

Pencarian Rute Terpendek dengan Menggunakan Algoritma Depth First, Breath First dan Hill Climbing (Study Comparative)

Feddy Setio Pribadi & Anggraini Mulwinda
Jurusan Teknik Elektro, Universitas Negeri Semarang
Feddystio@yahoo.com, airy1808@yahoo.com

Abstrak : Pencarian rute terpendek saat melakukan perjalanan merupakan hal yang perlu dilakukan selain menemukan kota tujuan juga untuk menghemat biaya perjalanan. Algoritma pencarian (*searching algorithm*) yang mendasari kerja dari software atau situs banyak modelnya, akan tetapi keefektifan suatu algortima pencarian dalam menemukan rute atau tujuan tergantung pada proses atau langkah-langkah yang di berikan oleh algortima itu sendiri. Penelitian ini akan memfokuskan pada penerapan 3 algoritma pencarian dalam penentuan rute terpendek yang paling optimum untuk dicapai. Hasil penelitian didapatkan bahwa Algoritma terbaik untuk mendapatkan rute terpendek dan paling efektif yang diterapkan pada jalur transportasi yang tersedia adalah Algoritma Breadth First dan Algoritma Hill Climbing.

Kata Kunci : Algoritma Depth First, Breath First , Hill Climbing.

1. Pendahuluan

1.1. Latar Belakang

Pencarian rute terpendek saat melakukan perjalanan merupakan hal yang perlu dilakukan selain menemukan kota tujuan. Alasan pencarian rute terpendek adalah meringkas perjalanan dan menghemat biaya perjalanan. Persolan lain dalam melakukan perjalanan yang efektif adalah penelusuran kota yang dilakukan oleh seorang *sales*, dimana seorang *sales* tersebut harus mengunjungi beberapa tempat untuk mendistribusikan barang, sehingga dia hanya akan sekali singgah di tempat tersebut sampai tempat terakhir yang akan dituju tercapai hingga kembali ke tempat asal.

Saat melakukan perjalanan ke tempat tujuan sering kali seseorang membawa peta sebagai petunjuk jalan. Penggunaan peta dalam bentuk ini secara visual mampu menggambarkan rute yang akan ditempuh dari kota asal ke kota tujuan. Pemakain peta kertas mempunyai kendala secara visual harus dapat mengurutkan rute mana yang akan di tempuh, salain itu penggunaan peta jenis ini tidak

memungkinkan untuk dapat memberikan suatu saran rute mana yang paling efektif yang dapat di lalui.

Perkembangan teknologi informasi memungkin penerapan sistem informasi dengan memasukan informasi keruangan. Dengan adanya informasi keruangan maka pencari informasi tidak hanya di suguh huruf dan angka akan tetapi visualisasi tempat dimana informasi tersebut dihasilkan. Peranan penerapan algoritma pencarian rute merupakan salah satu bagian yang membuat pencarian rute tervisualisasi melalui komputer ataupun *portable mobile comunication* marak dikembangkan. Beberapa perangkat lunak (*software*) routing dapat kita download secara gratis, sebagai contoh aplikasi situs map.google.com juga menyediakan layanan pencarian tempat dan rute, situs ini mencoba menyediakan layanan dinamis dimana pengguna dapat melakukan pencarian rute yang harus dilalui untuk sampai ke suatu tujuan.

Algoritma pencarian (*searching algorithm*) yang mendasari kerja dari software atau situs banyak modelnya, akan tetapi

kefektifan suatu algortima pencarian dalam menemukan rute atau tujuan tergantung pada proses atau langkah-langkah yang di berikan oleh algortima itu sendiri, sehingga ada algoritma tertentu yang sesuai untuk penacrian rute terpendek ada juga algoritma tertentu yang sesuai untuk pencarian perjalanan yang paling efektif dan efisien.

Dari latar belakang di atas muncul suatu permasalahan bahwa dalam melakukan suatu perjalanan hendaknya dapat menemukan suatu rute yang paling efektif dan efisien sehingga diharapkan dapat menekan biaya selama dalam perjalanan.

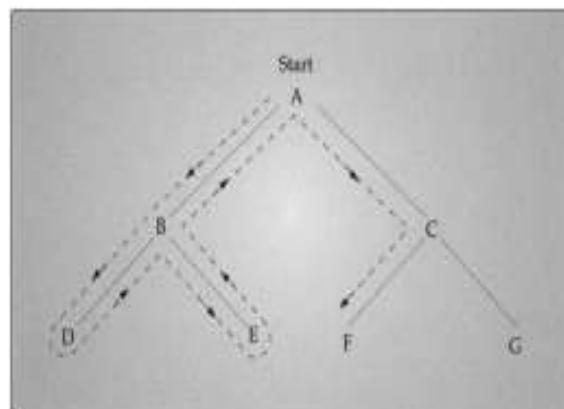
1.2. Tujuan

Penelitian ini bertujuan untuk menemukan rute yang efektif dan efisien dengan penerapan algortima pencarian yang tepat sehingga rute yang disarankan akan benar-benar menjadi rute yang terbaik.

Penelitian ini akan memfokuskan pada penerapan 3 algoritma pencarian rute yang diterapkan pada penentuan rute terpendek.

1.3. Tinjauan Pustaka

1.3.1. Metode Depth First



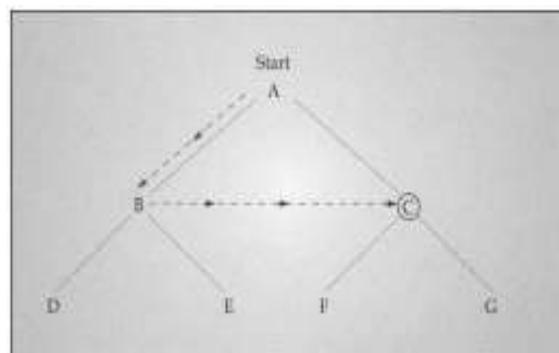
Gambar 1. Proses penelusuran dengan metode Depth First

Metode *Depth First search* mengeksplor setiap kemungkinan cabang yang mungkin akan menjadi sebuah solusi sebelum mengeksplor ke cabang yang lain. Pada gambar di atas menunjukkan suatu pencarian dengan metode *Depth First search* dimana F merupakan titik tujuan.

Pencarian dengan metode *Depth First* memcoba melintasi keseluruhan graf dengan urutan ABDBEBACF. Metode ini hampir mirip dengan model penelusuran dengan metode pohon. Penelusuran diawali dengan cabang yang kiri setiap node di lalui hingga mencapai node yang terakhir, jika node yang terakhir sudah dilalui akan tetapi belum menemukan titik tujuan maka penelusuran kembali melalui ke titik pangkal yaitu titik A. Pencarian dilanjutkan dengan menelusuri cabang sebelah kanan dengan menelusuri setiap node hingga menemukan titik tujuan. Prosedur ini diulang-ulang hingga setiap cabang dilalui dan sampai menemukan titik tujuan.

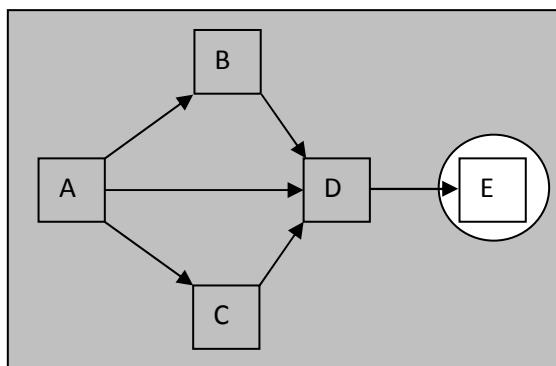
1.3.2. Metode Breadth First

Metode pencarian Breadth First merupakan kebalikan dari cara pencarian titik tujuan pada metode *Depth First*. Pada metode Breadth First penelusuran setiap titik dilakukan pada kedudukan titik pada level yang sama, seperti terlihat pada gambar, tujuan dari pencarian adalah menemukan titik C. Pencarian diawali dengan menelusuri titik pangkal yaitu titik A, karena titik A merupakan titik dengan level tertinggi maka penelusuran dilanjutkan dengan menjelajahi titik-titik pada level di bawahnya. Penelusuran selanjutnya adalah pada level kedua yaitu diawali dengan titik B, karena pada level ini terdapat titik lain yang selevel maka penelusuran dilanjutkan dengan menjelajahi titik C, dalam diagram ini titik C merupakan titik tujuan, maka dengan demikian proses pencarian selesai pada level dua.



Gambar 2. Proses penelusuran dengan metode Breadth First

1.3.3. Metode Hill Climbing



Gambar 3. 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 salalu akan mencari jalan yang lebih pintas untuk mencapai tujuannya. Pada Gambar 3 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 dipilih pada metode Hill Climbing akan dibandingkan ketiga jalur tersebut mana yang paling sedikit cost yang arus dikeluarkan, apakah rute yang pling pendek ataupun tingkat kemacetan yang paling kecil, pemilihan akan bergantung pada informasi yang diberikan pada peta yang akan dilalui.

1.3.4. Rute Terpendek

Perjalanan dengan mempertimbangkan penelusuran rute terpendek hanya mempertimbangkan jarak atau cost antar kota yang harus dilalui. Seperti ditunjukkan pada Gambar 3 Untuk menuju ke kota E dari kota asal yaitu kota A, algoritma pencarian yang efektif akan membandingkan jarak yang harus ditempuh jika melalui kota B kemudian baru ke kota E dengan jarak yang dilalui jika perjalanan dilakukan dari kota A lansung menuju ke kota E.

Pencarian rute terpendek hanya semata-mata mempertimbangkan jarak terpendek yang harus dilalui, pada kasus Gambar 3 Suatu metode bisa jadi akan memilih

menelusuri kota A ke kota D untuk mencapai tujuan akhir kota E, karena dengan pertimbangan perbandingan jarak tempuh dari kota A ke kota B lebih dekat daripada jarak yang harus ditempuh dari kota A ke kota D.

2. Bagian Inti

2.1. Metode

Pada penelitian ini ketiga metode terlebih dahulu akan di uji untuk menyelesaikan persoalan Traveling Salesman Problem. Pengujian yang kedua yaitu ketiga metode akan menyelesaikan persoalan pencarian rute terpedek yang harus di tempuh dari kota asal ke kota tujuan. Setelah selesai pengujian kemudian akan dilakukan analisis terhadap ketiga metode tersebut yang selanjutnya akan ditentukan performa terbaik yang ditujukan ketiga metode tersebut dalam menyelesaikan persoalan pencarian rute terpendek yang menjadi contoh kasus.

Penelitian ini mengambil contoh kasus perjalanan transportasi darat dengan menggunakan jalur Bus. Pada simulasi jalur bus yang digunakan adalah jalur transportasi yang ada antara kota pekalongan menuju ke kota yogyakarta.

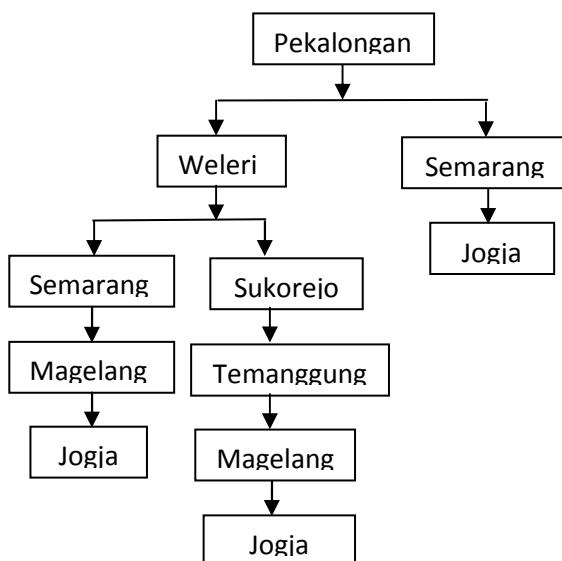
Adapun jalur trasportasi yang tersedia antara 2 kota tersebut adalah sebagai berikut :

Jalur 1 : Pekalongan – Weleri – Semarang – Magelang – Yogyakarta

Jalur 2 : Pekalongan – Weleri – Sukorejo – Temanggung – Magelang – Jogja

Jalur 3 : Pekalongan – Semarang – Jogja

Dari ketiga jalur tersebut selanjutnya dibentuk sebuah pohon keputusan (*decesion tree*) untuk mengetahui struktur pencarian, dari pohon keputusan tersebut kemudian di susun sebuah struktur data yang dapat dipahami oleh program komputer. Berikut adalah bentuk pohon keputusan yang berhasil di buat.



Gambar 4. Pohon Keputusan Rute Perjalanan Pekalongan – Jogja

2.2. Hasil dan Pembahasan

2.2.1. Hasil

Pohon keputusan disusun berdasarkan kasus yang ada, akan tetapi dalam membentuk pohon keputusan juga diperlukan analisis tentang struktur data yang nantinya akan diterapkan dalam bahasa pemrograman.

Dari pohon keputusan di atas selanjutnya dibuat susun struktur data yang diterjemahkan kedalam bahasa pemrograman, pada penlitian ini peneliti menggunakan bahasa C untuk mengembangkan program routing. Berikut adalah struktur data dari pohon keputusan:

```

void setup(void)
{
assert_flight("pekalongan", "weleri", 50);
assert_flight("weleri", "semarang", 40);
assert_flight("semarang", "magelang", 60);
assert_flight("magelang", "jogja", 50);
assert_flight("weleri", "sukorejo", 30);
assert_flight("sukorejo", "temanggung", 50);
assert_flight("temanggung", "magelang", 60);
assert_flight("pekalongan", "semarang", 110);
assert_flight("semarang", "jogja", 110);
}
  
```

Gambar 5. Struktur Data Percobaan 1

Asert_flight adalah fungsi yang digunakan untuk menyimpan data kota dan jarak antar kota, berikut adalah struktur dari fungsi asert_flight.

`Asert_flight("Kota A",Kota B",Jarak A – B)`

Dari beberapa percobaan dalam membentuk struktur data yang berada dari pohon keputusan didapatkan struktur data yang lain yaitu sebagai berikut.

```

void setup(void)
{
assert_flight("pekalongan", "weleri", 50);
assert_flight("weleri", "sukorejo", 30);
assert_flight("sukorejo", "temanggung", 50);
assert_flight("temanggung", "magelang", 60);
assert_flight("weleri", "semarang", 40);
assert_flight("semarang", "magelang", 60);
assert_flight("magelang", "jogja", 50);
assert_flight("pekalongan", "semarang", 110);
assert_flight("semarang", "jogja", 110);
}
  
```

Gambar 6. Struktur Data Percobaan 2

Pada struktur data 1, proses pembentukan struktur data berdasarkan penelusuran yang mendalam pada satu cabang kemudian dilanjutkan dengan cabang – cabang yang lain. Struktur data 2 proses pembentukan struktur data didasarkan pada pembentukan cabang – cabang yang ada (level 1), kemudian dilanjutkan pembentukan yang lebih mendalam pada cabang – cabang yang ada (level 2 dst).

Pada pengembangan 2 struktur data di atas didapatkan hasil bahwa setiap algoritma pencarian akan menghasilkan hal yang berbeda yang tergantung dari susunan struktur data yang dibentuk ketika diterapkan pada masing – masing algoritma. Gambar dibawah merupakan hasil yang didapatkan ketika 3 algoritma pencarian yaitu Depth First, Breadth First dan hill climbing dijalankan.

```
D:\Bank Proposal\dipa\pemula\routing\Project2.exe
From? pekalongan
To? jogja
pekalongan to weleri to semarang to jogja
Distance is 200.
```

Gambar 7. Hasil Pencarian dengan algoritma Depth First

```
D:\Bank Proposal\dipa\pemula\routing\Project3.exe
From? pekalongan
To? jogja
pekalongan to semarang to jogja
Distance is 220.
```

Gambar 8. Hasil Pencarian dengan algoritma Breadth First

```
D:\Bank Proposal\dipa\pemula\routing\Project1.exe
From? pekalongan
To? jogja
pekalongan to semarang to jogja
Distance is 220.
```

Gambar 9. Hasil Pencarian dengan algoritma Hill Climbing

```
D:\Bank Proposal\dipa\pemula\routing\Project2.exe
From? pekalongan
To? jogja
pekalongan to salatiga to solo to yogyakarta to magelang to jogja
Distance is 240.
```

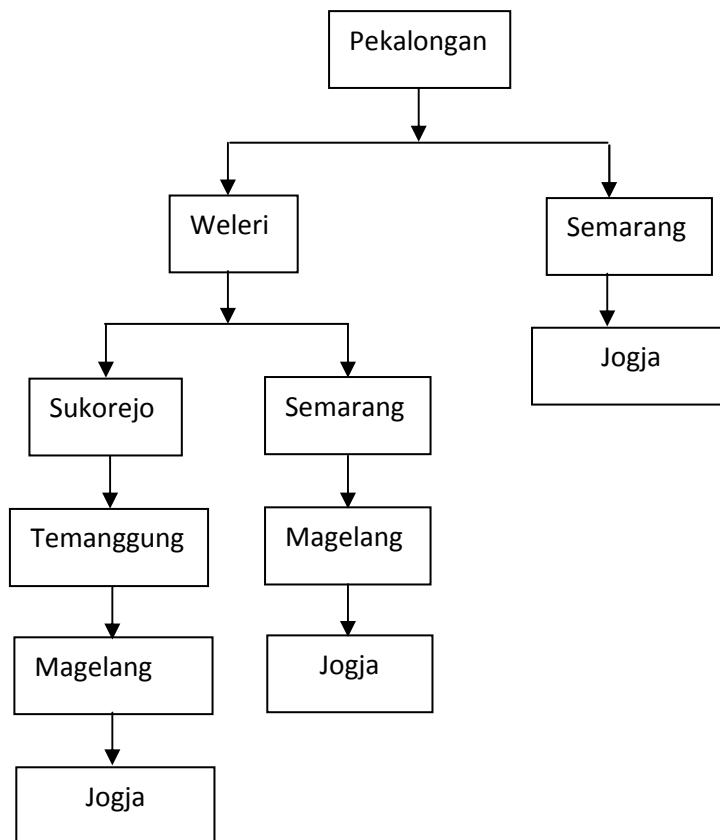
Gambar 10. Hasil Pencarian dengan algoritma Depth First dengan Struktur Data 2

2.2.2. Pembahasan

Algoritma pencarian *Depth First* mencoba untuk mengeksplor atau menjelajahi salah satu cabang terlebih dahulu, pada percobaan di atas cabang yang dijelajahi adalah cabang yang paling kiri sesuai juga dengan bentuk struktur datanya, pada Gambar 7. terlihat bahwa perjalanan dari kota Pekalongan Menuju ke Jogja akan melewati rute sebagai berikut : *Pekalongan – Weleri – Semarang – Jogja*

Pada hasil percobaan tersebut didapatkan rute dari pekalongan ke Jogja mengambil rute yang disusun pada cabang paling kanan. Pada percobaan ini belum mengindiasikan untuk menemukan rute terpendek ataupun yang paling efektif ketika akan bepergian dari kota Pekalongan ke kota Jogja melalui transportasi darat yaitu bus.

Pada percobaan selanjutnya susunan struktur data dibuat seperti pada gambar 12 yang diturunkan dari pohon keputusan sesuai Gambar 11. Hasil *running* program sesuai Gambar 10 didapatkan hasil bahwa algoritma pencarian *Depth First* digunakan untuk menelusuri kota – kota yang dilewati antara Kota Pekalongan ke Jogja, sehingga seorang *Sales* dapat memilih perjalanan ini untuk dapat sampai ke Jogja sembari memaksimalkan jumlah kota yang dapat disinggahi selama melakukan perjalanan (*kasus Traveling Salesman Problem*). Pada program ini masih mengalami kekurangan karena proses hanya melakukan pencarian pada dua titik yaitu titik asal dan tujuan. Pada program travling yang seharusnya program harus memuat titik – titik (node) yang harus dilewati kemudian ditelusuri perjalanan yang paling optimal dilakukan. Akan tetapi pad program ini hanya ditentukan sebagaimana orang bepergian dari satu kota ke kota lain sebagai tujuan, dan kota yang dilewati yang paling banyaklah yang menjadi acuan untuk dapat disinggahi.



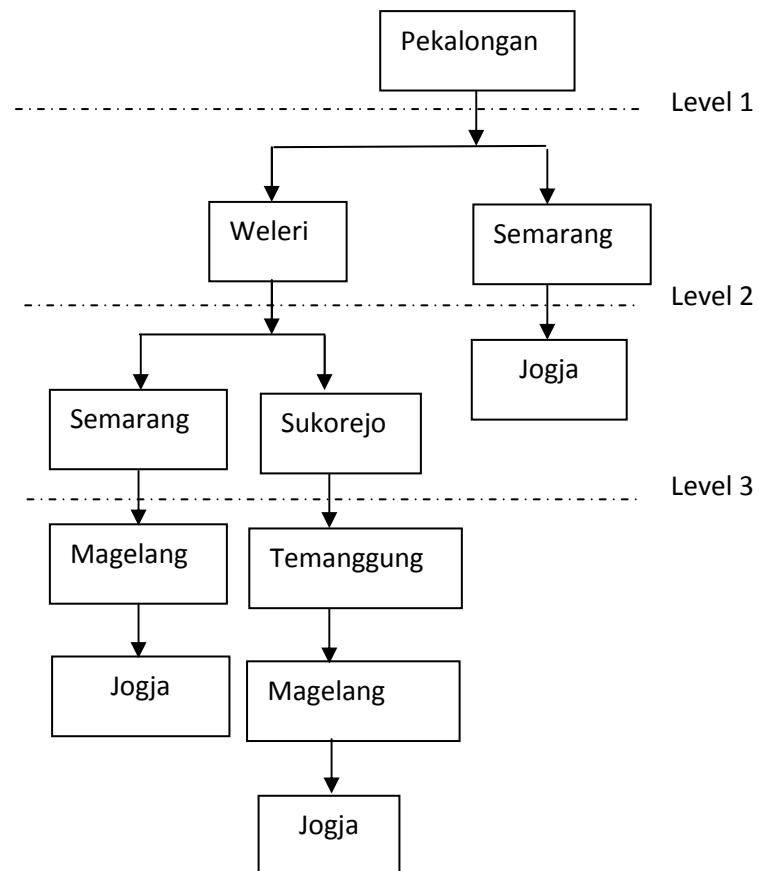
Gambar 11. Pohon Pencarian Untuk Struktur Data 2

Hasil yang didapat juga menggambarkan kelemahan dari proses pencarian algoritma *Depth First* yang akan selalu menjelajahi rute mulai dari kiri pohon, ketika susunan pohon keputusan dibuat sedemikian rupa sehingga setiap cabangnya mempunyai *goal* (tujuan) maka rute yang ditemukan adalah rute yang disusun pada cabang paling kiri, jika cabang paling kiri tidak ditemukan tujuan maka proses pencarian akan dilanjutkan pada cabang berikutnya.

Gambar 8 dan Gambar 9 menunjukkan hasil *running* program yang dibuat dengan algoritma *Breadth First* dan *Hill Climbing*. Pada gambar tersebut terlihat bahwa dua algoritma tersebut dapat digunakan untuk mendapatkan rute yang paling efektif ketika akan melakukan perjalanan darat dengan bus antara kota Pekalongan ke kota Jogja. Dua struktur data pada gambar 11 dan 12 yang diterapkan pada dua algoritma ini yaitu *Breadth First* dan *Hill Climbing* juga menghasilkan rute yang sama yaitu:

Pekalongan – Semarang – Jogja

Hasil rute diatas didapatkan dari proses pencarian rute yang paling efektif diantara 3 rute yang tersedia. Proses pendapatkan rute ini melalui proses penjelajahan tiap tingkat (level). Tiap level akan dijelahi apakah ada kota tujuan yang merupakan *goal* dari proses pencarian, jika ditemukan maka proses akan berhenti pada level tersebut, jika tujuan tidak ditemukan maka proses akan berlanjut pada level berikutnya. Gambar 12 berikut adalah proses penjelajahan tiap level yang dilakukan oleh 2 algoritma *Breadth First* dan *Hill Climbing* :



Gambar 12. Pohon Pencarian Dengan Pembagian Tiap Levelnya

Tiap level dari pohon keputusan ini diperiksa tiap *node* nya (kota), pada level 1 ditentukan untuk kota asal yaitu Pekalongan, setelah itu proses pencarian dilanjutkan ke level dibawahnya. Pada bagian tersebut ditemukan dua node yaitu Weleri dan Semarang, dari level 2 kemudian penelusuran dilanjutkan pada level 3, pada level ini ditemukan 3 node yaitu Semang, Sukorejo dan Jogja. Ketika di telusuri pada level ke 3 tujuan yaitu kota Jogja sudah ditemukan maka proses pencarian akan berakhir pada level ini.

3. Penutup

3.1. Kesimpulan

1. Algoritma terbaik untuk mendapatkan rute terpendek dan paling efektif yang diterapkan pada jalur transportasi yang tersedia adalah Algoritma Breadth First dan Algoritma Hill Climbing
2. Algoritma untuk mendapatkan titik – titik kota yang paling optimal untuk

disinggahi ketika melakukan perjalanan dari kota asal ke kota tujuan adalah Algoritma Depth First.

3.2. Saran

Adapun saran-saran yang dapat dikembangkan dari penggunaan algoritma pencarian ini adalah:

1. Penelitian lanjutan adalah penerapan algoritma pencarian ini untuk mendapatkan rute terpendek atau efektif ketika melakukan perjalanan dengan menggunakan kendaraan pribadi.
2. Pemanfaatan algoritma ini untuk menemukan jalur-jalur alternatif ketika melakukan perjalanan untuk mengunjungi beberapa kota sehingga perjalanan yang dilakukan dapat optimal baik dari segi waktu maupun biaya.

4. Daftar Pustaka

- Evan, 2008, Penggunaan *Depth First Search Dalam Pengambilan Keputusan Pada Klimaks Permainan Scrabble*, MAKALAH IF2251 STRATEGI ALGORITMIK TA-HUN 2008
- Lanny W, 2007 Pandjaitan, *Dasar-dasar komputasi cerdas*, Andi Offset, Yogyakarta.
- Riki Aditia, Fachrul Prasodjo, Irwansyah Ritonga, 2008, *Pencarian Jalur Dengan Breadth First Search Dan Depth First Search*, MAKALAH IF2251 STRATEGI ALGORITMIK TAHUN 2008
- Thiang, Handry Khoswanto, Felix Pasila, *Aplikasi Metode Hill Climbing Pada Standalone Robot Mobil Untuk Mencari Rute Terpendek*, Jurnal Informatif, Jurusan Teknik Elektro, Universitas Kristen Petra, Volume 3 2009.