

**PENERAPAN ALGORITMA *GREEDY* UNTUK
MENENTUKAN RUTE TERPENDEK ANTAR
KLINIK GIGI DI KOTA MEDAN
BERBASIS *MOBILE***

SKRIPSI

**TAING SAPITRI NURDIANTI
71153011**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

**PENERAPAN ALGORITMA *GREEDY* UNTUK
MENENTUKAN RUTE TERPENDEK ANTAR
KLINIK GIGI DI KOTA MEDAN
BERBASIS *MOBILE***

SKRIPSI

Diajukan Untuk Memenuhi Syarat Mencapai Gelar Sarjana Komputer

**TAING SAPITRI NURDIANTI
71153011**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

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 mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	: Taing Sapitri Nurdianti
Nomor Induk Mahasiswa	: 71153011
Program Studi	: Ilmu Komputer
Judul	: Penerapan Algoritma <i>Greedy</i> Untuk Menentukan Rute Terpendek Antar Klinik Gigi Di Kota Medan Berbasis <i>Mobile</i> .

dapat disetujui untuk segera di *munaqasyah* kan. Atas perhatiannya kami ucapkan terimakasih.

Medan, 13 Februari 2020 M
19 Rabiul Akhir 1441 H

Komisi Pembimbing

Pembimbing Skripsi I,

Pembimbing Skripsi II,

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

Yusuf Ramadhan Nasution, M.Kom
NIB. 110000075

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Taing Sapitri Nurdianti
Nim : 71153011
Program Studi : Ilmu Komputer
Judul : Penerapan Algoritma *Greedy* Untuk Menentukan
Rute Terpendek Antar Klinik Gigi Di Kota Medan
Berdasarkan *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, 13 Februari 2020

Taing Sapitri Nurdianti
NIM. 71153011



PENGESAHAN SKRIPSI

Nomor: 060/ST/ST.V/PP.01.1/02/2020

Judul : Penerapan Algoritma *Greedy* Untuk Menentukan Rute Terpendek Antar Klinik Gigi Di Kota Medan Berbasis *Mobile*
Nama : Taing Sapitri Nurdianti
Nomor Induk Mahasiswa : 71153011
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 : Kamis, 13 Februari 2020
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah,
Ketua,

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

Dewan Penguji,

Penguji I,

Penguji II,

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

Yusuf Ramadhan Nasution, M.Kom
NIB. 1100000075

Penguji III,

Penguji IV,

Muhammad Ikhsan, S.T, M.Kom.
NIP. 198304152011011008

Sriani, M.Kom.
NIB. 1100000108

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. H. M. Jamil, M.A.
NIP.1966091019990310

ABSTRAK

Kesehatan gigi dan mulut adalah indikator kesehatan masyarakat. Adanya masalah kesehatan pada gigi dan mulut menyebabkan penurunan fungsi kesehatan individu. Masalah kesehatan gigi dan mulut disebabkan oleh kebiasaan seseorang menyikat gigi secara tidak teratur. Banyak pilihan yang dapat dibuat untuk sampai ke lokasi di mana klinik gigi terdekat diinginkan, tetapi ada keadaan di mana klinik gigi yang direncanakan tidak buka. Oleh karena itu dibutuhkan rute terpendek untuk sampai ke klinik gigi lain. Data dalam bentuk jarak, waktu dan rute untuk sampai ke klinik gigi yang diinginkan diambil dari peta *Google*, dengan total 10 simpul untuk diterapkan ke peta (*maps*). Untuk hasil peta, sebuah algoritma digunakan, dalam hal ini algoritma *Greedy* digunakan untuk memecahkan masalah menemukan rute terpendek. Berdasarkan hasil penelitian dari klinik gigi terdekat dengan titik awal yang dicari menggunakan algoritma *Greedy* itu menghasilkan total jarak, waktu perjalanan dan titik yang dilalui.

Kata kunci: *Google Maps*, Algoritma *Greedy*, *MySQL*, Klinik Gigi, *Android Mobile*.

ABSTRACT

Dental and oral health is an indicator of public health. The presence of health problems in teeth and mouth leads to a decrease in individual health functions. Dental and oral health problems are caused by the habit of a person brushing teeth unregularly. Many options can be made to get to the location where the nearest dental clinic is desirable, but there are circumstances where the planned dental clinic is not open. Therefore it takes the shortest route to get to another dental clinic. Data in the form of distance, time and route to get to the desired dental clinic is taken from Google map, with a total of 10 vertices to apply to the map (maps). For the results of the map, an algorithm is used, in which case the Greedy algorithm is used to solve the problem of finding the shortest route. Based on the research results from the dental clinic closest to the starting point is sought using the Greedy algorithm it generates the total distance, travel time and the point passed.

Keywords :Google Maps, Greedy Algorithms, MySQL, Dental Clinic, Mobile Android.

KATA PENGANTAR



Assalamu'aalaikum Warahmatullahii Wabarakatu.

Alhamdulillahirobbil'alamin. Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena dengan izin-Nya lah penulis dapat menyelesaikan Skripsi ini. Shalawat dansalam semoga senantiasa tercurah kepada junjungan penulis, yakni Nabi besar Muhammad SAW.

Skripsi ini untuk melengkapi serta memenuhi syarat untuk menjalankan program mata kuliah yang ada di semester VIII Ilmu Komputer Universitas Islam Negeri Sumatera Utara untuk mendapat gelar Strata-1. Dalam mengerjakan Skripsi ini penulis menyadari adanya keterbatasan, kemampuan, dan wawasan penulis, maka dari itu penulis menerima masukan dan juga saran dari semua pihak.

Dalam penulisan ini banyak sekali kesulitan yang penulis hadapi, tetapi berkat bimbingan dan juga bantuan dari berbagai pihak akhirnya penulis dapat menyelesaikan Skripsi ini dengan judul : ***"Penerapan Algoritma Greedy Untuk Menentukan Rute Terpendek Antar Klinik Gigi di Kota Medan Berbasis Mobile"***.

Demi kelancaran dalam penyelesaian skripsi ini tidak lepas dari bantuan para pihak, khususnya Ayah dan Ibu, yaitu Pinayungan Lubis dan Sakira yang memberikan bantuan moral dan materi, serta Antusiasme dan doa yang begitu besar bagi penulis.

Penulis menyadari bahwa penyusunan skripsi ini berkat doa, perhatian, bantuan, bimbingan, motivasi dan dukungan dari berbagai pihak, sehingga dengan ketulusan dan kerendahan hati pada kesempatan ini, para penulis ingin mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Bapak Prof. Dr. Saidurrahman, M.Ag, selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.

2. Bapak Dr. H. M. Jamil, MA, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Dr. Mhd Furqan, S,Si., M.Comp.Sc selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan dan sebagai dosen pembimbing skripsi I serta pembimbing akademik yang telah berkontribusi membantu penulis untuk memberikan ide, saran, kritik, dan bimbingannya selama penulis mengerjakan skripsi ini.
4. Bapak Yusuf Ramadhan Nasution, M.Kom sebagai dosen pembimbing skripsi II yang telah berkontribusi membantu penulis untuk memberikan ide, saran, kritik, dan bimbingannya bagi penulis dalam mengerjakan skripsi ini.
5. Bapak Rakhmat Kurniawan, ST., M.Kom selaku Kepala Laboratorium yang telah membantu penulis untuk memberikan ide, saran, dan bimbingannya bagi penulis.
6. Teristimewa untuk kedua orangtua penulis, Ayah kami Pinayungan Lubis, dan Ibu kami Sakirah tercinta yang tidak pernah berhenti memberikan doa disetiap waktu dalam hidupnya dan selalu memberikan dorongan moril maupun materil bagi penulis sehingga bisa mengerjakan skripsi ini.
7. Adik-Adik kandung penulis, Lian Saputra Nurdiansyah Lubis dan Roni Ilham Nurdiansyah Lubis terimakasih untuk dukungan, doa dan semangatnya yang selalu diberikan untuk penulis.
8. Seluruh tenaga pengajar dan juga pegawai dari program studi S1 Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara.
9. Teman-teman kelas Ilmu Komputer 1 yang selalumemberikan dukungan serta arahan kepada penulis.
10. Kepada teman teman seperjuangan Angkatan I, Khairunnisa, Sapmalan Nasution, Rabiatul Adawiyah, dan lainnya yang tidak bias disebut satu persatu terimakasih telah tmemberikan kontribusi dalam penyelesaian skripsi ini.
11. Teruntuk Ahmad Fuadi Nasution, terimakasih telah memberikan dukungannya disetiap waktu, terimakasih untuk doa, semangat, dan

bantuannya dalam hal apapun, serta selalu menemani penulis hingga akhir penyelesaian akhir skripsi ini.

12. Sahabatku, Meilani, Lisda Asmidah Nasution, terimakasih atas doanya, dan bantuannya sehingga penulis dapat menyelesaikan skripsi ini.
13. Adik kos tercinta, Nurlaili Wisfa, terimakasih untuk doa, semangat, dan bantuannya atas nasihat dan pemberian saran kepada penulis hingga dapat menyelesaikan proposal skripsi ini.
14. Dan semua pihak yang telah membantu penulis tetapi tidak bisa disebutkan satu persatu.

Penulis menyadari sepenuhnya bahwa dalam melakukan skripsi ini masih jauh dari kata sempurna, untuk itu kritik dan saran yang bersifat membangun, penulis sangat mengharapkan dari pembaca. Semoga hasil skripsi ini akan menjadi amal bagi penulis dan bermanfaat bagi pembaca. Aamiin Ya Rabbal'alamin.

Medan, September 2020

Hormat Saya,

Penulis

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah.....	3
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA	5
2.1. Klinik Gigi	5
2.2. Pengertian Algoritma.....	5
2.3. Teori <i>Graph</i>	7
2.3.1 Jenis-jenis <i>Graph</i>	8
2.4. Lintasan Terpendek (<i>Shortest Path</i>)	13
2.5. Optimisasi	14
2.6. Algoritma <i>Greedy</i>	14
2.7. <i>Google Maps API</i>	18
2.8. <i>Google Maps</i>	19

2.8. <i>Flowchart</i>	19
2.10. <i>Android</i>	21
2.11. <i>Java</i>	24
BAB III METODOLOGI PENELITIAN	25
3.1. Tempat Dan Waktu.....	25
3.1.1. Tempat Penelitian	25
3.1.2 Waktu Dan Jadwal Pelaksanaan Penelitian.....	25
3.2 Bahan dan Alat Penelitian.....	26
3.2.1. <i>Hardware</i> (Perangkat Keras).....	26
3.2.2 <i>Software</i> (Perangkat Lunak)	26
3.3 Cara Kerja	26
3.3.1 Perencanaan	26
3.3.2 Teknik Pengumpulan Data	27
3.3.3. Analisa Kebutuhan.....	28
3.3.4. Perancangan.....	29
3.3.5. Pengujian	33
3.3.6 Penerapan	34
BAB IV HASIL DAN PEMBAHASAN.....	35
4.1 Pembahasan.....	35
4.1.1. Analisis Data.....	35
4.1.2. Representasi Data.....	39
4.1.3 Perancangan.....	60
4.2 Hasil.....	67
4.2.1. Implementasi Sistem	67
BAB V KESIMPULAN DAN SARAN	73

5.1	Kesimpulan	73
5.2	Saran	73
DAFTAR PUSTAKA.....		74

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
Gambar 2.1	<i>Graph G</i>	7
Gambar 2.2	<i>Graph Sederhana</i>	8
Gambar 2.3	<i>Graph Semu (Pseudo Graph)</i>	8
Gambar 2.4	<i>Graph Ganda (Multile graph)</i>	9
Gambar 2.5	<i>Graph Tidak Berarah (Undirect Graph)</i>	9
Gambar 2.6	<i>Graph Berarah (Direct Graph atau Digraph)</i>	10
Gambar 2.7	<i>Graph Berhingga (Limited Graph)</i>	10
Gambar 2.8	<i>Graph Tidak Berhingga (Unlimitied Graph)</i>	11
Gambar 2.9	<i>Graph Tidak Berbobot (Unweighted Graph)</i>	11
Gambar 2.10	<i>Graph Berbobot (Weighted Graph)</i>	11
Gambar 2.11	<i>Graph Rute Vertex Awal A dan Vertex Akhir F</i>	16
Gambar 2.12	<i>Graph Rute yang dipilih Vertex A ke Vertex B</i>	17
Gambar 2.13	<i>Graph Rute yang dipilih Vertex B ke Vertex A</i>	17
Gambar 2.14	<i>Graph Rute yang dipilih Vertex D ke Verte E</i>	17
Gambar 2.15	<i>Graph Rute yang dipilih Vertex A ke VertexF</i>	18
Gambar 3.1	Tahap Perencanaan	27
Gambar 3.2	<i>Flowchart Algoritma Greedy</i>	30

Gambar 3.3 Desain <i>Interface</i> Utama	31
Gambar 3.4 Desain Halaman Utama <i>User</i>	32
Gambar 3.5 Desain Halaman Menu Pengujian	33
Gambar 4.1 Tampilan <i>Graph</i> Klinik Gigi di Kota Medan.....	40
Gambar 4.2 <i>Graph</i> Rute dari <i>Vertex</i> A ke <i>Vertex</i> B	42
Gambar 4.3 <i>Graph</i> Rute dari <i>Vertex</i> B ke <i>Vertex</i> A	43
Gambar 4.4 <i>Graph</i> Rute dari <i>Vertex</i> C ke <i>Vertex</i> D	44
Gambar 4.5 <i>Graph</i> Rute dari <i>Vertex</i> D ke <i>Vertex</i> C	45
Gambar 4.6 <i>Graph</i> Rute dari <i>Vertex</i> E ke <i>Vertex</i> D	47
Gambar 4.7 <i>Graph</i> Rute dari <i>Vertex</i> F ke <i>Vertex</i> H.....	48
Gambar 4.8 <i>Graph</i> Rute dari <i>Vertex</i> G ke <i>Vertex</i> J	49
Gambar 4.9 <i>Graph</i> Rute dari <i>Vertex</i> H ke <i>Vertex</i> F.....	50
Gambar 4.10 <i>Graph</i> Rute dari <i>Vertex</i> I ke <i>Vertex</i> J.....	52
Gambar 4.11 <i>Graph</i> Rute dari <i>Vertex</i> J ke <i>Vertex</i> I.....	53
Gambar 4.12 <i>Maps</i> Rute dari <i>Vertex</i> A ke <i>Vertex</i> B.....	54
Gambar 4.13 <i>Maps</i> Rute dari <i>Vertex</i> B ke <i>Vertex</i> A.....	54
Gambar 4.14 <i>Maps</i> Rute dari <i>Vertex</i> C ke <i>Vertex</i> D.....	55
Gambar 4.15 <i>Maps</i> Rute dari <i>Vertex</i> D ke <i>Vertex</i> C.....	55
Gambar 4.16 <i>Maps</i> Rute dari <i>Vertex</i> E ke <i>Vertex</i> D.....	56
Gambar 4.17 <i>Maps</i> Rute dari <i>Vertex</i> F ke <i>Vertex</i> H.....	56
Gambar 4.18 <i>Maps</i> Rute dari <i>Vertex</i> G ke <i>Vertex</i> J.....	57

Gambar 4.19 <i>Maps</i> Rute dari <i>Vertex</i> H ke <i>Vertex</i> F	57
Gambar 4.20 <i>Maps</i> Rute dari <i>Vertex</i> I ke <i>Vertex</i> J	58
Gambar 4.21 <i>Maps</i> Rute dari <i>Vertex</i> J ke <i>Vertex</i> I	58
Gambar 4.22 Diagram Tampilan Halaman Awal	63
Gambar 4.23 Diagram Tampilan Menu Utama	63
Gambar 4.24 Diagram Menu Informasi Klinik	64
Gambar 4.25 Diagram Tampilan Menu Pengujian	64
Gambar 4.26 <i>Flowchart</i> Halaman Awal	65
Gambar 4.27 <i>Flowchart</i> Menu Utama	65
Gambar 4.28 <i>Flowchart</i> Menu Beranda	66
Gambar 4.29 <i>Flowchart</i> Menu Informasi Klinik.....	66
Gambar 4.30 <i>Flowchart</i> Menu Pengujian	67
Gambar 4.31 Tampilan Halaman Awal	68
Gambar 4.32 Tampilan Halaman Menu Utama	69
Gambar 4.33 Tampilan Halaman Beranda	70
Gambar 4.34 Tampilan Halaman Menu Informasi Klinik	71
Gambar 4.35 Tampilan Halaman Menu Pengujian	72

DAFTAR TABEL

Tabel	Judul Tabel	Halaman
Tabel 2.2	Simbol-simbol <i>Flowchart</i>	20
Tabel 3.1	Waktu dan Jadwal Penelitian	25
Tabel 4.1	Nama-Nama Klinik Gigi yang Menjadi <i>Vertex</i>	36
Tabel 4.2	Nama-nama Klinik yang Dilalui yang Menjadi <i>Vertex</i>	37
Tabel 4.3	Hasil Rute Terpendek Algoritma <i>Greedy</i>	59
Tabel 4.4	Tabel Klinik	60
Tabel 4.5	Tabel <i>Step Over</i>	62

LAMPIRAN

Lampiran	Judul Lampiran
1.	<i>Script</i> Program
2.	<i>Curriculum Vitae</i>
3.	Bukti bimbingan skripsi

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kesehatan memang merupakan elemen mahal bagi manusia, sehingga dibutuhkan kepekaan pribadi untuk menjaganya. Salah satu organ tubuh yang sering diabaikan untuk dijaga yaitu gigi dan mulut (Syindy Wulandari, 2018). Pemeriksaan gigi merupakan bagian yang penting untuk merawat gigi seseorang. Pemeriksaan gigi di klinik atau rumah sakit dapat membantu untuk mengetahui tingkat kesehatan gigi seseorang (Budiono & Santoso, 2016). Klinik Gigi merupakan tempat perawatan gigi untuk gigi tetap sehat dan gigi dapat menjalankan fungsinya.

Di kota Medan, Praktik dokter gigi atau klinik gigi sudah banyak dan ramai dikunjungi oleh masyarakat. Mereka membutuhkan klinik gigi yang bisa memberikan pelayanan dan perawatan yang lengkap menyeluruh dan memuaskan. Permasalahan kesehatan gigi dan mulut telah menjadi bagian penting untuk kita tetap sehat. Masalah kesehatan mulut telah menjadi bagian penting bagi kita untuk tetap sehat. Pertanyaan tentang biaya dan waktu telah menjadi faktor yang paling diperhitungkan selama perjalanan. Misalnya perjalanan untuk menemukan tempat klinik gigi, karena tidak semua orang tahu daerah yang mereka tuju, jadi sulit untuk menemukan rute tercepat menuju klinik gigi lainnya.

Sebuah hadits menjelaskan ajaran untuk menjaga kebersihan gigi yang terkandung dalam Hadits Nabi Muhammad Saw. Tujuannya adalah untuk mengingatkan bahwa manusia selalu bersih sebelum melakukan ibadah wajib (shalat). Hadits ini mengingatkan betapa pentingnya bagi manusia untuk menjaga kebersihan (gigi) untuk menghindari berbagai penyakit, tetapi ada kekhawatiran dari Nabi Muhammad Saw. Kita telah melihat bahwa hadits ini akan memberatkan umat Islam sehingga beliau tidak akan mewajibkannya, bahkan jika dalam kehidupannya Nabi Muhammad Saw sehari-hari beliau menyikat giginya

beberapa kali. Diriwayatkan dari Abu Hurairah radhiyallahu anhu, Nabi Muhammad shallallahu ‘alaihi wa sallam bersabda:

صَلَاةٌ كُلِّ مَعِ بِالسَّوَاكِ لَأَمَرْتُهُمْ أُمَّتِي عَلَى أَشَقِّ أَنْ لَوْ لَا

“Seandainyaaku tidak memberatkan umat-ku, niscaya aku perintahkan kalian untuk bersiwak setiap kali (hendak) shalat.” Dan riwayat Imam Malik menyebutkan, “Setiap (hendak) berwudhu.” (HR. Al-Bukhar-Muslim).

Informasi mudah diperoleh pada saat teknologi sudah berkembang. Manusia membutuhkan suatu bentuk informasi seluler yang mudah diakses dan digunakan oleh semua umur, baik oleh orang tua maupun anak-anak. Perkembangan perangkat seluler saat ini mengarah ke perangkat *smartphone* yang saat ini populer di tengah masyarakat. Kapasitasnya cukup besar, seperti dapat menemukan informasi dengan mudah, untuk pendidikan dan sebagai sarana hiburan.

Saat ini ponsel tidak hanya berfungsi sebagai sarana telekomunikasi, namun telah beralih menjadi gadget yang mampu melakukan banyak hal (Prabowo et al., 2013). Salah satu yang bisa dimanfaatkan melalui *smartphone* adalah pencarian rute terpendek (*shortest path*). *shortest path* adalah pencarian jalur terpendek antara simpul grafik, kemudian menghasilkan bobot minimum. Pencarian rute terpendek ini telah diterapkan di berbagai bidang untuk mengoptimalkan kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat proses (Nurul Fithriani Harahap, 2017). Pencarian rute terpendek sangat berguna untuk menghemat waktu. Semakin cepat kita pergi ke klinik gigi, semakin banyak waktu kita harus mengunjungi klinik gigi lainnya. Upaya untuk mengatasi masalah menentukan jadwal perjalanan yaitu menyediakan aplikasi *mobile* dengan menerapkan algoritma *Greedy*. Aplikasi *mobile* ini menyediakan informasi tempat klinik gigi, jarak, dan waktu tempuh.

Algoritma *Greedy* ini membentuk solusi langkah, pada setiap langkahnya tentu path tersebut akan memiliki banyak pilihan dan kemungkinan yang dapat di eksplorasi, dengan algoritma ini keputusan langkah yang di ambil berikutnya adalah yang paling menguntungkan pada keadaan sekarang (Sabaruddin, 2016).

Berdasarkan uraian latar belakang diatas, penulis memutuskan untuk mengambil judul ***“Penerapan Algoritma Greedy Untuk Menentukan Rute Terpendek Antar Klinik Gigi di Kota Medan Berbasis Mobile”***.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah dalam skripsi ini adalah sebagai berikut :

1. Bagaimana menerapkan algoritma *Greedy* dalam pencarian rute terpendek antar klinik gigi di kota Medan.
2. Bagaimana mengimplementasikan aplikasi yang dapat menguji algoritma *Greedy* dalam melakukan pencarian rute terdekat antar klinik gigi dikota Medan.

1.3. Batasan Masalah

Dalam penulisan skripsi ini terdapat batasan-batasan, agar apa yang diteliti sesuai dengan tujuan penelitian. Adapun batasan masalah adalah sebagai berikut:

1. Proses pencarian dilakukan dengan metode algoritma *Greedy*.
2. Simpul klinik gigi sebanyak 10 titik, yaitu pada Klinik Gigi di Kota Medan.
3. Aplikasi ini dirancang khusus untuk pengguna *Smartphone* dengan OS *Android* minimum versi android 6.0 (*Marshmallow*).
4. *Software* yang digunakan adalah *Visual Studio Code*, *Javasebagai* bahasa pemrograman, *Android Cordova* sebagai *Framework*, dan *MySQL* sebagai *database*.
5. Aplikasi ini menggunakan *Android*.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut:

1. Mengetahui proses algoritma *Greedy* dalam menentukan rute terpendek antar klinik gigi di kota Medan.

2. Mengetahui rute terpendek antar klinik gigi dalam sebuah aplikasi berbasis *mobile*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Untuk mendapatkan pengetahuan dari algoritma *Greedy* dalam melakukan pencarian rute terpendek antar klinik gigi di kota Medan.
2. Agar dapat melakukan pencarian rute terpendek dari sebuah aplikasi berbasis *mobile*.

BAB II

TINJAUAN PUSTAKA

2.1. Klinik Gigi

Klinik gigi dalam Peraturan Menteri Kesehatan nomor 920/Menkes/Per/XII/1986, didefinisikan sebagai sarana pelayanan kesehatan gigi dan mulut yang diberikan kepada masyarakat. Klinik adalah tempat yang dibangun untuk menyediakan layanan kesehatan bagi seluruh masyarakat. Klinik gigi adalah bangunan atau tempat yang dibangun untuk memberikan perawatan gigi kepada seluruh masyarakat, yang meliputi upaya pencegahan, perawatan dan pemulihan. (Depkes, R.I, 1996).

Ada 3 kelompok manusia yang harus ditangani dalam proses mengelola pelayanan kesehatan yaitu:

1. *Health operator / provider* (dokter gigi, perawat gigi dan sebagainya).
2. Penerima jasa (pasien dan keluarga pasien).
3. Tenaga admin kesehatan.

pentingnya mengelola layanan kesehatan gigi dapat dilihat dari fenomena yang selalu terjadi di organisasi layanan kesehatan, di mana dokter gigi, perawat atau perawat gigi selalu diharuskan untuk memberikan layanan berkualitas tinggi dengan metode perawatan terbaru dengan menggunakan teknologi dan obat-obatan yang dapat memberikan keamanan dan kepastian penyembuhan dan kesehatan pasien.

2.2. Pengertian Algoritma

Algoritma adalah metode efektif yang diekspresikan sebagai rangkaian terbatas. Algoritma juga merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan syarat untuk setiap permasalahan memiliki kriteria kondisi awal yang harus dipenuhi sebelum menjalankan sebuah algoritma. Algoritma juga memiliki pengulangan proses (iterasi), dan juga memiliki keputusan hingga keputusan selesai (Maulana, 2017).

Algoritma berbeda dengan logaritma. Logaritma merupakan operasi matematika yang merupakan kebalikan dari eksponen atau pemangkatan. contoh logaritma seperti $b^c = a$ ditulis sebagai $\log_b a = c$ (b disebut basis) (Maulana, 2017).

Sifat –sifat Algoritma

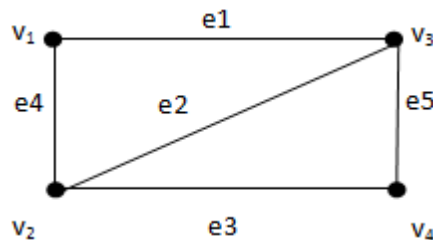
sifat utama dari algoritma adalah sebagai berikut:

1. *Input*: Algoritma memiliki input atau kondisi awal sebelum implementasinya, yang dapat berupa nilai variabel yang diambil dari set khusus.
2. *Output* : suatu algoritma akan menghasilkan *output* setelah implementasinya, atau algoritma tersebut akan mengubah kondisi awal menjadi kondisi akhir, di mana nilai *output* diperoleh dari nilai input yang telah diproses melalui algoritma.
3. *Definiteness*: langkah-langkah yang dituliskan dalam algoritma terdefinisikan dengan jelas jadi mudah dilakukan oleh pengguna algoritma.
4. *Finiteness* : Setiap algoritma harus memberikan hasil akhir atau keluaran setelah sejumlah langkah yang terbatas dilakukan terhadap setiap kondisi awal atau masukan yang diberikan.
5. *Effectiveness*: Setiap langkah dalam algoritma dapat dilakukan dalam suatu selang waktu tertentu sehingga pada akhirnya diperoleh solusi yang sesuai yang diharapkan.
6. *Generality* : langkah-langkah algoritma berlaku untuk setiap set input yang sesuai dengan masalah yang diberikan, bukan hanya untuk set tertentu.

2.3. Teori Graf

Graf adalah suatu diagram yang memuat informasi tertentu jika diinterpretasikan secara tepat (Chayatul Ichan, 2018). Dalam elemen Graf G didefinisikan sebagai pasangan himpunan (V, E) dan dinotasikan sebagai berikut $G = (V, E)$. V adalah himpunan tidak kosong dari sebuah simpul (*Node*) dan E adalah himpunan sisi (*edge*) yang menghubungkan antara sepasang simpul dan *Node* E (kosong). *Node* dalam suatu Graf dapat dilambangkan dengan huruf abjad seperti : a, b, c, d, ..., v, w, ... dengan bilangan asli seperti 1, 2, 3, ..., atau dapat digabung dari keduanya (Fitriyani, 2018).

Jumlah simpul dalam grafik disebut urutan dan jumlah tepi disebut ukuran. Dalam Grafik G , jika ada dua simpul yang terhubung ke tepi, tepi itu disebut beberapa sisi atau tepi ganda. Dan jika suatu sisi berasal dari suatu simpul dan kembali ke simpul aslinya, simpul itu disebut lingkaran. Grafik berikut G berisi titik set $V(G)$ dan sisi set $E(G)$ (Nurul Fithriani Harahap, 2017).



Gambar 2.1 *Graph G*

(Sumber : Khairani Ulfa Nst, 2017).

Berdasarkan gambar 2.1 grafik G memiliki 5 titik. Grafik G dengan himpunan titik dan sisi masing - masing dapat ditulis:

$$V(G) = \{ v1, v2, v3, v4, v5 \}$$

$$E(G) = \{ e1, e2, e3, e4, e5 \}$$

yaitu :

$$e1 = (v1, v3)$$

$$e2 = (v2, v3)$$

$$e3 = (v2, v4)$$

$$e4 = (v1, v2)$$

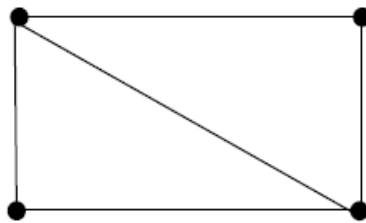
$$e5 = (v3, v4)$$

2.3.1. Jenis–Jenis *Graph*

Grafik terdiri dari beberapa jenis. Grafik berdasarkan ada atau tidak adanya sisi pada grafik, dapat diklasifikasikan menjadi dua jenis, yaitu :

1. *Graph* Sederhana (*SimpleGraph*)

Grafik yang tidak memiliki perulangan atau memiliki tepi ganda. Berikut adalah contoh grafik sederhana:



Gambar 2.2 *Graph* Sederhana

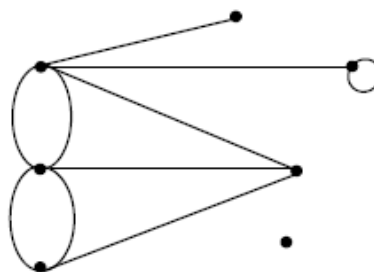
(Sumber : Khairani Ulfa Nst, 2017).

2. *Graph* Tidak Sederhana

Grafik tepi ganda atau loop disebut grafik tidak sederhana. Dalam grafik sederhana, dapat dibagi menjadi dua, yaitu:

a. Grafik Semu (*Pseudo Graph*)

grafik semu atau *pseudo graphs* adalah grafik yang memiliki loop, termasuk grafik yang memiliki perulangan dan tepi ganda karena pseudo-grafis lebih umum daripada beberapa grafik, *pseudo-edge* dapat dihubungkan ke dirinya sendiri.

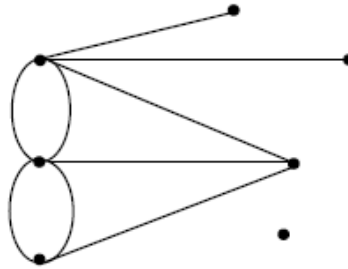


Gambar 2.3 *Graph* Semu (*Pseudo Graph*)

(Sumber : Khairani Ulfa Nst, 2017).

b. Grafik Ganda (*Multiple graph*)

Multiple graph adalah grafik tepi ganda. Tepi dapat menghubungkan sepasang simpul atau lebih dari dua.



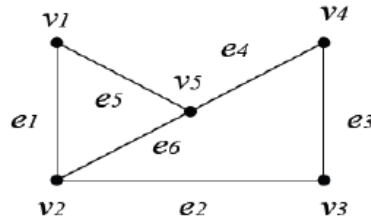
Gambar 2.4 *Graph Ganda (Multiple graph)*

(Sumber : Khairani Ulfa Nst, 2017).

Berdasarkan jenis garisnya, grafik dibedakan dalam 2 kategori adalah :

1. *Graph Tidak Berarah (Undirect Graph)*

Grafik tidak berarah adalah grafik yang energinya tidak memiliki arah atau panah. Dalam grafik yang tidak berorientasi, elemen E disebut *edge*.



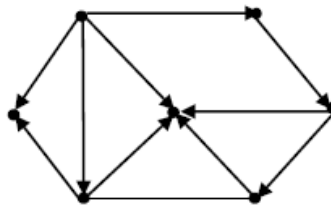
Gambar 2.5 *Graph Tidak Berarah (Undirect Graph)*

(Sumber : Khairani Ulfa Nst, 2017).

Grafik pada Gambar 2.5 adalah grafik tidak berarah dengan himpunan simpul $V(G) = \{v1, v2, v3, v4, v5\}$ dan himpunan sisi $E(G) = \{e1, e2, e3, e4, e5\}$ yaitu pasangan $\{(v1, v2), (v2, v3), (v3, v4), (v4, v5), (v5, v2)\}$.

2. *Graph Berarah (Direct Graph atau Digraph).*

Jika dalam grafik, semua energi menerima arah, grafik seperti ini disebut grafik terarah (grafik langsung) dengan notasi D. Grafik berarah D terdiri dari satu set simpul V dan satu set of arc: Terdiri dari pasangan simpul yang diurutkan yang ditunjukkan oleh (u, v) yang disebut sisi terarah (*arc*). Simpul u dalam *arc* ini disebut simpul awal dan simpul v disebut simpul akhir. Grafik terarah atau grafik ditulis dengan $D = (V, A)$.



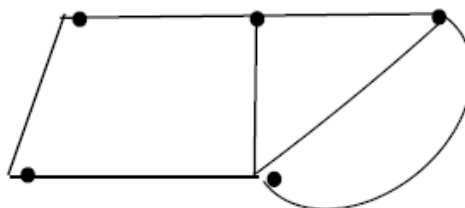
Gambar 2.6 *Graph Berarah (Direct Graph atau Digraph).*

(Sumber : Khairani Ulfa Nst, 2017).

Berdasarkan jumlah simpul pada suatu grafik, maka secara umum grafik dapat digolongkan menjadi dua jenis:

1. *Graph Berhingga (Limited Graph)*

Grafik berhingga adalah grafik yang memiliki jumlah simpul tidak terbatas atau masih dapat dihitung n . Berikut adalah contoh grafik yang sudah jadi.

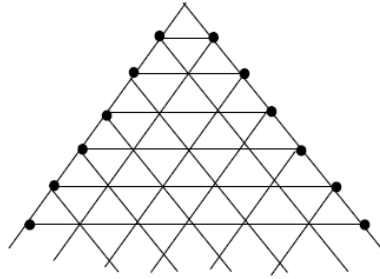


Gambar 2.7 *Graph Berhingga (Limited Graph)*

(Sumber : Khairani Ulfa Nst, 2017).

2. *Graph Tidak Terhingga (Unlimited Graph)*

Grafik tak terhingga adalah grafik dengan jumlah simpul tak terbatas, n tak terbatas. Berikut adalah contoh grafik yang sudah jadi.



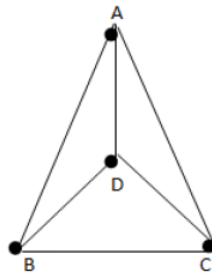
Gambar 2.8 *Graph Tidak Terhingga (Unlimited Graph)*

(Sumber : Khairani Ulfa Nst, 2017).

Berdasarkan bobotnya, grafik dibagi menjadi dua jenis, sebagai berikut:

1. *Graph* tidak berbobot (*Unweighted Graph*)

Grafik tidak berbobot atau grafik tidak berlabel adalah grafik yang setiap *edgenya* tidak memiliki nilai (bobot).

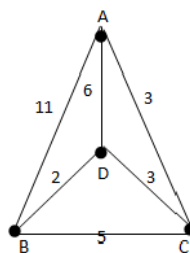


Gambar 2.9 *Graph Tidak Berbobot (unweighted Graph)*

(Sumber : Khairani Ulfa Nst, 2017).

2. *Graph* Berbobot (*Weighted Graph*)

Grafik berbobot atau berlabel adalah grafik yang setiap *edge* nya menerima nilai (berat), sehingga nilainya dapat dihitung pada grafik.



Gambar 2.10 *Graph Berbobot (Weighted Graph)*

(Sumber : Khairani Ulfa Nst, 2017).

Untuk mempelajari *Graf* secara lebih, perlu diketahui beberapa terminologi sebagai berikut:

1) Bertetangga (*Adjacent*)

Dua buah simpul pada grafik tidak berarah jika keduanya terhubung disebut bertetangga.

2) Bersisian (*Incident*)

Untuk setiap sisi $e = (v_j, v_k)$, sisi e seharusnya hidup berdampingan dengan simpul v_j dan v_k .

3) Simpul terkecil (*Isolated Vertex*)

Node yang tidak memiliki sisi berdampingan dalam arah *node* jarak jauh.

4) *Graph* Kosong (*Null Graph*)

Grafik yang himpunan sisinya merupakan himpunan kosong.

5) Derajat (*Degree*)

Tingkat simpul adalah jumlah sisi yang memotongnya.

6) Lintasan (*Path*)

Jalur yang panjangnya n dari peralihan awal menuju pergantian v_k dengan melewati peralihan sisi dan pergantian penuh..

7) Siklus/Sirkuit (*Cycle/Circuit*)

Jalur yang dimulai dan berakhir pada simpul yang sama.

8) Terhubung (*Connected*)

Grafik berorientasi adalah grafik terhubung jika, untuk setiap simpul grafik, ada jalur ke simpul ini.

9) *Upagraf* (*Subgraph*)

Upagraf merupakan bagian dari suatu grafik (*subset* suatu grafik).

10) *Upagraf* Merentang (*Spanning Subgraph*)

Sebuah *upagraf* yang berisi semua simpul dari grafik utama.

11) *Cut-Set*

Cut-Set grafik adalah jika anggota tertentu dari set tepi dihapus, yang mencegah grafik agar tidak terhubung..

12) *Graph* Berbobot (*Weighted Graph*)

Grafik berbobot adalah grafik yang setiap sisinya diberikan nilai / bobot (biaya).

2.4. Jalur Terpendek (*Shortest Path*)

Persoalan mencari lintasan terpendek di dalam graf merupakan salah satu persoalan optimasi. graf yang digunakan dalam pencarian lintasan terpendek adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya (Mahendra et al., 2019).

Asumsi yang digunakan di sini adalah bahwa semua bobot bernilai positif. Lintasan terpendek adalah jalur yang dilalui dari suatu *node* ke *node* lain dengan besar atau nilai pada sisi yang jumlah akhirnya dari *node* awal ke *node* akhir paling kecil. Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat lain. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf yang digunakan adalah graf yang berbobot yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot (Hayati & Yohanes, 2014).

Jalur terpendek adalah jalur minimum yang diperlukan untuk mencapai lokasi dari lokasi tertentu. Ada beberapa jenis masalah jalur minimum, yaitu :

- 1) Jalur minimum antara dua titik.
- 2) Jalur minimum antara semua pasangan titiknya.
- 3) Jalur minimum dari titik tertentu ke semua titik lainnya.
- 4) Jalur minimum antara dua titik yang melewati titik-titik tertentu.

Lintasan minimum tersebut dapat dicari menggunakan grafik. Grafik yang digunakan adalah grafik berbobot, yaitu grafik di mana setiap sisi menerima nilai atau bobot. Dalam hal ini, berat yang diharapkan adalah dalam bentuk jarak dan waktu (Dhea Fithaloka, 2017).

2.5. Optimisasi

Optimisasi merupakan kumpulan prinsip-prinsip matematika dan metode yang digunakan untuk menentukan penyelesaian dari persoalan kuantitatif dalam banyak disiplin ilmu, termasuk fisika, biologi, teknik, ekonomi, dan bisnis. Banyak persoalan dapat dirumuskan dan dipecahkan dengan menggunakan sekumpulan gagasan dan metode terpadu yang membentuk bidang optimisasi.

Istilah pemrograman matematika, secara luas identik dengan optimisasi, dimulai pada tahun 1940 sebelum pemrograman disamakan dengan pemrograman komputer. Pemrograman matematika meliputi studi tentang struktur matematika dari masalah optimisasi, penemuan metode untuk memecahkan persoalan ini, studi tentang sifat matematis dari metode ini, dan penerapan pada metode ini pada komputer. Komputer yang lebih cepat memperluas ukuran dan kompleksitas masalah optimisasi yang dapat dipecahkan. Pengembangan teknik optimisasi paralel dengan kemajuan tidak hanya dalam ilmu komputer tetapi juga dalam riset operasi, analisis numerik, teori permainan, matematika ekonomi, teori kontrol, dan kombinatorik.

Masalah optimisasi sering dimulai dengan serangkaian variabel dan parameter independen, dan mencakup persyaratan atau batasan pada nilai variabel independen yang dapat diterima. Pembatasan ini disebut sebagai kendala masalah. Elemen penting lain dari masalah optimisasi adalah ukuran "baik" yang disebut fungsi objektif yang tergantung pada variabel masalah. Solusi dari masalah optimisasi adalah himpunan nilai variabel yang memenuhi kendala, sehingga fungsi tujuan mencapai nilai optimal (Mhd Furqan, 2019).

2.6. Algoritma *Greedy*

Algoritma *Greedy* adalah algoritma untuk menemukan solusi optimal untuk masalah optimasi dengan menyelesaikan masalah langkah demi langkah. Algoritma *Greedy* merupakan algoritma yang dapat memecahkan masalah dengan membuat pilihan yang tampaknya terbaik pada saat tertentu. Banyak masalah optimasi yang dapat diselesaikan dengan menggunakan algoritma *Greedy*.

Beberapa masalah mungkin tidak menemukan solusi yang efisien, tetapi pada algoritma *Greedy* mampu memberikan solusi efisien yang dekat dengan optimal.

Solusi yang terbentuk dalam algoritma *Greedy* pada setiap langkah adalah sebagai berikut :

1. Pada setiap langkah yang dipilih harus menjadi pilihan terbaik, di mana pilihan yang telah dibuat dalam satu langkah tidak dapat diubah pada langkah berikutnya.
2. Pendekatan yang digunakan dalam algoritma *Greedy* terdiri dalam memilih optimum lokal pada setiap langkah. Masalah optimisasi dalam konteks algoritma *Greedy* diatur oleh elemen-elemen sebagai berikut:

Prinsip algoritma *greedy* adalah memilih keputusan yang terbaik dalam setiap langkahnya (Optimum local). Keputusan local yang di ambil akhirnya akan menjadi solusi optimum yang mutlak (Optimum global). Berikut adalah elemen dari algoritma *Greedy* (Hartanto et al., 2019):

1. Himpunan Kandidat

Set ini berisi elemen-elemen yang membentuk solusi. Untuk masalah jalur terpendek dalam grafik, set kandidat ini adalah himpunan simpul dalam grafik.

2. Himpunan Solusi

Set ini berisi kandidat yang dipilih sebagai solusi untuk masalah tersebut. Set ini adalah himpunan bagian dari himpunan kandidat.

3. Fungsi Seleksi

Fungsi Seleksi (*Selection Function*) Fungsi ini dinyatakan dengan predikat seleksi. Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya (Oktaviana & Naufal, 2017).

4. Fungsi Kelayakan

Fungsi yang memeriksa apakah kandidat yang dipilih dapat memberikan solusi yang bisa diterapkan.

5. Fungsi Objektif

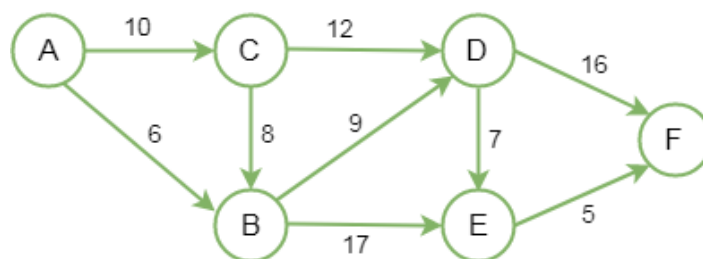
Fungsi yang memaksimalkan dan meminimum nilai-nilai solusi. Tujuannya adalah untuk memilih solusi terbaik dari masing-masing anggota set solusi.

Dalam algoritma *Greedy*, masalah tertentu tidak selalu berhasil memberikan solusi yang benar-benar optimal. Namun, algoritma *Greedy* tentu saja memberikan solusi yang mendekati nilai optimal (Nurul Fithriani Harahap, 2017).

Algoritma *greedy* untuk menemukan jalur terpendek dapat dirumuskan adalah sebagai berikut:

1. Periksa semua tepi yang berbatasan langsung dengan simpul a. Pilih tepi dengan bobot terkecil. Tepi ini pertama menjadi jalur terpendek, sebut saja $L(1)$.
2. Hitung jalur terpendek keduanya dengan cara sebagai berikut:
 - a) Hitung: $d(i) = \text{panjang } L(1) + \text{bobot tepi dari simpul terakhir } L(1) \text{ ke simpul } i \text{ yang lain.}$
 - b) Pilih $d(i)$ yang terkecil. Bandingkan $d(i)$ dengan bobot tepi (a, i) . Jika bobot tepi (a, i) kurang dari $d(i)$, maka $L(2) = L(1) \cup (\text{tepi simpul akhir } L(1) \text{ ke simpul } i)$.
3. Demikian pula, ulangi langkah 2 untuk menentukan jalur terpendek berikutnya.

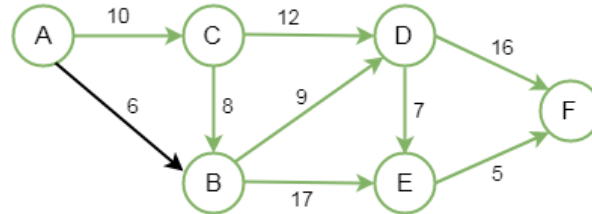
Contoh sederhana solusi grafik untuk menentukan rute terpendek menggunakan algoritma *Greedy* dengan titik awal A dan titik akhir F dapat dilihat pada gambar 2.11.



Gambar 2.11 Graph Rute Vertex Awal A dan Vertex Akhir F

(Sumber : Nurul Fithriyani Harahap, 2017)

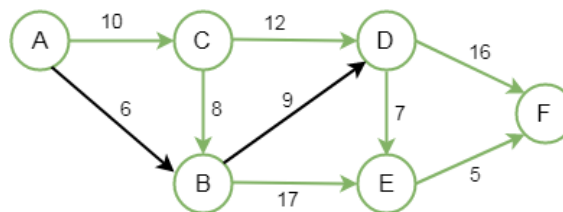
Jalur terpendek pertama yang dipilih dari tepi yang berbatasan langsung dengan titik A adalah titik B (AB) dengan bobot 6, ditunjukkan pada gambar 2.12.



Gambar 2.12 *Graph* Rute dari *Vertex* A ke *Vertex* B

(Sumber : Nurul Fithriyani Harahap, 2017)

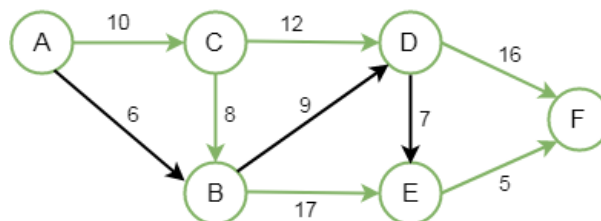
Kemudian pilih tepi yang dekat dengan tepi AB, yang merupakan tepi BD dengan berat 9 sehingga berat total adalah 15, tepi BE juga dekat dengan tepi AB dengan berat 17, sehingga total berat adalah 23. Jadi pilih tepi BD karena memiliki bobot yang lebih kecil, seperti yang ditunjukkan pada gambar 2.13.



Gambar 2.13 *Graph* Rute dari *Vertex* B ke *Vertex* D

(Sumber : Nurul Fithriyani Harahap, 2017)

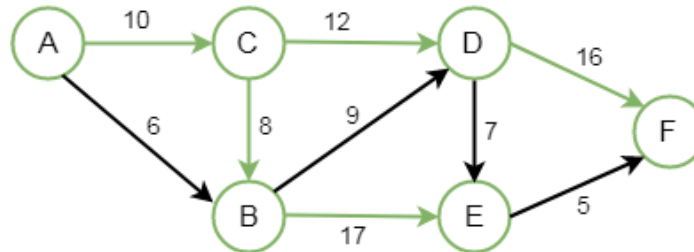
Kemudian pilih tepi yang dekat dengan tepi BD, yang merupakan tepi DF dengan berat 16 sehingga berat total 31, tepi DE juga dekat dengan tepi BD dengan berat 7, sehingga berat total adalah 22. Jadi pilih tepi ED karena memiliki bobot lebih kecil, seperti yang ditunjukkan pada gambar 2.14.



Gambar 2.14 *Graph* Rute yang dipilih dari *Vertex* D ke *Vertex* E

(Sumber : Nurul Fithriyani Harahap, 2017)

Kemudian pilih tepi yang berbatasan dengan tepi DE, yang merupakan tepi EF dengan berat 5 sehingga total beratnya adalah 27, seperti yang ditunjukkan pada gambar 2.15.



Gambar 2.15 Graph Rute dari Vertex E ke Vertex F
(Sumber : Nurul Fithriyani Harahap, 2017)

Hasil yang diperoleh dari contoh sederhana di atas menentukan rute terpendek menggunakan algoritma Greedy adalah $A \rightarrow B \rightarrow D \rightarrow E \rightarrow F$ dengan total jarak 27.

2.7. Google Maps API

Google Maps API adalah antarmuka aplikasi yang dapat diakses melalui *JavaScript* sehingga *Google Maps* dapat ditampilkan pada halaman web yang sedang dibuat. Menggunakan *Google Maps API* menghemat waktu dan uang untuk membuat aplikasi peta digital yang andal, karena pengguna hanya fokus pada data yang ingin mereka sesuaikan, karena data peta sudah disediakan oleh *Google Maps*. Ada 2 cara untuk mengakses data dari *Google Maps*, tergantung pada data yang ingin kita pulihkan dan dekripsi dari *Google Maps*, adalah sebagai berikut:

1. Tidak harus menggunakan *API Key* untuk akses data dari *Google Maps*.
2. Tidak harus menggunakan *API key* untuk akses data dari *Google Maps*.
3. Untuk pendaftaran *API key* harus dilakukan dengan data registrasi dalam bentuk nama *domain web* yang kita buat.

2.8. *Google Maps* (Peta *Google*)

Google Maps adalah layanan gratis yang diberikan oleh *Google* dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google Maps* dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript* (Rismayani & Ardimansyah, 2015).

Untuk dapat mengakses data dari *Google Maps*, kita memerlukan layanan pemetaan *web* gratis yang disediakan oleh *Google*, yang memiliki kapasitas banyak layanan pemetaan *web*. *Google Maps* juga memiliki properti sisi *server*, yaitu, peta yang disimpan di *server Google* dapat digunakan oleh pengguna.

2.9. *Flowchart* (Diagram Alur)

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek (Santoso & Nurmalina, 2017).


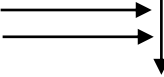



Flowchart membantu analis dan pemrogram memecahkan masalah dalam segmen yang lebih kecil dan membantu menganalisis alternatif lain dalam operasi. Bagan organisasi umumnya memfasilitasi penyelesaian masalah, terutama masalah yang perlu diselidiki dan dievaluasi lebih lanjut. Bagan organisasi adalah bentuk gambar/diagram yang memiliki satu atau dua arah aliran secara berurutan. Bagan organisasi digunakan untuk mewakili dan merancang program. Oleh karena itu, bagan organisasi harus dapat mewakili komponen dalam bahasa pemrograman.

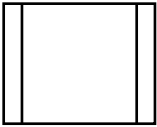
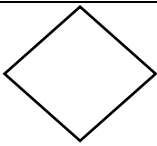
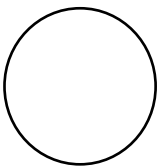
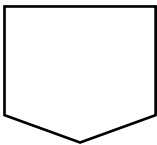



Pedoman untuk menggambar diagram alur, analisis sistem atau pemrograman adalah sebagai berikut :

- a. Bagan alir harus digambar dari atas ke bawah dan mulai dari sisi kiri halaman.

- b. Kegiatan dalam suatu bagan alir harus dilakukan dengan jelas.
- c. Kita harus menunjukkan di mana aktivitas akan dimulai dan di mana itu akan berakhir.
- d. Setiap aktivitas dalam bagan organisasi harus menggunakan kata yang mewakili pekerjaan, misalnya; "Siapkan" dokumen "menghitung" gaji.
- e. Setiap aktivitas dalam bagan organisasi harus dibentuk sebelumnya.
- f. Aktivitas terputus yang terhubung ke tempat lain harus diilustrasikan dengan jelas dengan simbol koneksi.
- g. Gunakan simbol bagan alur standar.

Tabel 2.2 Simbol-simbol *Flowchart*

No	Simbol	Keterangan
1	 Terminator (Simbol Terminal)	Permulaan dan akhir dari suatu program.
2	 Garis Alur (<i>Line Follower</i>)	Arah Aliran dari suatu Program.
3	 Simbol Persiapan	Proses instalasi atau definisi nilai awal.
4	 Simbol Proses	Proses Penghitungan atau pemrosesan data.
5	 <i>Input dan Output</i>	Memproses data <i>input/output</i> , parameter, informasi.

6	 <p>Simbol Sub Proses</p>	Awal subprogram/proses eksekusi subprogram.
7	 <p>Simbol Keputusan</p>	Perbandingan deklarasi, pemilihan data yang menyediakan opsi untuk langkah sebelumnya.
8	 <p><i>Connector</i></p>	Tautan ke bagan organisasi pada satu halaman.
9	 <p>Simbol Penghubung</p>	Koneksi bagian-bagian bagan organisasi yang ada di halaman yang berbeda.
10	 <p><i>Document</i></p>	<i>Input/output</i> dalam format cetak.
11	 <p><i>Disk Storage</i></p>	<i>Input/Output</i> yang menggunakan penyimpanan akses langsung.
12		Entri ekspres

	<i>punched card</i>	berasal dari kartu atau output tertulis di kartu.
--	---------------------	---

2.10. *Android*

Android adalah bagian dari perangkat lunak untuk perangkat seluler yang mencakup sistem operasi middleware dan aplikasi utama yang diterbitkan oleh *Google.Android SDK* (Perangkat Pengembangan Perangkat Lunak) menyediakan alat dan *API* yang diperlukan untuk mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Kisah *Android* dimulai pada 2005, *Google* kemudian tahun itu juga mulai membangun *platform Android* intensif. Pada 12 November 2007, *Google*, bekerja sama dengan *Open Handset Alliance* (OHA), sebuah konsorsium perangkat seluler terbuka, merilis *Google Android SDK*, setelah mengumumkan seminggu sebelumnya.

Android adalah sistem operasi bersifat *open source* berbasis *Linux* dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. *Android* awalnya dikembangkan oleh *Android, Inc.*, dengan dukungan finansial dari *Google*, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Ponsel Android* pertama mulai dijual pada bulan Oktober 2008. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance* yang merupakan konsorsium dari 34 perusahaan perangkat keras, perangkat lunak dan telekomunikasi (Surahman & Setiawan, 2017).

Android merupakan sistem operasi *mobile*. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga (Anwar et al., 2015).

Saat ini ada banyak *platform* untuk perangkat seluler, termasuk *Symbian*, *iOS*, *Windows Mobile*, *BlackBerry*, *Java Mobile Edition*, *Linux Mobile* (LiM) dan

banyak lagi. Namun ada beberapa hal tertentu yang menjadi kekuatan *Android*. Meskipun beberapa fitur yang telah muncul sebelumnya pada *platform* lain, *Android* adalah yang pertama menggabungkan hal-hal seperti berikut:

1. Kebebasan, pengembangan gratis tanpa biaya ke sistem karena didasarkan pada *Linux* dan *open source*. Produsen perangkat menyukai ini karena dapat membangun *platform* seperti yang diinginkan tanpa membayar royalti. Sehingga disukai pengembang perangkat lunak, karena *Android* dapat digunakan di perangkat apa pun dan tidak diikat oleh penyedia mana pun.
2. Arsitektur dari komponen dasar *Android* terinspirasi oleh teknologi mashup Internet. Bagian dalam suatu aplikasi dapat digunakan oleh aplikasi lain, bahkan dapat digantikan oleh komponen lain yang kompatibel dengan aplikasi yang sedang dikembangkan.
3. Banyak dukungan layanan, kemudahan penggunaan berbagai layanan dalam aplikasi seperti menggunakan layanan pencarian lokasi, *database SQL*, *browser* dan penggunaan peta. Semua ini telah terintegrasi dengan *Android*, yang membuatnya lebih mudah untuk mengembangkan aplikasi.

Siklus hidup aplikasi didefinisikan secara otomatis, setiap program dikelola di antara mereka dengan lapisan keamanan yang berbeda, sehingga pekerjaan sistem menjadi lebih stabil. Pengguna tidak perlu khawatir menggunakan aplikasi pada perangkat dengan memori terbatas.

Level *API* adalah nilai integer yang secara unik mengidentifikasi kerangka *API* yang direvisi yang ditawarkan oleh versi *platformAndroid*. Versi *Android API* yang dirilis adalah :

1. *Android Tipe 1.1*
2. *Android versi 1.5 (Cupcake)*
3. *Android versi 1.6 (Donut)*
4. *Android versi 2.0/2.1 (Eclair)*
5. *Android versi 2.2 (Froyo: Frozen Yoghurt)*
6. *Android versi 2.3 (Gingerbread)*
7. *Android versi 3.0/3.1 (Honeycomb)*

8. *Android* versi 4.0 ICS (*Ice Cream Sandwich*)
9. *Android* versi 4.1/4.2/4.3 (*Jelly Bean*)
10. *Android* versi 5.0.2 (*Lollipop*)
11. *Android* versi 6.0 (*Marshmallow*)
12. *Android* versi 7.0 (*Nougat*)
13. *Android* versi 8.0 (*Oreo*)
14. *Android* versi 9.0 (*Pie*)

Adapun level *API* penting untuk pengembang aplikasi *Android* meliputi::

1. Kerangka kerja peninjauan *API* maksimum yang mendukung.
2. Kerangka kerja *API* revisi yang diperlukan oleh aplikasi Versi *API* yang tidak kompatibel.
3. Setiap versi *platform* menyimpan pengidentifikasi tingkat *API* secara internal.

2.11. Pemrograman *Java* dan *JavaScript*

Java adalah bahasa pemrograman berorientasi objek murni yang didasarkan pada kemampuan terbaik dari bahasa pemrograman objek sebelumnya (*C ++*, *Ada*, *Simula*). *Java* diciptakan oleh James Gosling, pengembang Sun Microsystems pada tahun 1991.

Karakteristik *Java* antara lain :

1. Sederhana (*Simple*)
2. Berorientasi Objek (*Object Oriented*)
3. Terdistribusi (*Distributed*)
4. *Interpreted*
5. *Robust*
6. Aman (*Secure*)
7. *Architecture Neutral*
8. *Portable*
9. *Performance*
10. *Multithreaded*
11. Dinamis

JavaScript adalah bahasa *script* yang ditempelkan pada kode HTML dan proses pada sisi *klien*, sehingga kemampuan dokumen HTML menjadi lebih luas (Ripai, 2017)

BAB III

METODOLOGI PENELITIAN

3.1. Tempat Dan Waktu

3.1.1. Tempat Penelitian

Lokasi penelitian ini adalah di Laboratorium Sains dan Teknologi (SAINTEK) Kampus 1 Universitas Islam Negeri Sumatera Utara Medan, yang berlokasi di Jl IAIN no. 1, Gaharu, Kec. Medan Timur.

3.1.2. Waktu Dan Jadwal Pelaksanaan Penelitian

Waktu dan jadwal yang digunakan untuk penelitian tentang implementasi algoritma *greedy* menentukan rute terpendek antar klinik gigi berbasis *mobile* ini dilihat pada Tabel 3.1.

Tabel 3.1 Waktu dan Jadwal Penelitian

No.	Waktu	Jadwal Penelitian					
		Agustus	September	Oktober	November	Desember	Januari
1.	Perencanaan						
2.	Pengumpulan Data						
3.	Analisis Data dan Perancangan Sistem						
4.	Implementasi Sistem						
5.	Pengujian						

3.2. Bahan dan Alat Penelitian

Dalam pembuatan skripsi ini, alat dan bahan yang digunakan dalam penelitian tugas akhir ini diperlukan.

3.2.1. *Hardware* (Perangkat Keras)

Hardware (Perangkat Keras) yang diperlukan dalam penelitian ini adalah :

1. PC (*Personal Computer*) : *Processor core i3 Core 2,4 Ghz, Kapasitas Harddisk 500 GB, Besar Memory RAM 4 GB.*
2. *Android Device* : *Qualcomm Snapdragon 660, Memory Internal 128 GB, RAM 4 GB.*

3.2.1. *Software* (Perangkat Lunak)

Software (Perangkat Lunak) yang digunakan dalam penelitian ini adalah sebagai berikut.

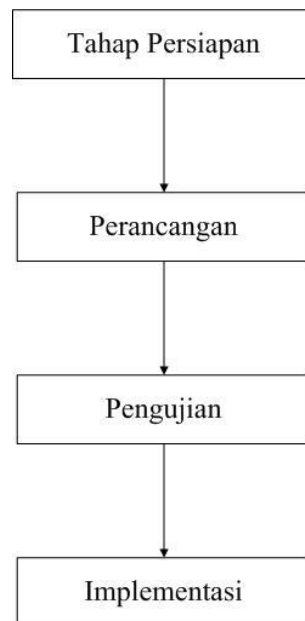
1. Sistem Operasi PC: *Windows 10, 64 bit*
2. Sistem Operasi *Android: Pie (9.0)*
3. *Google Maps*
4. *Android Cordova*
5. *Visual Studio Code*

3.3. Cara Kerja

Adapun cara kerja untuk penerapan algoritma *greedy* dalam menentukan rute terpendek adalah berikut ini :

3.3.1. Perencanaan

Pada bagian tahap ini, proses perencanaan kerja dalam penelitian ini berlangsung dalam beberapa tahap, yaitu tahap persiapan, desain, pengujian dan implementasi. Rencana kerja untuk penelitian ini diilustrasikan dalam diagram blok sebagai berikut.



Gambar 3.1 Tahap Perencanaan

- a. Pada tahap persiapan akan dilakukan serangkaian kegiatan sebelum memulai tahap pengumpulan data dan pengolahan data.
- b. Perancangan penelitian ini berisikan tentang pengembangan dari tahapan kebutuhan yang diubah kedalam diagram blok, *flowchart*, dan lainnya, agar peneliti memahami alur atau fungsi rancangan yang akan dibuat.
- c. Dalam fase pengujian ini, berbagai tes yang diterapkan pada langkah sebelumnya akan dilakukan dan akan menghasilkan hasil nyata.
- d. Tahap implementasi adalah penerapan semua proses yang dirancang dengan baik antara perangkat keras dan perangkat lunak yang telah digabungkan dengan kode sumber.

3.3.2. Teknik Pengumpulan Data

Pada titik ini, salah satu teknik pengumpulan data yang digunakan adalah menemukan referensi teoritis yang relevan dengan kasus atau masalah yang dihadapi (Nasution, 2018). Pencarian informasi dengan studi

literatur di beberapa jurnal ilmiah dan tugas akhir terkait dengan penerapan algoritma serakah untuk menentukan rute terpendek antara klinik gigi di Medan berdasarkan *mobile*, dapat memperkuat masalah serta landasan teori untuk melakukan penelitian.

3.3.3. Analisis Kebutuhan

Analisis kebutuhan Sistem adalah dimana beberapa kebutuhan bahan dalam sistem yang akan dipergunakan untuk menambah dan membantu jalan proses pembuatan suatu obyek (Nurul Fitria Insani, n.d.).

Analisis persyaratan sistem menjelaskan persyaratan sistem. Tujuannya adalah untuk benar-benar memahami kebutuhan sistem yang akan dibuat. Sistem yang dianalisis adalah sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan implementasi algoritma *greedy* berbasis *mobile*.

Saat merancang sistem ini, langkah-langkah yang harus diikuti terdiri dari memasukkan titik awal dan jarak klinik serta peta yang telah dianalisis. Kemudian akan diproses dan ditentukan rute terpendek menggunakan algoritma serakah, maka hasilnya akan ditampilkan. Hasil akhir algoritma akan diimplementasikan untuk mendapatkan hasil yang baik dari grafik yang telah dibentuk.

1. Aplikasi *mobile* yang akan dibangun harus dapat melakukan masukan/*input* seperti berikut:
 - a) *User* memilih lokasi klinik Gigi awal yang telah disediakan.
 - b) *User* mendapatkan informasi tempat klinik gigi terdekat yang akan dituju dari titik awal.
2. Aplikasi *mobile* yang akan dibangun harus mampu melakukan proses sebagai berikut :
 - a) Sistem mampu memproses hasil pencarian rute terpendek kepada *user* menuju tempat klinik gigi yang terdekat.

- b) Sistem mampu memproses rute pada *maps* sebagai sumber informasi pencarian rute terpendek bagi *user* untuk menuju tempat klinik gigi yang terdekat.
 - c) Sistem harus dapat menentukan panjang jarak serta waktu tempuh.
3. Aplikasi yang dibangun harus mampu melakukan keluaran/*output* sebagai berikut :
- a) Sistem menampilkan *maps* hasil proses pencarian rute terpendek menuju sebuah tempat klinik gigi terdekat dari titik awal di kota Medan.
 - b) Sistem menampilkan rute pada *maps* sebagai rute terpendek.

3.3.4. Perancangan

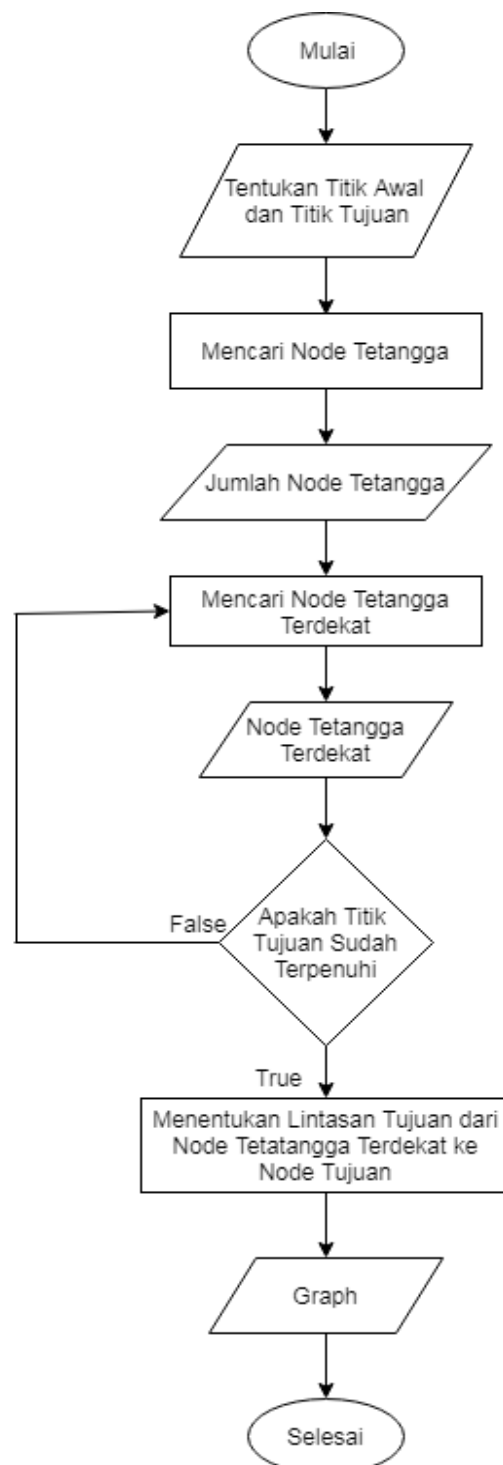
Perancangan sistem merupakan bagian yang penting dalam membuat suatu sistem ataupun aplikasi, perancangan sistem ini dimaksudkan untuk memberikan gambaran umum tentang aplikasi yang akan dibuat. Desain sistem adalah tahapan awal desain perangkat lunak. Perancangan ini dilakukan untuk mengetahui kondisi umum sistem.

A. *Flowchart*

Perancangan *Flowchart* sistem kerja berbasis *mobile* penentu rute terpendek menuju sebuah tempat klinik gigi di kota Medan dapat dilihat pada gambar *Flowchart* sebagai berikut :

1. *Flowchart* Algoritma *Greedy*

Rancangan *Flowchart* algoritma *greedy* untuk menentukan rute terpendek menuju sebuah tempat klinik gigi di kota Medan, berikut gambar *Flowchart* adalah :



Gambar 3.2 *Flowchart Algoritma Greedy*

B. Perancangan *Interface*

Desain ini dibuat sebagai dukungan untuk perantara pengguna dan aplikasi. Konsep struktural ini hanya dijelaskan secara umum dan garis besar. Menu-menu ini kemudian diterapkan sebagai tampilan pada halaman seluler. Antarmuka yang terkandung dalam sistem ini adalah halaman menu Beranda dan menu Pengujian.

1. Desain Tampilan *Interface Splash*

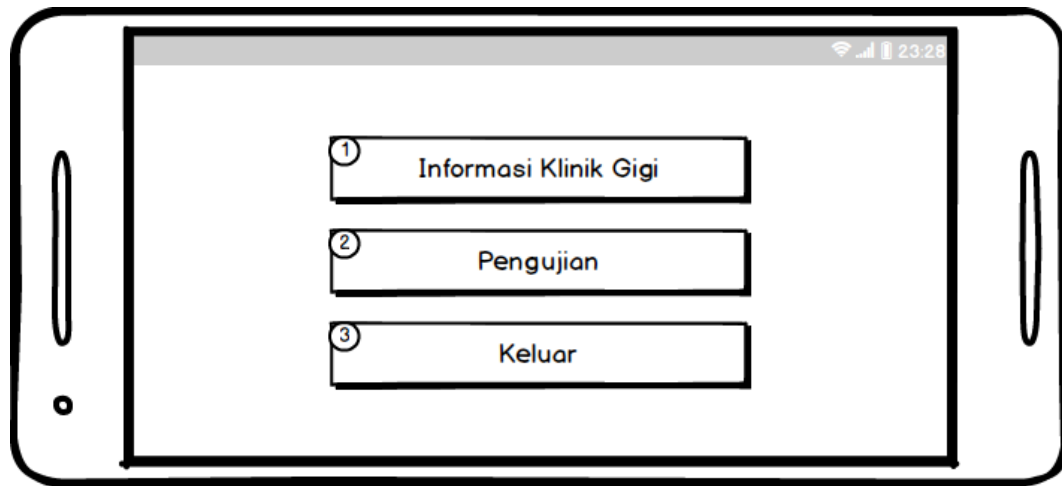
Rancangan *splash* ini merupakan tampilan yang terlihat sebelum masuk ke dalam menu utama aplikasi. Adapun rancangan tersebut dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3.3 Desain *Interface* Utama

2. Desain Tampilan Halaman Utama *User*

Halaman pengguna adalah halaman utama yang ditampilkan dari sistem aplikasi ketika aplikasi dijalankan. Tampilan desain halaman menu pengguna dapat dilihat pada gambar 3.4 sebagai berikut :



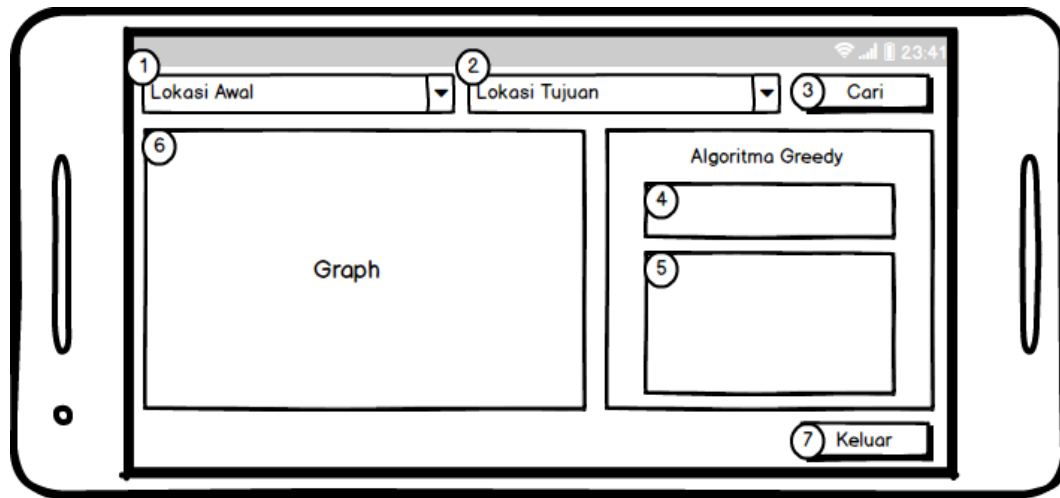
Gambar 3.4 Desain Halaman Utama *User*

Keterangan :

1. Button : Tombol Untuk Melihat Informasi Klinik Gigi.
2. Button : Tombol Untuk Pengujian.
3. Button : Tombol Untuk Keluar dari aplikasi.

3. Desain Halaman Menu Pengujian

Halaman Pengujian adalah halaman yang muncul ketika pengguna memilih menu Uji pada halaman utama pengguna. Pada halaman ini, pengguna dapat melihat peta untuk ditampilkan, pilih algoritma serakah. yang akan digunakan, dan lihat hasil total jarak, rute dan (jarak tempuh) dari algoritma. Desain halaman menu Uji terlihat pada gambar 3.5 adalah sebagai berikut :



Gambar 3.5 Desain Halaman Menu Pengujian

Keterangan :

1. *Combo Box* : Untuk menentukan *node* awal.
2. *Combo Box* : Untuk menentukan *node* akhir.
3. *Button* : Tombol untuk menampilkan hasil.
4. Label : Untuk menampilkan hasil pengujian menggunakan algoritma *Greedy*
hasil berupa : total jarak tempuh dan waktu jarak tempuh.
5. Untuk menampilkan rute algoritma *Greedy*.
6. Untuk menampilkan *maps*.
7. *Button* : Tombol untuk keluar dari menu pengujian.

3.3.5. Pengujian

Selama proses pengujian, akan membahas hasil pengujian aplikasi seluler yang dirancang dan diimplementasikan. Tes yang dilakukan untuk melihat apakah aplikasi seluler dapat bekerja sebagaimana mestinya, dan ini telah ditentukan dan dieksekusi sesuai dengan desain.

Untuk menjalankan aplikasi optimasi perjalanan antara klinik gigi di kota Medan dengan algoritma *Greedy*, sebelumnya perlu untuk mempersiapkan kebutuhan program yang akan diimplementasikan baik dari segi perangkat keras komputer dan perangkat lunak. Dalam penelitian ini, aplikasi mobile dibangun menggunakan bahasa pemrograman Java menggunakan perangkat lunak *Visual Studio Code*.

3.3.6. Penerapan

Pengaplikasian pencarian rute terpendek antara klinik gigi kota Medan menggunakan algoritma *Greedy* berbasis *mobile* ini nantinya akan diterapkan. Aplikasi *mobile* ini akan menghasilkan cara kerja algoritma *greedy* apakah algoritma ini adalah algoritma terbaik dalam menentukan rute terpendek antar klinik gigi di kota Medan.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Pembahasan

Beberapa langkah yang akan dibahas dalam penelitian ini adalah analisis data, representasi data, hasil analisis data dan desain berikut ini.

4.1.1. Analisis Data

Dalam langkah ini, analisis yang diperlukan untuk pembuatan aplikasi berbasis *android* untuk menentukan rute terpendek antar klinik gigi di kota Medan ini berupa analisis masalah, analisis persyaratan, dan analisis proses.

1. Analisis Masalah

Analisis adalah langkah yang bertujuan untuk mengidentifikasi masalah yang muncul saat membuat aplikasi, hal ini dilakukan agar selama proses desain, aplikasi tidak terjadi kesalahan sehingga sistem dapat berfungsi dengan baik dan diselesaikan tepat waktu. Dalam analisis sistem ini, sistem yang akan dianalisis yaitu :

- a. Bagaimanacara membuat aplikasi *android* untuk menentukan rute terpendek menuju sebuah klinik gigi di kota Medan.
- b. Bagaimana menampilkan sebuah *maps* klinik gigi yang ada pada aplikasi *android*.
- c. Bagaimana membangun aplikasi rute terpendek yang terpilih agar dapat tampil didalam *maps*.

Ada 10 nama klinik gigi di Medan dan titik-titik yang akan dilintasi yang telah dipilih dan akan diterapkan dalam maps sistem yang akan dilakukan, nama-nama klinik gigi dan titik-titik yang akan dilintasi terlihat maps tabel dibawah ini:

Tabel 4.1 Nama-nama Klinik Gigi yang Menjadi *Vertex*

No.	Nama-nama Klinik Gigi	Node	Alamat
1.	Osaka Dental Care	A	Osaka dental care Komplek Millenium Square, Jl. Kapten Muslim No.9, Helvetia Tim., Kec. Medan Helvetia, Kota Medan, Sumatera Utara 20123
2.	Klinik Gigi ProDental Plaza	B	Ruko Nusa Dua, Jl. Gatot Subroto, Sei Sikambing C. II, Kec. Medan Helvetia, Kota Medan, Sumatera Utara 20122
3.	Dokter Gigi Jenny	C	Jl. Sunggal Jl. Komp. Graha Sunggal No.26, Sunggal, Kec. Medan Sunggal, Kota Medan, Sumatera Utara 20128
4.	Klinik Gigi Drg. Lilys Mei	D	Jl. Ring Road Jl. Gagak Hitam No.50, Sunggal, Kec. Medan Sunggal, Kota Medan, Sumatera Utara 20122
5.	Klinik Pratama Senyum Jaya Abadi	E	Jl. Setia Budi No.117, Tj. Sari, Kec. Medan Selayang, Kota Medan, Sumatera Utara 20132
6.	Medan Dental Specialist Center	F	Jl. Kejaksaan No.8A, Petisah Tengah, Kec. Medan Petisah, Kota Medan, Sumatera Utara 20112
7.	Puri Dental Clinic	G	Jl. Denai Ruko Denai Boulevard No.1, Tegal Sari Mandala III, Kec. Medan Denai, Kota Medan, Sumatera Utara 20371
8.	Praktek drg. Imelda Grace	H	Jl. Tampilan Nauli, Ps. Merah Bar.,

	Marpaung		Kec. Medan Kota, Kota Medan, Sumatera Utara 20214
9.	Praktek Dokter Gigi drg Elvira Rosa Sinabutar M.Kes	I	Jl. Durung No.196, Sidorejo, Kec. Medan Tembung, Kota Medan, Sumatera Utara 20371
10.	Dokter Gigi Spesialis Ortodonti	J	Jl. Prof. HM. Yamin Sh No.273, Sei Kera Hilir I, Kec. Medan Perjuangan, Kota Medan, Sumatera Utara 20222

Tabel 4.2 Nama-nama Titik yang Dilalui yang Menjadi *Vertex*

No	Nama Titik yang Dilalui	Node	Alamat
1.	Helmi Clothing Distro Medan	A1	Jl. Amal Luhur No.81B, Dwi Kora, Kec. Medan Helvetia, Kota Medan, Sumatera Utara 20123
2.	Komplek SDH	A2	Jl. Tanah Tinggi, Sunggal, Kec. Medan Sunggal, Kota Medan, Sumatera Utara 20128
3.	Alfamidi Setia Budi 2	A3	Jl. Setia Budi Ps. II, Tj. Sari, Kec. Medan Selayang, Kota Medan, Sumatera Utara 20132
4.	Rumah Makan Pondok Gurih	A4	Jl. Brigjend Katamso No.33A, A U R, Kec. Medan Maimun, Kota Medan, Sumatera Utara 20159
5.	Perdana Bordir	A5	Jl. Denai No.12 B, Tegal Sari Mandala III, Kec. Medan Denai, Kota Medan, Sumatera Utara 20227
6.	PIONEER Ponsel	A6	Jln. W. Iskandar No. 97 (Simp. Perjuangan), Sidorejo, Kec. Medan

			Tembung, Kota Medan, Sumatera Utara 20222
--	--	--	---

2. Analisis Persyaratan

Analisis persyaratan dibagi menjadi 2 bagian yaitu analisis persyaratan fungsional dan analisis persyaratan non-fungsional.

2.1. Persyaratan Fungsional

Persyaratan fungsional di sini untuk menggambarkan aplikasi yang disediakan. Aplikasi ini melakukan perhitungan, menghasilkan rute dan waktu penelitian untuk klinik gigi terdekat di kota Medan dengan menerapkan algoritma *Greedy*, ada persyaratan fungsional tertentu sebagai berikut :

- a. Aplikasi ini menggunakan peta klinik gigi Medan yang telah dipilih dan dapat ditampilkan pada aplikasi sesuai dengan representasi peta yang ditentukan.
- b. Aplikasi yang dibangun menggunakan simpul dan tepi pada peta dan dapat ditampilkan pada aplikasi dengan nama dan rute untuk pergi ke klinik gigi di setiap sisi (tepi) yang terhubung ke simpul (*vertex*).
- c. Aplikasi ini menghasilkan pencarian rute terpendek dari *maps* menggunakan algoritma *Greedy*.

2.2. Persyaratan Non-Fungsional

Persyaratan non-fungsional adalah persyaratan yang berisi kinerja operasional dan kinerja aplikasi. Ada beberapa persyaratan non-fungsional yang harus dipenuhi yaitu :

a. *User Friendly* (Mudah Digunakan)

Untuk membangun aplikasi ini harus mudah digunakan (*user-friendly*), yang berarti bahwa sistem ini harus mudah digunakan oleh pengguna dengan tampilan yang sederhana dan mudah dimengerti.

b. Kinerja

Aplikasi *mobile* yang akan dibangun harus dapat menunjukkan hasil jalur terpendek dan rute Klinik Gigi dari algoritma yang diterapkan, yaitu algoritma *Greedy*.

c. Manajemen Kualitas

Aplikasi *mobile* yang akan dibangun harus berkualitas baik, mampu menampilkan peta, menampilkan jarak, waktu, dan hasil rute terpendek ke klinik gigi.

3. Analisis Proses

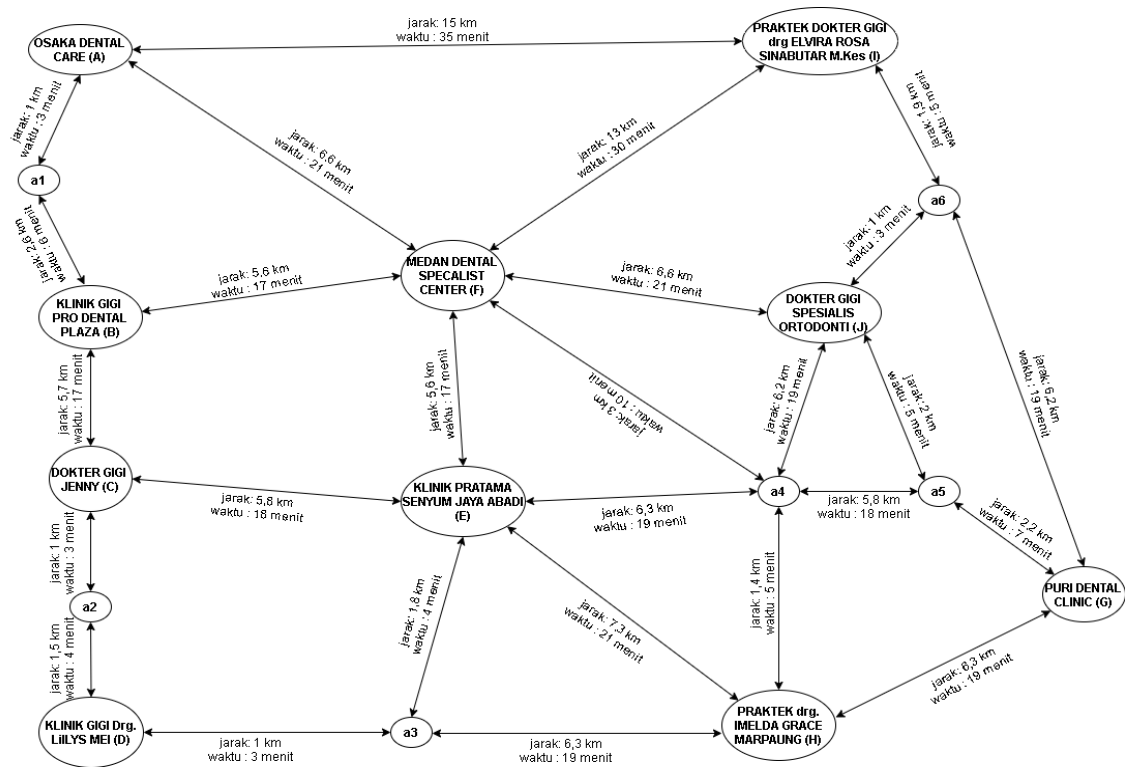
Aplikasi yang akan dibuat bahasa pemrogramannya adalah *Java* sebagai *software Androidnya* dan *PHP* untuk *Back End* nya. Algoritma yang digunakan untuk menentukan rute terpendek antar Klinik Gigi di Kota Medan adalah Algoritma *Greedy*.

4.1.2. Representasi Data

1. Hasil PengujianProses Algoritma *Greedy*

Pada algoritma *Greedy* langkah awal yang harus ditentukan adalah menentukan titik awal. Setelah dipilih titik awal yang berupa klinik gigi, kemudian klik tombol “Cari Klinik Terdekat” dimana tombol akan berfungsi untuk menampilkan hasil klinik terdekat atau rute terpendek dari titik awal, dan hasil pencarian akan menampilkan rute berupa *maps* serta menampilkan jarak, waktu tempuh, dan titik yang dilalui yang dibutuhkan algoritma tersebut dalam melakukan proses pencarian, jika ingin mengulang pencarian klinik gigi terdekat klik tombol “Pengujian Ulang” maka *user* dapat kembali memilih titik awal.

Berikut tampilan *graph* klinik gigi di kota Medan :



Gambar 4.1 Tampilan *Graph* Klinik Gigi di Kota Medan

2. Perhitungan Manual Algoritma *Greedy*

Tahap selanjutnya adalah melakukan proses perhitungan dengan algoritma *greedy*. Perhitungan akan dilakukan dari titik awal yang akan dipilih oleh *user* yaitu dari titik awal A sampai titik J. Maka dilakukan perhitungan dengan algoritma *greedy* adalah sebagai berikut:

Algoritma *greedy* untuk menemukan jalur terpendek dapat dirumuskan adalah sebagai berikut:

1. Periksa semua tepi yang berbatasan langsung dengan simpul a. Pilih tepi dengan bobot terkecil. Tepi ini pertama menjadi jalur terpendek, sebut saja L(1).
2. Hitung jalur terpendek keduanya dengan cara sebagai berikut:
 - a. Hitung: $d(i) = \text{panjang } L(1) + \text{bobot tepi dari simpul terakhir } L(1) \text{ ke simpul } i \text{ yang lain.}$

- b. Pilih $d(i)$ yang terkecil. Bandingkan $d(i)$ dengan bobot tepi (a, i) . Jika bobot tepi (a, i) kurang dari $d(i)$, maka $L(2) = L(1) \cup (\text{tepi simpul akhir } L(i) \text{ di simpul } i)$.
3. Demikian pula, ulangi langkah 2 untuk menentukan jalur terpendek berikutnya.

Contoh :

Proses perhitungan 1

Titik Awal : Klinik Osaka Dental Care (A)

Hasil Klinik Terdekat : Klinik Gigi ProDental Plaza (B)

Himpunan Kandidat : Klinik Osaka Dental Care (A) – Klinik Gigi ProDental Plaza (B)

Himpunan Solusi : 1. *Node A – a1 – B*

2. *Node A – F*

3. *Node A – I*

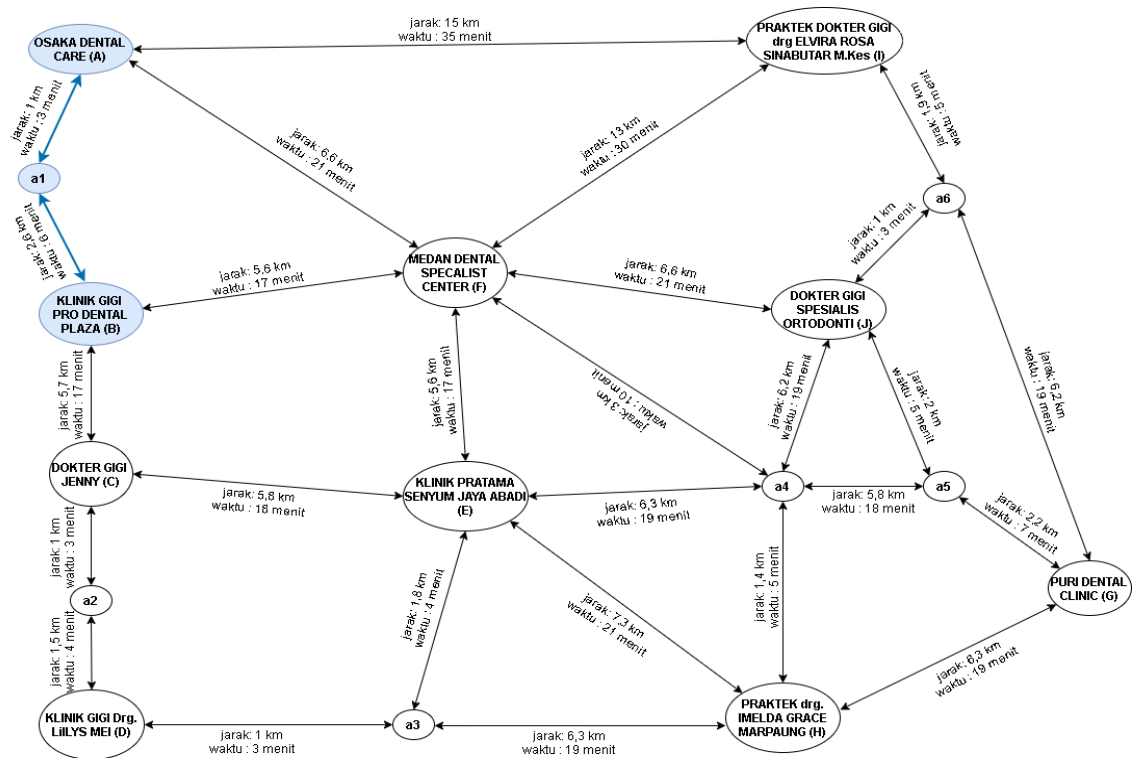
Fungsi Seleksi : *Node A – a1 – B*

Fungsi Kelayakan : *Node A – a1 – B*

Fungsi Objektif : Nilai minimum adalah *node A – a1 – B*

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1 + 2,7 = 3,7$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* adalah melalui node *A – a1 – B* dengan total jarak 3,7Km dengan waktu 9 menit ditunjukkan pada gambar 4.2 berikut :



Gambar 4.2 GraphRute dari Vertex A ke Vertex B

Proses perhitungan 2

Titik Awal : Klinik Gigi ProDental Plaza (B)

Hasil Klinik Terdekat : Klinik Osaka Dental Care (A)

Himpunan Kandidat : Klinik Gigi ProDental Plaza (B) – Klinik Osaka Dental Care (A)

Himpunan Solusi : 1. Node B–a1–A

2. Node B–F

3. Node B–C

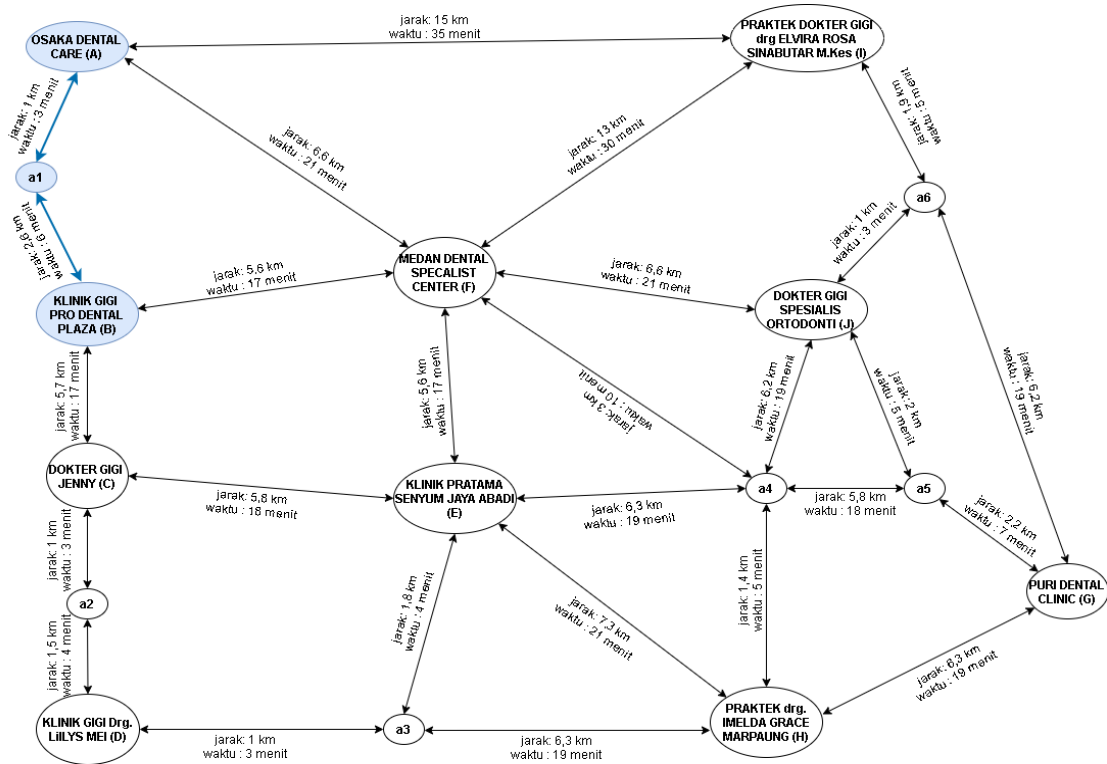
Fungsi Seleksi : Node B–a1–A

Fungsi Kelayakan : Node B–a1–A

Fungsi Objektif : Nilai minimum adalah nodeB–a1–A

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 2,6 + 1 = 3,7$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node B—a1—A dengan total jarak 3,7 Km dengan waktu 9 menit ditunjukkan pada gambar 4.3 berikut :



Gambar 4.3 *Graph* Rute dari Vertex B ke Vertex A

Proses perhitungan 3

Titik Awal : Dokter Gigi Jenny (C)

Hasil Klinik Terdekat : Klinik Gigi Drg. Lilys Mei(D)

Himpunan Kandidat : Dokter Gigi Jenny (C) – Klinik Gigi Drg. Lilys Mei (D)

Himpunan Solusi : 1. *Node* C—a2—D

2. *Node* C—B

3. *Node* C—E

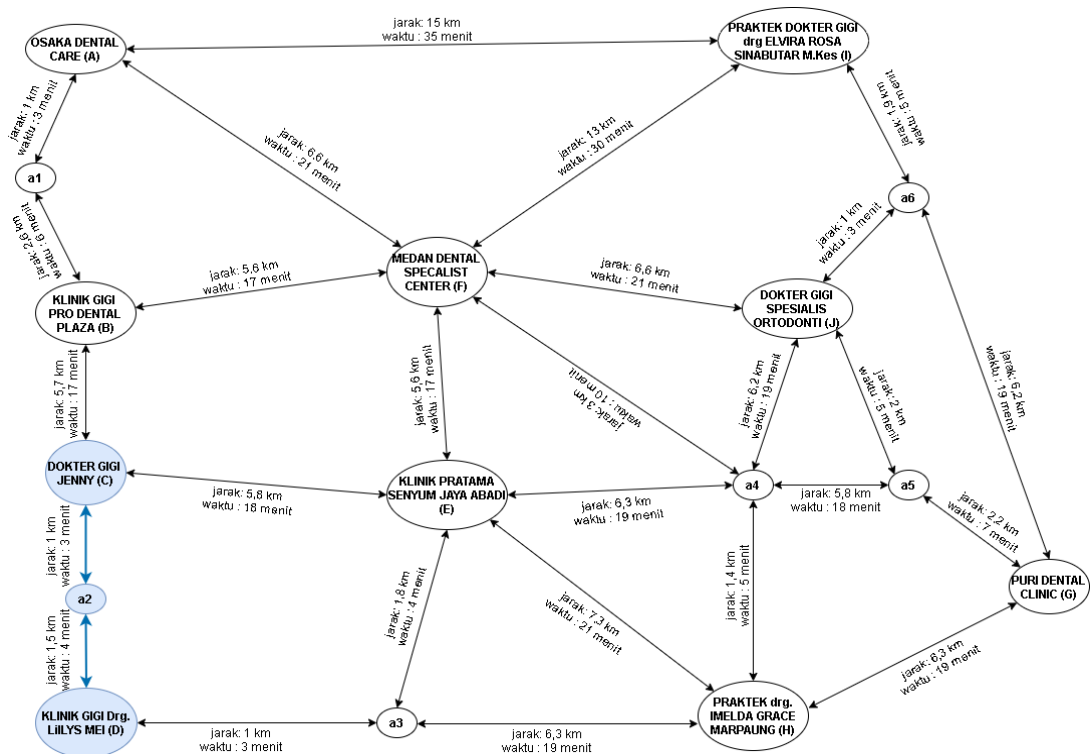
Fungsi Seleksi : *Node* C—a2—D

Fungsi Kelayakan : *Node C–a2–D*

Fungsi Objektif : Nilai minimum adalah *nodeC–a2–D*

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1 + 1,5 = 2,5$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node *C–a2–D* dengan total jarak 2,5Km dengan waktu 7 menit dapat dilihat pada gambar 4.4 berikut :



Gambar 4.4 Rute dari *Vertex C* ke *Vertex D*

Proses perhitungan 4

Titik Awal : Klinik Gigi Drg. Lilys Mei (D)

Hasil Klinik Terdekat : Dokter Gigi Jenny (C)

Himpunan Kandidat : Klinik Gigi Drg. Lilys Mei (D) – Dokter Gigi Jenny (C)

Himpunan Solusi : 1. *Node D–a2–C*

2. $NodeD-a3-E$

3. $NodeD-a3-H$

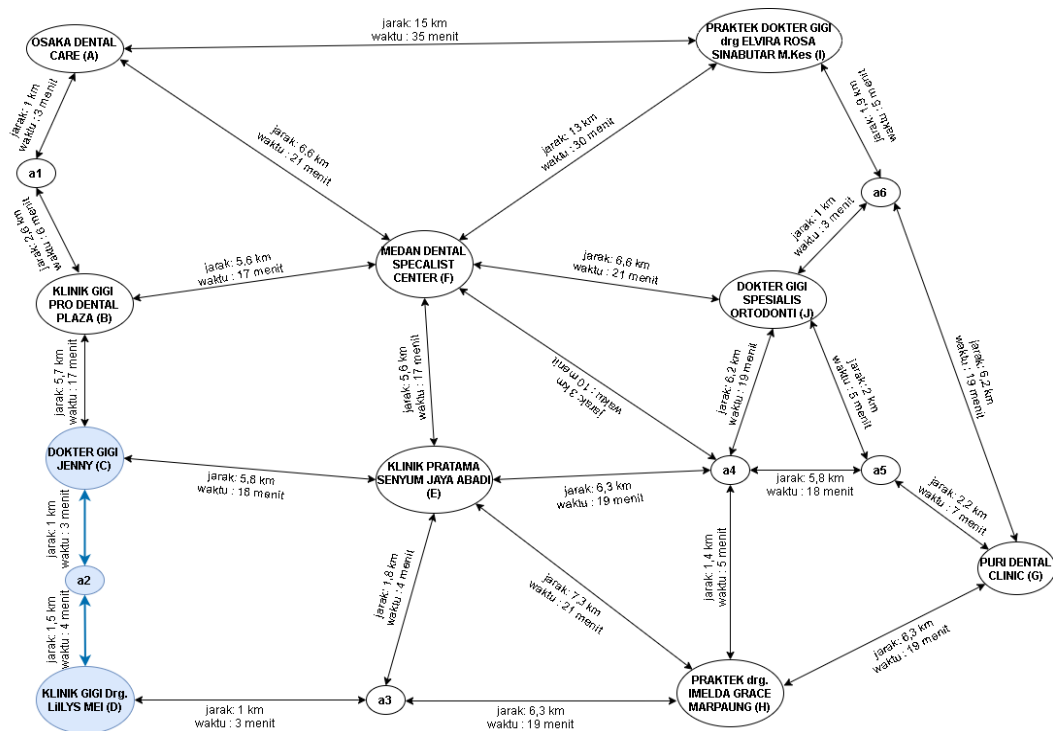
Fungsi Seleksi : $NodeD-a2-C$

Fungsi Kelayakan : $NodeD-a2-C$

Fungsi Objektif : Nilai minimum adalah $nodeD-a2-C$

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1,5 + 1 = 2,5$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node $D-a2-C$ dengan total jarak 2,5 Km dengan waktu 7 menit ditunjukkan pada gambar 4.5 berikut :



Gambar 4.5 *GraphRute* dari Vertex D ke Vertex C

Proses perhitungan 5

Titik Awal : Klinik Pratama Senyum Jaya Abadi (E)

Hasil Klinik Terdekat : Klinik Gigi Drg. Lilys Mei (D)

Himpunan Kandidat : Klinik Pratama Senyum Jaya Abadi (E) – Klinik Gigi Drg. Lilys Mei (D)

Himpunan Solusi : 1. *Node* E–a3–D
 2. *Node* E–F
 3. *Node* E–C
 4. *Node* E–H

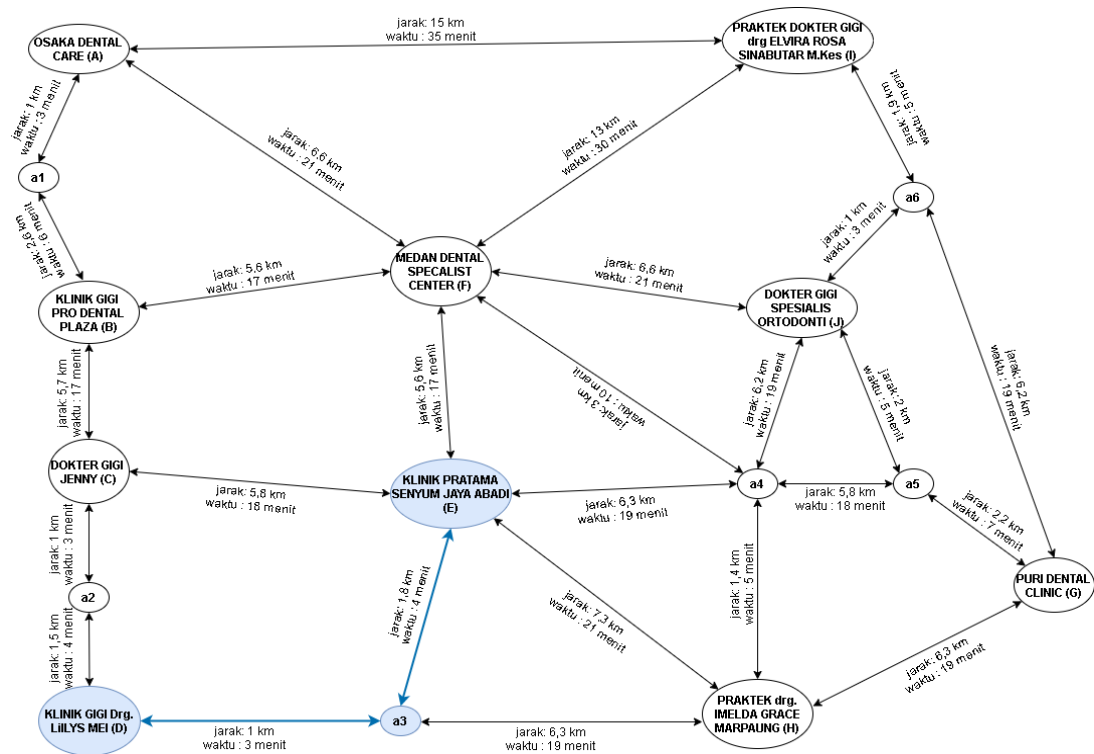
Fungsi Seleksi : *Node* E–a3–D

Fungsi Kelayakan : *Node* E–a3–D

Fungsi Objektif : Nilai minimum adalah *node* E–a3–D

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1 + 1,8 = 2,8$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node E–a3–D dengan total jarak 2,8 Km dengan waktu 7 menit ditunjukkan pada gambar 4.6 berikut :



Gambar 4.6 Graph Rute dari Vertex E ke Vertex D

Proses perhitungan 6

Titik Awal : Medan Dental Specialist Center (F)

Hasil Klinik Terdekat : Praktek drg. Imelda Grace Marpaung Mei (H)

Himpunan Kandidat : Medan Dental Specialist Center (F) – Praktek drg. Imelda Grace Marpaung Mei (H)

Himpunan Solusi : 1. Node F–a4–H

2. Node F–A

3. Node F–B

4. Node F–E

5. Node F–J

6. Node F–I

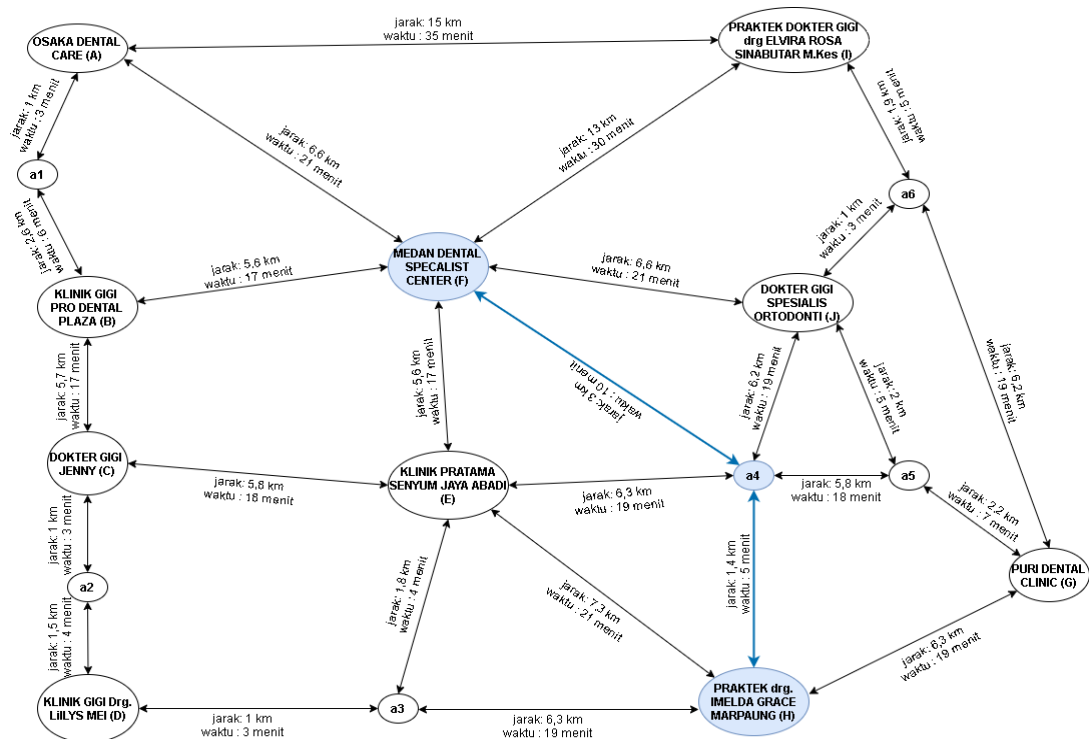
Fungsi Seleksi : Node F–a4–H

Fungsi Kelayakan : *Node F–a4–H*

Fungsi Objektif : Nilai minimum adalah *nodeF–a4–H*

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 3 + 1,4 = 4,4$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node *F–a4–H* dengan total jarak 4,4 Km dengan waktu 15 menit ditunjukkan pada gambar 4.7 berikut :



Gambar 4.7 *Graph* Rute dari *Vertex F* ke *Vertex H*

Proses perhitungan 7

Titik Awal : Puri Dental Clinic (G)

Hasil Klinik Terdekat : Dokter Gigi Spesialis Ortodonti (J)

Himpunan Kandidat : Puri Dental Clinic (G)– Dokter Gigi Spesialis Ortodonti (J)

Himpunan Solusi : 1. *Node G–a5–J*

2. Node G–H

3. Node G–a6–I

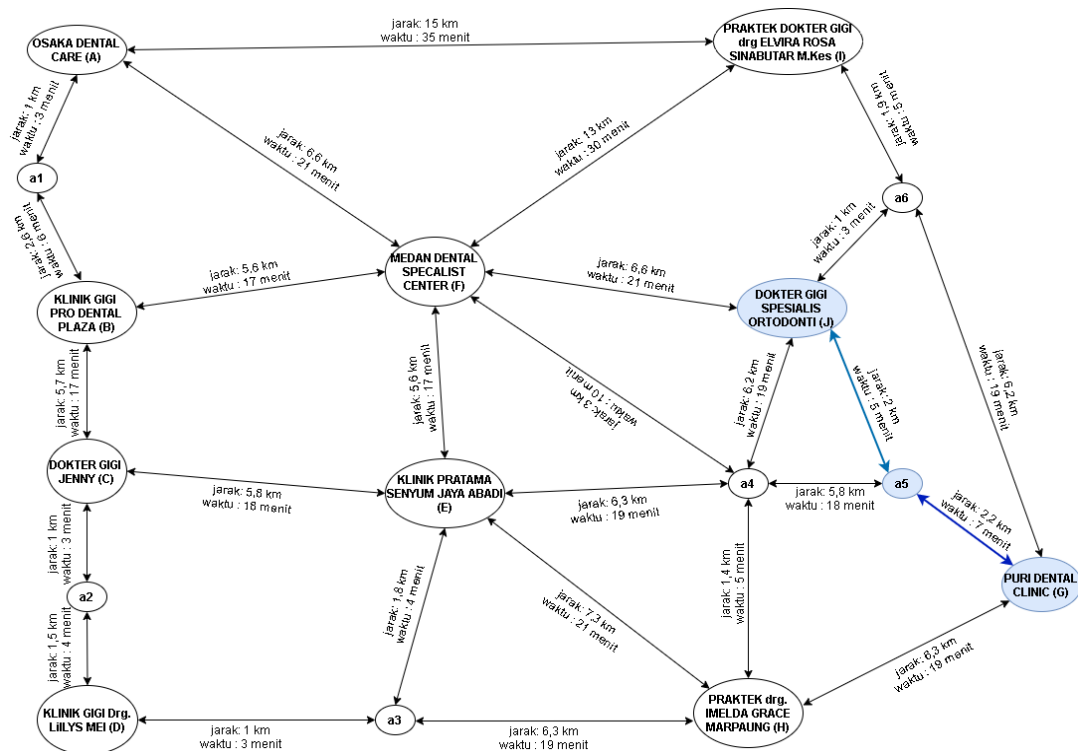
Fungsi Seleksi : Node G–a5–J

Fungsi Kelayakan : Node G–a5–J

Fungsi Objektif : Nilai minimum adalah node G–a5–J

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 3 + 1,2 = 4,2$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node G–a5–J dengan total jarak 4,2 Km dengan waktu 12 menit ditunjukkan pada gambar 4.8 berikut :



Gambar 4.8 GraphRute dari Vertex G ke Vertex J

Proses perhitungan 8

Titik Awal : Praktek drg. Imelda Grace Marpaung (H)

Hasil Klinik Terdekat : Medan Dental Specialist Center (F)

Himpunan Kandidat : Praktek drg. Imelda Grace Marpaung (H)— Medan Dental Specialist Center (F)

Himpunan Solusi : 1. *Node H—a4—F*
 2. *Node H—E*
 3. *Node H—a3—D*
 4. *Node H—G*

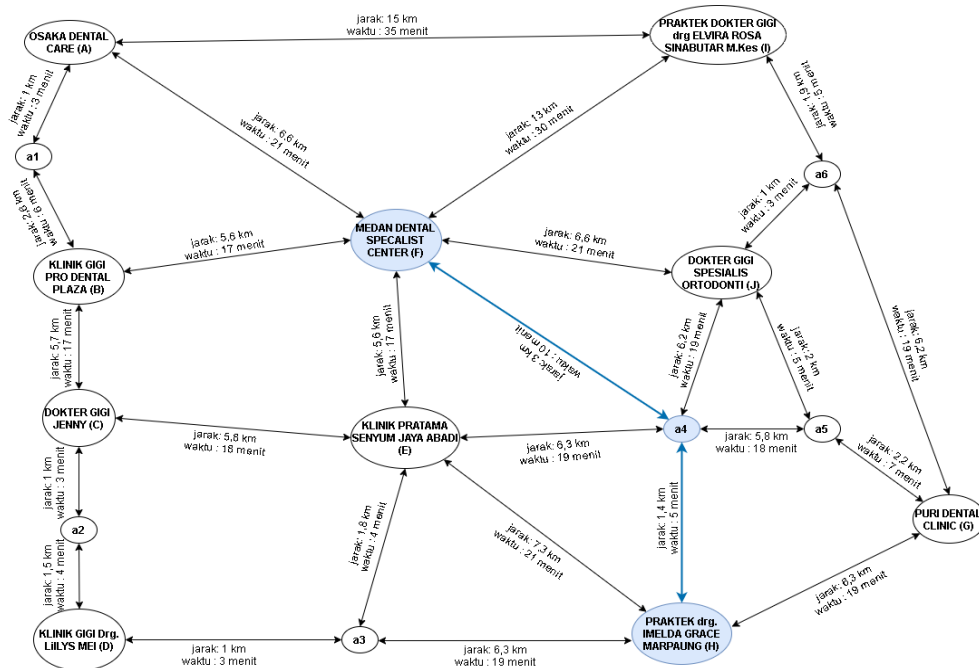
Fungsi Seleksi : *Node H—a4—F*

Fungsi Kelayakan : *Node H—a4—F*

Fungsi Objektif : Nilai minimum adalah *node H—a4—F*

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1,4 + 3 = 4,4$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node H—a4—F dengan total jarak 4,4 Km dengan waktu 15 menit ditunjukkan pada gambar 4.9 berikut :



Gambar 4.9 Graph Rute dari Vertex H ke Vertex F

Proses perhitungan 9

Titik Awal : Praktek drg. Imelda Grace Marpaung (I)

Hasil Klinik Terdekat : Medan Dental Specialist Center (J)

Himpunan Kandidat : Praktek drg. Imelda Grace Marpaung (I) – Medan Dental Specialist Center (J)

Himpunan Solusi : 1. *Node I–a6–J*
 2. *Node I–F*
 3. *Node I–A*
 4. *Node I–a6–G*

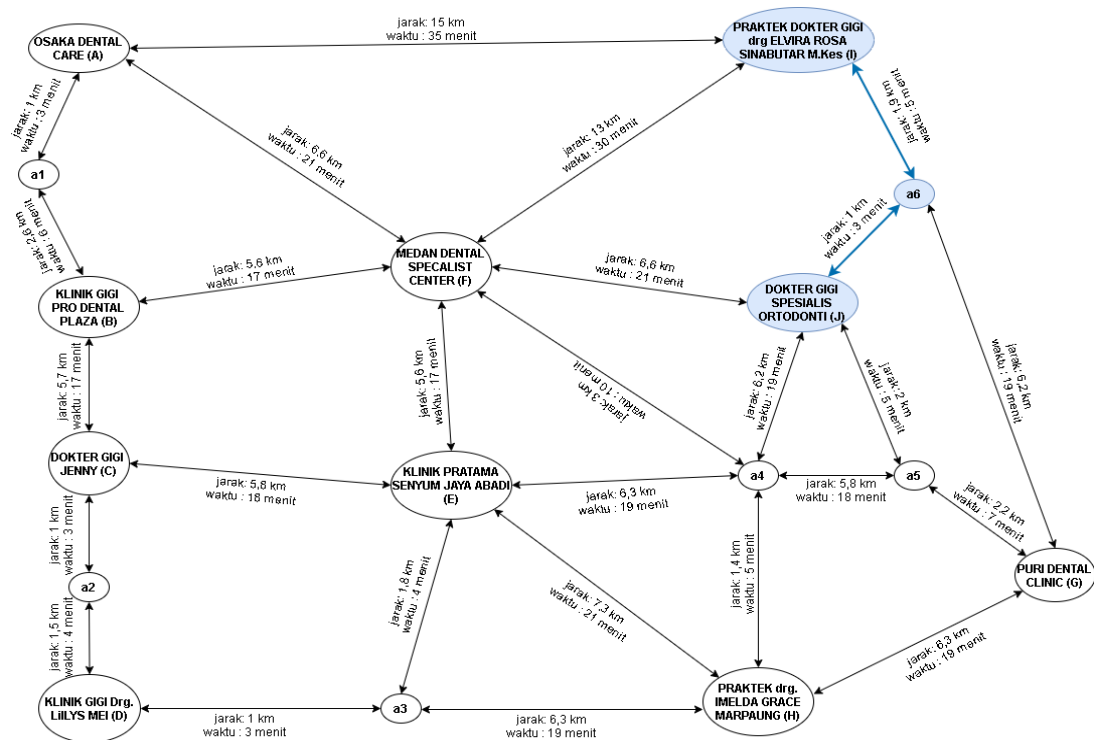
Fungsi Seleksi : *Node I–a6–J*

Fungsi Kelayakan : *Node I–a6–J*

Fungsi Objektif : Nilai minimum adalah *node I–a6–J*

Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1 + 1,9 = 2,9$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node *I–a6–J* dengan total jarak 2,9 Km dengan waktu 8 menit ditunjukkan pada gambar 4.10 berikut :



Gambar 4.10 Graph Rute dari Vertex I ke Vertex J

Proses perhitungan 10

Titik Awal : Medan Dental Specialist Center (J)

Hasil Klinik Terdekat : Praktek drg. Imelda Grace Marpaung (I)

Himpunan Kandidat : Medan Dental Specialist Center (J)– Praktek drg. Imelda Grace Marpaung (I)

Himpunan Solusi : 1. Node J–a6–I

2. Node J–F

3. Node J–a4–E

4. Node J–a4–H

4. Node J–a5–G

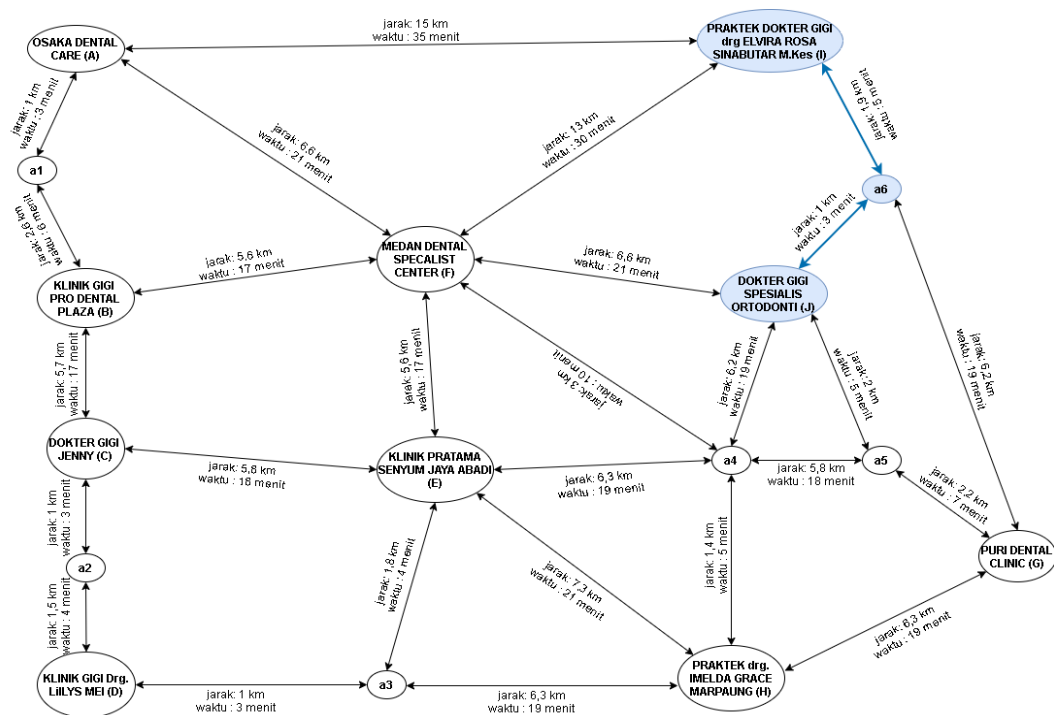
Fungsi Seleksi : Node J–a6–I

Fungsi Kelayakan : Node J–a6–I

Fungsi Objektif : Nilai minimum adalah $node J - a6 - I$

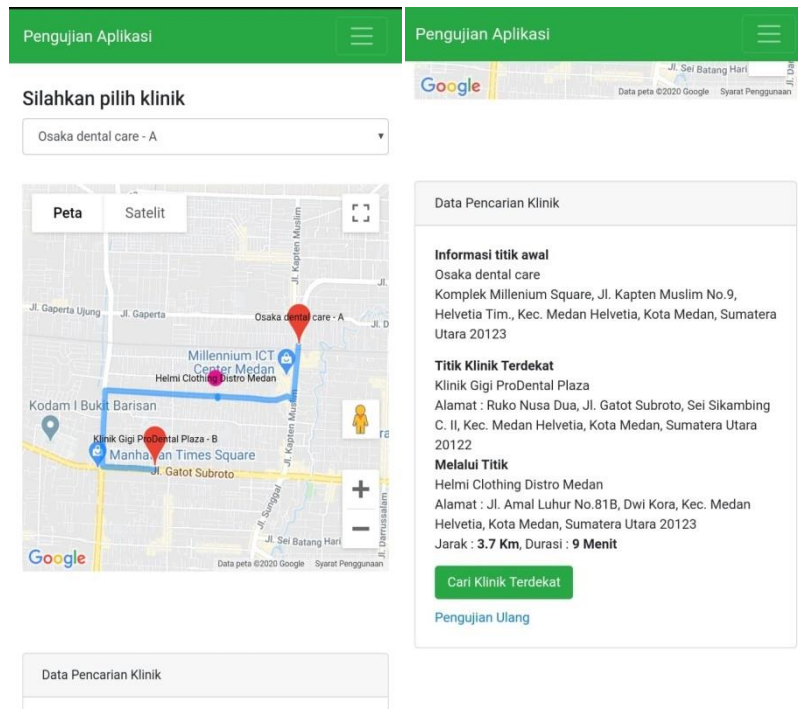
Penyelesaian :

Hitung $L(2) = \text{panjang } L(1) + d(i) = 1,9 + 1 = 2,9$. Hasil yang diperoleh dari perhitungan diatas untuk menentukan rute terpendek menggunakan algoritma *Greedy* ialah melalui node $J - a6 - I$ dengan total jarak 2,9 Km dengan waktu 8 menit ditunjukkan pada gambar 4.11 berikut :

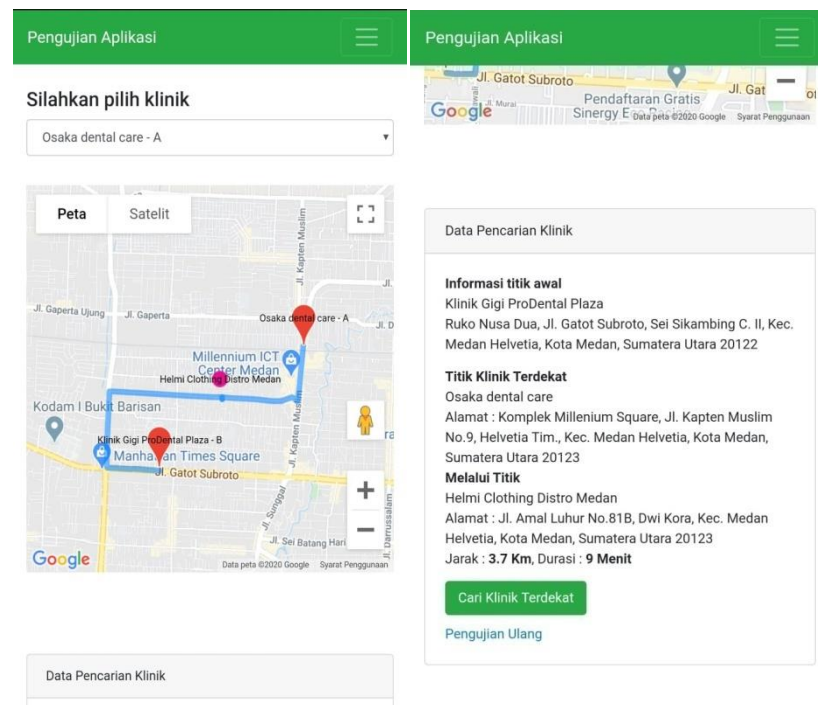


Gambar 4.11 Graph Rute dari Vertex J ke Vertex I

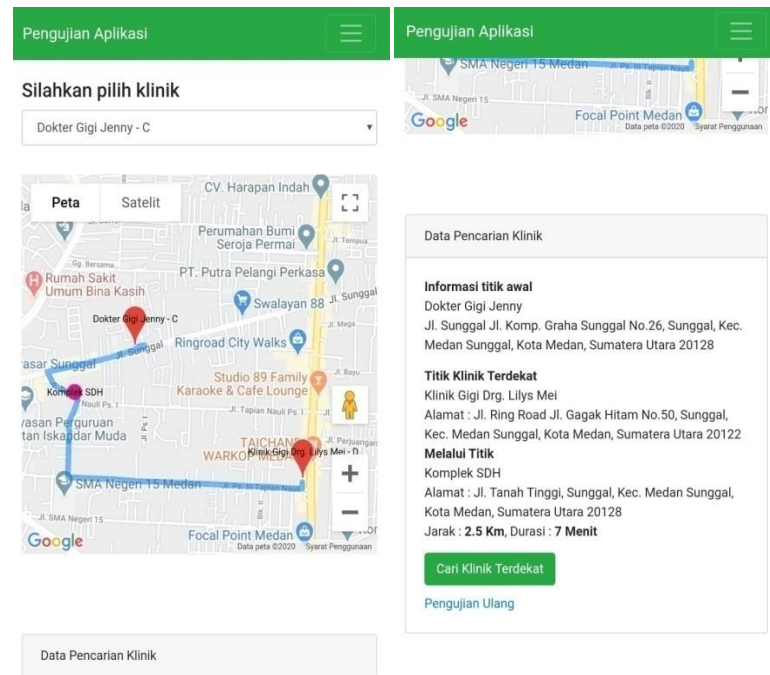
Setelah melakukan proses perhitungan manual menggunakan algoritma *greedy* dengan hasil *graph*, maka selanjutnya akan ditampilkan pengujian aplikasi yang dilakukan dari titik awal yang akan dipilih oleh *user* yaitu dari titik A sampai titik J, dengan hasil berupa *maps*. Pengujian pada aplikasi berupa *maps* ditunjukkan pada gambar berikut :



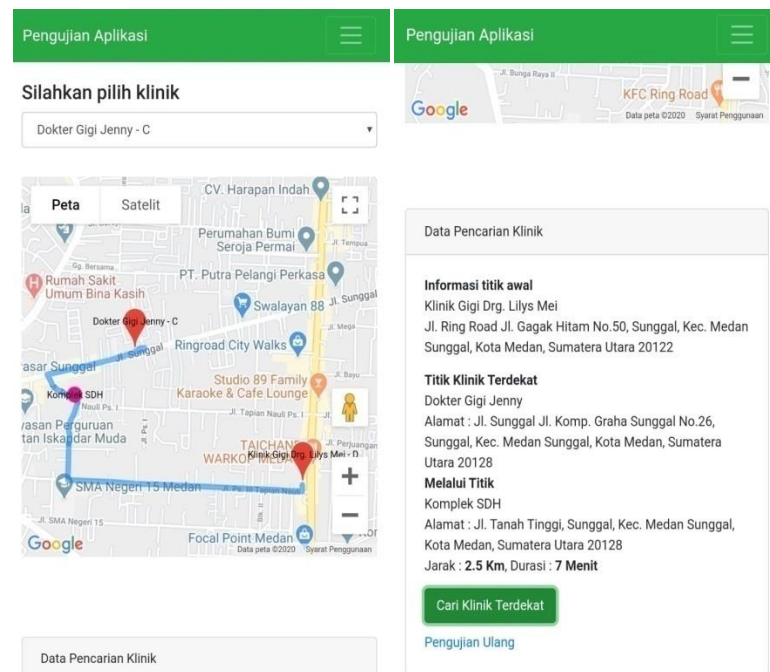
Gambar 4.12 Maps Rute dari Vertex A ke Vertex B



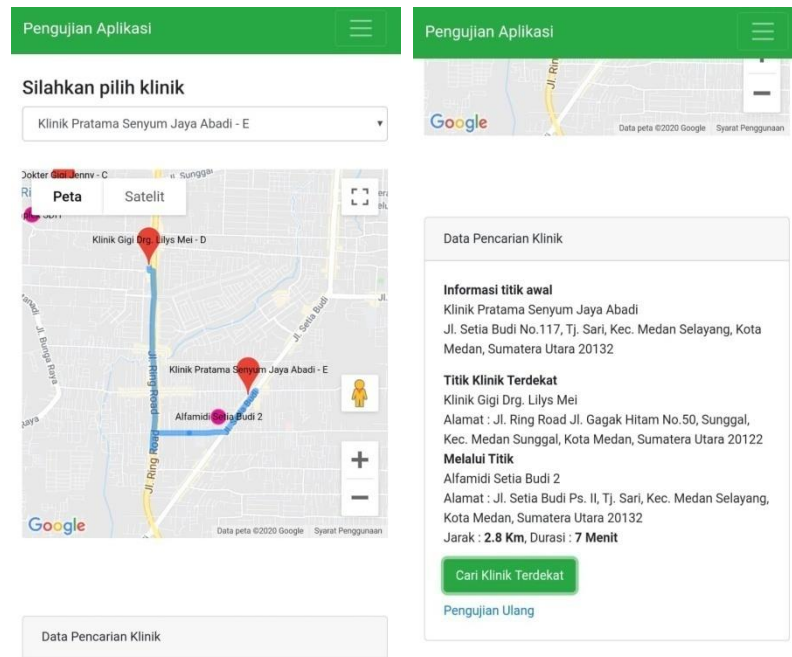
Gambar 4.13 Maps Rute dari VertexB ke VertexA



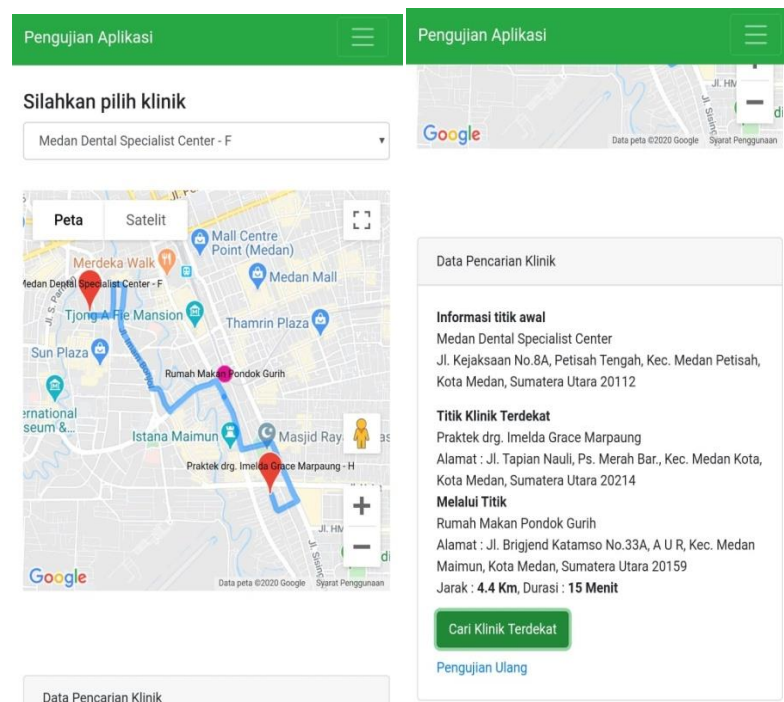
Gambar 4.14 Maps Rute dari Vertex C ke Vertex D



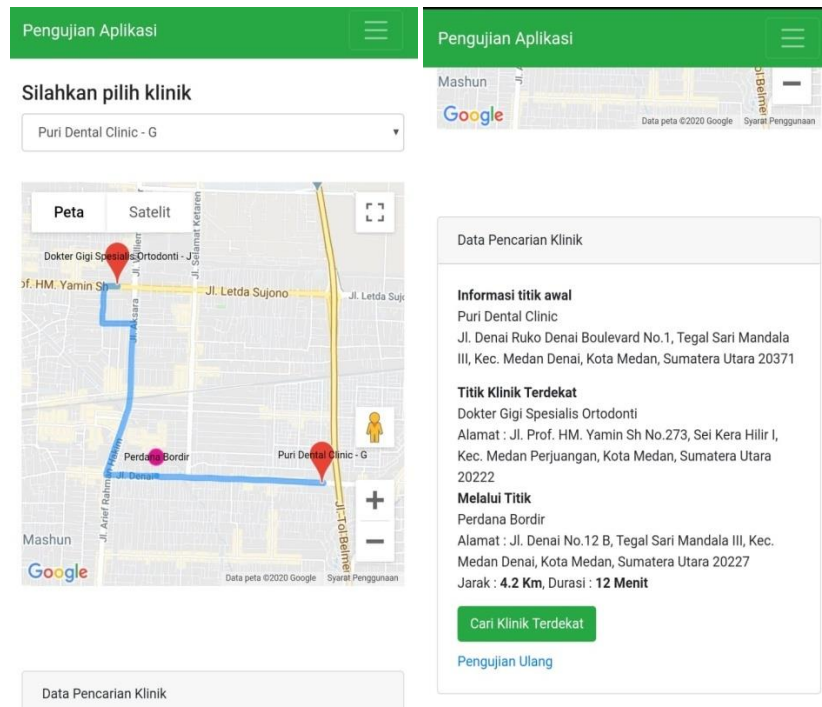
Gambar 4.15 Maps Rute dari Vertex D ke Vertex C



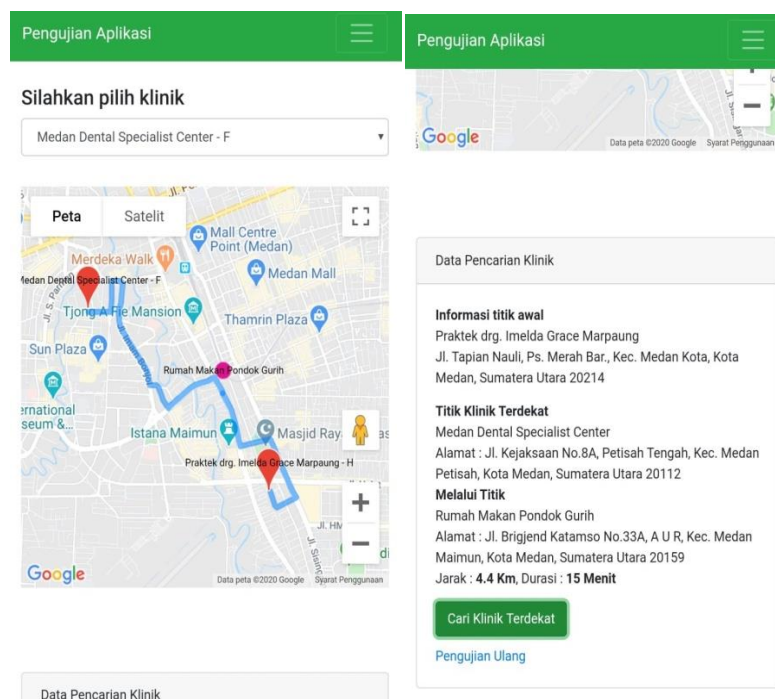
Gambar 4.16 Maps dari Vertex E ke Vertex D



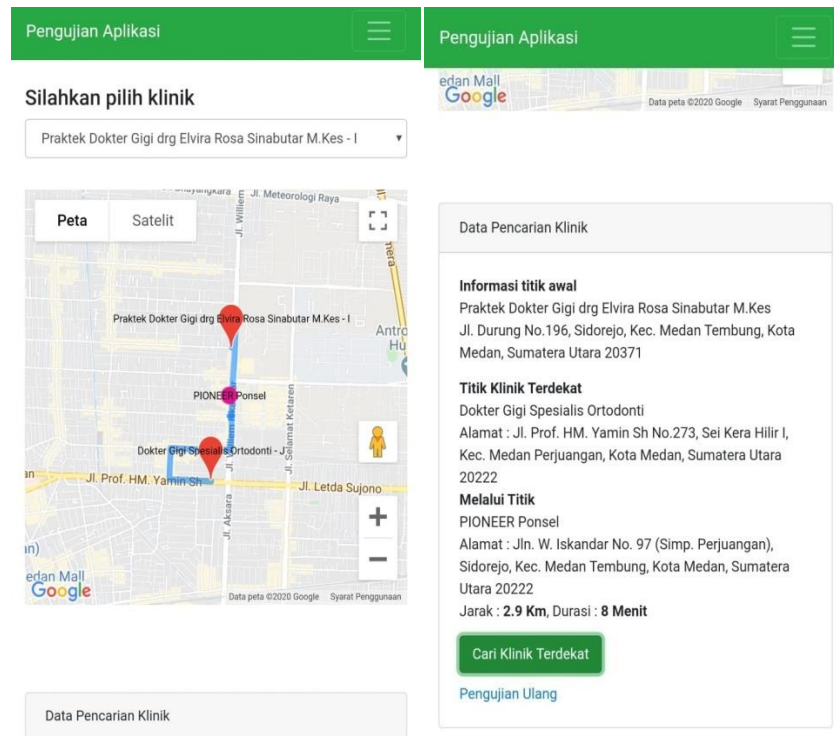
Gambar 4.17 Maps dari Vertex F ke Vertex H



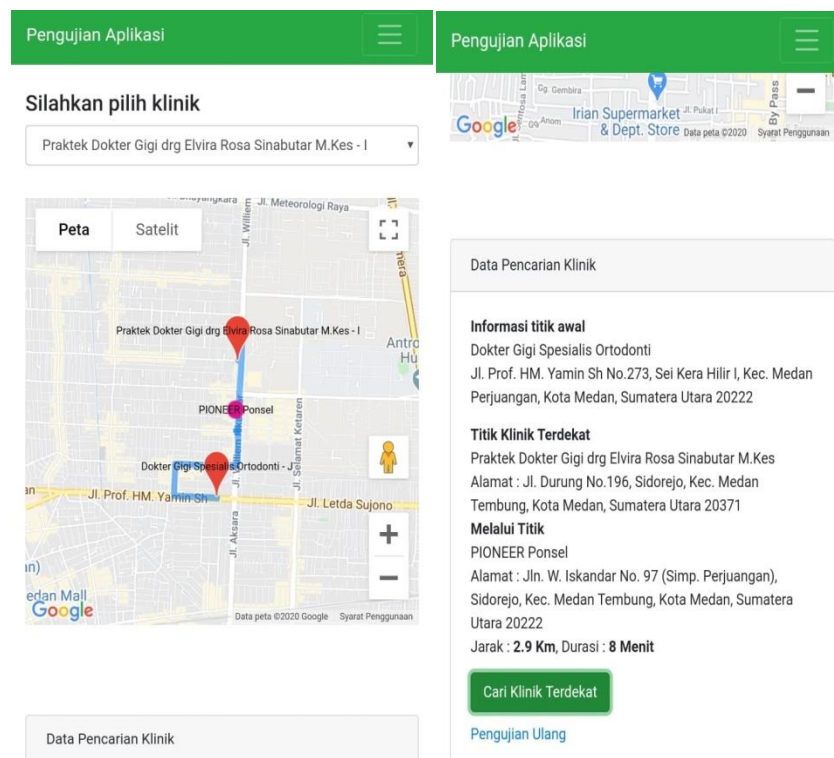
Gambar 4.18 Maps dari Vertex G ke Vertex J



Gambar 4.19 Maps dari Vertex H ke Vertex F



Gambar 4.20 Maps dari Vertex I ke Vertex J



Gambar 4.21 Maps dari Vertex J ke Vertex

3. Hasil Pengujian Jarak dan Waktu Algoritma Greedy

Proses pengujian dilakukan secara manual dengan bahasa pemrograman *Java*. Hasil uji coba yang dilakukan pada klinik gigi di kota Medan yang digunakan sebagai titik awal menuju tujuan menggunakan algoritma *Greedy* dapat dilihat pada Tabel 4.3 sebagai berikut :

Tabel 4.3 Hasil Rute Terpendek Algoritma Greedy

No	Rute	Node	Total Jarak	Total Waktu
1.	A-B	A-a1-B	$L(1) = A(a1) + B(a1)$ $= 1 + 2,6$ $= 3,7 \text{ Km}$	9 Menit
2.	B-A	B-a1-A	$L(1) = B(a1) + A(a1)$ $= 2,6 + 1$ $= 3,7 \text{ Km}$	9 Menit
3.	C-D	C-a2-D	$L(1) = C(a2) + D(a2)$ $= 1 + 1,5$ $= 2,5 \text{ Km}$	7 Menit
4.	D-C	D-a2-C	$L(1) = D(a2) + C(a2)$ $= 1,5 + 1$ $= 2,5 \text{ Km}$	7 Menit
5.	E-D	E-a3-D	$L(1) = E(a3) + D(a3)$ $= 1 + 1,8$ $= 2,8 \text{ Km}$	7 Menit
6.	F-H	F-a4-H	$L(1) = F(a4) + H(a4)$ $= 3 + 1,4$ $= 4,4 \text{ Km}$	15 Menit
7.	G-J	G-a5-J	$L(1) = G(a5) + J(a5)$ $= 3 + 1,2$ $= 4,2 \text{ Km}$	12 Menit

8.	H–F	H–a4–F	$L(1) = H(a4) + F(a4)$ $= 1,4 + 3$ $= 4,4 \text{ Km}$	15 Menit
9.	I–J	I–a6–J	$L(1) = I(a6) + J(a6)$ $= 1 + 1,9$ $= 2,9 \text{ Km}$	8 Menit
10.	J–I	J–a6–I	$L(1) = J(a6) + I(a6)$ $= 1,9 + 1$ $= 2,9 \text{ Km}$	8 Menit

4.1.3. Perancangan

1. Perancangan Database

Dalam pembuatan aplikasi ini *database* yang digunakan adalah *MySQL*.

Adapun *database* yang dibuat dalam perancangan pada aplikasi adalah:

a. Tabel Klinik

Tabel klinik digunakan untuk menyimpan data-data klinik gigi di kota Medan ke dalam *database*. Tabel ini berisi id, nama, alamat, lat, ing, lokasi terdekat dan *node*, dapat dilihat pada table 4.4 berikut :

Tabel 4.4 Tabel Klinik

Id	Nama	Alamat	Lat	Ing	Lokasi_terdekat	Node
1.	Osaka Dental Care	Text Data [118 B]	3.602301	98.645450	B	A
2.	Klinik Gigi ProDental Plaza	Text Data [108 B]	3.591522	98.632184	A	B
3.	Dokter Gigi Jenny	Text Data [104 B]	3.576391	98.617794	D	C

4.	Klinik Gigi Drg. Lilys Mei	Text Data [98 B]	3.570394	98.625749	C	D
5.	Klinik Pratama Senyum Jaya Abadi	Text Data [86 B]	3.558309	98.634893	D	E
6.	Medan Dental Specialist Center	Text Data [89 B]	3.587146	98.670967	H	F
7.	Puri Dental Clinic	Text Data [111 B]	3.581306	98.723101	J	G
8.	Praktek drg. Imelda Grace Marpaung	Text Data [83 B]	3.571547 5	98.687644 2	F	H
9.	Praktek Dokter Gigi drg Elvira Rosa Sinabutar M.Kes	Text Data [81 B]	3.609054	98.706719	J	I
10.	Dokter Gigi Spesialis Ortodonti	Text Data [104 B]	3.598175	98.704941	I	J

b. Tabel *Step Over*

Tabel *step over* digunakan untuk menyimpan data-data titik yang dilalui kedalam *database*. Tabel ini berisi id, kd_step, nama, alamat, lat, ing, dari dan ke, dapat dilihat pada table 4.5 berikut :

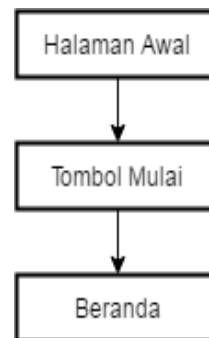
Tabel 4.5 Tabel *Step Over*

Id	kd_step	Nama	Alamat	Lat	Ing	Dari	ke
1	A	Helmi Clothing Distro Medan	Text Data [88 B]	3.597672	98.637738	A	B
3	C	Komplek SDH	Text Data [85 B]	3.573420	98.614891	C	D
4	D	Alfamidi Setia Budi 2	Text Data [86 B]	3.554619	98.632193	D	E
7	G	Rumah Makan Pondok Gurih	Text Data [87 B]	3.580141	98.683430	F	G
8	H	Perdana Bordir	Text Data [93 B]	3.581770	98.708389	F	–
11	K	PIONEER Ponsel	Text Data [106 B]	3.603377	98.706566	I	–

2. Diagram Tampilan *Interface*

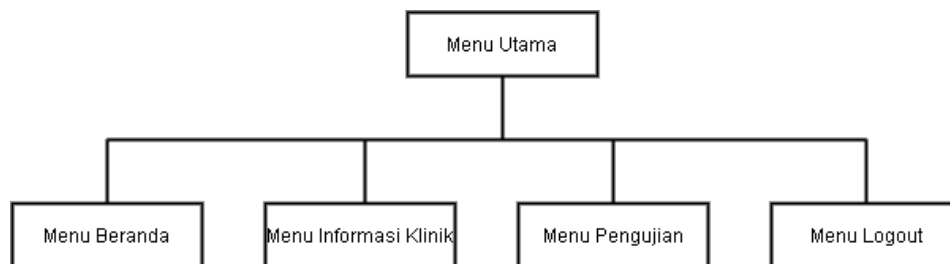
Diagram perancangan *interface* meliputi perancangan struktur menu dan perancangan tampilan pada tampilan user berbentuk diagram.

a. Diagram Tampilan Halaman Awal



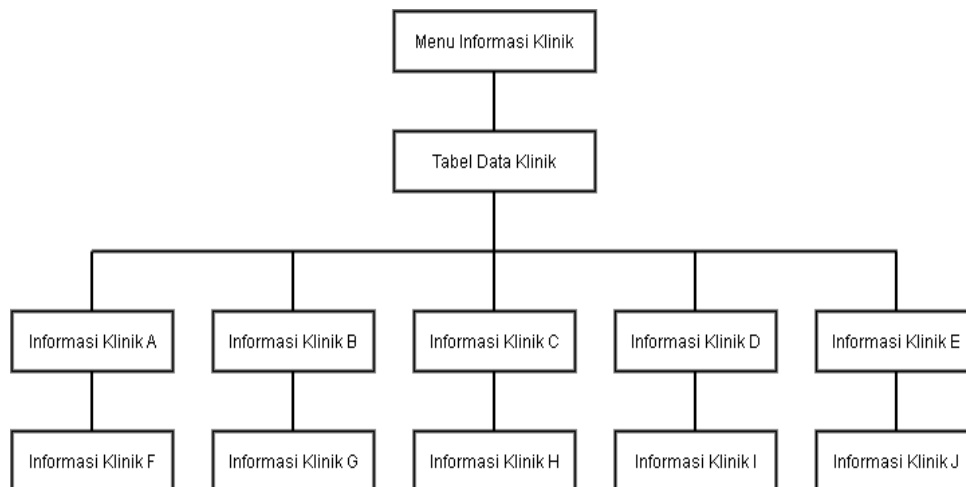
Gambar 4.22 Diagram Tampilan Halaman Awal

b. Diagram Tampilan Menu Utama



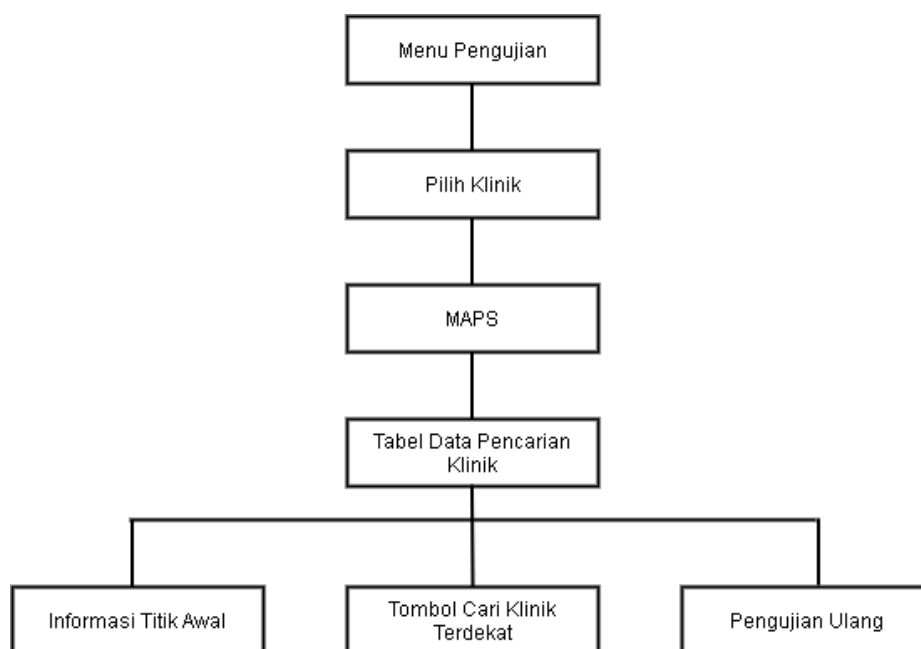
Gambar 4.23 Diagram Tampilan Menu Utama

c. Diagram Tampilan Menu Informasi Klinik



Gambar 4.24 Diagram Menu Informasi Klinik

d. Diagram Tampilan Menu Pengujian

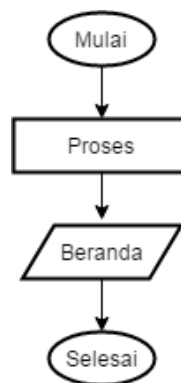


Gambar 4.25 Diagram Tampilan Menu Pengujian

3. Flowchart

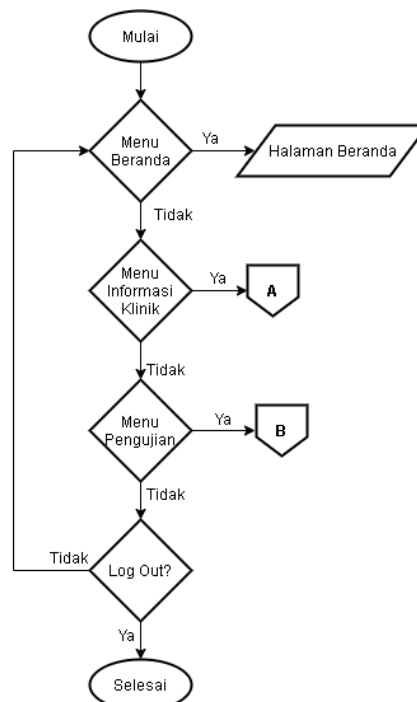
Setelah melakukan beberapa tahap rancangan pada penjelasan diatas, penjelasan tentang aliran sistem dari program kerja tersebut kemudian dibuat dalam bentuk diagram alur (*flowchart*) yang bertujuan untuk menggambarkan urutan dan hubungan proses dalam suatu program.

a. Flowchart Halaman Awal



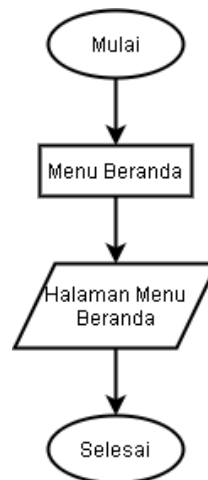
Gambar 4.26 Flowchart Halaman Awal

b. Flowchart Halaman Utama



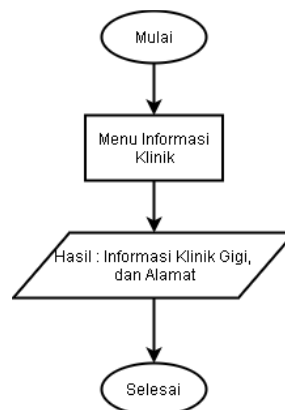
Gambar 4.27 Flowchart Menu Utama

c. *Flowchart* Menu Beranda



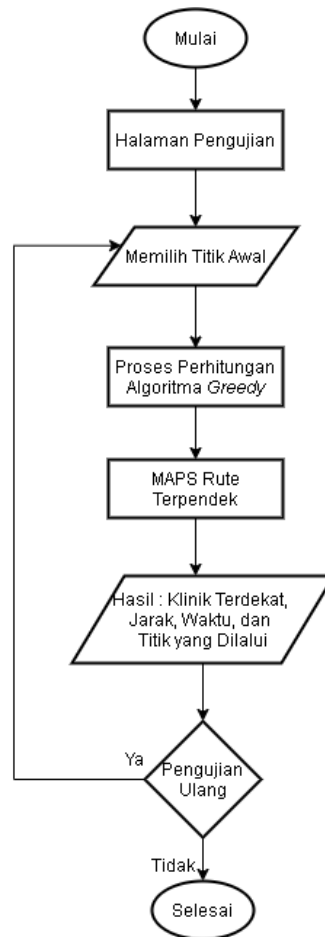
Gambar 4.28 *Flowchart* Menu Beranda

d. *Flowchart* Menu Informasi Klinik



Gambar 4.29 *Flowchart* Menu Informasi Klinik

e. *Flowchart* Menu Pengujian



Gambar 4.30 *Flowchart* Menu Pengujian

4.2. Hasil

4.2.1. Implementasi Sistem

Implementasi sistem merupakan kelanjutan dari tahap analisis dan desain sistem. Sistem ini dirancang menggunakan bahasa pemrograman *Java*. Dalam sistem ini, ada 4 (empat) tampilan menu, adalah menu home, menu informasi klinik, menu pengujian dan menu *logout*.

1. Tampilan Halaman Awal

Menu Halaman Awal adalah menu yang digunakan *user* untuk masuk ke dalam aplikasi menentukan rute terpendek antar klinik gigi di kota Medan.

Adapun tampilan halaman awal ditunjukkan pada gambar dibawah ini:

Aplikasi Rute Terpendek
Klinik Gigi di Kota Medan
dengan menggunakan Algoritma Greedy

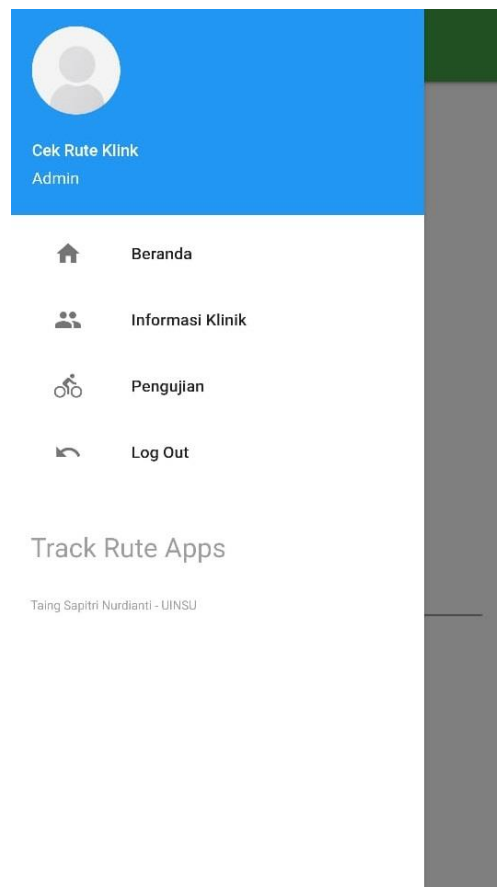


Status koneksi ke server : terhubung
Develop by : Taing Sapitri Nurdianti
Ilmu Komputer - Uinsu

Gambar 4.31 Tampilan Halaman Awal

2. Tampilan Menu Utama

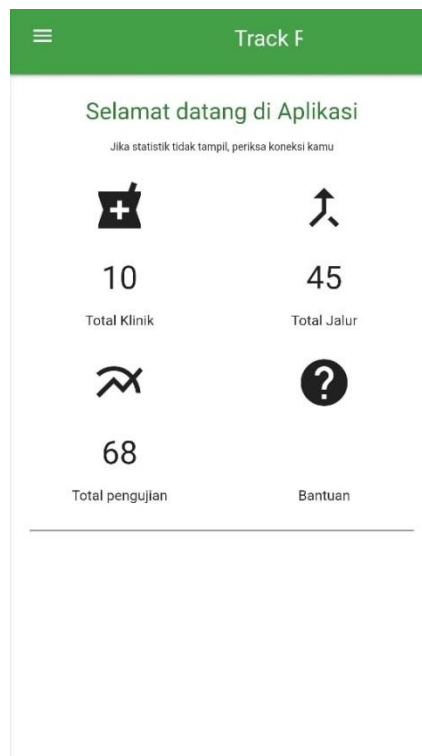
Tampilan halaman menu utama merupakan tampilan halaman yang terlihat pertama sekali pada saat user telah berhasil masukke aplikasi. Tampilan halaman menu utama dapat dilihat pada gambar dibawah ini :



Gambar 4.32 Tampilan Halaman Menu Utama

3. Tampilan Halaman Beranda

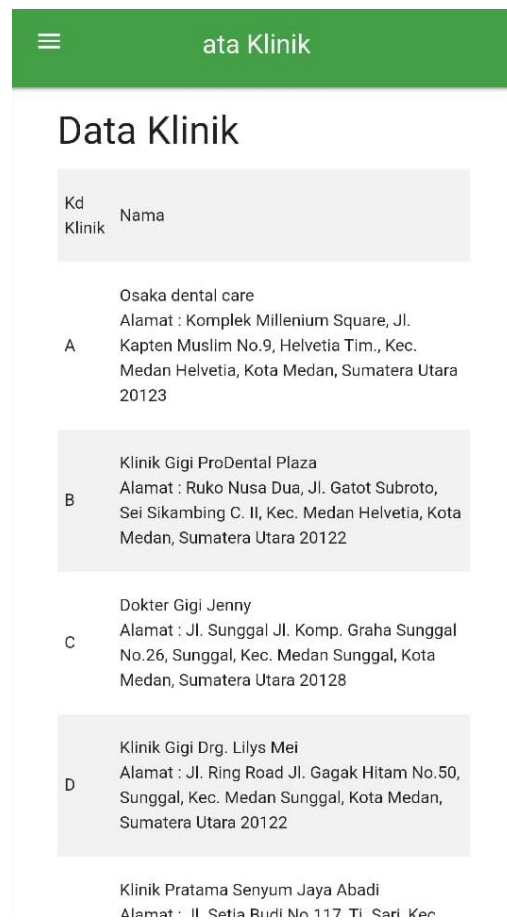
Tampilan halaman beranda adalah tampilan halaman berupa total klinik, total jalur, dan total pengujian. Tampilan halaman menu beranda dapat dilihat pada gambar dibawah ini :



Gambar 4.33 Tampilan Halaman Beranda

4. Tampilan Halaman Menu Informasi Klinik

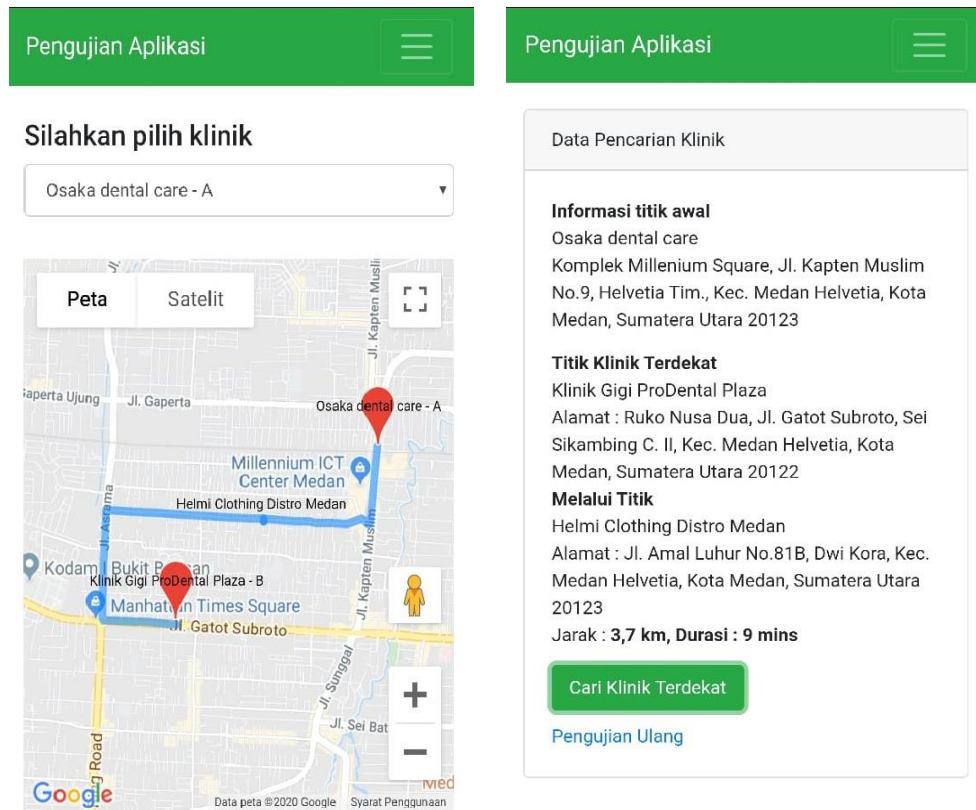
Tampilan halaman menu informasi klinik merupakan tampilan halaman berupa nama-nama 10 klinik gigi yang disediakan oleh admin. Tampilan halaman menu beranda dapat dilihat pada gambar dibawah ini :



Gambar 4.34 Tampilan Halaman Menu Informasi Klinik

5. Tampilan Halaman Menu Pengujian

Tampilan halaman menu pengujian merupakan halaman yang digunakan untuk memilih titik awal yang ada di aplikasi dipilih oleh *user*, kemudian pilih tombol “cari klinik terdekat” lalu menampilkan maps rute terpendek dari titik awal yang sudah dipilih, maka akan menampilkan hasil berupa jarak, waktu, dan titik yang dilalui. Tampilan halaman menu pengujian dapat dilihat pada gambar dibawah ini :



Gambar 4.35 Tampilan Halaman Menu Pengujian

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan studi literatur dan uji coba pada sistem ini, kesimpulan yang diperoleh adalah bahwa algoritma greedy merupakan algoritma untuk menemukan solusi optimal untuk masalah optimasi dengan menyelesaikan masalah langkah demi langkah. Dalam penelitian ini, ada langkah dalam proses menemukan rute terpendek antara klinik gigi Medan menggunakan algoritma *greedy* adalah menentukan pilihan rute yang terbaik tetapi tidak adanya pemilihan opsi sehingga dapat mengakibatkan konsekuensi dari pemilihan keputusan tersebut tidak dapat dipertanggungjawabkan secara penuh.

5.2. Saran

Saran yang dapat dipertimbangkan untuk penelitian dan pengembangan selanjutnya adalah sebagai berikut: sistem ini harus dikembangkan dengan meningkatkan jumlah simpul dan tepi, penelitian ini menggunakan 1 algoritma yang merupakan algoritma *greedy*, disarankan agar penelitian selanjutnya menambahkan algoritma terbaru yang dapat menentukan rute terpendek, pada penelitian ini aplikasi sudah *online* tetapi belum di *upload* di *Playstore*, disarankan agar penelitian selanjutnya aplikasi harus *online* dan sudah tersedia di *Playstore* agar *user* dapat menggunakannya kapan saja.

DAFTAR PUSTAKA

- Anwar, S., Nugroho, I., & Lestariningsih, E. (2015). Perancangan Dan Implementasi Aplikasi Mobile Semarang Guidance Pada Android. *Dinamik*, 20(2), 243541.
- Budiono, Y., & Santoso, H. B. (2016). Sistem Informasi Klinik Gigi Berbasis Web. *Eksis*, 09(01), 13–19.
- Fithaloka, Dhea. (2017). *Perbandingan Algoritma Greedy Dan Hill Climbing Untuk Menentukan Fasilitas Kesehatan Tingkat Pertama (Fktp) Terdekat Bagi Peserta Bpjs Kesehatan*.
- Fitriyani, R. E. (2018). *Analisa Pencarian Rute Tercepat Menuju Telaga Sarangan Menggunakan Algoritma Greedy*. 1(1), 1–5. <https://doi.org/10.31227/osf.io/7upkr>
- Furqan, Mhd. (2019). *Algoritma Hibrida Metode Heuristik Dan Eksak Untuk Menyelesaikan Persoalan Mixed Integer Nonliniar Programming Berskala Besar*.
- Hartanto, A. D., Mandala, A. S., P.L., D. R., Aminudin, S., & Yudirianto, A. (2019). Implementasi Algoritma Dijkstra Pada Game Pacman. *CCIT Journal*, 12(2), 170–176. <https://doi.org/10.33050/ccit.v12i2.687>
- Hayati, E. N., & Yohanes, A. (2014). Pencarian Rute Terpendek Menggunakan Algoritma Greedy. *Seminar Nasional IENACO*, 2337–4349.
- Ichana, Chayatul. (2018). Aplikasi Android Pencarian Mikrolet Kota Surabaya Dengan Menggunakan Algoritma Dijkstra. *Teknik Informatika, Fakultas Teknik*, 1, 6–8. <https://doi.org/10.16309/j.cnki.issn.1007-1776.2003.03.004>
- Mahendra, Y. D., Nuryanto, N., & Burhanuddin, A. (2019). Sistem Penentuan Jarak Terdekat Dalam Pengiriman Darah Di Pmi Kota Semarang Dengan Metode Algoritma Greedy. *Jurnal Komtika*, 2(2), 136–142. <https://doi.org/10.31603/komtika.v2i2.2601>
- Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman

- Menggunakan El-Goritma Berbasis Web. *Jurnal Teknik Mesin*, 6(2), 8.
<https://doi.org/10.22441/jtm.v6i2.1183>
- Nasution, Y. R. (2018). Penerapan Aplikasi Online Angket Persepsi Mahasiswa Terhadap Kinerja Dosen Uin Sumatera Utara Medan. *JISTech*, 3(2), 20–35. <https://doi.org/10.30829/jistech.v3i2.3158>
- Harahap, N. F. (2017). *Perbandingan Algoritma L-Deque Dan Algoritma Greedy Dalam Menentukan Rute Tependek Antar Tempat Wisata Di Kabupaten Tapanuli Tengah*.
- Nurul Fitria Insani. (n.d.). *Perancangan Aplikasi Pencarian Hotel Kota Bandung Jurnal*.
- Oktaviana, S., & Naufal, A. (2017). Algoritma Greedy untuk Optimalisasi Ruangan dalam Penyusunan Jadwal Perkuliahan. *Multinetics*, 3(1), 54. <https://doi.org/10.32722/multinetics.vol3.no.1.2017.pp.54-59>
- Prabowo, indra jaya krisna gede prabowo, Setiawan, A., & Prakoso, sabdo teguh. (2013). Pengembangan sistem mobile journal berbasis android untuk referensi belajar mahasiswa di lingkungan fakultas ekonomi uny. *Fakultas Pendidikan Dan Manajemen*, 1–3.
- Ripai, I. (2017). Rancangan Bangn Media Pembelajaran Menggunakan Android Untuk Mata Kuliah Pemrograman Internet Menggunakan Magazine App Marker. *Jurnal ICT Learning*, 3(1), 1–6.
- Rismayani, R., & Ardimansyah, A. (2015). Aplikasi Berbasis Mobile untuk Pencarian Rute Angkutan Umum Kota Makassar Menggunakan Algoritma Depth First Search. *Pekommas*, 18(3), 171–180. <https://doi.org/10.30818/jpkm.2015.1180303>
- Sabaruddin, R. (2016). Solusi Optimum Minmax 0/1 Knapsack Menggunakan Algoritma Greedy. *Jurnal Evolusi*, 23(45), 5–24.
- Santoso, S., & Nurmalina, R. (2017). Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut). *Jurnal Integrasi*, 9(1), 84–91.
- Surahman, S., & Setiawan, E. B. (2017). Aplikasi Mobile Driver Online Berbasis

Android Untuk Perusahaan Rental Kendaraan. *Jurnal ULTIMA InfoSys*, 8(1), 35–42. <https://doi.org/10.31937/si.v8i1.554>

Wulandari, Syindy. (2018). *Perbandingan Algoritma Brute-Force dan Algoritma A* untuk Mencari Rute Terpendek Antar Klinik Kecantikan di Kota Medan No Title*.

LAMPIRAN-LAMPIRAN

Lampiran 1

Listing Program Menggunakan *Visual Studio Code*

Fungsi untuk merender maps

```
varr directionsService = newgoogle.maps.DirectionsService;

        var diirectionsDisplay=new
        google.maps.DirectionsRenderer({suppressMarkers:true});

map = new google.maps.Map(document.getElementById('map'), {
    zoom: 17,
    center: {lat: 3.600969, lng: 98.697914}
});

directionsDisplay.setMap(map);
```

Fungsi untuk membuat marker lokasi klinik ke maps

```
$.post('http://api.haxors.or.id/taing/get_klinik_marker.php',function(data){

    let obj = JSON.parse(data);

    obj.forEach(myFunction);

    function myFunction(item, index){

        var marker = new google.maps.Marker({
            position : {lat : parseFloat(item.lat), lng:parseFloat(item.lng)},
            map: map,
            label : {
                text : item.nama+" - "+item.node,
                fontSize : "12px",
                color : "black"
            }
        });
    });
});
```


Fungsi untuk mencari jarak & waktu

```
service.getDistanceMatrix(  
{  
  origins: [start_ex_d],  
  destinations: [finish_ex_d],  
  travelMode: 'DRIVING',  
  unitSystem: 'google.maps.UnitSystem.METRIC',  
  avoidHighways: 'false',  
  avoidTolls: 'false',  
}, 'callback');
```

Fungsi untuk menampilkan rute ke maps

```
directionsService.route({  
  origin: start_ex,  
  destination: finish_ex,  
  travelMode: 'DRIVING',  
  waypoints : waypts,  
  optimizeWaypoints : true,  
}, 'function(response, status') {  
  if ($status === 'OK') {  
    'directionsDisplay'.setDirections(response);  
    $('#error').empty();  
    $('#error').removeClass();  
  } else {  
    $('#error').addClass("badge badge-danger");  
    $('#error').text("Tidak dapat menemukan nama lokasi, status  
error: "+status);  
  }  
});
```

Fungsi backend untuk menghitung jarak

```

<?php
header("Access-Control-Allow-Origin: *");

class data_respon{}

$data_r = new data_respon();

$node = $_POST['node'];

$kNode = $link ->query("SELECT * FROM tbl_node WHERE node='$node'
LIMIT 0,1;");

$fNode = $kNode -> fetch_array();

$nama = $fNode['nama'];
$alamat = $fNode['alamat'];
$lat = $fNode['lat'];
$lng = $fNode['lng'];
$lokasiTerdekat = $fNode['lokasi_terdekat'];

$data_r ->nama = $nama;
$data_r -> alamat = $alamat;
$data_r -> lat = $lat;
$data_r ->lng = $lng;
$data_r -> lokasi_terdekat = $lokasiTerdekat;

echo json_encode($data_r);

?>

```

Fungsi backend untuk memberikan data klinik

```

<?php
header("Access-Control-Allow-Origin: *");

include('db.php');

class data_respon{}

$data_r = new data_respon();

```

Proses Hitung

```
<?php
header("Access-Control-Allow-Origin: *");
include('db.php');
class data_respon{}
$data_r = new data_respon();

$start = $_POST['start'];
$finish = $_POST['finish'];
$jarak = $_POST['jarak'];
$durasi = $_POST['waktu'];
$waktu'' = datee("Y-m-d H:i:s");
```

cari start

```
$k_klinik_start = $link ->query("SELECT * FROM tbl_klinik WHERE
    kd_klinik='$start' LIMIT 0,1;");
$fks = $k_klinik_start ->farray();
$nm_ks = $fks['nama'];
```

cari finish

```
$k_klinik_finish = $link ->query("SELECT * FROM tbl_klinik WHERE
    kd_klinik='$finish' LIMIT 0,1;");
$fkf = $k_klinik_finish -> fetch_array();
$nm_kf = $fkf['nama'];
$id_record = "";
$stat_node = "";
//kueri data jalur
$kJalur1 = $link ->query("SELECT id FROM tbl_refrensi WHERE
    dari='$start' AND ke='$finish' LIMIT 0,1;");
$jhlhJalur = mysqli_num_rows($kJalur1);
if($jhlhJalur < 1){
    $finish_rev = $start;
    $kJalur2 = $link ->query("SELECT id FROM tbl_refrensi WHERE
        dari='$start_rev' AND ke='$finish_rev' LIMIT 0,1;");
    $fR2 = $kJalur2 -> fetch_array();
    $id_record = $fR2['id'];
    $stat_node = "mundur";
}
```

simpan ke data perhitungan

```
$link ->query("INSERT INTO tbl_perhitungan VALUES(null,'$start','$
    '$id_record','$waktu','$jarak','$durasi');");

$data_r ->start = $nm_ks;
$data_r -> finish = $nm_kf;
$data_r -> isiTabel = $tabel_akhir;
$data_r -> idRecord = $id_record;
$data_r -> statNode = $stat_node;
$data_r -> jlh_node = $jlhNode;
```

```
echo json_encode($data_r);
```

```
<?php
header("Access-Control-Allow-Origin: *");>]
```

Lampiran 2

DAFTAR RIWAYAT HIDUP (*CURRICULUM VITAE*)



Nama : Taing Sapitri Nurdianti
Nim : 71153011
Tempat, Tanggal Lahir : Jakarta, 09 September 1997
Jenis Kelamin : Perempuan
Alamat : Jl. Pendidikan No. 14
Kel/Desa : Sipolu-polu
Kecamatan : Panyabungan
Kabupaten : Mandailing Natal
Agama : Islam
Status Nikah : Belum Menikah
No HP : 082168525547
Nama Orangtua
Ayah : Pinayungan Lubis
Ibu : Sakirah

PENDIDIKAN FORMAL

2003-2009 : SD Negeri 088 Panyabungan
2009-2012 : SMP Negeri 2 Panyabungan
2012-2015 : SMK Negeri 1 Panyabungan
2015-2020 : Universitas Islam Negeri Sumatera Utara Medan