

# Penerapan Metode *Hill Climbing Search* untuk Pencarian Lokasi Terdekat pada Aplikasi Toko *Virtual* Berbasis Android

Muhammad Rezki Firdaus<sup>\*1</sup>, Syarif Indra Halim<sup>2</sup>, Dewi<sup>3</sup>

<sup>1,2,3</sup>STMIK Global Informatika MDP Jl. Rajawali No.14 Palembang

<sup>1,2</sup>PS Teknik Informatika, <sup>3</sup>PS Sistem Informasi STMIK Global Informatika MDP

e-mail: <sup>\*</sup>riezky\_firdauz@yahoo.com, <sup>2</sup>sindrah92@gmail.com, <sup>3</sup>dewi@mdp.ac.id

## Abstrak

Tujuan dari penerapan algoritma *Hill Climbing Search* pada aplikasi ini adalah untuk menghasilkan suatu aplikasi *smartphone* yang dapat mengetahui dan menampilkan posisi user dengan tempat-tempat tertentu. Aplikasi toko *virtual* ini dikembangkan dengan metodologi *Prototype*, dimana pada metodologi ini terdapat beberapa tahapan dalam pelaksanaannya yaitu merencanakan *prototype*, mendesain *prototype*, mengevaluasi *prototype*, membangun sistem, menguji sistem dan mengimplementasi sistem. Aplikasi ini dibuat dengan menggunakan aplikasi *eclipse* merupakan aplikasi pengembangan *Java/Android*, memiliki *plugins* yang dapat membuat *project* yang berbasis *Android ADT (Android Development Tools)* dan menggunakan *JSON (JavaScript Object Notation JavaScript)*. Dalam aplikasi ini terintegrasi dengan layanan *Google Map* dalam penentuan jalur antara user dengan suatu tempat, hal ini memudahkan aplikasi dalam penerapan algoritma yang berhubungan dengan layanan lokasi. Selain itu, keakuratan posisi user yang diterima *GPS Android* bisa meleset beberapa meter dari posisi user sekarang berada tergantung perangkat *hardware GPS* di *handset* user gunakan. Penerapan metode *Hill Climbing Search* sepenuhnya menggunakan informasi dari server yang telah disediakan, sehingga pemberian data kepada user sepenuhnya tergantung pada ketersediaan server, jika server tidak hidup maka aplikasi tidak dapat diakses oleh user.

**Kata kunci**—*Android, Algoritma Hill Climbing Search, Lokasi Tedekat, Prototype*

## Abstract

The purpose of implementation *Hill Climbing Search* algorithm in this application is to produce a *smartphone* application that can determine and display the user's position with certain places. This *virtual store* application is developed by using *Prototype* methodology, there are several stages in its implementation, they are plan *prototype*, desain *prototype*, evaluation *prototype*, build system, test system and implementation system. This application is made by using *Eclipse* application which is an application development *Java / Android*, has *plugins* that can create a *project* based on *Android ADT (Android Development Tools)* and use *JSON (JavaScript Object Notation JavaScript)*. This application used *GoogleMap* services for determining the path between the user with a place, it is easy for application in implementation algorithm which associated with the location service. Beside that, the accuracy of user's position that accepted *GPS Android* can miss a few meters away from the current user's position according the *GPS hardware device* in user's *handset*. The implementation of *Hill Climbing Search* algorithm fully uses the information from a server that has been provided, so the provision of data to the user fully dependent on the availability of server, if server is not alive then the application can 't be accessed by user.

**Keywords**—*Android, Hill Climbing Search Algorithm, Nearest Location, Prototype*

## 1. PENDAHULUAN

Pengaruh dan peran TI terhadap perkembangan bisnis *online* di Indonesia sangatlah besar. Contohnya dengan adanya pembuatan *website*, forum dan blog sebagai media yang bersifat sebagai jembatan antara penjual dan pembeli untuk memberi informasi sekaligus bertransaksi semakin hari semakin banyak. Hal ini menunjukkan perkembangan situs-situs yang memiliki wadah untuk bisnis *online* semakin berkembang setiap waktunya.

Semakin mudahnya bisnis *online* melalui fasilitas internet mendorong banyak orang menjalankan bisnis *online* ini, namun dengan terbatasnya pengetahuan mengenai bisnis *online* kebanyakan orang sulit mempublikasikannya. Seringkali juga kita mendapatkan informasi palsu yang banyak terdapat di beberapa *website*, forum dan layanan sosial. Ini sangat merugikan konsumen dalam melakukan transaksi dalam bisnis *online*.

Menentukan lokasi suatu toko atau daerah tempat produsen menjual produk barangnya sering kali menjadi suatu hambatan dalam mendapatkan lokasi aslinya. Seringkali hanya lokasi provinsi saja yang diberikan kepada para konsumen dalam mendapatkan informasi di beberapa *website*, forum dan layanan sosial. Dengan menampilkan informasi tentang lokasi tempat produsen menjual barangnya, konsumen dapat menuju ke lokasi produsen tanpa harus berkomunikasi dengan para produsen. Sehingga ini mengurangi tingkat kecurigaan terhadap informasi yang telah didapatnya.

Metode *Hill Climbing Search* adalah metode yang menentukan node-node yang telah diberi jarak antar node yang lain dengan membandingkan dengan node yang telah ada berdasarkan pemilihan jarak terdekat dari posisi sekarang. Kelebihannya tidak perlu memilih node yang telah diuji untuk dibandingkan lagi sehingga tidak memakan waktu pemrosesannya. Kekurangannya tidak mengetahui jarak node yang tidak diuji sebelumnya sehingga tidak bisa dibandingkan pada hasil yang dicapai[6].

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Merancang suatu aplikasi berbasis Android pada perangkat *mobile phone* untuk memberikan informasi yang akurat, mudah dan cepat mengenai lokasi penjual barang beserta keterangannya.
2. Menerapkan metode *Hill Climbing Search* untuk mencari lokasi terdekat pada *mobile phone*.

Batasan dalam penelitian ini yaitu sebagai berikut:

1. Calon produsen bisa mendaftar terlebih dahulu sebelum menampilkan barang yang akan dijual pada toko *virtual* beserta lokasinya yang diambil berdasarkan lokasi produsen saat mendaftar.
2. Pencarian barang menggunakan fitur GPS akan menampilkan lokasi toko *virtual* penjual barang yang dicari berdasarkan lokasi terdekat.
3. Lokasi produsen yang menjual barang dapat dilihat melalui aplikasi dengan fitur GPS.

Teori-teori pendukung penelitian ini yaitu sebagai berikut:

### 1.1 Teknologi Pencarian

Perkembangan teknologi perangkat *mobile phone* cukup pesat setiap tahunnya. Sebelumnya, *mobile phone* hanya difungsikan sebagai alat komunikasi suara dan alat pengiriman pesan saja. Dengan semakin bertambahnya keutuhan akan fitur-fitur baru untuk *mobile phone* memuat para *vendor* mengembangkan teknologi-teknologi baru untuk setiap produk mereka. Salah satu teknologi yang diaplikasikan di perangkat *mobile phone* saat ini yaitu teknologi *Global Positioning System* (GPS). Dengan memanfaatkan GPS, pengguna dapat mengetahui posisi keberadaannya secara *real time*[2].

*Location-Based Service* (LBS) memanfaatkan teknologi GPS dalam pengaplikasiannya. Selain dapat mengetahui posisi pengguna, aplikasi LBS juga dapat menentukan posisi tempat-

tempat tertentu. Dan dengan kombinasi ini, aplikasi LBS akan mencari rute untuk menghubungkan posisi pengguna dengan suatu tempat [2].

### 1.2 Global Positioning System (GPS)

GPS merupakan nama sebuah sistem navigasi global berbasis satelit (GNSS= Global Navigation Satellite System) yang dikembangkan oleh Departemen Pertahanan Amerika Serikat. Tetapi karena sistem ini adalah yang pertama kali serta satu-satunya di dunia yang berfungsi secara penuh saat ini dan dapat digunakan setiap saat oleh semua orang di dunia secara gratis, maka nama GPS menjadi terkenal dan sering dipakai sebagai nama alat navigasi berbasis satelit. Sistem ini menggunakan kelompok satelit yang diberi nama NAVSTAR (*Navigational Satellite Timing and Ranging*) [1].

### 1.3 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi pengembang untuk menciptakan aplikasi baru karena sistem operasi yang berbasis *open source*[5].

Android merupakan generasi baru *platform* yang memberikan banyak fasilitas pengembangan yang bebas terbuka. Arsitektur dari Android sendiri antara lain [5]:

1. *Applications and Widgets*
2. *Application Frameworks*
3. *Libraries*
4. *Android Run Time*
5. *Linux Kernel*

#### 1.3.1 Android Software Development Kit

Android SDK (*Software Development Kit*) menyediakan *tools* dan API (*Application Programming Interface*) yang diperlukan untuk aplikasi pengembangan *platform* Android dengan menggunakan bahasa pemrograman Java [3].

#### 1.3.2 Android Development Tool

Android *Development Tool (ADT)* adalah *plugin* yang didesain untuk IDE *eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE *eclipse*. Dengan menggunakan ADT untuk *eclipse* akan memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui *eclipse*. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang dirancang [5].

#### 1.3.3 Versi Sistem Operasi Android

Menurut Nazaruddin Safaat H [5], adapun versi-versi Android yang pernah dirilis adalah sebagai berikut :

1. Android versi 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*)
6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0 (*Honeycomb*)
8. Android versi 4.0 (*Ice Cream Sandwich*)
9. Android versi 4.1 (*Jelly Bean*)

#### 1.4 Eclipse

*Eclipse* adalah IDE (*Integrated Development*) merupakan aplikasi pengembangan Java/Android, *eclipse* memiliki *plugins* yang dapat membuat *project* yang berbasis Android ADT (*Android Development Tools*) [5].

#### 1.5 MySQL

MySQL merupakan sebuah program pembuat dan pengelola database atau yang sering disebut dengan DBMS (*Database Management System*). Sifat dari DBMS ini adalah *open source*. MySQL sebenarnya produk yang berjalan pada *platform Linux*, dengan adanya perkembangan dan banyaknya pengguna serta lisensi dari *database* ini adalah *open source* maka para pengembang merilis versi *Windows*[4].

#### 1.6 Metode Hill Climbing Search

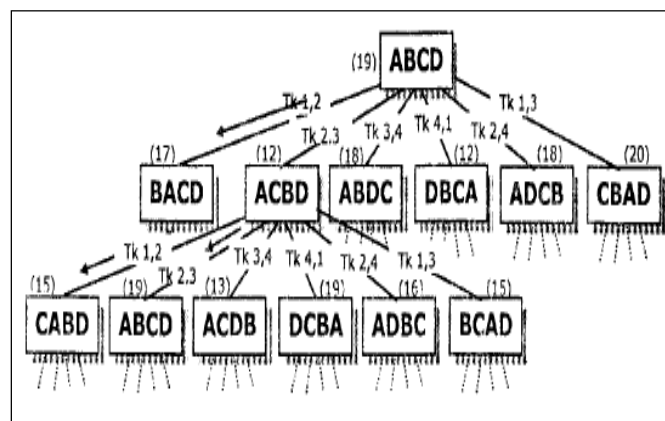
Metode ini hampir sama dengan metode pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin[6].

Ada dua macam metode *Hill Climbing Search*, yaitu *Simple Hill Climbing* dan *Steepest-ascent Hill Climbing*[6].

##### 1.6.1 Simple Hill Climbing Search

Algoritma untuk *Hill Climbing Search* adalah sebagai berikut [6] :

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kerjakan 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 baru.
  - b. Evaluasi keadaan baru tersebut.
    - Jika keadaan baru merupakan tujuan, keluar.
    - Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
    - Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan pencarian.



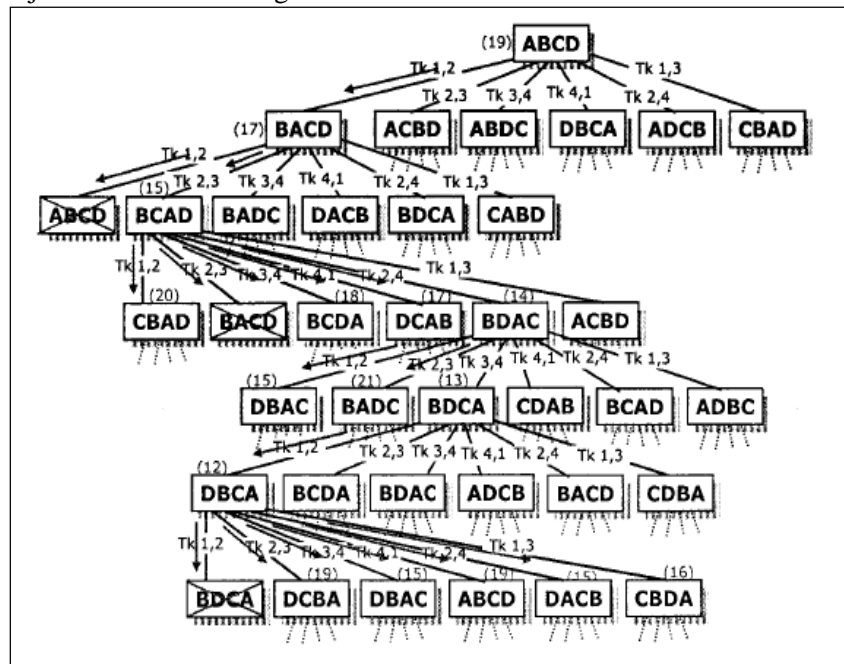
Gambar 1 *Simple Hill Climbing*

### 1.6.2 Steepest Ascent Hill Climbing Search

*Steepest-Ascent Hill Climbing Search* hampir sama dengan *Simple Hill Climbing Search* dan yang membedakan keduanya adalah pada gerakan pencarian yang tidak dimulai dari posisi paling kiri. Gerakan berikutnya dicari berdasarkan nilai heuristik terbaik. Dalam hal ini urutan penggunaan tidak menentukan penemuan solusi [6].

Adapun algoritma untuk *Steepest-Ascent Hill Climbing Search* adalah [6]:

1. Mulai dari keadaan awal, lakukan pengujian. Jika merupakan tujuan maka berhenti dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Ulangi hingga tujuan tercapai atau hingga pencarian tidak memberikan perubahan pada keadaan sekarang.
  - a. Tentukan SUCC sebagai nilai heuristik dari *successor-successor*.
  - b. Lakukan untuk tiap node yang digunakan oleh keadaan sekarang.
    - Gunakan node tersebut dan bentuk keadaan baru.
    - Evaluasi keadaan baru tersebut jika merupakan tujuan keluar. Jika bukan, bandingkan nilai keuristiknya dengan SUCC. Jika lebih baik, jadikan nilai heuristik keadaan baru tersebut sebagai SUCC, tetapi jika tidak lebih baik, nilai SUCC tidak berubah.
    - Jika SUCC lebih baik daripada nilai heuristik keadaan sekarang, ubah node SUCC menjadi keadaan sekarang.



Gambar 2 *Steepest Ascent Hill Climbing*

## 2. METODE PENELITIAN

Metode yang digunakan dalam pengembangan aplikasi ini menggunakan *prototype*. Dimulai dari hal analisis kebutuhan sistem, tahap perancangan sampai tahap implementasi semuanya menggunakan metode ini. *Prototype* membagi tahapan sebagai berikut :

1. Merencanakan *Prototype*  
 Pada tahapan ini, penulis melakukan perencanaan awal yaitu menentukan tujuan umum, manfaat, serta ruang lingkup. Tahapan ini meliputi mengumpulkan data-data mengenai

metode *Hill Climbing Search* ataupun informasi yang dibutuhkan dalam pengembangan aplikasi pada *mobile phone* yang akan dibuat.

2. Mendesain *Prototype*

Pada tahap ini, penulis menganalisis alur sistem dan membuat rancangan sementara tentang sistem informasi dan metode *Hill Climbing Search* dalam penerapan dalam proses pencarian jarak pada *mobile phone* dengan menggunakan GPS dari *google maps*.

3. Mengevaluasi *Prototype*

Pada tahap ini, penulis melakukan analisis kembali tahap perencanaan *prototype* dan tahap desain *prototype*, apabila kebutuhan tersebut dapat dipenuhi, maka proses akan lanjut ke tahap membangun, apabila tidak maka *prototype* direvisi dengan mengulang kembali tahap perencanaan dan tahap desain *prototype*.

4. Membangun Sistem

Pada tahap ini, setelah *prototype* yang sudah dirancang dan dievaluasi maka *prototype* dibangun untuk menjadi sistem atau aplikasi dengan memberikan kode program yang sesuai. Setelah tahap desain sementara selesai dikerjakan maka tahapan selanjutnya penulis kembali mengembangkan aplikasi ke dalam *mobile phone* yang sudah ditentukan sesuai kebutuhan *user* berdasarkan GUI, membuat simulasi server dengan menggunakan MYSQL pada XAMPP, menambahkan GPS *mobile phone* pada proses pencarian terdekat menggunakan metode *Hill Climbing Search*, merancang tampilan aplikasi menggunakan photoshop CS, serta melakukan pengkodean program desain aplikasi dengan menggunakan ADT (*Android Development Tools*) dan Android SDK.

5. Menguji Sistem

Sebelum masuk ke dalam proses pengujian aplikasi penulis harus mendaftar terlebih dahulu ke *Google API* untuk mendapatkan *API key* android. *API key* ditanam ke dalam program sebagai jalur akses ke *google maps* pada saat GPS program dijalankan. Penulis juga membuat node-node pada *google maps* untuk setiap persimpangan jalan yang ada dalam peta. Node-node yang telah disimpan kemudian akan ditampilkan sebagai jalur atau arah yang akan dilalui oleh metode *Hill Climbing Search*. Node-node tersebut digunakan sebagai perhitungan jarak terpendek dari lokasi sekarang ke tujuan. Pada proses pengujian sistem perangkat keras yang digunakan harus memiliki akses internet dimana semua aktifitas aplikasi menggunakan jalur internet.

6. Mengimplementasi Sistem

Setelah selesai melakukan pengujian, penulis melakukan pengimplementasian sistem pada daerah yang telah diberi node-node sebelumnya. Dalam proses pencarian lokasi terdekat penulis melakukan uji coba pada daerah yang memiliki banyaknya persimpangan pada peta. Pada daerah yang tidak memiliki persimpangan atau jalan penulis tidak dapat memberikan node pada daerah tersebut. Dikarena tidak ditampilkannya jalur secara *objectif* dari layanan *google maps* sendiri.

### 3. HASIL DAN PEMBAHASAN

Bagian ini berisi tentang langkah-langkah pengembangan aplikasi, yang terdiri dari lingkungan pengembangan aplikasi, analisis pencarian terdekat Algoritma *Hill Climbing Search*, Flowchart alur *Hill Climbing Search* pada program, implementasi dan analisis.

### 3.1 Lingkungan Pengembangan Sistem

Aplikasi yang dibuat adalah aplikasi yang berbasis Android. Aplikasi ini hanya dapat digunakan pada ponsel atau smartphone yang berbasis Android. Dalam pengembangan program aplikasi ini, jenis perangkat lunak dan perangkat keras yang digunakan adalah :

#### 3.2.1 Jenis Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan pada komputer sebagai berikut :

1. Windows 8.1 Pro
2. Eclipse
3. ADT (*Android Development Tool*)
4. SDK (*Software Development Kit*)
5. Photoshop CS
6. Notepad++

#### 3.2.2 Jenis Perangkat Keras yang Digunakan

Perangkat keras yang digunakan disajikan pada Tabel 1.

Tabel 1 Spesifikasi *Hardware*

<i>Hardware</i>	Komputer Laptop	<i>Handphone</i>
<i>Processor</i>	Intel Core i3 2,3Hz	1 GHz <i>Scorpion</i>
<i>RAM</i>	3,00 GB	512 MB
<i>Harddisk</i>	500 GB	400 MB

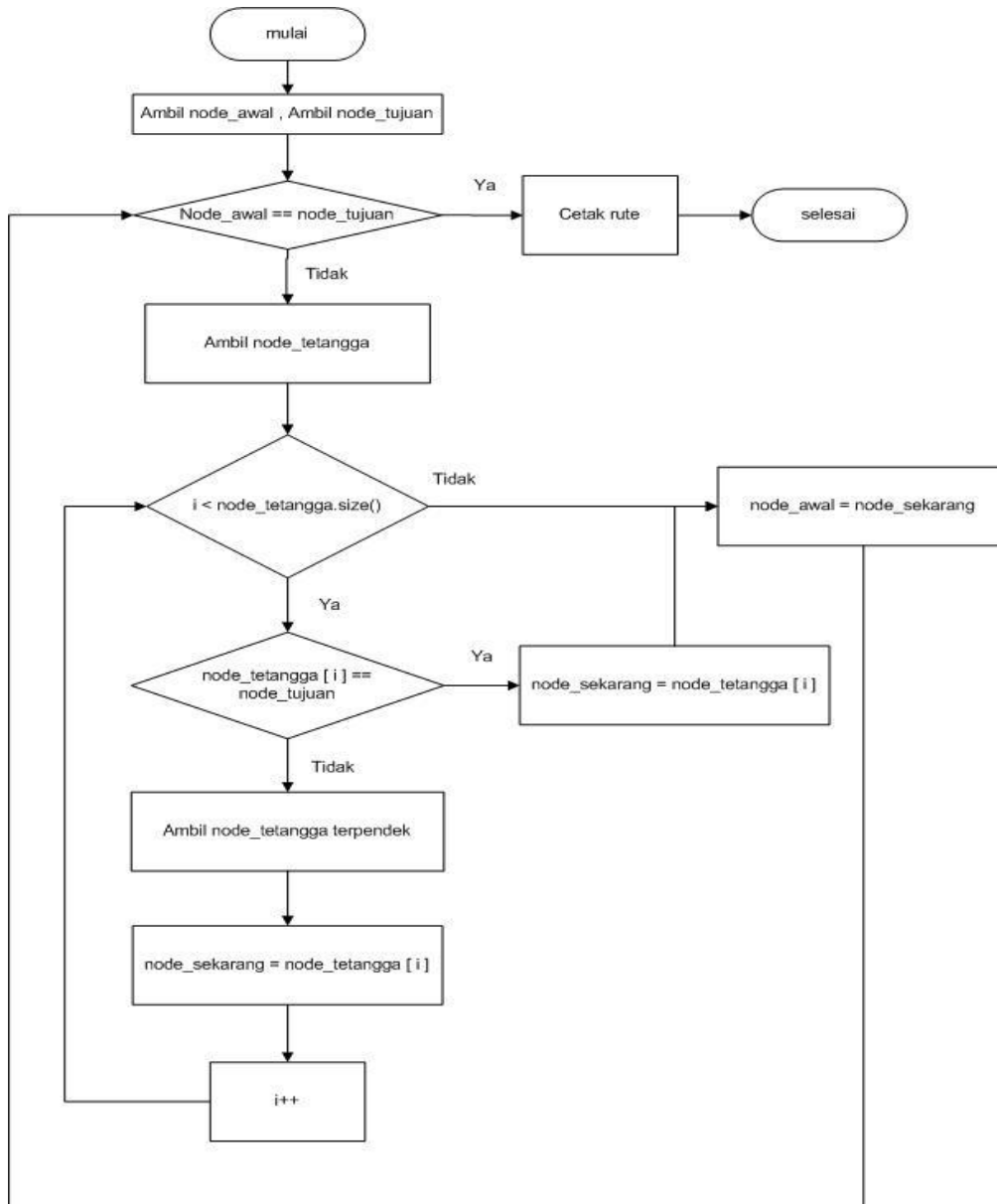
### 3.2 Analisis Pencarian Terdekat Algoritma *Hill Climbing Search*

*Hill Climbing Search* merupakan salah satu dari pencarian heuristik yang digunakan untuk pencarian atau perbandingan jarak pada aplikasi. Metode *Hill Climbing Search* terdiri dari dua jenis yaitu *Simple Hill Climbing Search* dan *Steepest-Ascent Hill Climbing*, pada penelitian ini digunakan metode *Steepest-Ascent Hill Climbing Search* untuk mencari jalur terpendek dari pencarian lokasi dikarenakan pencariannya dimulai dari nilai heuristik terbaik sehingga tidak menghabiskan memori yang banyak [6]

Dalam membuat jalur pencarian terdekat menggunakan metode *Hill Climbing Search* diperlukan node atau titik sebagai pembanding jaraknya [6]. Node-node didapat dari jalur yang memiliki dua atau lebih persimpangan jalur. Node-node sebelumnya disimpan terlebih dahulu sehingga dapat memberikan tingkat keakuratan dari jalur tersebut.

Dalam proses penerapannya, penulis membutuhkan lokasi dari setiap pengguna dan node untuk dijadikan sebagai jalur yang akan dibandingkan jaraknya. Jarak node yang telah didapat akan dibandingkan selalu dengan node yang lain sampai mendapatkan nilai terbaik dari node-node yang lain. Apabila dalam proses menemukan node yang terbaik dari yang sebelumnya maka posisi dari node awal akan berubah. Proses akan dilakukan sampai node yang diduduki adalah node tujuan. Node yang telah dilewati tidak akan dibaca lagi.

### 3.3 Flowchart Alur Hill Climbing Search pada Program



Gambar 3 Flowchart Alur Hill Climbing Search

### 3.4 Implementasi dan Analisis

#### 3.4.1 Keunggulan Program

1. Aplikasi dapat memberikan informasi yang lengkap tentang produsen yang menjual barangnya.
2. Aplikasi ini dilengkapi dengan pengolahan data barang dan toko yang relatif mudah digunakan oleh pengguna.
3. Aplikasi ini dapat menampilkan lokasi toko pengguna dari GPS Android yang telah terinstall.
4. Aplikasi ini dapat menampilkan lokasi terdekat dari toko produsen yang menjual barang dari hasil pencarian barang.
5. Aplikasi ini juga dapat diakses kapan saja dan dimana saja bagi pengguna *smartphone* Anroid.



### 3.4.2 Navigasi Diagram

Untuk tampilan antar muka aplikasi ini disajikan pada navigasi diagram. Navigasi diagram digunakan sebagai alur dari suatu aplikasi yang berfungsi untuk menggambarkan dengan sangat jelas mengenai hubungan rantai antara *layout* aplikasi (Gambar 4)



Gambar 4 Navigasi Diagram

### 3.4.3 Pengujian Metode Hill Climbing Search pada GPS Android

Untuk pengujian akan digunakan tingkat keakurasian dengan membandingkan jumlah ketepatan lokasi terpendek dari *Hill Climbing Search* dengan jumlah lokasi. Perhitungan tingkat keakurasian sebagai berikut :

$$\text{Akurasi} = \frac{\text{Jumlah Ketepatan Lokasi Terpendek}}{\text{Jumlah lokasi}} \times 100\%$$

Tabel 2 Pengujian Penerapan Metode *Hill Climbing Search*

No	Lokasi Awal	Lokasi Tujuan	Jarak HC	Jarak Google Map
1.	-2.95930061, 104.78015778	-2.963356, 104.780063	588m	600m
2.	-2.95930061, 104.78015778	-2.965713, 104.780749	1 km	1.1 km
3.	-2.95930061, 104.78015778	-2.962177, 104.774634	838m	850m
4.	-2.967084, 104.78015778	-2.963356, 104.780063	598m	650m
5.	-2.967084, 104.78015778	-2.969324, 104.77794	407m	500m
6.	-2.966849, 104.783689	-2.969099, 104.78208	456m	456m
7.	-2.966849, 104.783689	-2.963356, 104.780063	812m	750m
8.	-2.974177, 104.76427	-2.97287, 104.760794	1.25 km	1.1 km
9.	-2.974177, 104.76427	-2.968852, 104.766364	972m	800m
10.	-2.970984, 104.765557	-2.969827, 104.762832	296m	300m
Akurasi			$= \frac{7}{10} \times 100\% = 70\%$	

#### 4. KESIMPULAN

Beberapa kesimpulan yang dapat ditarik dari penelitian ini adalah:

1. Keakuratan posisi *user* yang diterima GPS Android bisa meleset beberapa meter dari posisi *user* sekarang berada tergantung perangkat *hardware* GPS di *handset* *user* gunakan.
2. Penerapan metode *Hill Climbing Search* sepenuhnya menggunakan informasi dari *server* yang telah disediakan sehingga pemberian data kepada *user* sepenuhnya tergantung pada ketersediaan *server*, jadi jika *server* tidak hidup maka aplikasi tidak dapat diakses oleh *user*.

#### 5. SARAN

Adapun saran yang dapat disampaikan untuk penelitian ini antara lain:

1. Diharapkan untuk kedepannya aplikasi ini dapat menampilkan jalur tercepat berdasarkan waktu dan tingkat kemacetan dari jalan raya.
2. Diharapkan lebih lanjut penerapan metode *Hill Climbing Search* dapat menampilkan jalur yang berbeda sehingga pengguna dapat memilih sendiri rute yang ingin dilalui.

#### DAFTAR PUSTAKA

- [1] Andre, 2009, *GPS bagi Pemula*, Jakarta.
- [2] B.R.Rompas, 2011, *Aplikasi Location-Based Service Pencarian Tempat di Kota Manado Berbasis Android*, Jurusan Teknik Electro-FT UNSRAT, Manado.
- [3] Mulyadi, 2010, *Membuat Aplikasi untuk Android Multimedia Center Publishing*, Yogyakarta.
- [4] Nugroho, Bunafit, 2009, *Latihan Membuat Aplikasi Web PHP dan MySQL dengan Dreamweaver*, Gava Media, Yogyakarta.
- [5] Sfaat H, Nazaruddin, 2012, *Pemograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android*, Informatika, Bandung.
- [6] Sri Kusumadewi, 2003, *Artificial Intelligence*, Graha Ilmu, Jogjakarta.