: Rp.	5.000
Total Biaya	: Rp.	1.605.000
5. Uang kuliah	Rp	4.200.000
		<u>6.555.000</u>

28/10
Dit.

Ukuran Toga : M



Hormat saya
SUTOYO
1624371084

Catatan :

- 1. Surat permohonan ini sah dan berlaku bila ;
 - a. Telah dicap Bukti Pelunasan dari UPT Perpustakaan UNPAB Medan.
 - b. Melampirkan Bukti Pembayaran Uang Kuliah aktif semester berjalan
- 2. Dibuat Rangkap 3 (tiga), untuk - Fakultas - untuk BPAA (asli) - Mhs.ybs.



Plagiarism Detector v. 1092 - Originality Report:

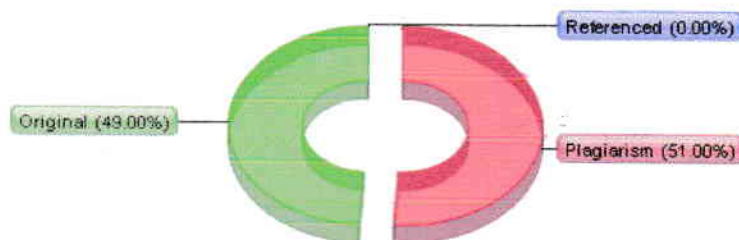
Analyzed document: 24/10/2019 17:11:13

"SUTOYO_1624371084_SISTEM KOMPUTER.docx"

Licensed to: Universitas Pembangunan Panca Budi_License4



Relation chart:



Distribution graph:

Comparison Preset: Rewrite. Detected language: Indonesian

Top sources of plagiarism:

% 10	wrds: 729	https://giocomp.wordpress.com/2014/09/09/materi-kuliah-kecerdasan-buatan-artificial-intell...
% 10	wrds: 665	https://dekapopass2.blogspot.com/2015/06/makalah-dasar-dasar-kecerdasan-buatan.html
% 8	wrds: 642	https://id.123dok.com/document/dzxd94yr-sistem-informasi-geografis-untuk-menentukan-rute-t...

Show other Sources:]

Processed resources details:

259 - Ok / 53 - Failed

Show other Sources:]

Important notes:

Wikipedia:	Google Books:	Ghostwriting services:	Anti-cheating:
[not detected]	[not detected]	[not detected]	[not detected]



YAYASAN PROF. DR. H. KADIRUN YAHYA
UNIVERSITAS PEMBANGUNAN PANCA BUDI
LABORATORIUM KOMPUTER
Jl. Jend. Gatot Subroto Km 4,5 Sei Sikambing Telp. 061-8455571
Medan - 20122

KARTU BEBAS PRAKTIKUM

Yang bertanda tangan dibawah ini Ka. Laboratorium Komputer dengan ini menerangkan bahwa :

Nama : SUTOYO
N.P.M. : 1624371084
Tingkat/Semester : Akhir
Fakultas : SAINS & TEKNOLOGI
Jurusan/Prodi : Sistem Komputer

Benar dan telah menyelesaikan urusan administrasi di Laboratorium Komputer Universitas Pembangunan Panca Budi Medan.

Medan, 23 Oktober 2019

Ka. Laboratorium



Fachrid Wadly, S. Kom

SURAT PERNYATAAN

Yang Bertanda Tangan Dibawah Ini :

N : SUTOYO
M : 1624371084
at/Tgl. Lahir : SIKARA-KARA I / 12 NOVEMBER 1995
at : Jl. Keluarga Gg Jadi
P : 082165771575
Orang Tua : SUWITO/SRINI
ltas : SAINS & TEKNOLOGI
ram Studi : Sistem Komputer
l : Analisis Metode Haversine Dalam Menentukan Pencarian Rute Terdekat Pada Graph

ama dengan surat ini menyatakan dengan sebenar - benarnya bahwa data yang tertera diatas adalah sudah benar
ai dengan ijazah pada pendidikan terakhir yang saya jalani. Maka dengan ini saya tidak akan melakukan
antutan kepada UNPAB. Apabila ada kesalahan data pada ijazah saya.

ikianlah surat pernyataan ini saya buat dengan sebenar - benarnya, tanpa ada paksaan dari pihak manapun dan
at dalam keadaan sadar. Jika terjadi kesalahan, Maka saya bersedia bertanggung jawab atas kelalaian saya.

Medan, 14 November 2019
Pernyataan
**METERAI
TEMPEL**
FC813BAHF081830590
6000
ENAM RIBU RUPIAH
SUTOYO
1624371084

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Sutoyo
IPM : 1624371084
Trodi : Sistem Komputer
Judul Skripsi : ANALISIS METODE HAVERSINE DALAM MENENTUKAN
PENCARIAN RUTE TERDEKAT PADA GRAPH

Dengan ini menyatakan bahwa :

1. Tugas Akhir/Skripsi saya bukan hasil Plagiat
2. Saya tidak akan menuntut perbaikan nilai indeks Prestasi Kumulatif (IPK) setelah ujian Sidang Meja Hijau
3. Skripsi saya dapat dipublikasikan oleh pihak lembaga, dan saya tidak akan menuntut akibat publikasi tersebut

Demikian pernyataan ini saya perbuat dengan sebenar-benarnya, terima kasih

Medan,

Yang membuat pernyataan



ABSTRAK

SUTOYO

ANALISIS METODE HAVERSINE DALAM MENENTUKAN PENCARIAN RUTE TERDEKAT PADA GRAPH

2019

Haversine adalah persamaan penting pada navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan garis bujur dan garis lintang. Haversine merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Hingga saat ini Haversine tersebut kini sudah mengalami pengembangan yaitu dengan menggunakan rumus sederhana, dimana dengan perhitungan komputer dapat memberikan tingkat presisi yang sangat akurat antar dua titik. Penelitian dengan Haversine akan diterapkan untuk menentukan lokasi terdekat. Graph adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Graph digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi objek-objek agar lebih mudah dimengerti. Pembuatan aplikasi ini menggunakan aplikasi Visual Studio 2010.

Kata Kunci : Haversine, Graph, Rute Terdekat, Kecerdasan Buatan

KATA PENGANTAR

Puji syukur kehadiran Allah SWT, karena dengan berkat dan kasih anugrah-Nya penulis masih diberikan kesehatan sehingga akhirnya skripsi ini dapat diselesaikan dengan baik dan sebagaimana mestinya.

Tugas akhir disusun berdasarkan hasil penelitian yang dilaksanakan pada Tanggal 05 November 2019 dengan judul : ” **ANALISIS METODE HAVERSINE DALAM MENENTUKAN PENCARIAN RUTE TERDEKAT PADA GRAPH** ”.

Dalam kesempatan ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada banyak pihak yang telah membantu dalam penyelesaian penyusunan Tugas Akhir ini. Penulis ingin mengucapkan terima kasih kepada :

1. Saya ucapkan terima kasih untuk kedua orang tua saya Bapak Suwito dan Ibu Sрни beserta keluarga saya tercinta, yang selalu memberikan semangat, do'a, motivasi dan membantu penelitian dari segi moril maupun materil sehingga dapat menyelesaikan penyusunan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E, M.M selaku Rektor Universitas Pembangunan Panca Budi Medan.
3. Rektor I Bapak Ir. Bhakti Alamsyah, M.T, Ph.D.
4. Ibu Sri Shindi Indira, ST., M.Sc., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
5. Bapak Eko Hariyanto, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
6. Bapak Andysah P. U. Siahaan, S.Kom., M.Kom, Ph.D, selaku Dosen Pembimbing I yang telah memberikan arahan dan membimbing dalam penyelesaian skripsi ini.
7. Bapak Zulham Sitorus, S.Kom., M.Kom, selaku Dosen Pembimbing II yang telah memberikan curahan pengetahuan, serta bimbingan dalam penyelesaian skripsi ini.
8. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

9. Seluruh staff dan karyawan pada Universitas Pembangunan Panca Budi Medan.

10. Dan seluruh teman-teman penulis dari program studi Sistem Komputer strata satu (S1) Fakultas Ilmu Komputer Universitas Pembangunan Panca Budi Medan

Penulis juga menyadari bahwa penyusunan Tugas Akhir ini belum sempurna baik dalam penulisan maupun isi disebabkan keterbatasan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun dari pembaca untuk kesempurnaan isi Tugas Akhir ini

Medan, 22 Oktober 2019

Penulis

Sutoyo

1624371084

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI.....	4
2.1 Kecerdasan Buatan.....	4
2.1.1 Lingkup Utama Kecerdasan Buatan.....	5
2.1.2 Keuntungan Kecerdasan Buatan	6
2.2 Algoritma Haversine	7
2.3 Graph.....	14
2.4 Visual Basic Net 2010.....	17
2.5 Pengertian UML.....	20
2.5.1 Use Case Diagram	22
2.5.2 Activity Diagram	24
2.5.3 Sequence Diagram.....	25
2.6 Pengertian Flowchart.....	26
BAB III METODE PENELITIAN	29
3.1 Tahapan Penelitian	29
3.2 Metode Pengumpulan Data	30
3.3 Analisis Sistem.....	30
3.4 Analisis Masalah	31
3.5 Analisis Kebutuhan Sistem	31
3.5.1 Kebutuhan Fungsional Sistem.....	32

3.5.2	Kebutuhan Non-Fungsional Sistem.....	32
3.6	Perhitungan Algoritma Haversine.....	33
3.7	Data Vektor.....	34
3.8	Flowchart.....	35
3.8.1	Flowchart Haversine.....	36
3.9	Perancangan Sistem Antarmuka.....	37
3.9.1	Perancangan Jarak Terpendek.....	37
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		38
4.1	Implementasi.....	38
4.1.1	Tampilan Halaman Awal.....	38
4.1.2	Tampilan Menu Buat Koordinat.....	39
4.1.3	Tampilan Menu Pilih Titik Awal.....	40
4.1.4	Tampilan Menu Pilih Titik Akhir.....	41
4.1.5	Tampilan Hasil Rute Terpendek.....	42
4.2	Pengujian Sistem.....	43
4.3	Kelebihan dan Kekurangan Sistem.....	57
BAB V PENUTUP.....		58
5.1	Kesimpulan.....	58
5.2	Saran.....	58
DAFTAR PUSTAKA		
BIOGRAFI PENULIS		

DAFTAR GAMBAR

Gambar 2.1 Contoh Graph Berarah dan Berbobot.....	15
Gambar 2.2 Contoh Graph tidak berarah dan berbobot.....	15
Gambar 2.3 Contoh graph berarah dan tidak berbobot.....	16
Gambar 2.4 Contoh Graph tidak berarah dan tidak berbobot.....	16
Gambar 2.5 Use Case Diagram.....	23
Gambar 2.6 Aktor.....	23
Gambar 2.7 Use Case.....	23
Gambar 2.8 Activity Diagram.....	24
Gambar 2.9 Sequence Diagram.....	25
Gambar 3.1 Tahapan Penelitian.....	29
Gambar 3.2 Diagram Fishbone.....	31
Gambar 3.3 Flowchart Haversine.....	36
Gambar 3.4 Interface Perancangan Jarak Terpendek.....	37
Gambar 4.1 Tampilan Halaman Awal.....	38
Gambar 4.2 Tampilan Input Buat Kordinat.....	39
Gambar 4.3 Tampilan isi titik awal pada bagian Start.....	40
Gambar 4.4 Tampilan Edit Data.....	41
Gambar 4.5 Tampilan hasil rute terpendek.....	42

DAFTAR TABEL

	Halaman
Tabel 2.1 Simbol-Simbol Flowchart	26
Tabel 3.1 Contoh Data Vektor	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan/Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan.

Path Block adalah kompleksitas logik dari suatu prosedur dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan himpunan jalur yang akan diuji. (Abdul Rouf, 2013). Lintasan terpendek merupakan salah satu masalah yang dapat diselesaikan dengan menggunakan graph. Jika diberikan sebuah graph berbobot, masalah lintasan terpendek adalah bagaimana kita mencari sebuah jalur pada graph yang meminimumkan jumlah bobot sisi pembentuk jalur tersebut.

Dalam permasalahan penentuan rute terpendek yang memerlukan waktu yang cukup lama, diperlukan sistem yang dapat menunjukkan suatu lokasi serta menentukan jalur terpendek menuju sebuah lokasi yang dicari. Teknik pencarian yang sering digunakan untuk menentukan jalur terpendek yaitu pencarian buta (*blind search*) dan pencarian heuristik (*heuristic search*). Algoritma *Haversine* berfungsi untuk mencapai tujuan dengan jarak tempuh terdekat, Algoritma *Haversine* memiliki suatu nilai heuristik yang digunakan sebagai dasar pertimbangan dimana estimasi nilai/biaya terkecil yang akan menentukan jarak tempuh terdekat.

Berdasarkan uraian diparagraf sebelumnya, maka penulis tertarik untuk melakukan penulisan skripsi dengan judul **“Implementasi Algoritma Haversine untuk Penentuan Jalur Terpendek”**.

1.2 Rumusan Masalah

Dalam pelaksanaan penelitian ini, adapun masalah yang diangkat, dibahas, dan diselesaikan adalah:

1. Bagaimana menggunakan algoritma Haversine untuk mencari jalur terpendek untuk mencapai tujuan?
2. Bagaimana membangun aplikasi penentu mencari jalur terpendek untuk mencapai tujuan Haversine?
3. Apakah algoritma Haversine lebih baik dan lebih cepat dan tepat untuk memutuskan jalur terpendek?

1.3 Batasan Masalah

Agar pembahasan penelitian ini tidak menyimpang dari apa yang telah dirumuskan, maka diperlukan batasan-batasan. Batasan-batasan dalam penelitian ini adalah:

1. Simulasi sistem menggunakan patch blok kotak persegi.
2. Jumlah panjang blok atau kotak pada simulasi berjumlah 25 dari kiri ke kanan, dan blok atas berjumlah 25 dari atas ke bawah.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mendapatkan jarak terpendek dengan algoritma Haversine .
2. Merancang simulasi yang dapat membantu pemahaman algoritma Haversine dalam pembelajar kecerdasan buatan.
3. Menentukan kelebihan dan kelemahan algoritma Haversine.

1.5 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penulisan skripsi ini adalah sebagai berikut:

1. Memberikan nilai perbandingan pada setiap kotak persegi dalam menentukan jarak terpendek.
2. Menampilkan nilai terkecil pada setiap jarak awal sampai pada jarak akhir sehingga muncul perbandingan antar setiap jalur.

BAB II

LANDASAN TEORI

2.1 Kecerdasan Buatan

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi (Dedi Nugraha, 2013). Hal Ini biasanya dilakukan dengan mengikuti/mencontoh karakteristik dan analogi berpikir dari kecerdasan/Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan (Dedi Nugraha, 2013).

Pengertian lain dari kecerdasan buatan adalah bagian ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat

diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

Agar komputer bisa bertindak seperti dan sebaik manusia, maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Untuk itu AI akan mencoba untuk memberikan beberapa metode untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin pintar (Dedi Nugraha, 2013).

2.1.1 Lingkup Utama Kecerdasan Buatan

- a. Sistem pakar. Komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki

keahlian untuk menyelesaikan masalah dengan meniru keahlian yang dimiliki para pakar

- b. Pengolahan bahasa alami. Dengan pengolahan bahasa alami ini diharapkan user mampu berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
- c. Pengenalan ucapan. Melalui pengenalan ucapan diharapkan manusia mampu berkomunikasi dengan komputer dengan menggunakan suara.
- d. Robotika dan Sistem sensor
- e. Computer vision, mencoba untuk dapat mengintrepetasikan gambar atau objek-objek tampak melalui komputer
- f. Intelligent Computer aid Instruction. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar

2.1.2 Keuntungan Kecerdasan Buatan

- a. Kecerdasan buatan lebih bersifat permanen. Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. Kecerdasan buatan tidak akan berubah sepanjang sistem komputer dan program tidak mengubahnya.
- b. Kecerdasan buatan lebih mudah diduplikasi dan disebar. Mentransfer pengetahuan manusia dari satu orang ke orang lain butuh proses dan waktu lama. Disamping itu suatu keahlian tidak akan pernah bisa diduplikasi secara lengkap. Sedangkan jika pengetahuan terletak pada suatu sistem

komputer, pengetahuan tersebut dapat ditransfer atau disalin dengan mudah dan cepat dari satu komputer ke komputer lain

- c. Kecerdasan buatan lebih murah dibanding dengan kecerdasan alami. Menyediakan layanan komputer akan lebih mudah dan lebih murah dibanding dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.
- d. Kecerdasan buatan bersifat konsisten. Hal ini disebabkan karena kecerdasan busatan adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami senantiasa berubah-ubah.
- e. Kecerdasan buatan dapat didokumentasikan. Keputusan yang dibuat komputer dapat didokumentasikan dengan mudah dengan melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.
- f. Kecerdasan buatan dapat mengerjakan pekerjaan lebih cepat dibanding dengan kecerdasan alami
- g. Kecerdasan buatan dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami.

2.2 Algoritma Haversine

Algoritma Haversine, dapat juga disebut sebagai AlgoritmaHaversine, merupakan salah satu contoh algoritmapencarian yang cukup populer di dunia. Haversine formula merupakan sebuahpersamaan yang penting dalam navigasi, dimanaformula ini memberikan jarak di antara dua titikpada lingkaran bola dari

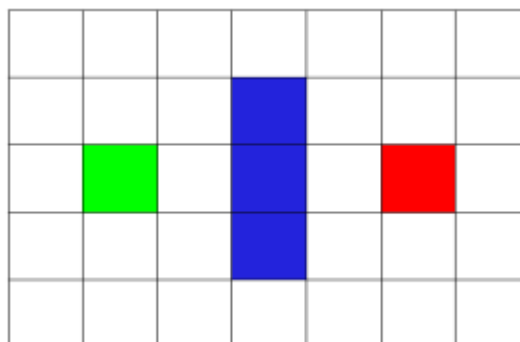
setiap garis bujur(longitude) dan garis lintang (latitude).Ini adalahkasus khusus dari sebuah formula yang lebih umum dalam trigonometri lingkaran bola, law of haversines, haversine formula menghubungkan sisi dan sudut dari sebuah segitiga bola (Harni Kusniyati, 2016).

Jika kita mengetikkan Algoritma Haversine pada sebuah mesin pencari, seperti google.com, maka akan kita temukan lebih dari sepuluh ribu literatur mengenai algoritma Haversine. Beberapa terminologi dasar yang terdapat pada algoritma ini adalah starting point, simpul (nodes), A, open list, closed list, harga (cost), halangan (unwalkable). Starting point adalah sebuah terminologi untuk posisi awal sebuah benda. A adalah simpul yang sedang dijalankan dalam algoritma pencarian jalan terpendek. Simpul adalah petak-petak kecil sebagai representasi dari area pathfinding. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga. Open list adalah tempat menyimpan data simpul yang mungkin diakses dari starting point maupun simpul yang sedang dijalankan. Closed list adalah tempat menyimpan data simpul sebelum A yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan. Harga (F) adalah nilai yang diperoleh dari penjumlahan nilai G, jumlah nilai tiap simpul dalam jalur terpendek dari starting point ke A, dan H, jumlah nilai perkiraan dari sebuah simpul ke simpul tujuan. Simpul tujuan yaitu simpul yang dituju. Rintangan adalah sebuah atribut yang menyatakan bahwa sebuah simpul tidak dapat dilalui oleh A. Prinsip algoritma ini adalah mencari jalur terpendek dari sebuah simpul awal (starting point) menuju simpul tujuan dengan memperhatikan harga (F) terkecil. Diawali dengan menempatkan A pada starting point, kemudian memasukkan seluruh

simpul yang bertetangga dan tidak memiliki atribut rintangan dengan A ke dalam open list. Kemudian mencari nilai H terkecil dari simpul-simpul dalam open list tersebut. Kemudian memindahkan A ke simpul yang memiliki nilai H terkecil. Simpul sebelum A disimpan sebagai parent dari A dan dimasukkan ke dalam closed list. Jika terdapat simpul lain yang bertetangga dengan A (yang sudah berpindah) namun belum termasuk ke dalam anggota open list, maka masukkan simpul-simpul tersebut ke dalam open list. Setelah itu, bandingkan nilai G yang ada dengan nilai G sebelumnya (pada langkah awal, tidak perlu dilakukan perbandingan nilai G). Jika nilai G sebelumnya lebih kecil maka A kembali ke posisi awal. Simpul yang pernah dicoba dimasukkan ke dalam closed list. Hal tersebut dilakukan berulang-ulang hingga terdapat solusi atau tidak ada lagi simpul lain yang berada pada open list (Harni Kusniyati, 2016).

Langkah 1: Area

Berikut adalah contoh simple arena yang akan kita gunakan. Warna hijau adalah starting point, warna merah adalah goal/end point, dan biru adalah penghalang. Goal dari aplikasi ini adalah mencari rute dari titik hijau ke merah tanpa melewati penghalang biru



Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

Langkah 2 : Movement Cost / Biaya Pergerakan

Kita asumsikan setiap langkah dari hijau adalah legal baik vertikal, horizontal, maupun diagonal dengan catatan tidak membentur tembok. Setiap langkah yang diizinkan kita berikan nilai G dimana G adalah cost atau biaya dalam setiap langkah. Dalam kasus ini kita akan berikan nilai 10 untuk setiap langkah vertikal maupun horizontal, dan 14 untuk diagonal. Nilai 14 kita dapatkan dari perhitungan pitagoras dimana $14,1421 = \sqrt{\sqrt{10} + \sqrt{10}}$. Hasil data nilai G ini selanjutnya kita gambarkan sbb :

G=14	G=10	G=14				
G=10		G=10				
G=14	G=10	G=14				

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

Selain dari perhitungan tersebut, kita dapat mengalikan dengan konstanta tertentu untuk memanipulasi biaya, misal : ketika melewati sungai maka $G = G * 2$.

Langkah 3 : Estimated Movement / Estimasi gerakan

Langkah selanjutnya kita hitung biaya estimasi pergerakan dan kita simbolkan dengan H. Nilai H ini secara singkat adalah nilai jarak / estimasi biaya dari

pergerakan dari suatu titik terhadap titik finish dengan mengabaikan penghalang yang ada. Untuk lebih jelasnya silahkan lihat gambar di bawah :

H=60	H=50	H=40				
H=50		H=30	H=20	H=10		
H=60	H=50	H=40				

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

Langkah 4 : Scoring / Penilaian

Setelah nilai G dan H kita dapatkan, maka kita berikan skor dari masing-masing titik yang akan dilalui. Skor kita lambangkan misalnya dengan F dimana nilai $F = G + H$. Nilai F selanjutnya kita masukkan dalam setiap titik dari setiap langkah yang akan dilalui. Untuk lebih jelasnya lihat gambar di bawah :

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40				
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54				

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

Dari setiap nilai tersebut kita ambil keputusan dengan mengambil langkah dengan nilai F terkecil.

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40				
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54				

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

Langkah 5 : Looping / Perulangan

Setelah pergerakan pertama selesai selanjutnya lakukan perulangan dari langkah 1 sampai 4. Untuk lebih jelasnya setiap pergerakan akan digambarkan di bawah :

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40				
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54				
	G = 28 H = 60 F = 88	G = 24 H = 50 F = 74	G = 28 H = 40 F = 68			

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40				
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54		G = 42 H = 20 F = 62		
	G = 28 H = 60 F = 88	G = 24 H = 50 F = 74	G = 28 H = 40 F = 68	G = 38 H = 30 F = 68		

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40				
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54		G = 42 H = 20 F = 62		
	G = 28 H = 60 F = 88	G = 24 H = 50 F = 74	G = 28 H = 40 F = 68	G = 38 H = 30 F = 68		

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 50 F = 54				
G = 10 H = 50 F = 60		G = 10 H = 30 F = 40		G = 52 H = 10 F = 62		
G = 14 H = 60 F = 74	G = 10 H = 50 F = 60	G = 14 H = 40 F = 54		G = 42 H = 20 F = 62	G = 52 H = 10 F = 62	
	G = 28 H = 60 F = 88	G = 24 H = 50 F = 74	G = 28 H = 40 F = 68	G = 38 H = 30 F = 68		

Sumber: (<http://www.policyalmanac.org/games/aHaverSineTutorial.html>)

2.3 Graph

Graph adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat. Graph digunakan untuk menggambarkan berbagai macam struktur yang ada. Tujuannya adalah sebagai visualisasi objek-objek agar lebih mudah dimengerti.

Graph G didefinisikan sebagai pasangan himpunan (V, E) , ditulis dengan notasi $G = (V, E)$, yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (vertices atau node) dan E adalah himpunan sisi (edges atau arcs) yang menghubungkan sepasang simpul [6]. Menurut arah dan bobotnya, graph dibagi menjadi empat bagian yaitu:

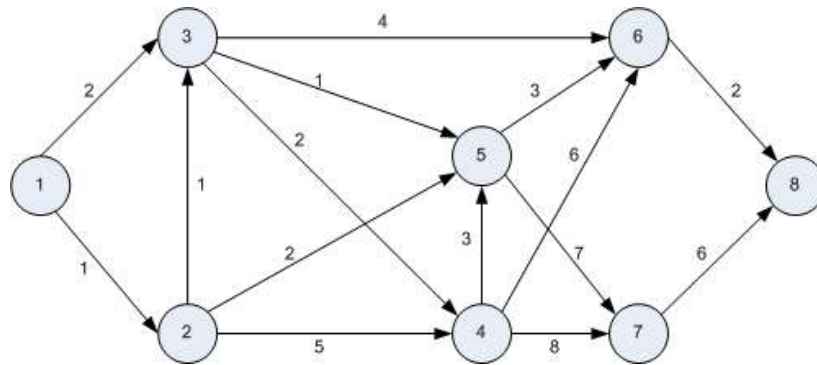
1. Graph berarah dan berbobot adalah graf yang tiap verteks (titiknya) diwakili oleh sebuah anak panah yaitu sebuah kurva berarah dan setiap sisinya memiliki bobot. Sebagai contoh pada gambar ditunjukkan:

$$V(G) = \{1, 2, 3, 4, 5, 6, 7, 8\}$$

$$E(G) = \{[1, 2], [1, 3], [2, 3], [2, 4], [2, 5], [3, 4], [3, 5], [3, 6], [4, 5], [4, 6], [4, 7], [5, 6], [5, 7], [6, 8], [7, 8]\}$$

Sumber: (<https://text-id.123dok.com/document/ozlxkngz4-graph-pencarian-rute-terpendek-dengan-algoritma-dijkstra.html>)

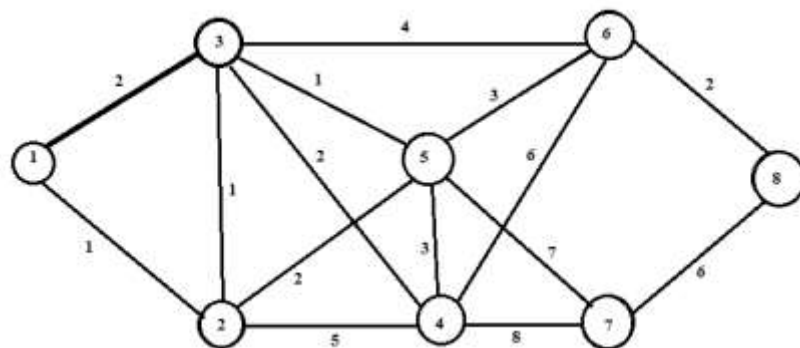
Setiap sisi pada titik graph memiliki bobot yang berbeda-beda.



Gambar 2.1. Contoh Graph Berarah dan Berbobot

Sumber: (<https://text-id.123dok.com/document/ozlxkngz4-graph-pencarian-rute-terpendek-dengan-algoritma-dijkstra.html>)

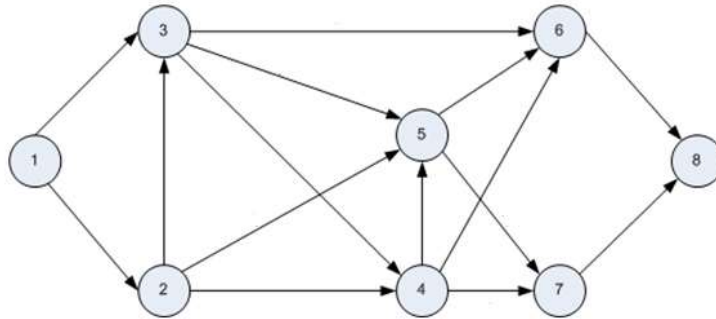
2. Graph tidak berarah dan berbobot setiap sisi graph tidak terdapat anak panah atau sebuah kurva berarah. Disini urutan setiap simpulnya yang terhubung dari setiap verteks tidak diperhatikan. Maka, $E(G) = \{[1,2]\}$ sama dengan $E(G) = \{[2,1]\}$. Dan setiap sisinya memiliki bobot atau nilai yang berbeda.



Gambar 2.2. Contoh Graph tidak berarah dan berbobot

Sumber: (<https://text-id.123dok.com/document/ozlxkngz4-graph-pencarian-rute-terpendek-dengan-algoritma-dijkstra.html>)

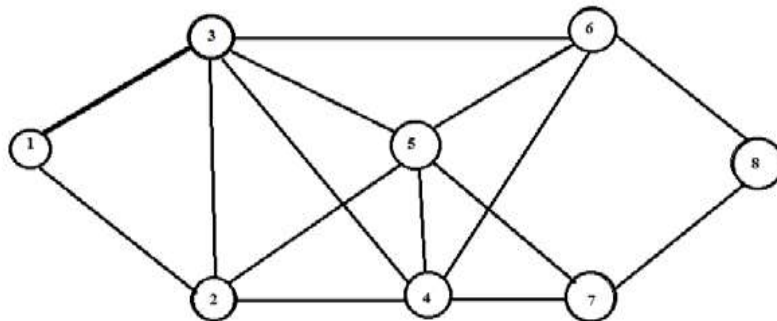
3. Graph berarah dan tidak berbobot adalah graph yang setiap verteksnya diwakili sebuah anak panah yang menunjukkan sebuah arah, tapi setiap sisinya tidak memiliki bobot atau nilai.



Gambar 2.3. Contoh graph berarah dan tidak berbobot

Sumber: (<https://text-id.123dok.com/document/ozlxkngz4-graph-pencarian-rute-terpendek-dengan-algoritma-dijkstra.html>)

4. Graph tidak berarah dan tidak berbobot adalah graph yang setiap verteks tidak diwakili sebuah anak panah dan setiap sisinya tidak memiliki bobot atau nilai.



Gambar 2.4. Contoh Graph tidak berarah dan tidak berbobot.

Sumber: (<https://text-id.123dok.com/document/ozlxkngz4-graph-pencarian-rute-terpendek-dengan-algoritma-dijkstra.html>)

2.4 Visual Basic Net 2010

Merupakan sebuah bahasa pemrograman dan sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. Beberapa kemampuan atau manfaat dari Visual Basic diantaranya (DP Oktavian, 2010:10):

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat obyek-obyek pembantu program, seperti Control Active X, File Help, Aplikasi Internet dan sebagainya.
3. Menguji program (debugging) dan menghasilkan program akhir berakhiran "EXE" yang bersifat executable atau dapat langsung dijalankan.

Keistimewaan utama dari Visual Basic adalah:

4. Menggunakan platform pembuatan program yang diberi nama developer studio, yang memiliki tampilan seperti C++ dan visual J++.
5. Memiliki kompiler handal yang dapat menghasilkan File Executable yang lebih cepat dan efisien.
6. Memiliki tambahan saran wizard yang baru. Tambahan kontrol-kontrol baru dan lebih canggih serta peningkatan kaidah struktur bahasa Visual Basic.
7. Kemampuan membuat Active X dan fasilitas internet yang lebih banyak.
8. Sarana akses yang lebih cepat dan andal untuk membuat aplikasi database yang berkemampuan tinggi.

9. Visual Basic.net memiliki beberapa versi baru edisi yang disesuaikan dengan kebutuhan pemakainya.

Dalam pemrograman berbasis OOP (Object Oriented Programming), sebuah program dibagi menjadi bagian-bagian kecil yang disebut dengan obyek. Setiap obyek memiliki entiti terpisah dengan entiti-entiti lain dalam lingkungannya. Obyek-obyek yang terpisah ini dapat diolah sendiri-sendiri, dan setiap obyek memiliki sekumpulan sifat dan metode yang melakukan fungsi tertentu sesuai dengan yang telah diprogramkan kepadanya (DP Oktavian, 2010:10 – google.book.com).

Adapun obyek-obyek yang dipergunakan dalam program ini adalah:

1. Project

Project adalah sekumpulan modul. Jadi project merupakan aplikasi itu sendiri. Project disimpan dalam file yang berakhiran VBP. Jika kita akan melaksanakan pembuatan program aplikasi, akan terdapat jendela project yang berisi semua file yang dibutuhkan menjalankan program aplikasi Visual Basic.net pada saat pembuatan program aplikasi baru maka jendela project otomatis akan berisi object form1. Pada jendela project terdapat tiga icon yaitu View Code, View Object, dan Toggle Folders. Icon View Code dipakai untuk menampilkan jendela editor kode program. Icon View Object dipakai untuk menampilkan bentuk formulir (form) dan icon Toggle Folders digunakan untuk menampilkan folder

2. Form

Form adalah jendela yang dipakai untuk membuat user interface/tampilan. Secara otomatis akan tersedia form yang baru jika membuat suatu program aplikasi yang baru, dengan nama Form1. pada umumnya dalam suatu form terdapat garis titik-titik yang disebut dengan Grid. Untuk lebih memahami form ini maka di bawah ini terdapat gambar jendela form.

3. Toolbox

Toolbox adalah kumpulan dari obyek yang digunakan untuk membuat user interface (tampilan) serta control bagi program aplikasi. Untuk menempatkan control pada suatu form dapat dilakukan dengan klik ganda control dalam toolbox, kemudian mengubah besar dan ukurannya serta memindahkannya dengan metode Drag and Drop atau dengan cara mengklik kontrol toolbox, kemudian pindahkan pointer mouse jendela form. Kursor berubah menjadi Crosshair lalu tempatkan pada sudut kiri atas dimana kita inginkan kontrol tersebut diletakkan, tekan tombol mouse kiri dan tahan ketika menyeret kursor ke arah sudut kanan bawah.

4. Properties

Properties berisikan daftar struktur setting properti yang digunakan pada sebuah object terpilih. Kotak drop-down pada bagian atas jendela berisi daftar semua object pada form yang aktif. Ada tab tampilan, yaitu alphabetic (urut abjad) dan categorized (urut berdasarkan kelompok).

5. Kode Program

Kode program adalah serangkaian tulisan perintah yang akan dilaksanakan jika suatu obyek dijalankan. Kode program ini mengontrol dan menentukan jalannya suatu obyek.

6. Event

Event adalah peristiwa atau kejadian yang diterima suatu obyek, misalnya klik, seret, tunjuk, dan lain sebagainya.

7. Metode (Methods)

Metode adalah serangkaian perintah yang sudah tersedia pada suatu obyek yang dapat diminta untuk mengerjakan tugas khusus.

8. Module

Module dapat disejajarkan dengan form, tetapi module tidak mengandung obyek. Module berisikan prosedur umum, deklarasi variabel dan definisi konstanta yang digunakan oleh aplikasi.

2.5 Pengertian UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga

menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET (Prastuti Sulistyorini, 2012).

Unified Modeling Language (UML) adalah kumpulan notasi grafis yang didukung oleh sebuah model tunggal, yang membantu dalam menjelaskan dan merancang sistem perangkat lunak, khususnya sistem perangkat lunak dibangun menggunakan gaya berorientasi objek. UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang kita sebut sebuah model (Andy Prasetyo Utomo, 2013).

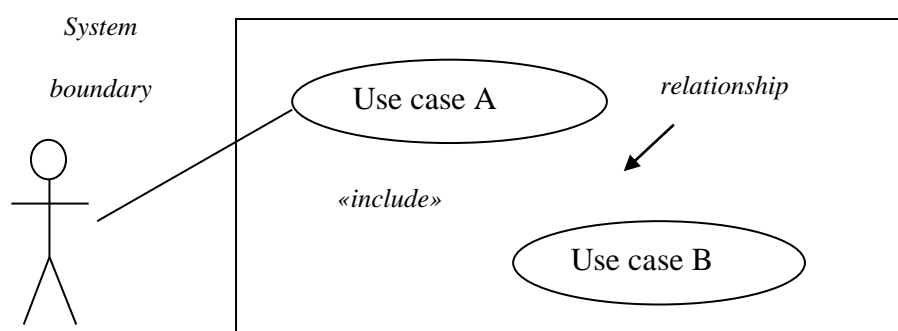
Dengan menggunakan model ini diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk faktor-faktor seperti *scalability*, *robustness*, *security*, dan sebagainya. Untuk melakukan pemodelan sistem perangkat lunak secara visual digunakan UML (*Unified Modelling Language*) yang digambarkan secara elektronik lewat sarana perangkat lunak *Rational Rose*. Sebagai mana telah diterapkan oleh Gufran (2012) di mana UML diterapkan untuk mengukur kinerja mahasiswa menggunakan pendekatan berorientasi objek. Selanjutnya Jakimi dan Koutbi (2009) menerapkan pendekatan UML untuk skenario rekayasa dan kode generasi.

2.5.1 Use Case Diagram

Use case merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Dalam tehnik ini tidak diterangkan cara kerja sistem secara internal maupun implementasinya. Yang ditunjukkan adalah langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak (Nyimas Artina, 2006).

Diagram *Use Case* merupakan diagram yang menggambarkan fungsi berupa komponen, kelas, atau kejadian yang ada dalam *system* (Ade Sutedi *et al*, 2015). *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A.S dan M. Shalahuddin, 2014).

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.



Gambar 2.5. Use Case Diagram

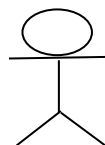
Sumber : (E Triandini, IG Suardika, 2012 – google.book.com)

Terdapat 2 bagian utama dalam *use case modeling* sebagaimana dijelaskan sebagai berikut:

- **Aktor**

Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat

itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.



Gambar 2.6. Aktor

Sumber : (E Triandini, IG Suardika, 2012 – google.book.com)

- *Use Case*

Use case merupakan fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

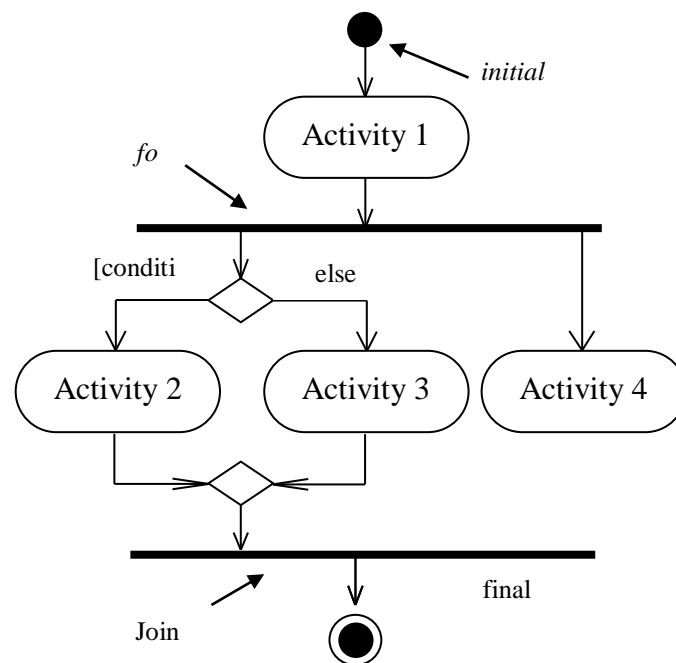


Gambar 2.7. Use Case

Sumber : (E Triandini, IG Suardika, 2012 – google.book.com)

2.5.2 Activity Diagram

Activity diagrams menggambarkan aliran kerja atau aktivitas sari sebuah sistem atau proses bisnis. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa A.S dan M. Shalahuddin, 2014).



Gambar 2.8. Activity Diagram

Sumber : (E Triandini, IG Suardika, 2012 – google.book.com)

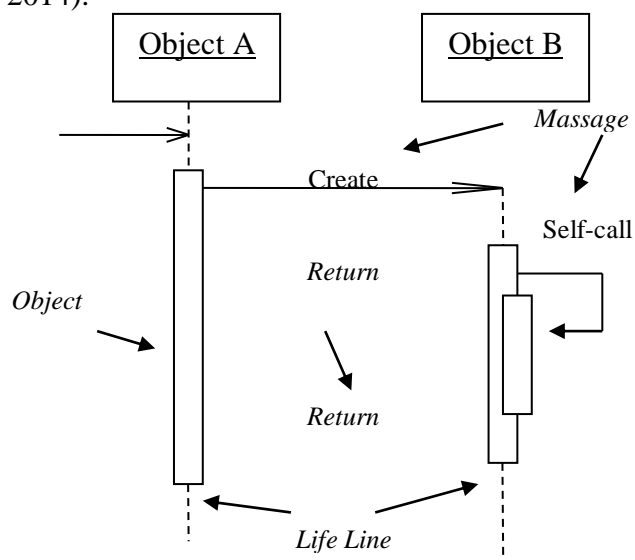
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* di mana setiap aktivitas dianggap memiliki antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

2.5.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa A.S dan M. Shalahuddin, 2014).



Gambar 2.9. Sequence Diagram

Sumber : (E Triandini, IG Suardika, 2012 – google.book.com)

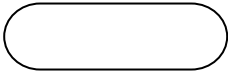



2.6 Pengertian Flowchart


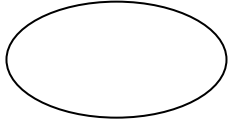
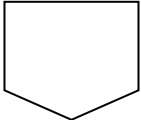

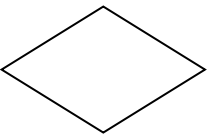
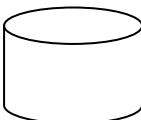
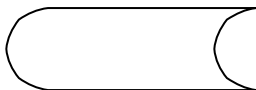


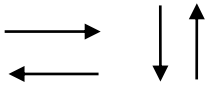
Menurut (Sariadin Siallagan, 2013), Flowchart adalah suatu diagram alir yang mempergunakan simbol atau tanda untuk menyelesaikan masalah. Dalam hal ini, penyelesaian masalah menggunakan simbol-simbol yang telah disepakati.



Menurut (Abdillah Baraja, 2012) Flowchart adalah representasi grafik yang menggambarkan setiap langkah yang akan dilakukan dalam suatu proses, yang merupakan alat bantu yang banyak digunakan untuk menggambarkan sistem secara pisikal.

Bagan alir (flowchart) adalah bagan (chart) yang menunjukkan alir (flow) di dalam program atau prosedur system secara logika. Digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

Tabel 2.1. Simbol-Simbol Flowchart

NO	SIMBOL	FUNGSI
1.		Terminal menyatakan awal atau akhir dari suatu logaritma.
2.		Menyatakan proses.
3.		Proses yang terdefenisi atau sub program.
4.		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.

5.		Menyatakan masukan dan keluaran (input/output).
6.		Menyatakan penyambung ke simbol lain dalam satu halaman.
7.		Menyatakan penyambung ke halaman lainnya.
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>decision</i> (keputusan) yang digunakan untuk penyeleksian kondisi didalam program.
10.		Menyatakan media prnyimpanan drum magnetik.
11.		Menyatakan input/output menggunakan disket.
12.		Menyatakan operasi yang dilakakukan secara manual.
13.		Menyatakan input/output dari kartu plong.
14.		Menyatakan aliran pekerjaan (proses).

15.		Multidocument (banyak dokumen).
16.		Delay (penundaan atau kelambatan).

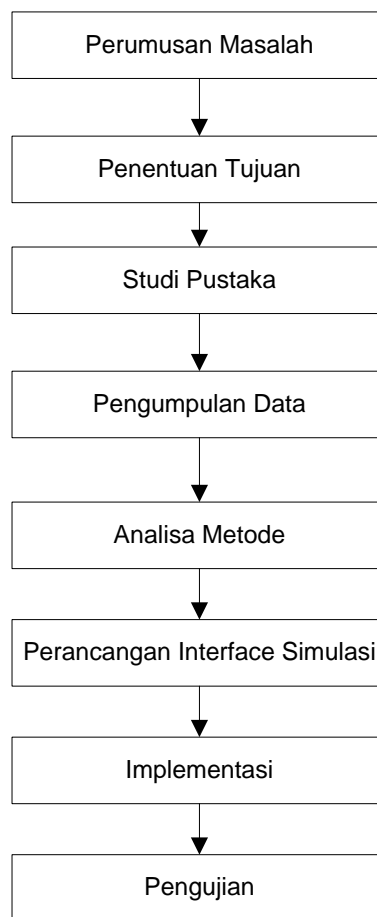
Sumber :Abdillah Baraja, 2012

BAB III

METODE PENELITIAN

1.1 Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan oleh penulis ini dengan judul Analisis Metode Haversine Dalam Menentukan Pencarian Rute Terdekat Pada Graph adalah sebagai berikut:



Gambar 3.1. Tahapan Penelitian

1.2 Metode Pengumpulan Data

Adapun metodologi pengumpulan data pada penelitian ini adalah sebagai berikut:

1. Metode Observasi

Metode observasi adalah metode pengumpulan data dengan melakukan pengamatan langsung terhadap obyek yang diteliti untuk mengumpulkan data dan informasi yang berkaitan dengan permasalahan yang ada. Informasi yang berkaitan adalah proses jalannya sistem tersebut.

2. Metode Studi Literatur

Studi literatur dapat dilakukan dengan mempelajari sumber-sumber buku yang berkaitan melalui membaca buku-buku dari perpustakaan dan mencari manfaat referensi dari internet.

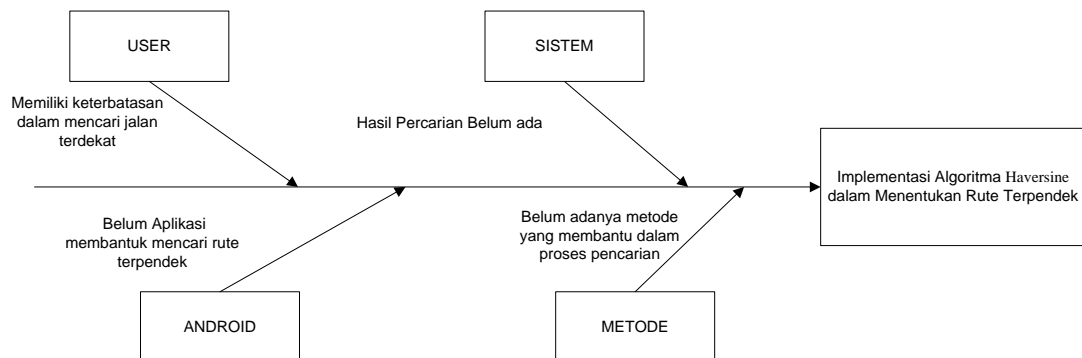
1.3 Analisis Sitem

Dalam perancangan sebuah sistem diperlukan analisis untuk keperluan sistem. Dengan adanya analisis sistem, sistem yang dirancang diharapkan akan lebih baik dan memudahkan dalam pengembangan sistem selanjutnya. Tujuan dari analisa sistem ini sendiri adalah agar sistem yang dirancang menjadi tepat guna dan ketahanan dari sitem tersebut akan lebih terjaga.

Sistem ini akan dapat digunakan oleh siapapun untuk mencari rute terpendek menuju salah satu Tujuan yang ada. Sistem ini dirancang menggunakan Algoritma Haversine untuk mencari retu terpendeknya.

1.4 Analisis Masalah

Masalah utama adalah untuk pencarian rute terpendek menuju lokasi Tujuan masih menggunakan pencarian manual oleh siapapun. Secara umum kemampuan manusia untuk memproses data masih lambat, sehingga untuk menentukan rute terpendek yang ingin dilalui masih kurang efektif. Pencarian rute terpendek menuju lokasi Tujuan ini juga masih belum adanya dukungan dokumentasi yang memadai. Dengan menggunakan metode secara manual juga masih mengalami keterbatasan dalam proses analisisnya, tidak ada panduan yang pasti untuk menganalisis jalur yang tepat. Seluruh masalah tersebut di muat dalam diagram konteks pada gambar 3.2.



Gambar 3.2. Diagram Fishbone

1.5 Analisis Kebutuhan Sistem

Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan. Kebutuhan fungsional dan kebutuhan nonfungsional. Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya

dilakukan oleh sistem. Sedangkan kebutuhan nonfungsional adalah kebutuhan yang menitik beratkan pada properti perilaku yang dimiliki oleh sistem.

1.5.1 Kebutuhan Fungsional Sistem

Menggunakan analisis diharapkan suatu sistem dapat diuraikan secara utuh menjadi komponen-komponen dasar dengan tujuan identifikasi, mengevaluasi permasalahan dan kebutuhan yang diharapkan, dan analisis ini juga dilakukan untuk menjamin bahwa sistem yang dibangun sesuai dengan kebutuhan penelitian.

Hasil analisis kebutuhan sistem adalah :

- a. Sistem dapat memberikan rute terpendek untuk menuju lokasi Tujuan dengan menggunakan algoritma Haversine,
- b. Sistem dapat digunakan setiap siapapun sebagai media untuk memberikan informasi serta lokasi Tujuan yang terdekat.
- c. Sistem dapat memberikan informasi lokasi Tujuan.

1.5.2 Kebutuhan Non-Fungsional Sistem

Analisis non fungsional menunjukkan kesulitan saat pengembangan sistem sehingga sistem tidak terlalu berjalan optimal namun sistem tersebut dapat dijalankan. Adapun kebutuhan non fungsional dari sistem yang dirancang mencakup:

- a. Dari segi performance, sistem ini dirancang:

Tampilan/interface yang menarik

- b. Dari segi delay, sistem ini dirancang dengan waktu tunda yang sesuai dengan waktu tunda dikarenakan proses ini membutuhkan sebuah koneksi internet sehingga waktu loading tergantung dari koneksi internet tersebut seperti untuk menampilkan peta dari hasil pencarian rute terpendek dikarenakan peta yang ditampilkan berasal dari API Google
- c. Dari segi Informasi, sistem ini dirancang :
- Untuk menginformasikan bagaimana cara menambahkan data lokasi.
Digunakan untuk menginformasikan apabila password yang telah dimasukan pengguna salah.
- d. Dari segi keamanan/security, sistem ini dirancang dengan proses login untuk hak akses halaman admin pada web server.

1.6 Perhitungan Algoritma Haversine

Dimisalkan titik awal adalah A, dan titik tujuan atau akhir adalah D, terlebih dahulu ketahui titik A, dan titik apa saja yang dilalui oleh titik A.

$$\text{Titik Awal} = A (22, 29)$$

$$\text{Titik Akhir} = B(39, 34)$$

Untuk menghitung jarak dengan metode Haversine dapat dilihat pada perhitungan berikut ini.

$$H = \text{ACOS}((\text{SIN}(22) * \text{SIN}(39)) + (\text{COS}(22) * \text{COS}(39) * \text{COS}(34-29))) * 6371 /$$

110

$$H = 17,71976493$$

1.7 Data Vektor

Model data vektor diwakili oleh symbol-simbol atau yang selanjutnya dalam Sistem Informasi Geografis dikenal dengan feature, seperti feature titik (point), feature garis (line), feature area (surface) (Budi Santosa, 2012).

Tujuan dari perancangan database server pada aplikasi ini adalah untuk mendata jarak dari titik awal ke titik tujuan. Disamping itu database yang diletakkan pada notepadagar sistem dapat diproses dengan cepat.

Tabel 3.1. Contoh Data Vektor

No.	Titik	X	Y	Cabang
1	A	20	20	B H
2	B	40	20	A C D
3	C	50	58	B D E H X
4	D	63	20	B C F
5	E	77	53	C F G J X
6	F	83	20	D E G V
7	G	96	45	E F J V
8	H	20	86	A C L X
9	I	107	98	J K L N X Z
10	J	100	76	E G I X
11	K	85	146	I L M N Y
12	L	20	144	H I K X Y
13	M	145	200	K N S Y
14	N	140	137	I K M O S
15	O	157	72	N P T
16	P	175	105	O Q R S
17	Q	200	80	P R S
18	R	200	20	P Q T
19	S	200	200	M N P Q
20	T	160	20	O R U W Z
21	U	128	20	T V W
22	V	100	20	F G U W
23	W	115	40	T U V

24	X	67	91	C E H I J L
25	Y	20	200	K L M
26	Z	125	56	I T

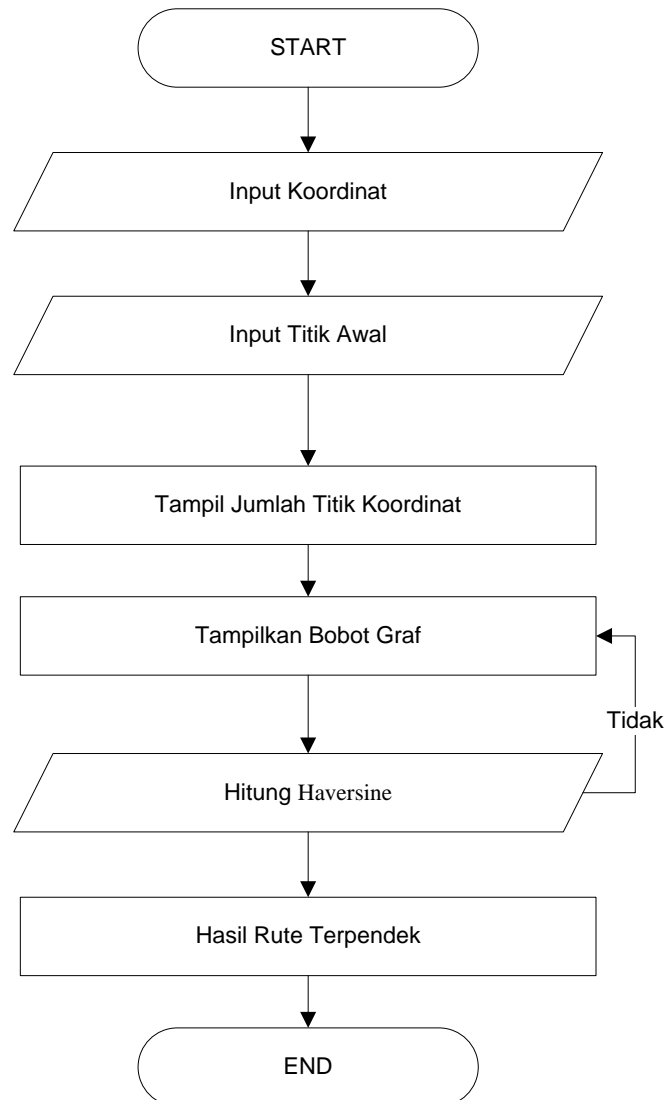
Tabel 3.1 menjelaskan bahwa pada perancangan sistem alamat tujuan Jln. Let.Jend. Jamin Ginting diganti menjadi graf A, dan Jl. Sei Bahorok Kel. Pujidadi diganti menjadi graf B. Adapun tujuan dari pembuatan tabel data vektor ini adalah untuk meminimalisasikan tampilan pada program agar tampilan program menjadi simpel dan cepat.

1.8 Flowchart

Flowchart (Bagan Alir Program) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Dengan kata lain flowchart (bagan alir program) dapat didefinisikan sebagai sebuah bagan yang menunjukkan aliran di dalam program atau prosedur sistem secara logika. Bagan alir sistem menunjukkan apa yang dikerjakan di sistem.

1.8.1 Flowchat Haversine

Berikut ini adalah flowchart untuk menentukan proses perhitungan rute terpendek dengan metode Haversine. Flowchart dapat dilihat pada Gambar 3.3.

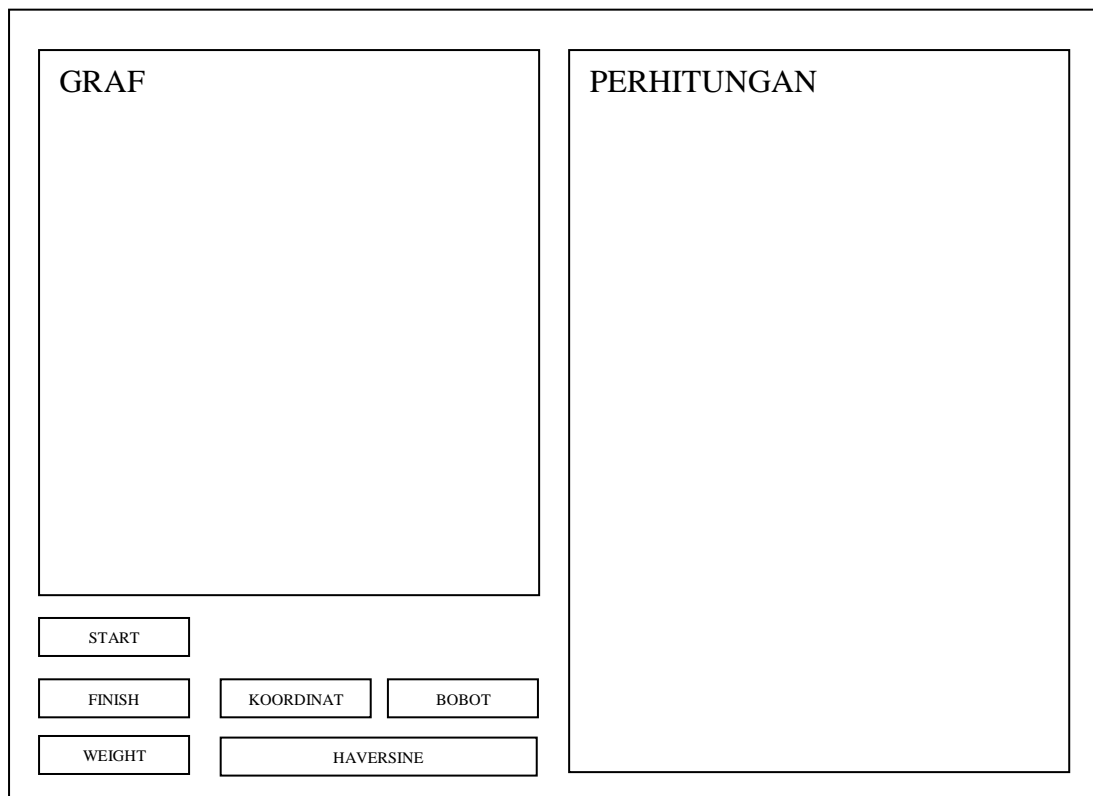


Gambar 3.3. Flowchart Haversine

1.9 Perancangan Sistem Antarmuka

1.9.1 Perancangan Jarak Terpendek

Berikut ini adalah tampilan hasil perancangan antarmuka dari rute terpendek dengan metode Haversine.



The interface consists of a main container with two large empty rectangular areas. The left area is labeled 'GRAF' and the right area is labeled 'PERHITUNGAN'. Below these areas is a control panel with several buttons:

- A single button labeled 'START'.
- Two buttons labeled 'FINISH' and 'BOBOT'.
- A button labeled 'KOORDINAT'.
- A button labeled 'WEIGHT'.
- A wide button labeled 'HAVERSINE'.

Gambar 3.4. Interface Perancangan Jarak Terpendek

Halaman perancangan ini adalah halaman untuk user mencari rute terpendek dengan algoritma Haversine. User tinggal menginput lokasi awal dan lokasi tujuan dan tinggal mengklik cari rute untuk menentukan rute terpendek apa yang akan digunakan.

BAB IV

HASIL DAN PEMBAHASAN

1.1 Implementasi

Pada bagian ini akan dilakukan proses pengujian aplikasi yang sudah dibuat berdasarkan perancangan sebelumnya. Aplikasi ini dibuat menggunakan Visual Basic.NET 2010.

1.1.1 Tampilan Halaman Awal

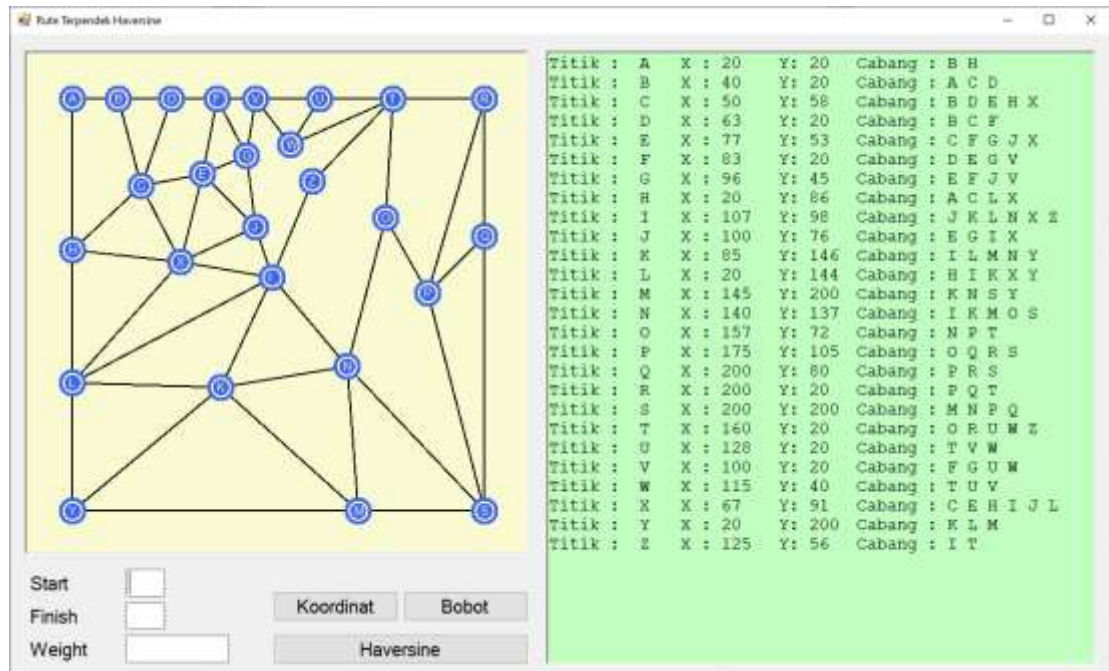
Tampilan menu awal adalah tampilan yang pertama kali ditampilkan. Halaman ini memiliki beberapa fungsi untuk menghubungkan ke tampilan lainnya. Berikut adalah tampilan halaman awal dari program aplikasi.



Gambar 4.1. Tampilan Halaman Awal

1.1.2 Tampilan Menu Buat Kordinat

Halaman Menu Kordinat digunakan untuk menginput semua data tempat pelanggan. Berikut adalah tampilan halaman menu buat koordinat.

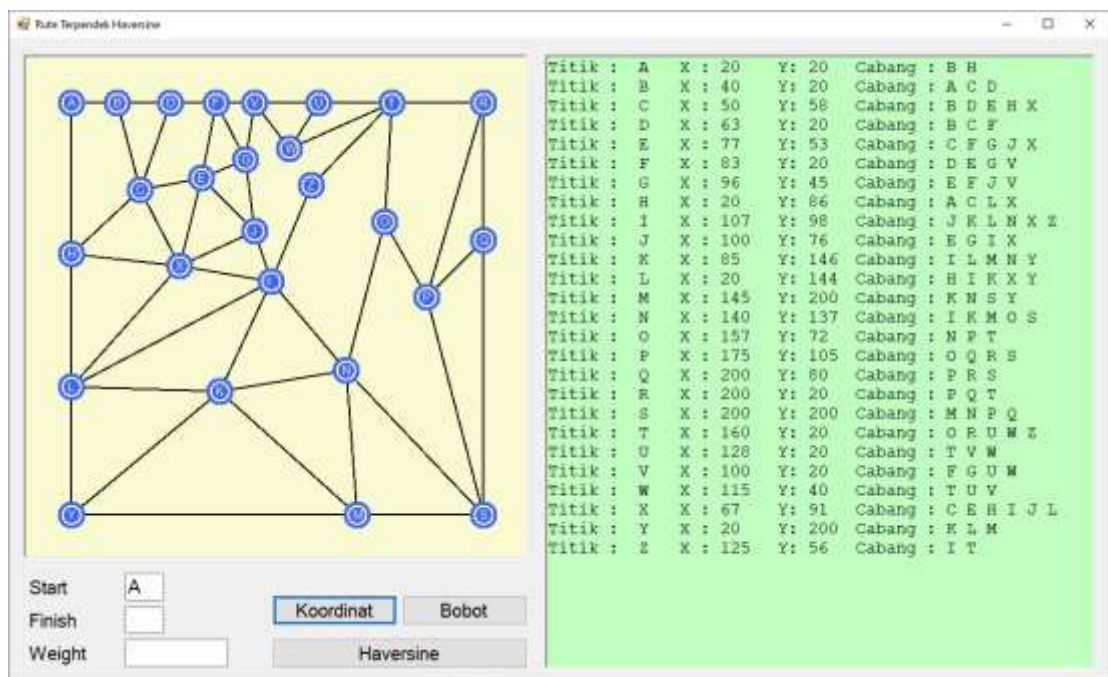


Gambar 4.2. Tampilan *Input* Buat Kordinat

Pada proses buat koordinat, ada 26 titik yang akan diciptakan dari mulai A hingga Z, setiap titik memiliki koordinat dan percabangan masing-masing. Pada kasus ini semua titik dianggap memiliki jalur dua arah. Sebagai contoh pada titik AB, ada rute dari A ke B dan sebaliknya ada rute dari B ke A.

1.1.3 Tampilan Menu Pilih Titik Awal

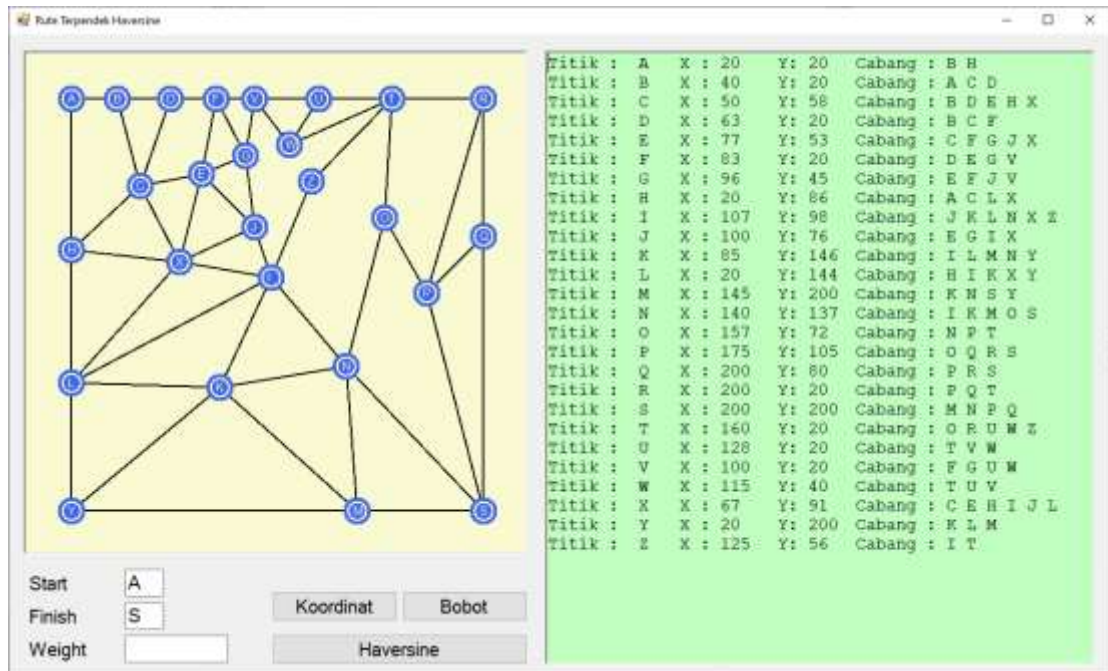
Tampilan ini digunakan untuk menginputkan data titik awal dalam proses penentuan rute terpendek. Berikut adalah tampilan proses data titik awal :



Gambar 4.3. Tampilan isi titik awal pada bagian Start

1.1.4 Tampilan Menu Pilih Titik Akhir

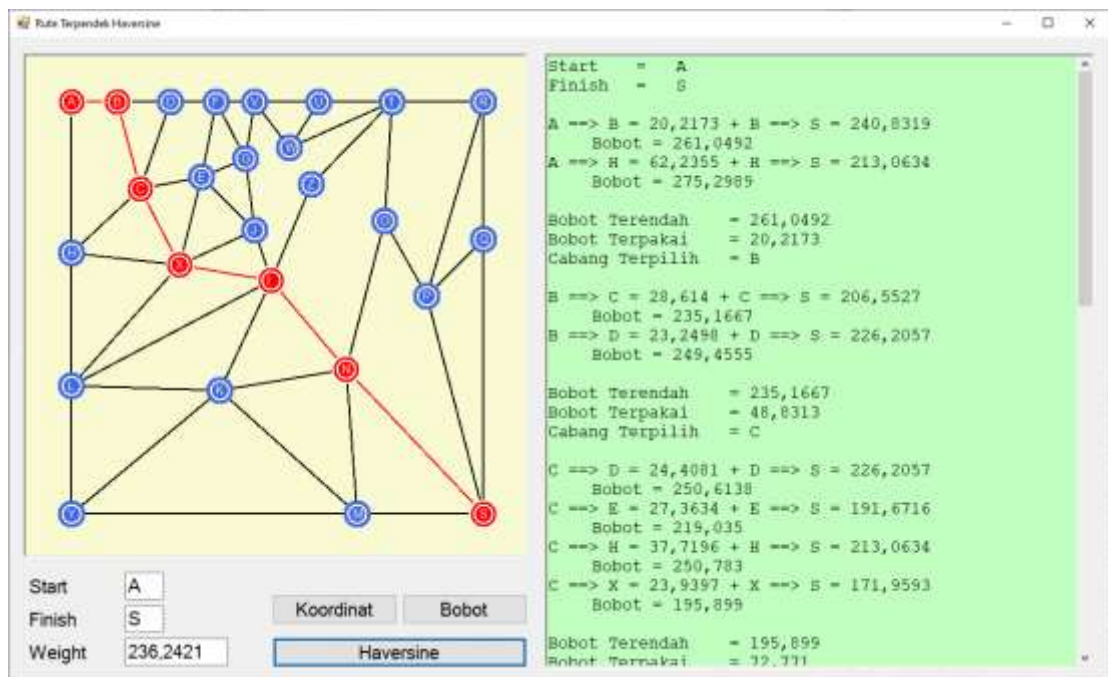
Tampilan ini digunakan untuk menginputkan data titik akhir (Tujuan) dalam proses penentuan rute terpendek. Berikut adalah tampilan proses data titik akhir:



Gambar 4.4. Tampilan Edit Data

1.1.5 Tampilan Hasil Rute Terpendek

Tampilan halaman rute terpendek adalah tampilan hasil perhitungan dari titik awal ke titik akhir, yang dimana titik awal adalah A dan titik akhir adalah S sebagai berikut.



Gambar 4.5. Tampilan hasil rute terpendek

Adapun hasil dari proses perhitungan rute terpendek menggunakan metode Haversine dengan titik awal A dan tujuan akhir S, rute yang dilalui adalah A-B-C-X-I-N-S.

1.2 Pengujian Sistem

Pengujian ini akan menghitung rute terdekat dari beberapa titik yang akan ditentukan. Adapun tujuan dari pembuatan tabel data vektor ini adalah untuk menentukan bobot jarak dari dua titik.

Titik [A] ke Titik [B] = 20

Titik [A] ke Titik [H] = 66

Titik [B] ke Titik [A] = 20

Titik [B] ke Titik [C] = 39,2938

Titik [B] ke Titik [D] = 23

Titik [C] ke Titik [B] = 39,2938

Titik [C] ke Titik [D] = 40,1622

Titik [C] ke Titik [E] = 27,4591

Titik [C] ke Titik [H] = 41,0366

Titik [C] ke Titik [X] = 37,1214

Titik [D] ke Titik [B] = 23

Titik [D] ke Titik [C] = 40,1622

Titik [D] ke Titik [F] = 20

Titik [E] ke Titik [C] = 27,4591

Titik [E] ke Titik [F] = 33,541

Titik [E] ke Titik [G] = 20,6155

Titik [E] ke Titik [J] = 32,5269

Titik [E] ke Titik [X] = 39,2938

Titik [F] ke Titik [D] = 20

Titik [F] ke Titik [E] = 33,541

Titik [F] ke Titik [G] = 28,178

Titik [F] ke Titik [V] = 17

Titik [G] ke Titik [E] = 20,6155

Titik [G] ke Titik [F] = 28,178

Titik [G] ke Titik [J] = 31,257

Titik [G] ke Titik [V] = 25,318

Titik [H] ke Titik [A] = 66

Titik [H] ke Titik [C] = 41,0366

Titik [H] ke Titik [L] = 58

Titik [H] ke Titik [X] = 47,2652

Titik [I] ke Titik [J] = 23,0868
 Titik [I] ke Titik [K] = 52,8015
 Titik [I] ke Titik [L] = 98,4124
 Titik [I] ke Titik [N] = 51,0882
 Titik [I] ke Titik [X] = 40,6079
 Titik [I] ke Titik [Z] = 45,6946

Titik [J] ke Titik [E] = 32,5269
 Titik [J] ke Titik [G] = 31,257
 Titik [J] ke Titik [I] = 23,0868
 Titik [J] ke Titik [X] = 36,2491

Titik [K] ke Titik [I] = 52,8015
 Titik [K] ke Titik [L] = 65,0308
 Titik [K] ke Titik [M] = 80,7217
 Titik [K] ke Titik [N] = 55,7315
 Titik [K] ke Titik [Y] = 84,5044

Titik [L] ke Titik [H] = 58
 Titik [L] ke Titik [I] = 98,4124
 Titik [L] ke Titik [K] = 65,0308
 Titik [L] ke Titik [X] = 70,8378
 Titik [L] ke Titik [Y] = 56

Titik [M] ke Titik [K] = 80,7217
 Titik [M] ke Titik [N] = 63,1981
 Titik [M] ke Titik [S] = 55
 Titik [M] ke Titik [Y] = 125

Titik [N] ke Titik [I] = 51,0882
 Titik [N] ke Titik [K] = 55,7315
 Titik [N] ke Titik [M] = 63,1981
 Titik [N] ke Titik [O] = 67,1863
 Titik [N] ke Titik [S] = 87

Titik [O] ke Titik [N] = 67,1863
 Titik [O] ke Titik [P] = 37,5899
 Titik [O] ke Titik [T] = 52,0865

Titik [P] ke Titik [O] = 37,5899
 Titik [P] ke Titik [Q] = 35,3553
 Titik [P] ke Titik [R] = 88,6002
 Titik [P] ke Titik [S] = 98,2344

Titik [Q] ke Titik [P] = 35,3553
 Titik [Q] ke Titik [R] = 60

Titik [Q] ke Titik [S] = 120

Titik [R] ke Titik [P] = 88,6002
Titik [R] ke Titik [Q] = 60
Titik [R] ke Titik [T] = 40

Titik [S] ke Titik [M] = 55
Titik [S] ke Titik [N] = 87
Titik [S] ke Titik [P] = 98,2344
Titik [S] ke Titik [Q] = 120

Titik [T] ke Titik [O] = 52,0865
Titik [T] ke Titik [R] = 40
Titik [T] ke Titik [U] = 32
Titik [T] ke Titik [W] = 49,2443
Titik [T] ke Titik [Z] = 50,2096

Titik [U] ke Titik [T] = 32
Titik [U] ke Titik [V] = 28
Titik [U] ke Titik [W] = 23,8537

Titik [V] ke Titik [F] = 17
Titik [V] ke Titik [G] = 25,318
Titik [V] ke Titik [U] = 28
Titik [V] ke Titik [W] = 25

Titik [W] ke Titik [T] = 49,2443
Titik [W] ke Titik [U] = 23,8537
Titik [W] ke Titik [V] = 25

Titik [X] ke Titik [C] = 37,1214
Titik [X] ke Titik [E] = 39,2938
Titik [X] ke Titik [H] = 47,2652
Titik [X] ke Titik [I] = 40,6079
Titik [X] ke Titik [J] = 36,2491
Titik [X] ke Titik [L] = 70,8378

Titik [Y] ke Titik [K] = 84,5044
Titik [Y] ke Titik [L] = 56
Titik [Y] ke Titik [M] = 125

Titik [Z] ke Titik [I] = 45,6946
Titik [Z] ke Titik [T] = 50,2096

Hasil perhitungan dalam menentukan rute terpendak dari titik A ke S dapat dilihat pada perhitungan berikut ini.

Start = A
Finish = S

A ==> B = 20,2173 + B ==> S = 240,8319
Bobot = 261,0492
A ==> H = 62,2355 + H ==> S = 213,0634
Bobot = 275,2989

Bobot Terendah = 261,0492
Bobot Terpakai = 20,2173
Cabang Terpilih = B

B ==> C = 28,614 + C ==> S = 206,5527
Bobot = 235,1667
B ==> D = 23,2498 + D ==> S = 226,2057
Bobot = 249,4555

Bobot Terendah = 235,1667
Bobot Terpakai = 48,8313
Cabang Terpilih = C

C ==> D = 24,4081 + D ==> S = 226,2057
Bobot = 250,6138
C ==> E = 27,3634 + E ==> S = 191,6716
Bobot = 219,035
C ==> H = 37,7196 + H ==> S = 213,0634
Bobot = 250,783
C ==> X = 23,9397 + X ==> S = 171,9593
Bobot = 195,899

Bobot Terendah = 195,899
Bobot Terpakai = 72,771
Cabang Terpilih = X

X ==> E = 15,1071 + E ==> S = 191,6716
Bobot = 206,7787
X ==> H = 47,6211 + H ==> S = 213,0634
Bobot = 260,6845
X ==> I = 40,3577 + I ==> S = 138,0326
Bobot = 178,3903
X ==> J = 33,1118 + J ==> S = 159,2985

Bobot = 192,4103
 X ==> L = 58,2277 + L ==> S = 188,5099
 Bobot = 246,7376

Bobot Terendah = 178,3903
 Bobot Terpakai = 113,1287
 Cabang Terpilih = I

I ==> J = 8,6574 + J ==> S = 159,2985
 Bobot = 167,9559
 I ==> K = 20,8968 + K ==> S = 127,0472
 Bobot = 147,944
 I ==> L = 83,0631 + L ==> S = 188,5099
 Bobot = 271,573
 I ==> N = 38,3424 + N ==> S = 87
 Bobot = 125,3424
 I ==> Z = 25,0411 + Z ==> S = 162,3607
 Bobot = 187,4018

Bobot Terendah = 125,3424
 Bobot Terpakai = 151,4711
 Cabang Terpilih = N

N ==> K = 55,5393 + K ==> S = 127,0472
 Bobot = 182,5865
 N ==> M = 49,723 + M ==> S = 55
 Bobot = 104,723
 N ==> O = 57,306 + O ==> S = 135,0296
 Bobot = 192,3356
 N ==> S = 84,771 + S ==> S = 0
 Bobot = 84,771

Bobot Terendah = 84,771
 Bobot Terpakai = 236,2421
 Cabang Terpilih = S

Rute yang dilalui = A-B-C-X-I-N-S

Hasil perhitungan dalam menentukan rute terpendak dari titik R ke Y dapat dilihat pada perhitungan berikut ini.

Start = R
Finish = Y

R ==> P = 87,9774 + P ==> Y = 181,7966
Bobot = 269,774
R ==> Q = 56,6575 + Q ==> Y = 216,3331
Bobot = 272,9906
R ==> T = 40,4345 + T ==> Y = 228,0351
Bobot = 268,4696

Bobot Terendah = 268,4696
Bobot Terpakai = 40,4345
Cabang Terpilih = T

T ==> O = 48,751 + O ==> Y = 187,4913
Bobot = 236,2423
T ==> U = 32,3476 + U ==> Y = 209,9143
Bobot = 242,2619
T ==> W = 47,4187 + W ==> Y = 186,0779
Bobot = 233,4966
T ==> Z = 44,7379 + Z ==> Y = 178,2162
Bobot = 222,9541

Bobot Terendah = 222,9541
Bobot Terpakai = 85,1724
Cabang Terpilih = Z

Z ==> I = 25,0411 + I ==> Y = 134,0634
Bobot = 159,1045

Bobot Terendah = 159,1045
Bobot Terpakai = 110,2135
Cabang Terpilih = I

I ==> J = 8,6574 + J ==> Y = 147,5669
Bobot = 156,2243
I ==> K = 20,8968 + K ==> Y = 84,5044
Bobot = 105,4012
I ==> L = 83,0631 + L ==> Y = 56
Bobot = 139,0631
I ==> N = 38,3424 + N ==> Y = 135,5323

Bobot = 173,8747
 I ==> X = 40,3577 + X ==> Y = 118,7013
 Bobot = 159,059

Bobot Terendah = 105,4012
 Bobot Terpakai = 131,1103
 Cabang Terpilih = K

K ==> L = 65,7093 + L ==> Y = 56
 Bobot = 121,7093

K ==> M = 58,6635 + M ==> Y = 125
 Bobot = 183,6635

K ==> N = 55,5393 + N ==> Y = 135,5323
 Bobot = 191,0716

K ==> Y = 67,8456 + Y ==> Y = 0
 Bobot = 67,8456

Bobot Terendah = 67,8456
 Bobot Terpakai = 198,9559
 Cabang Terpilih = Y

Rute yang dilalui = R-T-Z-I-K-Y

Hasil perhitungan dalam menentukan rute terpendak dari titik V ke M dapat dilihat pada perhitungan berikut ini.

Start = V
 Finish = M

V ==> F = 17,1847 + F ==> M = 190,3786
 Bobot = 207,5633

V ==> G = 5,2699 + G ==> M = 162,5608
 Bobot = 167,8307

V ==> U = 28,3042 + U ==> M = 180,801
 Bobot = 209,1052

V ==> W = 16,1236 + W ==> M = 162,7882
 Bobot = 178,9118

Bobot Terendah = 167,8307
 Bobot Terpakai = 5,2699
 Cabang Terpilih = G

$G \implies E = 19,1656 + E \implies M = 161,966$
 Bobot = 181,1316
 $G \implies F = 12,8303 + F \implies M = 190,3786$
 Bobot = 203,2089
 $G \implies J = 5,8108 + J \implies M = 131,9129$
 Bobot = 137,7237

Bobot Terendah = 137,7237
 Bobot Terpakai = 11,0807
 Cabang Terpilih = J

$J \implies E = 22,7852 + E \implies M = 161,966$
 Bobot = 184,7512
 $J \implies I = 8,6574 + I \implies M = 108,8485$
 Bobot = 117,5059
 $J \implies X = 33,1118 + X \implies M = 134,0336$
 Bobot = 167,1454

Bobot Terendah = 117,5059
 Bobot Terpakai = 19,7381
 Cabang Terpilih = I

$I \implies K = 20,8968 + K \implies M = 80,7217$
 Bobot = 101,6185
 $I \implies L = 83,0631 + L \implies M = 136,9708$
 Bobot = 220,0339
 $I \implies N = 38,3424 + N \implies M = 63,1981$
 Bobot = 101,5405
 $I \implies X = 40,3577 + X \implies M = 134,0336$
 Bobot = 174,3913
 $I \implies Z = 25,0411 + Z \implies M = 145,3823$
 Bobot = 170,4234

Bobot Terendah = 101,5405
 Bobot Terpakai = 58,0805
 Cabang Terpilih = N

$N \implies K = 55,5393 + K \implies M = 80,7217$
 Bobot = 136,261
 $N \implies M = 49,723 + M \implies M = 0$
 Bobot = 49,723
 $N \implies O = 57,306 + O \implies M = 128,5613$
 Bobot = 185,8673
 $N \implies S = 84,771 + S \implies M = 55$
 Bobot = 139,771

Bobot Terendah = 49,723
 Bobot Terpakai = 107,8035
 Cabang Terpilih = M

Rute yang dilalui = V-G-J-I-N-M

Hasil perhitungan dalam menentukan rute terpendak dari titik H ke Q dapat dilihat pada perhitungan berikut ini.

Start = H
 Finish = Q

H ==> A = 62,2355 + A ==> Q = 189,7367
 Bobot = 251,9722
 H ==> C = 37,7196 + C ==> Q = 151,6047
 Bobot = 189,3243
 H ==> L = 54,7923 + L ==> Q = 191,0393
 Bobot = 245,8316
 H ==> X = 47,6211 + X ==> Q = 133,4541
 Bobot = 181,0752

Bobot Terendah = 181,0752
 Bobot Terpakai = 47,6211
 Cabang Terpilih = X

X ==> C = 23,9397 + C ==> Q = 151,6047
 Bobot = 175,5444
 X ==> E = 15,1071 + E ==> Q = 125,9286
 Bobot = 141,0357
 X ==> I = 40,3577 + I ==> Q = 94,7259
 Bobot = 135,0836
 X ==> J = 33,1118 + J ==> Q = 100,08
 Bobot = 133,1918
 X ==> L = 58,2277 + L ==> Q = 191,0393
 Bobot = 249,267

Bobot Terendah = 133,1918
 Bobot Terpakai = 80,7329
 Cabang Terpilih = J

J ==> E = 22,7852 + E ==> Q = 125,9286

Bobot = 148,7138
 J ==> G = 5,8108 + G ==> Q = 109,7315
 Bobot = 115,5423
 J ==> I = 8,6574 + I ==> Q = 94,7259
 Bobot = 103,3833

Bobot Terendah = 103,3833
 Bobot Terpakai = 89,3903
 Cabang Terpilih = I

I ==> K = 20,8968 + K ==> Q = 132,5934
 Bobot = 153,4902
 I ==> L = 83,0631 + L ==> Q = 191,0393
 Bobot = 274,1024
 I ==> N = 38,3424 + N ==> Q = 82,7587
 Bobot = 121,1011
 I ==> Z = 25,0411 + Z ==> Q = 78,7464
 Bobot = 103,7875

Bobot Terendah = 103,7875
 Bobot Terpakai = 114,4314
 Cabang Terpilih = Z

Z ==> T = 44,7379 + T ==> Q = 72,111
 Bobot = 116,8489

Bobot Terendah = 116,8489
 Bobot Terpakai = 159,1693
 Cabang Terpilih = T

T ==> O = 48,751 + O ==> Q = 43,7379
 Bobot = 92,4889
 T ==> R = 40,4345 + R ==> Q = 60
 Bobot = 100,4345
 T ==> U = 32,3476 + U ==> Q = 93,723
 Bobot = 126,0706
 T ==> W = 47,4187 + W ==> Q = 93,9415
 Bobot = 141,3602

Bobot Terendah = 92,4889
 Bobot Terpakai = 207,9203
 Cabang Terpilih = O

O ==> N = 57,306 + N ==> Q = 82,7587
 Bobot = 140,0647
 O ==> P = 36,9685 + P ==> Q = 35,3553
 Bobot = 72,3238

Bobot Terendah = 72,3238
 Bobot Terpakai = 244,8888
 Cabang Terpilih = P

P ==> Q = 35,4358 + Q ==> Q = 0
 Bobot = 35,4358

P ==> R = 87,9774 + R ==> Q = 60
 Bobot = 147,9774

P ==> S = 97,443 + S ==> Q = 120
 Bobot = 217,443

Bobot Terendah = 35,4358
 Bobot Terpakai = 280,3246
 Cabang Terpilih = Q

Rute yang dilalui = H-X-J-I-Z-T-O-P-Q

Hasil perhitungan dalam menentukan rute terpendak dari titik S ke B dapat dilihat pada perhitungan berikut ini.

Start = S
 Finish = B

S ==> M = 55,5975 + M ==> B = 208,3867
 Bobot = 263,9842

S ==> N = 84,771 + N ==> B = 153,9123
 Bobot = 238,6833

S ==> P = 97,443 + P ==> B = 159,5306
 Bobot = 256,9736

S ==> Q = 110,1207 + Q ==> B = 170,8801
 Bobot = 281,0008

Bobot Terendah = 238,6833
 Bobot Terpakai = 84,771
 Cabang Terpilih = N

N ==> I = 38,3424 + I ==> B = 102,8251
 Bobot = 141,1675

N ==> K = 55,5393 + K ==> B = 133,7946
 Bobot = 189,3339

$$N \implies M = 49,723 + M \implies B = 208,3867$$

$$\text{Bobot} = 258,1097$$

$$N \implies O = 57,306 + O \implies B = 128,0352$$

$$\text{Bobot} = 185,3412$$

$$\text{Bobot Terendah} = 141,1675$$

$$\text{Bobot Terpakai} = 123,1134$$

$$\text{Cabang Terpilih} = I$$

$$I \implies J = 8,6574 + J \implies B = 82,0731$$

$$\text{Bobot} = 90,7305$$

$$I \implies K = 20,8968 + K \implies B = 133,7946$$

$$\text{Bobot} = 154,6914$$

$$I \implies L = 83,0631 + L \implies B = 125,6025$$

$$\text{Bobot} = 208,6656$$

$$I \implies X = 40,3577 + X \implies B = 75,9605$$

$$\text{Bobot} = 116,3182$$

$$I \implies Z = 25,0411 + Z \implies B = 92,3093$$

$$\text{Bobot} = 117,3504$$

$$\text{Bobot Terendah} = 90,7305$$

$$\text{Bobot Terpakai} = 131,7708$$

$$\text{Cabang Terpilih} = J$$

$$J \implies E = 22,7852 + E \implies B = 49,5782$$

$$\text{Bobot} = 72,3634$$

$$J \implies G = 5,8108 + G \implies B = 61,327$$

$$\text{Bobot} = 67,1378$$

$$J \implies X = 33,1118 + X \implies B = 75,9605$$

$$\text{Bobot} = 109,0723$$

$$\text{Bobot Terendah} = 67,1378$$

$$\text{Bobot Terpakai} = 137,5816$$

$$\text{Cabang Terpilih} = G$$

$$G \implies E = 19,1656 + E \implies B = 49,5782$$

$$\text{Bobot} = 68,7438$$

$$G \implies F = 12,8303 + F \implies B = 43$$

$$\text{Bobot} = 55,8303$$

$$G \implies V = 5,2699 + V \implies B = 60$$

$$\text{Bobot} = 65,2699$$

$$\text{Bobot Terendah} = 55,8303$$

$$\text{Bobot Terpakai} = 150,4119$$

$$\text{Cabang Terpilih} = F$$

$$F \implies D = 20,2173 + D \implies B = 23$$

Bobot = 43,2173
 F ==> E = 8,1569 + E ==> B = 49,5782
 Bobot = 57,7351
 F ==> V = 17,1847 + V ==> B = 60
 Bobot = 77,1847

Bobot Terendah = 43,2173
 Bobot Terpakai = 170,6292
 Cabang Terpilih = D

D ==> B = 23,2498 + B ==> B = 0
 Bobot = 23,2498
 D ==> C = 24,4081 + C ==> B = 39,2938
 Bobot = 63,7019

Bobot Terendah = 23,2498
 Bobot Terpakai = 193,879
 Cabang Terpilih = B

Rute yang dilalui = S-N-I-J-G-F-D-B

Hasil perhitungan dalam menentukan rute terpendak dari titik S ke B dapat dilihat pada perhitungan berikut ini.

Start = C
 Finish = P

C ==> B = 28,614 + B ==> P = 159,5306
 Bobot = 188,1446
 C ==> D = 24,4081 + D ==> P = 140,6023
 Bobot = 165,0104
 C ==> E = 27,3634 + E ==> P = 110,9414
 Bobot = 138,3048
 C ==> H = 37,7196 + H ==> P = 156,1602
 Bobot = 193,8798
 C ==> X = 23,9397 + X ==> P = 108,9036
 Bobot = 132,8433

Bobot Terendah = 132,8433
 Bobot Terpakai = 23,9397
 Cabang Terpilih = X

$X \Rightarrow E = 15,1071 + E \Rightarrow P = 110,9414$
 Bobot = 126,0485
 $X \Rightarrow H = 47,6211 + H \Rightarrow P = 156,1602$
 Bobot = 203,7813
 $X \Rightarrow I = 40,3577 + I \Rightarrow P = 68,3593$
 Bobot = 108,717
 $X \Rightarrow J = 33,1118 + J \Rightarrow P = 80,4114$
 Bobot = 113,5232
 $X \Rightarrow L = 58,2277 + L \Rightarrow P = 159,8312$
 Bobot = 218,0589

Bobot Terendah = 108,717
 Bobot Terpakai = 64,2974
 Cabang Terpilih = I

$I \Rightarrow J = 8,6574 + J \Rightarrow P = 80,4114$
 Bobot = 89,0688
 $I \Rightarrow K = 20,8968 + K \Rightarrow P = 98,8989$
 Bobot = 119,7957
 $I \Rightarrow L = 83,0631 + L \Rightarrow P = 159,8312$
 Bobot = 242,8943
 $I \Rightarrow N = 38,3424 + N \Rightarrow P = 47,4236$
 Bobot = 85,766
 $I \Rightarrow Z = 25,0411 + Z \Rightarrow P = 70,0071$
 Bobot = 95,0482

Bobot Terendah = 85,766
 Bobot Terpakai = 102,6398
 Cabang Terpilih = N

$N \Rightarrow K = 55,5393 + K \Rightarrow P = 98,8989$
 Bobot = 154,4382
 $N \Rightarrow M = 49,723 + M \Rightarrow P = 99,6243$
 Bobot = 149,3473
 $N \Rightarrow O = 57,306 + O \Rightarrow P = 37,5899$
 Bobot = 94,8959
 $N \Rightarrow S = 84,771 + S \Rightarrow P = 98,2344$
 Bobot = 183,0054

Bobot Terendah = 94,8959
 Bobot Terpakai = 159,9458
 Cabang Terpilih = O

$O \Rightarrow P = 36,9685 + P \Rightarrow P = 0$
 Bobot = 36,9685
 $O \Rightarrow T = 48,751 + T \Rightarrow P = 86,3134$

Bobot = 135,0644

Bobot Terendah = 36,9685
Bobot Terpakai = 196,9143
Cabang Terpilih = P

Rute yang dilalui = C-X-I-N-O-P

1.3 Kelebihan dan Kekurangan Sistem

Adapun kelebihan dan kekurangan dari Pencarian jalur terdekat di Kota Binjai Menggunakan Metode Haversine ini adalah sebagai berikut:

1. Kelebihan Sistem
 - a. Lebih mudah diakses.
 - b. Proses penginputan lokasi lebih cepat.
 - c. Proses pencarian lokasi lebih mudah
2. Kekurangan Sistem
 - a. Hanya mencari lokasi.
 - b. Sebaiknya dapat digunakan pada Android.
 - c. Hanya membahas lokasi secara koordinat, untuk lokasi-lokasi real seperti Google Map belum ada.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil aplikasi rancangan pencarian rute terpendek menggunakan algoritma Haversine, maka didapat beberapa kesimpulan seperti berikut :

1. Dengan adanya aplikasi pencarian rute terpendek pemahaman tentang algoritma Haversine ini menjadi lebih mudah.
2. Algoritma Haversine yang digunakan dalam proses pencarian rute terpendek ini sangat membantu dan mudah untuk dimengerti oleh penulis.

5.2 Saran

Berikut adalah saran dari penulis agar aplikasi rancangan pencarian rute terpendek menggunakan algoritma dapat bermanfaat dan dikembangkan menjadi lebih baik lagi :

1. Di masa yang akan datang, sebaiknya aplikasi ini dikembangkan dengan menerapkan sebuah metode lainnya serta diterapkan untuk rute distribusi terdekat.
2. Aplikasi ini akan lebih baik jika dirancang dengan aplikasi android.

DAFTAR PUSTAKA

- Nugraha, Dedi, Sugito Sugito, and Dwi Ispriyanti. "Penentuan Model Sistem Antrean Kendaraan Di Gerbang Tol Banyumanik Semarang." *Jurnal Gaussian 2.2* (2013): 89-97.
- Harni Kusniyati, Jurnal, Redaksi Tim. "Aplikasi Pencarian Ustadz untuk Wilayah DKI Jakarta Menggunakan Algoritma Haversine Formula Berbasis Android." *Petir 9.2* (2016): 102-111.
- Oktavian, Diar Puji. *Menjadi Programmer jempolan menggunakan PHP*. Penerbit Mediakom, 2010.
- Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 8-18.
- Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." *Seminar Nasional Teknologi Informasi Dan Multimedia*, ISSN. 2015.
- Putera, A., Siahaan, U., & Rahim, R. (2016). Dynamic key matrix of hill cipher using genetic algorithm. *Int. J. Secur. Its Appl*, 10(8), 173-180.
- Putra, Randi Rian, and Cendra Wadisman. "Implementasi Data Mining Pemilihan Pelanggan Potensial Menggunakan Algoritma K Means." *INTECOMS: Journal of Information Technology and Computer Science 1.1* (2018): 72-77.
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Rahim, R. (2018, October). A Novelty Once Methode Power System Policies Based On SCS (Solar Cell System). In *International Conference of ASEAN Prespective and Policy (ICAP)* (Vol. 1, No. 1, pp. 195-198).
- Rizal, Chairul. "Pengaruh Varietas dan Pupuk Petroganik Terhadap Pertumbuhan, Produksi dan Viabilitas Benih Jagung (*Zea mays L.*)" *ETD Unsyiah* (2013).
- Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. *CESS (Journal of Computer Engineering, System and Science)*, 3(1), 45-49. kbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.

- Sari, R. D., Supiyandi, A. P. U., Siahaan, M. M., & Ginting, R. B. (2017). A Review of IP and MAC Address Filtering in Wireless Network Security. *Int. J. Sci. Res. Sci. Technol*, 3(6), 470-473.
- Siahaan, MD Lesmana, Melva Sari Panjaitan, and Andysah Putera Utama Siahaan. "MikroTik bandwidth management to gain the users prosperity prevalent." *Int. J. Eng. Trends Technol* 42.5 (2016): 218-222.
- Sidik, A. P. (2018). Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol.
- Sumartono, I., Siahaan, A. P. U., & Mayasari, N. (2016). An overview of the RC4 algorithm. *IOSR J. Comput. Eng*, 18(6), 67-73.
- Supiyandi, S., Hermansyah, H., & Sembiring, K. A. (2020). Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video. *Jurnal Media Informatika Budidarma*, 4(2), 340-346.
- Sitorus, Z. (2018). Kebutuhan Web Service untuk Sinkronisasi Data Antar Sistem Informasi dalam Universitas. *Jurnal Teknik dan Informatika*, 5(2), 87-90.
- Sarif, M. I. (2017). Penemuan Aturan yang Berkaitan dengan Pola dalam Deret Berkala (Time Series).
- Sulistiyorini, Prastuti, Christian Yulianto Rusli, and Indrayanti Indrayanti. "Aplikasi Pengukuran Kualitas Pelayanan Publik Secara Online Untuk Mendukung Smart Government Kota Pekalongan (Online Public Service Quality Measurement Application To Support Smart Government Pekalongan City)." *Jurnal Litbang Kota Pekalongan* 13 (2017).
- Utomo, Andy Prasetyo. "Analisa dan perancangan sistem informasi parkir di Universitas Muria Kudus." *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer* 3.1 (2013): 17-24.
- Sutedi, Ade, Cepy Slamet, and Dhami Johar Damiri. "Rancang Bangun Open Access Journal Menggunakan Metode Object Oriented Dengan Pendekatan Uml-Based Web Engineering." *Jurnal Algoritma* 12.1 (2015).
- Rosa A.S dan M. Shalahuddin, Alfarisyi, Muhammad Imam, Rispianda Rispianda, and Khuria Amila. "Rancangan sistem informasi layanan alumni ITENAS berbasis web." *Reka Integra* 2.1 (2014)..
- Triandini, Evi, and I. Gede Suardika. *Step by Step Desain Proyek Menggunakan UML*. Penerbit Andi, 2012..
- Irawan, Yunus, and Abdillah Baraja. "Analisis dan perancangan jaringan komputer Sekolah dasar islam sains dan teknologi ibnu qoyyim surakarta." *IJNS- Indonesian Journal on Networking and Security* 4.3 (2012)..