

## Perbandingan Algoritma *Greedy* dan *Hill Climbing* Untuk Menentukan Fasilitas Kesehatan Tingkat Pertama (FKTP) Terdekat Bagi Peserta BPJS Kesehatan

Dhea Fithaloka<sup>1</sup>, Mohammad Andri Budiman<sup>2</sup>, Dian Rachmawati<sup>3</sup>

<sup>1,2,3</sup>Program Studi S1 Ilmu Komputer, Fasilkom-TI, Universitas Sumatera Utara  
Jalan Universitas No. 9 Kampus Universitas Sumatera Utara Medan 20155

<sup>1</sup>dheafithaloka29@gmail.com

<sup>2</sup>mandrib@usu.ac.id

<sup>3</sup>dian.rachmawati@usu.ac.id

### Abstrak

Kebutuhan pencarian Fasilitas Kesehatan Tingkat Pertama di Kota Medan termasuk cukup besar, dimana Fasilitas Kesehatan Tingkat Pertama melayani sesuai keluhan pasien, seperti penyakit umum, rawat jalan dan rawat inap, konsultasi, obat-obat dan lain sebagainya. Terdapat pilihan wilayah yang dapat ditempuh untuk menuju Fasilitas Kesehatan Tingkat Pertama terdekat yang diinginkan, terdapat 21 wilayah Fasilitas Kesehatan Tingkat Pertama di kota medan. Dalam pencarian Fasilitas Kesehatan Tingkat Pertama terdekat di Kota Medan tersebut akan diterapkan kedalam sebuah graf. Dalam menyelesaikan graf diperlukan pula algoritma, algoritma yang akan digunakan yaitu algoritma *Hill Climbing* dan algoritma *Greedy*, dimana algoritma *Hill Climbing* adalah suatu metode untuk mencari dan menentukan rute yang paling singkat dengan memperkecil tempat yang ditinggahi dengan menggunakan cara *heuristic* dan algoritma *Greedy* memberikan solusi memecahkan masalah dengan membuat pilihan optimum lokal. Berdasarkan Hasil pencarian Fasilitas Kesehatan Tingkat Pertama di Kota Medan dengan menggunakan algoritma *Hill Climbing* dan algoritma *Greedy* menunjukkan hasil yang berbeda dan *running time* yang berbeda dimana algoritma *Hill Climbing* memiliki nilai *running time* yang lebih cepat serta menunjukkan hasil yang sesuai dengan tujuan dibandingkan algoritma *Greedy*.

**Kata kunci**— Fasilitas Kesehatan Tingkat Pertama, Graf, Algoritma *Hill Climbing*, Algoritma *Greedy*, *Running Time*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Asuransi merupakan penjaminan yang berlaku dalam masyarakat modern. Masyarakat modern sekarang ini menginginkan rasa aman (*safety*) bagi kelangsungan kehidupannya. Sistem asuransi dapat memenuhi kebutuhan akan rasa aman itu untuk melindungi (*protect*) beberapa aspek kehidupan masyarakat, seperti perlindungan bagi biaya kesehatan, hari tua, atau pendidikan anak. Bila terjadi sesuatu hal yang tidak diinginkan pada diri peserta, sudah ada

penopang (*backup*) yang dapat meringankan resiko. Saat ini pemerintah mempercayakan BPJS Kesehatan sebagai penyelenggara jaminan pemeliharaan kesehatan seluruh rakyat Indonesia. BPJS Kesehatan adalah suatu badan hukum publik yang dulunya adalah badan usaha milik negara dengan nama PT. Askes, dimana sejak 1 Januari 2014 mulai beroperasi dengan menangani peserta PT. Askes terdahulu, peserta dari PT. Jamsostek, TNI/POLRI, Jamkesmas, Jamkesda dan peserta individu.

Seluruh peserta BPJS Kesehatan wajib terdaftar di salah satu Fasilitas Kesehatan Tingkat Pertama yang terdekat dengan

domisili tempat tinggalnya. Fasilitas Kesehatan Pertama yang disediakan oleh BPJS Kesehatan adalah Puskesmas, Klinik Swasta, Klinik TNI/Polri dan Balai Pengobatan. Pelayanan Kesehatan bagi Peserta BPJS Kesehatan dilakukan sistem rujukan berjenjang. Rujukan berjenjang yaitu dari Fasilitas Kesehatan Tingkat Pertama kemudian di rujuk ke Fasilitas Kesehatan Tingkat Lanjutan. Pemilihan Fasilitas Tingkat Pertama dilakukan saat peserta BPJS Kesehatan melakukan pendaftaran. Apabila FKTP yang dipilih tutup ataupun penuh, maka peserta dapat melakukan pelayanan ke FKTP terdekat lainnya.

Hambatan dalam mencari FKTP terdekat membuat masyarakat sulit untuk menentukan fasilitas kesehatan yang terdekat dengan lokasi tempat tinggalnya ataupun lokasi keberadaan saat membutuhkan pelayanan kesehatan. Dengan menggunakan algoritma *Greedy* dan *Hill Climbing* dapat membantu masyarakat mencari lokasi terdekat tentang lokasi pelayanan FKTP.

Secara umum, pencarian jalur terpendek dapat dibagi menjadi 2 (dua) metode, yaitu: metode konvensional dan metode heuristik. Metode konvensional merupakan metode yang menggunakan perhitungan matematik biasa, pada pencarian lintasan terpendek hanya dapat diselesaikan untuk 5 (lima) sampai 10 (sepuluh) *verteks*, untuk *verteks* yang lebih banyak metode *heuristik* lebih variatif dan waktu perhitungan yang diperlukan lebih singkat, karena metode *heuristik* menggunakan metode pendekatan dan melakukan pencarian [Lubis, 2009].

Algoritma *Hill Climbing* dapat diterapkan untuk kedua diskrit dan masalah kontinyu [Dowland, 1991]. Metode ini hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi *heuristik*. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi *heuristik* ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin [Sri Kusumadewi, 2003].

Algoritma *Greedy* merupakan salah satu algoritma yang sering digunakan untuk memecahkan permasalahan optimasi. Pada setiap langkah, kita harus membuat pilihan optimum lokal (*local optimum*), yang dapat diartikan sebagai keputusan terbaik yang diambil pada langkah tersebut. (Munir, 2007)

Dari pemaparan di atas, penulis melakukan penelitian yang berjudul “Perbandingan Algoritma *Greedy* dan *Hill Climbing* Untuk Menentukan Fasilitas Kesehatan Tingkat Pertama (FKTP) Terdekat Bagi Peserta BPJS Kesehatan”.

## 1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, perumusan masalah yang akan dibahas ialah menentukan fasilitas kesehatan tingkat pertama (FKTP) yang terdekat bagi peserta BPJS Kesehatan dengan menggunakan Algoritma *Greedy* dan Algoritma *Hill Climbing*.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah :

1. Data fasilitas kesehatan tingkat pertama yang digunakan adalah fasilitas kesehatan tingkat pertama (Puskesmas/Klinik) di wilayah Kota Medan, yaitu Kecamatan Medan Barat, Medan Helvetia, dan Medan Timur.
2. Proses pencarian dilakukan dengan metode Algoritma *Greedy* dan Algoritma *Hill Climbing*.
3. Graf yang digunakan hanya graf statis saja.
4. Bobot yang digunakan adalah jarak.
5. Titik tujuan yang akan dikunjungi sebanyak 32 titik yaitu pada Puskesmas/Klinik pada tiga wilayah kecamatan yang dipilih
6. Parameter:
  - a. Titik awal adalah fasilitas kesehatan tingkat pertama
  - b. Satuan waktu (*running time* dari *software* aplikasi) dalam menentukan jarak terpendek (ms).
  - c. Keoptimalan jarak terpendek (km)
7. Bahasa pemrograman menggunakan C# dan menggunakan *database management system* SQLite.

#### 1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang sebuah aplikasi yang dapat membantu peserta BPJS Kesehatan dalam menentukan FKTP terdekat dengan menggunakan Algoritma *Greedy* dan *Hill Climbing*.

#### 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi peserta BPJS Kesehatan dalam menentukan FKTP terdekat dengan domisilinya, juga memudahkan bagi petugas BPJS Kesehatan dalam memilihkan FKTP untuk pesertanya.

#### 1.6 Metodologi Penelitian

Penelitian ini menerapkan beberapa metode penelitian sebagai berikut:

##### 1. Studi Literatur

Pada tahap ini dilakukan pengumpulan referensi yang diperlukan dalam penelitian. Hal ini dilakukan untuk memperoleh informasi dan data yang diperlukan untuk penelitian ini. Referensi yang digunakan dapat berupa buku, jurnal, artikel, situs internet yang berkaitan dengan penelitian ini dengan beberapa topik seperti pengertian algoritma, graf, jenis- jenis graf, *shortest path*, algoritma *Greedy* dan *Hill Climbing*.

##### 2. Pengumpulan dan Analisis Data

Pada tahap ini dilakukan pengumpulan dan analisa data yang berhubungan dengan penelitian ini seperti fungsi algoritma *Greedy* dan *Hill Climbing*., serta data jarak FKTP.

##### 3. Perancangan Sistem

Pada tahap ini dilakukan perancangan *GUI* (*Graphic User Interface*), *Unified Modeling Language* (*UML*), dan struktur program sistem pengamanan teks.

##### 4. Implementasi Sistem

Sistem diimplementasikan dengan menggunakan Algoritma algoritma *Greedy* dan *Hill Climbing*.

##### 5. Pengujian Sistem

Pada tahap ini akan dilakukan pengujian terhadap sistem yang telah dikembangkan.

##### 6. Dokumentasi Sistem

Melakukan pembuatan dokumentasi sistem mulai dari tahap awal hingga pengujian sistem.

## 2. TINJAUAN TEORETIS

### 2.1 Algoritma

Algoritma merupakan suatu langkah langkah untuk menyelesaikan masalah yang disusun secara sistematis, tanpa memperhatikan bentuk yang akan digunakan sebagai implementasinya, sehingga suatu algoritma dapat menjelaskan “bagaimana” langkah – langkah dalam melaksanakan suatu fungsi yang dapat diimplementasikan dengan suatu program atau suatu komponen fisik [Hartono, 2007].

Menurut Donald E.Knuth, terdapat ciri-ciri algoritma yaitu:

1. Algoritma memiliki langkah yang terbatas dan harus berhenti saat pada langkah terakhir.
2. Setiap langkah harus didefinisikan dengan tepat dan tidak ambiguous (tidak berarti dua).
3. Algoritma memiliki nol atau lebih masukan (input). Masukan ialah besaran yang diberikan pada algoritma sebelum algoritma mulai bekerja.

### 2.2 Teori Dasar Graf

Teori graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek-objek tersebut. Representasi visual dari graf adalah dengan menguyatakan objek dinyatakan sebagai noktah, bulatan atau titik, sedangkan hubungan antara objek dinyatakan dengan garis. Sesungguhnya peta adalah sebuah graf, yang dalam hal ini kota dinyatakan sebagai bulatan sedangkan jalan dinyatakan sebagai garis [Mediputra, 2010].

Graf adalah kumpulan node (simpul) di dalam bidang dua dimensi yang dihubungkan dengan sekumpulan garis (sisi). Graf dapat digunakan untuk merepresentasika objek-objek diskrit dan hubungan antara objek-objek tersebut. Representasi visual dari graf adalah

dengan menyatakan node, bulatan atau titik (*vertex*), sedangkan hubungan antara objek dengan garis dinyatakan dengan baris (*edge*) [Munir, 2010].

### 2.3 Lintasan Terpendek (Shortest Path)

Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat tertentu.

Ada beberapa macam persoalan lintasan minimum, antara lain :

1. Lintasan minimum antara dua buah titik.
2. Lintasan minimum antara semua pasangan titik.
3. Lintasan minimum dari titik tertentu ke semua titik yang lain
4. Lintasan minimum antara dua buah titik yang melalui beberapa titik tertentu.

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. Dalam kasus ini, bobot yang dimaksud berupa jarak dan waktu kemacetan terjadi [Prana, 2010].

### 2.4 Algoritma Greedy

Dalam bahasa Inggris, *Greedy* berarti rakus, tamak, atau loba. Definisi ini sangat sesuai dengan prinsip algoritma *Greedy*, yaitu “*take what you can get now!*”. Algoritma *Greedy* membentuk solusi langkah per langkah (*step by step*). Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Algoritma *Greedy* merupakan salah satu algoritma yang sering digunakan untuk memecahkan permasalahan optimasi. Pada setiap langkah, kita harus membuat pilihan optimum lokal (*local optimum*), yang dapat diartikan sebagai keputusan terbaik yang diambil pada langkah tersebut. [Munir, 2007].

Algoritma *Greedy* membuat list terurut dari lintasan yang dikenal, yang biasa disebut batas. Jalur di batas diwakili oleh tiga komponen: *vertex* terakhir di jalan, biaya jalan, dan *vertex* berikutnya ke yang terakhir di jalan. Dengan menyimpan *vertex* berikutnya ke yang terakhir, jalur dari *vertex* akhir ke

sumber dapat direkonstruksi (dengan mengikuti nilai-nilai sebelumnya untuk jalur lainnya titik terakhir semua jalan ke sumber). Awalnya, batas hanya berisi simpul sumber (jalur biaya nol). [Adamson & Tick, 1991].

Algoritma *Greedy* biasanya dilakukan dalam beberapa langkah, maka dalam satu permasalahan optimasi kita akan membuat beberapa pilihan optimum lokal, sesuai dengan banyaknya langkah yang harus dilakukan. Diharapkan bahwa setiap kali diambil pilihan optimum lokal, langkah-langkah setelahnya akan mengarah ke solusi optimum global. Perlu diingat bahwa algoritma *Greedy* hanya memakai dua macam persoalan optimasi, yaitu maksimasi (*maximization*) dan minimasi (*minimization*). [Hayati, 2014].

Persoalan optimasi dalam konteks algoritma *Greedy* disusun oleh elemen-elemen sebagai berikut:

a. Himpunan kandidat, C.

Himpunan ini berisi elemen-elemen pembentuk solusi. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.

b. Himpunan solusi, S.

Merupakan himpunan dari kandidat-kandidat yang terpilih sebagai solusi persoalan. Himpunan solusi adalah himpunan bagian dari himpunan kandidat.

c. Fungsi seleksi

Merupakan fungsi yang pada setiap langkah memilih kandidat yang paling mungkin untuk mendapatkan solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya.

d. Fungsi kelayakan (feasible).

Fungsi ini akan memeriksa kelayakan suatu kandidat yang telah dipilih. Dalam arti, kandidat tersebut dan himpunan solusi yang terbentuk tidak melanggar constraints yang ada. Bila kandidat layak, maka kandidat tersebut akan dimasukkan ke dalam himpunan solusi, dan jika kandidat tersebut tidak layak, maka kandidat akan dibuang dan tidak akan dipertimbangkan lagi dalam pencarian solusi optimum.

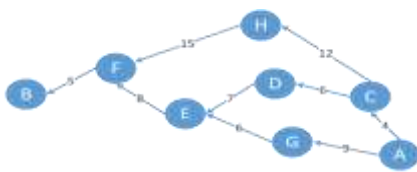
e. Fungsi obyektif.

Fungsi ini akan membuat nilai solusi maksimum atau minimum, sesuai dengan

jenis optimasi apa yang dibutuhkan. [Defindal, 2010].

Dengan kata lain, algoritma *Greedy* akan melakukan pencarian sebuah himpunan bagian S dari himpunan kandidat C. Himpunan bagian S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi obyektif. [Defindal, 2010].

Dalam contoh berikut jika diberikan sebuah pencarian berdasarkan Gambar 1 dengan graf searah, Melalui titik A menuju ke titik E.



**Gambar 1 Contoh Kasus Pencarian Berdasar Algoritma Greedy**

Dikarekan algoritma *Greedy* hanya berfokus pada local maximum dimana algoritma ini hanya memilih rute berdasarkan jarak-jarak terkecil yang dilalui.

Melalui titik A → C, dengan jarak 4

Melalui titik C → D, dengan jarak 6

Melalui titik D → E, dengan jarak 7

Sehingga jarak yang didapat adalah  $4 + 6 + 7 = 17$ .

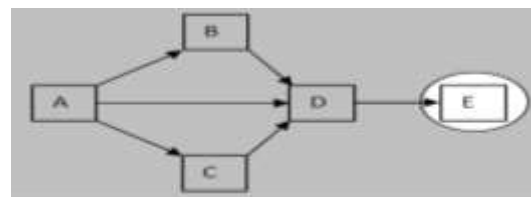
## 2.5 Algoritma Hill Climbing

Metode *Hill Climbing* adalah suatu metode untuk mencari dan menentukan rute yang paling singkat dengan memperkecil jumlah kota atau tempat yang disinggahi dengan menggunakan cara *heuristic*. Cara kerjanya adalah menentukan langkah berikutnya dengan menempatkan node yang akan muncul sedekat mungkin dengan sasarannya.

Algoritma untuk *Hill Climbing* adalah sebagai berikut:

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kerjakanlah langkah-langkah berikut sampai solusinya ditemukan, atau sampai tidak ada node baru yang akan diaplikasikan pada keadaan sekarang:

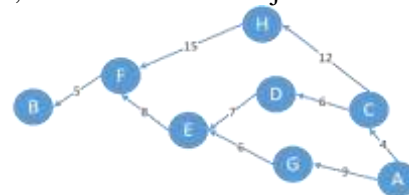
- a. Cari *node* yang belum pernah digunakan; gunakan node ini untuk mendapatkan keadaan yang baru.
- b. Evaluasi keadaan baru tersebut.
  - i. Jika keadaan baru merupakan tujuan, keluar.
  - ii. Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikanlah keadaan baru tersebut menjadi keadaan sekarang.
  - iii. Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan pencarian.



**Gambar 2 Proses penelusuran dengan metode Hill Climbing.**

Metode *Hill Climbing* terinspirasi akan langkah-langkah yang dilakukan oleh para pendaki dalam menemukan camp mereka yang terletak diatas lereng gunung bagian atas. Para pendaki selalu akan mencari jalan yang lebih pintas untuk mencapai tujuannya. Pada Gambar 2 diilustrasikan bahwa untuk mencapai titik E dari titik A mempunyai 3 alternatif jalur yaitu A-B-D-E, A-C-D-E, dan A-D-E. Penentuan rute yang akan dipilih pada metode *Hill Climbing* akan dibandingkan ketiga jalur tersebut mana yang paling sedikit *cost* yang harus dikeluarkan, apakah rute yang paling pendek ataupun tingkat kemacetan yang paling kecil, pemilihan akan bergantung pada informasi yang diberikan pada peta yang akan dilalui [Pribadi, 2010].

Dalam contoh berikut jika diberikan sebuah pencarian berdasarkan Gambar 3 dengan graf searah, melalui titik A menuju ke titik E.



**Gambar 3 Contoh Kasus Pencarian Berdasar Algoritma Hill Climbing**

Melalui titik A → C, dengan jarak 4  
 Melalui titik A → G, dengan jarak 9  
 Algoritma *Hill Climbing* akan memilih jarak terkecil untuk dilalui, sehingga keputusan pertama akan melalui titik A → C.

Melalui titik C → D, dengan jarak 6, total jarak 10.

Algoritma *Hill Climbing* akan memilih satu jarak sebelum, dikarenakan jarak antara C → D lebih besar di banding dengan jarak antara A → G.

Melalui titik A → G, dengan jarak 9

Melalui titik G → E, dengan jarak 6, total jarak 15.

Didapat titik E dengan total terkecil dan titik tujuan.

### 3. ANALISIS DAN PERANCANGAN

#### 3.1 Analisis Sistem

##### 1) Analisis Masalah

Banyak cara untuk mencapai suatu tempat yang diinginkan dengan mudah, dalam hal ini untuk menuju Fasilitas Kesehatan Tingkat Pertama yang diinginkan diperlukan suatu cara yang cepat dalam mencari jarak terpendeknya dan dapat memilih solusi yang paling optimal, sehingga dapat menghemat waktu dan membuat perjalanan lebih efektif dan efisien. Dengan dasar ini penulis mencari suatu cara untuk mencari Fasilitas Kesehatan Tingkat Pertama terdekat di Kota Medan dengan menggunakan dua buah algoritma yaitu Algoritma *Greedy* dan *Hill Climbing*.

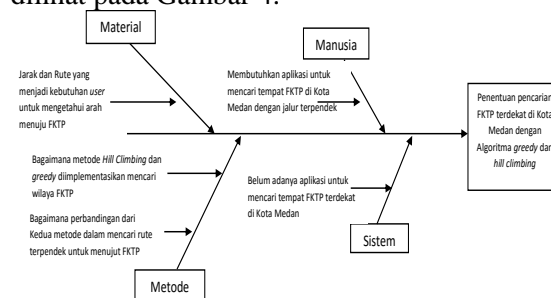
Permasalahan pada penelitian ini, bukan hanya sebatas pencarian jarak terdekat dengan dua buah algoritma *Greedy* dan *Hill Climbing*, tetapi juga dengan membandingkan jarak dan hasil kerja kedua algoritma tersebut dan dalam pengaplikasiannya graf dibentuk dengan mengikuti peta Kota Medan, dimana *vertex* ditentukan berdasarkan beberapa nama Fasilitas Kesehatan Tingkat Pertama yang telah di pilih yang terdapat di Kota Medan, dan *edge* merupakan panjang jarak jalan yang akan dilalui nantinya dari Fasilitas Kesehatan Tingkat Pertama satu ke Fasilitas Kesehatan Tingkat Pertama lainnya.

Terdapat 3 Wilayah Fasilitas Kesehatan Tingkat Pertama di Kota Medan yang telah dipilih dan akan diterapkan kedalam graf pada sistem yang dibuat, nama-nama Fasilitas Kesehatan Tingkat Pertama yang digunakan yaitu :

1. Wilayah Medan Barat
2. Wilayah Medan Helvetia
3. Wilayah Medan Timur

##### 2) Diagram Ishikawa

Dalam mengidentifikasi suatu masalah dapat menggunakan diagram *Ishikawa* (*fishbone diagram*). Diagram *Ishikawa* merupakan suatu alat visual untuk mengidentifikasi, mengeksplorasi, dan secara grafik menggambarkan secara detail semua penyebab yang berhubungan dengan suatu permasalahan. Diagram *Ishikawa* bentuknya menyerupai kerangka tulang ikan yang bagian-bagiannya meliputi kepala, sirip, dan duri dimana digambarkan seperti susunan tulang ikan yang terbagi atas dua bagian yaitu bagian kepala ikan yang merupakan masalah, bagian utama tulang belakang merupakan kemungkinan-kemungkinan penyebab masalah. Diagram *Ishikawa* sistem ini dapat dilihat pada Gambar 4.



Gambar 4. Diagram *Ishikawa*

Diagram *Ishikawa* diatas terbagi atas 2 (dua) bagian yaitu *head* dan *bone*. *Bone* terdiri dari 4 aspek yaitu material, metode, user dan sistem. Material adalah apa saja yang diperlukan dalam menjalankan sistem yaitu jarak dan rute yang digunakan untuk mengetahui seberapa jauh jalan yang akan ditempuh dan rute yang akan dilalui. Metode adalah kebutuhan yang spesifik dari proses yang terdiri dari 3 (tiga) bagian yaitu bagian

algoritma *Hill Climbing*, algoritma *Greedy* dan Hasil Pencarian dan Perbandingan kedua algoritma yang akan digunakan untuk menjalankan hasil graf yang terbentuk. *User* adalah apa saja yang akan diketahui dari sistem yang akan dikerjakan dimana user sebelumnya belum mengetahui cara kerja sistem. Sistem adalah hal yang akan dilakukan atau dibuat yaitu dengan membuat aplikasi pencarian Fasilitas Kesehatan Tingkat Pertama terdekat di Kota Medan dengan menggunakan algoritma *Hill Climbing* dan algoritma *Greedy*, melakukan perbandingan dari hasil kerja kedua algoritma tersebut.

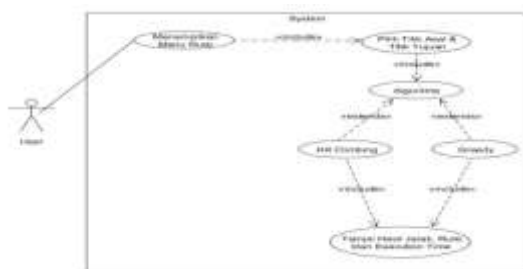
### 3.2 Perancangan Sistem

Pada bagian ini digunakan *Unified Modeling Language (UML)* sebagai bahasa spesifikasi standar suatu model yang berfungsi untuk membantu merancang sistem. Sistem ini dibangun dengan membuat use case diagram, sequence diagram dan activity diagram.

#### 1) Use Case Diagram

*Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga *user* paham dan mengerti mengenai kegunaan sistem yang dibangun, dimana penggambaran sistem dari sudut pandang user itu sendiri sehingga *use case* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian.

Berikut *use case* diagram dapat dilihat pada Gambar 5



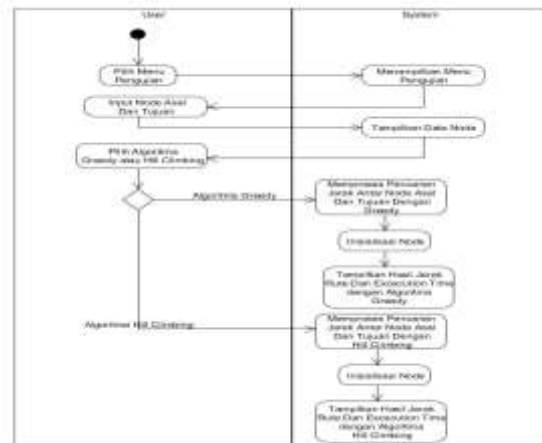
Gambar 5 Use Case Diagram

#### 2) Activity Diagram

*Activity* diagram dapat menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses, serta dapat menggambarkan alur aktivitas kerja pada sistem yang sedang dirancang ataupun berjalan, bagaimana

masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana berakhir.

*Activity* diagram pada Gambar 6. menjelaskan proses kerja dari sistem untuk proses menentukan titik awal dan akhir serta memilih algoritma yang akan digunakan oleh user. *Activity* diagram sistem pada aplikasi yang dibuat dapat dilihat pada Gambar 6.

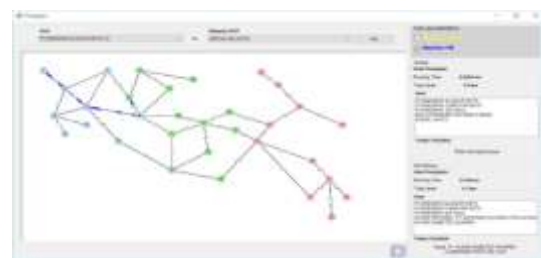


Gambar 6 Activity Diagram Cara Kerja Sistem

## 4. IMPLEMENTASI DAN PENGUJIAN

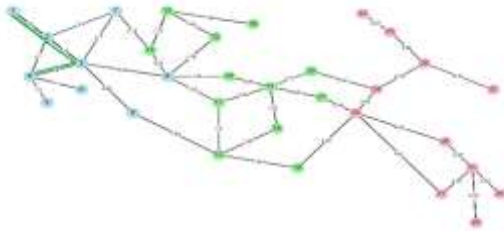
### 4.1 Pengujian Algoritma Hill Climbing

Adapun algoritma *Hill Climbing* menghasilkan rute-rute baru sampai rute tersebut merupakan rute terpendek dan berhasil mencapai tujuan. Proses perhitungan algoritma ini diilustrasikan dengan menggunakan tree. Pada awalnya tree hanya berisi titik awal pencarian.



Gambar 7 Tampilan Pengujian Pencarian Wilayah FKTP dengan Algoritma *Hill Climbing* (Rute Biru)

Sebagai contoh pengujian dilakukan proses pencarian FKTP dengan Asal FKTP Puskesmas Glugur Kota menuju wilayah FKTP ke Medan Helvetia dengan



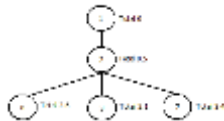
menggunakan algoritma *Hill Climbing* dapat dilihat pada Gambar 7.



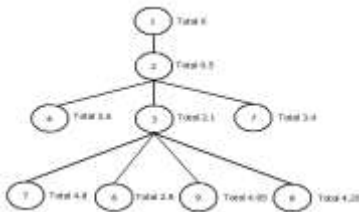
Dengan menggunakan titik awal ini, dilakukan proses menghasilkan cabang dari *tree* yang baru berdasarkan titik yang terhubung dengan titik 1, dimana 1 merupakan Puskesmas Glugur Kota.



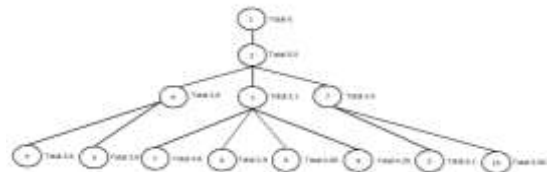
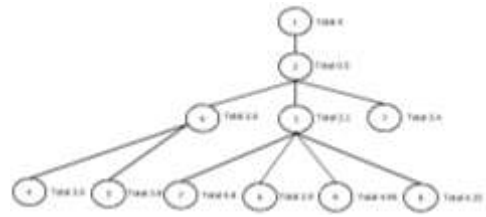
Dari cabang yang terbuka yaitu titik 2, dikembangkan lagi dengan cara melihat titik-titik yang terhubung dengan titik 2 dan menghitung total jaraknya. Dimana titik 2 tersebut merupakan FKTP yang terdekat dengan jarak terpendek yaitu Puskesmas P. Brayan Kota.



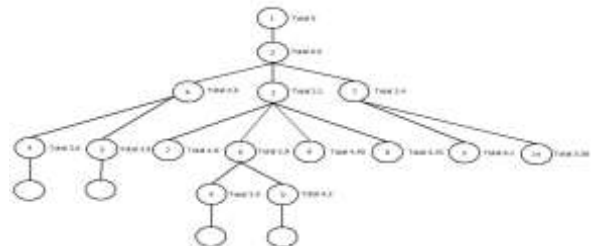
Lalu selanjutnya terlihat posisi *tree* dimana terdapat lebih dari satu cabang, maka dipilih cabang dengan nilai total terkecil, dimana dalam kondisi ini adalah titik 3, yaitu Puskesmas Sei Agul.



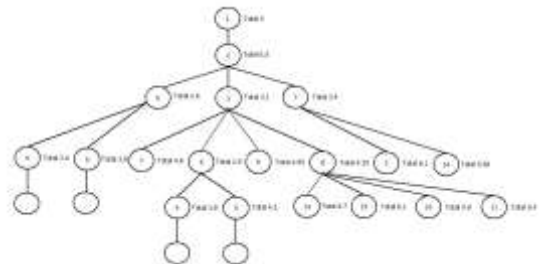
Dari titik 3 maka dikembangkan lagi cabangnya dengan memeriksa titik yang terhubung. Dalam kondisi *tree* yang sekarang terdapat enam titik yang bisa diseleksi yaitu titik 6 dengan total 2.6, titik 7 dengan total 3.4, titik 7 dengan total 4.8, titik 6 dengan total 2.9, titik 9 dengan total 4.95 dan titik 8 dengan total 4.35. Dari keseluruhan titik yang bisa dikembangkan, titik 6 dengan total 2.6 merupakan cabang dengan nilai total terendah. Maka dari titik 6 total 2.6 ini, dilakukan lagi pengembangan.



Dilakukan proses yang sama terhadap cabang dengan nilai total terkecil yaitu titik 7 dengan total 3.4. Dikarenakan titik 4 dan 5 tidak lagi memiliki hubungan terhadap titik lain, maka dapat dikondisikan sebagai berikut.



Dari kondisi *tree* tersebut dapat disimpulkan bahwa titik 8 merupakan titik dengan total terendah, maka dilakukan proses pengembangan.

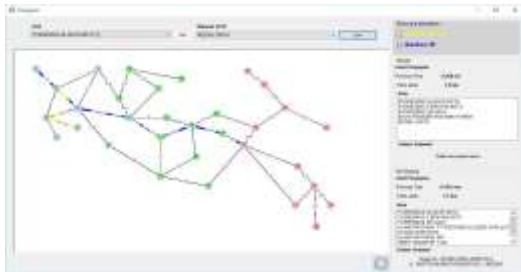


Pada kondisi *tree* tersebut, didapatkan bahwa titik 14 merupakan titik dengan nilai total terkecil, sekaligus titik yang berada pada wilayah tujuan. Dengan demikian didapatkan rute terpendek 1-2-3-8-14 dengan total jarak 4.7 km yaitu Faskes Klinik Diabetes Dharma. Dimana hasil pengujian pada aplikasi didapatkan hasil yang sama dengan proses perhitungan manual.

4.2 Pengujian Algoritma *Greedy*



Proses pencarian Fasilitas Kesehatan dengan menggunakan algoritma *Greedy* dapat dilihat pada Gambar 8.

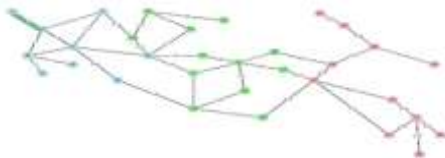


**Gambar 8 Tampilan Pengujian Pencarian Fasilitas Kesehatan dengan Algoritma Greedy (Rute Kuning)**

Algoritma *Greedy* hanya berfokus pada local maximum dimana algoritma ini hanya memilih rute berdasarkan jarak-jarak terkecil yang dilalui bahkan terkadang tidak dapat menemukan hasil yang diharapkan.

Dicontohkan titik awal adalah titik 1 yaitu Puskesmas Glugur Kota dan titik tujuan merupakan fasilitas kesehatan BPJS wilayah Helvetia yang ditandai dengan titik berwarna hijau.

Langkah pertama



Langkah kedua, algoritma ini memilih titik ke - 3 dikarenakan jarak tersebut lebih kecil dari jarak yang dapat dilalui titik ke - 2, dimana terdapat tiga titik yaitu titik 6, 3 dan titik 7.

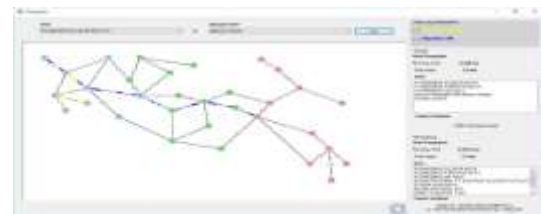


Langkah ketiga, algoritma ini memilih titik ke - 6 dikarenakan jarak tersebut lebih kecil dari jarak yang dapat dilalui titik tujuan lain, dimana titik 9, 8 dan 7 memiliki jarak yang lebih besar.

Langkah Keempat, Dari proses tersebut, algoritma *Greedy* menghasilkan rute 1-2-3-6-4

yang mana tidak berhasil mencapai wilayah Helvetia.

Untuk mencari real running time harus dilakukan proses pengujian sistem dengan algoritma *Hill Climbing* dan algoritma *Greedy* dimana sebelum mengetahui real running timenya user harus memilih titik awal dan titik tujuan yang akan dikunjungi. Akan dilakukan satu kali proses pengujian. Proses pengujian pertama akan dicari Puskesmas Glugur Kota ke Wilayah FKTP Medan Timur dengan menggunakan algoritma *Hill Climbing* dan algoritma *Greedy* dan dapat dilihat pada Gambar 9.



**Gambar 9 Hasil Pengujian Pertama Pencarian FKTP**

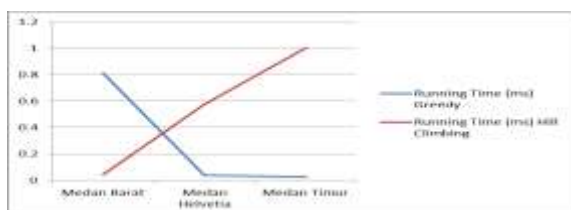
Dari hasil pengujian didapatkan total jarak dengan algoritma *Hill Climbing* dan algoritma *Greedy* yaitu 3,9 km dan 7,5 km serta *execution time* yang didapat algoritma *Hill Climbing* yaitu 0,268 ms dan *execution time* yang didapat algoritma *Greedy* yaitu 0.5016 ms. Jadi untuk percobaan pertama algoritma *Hill Climbing* dan algoritma *Greedy* dengan tujuan yang sama namun jarak dan *execution time* yang berbeda yaitu algoritma *Hill Climbing* lebih cepat dari algoritma *Greedy*.

## 5. HASIL PENGUJIAN

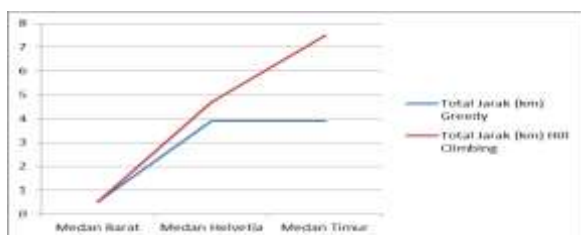
Adapun tujuan dari pengujian ini ialah untuk mengetahui perbandingan dari hasil Algoritma *Greedy* dan *Hill Climbing* dalam mencari tujuan yang sama, sehingga dapat dilihat algoritma mana yang lebih efektif dalam hal pencapaian tujuan dan waktu pencarian tujuan. Perbandingan algoritma yang di amati tersebut dapat di lihat pada Tabel 1.

**Tabel 1 Hasil Pengujian Algoritma Greedy dan Hill Climbing Dengan Tujuan Yang Sama**

No	Faskes Asal	Daerah Tujuan	Running Time (ms)		Total Jarak (km)		Hasil	
			Greedy	Hill Climbing	Greedy	Hill Climbing	Greedy	Hill Climbing
1	Puskesmas Glugur Kota	Medan Barat	0.8128	0.0409	0.5	0.5	Mencapai Tujuan	Mencapai Tujuan
2		Medan Helvetia	0.0384	0.5766	3.9	4.7	Tidak Mencapai Tujuan	Mencapai Tujuan
3		Medan Timur	0.0243	1.0054	3.9	7.5	Tidak Mencapai Tujuan	Mencapai Tujuan



**Gambar 10 Grafik Perbandingan Running Time Algoritma Greedy dan Hill Climbing**



**Gambar 11 Grafik Perbandingan Total Jarak Algoritma Greedy dan Hill Climbing**

Berdasarkan Tabel 1 hasil pengujian dilakukan kepada kedua algoritma dengan FKTP asal menuju tiga daerah tujuan yang berbeda. Yang pertama, kedua algoritma berhasil menuju tujuan dengan tepat dengan jarak tempuh sama akan tetapi *Hill Climbing* lebih cepat mendapatkan hasil dengan *running time* 0.0409 ms. Yang kedua, *Greedy* menghasilkan *running time* dan jarak tempuh lebih kecil daripada *Hill Climbing*, akan tetapi *Greedy* tidak menuju ke daerah tujuan sehingga *Greedy* tidak berhasil mencapai tujuannya. Hal yang sama juga terjadi pada percobaan ketiga. Ditinjau dari hasil pengujian tersebut bahwa semakin jauh jarak yang ditempuh, maka *Greedy* semakin tidak mencapai tujuan, sedangkan *Hill Climbing* mencapai tujuan dengan jarak tempuh yang semakin besar dan memerlukan *running time* yang lebih besar juga daripada *Greedy*.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Hasil pengujian dari pencarian jalan terpendek dapat diselesaikan dengan menggunakan algoritma *Hill Climbing* namun tidak dapat diselesaikan dengan algoritma *Greedy* dikarenakan *Hill Climbing* selalu mencapai tujuan sedangkan *Greedy* tidak mencapai tujuan karena *Greedy* saat melakukan pencarian tidak melihat tujuan hanya melihat jarak terdekat mana yang paling kecil.
2. Dari hasil pengujian pencarian jalan terpendek algoritma *Hill Climbing* memiliki execution time yang lebih cepat dalam pengujian daripada algoritma *Greedy*, dengan tujuan yang sama untuk *Hill Climbing* 0.3948 ms sedangkan *Greedy* 0.8358 ms.
3. Proses pencarian jalan terpendek dengan algoritma *Hill Climbing* dan algoritma *Greedy* menunjukkan bahwa menggunakan algoritma *Greedy* tidak cocok untuk mencari rute terdekat dari tempat tujuan asal mencapai daerah tujuan akhir.

### 6.2 Saran

Saran yang dapat diberikan pada penulis untuk pengembangan dan perbaikan sistem lebih lanjut adalah:

1. Untuk penelitian selanjutnya diharapkan penulis dapat mengimplementasikan algoritma tersebut dapat pencarian jalan terpendek dengan kasus yang berbeda.
2. Sistem yang dirancang diharapkan dapat langsung menunjukkan lokasi melalui google map atau gambar peta yang dapat langsung dilihat oleh user.

## DAFTAR PUSTAKA

- [1] Adamson P., Tick, E. 1991, *Greedy Partitioned Algorithms for the Shortest-Path Problem*. International Journal of Parallel Programming, Vol 20, No. 4.

- [2] Dowsland, Kathryn A. 1991. Hill-Climbing, Simulated Annealing And The Steiner Problem In Graphs, *Engineering Optimization*, 17:1-2, 91-107
- [3] Defindal, dkk.. 2010. Algoritma *Greedy* Untuk Menentukan Lintasan Terpendek. Makalah. Bandung: Institut Teknologi Bandung.
- [4] Hayati, Enty Nur., dkk. 2014. Pencarian Rute Terpendek Menggunakan Algoritma *Greedy*. Makalah. Semarang: Universitas Stukubank Semarang.
- [5] Lubis, Henny Syahriza. 2009. Perbandingan Algoritma *Greedy* dan Dijkstra Untuk Menentukan Lintasan Terpendek. Skripsi. Universitas Sumatera Utara.
- [6] Munir, R. 2010. Matematika Diskrit. Edisi Ke 3. Teknik Informatika : Bandung.
- [7] Munir, R. 2007. Strategi Algoritma . Teknik Informatika: Bandung.
- [8] Pribadi, Feddy Setio, dkk. 2010. Pencarian Rute Terpendek dengan Menggunakan Algoritma Depth First, Breath First dan *Hill Climbing* (Study Comparative). Jurnal. Semarang: Universitas Negeri Semarang.
- [9] Sri Kusumadewi 2003, Artificial Intelligence, Graha Ilmu, Jogjakarta.

