

IMPLEMENTASI ALGORITMA *STEEPEST ASCENT HILL CLIMBING* (SAHC) UNTUK APLIKASI PENCARIAN RUTE TERPENDEK BERBASIS *MOBILE*

SKRIPSI

RAZZAQ H.NUR WIJAYA

71153002



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA**

MEDAN

2019

IMPLEMENTASI ALGORITMA *STEEPEST ASCENT HILL CLIMBING* (SAHC) UNTUK APLIKASI PENCARIAN RUTE TERPENDEK BERBASIS *MOBILE*

SKRIPSI

Diajukan untuk memenuhi syarat mencapai gelar komputer

RAZZAQ H.NUR WIJAYA

71153002



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA**

MEDAN

2019

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi
Lamp : -

Kepada Yth :
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara:

Nama : Razzaq H.Nur Wijaya

Nomor Induk Mahasiswa : 71153002

Program Studi : Ilmu Komputer

Judul : Implementasi Algoritma *Steepest Ascent*
Hill Climbing (SAHC) Untuk Aplikasi
Pencarian Rute Terpendek berbasis *Mobile*

dapat disetujui untuk segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terimakasih

Medan, 11 November 2019
14 Rabiul Awal 1441 H

Komisi Pembimbing

Dosen Pembimbing I

Dosen Pembimbing

Dr. Mhd. Furqan, S.Si., M.Comp.Sc
NIP. 198008062006041003

Armansyah, M.Kom
NIB. 2004108401

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertandatangan di bawah ini:

Nama : Razzaq H.Nur Wijaya

Nomor Induk Mahasiswa : 71153002

Program Studi : Ilmu Komputer

Judul : Implementasi Algoritma *Steepest Ascent Hill Climbing*
(SAHC) Untuk Aplikasi Pencarian Rute Terpendek
berbasis *Mobile*

menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 11 November 2019

Razzaq H.Nur Wijaya
NIM. 71153002



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url : <https://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESEHAN SKRIPSI

Nomor: 008/ST.V/PP.01.1/01/2020

Judul : Implementasi Algoritma *Steepest Ascent Hill Climbing* (SAHC)
Untuk Aplikasi Pencarian Rute Terpendek berbasis *Mobile*
Nama : Razzaq H.Nur Wijaya
Nomor Induk Mahasiswa : 71153002
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan dinyatakan **LULUS**.

Pada hari /tanggal : Senin, 11 November 2019
Tempat : Ruang Sidang FST

Tim Ujian Munaqasyah,
Ketua,

Dr. Mhd. Furqan, S.Si., M..Comp.Sc
NIP. 19800806 200604 1 003

Dewan Penguji,

Penguji I,

Penguji ,

Dr. Mhd. Furqan, S.Si., M..Comp.Sc
NIP. 19800806 200604 1 003

Armansyah, M.Kom
NIB. 2004108401

Penguji I,

Penguji IV,

Yusuf Ramadhan Nasution, M.Kom
NIB. 2025058501

Abdul Halim Hasugian M.Kom
NIB. 0127038801

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan

Dr. H. M. Jamil, MA
NIP. 196609101999031002

ABSTRAK

Pencarian dan penentuan rute terpendek merupakan masalah yang rumit, mencari rute terpendek dari sejumlah objek wisata dan jarak antar objek wisata. Dengan akses jalur yang bervariasi, pencarian rute terpendek menjadi pilihan yang tepat menggunakan sebuah aplikasi *mobile*. Berdasarkan hasil Implementasi Algoritma *Steepest Ascent Hill Climbing* (SAHC) Untuk Aplikasi Pencarian Rute Terpendek berbasis *Mobile* di Kabupaten Humbang Hasundutan. Maka didapat kesimpulan bahwa pencarian rute terpendek berbasis *Mobile* dapat diselesaikan dengan menggunakan algoritma *Steepest Ascent Hill Climbing*. Pada proses perhitungan manual dengan menggunakan algoritma *Steepest Ascent Hill Climbing* pada node asal Humbang terdapat nilai heuristik sebesar 0.0896184808, pada node asal simpang tiga terdapat nilai heuristik 0.1693780561, pada node asal sipinsur terdapat nilai heuristik sebesar 0.367474152, pada node asal air terjun janji terdapat nilai heuristik sebesar 0.3043253189, dan yang terakhir pada node asal air terjun sibabo terdapat nilai heuristik sebesar 0.3823982675. Maka hasil rute terpendek dari Sipinsur Geosite (F) ke Air terjun simolap (B) adalah $F \rightarrow D \rightarrow B$ (Sipinsur GeoSite – simpang 4 – Air terjun simolap) total jaraknya adalah 51 km dan waktunya 1 jam 34 menit. Sehingga hasil pengujian proses algoritma *Steepest Ascent Hill Climbing* dengan sistem sesuai dengan proses perhitungan manual algoritma *Steepest Ascent Hill Climbing*.

Kata-kata Kunci: Algoritma *Steepest Ascent Hill Climbing* (SAHC), *Mobile*, Kecerdasan Buatan, Pencarian Rute Terpendek, Nilai Heuristik

ABSTRACT

Searching and determining the shortest route is a complicated problem, finding the shortest route from a number of tourist objects and the distance between tourist objects. With varied path access, finding the shortest route is the right choice using a mobile application. Based on the results of the Implementation of the Steepest Ascent Hill Climbing (SAHC) Algorithm for the Mobile-based Shortest Route Search Application in Humbang Hasundutan Regency. Then it can be concluded that the search for the shortest route based on Mobile can be completed using the Steepest Ascent Hill Climbing algorithm. In the manual calculation process by using the Steepest Ascent Hill Climbing algorithm on the origin node Humbang there is a heuristic value of 0.0896184808, at the origin node intersection there is a heuristic value of 0.1693780561, at the origin node sipinsur there is a heuristic value of 0.367474152, at the origin node of the promise there is a heuristic value amounted to 0.3043253189, and the last at the node of origin Sibabo waterfall there is a heuristic value of 0.3823982675. So the shortest route results from Sipinsur Geosite (F) to Simolap Waterfall (B) is $F \rightarrow D \rightarrow B$ (Sipinsur GeoSite - intersection 4 - Simolap Waterfall) the total distance is 51 km and the time is 1 hour 34 minutes. So the results of testing the process of the Steepest Ascent Hill Climbing algorithm with the system in accordance with the manual calculation process of the Steepest Ascent Hill Climbing algorithm.

Keywords : Steepest Ascent Hill Climbing (SAHC) Algorithm,
Mobile, Artificial Intelligence, Shortest Route Search, Heuristic
Value

KATA PENGANTAR

Assalamu'alaikum wr.wb.

Alhamdulillah, puji syukur kehadiran Allah SWT, yang telah melimpahkan rahmat, hidayah sertakarunia-Nya, dan menganugerahkan petunjuk untuk memudahkan dan melancarkan sehingga penulis dapat menyelesaikan proposal skripsi ini dengan judul “Perbandingan Algoritma *Dijkstra* dan Algoritma *Steepest Ascent Hill Climbing* Dalam Menentukan Rute Terpendek (Studi Kasus : Antar Lokasi Objek Wisata di Kabupaten Humbang Hasundutan”. Tak lupa juga sholawat serta salam semoga senantiasa tercurahkan kepada uswatun hasanah kita, Nabi Muhammad SAW beserta keluarga, para sahabat, dan pengikutnya termasuk kita semua yang senantiasa menantikan syafa'atnya kelak di hari akhir.

Demi kelancaran dalam penyelesaian laporan proposal skripsi ini tidak terlepas dari bantuan pihak terutama kepada Ayah dan Ibu yaitu Hasquizal dan Nurainun yang telah memberikan bantuan moril maupun materil, semangat dan do'a yang begitu besar kepada penulis.

Penulis menyadari bahwa tersusunnya proposal skripsi ini atas do'a, perhatian, bantuan, bimbingan, motivasi serta dukungan dari berbagai pihak, sehingga dengan keikhlasan dan kerendahan hati pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Saidurrahman, M.Ag, selaku Rektor Universitas Islam Negeri Sumatera Utara.
2. Bapak Dr. H. M. Jamil, MA, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
3. Bapak Dr. Mhd Furqan, S,Si., M.Comp.Sc selaku Ketua prodi Komputer dan dosen fakultas dan universitas akademik saya serta selaku dosen pembimbing skripsi I yang telah berkontribusi membantu penulis dalam memberikan ide, saran, keritik, dan bimbingannya kepada penulis selama penulis mengerjakan proposal skripsi ini.

4. Bapak Armansyah M.Kom sebagai dosen pembimbing skripsi yang telah berkontribusi membantu penulis dalam memberikan ide, saran, keritik, dan bimbingannya kepada penulis selama penulis mengerjakan proposal skripsi ini.
5. Seluruh tenaga pengajar dan pegawai program studi S1 Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
6. Teristimewa penulis sampaikan terima kasih dengan setulus hati kepada kedua orang tua tercinta, ayahanda Hasquizal dan ibunda Nurainun. Karena atas doa, kasih sayang, motivasi dan dukungannya saya dapat menyelesaikan studi sampai ke bangku sarjana.
7. Istri tercinta Dinda Syahrani Nst, S.Pd yang tanpa henti membantu dan menyemangati saya, memberikan perhatian serta memberikan motivasi kepada saya sehingga saya tersadar berusaha lebih baik dalam menyelesaikan Skripsi ini.
8. Dan semua pihak yang telah membantu penulis namun tidak dapat disebutkan satu persatu.

Penulis menyadari sepenuhnya bahwa dalam pembuatan proposal skripsi ini masih jauh dari kesempurnaan, untuk itu kritik dan saran yang bersifat membangun sangat penulis harapkan dari pembaca. Semoga hasil proposal skripsi ini menjadi Ibadah bagi penulis dan bermanfaat bagi pembaca. AamnYa Rabbal'alamin.

Medan, 11 November 2019

Hormat saya,

Razzaq H.Nur Wijaya

DAFTAR ISI

PERSETUJUAN SKRIPSI	ii
SURAT PERNYATAAN KEASLIAN SKRIPSI.....	i
PENGESAHAN SKRIPSI	iv
KATA PENGANTAR	v
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
ABSTRAKSI	xi
ABSTRACT.....	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
BAB PEMBAHASAN	5
2.1 Implementasi.....	5
2.1.1 Defenisi.....	5
2.2 Konsep Dasar Graf.....	5
2.2.1 Pengertian Graf.....	5
2.2.2 Istilah-Istilah Dalam Graf.....	6
2.2.3 Jenis-Jenis Graf	8
2.2.4 Sejarah Graf.....	9
2.2.5 Struktur Graf (Graph).....	10
2.3 Algoritma <i>Steepest Ascent Hill Climbing</i>	11
2.4 Contoh Algoritma	14
2.5 <i>Shortest Path</i>	15
2.6 Aplikasi	16
2.6.1 Aplikasi	16
2.7 <i>Mobile dan Android</i>	17

2.7.1 <i>Mobile</i>	17
2.7.2 <i>Android</i>	18
2.8 <i>Java, JavaScript dan PHP</i>	19
2.8.1 <i>Java</i>	19
2.8.2 <i>JavaScript</i>	21
2.8.3 <i>PHP</i>	22
2.8.3.1 <i>Sejarah PHP</i>	22
2.8.3.2 <i>Pengertian PHP</i>	23
2.9 <i>Android Studio</i>	23
2.10 <i>MySQL Database</i>	24
2.10.1 <i>Sejarah MySQL</i>	24
2.10.2 <i>Pengertian MySQL</i>	25
2.10.3 <i>Fungsi MySQL</i>	25
2.10.4 <i>Kelebihan dan Kekurangan MySQL</i>	26
2.10.4.1 <i>Kelebihan</i>	26
2.10.4.2 <i>Kekurangan</i>	26
2.11 <i>Penelitian yang Relevan</i>	27
BAB I METODOLOGI PENELITIAN	31
3.1 <i>Tempat dan Waktu</i>	31
3.1.1 <i>Tempat Penelitian</i>	31
3.1.2 <i>Waktu & Jadwal Pelaksanaan Penelitian</i>	31
3.2 <i>Teknik Pengumpulan Data</i>	32
3.2.1 <i>Studi Literatur</i>	32
3.2.2 <i>Observasi</i>	32
3.3 <i>Cara Kerja</i>	32
3.3.1 <i>Sistem</i>	32
3.3.2 <i>Analisis Kebutuhan</i>	33
3.3.3 <i>Tampilan Rancangan Sistem</i>	35
3.3.3 <i>Pengujian</i>	35
3.3.4 <i>Penerapan/Penggunaan</i>	36
BAB IV HASIL DAN PEMBAHASAN	37
4.1 <i>Pembahasan</i>	37
4.1.1 <i>Analisis Data</i>	37
4.1.2 <i>Hasil Analisis Data</i>	39

4.1.3 Perancangan	40
4.1.3.1 <i>Flowchart</i> Sistem	41
4.1.3.2 <i>Flowchart</i> Algoritma <i>Steepest Ascent Hill Climbing</i>	42
4.1.4.1 Pengujian	43
4.1.4.2 Perhitungan Manual Algoritma :	
<i>Steepest Ascent Hill Climbing</i>	45
BAB V KESIMPULAN DAN SARAN	48
5.1 Kesimpulan	48
5.2 Saran	48
DAFTAR PUSTAKA	51

DAFTAR GAMBAR

Gambar	Nama Gambar	Hal
2.1	Soal Algoritma <i>Steepest Ascent Hill Climbing</i>	14
2.2	Hasil Pencarian.....	14
3.1	Graf Rute di Kabupaten Humbang Hasundutan	34
3.2	<i>Flowchart</i> Algoritma SAHC.....	35
3.3	Rancangan Tampilan Aplikasi	36
4.1	<i>Web Google Maps</i> Untuk Mendapatkan Nilai Latitude dan Longitude	37
4.2	Input Objek Wisata.....	38
4.3	Objek Wisata yang Sudah di Input.....	38
4.4	Nilai Latitude dan Longitude pada Objek Wisata.....	39
4.5	<i>Flowchart</i> Sistem.....	41
4.6	<i>Flowchart</i> Algoritma <i>Steepest Ascent Hill Climbing</i>	42
4.7	Pengujian Proses Algoritma <i>Steepest Ascent</i> <i>Hill Climbing</i>	43
4.8	Hasil Rute Terpendek yang Dilintasi	44
4.9	<i>Node D</i> Menjadi <i>Current Node</i>	46
4.10	Hasil Pencarian Berhenti	47

DAFTAR TABEL

Tabel	Nama Tabel	Hal
2.1	Perjalanan Java dari Awal hingga Akhir.....	21
3.1	Waktu dan Jadwal Pelaksanaan Penelitian	31
4.1	Hasil Analisis Data.....	40
4.2	Nama Objek Wisata di Kab. Humbahas.....	45
4.3	Perhitungan Manual Algoritma <i>SAHC</i>	46

DAFTAR LAMPIRAN

Lampiran	Judul Lampir
1.	Listing Program
2.	Kartu Bimbingan
3.	Daftar Riwayat Hidup

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pencarian rute terpendek merupakan masalah yang rumit, sudah sering dibahas oleh peminat teknologi kecerdasan buatan. Hingga saat ini permasalahan pencarian rute tetap dilakukan melalui riset-riset yang ada. Sering kali pengguna jalan ditawarkan pada beberapa jalur atau jalan yang berbeda. Pemilihan rute secara manual sering kali kurang efektif karena bersifat secara analitik atau perkiraan dimana pengguna melakukan tahap coba-coba. Pencarian rute terpendek akan menjadi pilihan yang tepat manakala menggunakan sebuah aplikasi berbasis *mobile* yang memberikan solusi secara cepat dan mudah. Dalam teknologi kecerdasan buatan pencarian rute dapat dilakukan menggunakan algoritma pencarian. Ada banyak algoritma yang ditawarkan baik yang menggunakan teknik heuristik maupun secara serampangan yang sering disebut dengan *blind search*. Diantaranya yang menggunakan teknik heuristik adalah algoritma *Steepest Ascent Hill Climbing*. Algoritma *Steepest Ascent Hill Climbing* adalah metode algoritma yang banyak digunakan untuk permasalahan optimasi. Salah satu penerapannya adalah mencari rute yang terpendek dengan memaksimumkan atau meminimumkan nilai dari fungsi optimasi yang ada. Secara harfiah *Steepest* berarti paling tinggi. Sedangkan *Ascent* berarti kenaikan. Dengan demikian *Steepest Ascent* berarti kenaikan paling tinggi. Jadi prinsip dasar dari metode ini adalah mencari kenaikan paling tinggi dari keadaan sekitar untuk mencapai nilai yang paling optimal. Adapun ayat yang menjelaskan tentang kecerdasan buatan manusia terdapat dalam surah Fathir ayat 43 yang berbunyi :

أَسْتَكْبَرُوا فِي الْأَرْضِ وَمَكْرَ السَّيِّئِ وَلَا يَحِيقُ الْمَكْرَ السَّيِّئِ إِلَّا بِأَهْلِهِ
فَهَلْ يَنْظُرُونَ إِلَّا سُنَّتَ الْأَوَّلِينَ فَلَن تَجِدَ لِسُنَّتِ اللَّهِ تَبْدِيلًا وَلَن تَجِدَ
لِسُنَّتِ اللَّهِ تَحْوِيلًا ﴿٤٣﴾

Artinya : Karena kesombongan (mereka) di muka bumi dan karena rencana (mereka) yang jahat. Rencana yang jahat itu tidak akan menimpa selain orang yang merencanakannya sendiri. Tiadalah yang mereka nanti-nantikan melainkan sunnah kepada orang-orang yang terdahulu. Maka sekali-kali kamu tidak akan mendapatkan bagi sunnah allah, dan sekali-kali tidak pula akan menemui penyimpangan bagi sunnah allah itu.

Algoritma *Steepest Ascent Hill Climbing* merupakan algoritma yang menggunakan estimasi jarak terdekat (cost / jarak sebenarnya) untuk mencapai tujuan (*goal*). Dalam penelitian ini algoritma *Steepest Ascent Hill Climbing* akan menggunakan aplikasi *mobile* untuk pencarian rute objek wisata terdekat yang berada dikabupaten Humbang Hasundutan.

Algoritma *Steepest Ascent Hill Climbing* pernah dilakukan oleh Lipian Alfha Zemma, Herfina, Arie Qur'ania dalam penelitiannya berjudul Penerapan Metode *Steepest Ascent Hill Climbing* Pada Pencarian Rute Terdekat Fasilitas Pelayanan Darurat Di Kota Bogor Berbasis Android. Pada penelitian itu diperlukan beberapa data, yaitu jarak sesungguhnya antara titik/*node* yang berhubungan yang merupakan biaya/cost antara *node* dan koordinat setiap titik/*node*. Dengan menggunakan data-data tersebut maka pencarian rute terdekat bisa diterapkan. Proses implementasi algoritma SAHC (*Steepest Ascent Hill Climbing*) tidak bisa secara langsung dimasukkan pada pencarian rute di google maps, karena google maps sendiri tidak menyediakan layanan untuk memasukan serta menjalankan fungsi perubahan jalur yang diinginkan. Maka dari itu solusi untuk menerapkan algoritma SAHC (*Steepest Ascent Hill Climbing*) pada penelitian ini dilakukan secara manual dengan menggunakan data-data yang sudah didapatkan lalu di buat garis sesuai hasil perhitungan manual pada rute google maps dengan memasukan titik-titik koordinat yang telah terpilih dan menjadi rute terdekat ke dalam coding program di android studio.

Penelitian yang dilakukan oleh Harefa, dalam visualisasi barang menggunakan algoritma *Steepest Ascent Hill Climbing* untuk memvisualisasikan penyusunan barang pada sebuah gudang. Metode tersebut diterapkan untuk mengurutkan barang berdasarkan beratnya, sehingga urutan barang dimulai dari

barang yang memiliki berat paling besar hingga paling kecil. Barang kemudian disusun berdasarkan urutannya dengan menggunakan metode *block stacking*, yaitu barang akan disusun di atas barang sebelumnya dengan syarat berat barang yang akan disusun tidak boleh melebihi batas maksimum beban yang dapat ditampung barang yang berada di bawahnya. Hal-hal yang diperhatikan pada proses penyusunan barang di penelitian ini adalah beban maksimum yang dapat ditampung barang dan kapasitas ruang.

Kabupaten Humbang Hasundutan, yang berada di Sumatera Utara yang memiliki keindahan alam dan wisata bahari serta peninggalan sejarah yang sampai saat ini aktif dijadikan sebagai objek wisata. Beberapa objek wisata yang dapat dikunjungi di antara nya, 1) Air terjun sibabo; 2) Air terjun simolap; 3) Humbang Hasundutan; 4) Air terjun janji; 5) Sipinsur Geosite.

Keindahan alam dan wisata bahari yang ada memiliki potensi untuk dikembangkan. Pengembangan potensi objek wisata tentunya memerlukan sarana pendukung. Aplikasi pencarian rute terpendek merupakan salah satu pendukung manakala di rancang dengan teknologi yang sesuai.

Dalam mendukung perkembangan kabupaten dan objek wisata tersebut, disini peneliti ingin merancang aplikasi yang dimaksud diatas. Dengan judul penelitian “Implementasi Algoritma *Steepest Ascent Hill Climbing* (SAHC) Untuk Aplikasi Pencarian Rute Terpendek berbasis *Mobile*” diharapkan dapat berkontribusi dalam mengembangkan Kabupaten Humbang Hasundutan Khususnya pada bidang pariwisata.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah :

1. Bagaimana Menentukan Rute Terdekat dengan Algoritma *Steepest Ascent Hill Climbing*.
2. Bagaimana mengimplementasikan Algoritma *Steepest Ascent Hill Climbing* pada Aplikasi Berbasis *Mobile*.

1.3 Batasan Masalah

Terdapat beberapa batasan masalah pada penelitian ini yaitu:

1. Studi kasus yang dilakukan peneliti pada Kab.Humbang Hasundutan.
2. Peneliti menggunakan algoritma *Steepest Ascent Hill Climbing*.
3. Pengimplementasian pada penelitian ini berbasis *Android*
4. Pengambilan data menggunakan Aplikasi *Google maps*
5. Graf/rute menggunakan bantuan *Google maps*
6. Aplikasi yang dibuat peneliti ke dalam aplikasi *mobile* dengan bahasa pemrogram *Java,Java Script, Php* dan *Tools* untuk memampikan interface aplikasi menggunakan *Android Studio*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengimplementasikan algoritma *Steepest Ascent Hill Climbing* pada aplikasi pencarian rute terpendek berbasis *Mobile Android*.

1.5 Manfaat Penelitian

Manfaat penelitian ini adalah :

1. Membantu pemerintah daerah dalam menyediakan sarana pendukung untuk pengembangan potensi pariwisata daerah.
2. Memberikan kontribusi dalam penelitian maupun pendidikan berupa referensi untuk riset-riset yang menggunakan metode yang sama.

BAB

LANDASAN TEORI

2.1 Implementasi

2.1.1 Defenisi

Secara etimologis pengertian implementasi menurut Kamus Webster yang dikutip oleh Solichin Abdul Wahab (2004) (Webster dalam Wahab (2004:64) adalah : Konsep implementasi berasal dari bahasa inggris yaitu to implement.

Dalam kamus besar webster, to implement (mengimplementasikan) berarti to provide the means for carrying out (menyediakan sarana untuk melaksanakan sesuatu) dan to give practical effect to (untuk menimbulkan dampak/akibat terhadap sesuatu) Implementasi berasal dari bahasa Inggris yaitu to implement yang berarti mengimplementasikan. Implementasi merupakan penyediaan sarana untuk melaksanakan sesuatu yang menimbulkan dampak atau akibat terhadap sesuatu. Sesuatu tersebut dilakukan untuk menimbulkan dampak atau akibat itu dapat berupa undang–undang, Peraturan Pemerintah, Keputusan Peradilan dan Kebijakan yang dibuat oleh Lembaga–Lembaga Pemerintah dalam kehidupan kenegaraan.

Implementasi adalah pelaksanaan atau penerapan. Artinya yang dilaksanakan dan diterapkan adalah kurikulum yang telah dirancang atau di desain untuk dijalankan sepenuhnya. Maka implementasi kurikulum juga dituntut untuk melaksanakan sepenuhnya apa yang telah direncanakan dalam kurikulumnya, permasalahan besar yang akan terjadi apabila yang dilaksanakan bertolak belakang atau menyimpang dari yang telah dirancang maka terjadilah kesia - siaan antara rancangan dengan implementasi.

2.2 Konsep Dasar Graf

2.2.1 Pengertian Graf

Pengertian Graf Dalam konsep dasar graf diketahui bahwa definisi dari graf dan unsur-unsurnya akan disusun dengan menggunakan bahasa himpunan.

Oleh sebab itu, sebelum membahas mengenai definisi dari graf, akan dijelaskan terlebih dahulu mengenai syarat dari sebuah himpunan. Dimana, suatu himpunan sendiri disyaratkan bahwa setiap elemen didalamnya hanya akan muncul sekali saja. Graf sendiri merupakan “alat” yang digunakan untuk merepresentasikan objek-objek pada suatu diskrit dan hubungan yang terdiri antara objek-objek tersebut.

Graf (G) didefinisikan dengan sebuah pasangan himbunan yang terdiri dari himpunan tak kosong dari simpul-simpul yang biasanya disimbolkan dengan (V /vertice), dan himpunan rusuk yang disimbolkan dengan (E /edge). Dimana, baik V/E menghasilkan sepasang simpul pada G . Kemudian, himpunan simpul pada G akan dinotasikan sebagai V , dan himpunan rusuk pada G akan dinotasikan dengan E . Sehingga, akan menghasilkan rumus sebagai berikut $G=(V, E)$.

Sedangkan menurut Siang (2002: 187), menyatakan bahwa pada suatu graf G akan terdoro dari dua himpunan yang berhingga. Sehingga, suatu graf G terdiri dari himpunan V dan E , yang kemudian akan dituliskan dengan (V, E) , dengan V yang merupakan himpunan berhingga, dan E yang terdiri dari himpunan rusuk yang bersisian dengan V .

Dengan demikian, dapat disimpulkan bahwa graf G merupakan sebuah pasangan himpunan yang mana di dalamnya akan terdiri dari himpunan tak kosong dan himpunan rusuk yang keduanya kemudian disimbolkan dengan (V dan E). Kemudian, untuk mempermudah penjelasan mengenai graf G dapat dilakukan dengan hubungan dua titik, dua sisi dan simpul diberikan nama tertentu.

2.2.2 Istilah-Istilah dalam Graf

Istilah-istilah dalam Graf Terdapat beberapa istilah yang biasanya sangat sering digunakan dalam graf G , diantaranya sebagai berikut:

1. Gelang (Loop): merupakan sebuah rusuk yag dapat dikatakan gelang jika ujung rusuknya memiliki awal dan akhir pada simpul yang sama.

2. Rusuk Ganda (Multiple Edges): merupakan sebuah penamaan yang diberikan pada suatu graf G yang mempunyai lebih dari satu rusuk dengan sepasang simpul yang sama.

3. Bertetangga (Adjacent): merupakan dua buah simpul yang terdapat pada graf yang tidak berarah G yang dikatakan bertetangga apabila keduanya terhubung secara langsung dengan sebuah rusuk yang sama.

4. Bersisian (Incident): jadi untuk sembarang rusuk $e = (u, v)$, rusuk e akan dikatakan berisikan dengan simpul u dan simpul v .

5. Graff kosong (Null Graph atau Empty Graph): merupakan suatu graf G yang himpunan dalam rusuknya merupakan sebuah himpunan kosong.

6. Perjalanan (Walk): merupakan sebuah barisan berganti-ganti diantaranya simpul dan juga rusuk dari G , yang akan diawal dengan simpul u dan kemudian diakhiri dengan simpul v .

7. Lintasan (Path): merupakan sebuah barisan yang berselang-seling antara simpulsimpul dan rusuk-rusuk dari sebuah graf G .

8. Siklus (Cycle) dan Sirkuit (Cirkuit): merupakan sebuah lintasan yang akan berawal dan berakhir pada sebuah simpul yang sama.

9. Terhubung (Connected): dua simpul dalam graf G , akan dikatakan terhubung apabila terdapat sebuah lintasan, apabila setiap simpul dalam graf terhubung, maka graf tersebut disebut dengan graf terhubung.

10. Graf Berbobot (Weighted Graph): merupakan sebuah graf yang pada setiap rusuknya diberi sebuah harga atau bobot, dimana pada tiap-tiap rusuknya akan memiliki bobot yang berbeda-beda, tergantung kepada masalah yang dimodelkan dengan graf tersebut.

2.2.3 Jenis-Jenis Graf

Jenis-jenis Graf Menurut Munir (2005: 357), yang menyatakan bahwa graf G akan dikelompokkan kedalam beberapa jenis yang sesuai dengan sudut pandang pengelompokannya. Dimana, pengelompokan graf dapat dinilai melalui tidak adanya rusuk yang ganda, berdasarkan dengan jumlah simpul, maupun berdasarkan dari orientasi arah pada rusuk tersebut.

Secara umum, graf sendiri digolongkan kedalam dua jenis, yaitu graf sederhana dan graf tidak sederhana, yang akan dijelaskan sebagai berikut:

1. Graf Sederhana (Simple Graph) Menurut Harju (2012), graf sederhana tidak memiliki rusuk ganda dan atau, gelang. Dimana, rusuk merupakan sebuah pasangan yang tidak terurut (unordered pairs). Selain itu, menurut Munir (2005), graf sederhana juga dapat diartikan sebagai graf yang terdiri dari himpunan tak kosong simpul-simpul dan himpunan pasangan tidak terurut yang berbeda yang kemudian disebut sebagai rusuk. Dimana, menurut Siang (2002) dalam beberapa graf khusus sering menggunakan graf, sebagai berikut:
 - a. Graf Lengkap (Complete Graph): merupakan sebuah graf sederhana yang pada setiap dua simpulnya saling bertetangga.
 - b. Graf Lingkaran: merupakan sebuah graf sederhana yang pada setiap simpulnya berderajat dua.
 - c. Graf Teratur(Regular Graph): merupakan sebuah graf yang pada setiap simpulnya memiliki derajat yang sama.
 - d. Graf Bipartit (Bipartit Graph): merupakan sebuah graf yang mana simpul V_1 terhubung dengan simpul pada V_2 yang kemudian dinyatakan dengan $G(V_1, V_2)$.
2. Graf Tak Sederhana (Unsimple Graph)

Graf Tak Sederhana (Unsimple Graph) Harju (2012), menyatakan bahwa graf yang didalamnya mengandung rusuk ganda atau gelang akan dinamakan dengan graf tak sederhana (unsimple graph). Terdapat dua

macam graf dalam graf tak sederhana, yaitu graf ganda (multigraph) dan graf semu (pseudigraph). Dimana, graf ganda merupakan graf yang didalamnya mengandung rusuk ganda. Sedangkan, graf semu merupakan graf yang didalamnya mengandung gelang (loop).

Selain itu, Menurut Bondy dan Murty (1982), graf tak sederhana juga dapat dikelompokkan berdasarkan orientasi arah pada rusuknya. Dimana, orientasi arah tersebut dibagi menjadi dua yaitu graf tak berarah dan graf berarah.

- a. Graf Tak Berarah (Undirected Graph): merupakan sebuah graf yang rusuknya tidak mempunyai orientasi arah.
- b. Graf berarah (Directed Graph): merupakan sebuah graf yang setiap rusuknya mempunyai orientasi arah.

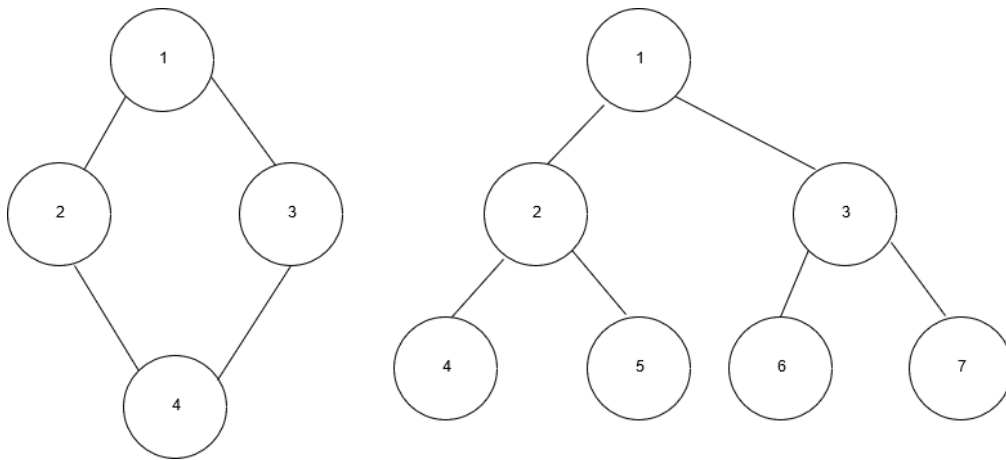
2.2.4 Sejarah Graf

Menurut catatan sejarah, graf pertama kali digunakan untuk menyelesaikan permasalahan jembatan *Konigsberg* pada tahun 1736. Masalah jembatan *Konigsberg* ini adalah : mungkinkah melalui ketujuh buah jembatan itu masing-masing tepat satu kali dan kembali lagi ketempat semula. Kemudian pada tahun 1736 seorang matematikawan Swiss, L.Euler, adalah orang pertama yang berhasil menemukan jembatan masalah itu dengan memodelkan masalah ini ke dalam graf. Euler mengungkapkan bahwa tidak mungkin seseorang berjalan melewati tepat satu kali masing-masing jembatan dan kembali lagi ke tempat semula karena pada graf model jembatan *Konigsberg* itu tidak semua simpul berderajat genap (derajat sebuah simpul adalah jumlah sisi yang berkaitan dengan simpul yang bersangkutan).

Pada sebuah graf dapat dilalui setiap sisinya masing-masing satu kali dan kembali lagi ke tempat semula, maka graf tersebut dikatakan memiliki sirkuit Euler. Sampai saat ini graf telah berkembang penggunaannya menjadi jauh lebih luas.

2.2.5 Struktur Graf (Graph)

Graf merupakan struktur data yang terdiri atas kumpulan *vertex* (V) dan *edge* E , biasa ditulis sebagai $G = (V, E)$, dimana *vertex* adalah *node* paragraf dan *edge* adalah rusuk atau jaring yang menghubungkan dua *node*. Jaringan terdefinisi melalui pasangan *node* (v, w) , dimana v disebut *tail* dan w disebut *head* dari jaringan tersebut. Contoh beberapa graf diberikan berikut ini : (Suarga. 2012 : 224-226) dalam buku algoritma dan pemograman.



Pada graf tersebut dapat didefinisikan himpunan *vertex* dan himpunan *edge* sebagai berikut :

$$V(G_1) = \{ 1, 2, 3, 4 \}; \quad E(G_1) = \{ (1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4) \}$$

$$V(G_2) = \{ 1, 2, 3, 4, 5, 6, 7 \}; \quad E(G_2) = \{ (1, 2), (1, 3), (2, 4), (2, 5), (3, 6), (3, 7) \}$$

Ada dua kategori graf yaitu :

1. *Undirected graph* : graf dimana edge tidak memiliki arah, seperti G_1 dan G_2
2. *Directed graph* : graf dimana edge memiliki arah

Beberapa istilah yang sering digunakan dalam masalah graf antara lain sebagai berikut :

1. *Adjacent vertex* : adalah dua *node* berdekatan, terhubung langsung oleh vertex

2. *Path* : jalur melalui edge yang menghubungkan satu *vertex* ke *vertex* yang lain, panjang suatu jalur ditentukan oleh jumlah jaringan (*edge*) yang menghubungkan dua *vertex*
3. *Completed graph* : adalah graf dimana semua *vertex* terhubung langsung satu dengan yang lain.
4. *Weighted graph* : graf yang setiap *edge* nya memiliki bobot atau nilai
5. *Cycle* : adalah jalur yang mulai dari suatu *vertex* dan berakhir pada *vertex* yang sama.

2.3 Algoritma *Steepest Ascent Hill Climbing*

Metode heuristik adalah bagian bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian (Mutakhiroh et al. 2007). Asal mula kata heuristik berasal dari kata kerja bahasa Yunani, yaitu heurisken yang artinya ‘mencari’ atau ‘menemukan’. Proses pencarian pada metode ini dilakukan dengan selektif dan dapat memberikan panduan untuk proses pencarian berikutnya yang memiliki kemungkinan sukses paling besar (Berlianty et al. 2010). Beberapa diantaranya adalah algoritma Genetika, Algoritma Genetika menggunakan prinsip menemukan solusi neighborhood berdasarkan seleksi alam dan mekanisme genetik alami yang dapat digunakan untuk menyelesaikan masalah optimisasi .(furqan.mhd,dkk.,2018) algoritma A*, algoritma Simple Hill Climbing dan algoritma Steepest – Ascent Hill Climbing.

Cara kerja dari algoritma Stepest Ascent Hill Climbing adalah sebagai berikut, lakukan tahap-tahap ini sampai diperoleh goal state atau sampai tidak ada lagi operator (state) yang ingin digunakan:

1. Pilih operator bebas untuk digunakan sebagai current state. Jadikan operator tersebut untuk mencari new state.
2. Evaluasi current state dan dapatkan semua successor untuk dijadikan sebagai next state. Kemudian evaluasi setiap successor dan beri nilainya.
3. Jika ada satu successor yang ditemukan memiliki nilai terbaik dari current statenya, maka gunakan successor itu menjadi new current state. Lakukan operasi 1 sampai 3 hingga ditemukan current state sama dengan goal state.

Steepest Ascent Climbing adalah jenis algoritma pencarian berdasarkan nilai *heuristic* terbaik. Dalam hal ini penggunaan operator tidak menentukan penemuan solusi. *Steepest Ascent Climbing* merupakan metode algoritma yang banyak digunakan untuk permasalahan optimal.

Metode *Steepest Ascent Climbing* merupakan pengembangan dari metode *Simple hill climbing*. Bedanya adalah *Simple hill climbing* menentukan *next state* dengan membandingkan *current state* dengan salah satu *successor* dan *successor* pertama yang lebih baik akan dipilih menjadi *next state*. Sedangkan pada *Steepest Ascent Climbing* akan membandingkan *current state* dengan semua *successor* yang ada didekatnya sehingga dalam *Steepest Ascent Climbing* *next state* nya merupakan *successor* yang paling baik atau paling mendekati tujuan. (Wisnu, Adam, Risa. 2015 : 23) dalam jurnal Pembuatan Game *Slider Puzzle* Menggunakan Metode *Steepest Ascent Hill Climbing* berbasis Android.

Steepest Ascent Hill Climbing memiliki kemiripan dengan algoritma *Simple Hill Climbing*, namun gerakan pencarian tidak dimulai dari posisi paling kiri. Gerakan pencarian selanjutnya ditentukan berdasarkan nilai heuristik terbaik. Dalam hal ini urutan penggunaan operator tidak menentukan penemuan solusi (Kusumadewi *et al.* 2005 : 58) dalam buku Aplikasi Logika *Fuzzy* untuk Pendukung, Keputusan Bedanya adalah *Simple Hill Climbing* menentukan titik selanjutnya dengan membandingkan titik saat ini dengan satu *successor* (titik persimpangan) dan *successor* pertama yang lebih baik akan dipilih menjadi titik selanjutnya. Sedangkan *Steepest Ascent Hill Climbing*, akan membandingkan titik selanjutnya dengan semua *successor* yang ada di dekatnya sehingga dalam *Steepest Ascent Hill Climbing* titik selanjutnya adalah *successor* yang paling baik dan mendekati tujuan.

Algoritma ini memeriksa titik (node), yaitu biaya (cost) yang dibutuhkan untuk mencapai sebuah node dan heuristik node / $h(n)$ yaitu cost yang didapat dari node ke tujuan. Sehingga dapat dirumuskan sebagai berikut :

$$H(n) = \sqrt{(X_{asal} - X_{tujuan})^2 + (Y_{asal} - Y_{tujuan})^2}$$

Keterangan :

- $h(n)$ = perkiraan heuristik atau cost atau path dari node n ke tujuan.
Untuk menentukan nilai $h(n)$ ditunjukkan oleh persamaan 1, berikut :

$$1. h(n) =$$

dimana :

$h(n)$: nilai heuristik untuk node / titik n

Xasal : nilai Latitude asal dari node/titik asal

Xtujuan : nilai Latitude tujuan dari node/titik tujuan

Yasal : nilai Longatititude asal dari node/titik asal

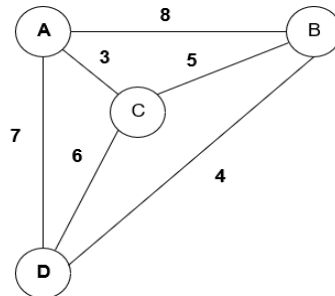
Ytujuan : nilai longatititude tujuan dari node/titik tujuan

Langkah-langkah algoritma *Steepest Ascent Hill Climbing* adalah sebagai berikut :

1. Mulai dari keadaan awal dalam melakukan pengujian. Jika merupakan suatu tujuan, maka berhenti. Jika tidak lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kemudian ulangi hingga solusi atau tujuan ditemukan atau hingga iterasi tidak memberikan perubahan pada keadaan sekarang :
 - a. Tentukan *SUCC* sebagai nilai heuristik terbaik dari *successor* – *successor*.
 - b. Lakukan untuk masing – masing *node* yang digunakan oleh keadaan sekarang:
 - c. Gunakan *node* tersebut dan bentuk keadaan baru.
 - d. Evaluasi keadaan terbaru tersebut. Jika merupakan tujuan maka keluar. Jika tidak, maka bandingkan nilai heuristiknya dengan *SUCC*. Jika lebih baik, tetapkan nilai heuristik keadaan baru tersebut sebagai *SUCC*. Namun jika tidak lebih baik, maka nilai *SUCC* tidak berubah.
 - e. Jika *SUCC* lebih baik dari nilai heuristik keadaan sekarang, maka ubah *node SUCC*, menjadi keadaan sekarang.

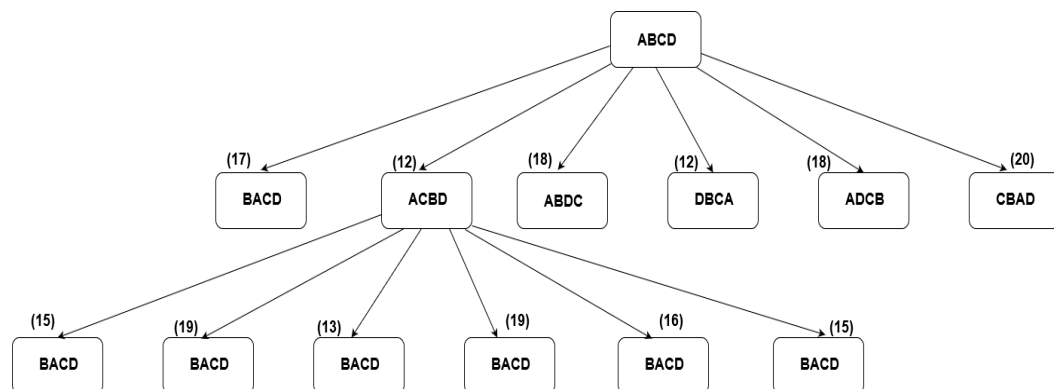
2.4 Contoh Algoritma *Steepest Ascent Hill Climbing*

Adapun contoh pada algoritma *Steepest Ascent Hill Climbing* adalah :



Gambar 2.1 Soal algoritma *Steepest Ascent Hill Climbing*

Misalkan pada contoh kasus TSP ada 4 kota dengan jarak antara tiap –tiap kota seperti terlihat pada Gambar 2.11. Maka pencarian dengan Steepest Ascent Hill Climbing ini dapat dilihat pada Gambar 2.12. Terlihat bahwa, keadaan awal, lintasan terpilih adalah ABCD(19). Pada level pertama, akan memilih nilai heuristik terbaik dari keenam succesor yang ada, yaitu BACD(17), ACBD(12), ABDC(18), DBCA(12), ADCB(18) atau CBAD(20). Maka yang terpilih adalah ACBD, karena memiliki nilai heuristik paling kecil (=12). Selanjutnya dari ACBD ini akan dipilih nilai heuristik terbaik dari succesor yaitu CABD(15), ABCD(19), ACBD(13), DCBA(19), ADBC(16) atau BCAD(15). Ternyata diketahui dari keenam succesor tersebut memiliki nilai heuristik yang lebih besar dibanding dengan ACBD. Sehingga tidak akan ada perubahan nilai keadaan (tetap ACBD). Hasil yang diperoleh, lintasannya adalah ACBD(12) (Kusumadewi et al. 2005).



Gambar 2.2 Hasil Pencarian (Kusumadewi et al. 2005)

2.5 Shortest Path

Lintasan terpendek (*shortest path*) adalah lintasan minimum yang diperlukan dalam mencapai suatu tempat dari tempat tertentu. Dalam menemukan lintasan terpendek antara dua atau lebih simpul yang berhubungan. Persoalan mencari lintasan terpendek merupakan salah satu persoalan optimasi. Persoalan ini biasanya direpresentasikan dalam bentuk graf, dimana keadaan yang berhubungan dengan ruang lingkup pencarian direpresentasikan oleh verteks dan transisi yang terjadi digambarkan dalam bentuk edge. Perpindahan dari satu verteks ke verteks yang lainnya dapat dilakukan dengan cara menerapkan operator logika yang akan memanipulasi keadaan awal atau dengan menggunakan *mobile agent* yang bergerak dari satu verteks ke verteks lainnya. Persoalan lintasan terpendek banyak dijumpai di kehidupan sehari-hari. Aplikasi yang paling sering ditemui adalah pada bidang transportasi dan komunikasi, seperti pada pencarian rute terbaik untuk menempuh dua kota atau untuk mengetahui dan menelusuri proses pengiriman paket data komunikasi dalam suatu jaringan komunikasi agar dihasilkan suatu proses yang paling cepat.

Shortest path merupakan pencarian rute atau *path* terpendek antara *node* yang ada pada *graph*, biaya (*cost*) yang dihasilkan adalah minimum. Sedangkan menurut dublin (2009 : 210) dalam buku mencari rute terpendek menemukan jalan terpendek dari total panjang jalan antara dua *node* grafik diarahkan dengan panjang berkaitan dengan tepi masing-masing. Dalam pencarian lintasan terpendek masalah yang dihadapi adalah mencari lintasan mana yang akan dilalui sehingga didapat lintasan yang paling pendek dari satu verteks ke verteks yang lain. Ada beberapa macam persoalan lintasan terpendek, antara lain : 1) Lintasan terpendek antara dua buah vertex; 2). Lintasan terpendek antara semua pasangan vertex; 3) Lintasan terpendek dari verteks tertentu ke semua verteks yang lain; 4) Lintasan terpendek antara dua buah verteks yang melalui beberapa verteks tertentu.

2.6 Aplikasi

2.6.1 Aplikasi

Menurut Hasan Abdurahman dan Asep Ririh Riswaya (2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Pengertian aplikasi menurut para ahli :

- a. Pengertian aplikasi menurut Jogiyanto (1999:12) adalah penggunaan dalam suatu komputer, instruksi(*instruction*) atau pernyataan(*statement*) yang disusun sedemikian sehingga komputer dapat memproses input menjadi output.
- b. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.
- c. Menurut Wikipedia, aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

2.7 Mobile dan Android

2.7.1 Mobile

Menurut Pressman dan Bruce (2014:9), mobile adalah rancangan khusus untuk platform mobile (misalnya iOS, android, atau windows mobile). Dalam banyak kasus, mobile memiliki user interface dengan mekanisme interaksi unik yang disediakan oleh platform mobile, interoperabilitas dengan sumber daya berbasis web yang menyediakan akses ke beragam informasi yang relevan dengan mobile, dan kemampuan pemrosesan lokal untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk platform mobile. Selain itu mobile menyediakan kemampuan penyimpanan persisten dalam platform.

Aplikasi mobile merupakan peningkatan dari sistem perangkat lunak terpadu yang umumnya ditemukan pada PC Desktop. Pada awal kemunculannya, aplikasi menyediakan fungsionalitas yang terbatas dan terisolasi seperti permainan, kalkulator atau mobile Web browsing sehingga aplikasi sangat dihindari karena multitasking yang ‘memakan’ sumber daya hardware perangkat mobile yang awalnya cukup terbatas. Namun, jaman sekarang ini dengan dukungan sumber daya hardware yang lebih tinggi, aplikasi telah menjadi hal yang tak terpisahkan dengan perangkat mobile sebab mereka mampu melakukan apapun dengan mudah.

Aplikasi mobile paling sederhana mengambil aplikasi berbasis PC dan porting ke perangkat mobile. Kini para ahli mengembangkan aplikasi khusus untuk lingkungan mobile, mengambil keuntungan dari keterbatasan dan keuntungan. Sebagai contoh, aplikasi yang menggunakan fitur-fitur berbasis lokasi inheren dibangun dari chip micro mobile, ini mengingatkan bahwa smartphone mobile tidak memiliki konsep yang dengan PC komputer.

Kemudian mulailah muncul berbagai aplikasi – aplikasi yang di buat oleh para developer. Mulai dari aplikasi mobile yang membantu para penggunanya untuk lebih mudah berinteraksi satu sama lain, membantu para penggunanya dalam hal jual – beli barang, hingga aplikasi petunjuk jalan. Aplikasi – aplikasi ini di buat Karena banyaknya keluhan – keluhan dari masyarakat sendiri yang

membuat developer untuk membangun aplikasi – aplikasi tersebut. Dan hingga munculnya aplikasi yang menggunakan suara untuk dapat menggunakan aplikasi tersebut contohnya seperti *Google Assistant* yang dibuat oleh Google. Dari aplikasi ini akan membuka peluang baru lagi bagi para developer untuk terus mengembangkan aplikasi – aplikasi seperti *Google Assistant* ini.

2.7.2 Android

Menurut Nazrudin safaat H (2012:1) menyatakan android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyiapkan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Android adalah Platform/aplikasi yang bebas untuk *developer*. Tidak ada lisensi atau biaya atau biaya royalti untuk dikembangkan pada platform Android. Android merupakan generasi baru platform *mobile*, platform yang memberikan pengembang untuk melakukan sesuai yang diharapkannya. Sistem operasi yang mendasari Android di lisensikan dibawah GNU, General Public Lisensi Versi 2 (GPLv2), yang sering dikenal dengan istilah “copyleft” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh dibawah term. Android juga didistribusikan dibawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya (Safaat, N., 2012).

Android adalah sebuah sistem operasi untuk ponsel yang berbasis Linux. Android SDK (Software Development Kit) menyediakan tools dan API (*Application Programming Interface*) yang diperlukan bagi para pengembang untuk membuat dan mengembangkan aplikasi yang digunakan pada ponsel bersistem operasi Android dengan menggunakan bahasa pemrograman Java. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras,

piranti lunak, dan piranti telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia(Hermawan, S., 2011).

Mobile kata sifat yang berarti dapat bergerak bahkan dapat digerakkan dengan bebas dan mudah. Namun mobile dapat pula diartikan sebuah benda yang berteknologi tinggi dan dapat bergerak tanpa menggunakan kabel, contohnya seperti *smartphone*. Di dalam *smartphone* terdapat sistem operasi android yang berbasis Linux menyediakan tools yang diperlukan bagi para pengembang untuk membuat dan mengembangkan aplikasi yang digunakan pada ponsel bersistem operasi Android dengan menggunakan bahasa pemrograman Java.

2.8 Java, JavaScript dan PHP

Dalam membuat aplikasi ini terdapat 3 bahasa pemrograman yang di pakai yaitu *Java, JavaScript dan Php*.

2.8.1 Java

Java adalah bahasa pemrograman berorientasi objek murni dibuat berdasarkan kemampuan - kemampuan terbaik, bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991.

Bahasa pemrograman java pada awalnya dikhususkan untuk aplikasi berbasis internet, namun sekarang aplikasi-aplikasi java sudah dapat digunakan untuk tidak hanya berbasis Web saja tetapi dari basis Desktop hingga aplikasi Mobile sudah menggunakannya. Java itu portable yaitu dapat dijalankan di berbagai platform yaitu Windows, Linux, Unix, MacOS, Android (Mobile). Kunci dari portabilitas ini adalah keluaran hasil kompilasi java bukanlah file executable melainkan berbentuk bytecode. Bytecode inilah yang akan dieksekusi oleh JRE (Java Runtime Environment) yang disebut JVM (Java Virtual Machine) dan telah diinstall pada setiap sistem operasi yang akan digunakan, sehingga JVM merupakan interpreter bagi bytecode. interpreter merupakan perangkat lunak yang berfungsi

melakukan eksekusi sejumlah instruksi yang ditulis dalam suatu bahasa pemrograman. (Bambang, 2011 : 2).

Berikut Tabel Perjalanan Java dari Awal hingga Kini :

Tahun	Versi	Perkembangan Java
1991-1994	-	Pertama kali diciptakan dengan nama Oak.
1995	-	Nama Oak diganti menjadi Java.
1996	Java kit 1.0	Sistem operasi Java diluncurkan.
1997-2000	Java kit 1.1, Java kit 1.2, Java kit 1.3	Java Community Process (JCP) dibentuk untuk membuat mekanisme pengembangan dari pihak luar.
2001-2004	J2SE 1.4. J2SE 5.0	Website Java.com dirilis supaya pengguna bisa download dan menjalankan aplikasi Java di desktop dan laptop.
2005	-	10 tahun Java, Java telah digunakan oleh 4,5 juta developer dan 2,5 miliar perangkat.
2006-2008	JAVA SE 6	- Java tersedia dalam versi open source di bawah GNU (General Public License). – Pertama kalinya Java Virtual Machine (JVM) diluncurkan.
2009-2011	JAVA SE 7	Oracle mengakuisisi Sun. Oracle launching Java Magazine untuk publikasi teknikal bagi para developer.
2012-2014	JAVA SE 8	Publikasi Java 8 di terjemahkan dalam 8 bahasa dunia

2015	-	- Java menjadi platform bahasa pemrograman nomor satu di dunia. – Telah dijalankan di 13 miliar perangkat.
2016	-	java Magazine telah di subscribe 250.000 orang dan dijalankan di lebih dari 15 miliar perangkat.
2017	JAVA SE 9	Ada 38 juta JVM yang aktif digunakan.
2018	JAVA SE 10, JAVA SE 11	Fitur terbaru termasuk HTTP client untuk website.
2019	JAVA SE 12	Java telah digunakan berbagai aplikasi mobile, seperti Spotify, Siemens, NASA Open Code Project, Minecraft, dll

Tabel 2.1 Perjalanan Java Dari Awal Hingga Kini

2.8.2. JavaScript

JavaScript adalah *JavaScript* adalah bahasa pemrograman web bersifat Client Side Programming Language. Client Side Programming Language adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh client. Aplikasi client yang dimaksud merujuk kepada web browser seperti *Google Chrome*, *Mozilla Firefox*, *Opera Mini* dan sebagainya. *JavaScript* pertama kali dikembangkan pada pertengahan dekade 90'an. Meskipun memiliki nama yang hampir serupa, JavaScript berbeda dengan bahasa pemrograman Java. Untuk penulisannya, *JavaScript* dapat disisipkan di dalam dokumen *HTML* ataupun dijadikan dokumen tersendiri yang kemudian diasosiasikan dengan dokumen lain yang dituju. *JavaScript* mengimplementasikan fitur yang dirancang untuk mengendalikan bagaimana sebuah halaman web berinteraksi dengan penggunanya (Henderson, 2009:256).

2.8.3 PHP

2.8.3.1 Sejarah PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah dplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan

ditambahkan pada versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017.

2.8.3.2 Pengertian PHP

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman open source yang sangat cocok atau dikhususkan untuk pengembangan web dan dapat ditanamkan pada sebuah skripsi *HTML*. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti *C*, *Java*, dan *Perl* serta mudah untuk dipelajari. *PHP* merupakan bahasa scripting *server – side*, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain *PHP* adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode *HTML*". Menurut Kustiyaningsih (2011:114), "PHP (atau resminya *PHP: Hypertext Preprocessor*) adalah skrip bersifat server – side yang ditambahkan ke dalam *HTML*".

2.9 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada *IntelliJ IDEA*. Selain sebagai editor kode dan fitur developer *IntelliJ* yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, seperti:

- Sistem build berbasis *Gradle* yang fleksibel
- Emulator yang cepat dan kaya fitur

- Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android
- Terapkan Perubahan agar dapat melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- Template kode dan integrasi GitHub dapat membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- Framework dan fitur pengujian yang lengkap
- Fitur lint dapat merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- Dukungan C++ dan NDK
- Dukungan bawaan pada Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

2.10 MySQL Database

2.10.1 Sejarah MySQL

MySQL AB merupakan perusahaan komersial Swedia yang mensponsori dan yang memiliki MySQL. Pendiri MySQL AB adalah dua orang Swedia yang bernama David Axmark, Allan Larsson dan satu orang Finlandia bernama Michael “Monty”. Setiap pengguna MySQL dapat menggunakannya secara bebas yang didistribusikan gratis dibawah lisensi GPL(*General Public License*) namun tidak boleh menjadikan produk turunan yang bersifat komersial.

Pada saat ini MySQL merupakan database server yang sangat terkenal di dunia, semua itu tak lain karena bahasa dasar yang digunakan untuk mengakses database yaitu SQL. SQL (*Structured Query Language*) pertama kali diterapkan pada sebuah proyek riset pada laboratorium riset San Jose, IBM yang bernama system R. Kemudian SQL juga dikembangkan oleh Oracle, Informix dan Sybase. Dengan menggunakan SQL, proses pengaksesan database lebih user-friendly dibandingkan dengan yang lain, misalnya dBase atau Clipper karena mereka masih menggunakan perintah-perintah pemrograman murni.

2.10.2 Pengertian *MySQL*

MySQL adalah sebuah database manajemen sistem (DBMS) populer dan memiliki fungsi sebagai relational database manajemen sistem (RDBMS). Selain itu *MySQL* software merupakan aplikasi yang sifatnya *open source* serta server basis data *MySQL* memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur client server atau *embedded systems*. Dikarenakan faktor *open source* dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data.

2.10.3 Fungsi *MySQL*

Database *MySQL* sebagai wadah yang menyimpan string (text based) data. Gambar, media, file, file audio, dan hal hal alam yang benar benar tidak harus disimpan dalam database. Caranya adalah dengan meletakkan file file pada server dalam foler dan hanya referensi nama dan path di database tersebut.

Berikut adalah beberapa jenis aplikasi PHP dan *MySQL* dapat membantu anda membuat:

1. Situs Blog
2. Custom Database Driver Website Dinamis (seperti developPHP)
3. E-commerce dan Pemrograman Custom Toko Online
4. Jaringan Sosial dan omunitas
5. Guestbook
6. Seluruh Website Dinamis
7. Klien dan informasi Pelanggan.

2.10.4. Kelebihan dan Kekurangan *MySQL*

2.10.4.1. Kelebihan

Adapun kelebihan MySQL dalam penggunaannya dalam database adalah:

- Free atau gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya.
- MySQL stabil dan tangguh dalam pengoperasiannya
- My SQL mempunyai sistem keamanan yang cukup baik
- Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas
- Sangat fleksibel dengan berbagai macam program
- Perkembangan dari MySQL sangat cepat

2.10.4.2 Kekurangan

Kekurangan yang dimiliki oleh MySQL, diantaranya:

- Kurang mendukung koneksi bahasa pemrograman seperti Visual basic atau biasa kita kenal dengan sebutan VB, Foxpro, Delphi dan lain-lain sebab koneksi ini menyebabkan field yang dibaca harus sesuai dengan koneksi dari bahasa pemrograman visual tersebut.
- Data yang dapat ditangani belum besar dan belum mendukung *widowing function*.

2.11 Penelitian yang Relevan

Berikut penelitian tentang jarak terpendek (*shortest path*) yang membahas algoritma *Dijkstra* dan algoritma *Steepest Ascent Hill Climbing* :

1. Hairul Anam, Feby Sabilhul Hanafi, Ahmad Fauzal Adifia (2018) dalam judul penelitian penerapan metode *Steepest Ascent Hill Climbing* pada permainan puzzle. Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa metode *Steepest Ascent Hill Climbing* bisa diterapkan dalam permainan puzzle dengan mempertimbangkan nilai *heuristik* dari langkah selanjutnya. Penyelesaian puzzle dengan *Steepest Ascent Hill Climbing* pada inisialisasi state awal yang diberikan karena hal itu akan berpengaruh pada nilai *heuristik* yang dimiliki. Oleh karena itu adakalanya puzzle mencapai posisi terbaik yang diinginkan bisa disebut *global optimum*. Namun terkadang akan mencapai *lokal optimum* yaitu ketika posisi puzzle tidak mencapai *goal state*.
2. Thiang, Handry Khoswanto, 2018, dalam judul penelitian Implementasi metode *Steepest Ascent Hill Climbing* pada mikrokontroler MCS51 untuk robot mobil pencari rute terpendek. Peneliti menunjukkan metode *Steepest Ascent Hill Climbing* telah berhasil dplementasi pada mikrokontroler AT89S52 dalam mencari rute terpendek. Kedua robot dapat bergerak sesuai dengan rute yang telah didapat dengan menggunakan metode *Steepest Ascent Hill Climbing* dan juga kedua robot dapat bekerja sama dan berkomunikasi dengan baik. Tetapi karena area yang dirancang untuk pengujian berbentuk simetris, maka seharusnya terdapat lebih dari satu solusi berupa rute terpendek. Pada kasus ini, metode *Steepest Ascent Hill Climbing* hanya dapat menemukan satu solusi saja. Metode *Steepest Ascent Hill Climbing* tidak dapat digunakan untuk menentukan rute yang terpendek bila titik yang dievaluasi mempunyai jarak yang sama. Untuk pengembangan selanjutnya, sebaiknya area untuk pengujian, jangan dibuat simetris sehingga bisa terlihat jelas keberhasilan metode *Steepest Ascent Hill Climbing* dalam pencarian rute terpendek.

3. Elvina, Lukman Hakim (2018), dalam judul penelitian modifikasi algoritma *Steepest Ascent Hill Climbing* dan *backtracking* untuk pencarian lintasan kritis proyek. Hasil dari penelitian ini menunjukkan bahwa angka keberhasilan dalam menangani 5 *test-case* adalah sebesar 80%. Hal ini disebabkan tidak adanya proses *backtracking* dari modifikasi algoritma *Steepest Ascent Hill Climbing* untuk membandingkan setiap kemungkinan lintasan kritis yang ada. Namun dengan tidak adanya proses *backtracking* pada modifikasi algoritma *Steepest Ascent Hill Climbing* menjadikan algoritma ini mempunyai *process cost* yang lebih rendah jika dibandingkan dengan algoritma yang tidak dimodifikasi.
4. Hendra Waskita, (2014) dalam judul penelitian penerapan algoritma *Steepest Ascent Hill Climbing* pada penjadwalan kelas praktikum. Hasil penelitian yaitu pada penelitian ini algoritma dapat diterapkan dengan baik dan dapat menghasilkan solusi sesuai dengan tujuan yang diharapkan atau mendekati solusi tujuan. Berdasarkan pengujian yang dilakukan, metode *Steepest Ascent Hill Climbing* lebih efisien sebesar 35,72% dari pada menggunakan metode random. Penjadwalan menggunakan sistem penjadwalan secara terkomputerisasi akan lebih mudah dan lebih cepat. Penjadwalan di simpan di *database* sehingga memudahkan dalam *backup* data. Data saling terintegrasi sehingga meminimalkan terjadinya seorang pengajar mengajar pada hari dan jam yang sama.
5. Gusri Alif Radityo (2016) dengan judul penelitian Rancang Bangun Aplikasi Informasi Rute Terpendek Rumah Sakit, Rumah Bersalin dan Puskesmas di Palembang dengan Algoritma *Steepest Ascent Hill Climbing* Berbasis Android. Dari data yang telah diujikan dapat disimpulkan bahwa penerapan algoritma ini efektif untuk pembuatan aplikasi informasi rute terpendek tetapi kurang efisien karena aplikasi ini dibutuhkan dalam keadaan darurat sehingga diperlukan waktu yang cepat untuk mendapatkan informasi ke lokasi tujuan.
6. Jemmy Sihombing (2014) dalam skripsi yang berjudul Perancangan Aplikasi Pencarian Jalur Terpendek. Hasil penelitian yang diperoleh

adalah aplikasi dapat menemukan jalur terpendek dengan cukup baik dengan menggunakan Metode *Steepest Ascent Hill Climbing*.

7. Endah Yuniawati (2015) dalam skripsi nya yang berjudul Optimasi Rute Distribusi Menggunakan *Steepest Ascent Hill Climbing*. Hasil penelitiannya yang diperoleh adalah dari rute sebelum dan sesudah dilakukan perhitungan dalam menggunakan *Steepest Ascent Hill Climbing* didapatkan jarak yang lebih optimal dengan selisih jarak 10,3 Km yaitu sekitar 20,12%. Setelah menggunakan *Steepest Ascent Hill Climbing* didapatkan jarak yang lebih optimal dengan selisih jarak 10,7 Km yaitu sekitar 16,54%. Sehingga dapat disimpulkan bahwa setiap harinya dapat menghemat 18,11% untuk biaya transportasi.
8. Ferdi Ninaber (2009) dalam judul penelitiannya yaitu Robot Mobil Pencarian Rute Terpendek Menggunakan Metode *Steepest Ascent Hill Climbing*. Hasil penelitiannya dapat disimpulkan bahwa untuk bergerak menuju tujuan, robot mobil akan bergerak mengikuti garis rute yang telah didapatkan dengan menggunakan metode *Steepest Ascent Hill Climbing*. Dari hasil pengujian terlihat bahwa robot dapat menentukan rute terpendek berdasarkan algoritma *Steepest Ascent Hill Climbing*. Akan tetapi karena map yang digunakan dalam pengujian adalah map yang simetris maka ada banyak solusi untuk rute terpendek. Metode ini hanya dapat menemukan salah satu dari rute tersebut sehingga belum dapat dikatakan sebagai rute terpendek. Dari hasil pengujian juga terlihat bahwa setelah mendapatkan rute terpendek, robot dapat bergerak sesuai dengan rute yang dihasilkan.
9. David Abraham (2015), dalam judul penelitiannya Penyelesaian aslah 8-Puzzle Dengan Algoritma *Steepest Ascent Hill Climbing*. Hasil penelitiannya bahwa solusi 8 pazzle cepat diperoleh dengan menggunakan prinsip array dan variasi algoritma *Steepest Ascent Hill Climbing* dengan memilih kemiringan yang paling tajam atau curam dengan parameter heuristic posisi yang benar dan heuristic jarak serta dikombinasikan dengan LogList sebagai penyimpanan state yang pernah dilalui untuk menanggulangi permasalahan pada algoritma itu sendiri.

10. Lipian Alfa (2014), dalam judul penelitiannya Penerapan Metode *Steepest Ascent Hill Climbing* dalam Model Pencarian Rute Terdekat Fasilitas Pelayanan Darurat di Kota Bogor Berbasis Android. Hasil penelitiannya bahwa pada penelitian ini dilakukan secara manual dengan menggunakan data-data yang sudah di dapatkan lalu di buatkan garis sesuai hasil perhitungan manual pada rute *google maps* dengan memasukkan titik-titik koordinat yang telah terpilih dan menjadi rute terdekat ke dalam coding program di android studio.

BAB I

METODOLOGI PENELITIAN

3.1 Tempat dan Waktu

3.1.1 Tempat Penelitian

Penelitian ini dilakukan di tempat objek wisata yang ada di Kabupaten Humbang Hasundutan diantaranya berupa : 1) Bakkara; 2) Istana Makam Sisingamangaraja; 3) Air Terjun Aek Sipangolu; 4) Sipinsur dan ; 5) Air Terjun Binanga Janji.

3.1.2 Waktu & Jadwal Pelaksanaan Penelitian

Adapun waktu dan jadwal pelaksanaan penelitian seperti tabel di bawah ini:

Keterangan	Juni				Juli			Oktober	November
	I		I	IV	I		I		
Studi Literatur									
Studi Observasi									
Pengumpulan Data									
Analisis Kebutuhan									
Perancangan									
Pembuatan									
Selesai									

Tabel 3.1 Waktu dan Jadwal Pelaksanaan Penelitian

3.2 Teknik Pengumpulan Data

3.2.1. Studi Literatur

Dalam penelitian ini teknik pengumpulan data yang dilakukan adalah mencari referensi teori yang relevan dengan kasus permasalahan yang didapatkan. Pada pencarian informasi dengan studi pustaka pada beberapa jurnal ilmiah dan skripsi yang berhubungan dengan sistem sistem pencarian rute terpendek menggunakan algoritma *Steepest Ascent Hill Climbing* dapat memperkuat permasalahan serta sebagai dasar teori dalam melakukan penelitian.

3.2.2. Observasi

Dalam penelitian ini penulis melakukan observasi lapangan langsung dengan melakukan perjalanan menggunakan kilometer kendaraan dan *Stopwatch* untuk mengetahui jarak dan waktu yang diperlukan untuk sampai di beberapa lokasi objek wisata Kabupaten Humbang Hasundutan serta menggunakan aplikasi *google maps* untuk mengetahui ke akuratan data jarak, waktu, rute serta nilai longitude dan Latitude di setiap titik lokasi objek wisata. Hasil observasi berupa data jarak, waktu dan rute menuju lokasi objek wisata Kabupaten Humbang Hasundutan yang diinginkan.

3.3 Cara Kerja

3.3.1 Sistem

Sistem yang dibangun bertujuan untuk mengimplementasikan algoritma *Steepest Acent Hill Climbing* dalam menentukan rute terpendek berbasis *mobile*. Pada penelitian ini perancangan dilakukan untuk mengetahui rute, jarak, dan waktu yang dibutuhkan untuk menempuh titik lokasi awal dan titik lokasi tujuan yang sudah ditentukan.

3.3.2 Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan dan diperlukan pada pembuatan aplikasi yang dapat di rangkai, adapun analisis kebutuhannya berupa perangkat keras,perangkat lunak,dan kebutuhan pengguna (user).

Analisis kebutuhan merupakan pernyataan tentang apa yang harus dikerjakan oleh sistem, dan karakteristik apa yang harus dimiliki sistem. Analisis kebutuhan sistem terbagi atas dua bagian, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

A. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang merupakan layanan Pada Sistem aplikasi yang harus disediakan, serta memasukan data yang sudah di dapat dalam observasi yang sudah dilakukan.

Dalam analisis kebutuhan fungsional disini menjelaskan tentang sistem yang disediakan. Sistem ini melakukan perhitungan, untuk menerapkan sistem tersebut, kebutuhan secara fungsional harus dipenuhi sebagai berikut adalah :

1. Menggunakan graf lokasi objek wisata di Kabupaten Humbang Hasundutan yang telah ditentukan.
2. User adalah orang yang menjalankan aplikasi tersebut agar aplikasinya dapat berjalan di dalam *smartphone* berbasis android.
3. Aplikasi menyediakan lokasi titik awal dan titik tujuan yang akan kita tuju.
4. hasil rute yang didapatkan akan dilalui oleh *graf* dengan algoritma *Steepest Ascent Hill Climbing*.

B. Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional digunakan dalam bentuk kebutuhan perangkat yang dibutuhkan sistem serta terbagi untuk pengembangan

atau penggunaannya. Kebutuhan non-fungsional dipenuhi antara lain sebagai berikut:

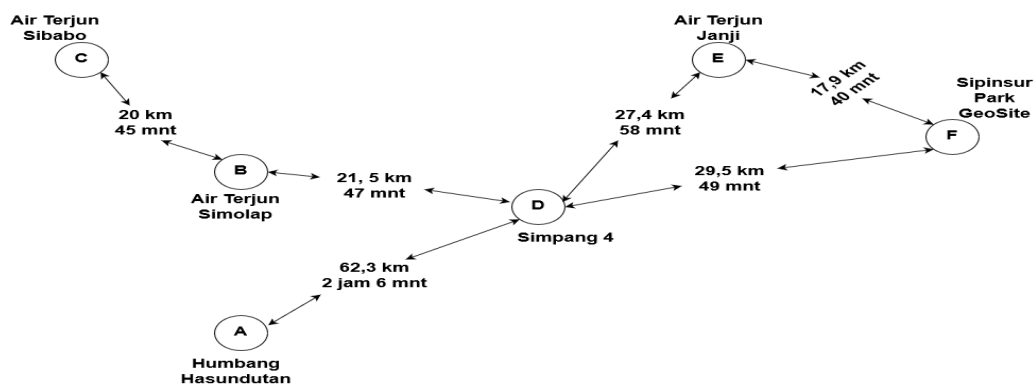
1. Alat

Aplikasi yang akan dibangun dapat dijalankan dengan *smartphone* berbasis android dan di instal kedalam *smartphone* tersebut.

2. Perangkat lunak

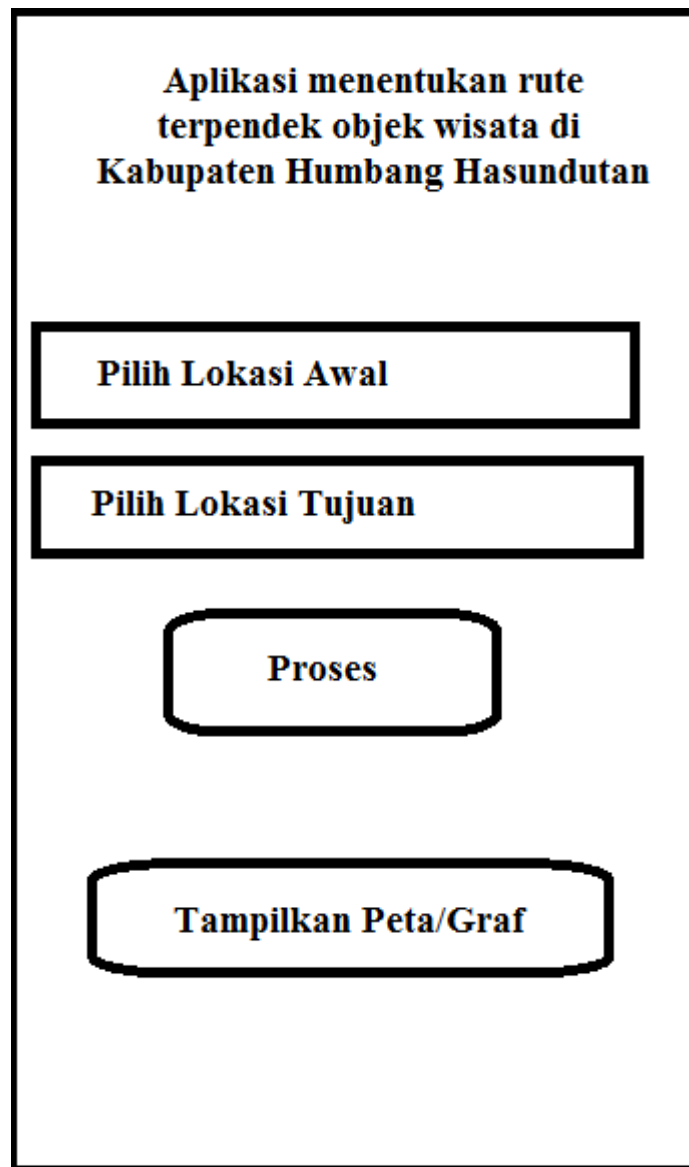
Aplikasi dibangun dengan menggunakan bahasa pemrograman bahasa java metode pemodelannya menggunakan UML (*Unified Modeling Language*) agar dapat mengembangkan perangkat lunak dan dijalankan disemua platform menggunakan *Android Studio* serta menggunakan *MySQL*, yang dinyatakan sebagai media penyimpanan utama untuk data aplikasi atau pengguna, atau juga bisa menggunakannya untuk proses *caching* serta menyediakan data yang diambil dari cloud.

Kemudian data yang sudah dikumpulkan dibuat kedalam Graf. Berikut adalah graf lokasi objek wisata yang berada di Kabupaten Humbang Hasundutan.



Gambar 3.1 Graf Rute Wisata di Kabupaten Humbang Hasundutan

3.3.3 Tampilan Rancangan Sistem



Gambar 3.2 Rancangan Tampilan Aplikasi

3.3.4 Pengujian

Tahap pengujian pencarian rute terpendek ini dilakukan dengan menggunakan algoritma *Steepest Ascent Hill Climbing* dengan cara melakukan uji coba langsung perjalanan dari satu objek wisata menuju objek wisata lain. Berapa jam, berapa kilometer yang harus ditempuh dan jalan mana saja yang akan dilalui pengunjung

menuju wisata tersebut. Pengujian ini dilakukan untuk memudahkan para wisatawan yang datang.

3.3.5 Penerapan/Penggunaan

Penerapan atau penggunaan dalam penelitian ini user dapat mengunduh dan menginstall aplikasinya ke dalam *smartphone* berbasis android kemudian user menentukan lokasi awal dan lokasi yang mau dituju kemudian muncul rute terpendek, jarak dan waktu tempuh di dalam aplikasi.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

4.1.1 Analisis Data

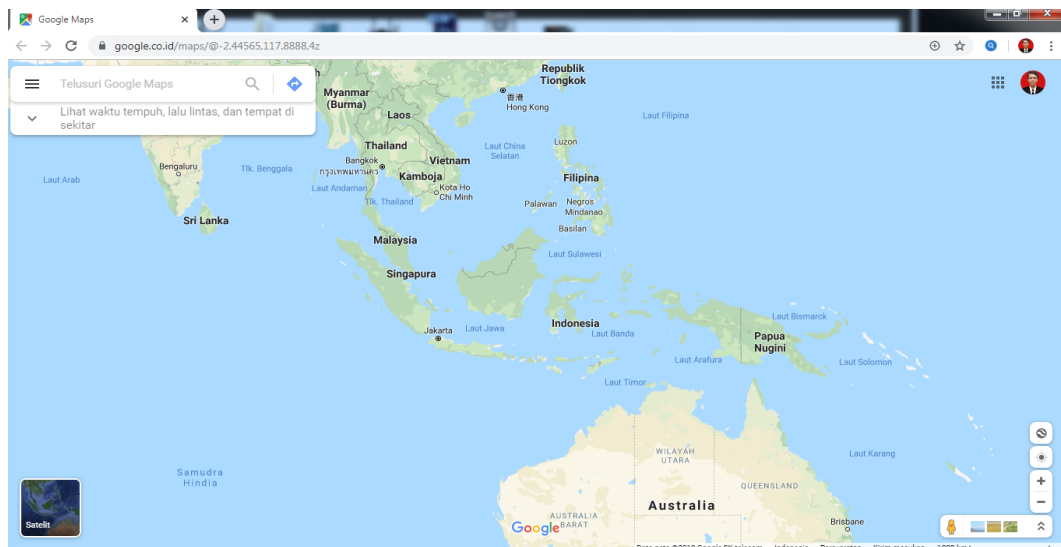
Analisis data dalam penelitian ini mengambil data berupa nilai Longitude dan Latitude pada *google maps* untuk menghitung dan mencari nilai heuristik menggunakan rumus :

$$H(n) = \sqrt{(X_{asal} - X_{tujuan})^2 + (Y_{asal} - Y_{tujuan})^2}$$

Hal ini bertujuan agar dapat mencocokkan data yang telah di dapatkan dalam beberapa literatur serta data-data lain yang telah disiapkan.

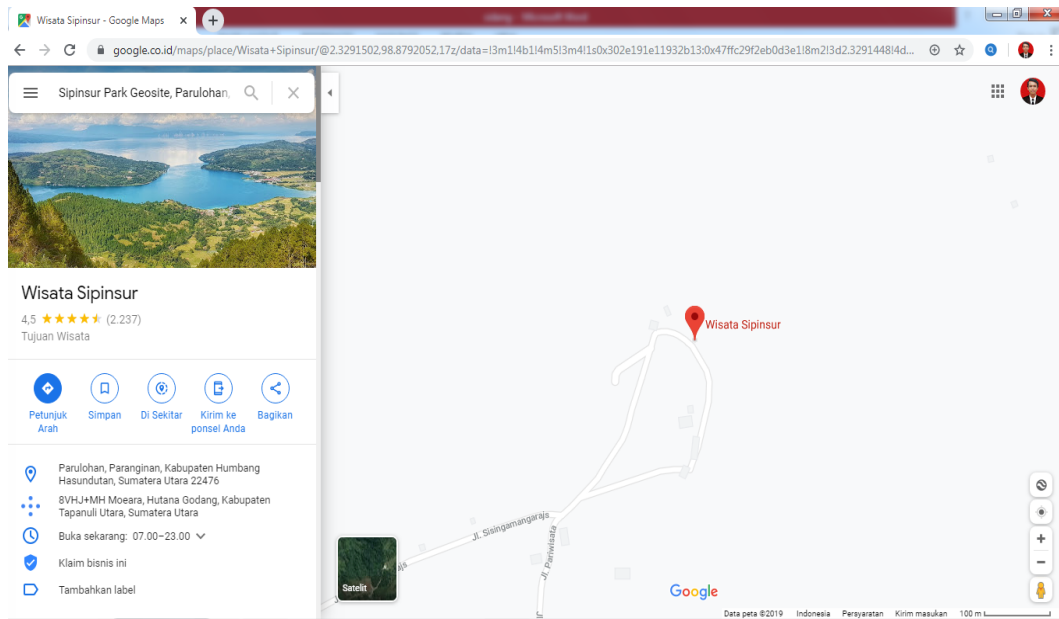
Berikut cara untuk mendapatkan nilai Latitude dan Longitude pada setiap titik Objek wisata Kabupaten Humbang Hasundutan menggunakan bantuan *GoogleMaps* :

1. Buka Web *GoogleMaps*



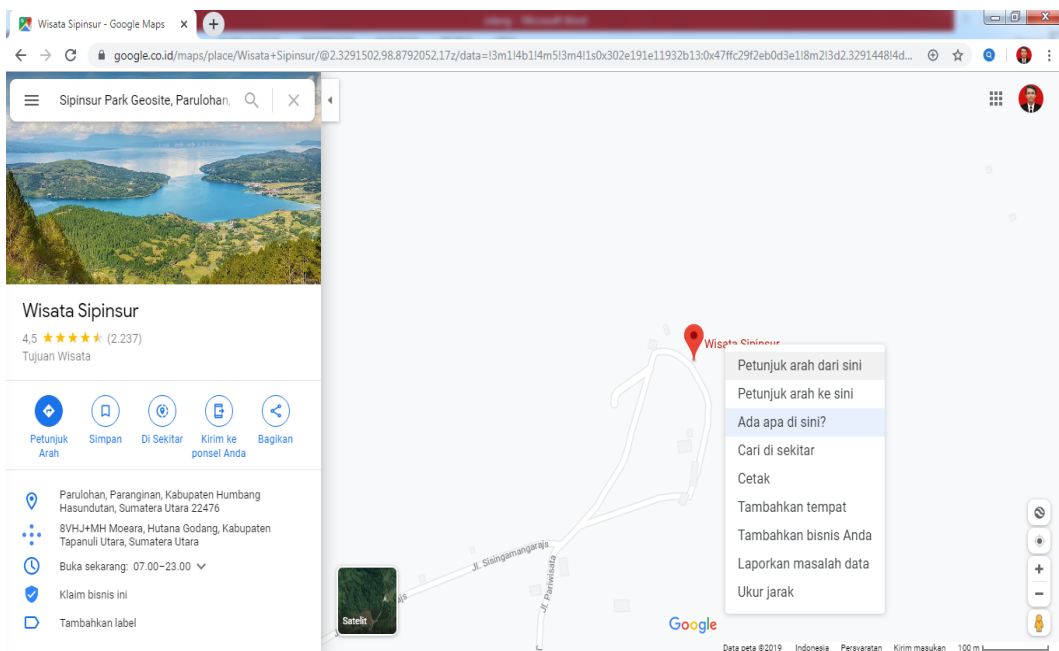
Gambar 4.1 Web *Google Maps* untuk mendapatkan nilai Latitude dan longitude

2. Input Objek Wisata



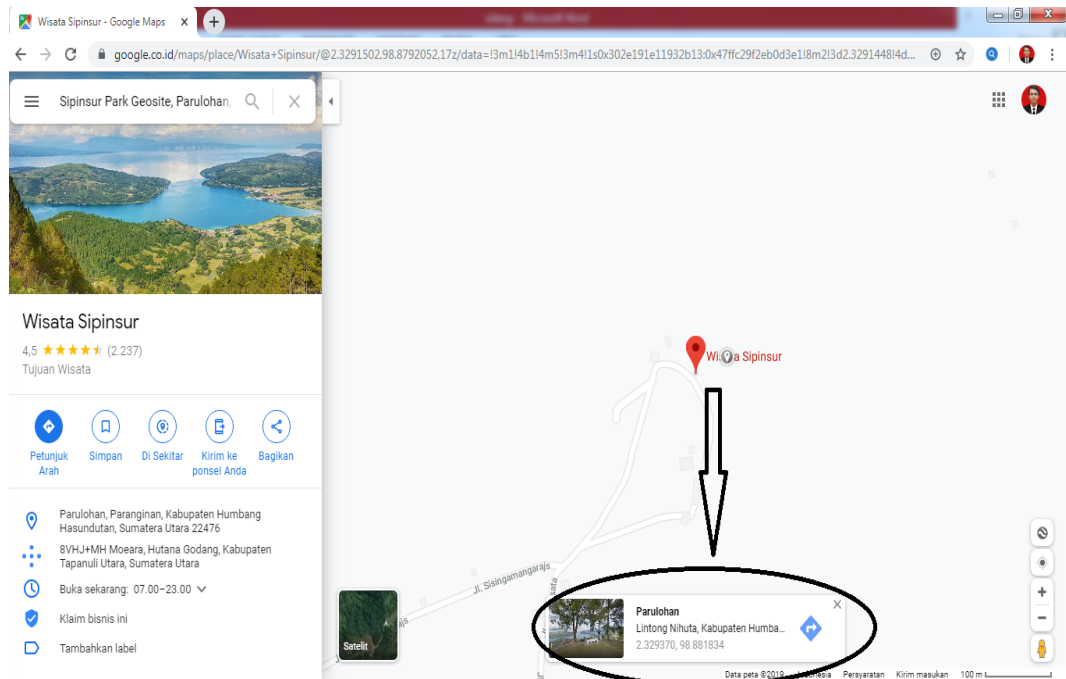
Gambar 4.2 Input Objek Wisata

3. Kemudian Klik kanan pada tanda merah Objek Wisata yang sudah dinput dan pilih bacaan “ada apa disini?”



Gambar 4.3 Objek Wisata yang Sudah Dinput

4. Kemudian akan muncul nilai Latitude dan Longitude pada Objek wisata tersebut.



Gambar 4.4 Nilai Latitude dan Longitude pada Objek Wisata

4.1.2 Hasil Analisis Data

Adapun hasil analisis data nilai Latitude dan Longitude serta hasil nilai heuristik pada penelitian ini adalah :

Lokasi Awal	Latitude (x)	Longitude (y)	Lokasi tujuan	Latitude (x)	Longitude (y)	Heuristik
Humbang	2.198849	98.572106	Sipinsur	2.329149	98.881393	0.3356136743
Simpang 4	2.256828	98.687397	Sipinsur	2.329149	98.881393	0.2070099884
Air terjun simolap	2.270737	98.518591	Sipinsur	2.329149	98.881393	0.3674741527
Air terjun janji	2.339474	98.815052	Sipinsur	2.329149	98.881393	0.0671396597
Air terjun sibabo	2.348579	98.426052	Sipinsur	2.329149	98.881393	0.4557553633
Simpang 4	2.256828	98.687397	Humbang	2.198849	98.572106	0.129048747
Air terjun simolap	2.270737	98.518591	Humbang	2.198849	98.572106	0.0896199738
Air terjun	2.339474	98.815052	Humbang	2.198849	98.572106	0.50673

janji						13538
Air terjun sibabo	2.348579	98.426052	Humbang	2.198849	98.572106	0.2091670237
Sipinsur	2.329149	98.881393	Humbang	2.198849	98.572106	0.5152080632
Humbang	2.198849	98.572106	Air terjun simolap	2.270737	98.518591	0.0896184808
Simpang 4	2.256828	98.687397	Air terjun simolap	2.270737	98.518591	0.1693780561
Sipinsur	2.329149	98.881393	Air terjun simolap	2.270737	98.518591	0.3674741527
Air terjun janji	2.339474	98.815052	Air terjun simolap	2.270737	98.518591	0.3043253189
Air terjun sibabo	2.348579	98.426052	Air terjun simolap	2.270737	98.518591	0.3823982675
Humbang	2.198849	98.572106	Air terjun janji	2.339474	98.815052	0.2807100809
Simpang 4	2.256828	98.687397	Air terjun janji	2.339474	98.815052	0.1520728783
Air terjun simolap	2.270737	98.518591	Air terjun janji	2.339474	98.815052	0.3043253189
Sipinsur	2.329149	98.881393	Air terjun janji	2.339474	98.815052	0.0663410001
Air terjun sibabo	2.348579	98.426052	Air terjun janji	2.339474	98.815052	0.38910654199
Humbang	2.198849	98.572106	Air terjun sibabo	2.348579	98.426052	0.2091670237
Simpang 4	2.256828	98.687397	Air terjun sibabo	2.348579	98.426052	0.2769755495
Air terjun simolap	2.270737	98.518591	Air terjun sibabo	2.348579	98.426052	0.0925389999
Air terjun janji	2.339474	98.815052	Air terjun sibabo	2.348579	98.426052	0.38910654199
Sipinsur	2.329149	98.881393	Air terjun sibabo	2.348579	98.426052	0.4557553633

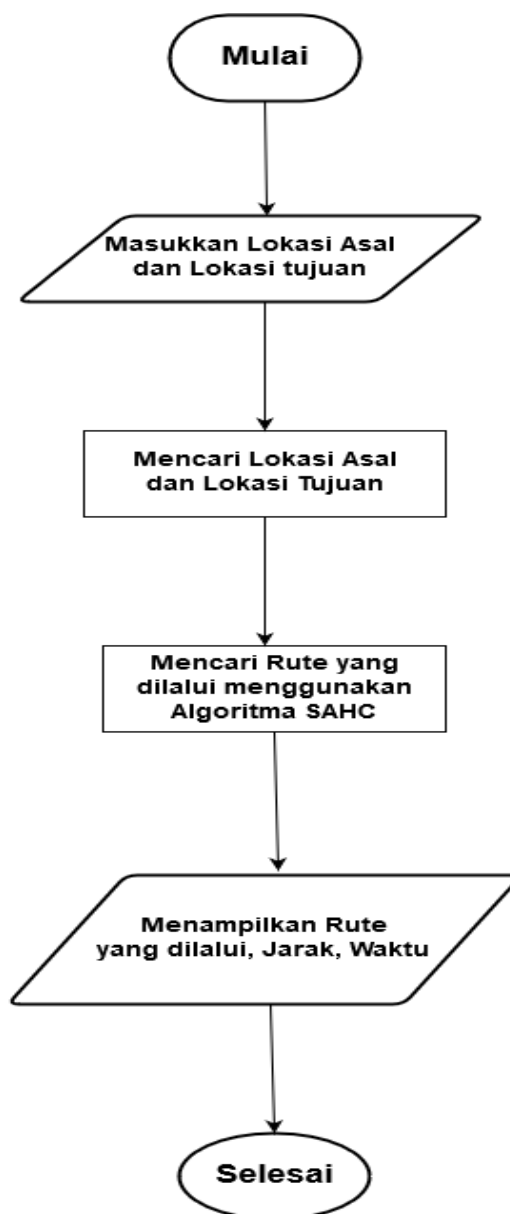
Tabel 4.1 Hasil Analisis Data

4.1.3 Perancangan

Pada penelitian ini untuk merancang sistem pencarian jarak terdekat setiap lokasi objek wisata di Kabupaten Humbang Hasundutan menggunakan *Flowchart* Sistem dan *Flowchart* pada Algoritma *Steepest Ascent Hill Climbing*.

4.1.3.1 *Flowchart* Sistem

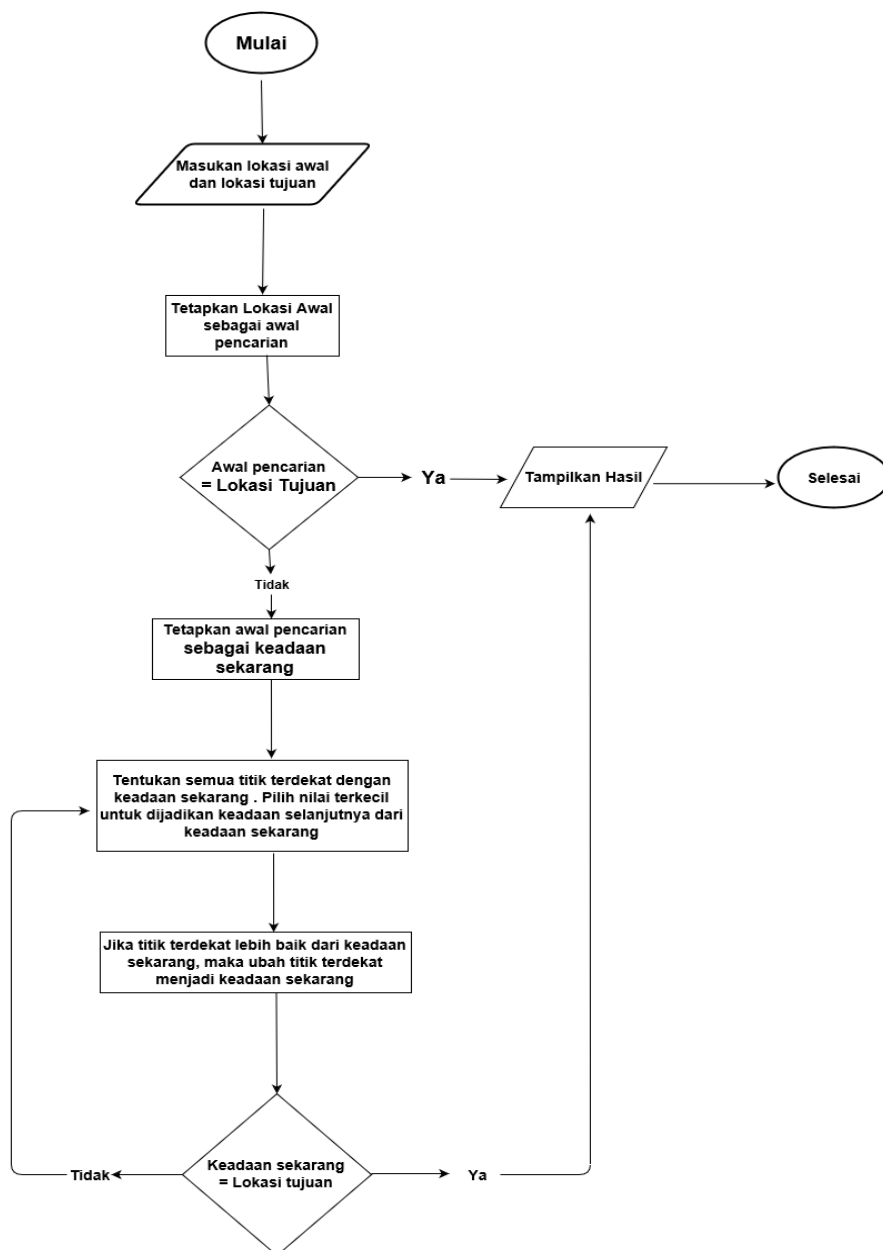
Flowchart ini untuk melakukan pencarian jarak terdekat antar lokasi di objek wisata Kabupaten Humbang Hasundutan yang dapat dilihat pada Gambar 4 Pada flowchart ini terdapat tiga masukan yaitu : lokasi asal objek wisata, lokasi tujuan objek wisata, dengan keluarannya yang akan menghasilkan total jarak, rute, dan waktu eksekusi.



Gambar 4.5 *Flowchart Sistem*

4.1.3.2 *Flowchart Algoritma Steepest Ascent Hill Climbing*

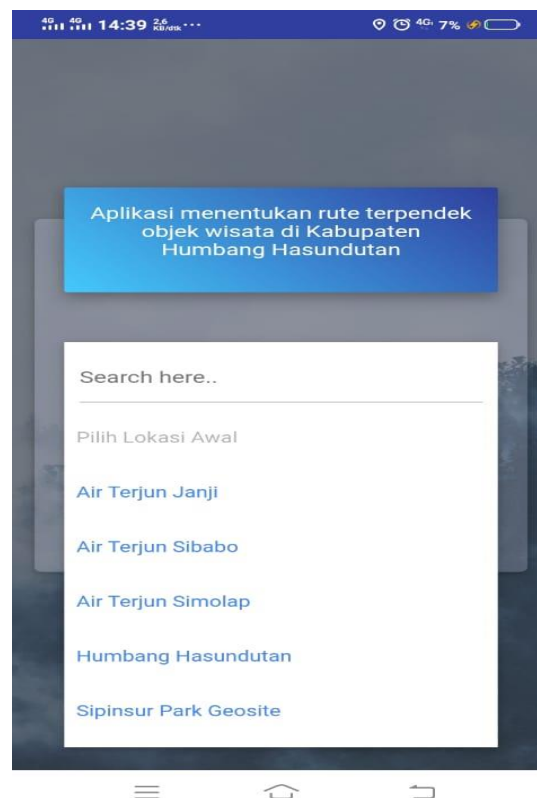
Flowchart ini untuk menggambarkan proses pencarian jarak terdekat disetiap lokasi objek wisata Kabupaten Humbang Hasundutan menggunakan algoritma *Steepest Ascent Hill Climbing* yang terdapat dalam Gambar 3.2.



Gambar 4.6 *Flowchart* Algoritma *Steepest Ascent Hill Climbing*

4.1.4 Pengujian

Pengujian adalah tahap lanjutan setelah dilakukannya analisis data. Pengujian dilakukan dengan tujuan membuktikan sistem yang dibuat telah berjalan dengan baik. Pengujian dilakukan terhadap algoritma *Steepest Ascent Hill Climbing* dalam menentukan rute terpendek berbasis *Mobile* dengan implementasi hasil jarak terpendek dan waktu proses pada algoritma. Titik asal dan titik tujuan terdapat sebanyak 5 titik yang disediakan. pada Gambar 4.6

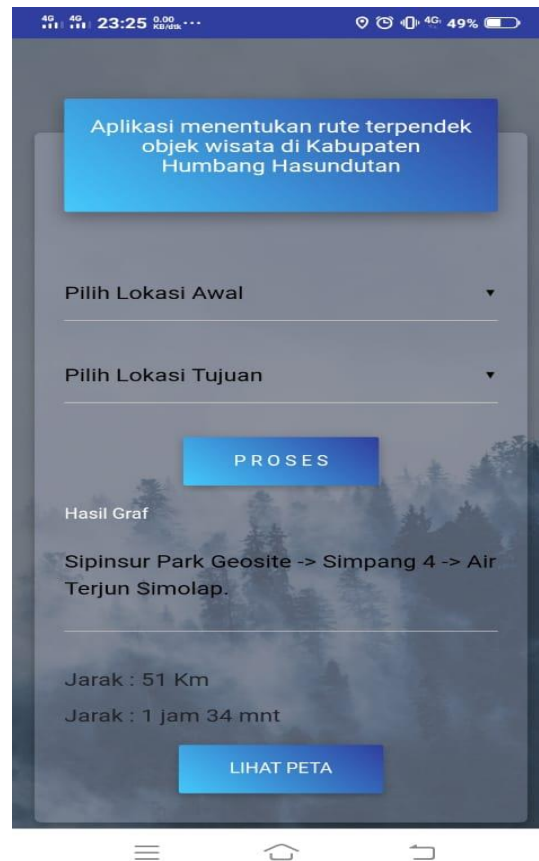


Gambar 4.7 Pengujian Proses Algoritma SAHC

Pada tahap pengujian ini user memilih Lokasi awal dan lokasi tujuan yang diinginkan kemudian tekan tombol proses untuk melakukan pencarian rute terpendek, total jarak, waktu dan graf/rute yang akan ditampilkan menggunakan Algoritma *Steepest Ascent Hill Climbing* ditunjukkan gambar 4.7. Berdasarkan gambar 4.8 diketahui bahwa dari Sipinsur Geosite (lokasi awal) ke Air Terjun Simolap (lokasi tujuan) hasil

rute terpendek yang dilintasi adalah Sipinsur Geosite → simpang 4 → Air terjun simolap, dengan total jarak 51 km dan waktu 1 jam 34 menit.

Gambar 4.8



Gambar 4.8 Hasil Rute Terpendek yang Dilintasi

Kode Titik	Nama Titik
A	Humbang Hasundutan
B	Air Terjun Simolap
C	Air Terjun Sibabo
D	Simpang 4
E	Air Terjun Janji
F	Sipinsur Geosite

Tabel 4.2 Nama Objek Wisata di Kabupaten Humbang Hasundutan

4.1.5 Perhitungan Manual Algoritma *Steepest Ascent Hill Climbing*

Dalam pengujian ini, akan dilihat hasil dari perhitungan manual algoritma *Steepest Ascent Hill Climbing*, apakah sesuai dengan Implementasi Sistem. Adapun perhitungan manual algoritma *Steepest Ascent Hill Climbing* sebagai berikut :

Tahap 1 :

Pengujian sistem yang akan dicari yaitu rute terdekat dengan hitungan manual dari *node* F menuju *node* B pada gambar 3.1.

Tahap 2 :

Mendapatkan nilai heuristik dengan menggunakan titik koordinat Longitude dan Latitude pada *Google Maps* lalu dihitung dengan menggunakan persamaan sebagaimana yang tertera dibawah ini :

$$H(n) = \sqrt{(X_{asal} - X_{tujuan})^2 + (Y_{asal} - Y_{tujuan})^2}$$

Berikut adalah hasil perhitungan untuk mendapatkan nilai heuristik $h(n)$ pada tabel (4.3) dibawah ini :

<i>Node</i> Asal	Latitude x	Longitude Y	<i>Node</i> tujuan	Latitude X	Longitude Y	Nilai Heuristik
Humbang	2.198849	98.572106	Air terjun simolap	2.270737	98.518591	0.089618480 8
Simpang tiga	2.256828	98.687397	Air terjun simolap	2.270737	98.518591	0.169378056 1
Sipinsur	2.329149	98.881393	Air terjun simolap	2.270737	98.518591	0.367474152 7
Air terjun janji	2.339474	98.815052	Air terjun simolap	2.270737	98.518591	0.304325318 9
Air terjun sibabo	2.348579	98.426052	Air terjun simolap	2.270737	98.518591	0.382398267 5

Tabel 4.3 Perhitungan Manual Algoritma *Steepest Ascent Hill Climbing*

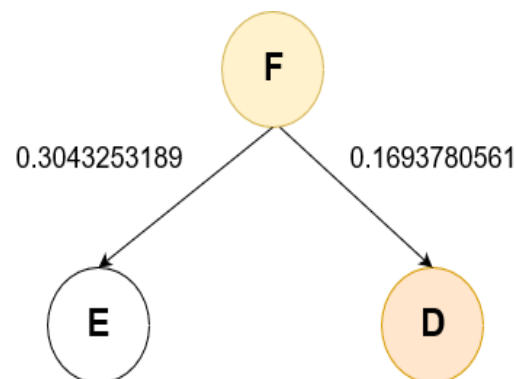
Tahap 3 :

Mulai dari *node* awal F. Pilih semua *node* yang terhubung dengan *node* awal F yaitu *node* E dan D yang ditunjukkan gambar 4.7 diketahui bahwa nilai heuristic masing – masing *node* adalah :

D : 0.1693780516

E : 0.3043253189

Lakukan pemilihan *node* yang bernilai heuristic terkecil yaitu *node* D. Maka $h(d) = 0.2769755495$ *node* D menjadi *current node* (lihat gambar 4.7)



Gambar 4.9 Node D Menjadi *Current Node*

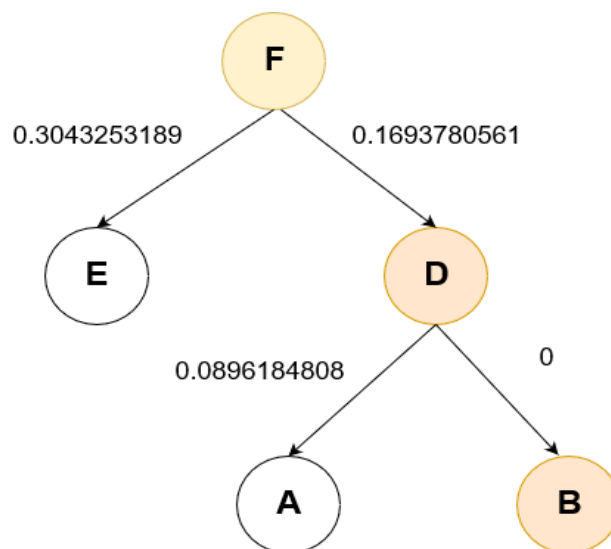
Tahap 4 :

Mulai dari *current node* = D . Pilih semua *node* yang terhubung dengan *node* D yaitu *node* A dan B, berdasarkan perhitungan di tabel (4.4) diketahui nilai heuristik masing – masing *node* adalah :

A : 0.2091670237

B : 0

Dikarenakan *current node* B nilai heuristik yang terdapat *node* B = 0 dan *node* B merupakan tujuan. Maka hasil proses pencarian berhenti lihat gambar (4.9) (lihat Gambar 4.9).



Gambar 4.10 Hasil Pencarian Berhenti

Berdasarkan hasil dari pengujian proses perhitungan manual menggunakan algoritma *Steepest Ascent Hill Climbing* yang di tunjukkan oleh Gambar 4.9 maka hasil rute terpendek dari Sipinsur Geosite (F) ke Air terjun simolap (B) adalah F → D → B (Sipinsur GeoSite – simpang 4 – Air terjun simolap) total jaraknya adalah 51 km dan waktu nya 1 jam 34 menit. Sehingga hasil pengujian proses algoritma *Steepest Ascent Hill Climbing* dengan sistem sesuai dengan proses perhitungan manual algoritma *Steepest Ascent Hill Climbing* (lihat Gambar 4.9 dan Tabel 4.3).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil Implementasi Algoritma *Steepest Ascent Hill Climbing* (SAHC) Untuk Aplikasi Pencarian Rute Terpendek berbasis *Mobile* di Kabupaten Humbang Hasundutan. Maka didapat kesimpulan bahwa pencarian rute terpendek berbasis *Mobile* dapat diselesaikan dengan menggunakan algoritma *Steepest Ascent Hill Climbing*. Pada proses perhitungan manual dengan menggunakan algoritma *Steepest Ascent Hill Climbing* pada node asal Humbang terdapat nilai heuristik sebesar 0.0896184808, pada node asal simpang tiga terdapat nilai heuristik 0.1693780561, pada node asal sipinsur terdapat nilai heuristik sebesar 0.367474152, pada node asal air terjun janji terdapat nilai heuristik sebesar 0.3043253189, dan yang terakhir pada node asal air terjun sibabo terdapat nilai heuristik sebesar 0.3823982675. Maka hasil rute terpendek dari Sipinsur Geosite (F) ke Air terjun simolap (B) adalah $F \rightarrow D \rightarrow B$ (Sipinsur GeoSite – simpang 4 – Air terjun simolap) total jaraknya adalah 51 km dan waktu nya 1 jam 34 menit. Sehingga hasil pengujian proses algoritma *Steepest Ascent Hill Climbing* dengan sistem sesuai dengan proses perhitungan manual algoritma *Steepest Ascent Hill Climbing*.

5.2 Saran

Adapun saran-saran yang dapat dipertimbangkan untuk penelitian maupun pengembangan sistem kedepannya, diantaranya adalah :

1. Aplikasi yang di ciptakan hanya digunakan pada Smartphone berbasis android, serta dapat dikembangkan di setiap OS yang berbeda.
2. Untuk penelitian selanjutnya diharapkan dapat mengimplementasikan algoritma *Steepest Ascent Hill Climbing* pada kasus yang berbeda. Serta dapat menambahkan simpul dan sisi agar jarak yang dihasilkan lebih varaitif.

3. Membuat tampilan aplikasi android yang menarik agar pengguna lebih mudah untuk menggunakan aplikasi menentuka rute terpendek.

DAFTAR PUSTAKA

- Aditya Rahmatullah, 2016, *Belajar UML – Use Case Diagram*, Jakarta : Rineka Cipta
- Agustin, 2011, *Algoritma Diagram Alir dan Pseudo-Code*, Jakarta : Rineka Cipta
- Amras Mauluddin, Hadi Prasetyo, dll, 2016, Implementasi Algoritma *Steepest Ascent Hill Climbing* Pada Permainan *Slide Puzzle* Berbasis Android, ISSN : 2089-9815
- Arifianto, Teguh. (2011). *Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT*, Yogyakarta: Andi Publisher
- Bambang Haryanto, 2011, *Esensi-Esensi Bahasa Pemrograman Java*, Yogyakarta : Andi
- Berlianty, I. & Arifin, M. 2010. *Teknik – Teknik Optimasi Heuristik*. Graha Ilmu: Yogyakarta.
- Budi Sutedjo, 2016, *Algoritma & Teknik Pemrograman*, Yogyakarta : Andi Offset
- Dublin, 2009, *Mencari Rute Terpendek*, Jakarta: Rineka Cipta
- Hasan Abdurrahman dan Asep Ririh Riswaya, 2014, *Penjelasan Mengenai Aplikasi*, Bandung
- Henderson, H, 2009, *Encyclopedia Of Computer Science and Teknologi*, Newyork : Facts on File
- Hermawan S, Stephanus, 2011, *Mudah Membuat Aplikasi Android*, Yogyakarta : Andi Offset.
- Jesa ariawan dan Sri Wahyuni, 2015, *Aplikasi Pengajuan Lembar Karyawan Berbasis Web*, Jurnal Sisfotek Global, Vol 5 No. 1
- Kusumadewi, 2005, *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*, Yogyakarta : Graha Ilmu
- Furqan, Mawengkang, Salim, 2018, *A Reviem of Prim And Genetical Algorithms in finding anddetermining routes on connected wegthed Graphs. International Journal of Civil Engineering and Technology (IJCIET)*, ISSN : 0976-6308, Vol 9.

- Mutakhirah, I., Saptono, F., Hasanah, N., Wiryadinata, Romi. 2007. *Pemanfaatan Metode Heuristik dalam Pencarian Jalur Terpendek dengan Algoritma*
- Ningati, 2014, *Sistematika informasi Geografis (SIG)*, Jakarta : Rineka Cipta
- Nazaruddin Safaat H. 2011. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Informatika*. Bandung
- Suarga, 2012, *Algoritma dan Pemrograman*, Yogyakarta : C.V Andi Ovset
- Thiang, 2009, *Robot Pencarian Rute Terpendek Menggunakan Metode Steepest Ascent Hill Climbing*, Yogyakarta, ISSN : 1907-5022
- Wisnu, Adam, Risa, 2015, *Pembuatan Game Slider Puzzle Menggunakan Metode Steepest Ascent Hill Climbing berbasis Android*, ISSN : 1979-8911, Vol. IX No 1
- Zhan F. Benjamin, 1997, *There Fastest Shortest Path lgorithms on Real Procedures. International Journal of Geographic and Decision Analysis*

LAMPIRAN-LAMPIRAN

Lampiran 1 Listing Program

1. Listing Program Pengguna

```
1 <?php
2 include "koneksi.php";
3 $awal = $_GET['titikawal'];
4 $akhir = $_GET['titikakhir'];
5 ?>
6 <!DOCTYPE html>
7 <html class="full-height">
8 <head>
9 <title>GIS</title>
10 <!-- Font Awesome -->
11 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
12 <!-- Bootstrap core CSS -->
13 <link rel="stylesheet" href="css/bootstrap.min.css">
14 <!-- Material Design Bootstrap -->
15 <link rel="stylesheet" href="css/mdb.css">
16
17 <!-- Your custom styles (optional) -->
18 <style>
19     .intro-2 {
20         background: url("img/img1.jpg")no-repeat center center;
21         background-size: cover;
22     }
23     .top-nav-collapse {
24         background-color: #3f51b5 !important;
25     }
26     .navbar:not(.top-nav-collapse) {
27         background: transparent !important;
28     }
29     @media (max-width: 768px) {
30         .navbar:not(.top-nav-collapse) {
31             background: #3f51b5 !important;
32         }
33     }
34
35     .card {
```

```

36     background-color: rgba(229, 228, 255, 0.2);
37   }
38
39   .md-form .prefix {
40     font-size: 1.5rem;
41     margin-top: 1rem;
42   }
43   .md-form label {
44     color: #ffffff;
45   }
46   h6 {
47     line-height: 1.7;
48   }
49   @media (max-width: 740px) {
50     .full-height,
51     .full-height body,
52     .full-height header,
53     .full-height header .view {
54       height: 750px;
55     }
56   }
57   @media (min-width: 741px) and (max-height: 638px) {
58     .full-height,
59     .full-height body,
60     .full-height header,
61     .full-height header .view {
62       height: 750px;
63     }
64   }
65
66   .card {
67     margin-top: 30px;
68     /*margin-bottom: -45px;*/
69

```

```

70   }
71
72   .md-form input[type=text]:focus:not([readonly]),
73   .md-form input[type=password]:focus:not([readonly]) {
74     border-bottom: 1px solid #8EDEF8;
75     box-shadow: 0 1px 0 0 #8EDEF8;
76   }
77   .md-form input[type=text]:focus:not([readonly])+label,
78   .md-form input[type=password]:focus:not([readonly])+label {
79     color: #8EDEF8;
80   }
81
82   .md-form .form-control {
83     color: #fff;
84   }
85
86 </style>
87
88 </head>
89
90
91 <body class="fixed-gn white-skin">
92
93   <!--Main Navigation-->
94   <header>
95     <!--Intro Section-->
96     <section class="view intro-2 htm-stylish-strong">
97       <div class="full-hg-img flex-center">
98         <div class="container">
99           <div class="row">
100             <div class="col-xl-5 col-lg-6 col-md-10 col-sm-12 mx-auto mt-lg-5">
101               <form action="" method="GET">
102                 <!--Form with header-->
103                 <div class="card wow fadeIn" data-wow-delay="0.6s">
104                   <div class="card-body">

```

```

106 <!--Header-->
107 <div class="form-header blue-gradient wow bounceInLeft" data-wow-delay="0.9s">
108   <h3>Silahkan Pilih Titik</h3>
109 </div>
110
111 <!--Body-->
112 <div class="md-form wow fadeOut" data-wow-delay="0.7s">
113   <select class="mdb-select" name="titikawal" searchable="Search here..">
114     <option value="" disabled selected>Pilih Titik Awal</option>
115     <option>Humbang Hasundutan</option>
116   </select>
117 </div>
118 <div class="md-form wow fadeOut" data-wow-delay="0.7s">
119   <select class="mdb-select" name="titikakhir" searchable="Search here..">
120     <option value="" disabled selected>Pilih Titik Akhir</option>
121   <?php
122     $stmt = $conn->prepare("SELECT * from lokasi");
123     $stmt->execute();
124     $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
125     foreach($stmt->fetchAll() as $k=>$dt) {
126     }
127     <option><?php echo $dt['nm1']; <?></option>
128   <?php <?>
129   </select>
130 </div>
131
132 <div class="text-center wow bounceInRight" data-wow-delay="0.9s">
133   <input type="submit" name="masuk" class="btn blue-gradient waves-effect waves-light" value="P R O S E S">
134 </div>
135 <?php
136 if($awal != "" && $akhir != ""){
137   $stmt = $conn->prepare("SELECT * from lokasi where nm = '$awal' and nm1 = '$akhir'");
138   $stmt->execute();
139   $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
140   foreach($stmt->fetchAll() as $k=>$v) {
141     <?>
142
143     <br>
144     <center></center>
145     <br>
146     <div class="text-center wow bounceInRight" data-wow-delay="0.9s">
147       <a href="<?php echo $v['alamat']; <?>" target="_blank" class="btn blue-gradient waves-effect waves-light">Lihat Peta</a>
148     </div>
149
150     <?php
151     }
152   }
153   <?>
154 </div>
155 </div>
156 <!--/Form with header-->
157 </form>
158 </div>
159 </div>
160 </div>
161 </div>
162 </section>
163
164 </header>
165 <!--Main Navigation-->
166
167 <!-- SCRIPTS -->
168 <!-- JQuery -->
169 <script src="js/jquery-3.2.1.min.js"></script>
170 <!-- Bootstrap tooltips -->
171 <script type="text/javascript" src="js/popper.min.js"></script>
172 <!-- Bootstrap core JavaScript -->
173 <script type="text/javascript" src="js/bootstrap.js"></script>
174 <!-- MDB core JavaScript -->
175 <script type="text/javascript" src="js/mdb.min.js"></script>

```

```

175 <script type="text/javascript" src="js/mdb.min.js"></script>
176 <script>
177     new WOW().init();
178
179     // Material Select Initialization
180     $(document).ready(function () {
181         $('#select[name="datatables_length"]').material_select();
182     });
183     // Material Select Initialization
184     $(document).ready(function () {
185         $('#.mdb-select').material_select();
186     });
187 </script>
188 </body>
189 </html>

```

2. Listing Map

```

1 <?php
2 @session_start();
3 include "koneksi.php";
4 if(@$_SESSION['nama']){
5 $var = @$_SESSION['nama'];
6 $id = @$_SESSION['id'];
7 }?>
8 <!DOCTYPE html>
9 <html lang="en">
10
11 <head>
12 <meta charset="utf-8">
13 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
14 <meta http-equiv="x-ua-compatible" content="ie=edge">
15 <title>GIS</title>
16 <!-- Font Awesome -->
17 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
18 <!-- Bootstrap core CSS -->
19 <link rel="stylesheet" href="css/bootstrap.min.css">
20 <!-- Material Design Bootstrap -->
21 <link rel="stylesheet" href="css/mdb.css">
22 <!-- DataTables.net -->
23 <link rel="stylesheet" type="text/css" href="js/vendor/datatables/css/dataTables.bootstrap4.min.css"/>
24
25 <!-- Your custom styles (optional) -->
26 <style>
27
28 </style>
29 </head>
30
31 <body class="fixed-gn white-skin">
32
33 <!--Main Navigation-->
34 <header>

```



```

103 </div>
104 <div class="text-center">
105 <div class="img-form form-sm">
106 <input type="text" name="longal" class="form-control" required="">
107 <label for="form1" class="">Longitude Akhir</label>
108 </div>
109 </div>
110 <div class="text-center">
111 <div class="file-field">
112 <div class="btn btn-primary btn-sm waves-effect waves-light">
113 <span>Pilih Foto</span>
114 <input type="file" name="foto1">
115 </div>
116 <div class="file-path-wrapper">
117 <input class="file-path validate" type="text" placeholder="Foto Peta">
118 </div>
119 </div>
120 </div>
121 </div>
122
123 <!--Footer-->
124 <div class="modal-footer">
125 <a type="button" class="btn btn-outline-secondary-modal waves-effect" data-dismiss="modal">Kembali</a>
126 <input type="submit" name="simpan" class="btn btn-primary-modal" value="Simpan">
127 </div>
128 <?php
129 if(isset($_POST['simpan'])) {
130
131     $aa = $_POST['nm'];
132     $bb = $_POST['nml'];
133     $cc = $_POST['alamat'];
134     $dd = $_POST['lat'];
135     $ee = $_POST['longa'];
136     $ff = $_POST['lat1'];
137     $gg = $_POST['longal'];
138
139     $ab = pow($dd - $ff,2);
140     $ac = pow($ee - $gg,2);
141     $heuristik = sqrt($ab + $ac);
142
143     $nama_baru = $_FILES["foto1"]["name"];
144
145     move_uploaded_file($_FILES["foto1"]["tmp_name"], "img/".$_FILES["foto1"]["name"]);
146
147     $stmt = $conn->prepare('INSERT INTO lokasi (nm, nml, alamat, latitude, longitude, latitude1, longitude1, gambar, heuristik)
148 VALUES (:aa, :bb, :cc, :dd, :ee, :ff, :gg, :hh, :ii)');
149 $stmt->bindParam(':aa', $aa);
150 $stmt->bindParam(':bb', $bb);
151 $stmt->bindParam(':cc', $cc);
152 $stmt->bindParam(':dd', $dd);
153 $stmt->bindParam(':ee', $ee);
154 $stmt->bindParam(':ff', $ff);
155 $stmt->bindParam(':gg', $gg);
156 $stmt->bindParam(':hh', $nama_baru);
157 $stmt->bindParam(':ii', $heuristik);
158
159 $stmt->execute();
160 ?>
161 <script type="text/javascript">
162 window.location.href="map";
163 </script>
164 <?php
165 }
166 ?>
167 </form>
168 </div>
169 <!--/.Content-->
170 </div>
171

```



```

171 </div>
172 <!--/.Tutup Modal Tambah-->
173
174 <div class="card">
175     <div class="card-body">
176         <div style="overflow-x:scroll; padding: 3px; scroll-color:hidden; border: 0px solid #ccc;">
177             <table id="datatables" class="table table-striped table-bordered" cellspacing="0" width="100%">
178                 <thead>
179                     <tr>
180                         <th>No</th>
181                         <th>lokasi Awal</th>
182                         <th>lokasi Tujuan</th>
183                         <th>Alamat</th>
184                         <th>Latitude (X)</th>
185                         <th>Longitude (Y)</th>
186                         <th>Latitude (X2)</th>
187                         <th>Longitude (Y2)</th>
188                         <th>Heuristik</th>
189                         <th>Aksi</th>
190                     </tr>
191                 </thead>
192                 <tbody>
193                     <?php
194                     $stmt = $conn->prepare("SELECT * from lokasi");
195                     $stmt->execute();
196                     $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
197                     $no = 1;
198                     foreach($stmt->fetchAll() as $k=>$v) {
199                         ?>
200                     <tr>
201                         <td>?<?php echo $no++; ?></td>
202                         <td>?<?php echo $v['nm']; ?></td>
203                         <td>?<?php echo $v['nm1']; ?></td>
204                         <td>

```

```

205         <a href="?<?php echo $v['alamat']; ?>" target="_blank" class="btn btn-sm bg-success">
206         <i class="fa fa-map"></i>
207         </a>
208         </td>
209         <td>?<?php echo $v['latitude']; ?></td>
210         <td>?<?php echo $v['longitude']; ?></td>
211         <td>?<?php echo $v['latitude1']; ?></td>
212         <td>?<?php echo $v['longitude1']; ?></td>
213         <td>?<?php echo $v['heuristik']; ?></td>
214         <td>
215         <!--
216         <a href="#" class="btn btn-sm bg-warning" data-toggle="modal" data-target="#edit?<?php echo $v['id']; ?>">
217         <i class="fa fa-edit"></i>
218         </a>
219         -->
220         <a href="#" class="btn btn-sm bg-red" data-toggle="modal" data-target="#hapus?<?php echo $v['id']; ?>">
221         <i class="fa fa-trash"></i>
222         </a>
223         </td>
224     </tr>
225
226 <!-- Modal Ubah -->
227 <div class="modal fade" id="edit?<?php echo $v['id']; ?>" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
228     <div class="modal-dialog modal-lg modal-notify modal-warning" role="document">
229         <!--Content-->
230         <div class="modal-content">
231             <!--Header-->
232             <form action="editmap" method="GET">
233             <div class="modal-header">
234                 <p class="heading lead">Ubah Data</p>
235
236                 <button type="button" class="close animated rotateIn" data-dismiss="modal" aria-label="Close">
237                 <span aria-hidden="true" class="white-text">&times;</span>
238                 </button>

```

```

239     </div>
240
241     <!--Body-->
242     <?php
243     $kd = $v['id'];
244     $stmt = $conn->prepare("SELECT * from lokasi where id='$kd'");
245     $stmt->execute();
246     $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
247     foreach($stmt->fetchAll() as $k=>$s) {
248     ?>
249     <div class="modal-body">
250     <div class="text-center">
251     <div class="md-form form-sm">
252     <input type="text" value="<?php echo $s['nm']; ?>" name="nm" class="form-control" required="">
253     <label for="form1" class="">Nama Lokasi</label>
254     </div>
255
256     </div>
257     <div class="text-center">
258     <div class="md-form form-sm">
259     <input type="text" value="<?php echo $s['alamat']; ?>" name="alamat" class="form-control" required="">
260     <label for="form1" class="">Alamat</label>
261     </div>
262
263     </div>
264     <div class="text-center">
265     <div class="md-form form-sm">
266     <input type="text" value="<?php echo $s['latitude']; ?>" name="lat" class="form-control" required="">
267     <label for="form1" class="">Latitude</label>
268     </div>
269
270     </div>
271     <div class="text-center">
272     <div class="md-form form-sm">
273     <input type="text" value="<?php echo $s['longitude']; ?>" name="longg" class="form-control" required="">
274     <label for="form1" class="">Longitude</label>

```

```

273     </div>
274     </div>
275     </div>
276     <input type="hidden" name="id" value="<?php echo $s['id']; ?>">
277     <?php } ?>
278     <!--Footer-->
279     <div class="modal-footer">
280     <a type="button" class="btn btn-outline-secondary-modal waves-effect" data-dismiss="modal">Kembali</a>
281     <button type="submit" name="ubah" class="btn btn-primary-modal">Ubah</button>
282     </div>
283     </form>
284     </div>
285     <!--/.Content-->
286 </div>
287 </div>
288 <!--/Tutup Modal Ubah-->
289 <!-- Modal Hapus -->
290 <div class="modal fade" id="hapus<?php echo $v['id']; ?>" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
291 <div class="modal-dialog modal-lg modal-notify modal-danger" role="document">
292 <!--Content-->
293 <div class="modal-content">
294 <!--Header-->
295 <div class="modal-header">
296 <p class="heading lead">Hapus Data</p>
297
298 <button type="button" class="close animated rotateIn" data-dismiss="modal" aria-label="Close">
299 <span aria-hidden="true" class="white-text">&times;</span>
300 </button>
301 </div>
302
303 <!--Body-->
304 <form action="hapusmap" method="GET">
305 <?php
306 $kd = $v['id'];

```

```

307     ?>
308     <div class="modal-body">
309         <div class="text-center">
310             <div class="md-form form-gm">
311                 <input type="hidden" value="<?php echo $kd; ?>" name="kd">
312                 <p class="fa fa-warning animated flash" style="font-size: 170px;"></p>
313                 <p style="font-size: 20px;">Apakah anda yakin ingin menghapus data ini?</p>
314             </div>
315         </div>
316     </div>
317     <!--Footer-->
318     <div class="modal-footer">
319         <a type="button" class="btn btn-outline-secondary-modal waves-effect" data-dismiss="modal">Kembali</a>
320         <button type="submit" name="hapus" class="btn btn-primary-modal">Hapus</button>
321     </div>
322 </form>
323 </div>
324 <!--/.Content-->
325 </div>
326 </div>
327 <!--/.Tutup Modal Ubah-->
328 <?php } ?>
329 </tbody>
330 </table>
331 </div>
332 </div>
333 <div>
334
335
336
337 n: Basic examples-->
338
339
340 t-->

```

```

344 ->
345 >
346 </script>
347 </script>
348 </script>
349 </script>
350 </script>
351 </script>
352 </script>
353 </script>
354 </script>
355 </script>
356 </script>
357
358 Initialization
359 collapse").sideNav();
360
361 ner = document.getElementById('slide-out');
362 size(container, {
363     speed: 2,
364     propagation: true,
365     scrollbarLength: 20
366
367
368 }).ready(function() {
369     $('#datatables').DataTable({
370         'aging' : true,
371         'lengthChange': false,
372         'searching' : true,
373         'ordering' : true,
374         'info' : false,
375         'autoWidth' : true
376
377
378

```

```
376
377
378
379
380 1 Select Initialization
381  .ready(function () {
382     ect[name="datatables_length"]').material_select();
383
384 1 Select Initialization
385  .ready(function () {
386     b-select').material_select();
387
388
389
390
391  text/javascript">
392  nis() {
393     , document.getElementById("jenis").value;
394     etElementById("coba").value=tes2;
395
396
397
398
399
400 
401
402
403  text/javascript">
404     n.href="/erwin/logout";
405
406 
407
408
```

Lampiran 2

Kartu Bimbingan

KARTU BIMBINGAN SKRIPSI
Semester Gasal/Genap Tahun Akademik

Nama : Rizka Nur Wijaya	Pembimbing I : Dr. Mhd Fuzqan, S.Si., M. Comp. Sc
NIM : 7115302	Pembimbing II : ARMANSYAH, N.Kom.
Prog. Studi : Ilmu Komput	SK Pembimbing :
Judul Skripsi : Implementasi Algoritma Steepest Ascent Hill Climbing (SAHC) Untuk Aplikasi Pencarian Rute Terpendek Berbasis Mobile	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	08/05-2019	Perbaiki tulisan bacaan dan halaman		20/04/19	Mencari jurnal yang sesuai dengan penelitian	
II	21/05-2019	Tambahkan surral dan penelitian yang sudah dilakukan		05/15/19	Revisi cover dan Bab I	
III	16/07-2019	Revisi Bab 2,3		15/16/19	Revisi Penulisan dan tanda baca Halaman dan lain	
IV	17/08-2019	Tambahkan Jurnal di Daftar Pustaka		22/16/19	Buat sebuah graf rute terpendek	
V	23/18/19	Acc Sempro		29/18/19	Acc sempro	

Buku Laporan Kegiatan Akademik Mahasiswa Fakultas SAINTEK UIN-SU Medan | 28

VI	20/9/19	Revisi BAB IV - Analisis data - Program algoritma	Furqan	26/9/19	Revisi Plancher's Algoritma	
VII	13/10/19	Revisi sub-bab di bab IV	Furqan	17/10/19	hitung nilai heuristik disetiap etik.	
VIII	29/11/19	Revisi Kesimpulan	Furqan	30/10/19	Revisi kebutuhan Fungsional dan non Fungsional	
IX	5/11/19	Perbaiki Abstrak	Furqan	7/11/19	ACC sidang.	
X	9/11/2019	ACC sidang	Furqan			

Medan, 09 NOV 2019.

An. Dekan

Ketua Jurusan/Program Studi

ILMU KOMPUTER

Dr. Mhd Furqan, S.Si., M.Comp.Sc.

NIP. 190000620060411003

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

Lampiran ke 3

Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP CURRICULUM VITAE

DATA PRIBADI:

Nama : Razzaq H nur Wijaya.,S.Kom
Tempat/ TglLahir : Helvetia / 26 Mei 1997
JenisKelamin : Laki-laki
Agama : Islam
Status : Menikah
Tinggi Badan : 174
Alamat : Dusun VIII Jl.Bambu Gg.Kenanga
Hp : 081360174766
e-mail : razzaqnurwijaya@gmail.com



RIWAYAT PENDIDIKAN :

Tahun	Lembaga Pendidikan
2015 -2019	Universitas Islam Negeri Sumatera Utara Fak.Sains & Teknologi Jurusan Ilmu Komputer
2012 -2015	SMK NEGERI 1 Percut Sei Tuan
2009 -2012	MTs Al-Washliyah Tembung
2003 -2009	SD Negeri101602 Helvetia

PENDIDIKAN TERAKHIR :

Instansi	UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
Gelar	S.Kom (Sarjana Ilmu Kompute)
Program Studi	Ilmu Komputer
Judul Skripsi	Implementasi Algoritma <i>Steepest Ascent Hill Climbing</i> (SAHC) pada aplikasi pencarian rute terpendek berbasis <i>Mobile</i>
Tanggal Kelulusan	11 November 2019
IPK	3.3 (Skala 4)

PENGALAMAN ORGANISASI/KEPANITIAAN :

Tahun	Jenis Kegiatan dan Peran
2016-2017	Ketua Bidang Organisasi dan Pemberdayaan di Himpunan Mahasiswa Jurusan (HMJ)
2016	Ketua Panitia PORSENI ILKOMP seluruh Mahasiswa Kota Medan
2016	Panitia Penggalangan Dana untuk korban Sinabung

LAIN-LAIN :

Computer Skill	Mampu mengoperasikan Sistem Operasi Ubuntu, Mikrotik, Linux, Windows Ms.Office (Word, Excel, Power Point, dll), Mengetahui Bahasa Pemrograman Php, Internet dan software lainnya
Language Skill	Mampu menggunakan Bahasa Indonesia dengan baik dan menggunakan Bahasa Inggris untuk lisan dan tulisan secara pasif
Social Skill	Saya seorang yang kreatif, jujur, suka bekerja keras, bertanggung jawab, teliti, dan disiplin. Saya mampu bekerjasama dengan tim/kelompok dan berkomunikasi yang baik. Saya dapat belajar hal baru dengan baik dan mampu menyesuaikan diri dengan lingkungan baru
Skill Other	Mampu memperbaiki segala kerusakan pada Laptop/Pc, Maintenance Jaringan Internet dan CCTV

PENGALAMAN KERJA

Tahun	PERUSAHAAN/INSTANSI
2016-2017	Sebagai Tata Usaha di Sekolah SD Negeri 106812 Bandar Klippa <ul style="list-style-type: none"> - Menginput data DAPODIK Sekolah - Mengerjakan laporan dana BOS Sekolah
2019	Sebagai PPNP (Operator Komputer) di instansi Negeri Kementerian ATR/BPN Kabupaten Humbang Hasundutan Prov. Sumatera Utara <ul style="list-style-type: none"> - Mendata yang ingin mendaftarkan sertifikat tanah - Memeriksa kelengkapan berkas - Melakukan penyerahan sertifikat yang sudah selesai
2019	Suervisor PT. Wiratama Adji Jaya <ul style="list-style-type: none"> - Mengawasi Pekerjaan Lapangan (General Worker) - Membuat Laporan Absen dan lembur harian serta bulanan - Membuat perhitungan gaji setiap bulan berdasarkan absen dan lembur - Merekrut calon pekerja - Mendaftarkan Kontrak ke Disnaker - Memberikan Gaji kepada Karyawan

KEGIATAN SEMINAR/PELATIHAN

Tahun	Nama Seminar/Pelatihan	Instansi
2019	Cyber Security	BBPSDMP Kominfo Medan
2019	Desain Grafis	BBPSDMP kominfo medan & LSP Komputer

Demikianlah riwayat hidup ini dibuat dengan sebenarnya.