Pengembangan Aplikasi Pencarian Rute Terbaik Pada Sistem Operasi Android (Studi Kasus Rute Trans-Jogja)

On Development the Search Best Application in Android Operating System (Case Study Trans-Jogja Route)

Aris Priyantoro, Khabib Mustofa
Jurusan Ilmu Komputer dan Elektronika, FMIPA, Universitas Gadjah Mada
E-mail: aris@pituruh.com

Abstrak

Layanan Trans-Jogja memiliki banyak rute perjalanan, dimana setiap rutenya akan melewati banyak tempat pemberhentian (*shelter*). Pengguna Trans-Jogja hanya dikenakan satu kali pembayaran apabila tidak keluar dari *shelter* meskipun harus berpindah-pindah jalur. Pengguna akan kesulitan menghitung waktu total perjalanan, total jarak ataupun jumlah pergantian bis dikarenakan tidak adanya informasi ataupun jadwal setiap armada yang lengkap pada setiap *shelter*.

Penelitian ini berhasil membangun aplikasi pencarian rute terbaik (berdasarkan jadwal keberangkatan, jarak terdekat dan jumlah pergantian armada) berbasis Google Map pada sistem operasi Android dengan studi kasus Trans-Jogja. Pengguna dalam aplikasi ini akan dapat menentukan *node* awal pencarian dan *node* akhir pencarian dengan beberapa cara, yaitu manual menunjuk *shelter*, otomatis menunjuk *shelter* awal berdasarkan posisi saat ini, dan menunjuk *shelter* terdekat dengan *Point Of Interest* (POI) tertentu. Berdasarkan hasil pengujian dapat disimpulkan bahwa waktu yang diperlukan untuk merespon suatu pencarian sangat bergantung dengan kualitas koneksi internet yang digunakan.

Kata kunci: depth-first search, rute terbaik, android, google map

Abstract

Trans-Jogja has a lot of routes, every route will pass through many stops (shelter). Trans-Jogja only charges one time payment if the customer doesn't go out of the shelter eventhough many routes will be passed. Customer will be difficult to calculate total trip time, total distance, or the number of replacement busses due to lack of information or a complete schedule of each fleet at each shelter.

This study succeeded in development of best route service (based on the scheduled departure time, shortest distance and the number of fleet turnover) based on Google Map in Android operating system with a case study of Trans-Jogja routes. Customer in this application will be able to choose initial shelter and destinationshelter in several ways, the manual refers shelter, automatically appoint the nearest shelter based on current position, and pointed to the nearest shelter using a Point Of Interest (POI). Based on the test results can be concluded that the respond time to a search is dependent on the quality of internet connection used.

Keywords: depth-first search, best route, android, google map

1. Pendahuluan

Kota Yogyakarta dikenal dengan berbagai macam jenis layanan transportasi, diantaranya adalah bis kota, taksi, becak, andong, dan yang paling baru adalah Trans-Jogja. Trans-Jogja memiliki keunikan tersendiri jika dibandingkan dengan jenis transportasi lainnya. Layanan ini memiliki banyak rute perjalanan, dimana setiap rutenya akan melewati

banyak tempat pemberhentian (*shelter*). Pengguna Trans-Jogja hanya dikenakan satu kali pembayaran apabila tidak keluar dari *shelter* meskipun harus berpindah-pindah jalur. Seseorang dimungkinkan untuk menghabiskan waktunya di satu bis Trans-Jogja dan tidak tahu harus turun di *shelter* mana dan berganti bis jalur berapa untuk sampai di tempat tujuan. Setiap *shelter* memiliki jadwal keberangkatan yang berbeda-beda untuk setiap armada yang masuk. Pengguna akan kesulitan menghitung waktu total perjalanan dikarenakan tidak adanya informasi jadwal setiap armada pada setiap *shelter*.

Pengguna membutuhkan informasi rute terbaik dari titik awal menuju titik tertentu. Titik awal dan titik tujuan dapat ditentukan sendiri oleh pengguna maupun berdasarkan posisi koordinat saat ini. Rute terbaik menurut preferensi pengguna dapat berbeda-beda. Rute terbaik berdasarkan waktu tercepat menurut sebagian pengguna merupakan rute terbaik, terdapat pula pengguna lain yang menganggap rute terdekat adalah rute terbaik, dan ada juga pengguna yang mengganggap pergantian armada bis yang paling sedikit adalah rute terbaik.

Penelitian tentang rute tercepat dan rute terdekat sudah banyak dilakukan sebelumnya. Muslim (2006) dalam tesisnya mengimplementasikan penentuan rute terbaik dengan basis sistem informasi geografis. Penentuan rute ini dibagi ke dalam tiga bagian yaitu rute terdekat, rute tercepat dan rute termurah. Proses pencarian jalur terbaik menggunakan fungsi-fungsi *built-in* dari aplikasi ArcView. Friedrich, (2001) dalam jurnal yang berjudul *Timetable-based Transit Assignment Using Branch & Bound* mengemukakan tentang pencarian jalur tercepat menggunakan jadwal pada transportasi umum. Algoritma yang digunakan dalam jurnal tersebut adalah *Branch and Bound*. Struktur data khusus digunakan untuk memberikan bantuan seperti pencarian segmen koneksi selanjutnya ketika diberikan sebuah *node* dan waktu atau mendapatkan semua koneksi yang berasal dari sebuah *node* yang tetap. Dalam jurnal tersebut diketahui bahwa pendekatan yang dilakukan dapat mengurangi waktu komputasi secara signifikan, selain itu juga dimungkinkan untuk menggunakan jadwal pada jaringan yang besar. Purnadi (2010) dalam skripsinya membuat aplikasi peta *mobile* untuk pencarian jalur terpendek pada sistem operasi Android. Algoritma yang dipakai sebagai dasar pencarian rute terpendek adalah algoritma Dijkstra. Dalam aplikasi yang dibuat dimungkinkan untuk menentukan arah *edge* yang dibuat. Jarak antar *node* diukur menggunakan skala dari gambar peta dikalikan dengan panjang *edge*.

2. Metode Penelitian

Penelitian ini akan membangun aplikasi pencarian rute terbaik (berdasarkan jadwal keberangkatan, jarak terdekat dan jumlah pergantian armada) berbasis Google Map pada sistem operasi Android dengan studi kasus Trans-Jogja. Pengguna dalam aplikasi ini akan dapat menentukan *node* awal pencarian dan *node* akhir pencarian dengan beberapa cara, yaitu manual menunjuk *shelter*, otomatis menunjuk *shelter* awal berdasarkan posisi saat ini, dan menunjuk *shelter* terdekat dengan *Point Of Interest* (POI) tertentu.

Algoritma yang digunakan dalam penelitian ini adalah algoritma *Depth-First Search* (DFS). Algoritma ini digunakan untuk mencari semua solusi jalur yang ditemukan dari *node* awal sampai dengan *node* akhir. Setelah didapatkan semua solusi yang mungkin, kemudian dicari rute terbaik diantara solusi tersebut..

2.1 Rute Terbaik

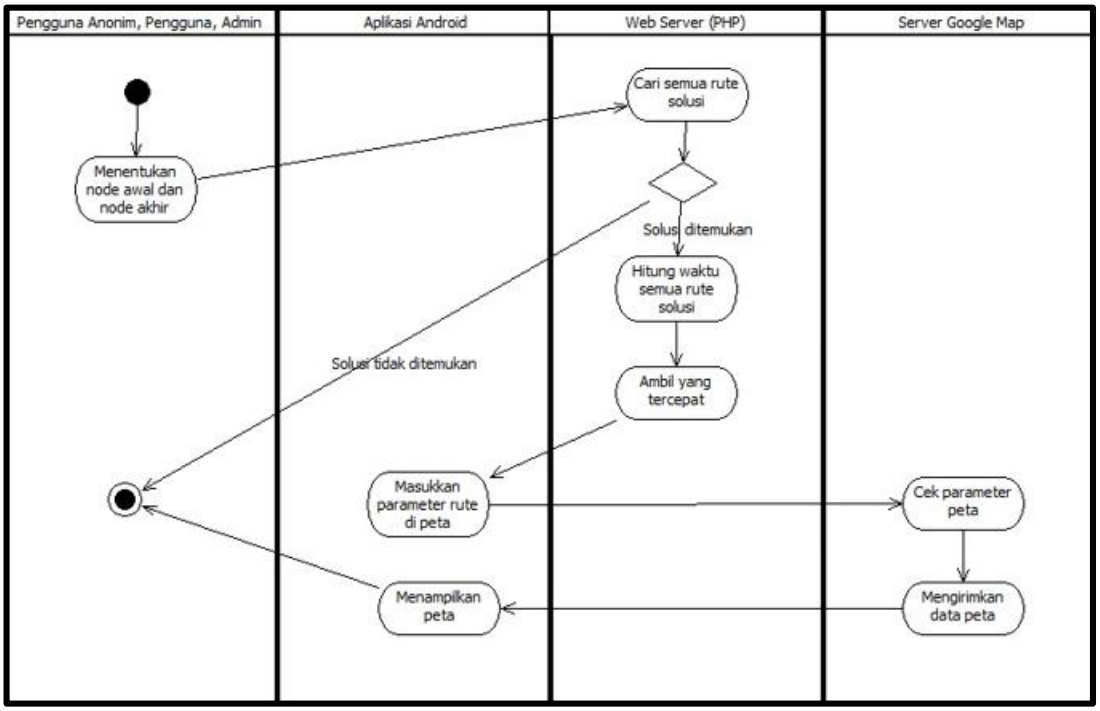
Rute terbaik pada penelitian ini adalah rute yang dipilih oleh pengguna sistem. Sistem menawarkan tiga jenis solusi terhadap pencarian rute, yaitu rute tercepat berdasarkan jadwal,

rute terdekat berdasarkan jarak dan rute dengan pergantian bis minimal. Hal ini akan memudahkan pengguna yang mempunyai preferensi berbeda terhadap suatu rute. Terdapat sebagian pengguna yang tidak menginginkan rute tercepat berdasarkan jadwal, tetapi menginginkan rute terdekat. Kasus lain, terdapat juga pengguna yang menginginkan rute yang dilewati mempunyai pergantian armada bis paling sedikit.

2.1.1 Pencarian Rute Tercepat Berdasarkan Jadwal Keberangkatan

Masalah pencarian rute tercepat berdasarkan jadwal keberangkatan armada angkutan umum (dalam hal ini Trans-Jogja) dapat diselesaikan dengan berbagai cara. Dalam penelitian ini akan menggunakan algoritma *Depth-First Search* yang dimodifikasi lebih jauh.

Ide utama untuk menyelesaikan masalah pada sub bab ini adalah menghasilkan semua kombinasi rute dari *node* awal menuju *node* akhir. Dari semua kombinasi rute tersebut kemudian dihitung waktu keberangkatan pada setiap *node* sehingga waktu keberangkatan armada pada *node* selanjutnya harus lebih besar dari jadwal keberangkatan *node* sekarang ditambah dengan waktu perjalanan ke *node* selanjutnya. Setelah semua rute dihitung waktu keberangkatan dari *node* awal sampai *node* tujuan maka data diurutkan dengan waktu tercepat sampai *node* tujuan berada paling atas. Dari urutan yang telah dibuat diambil yang paling atas sebagai rute dengan waktu tercepat dari *node* awal sampai *node* tujuan. Gambar 1 menunjukkan activity diagram untuk pencarian rute tercepat berdasarkan jadwal keberangkatan.

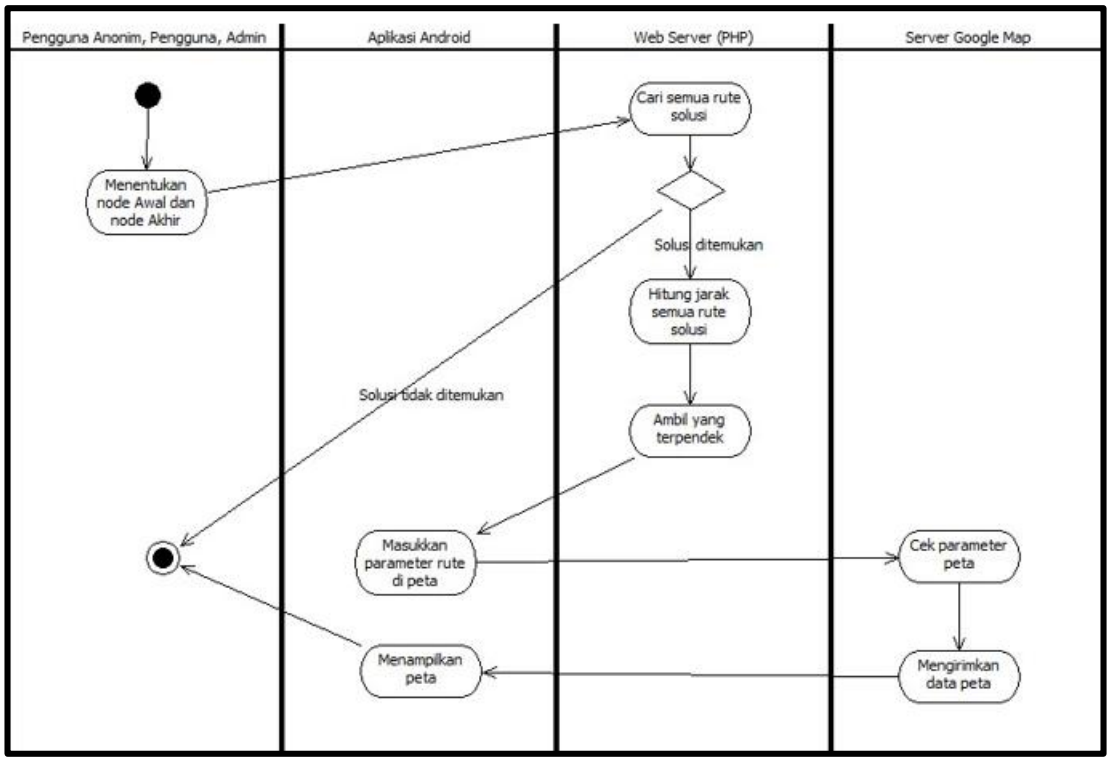


Gambar 1. Activity Diagram Pencarian Rute Tercepat Berdasarkan Jadwal.

2.1.2 Pencarian Rute Terdekat

Penelitian ini memiliki juga kemampuan untuk mencari rute terdekat. Algoritma yang digunakan untuk pencarian rute terdekat adalah *Depth-First Seach* (DFS). Algoritma yang digunakan sama dengan algoritma yang dipakai untuk pencarian rute tercepat. Alasan penggunaan algoritma yang sama adalah penggunaan sumber daya pada sistem yang efisien.

Algoritma DFS yang telah dimodifikasi menghasilkan daftar semua solusi *node* yang diperoleh. Dari hasil ini sistem tinggal mencari rute terdekat dari semua solusi. Sistem tidak perlu lagi untuk mencari solusi baru dan menghitung kembali rute yang terpendek. Gambar 2 adalah activity diagram pencarian rute terdekat.

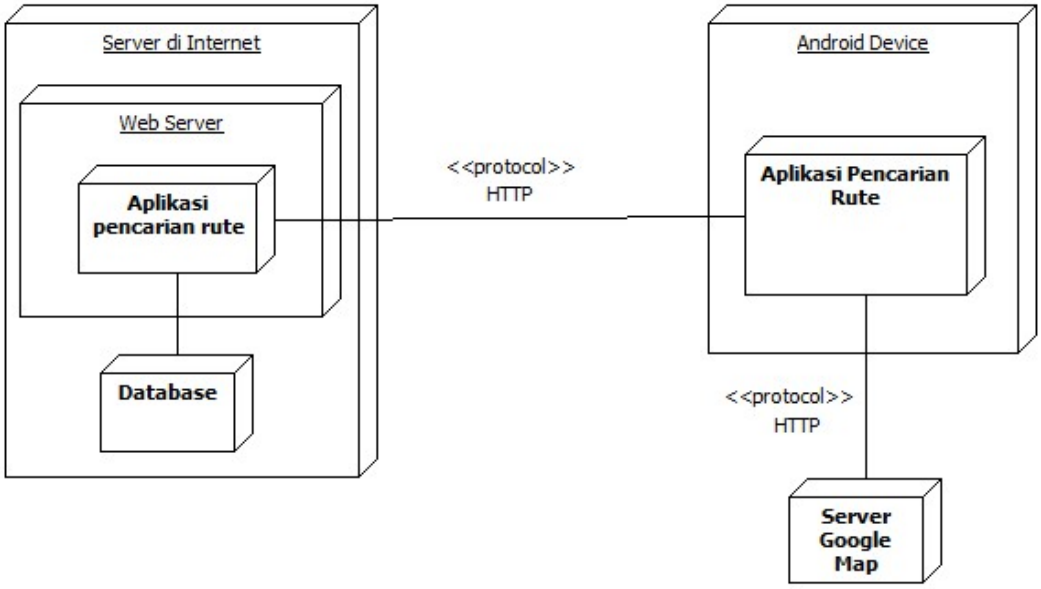


Gambar 2 Pencarian rute terdekat.

2.2 Deployment Diagram

Arsitektur yang digunakan dalam penelitian ini bersifat client-server. Aplikasi Android yang dihasilkan merupakan *client* yang memperoleh data dari *server*. Aplikasi yang berfungsi sebagai *server* adalah aplikasi web menggunakan bahasa pemrograman PHP. Terdapat tiga alasan pemilihan arsitektur client-server pada penelitian ini. Pertama, pemutahiran data mudah dilakukan, sehingga pengguna aplikasi Android tidak perlu melakukan update aplikasi ketika ada perubahan data. Kedua, basisdata yang ada pada sistem operasi Android (SQLite) belum mempunyai fasilitas yang dibutuhkan dalam membangun sistem ini, seperti stored procedures, view dan function. Alasan terakhir adalah fasilitas *Point Of Interest* (POI) membutuhkan basisdata penyimpanan yang terpusat.

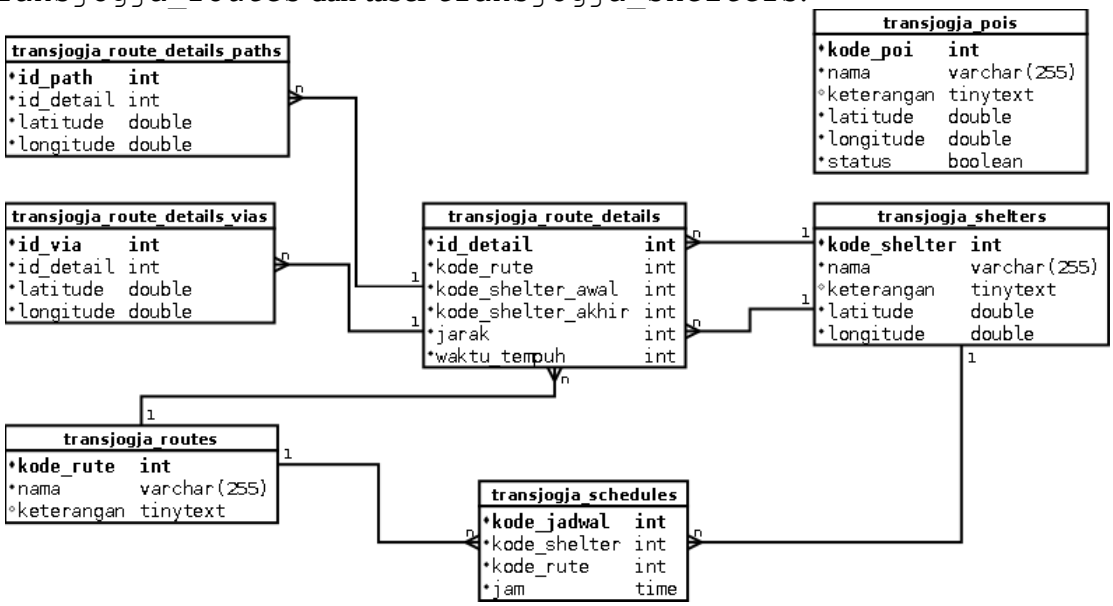
Sistem akan dipasang pada dua tempat yang berbeda (Gambar 3). Tempat yang pertama adalah di server yang dapat diakses oleh semua pengguna internet. Aplikasi ini hanya perlu dipasang sekali saja. Aplikasi ini akan membutuhkan database pada server yang sama. Tempat kedua adalah pemasangan aplikasi khusus untuk pencarian rute pada masing-masing handset Android pengguna. Protokol yang digunakan untuk berkomunikasi dari aplikasi di android dengan aplikasi di server adalah http. Protokol http juga digunakan untuk berkomunikasi antara android dengan server Google Map.



Gambar 3. *Deployment Diagram*

2.3 Desain Basis Data

Desain basis data yang akan digunakan dalam aplikasi ditunjukkan dalam Gambar 4. Dalam desain tersebut terdapat tujuh tabel. Tabel utama yang akan digunakan adalah tabel `transjogja_route_details`. Tabel ini mengambil data master dari tabel `transjogja_routes` dan tabel `transjogja_shelters`.



Gambar 4. Desain Basis Data

Tabel `transjogja_route_details_via` digunakan untuk membantu menggambar jalur yang menghubungkan antara dua *shelter*. Tabel ini akan menentukan belokan mana yang akan dilewati diantara dua *shelter*. Sedangkan tabel `POI` digunakan sebagai bantuan untuk penentuan lokasi *shelter* yang terdekat. Tabel `transjogja_route_details_paths` merupakan kumpulan data koordinat yang digunakan untuk menyusun *path* antar *shelter*. Data pada tabel `transjogja_shelter` dan tabel `transjogja_routes` dibuat jadwal keberangkatan pada tabel `transjogja_schedules`.

Desain basis data pada Gambar 5 digunakan untuk menyimpan data-data seperti pada Gambar 4. Kolom 1A, 1C, 2A dan 2C disimpan dalam tabel `transjogja_shelters`. Tabel `transjogja_routes` menyimpan data 3A dan 3C. Jadwal keberangkatan pada tiap-tiap kolom dalam *range* 5B sampai 11K untuk jalur 1A (kolom 5A) disimpan pada tabel `transjogja_schedules`.

	A	B	C	D	E	F	G	H	I	J	K
1	**HALTE**		**TERMINAL PRAMBANAN**								
2	**1001**		Pintu Keluar Terminal Prambanan								
3	1A		PRAMBANAN - BANDARA ADISUCIPTO - JANTI- AMBARUKMO- RS.BETHESDA- WANITATAMA - TUGU- MALIOBORO- TAMAN PINTAR- KUSUMANEGARA- GEMBIRALOKA- JEC - JANTI- MAGUWO- BANDARA ADI SUCIPTO - KALASAN - PRAMBANAN.								
4	RUTE WAKTU (WIB)										
5	1A	5:30	5:44	5:58	6:12	6:26	6:40	6:54	7:08	7:36	7:50
6		8:04	8:18	8:32	8:46	9:00	9:14	9:42	9:56	10:10	10:24
7		10:38	10:52	11:06	11:20	11:48	12:02	12:16	12:30	12:44	12:58
8		13:12	13:26	13:54	14:08	14:22	14:36	14:50	15:04	15:18	15:32
9		16:00	16:14	16:28	16:42	16:56	17:10	17:24	17:38	18:06	18:20
10		18:34	18:48	19:02	19:16	19:30	19:44	20:12	20:26	20:40	20:54
11		21:08	21:22								

Gambar 5. Contoh Data Mentah

2.4 Implementasi Algoritma DFS

Algoritma DFS dibuat menggunakan bahasa pemrograman PHP. Algoritma DFS membutuhkan graf. *Shelter-shelter* yang berhubungan satu-sama lain membentuk sebuah graf yang akan digunakan pada algoritman DFS. *Class*`Graph` digunakan untuk menyimpan informasi graf tersebut didalam memori. Kutipan kode program untuk menyusun graf ditunjukkan pada Gambar 6.

```
$_graph = new Graph();
$route_details = $this->route_details->get_all();
foreach($route_details as $detail) {
    $_graph->AddEdge($detail->kode_shelter_awal,
                     $detail->kode_shelter_akhir);
}
```

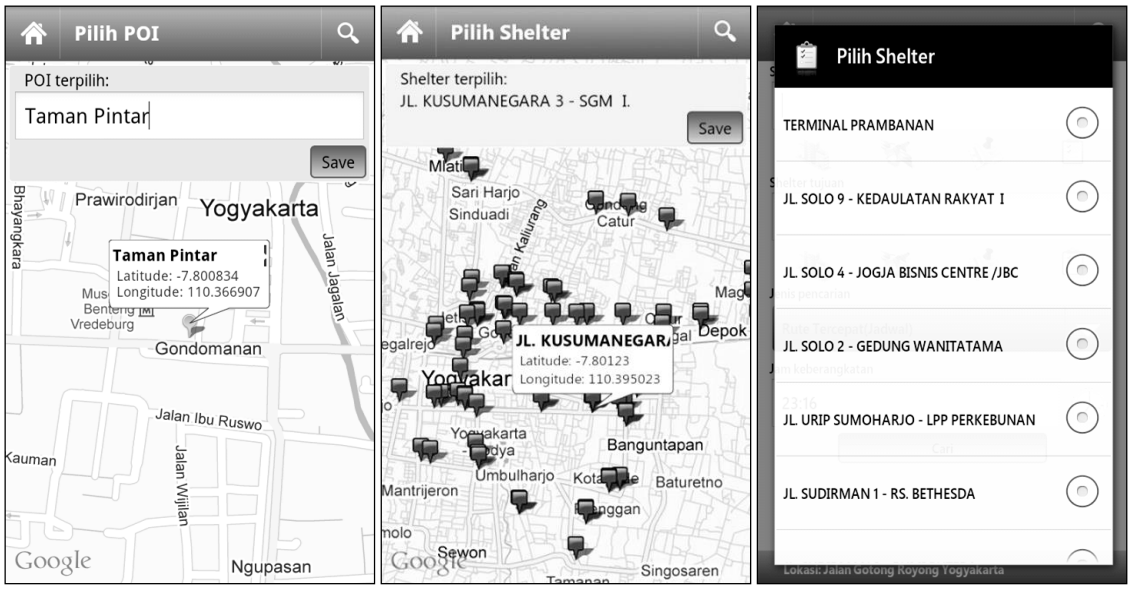
Gambar 6. Kode Penyusun Graf

Proses pencarian solusi pada algoritma DFS dimulai dengan mendefinisikan *shelter* awal dan *shelter* tujuan. Fungsi rekursi `DepthFirst` (Gambar 7) digunakan untuk mencari semua solusi yang mungkin dari *shelter* awal menuju *shelter* tujuan.

```
public function DepthFirst($graph, $visited) {
    $nodes = $graph->AdjacentNodes(end($visited));
    if (is_null($nodes)) return;
    foreach ($nodes as $node) {
        if (in_array($node, $visited))
            continue;
        if ($node == $this->_target) {
            $visited[] = $node;
            $this->_solution[] = $visited;
            array_pop($visited);
            break;
        }
    }
    foreach ($nodes as $node) {
        if (in_array($node, $visited) ||
            ($node == $this->_target))
            continue;
        $visited[] = $node;
        $this->DepthFirst($graph,$visited);
        array_pop($visited);
    }
}
```

Gambar 7. Fungsi rekusi `DepthFirst`

Gambar 8. Tampilan Pencarian Rute



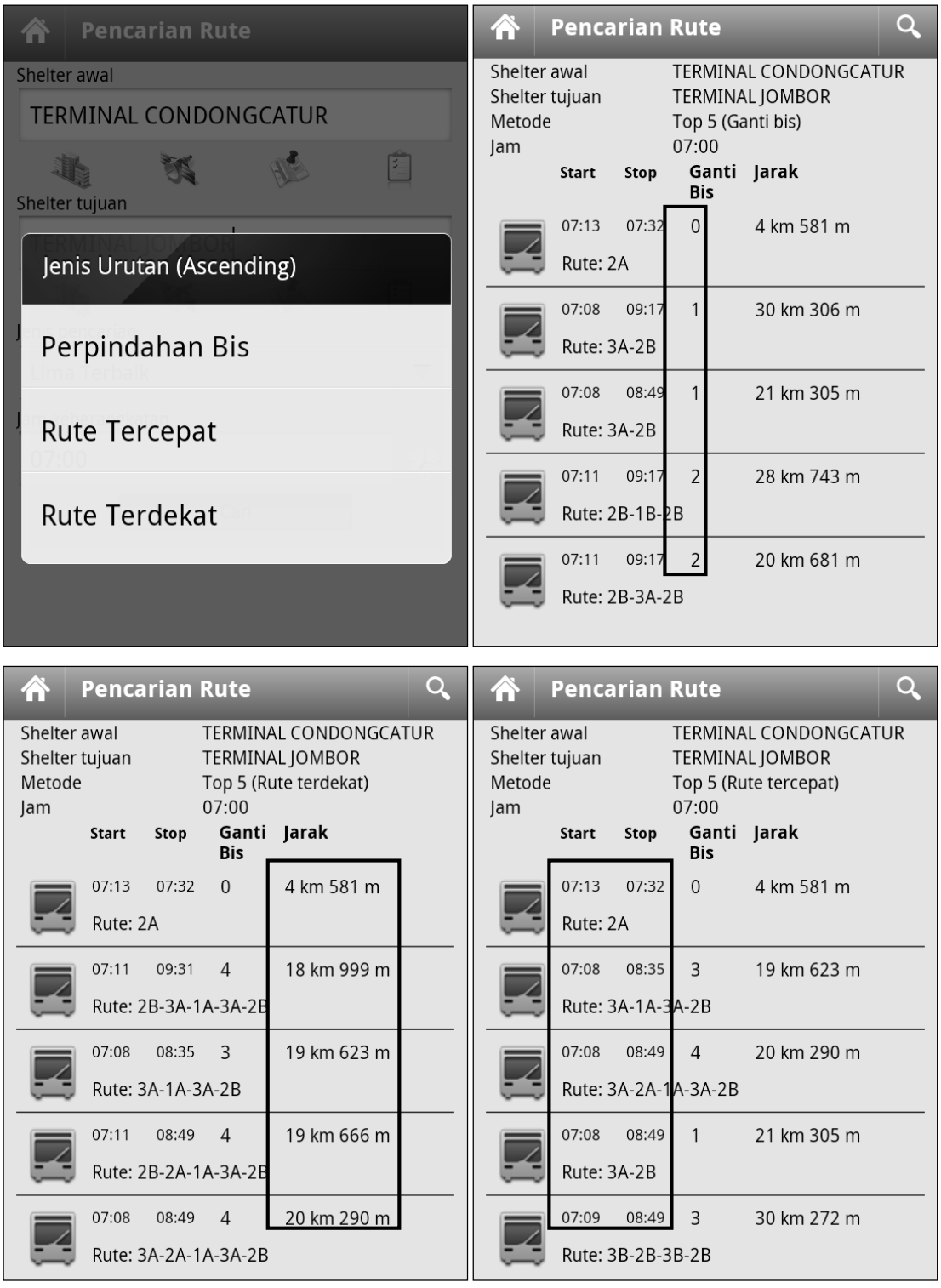
Gambar 9. Jenis-Jenis Penentuan Shelter

```
<?xml version="1.0" encoding="utf-8"?>
<TJRoute>
<Error></Error>
<StartShelter>5001</StartShelter>
<StopShelter>1126</StopShelter>
<Method>2</Method>
<TimeSearch>07:00</TimeSearch>
<Top5>
  <Top>
<start>07:11:00</start>
<end>07:56:00</end>
<distance>11 km 234 m</distance>
<buschanges>0</buschanges>
<solution>5001,2225,2224,2223,2121,2217,1126</solution>
<routes>2B</routes>
</Top>
  . . . . .
<Top>
<start>07:08:00</start>
<end>07:56:00</end>
<distance>11 km 858 m</distance>
<buschanges>1</buschanges>
<solution>5001,3129,2223,2121,2217,1126</solution>
<routes>3A-2B</routes>
</Top>
</Top5></TJRoute>
```

Gambar 10. XML Pencarian Lima Rute Terbaik

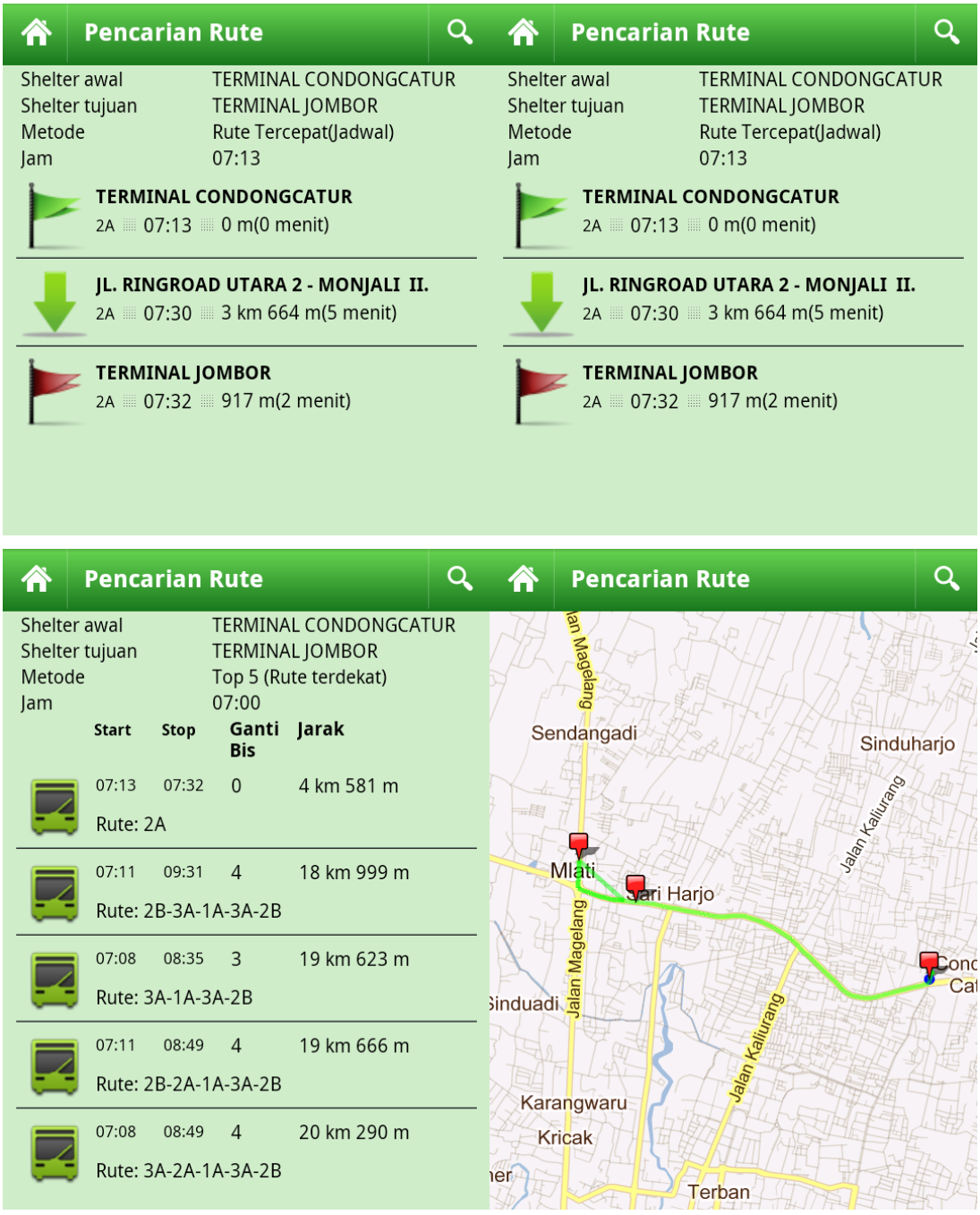
2.5 Antar Muka Pencarian Rute

Tampilan awal pencarian rute ditunjukkan pada Gambar 8. Aplikasi Android secara otomatis akan mendeteksi lokasi terdekat saat itu, apabila tidak memperoleh informasi nama jalan maka ditampilkan informasi *latitude* dan *longitude*. Informasi *shelter* awal dan *shelter* tujuan adalah hal utama yang harus ditentukan. Jenis pencarian dan waktu pencarian juga menentukan hasil yang akan diperoleh.



Gambar 11. Pencarian Lima Rute Terbaik

Cara menentukan *shelter* awal dan *shelter* tujuan ada lima, yaitu mengetikkan kata kunci pada *textbox*(akan ditampilkan daftar *shelter* berdasarkan kata kunci), menunjuk pada POI tertentu, menggunakan informasi koordinat dari GPS/BTS, menunjuk *shelter* tertentu pada peta dan memilih dari daftar *shelter*. Gambar 9 menunjukkan jenis-jenis penentuan *shelter*.



Gambar 12. Detail Hasil Pencarian Lima Terbaik

2.6 Hasil Pencarian Rute Terbaik

Solusi-solusi yang dihasilkan oleh algoritma DFS digunakan sebagai masukan pada pencarian rute terbaik. Pemilihan rute terbaik didasarkan pada jenis urutan yang telah dipilih. Urutan tersebut adalah Pergantian Bis, Rute Tercepat dan Rute Terdekat. Aplikasi Android akan meminta data lima rute terbaik. Proses pencarian lima rute terbaik dilakukan oleh skrip php di webserver. Semua solusi yang dihasilkan oleh algoritma DFS dicek satu-persatu dan dibandingkan dengan data awal. Setelah lima data terbaik diperoleh maka hasil tersebut dikembalikan ke aplikasi Android menggunakan format xml (gambar 10).

Tampilan layar untuk memilih jenis urutan pencarian lima rute terbaik ditampilkan pada Gambar 11. Hasil pencarian lima rute terbaik juga ditampilkan pada Gambar 11. Hasil

pencarian yang ditampilkan adalah informasi jam keberangkatan, jam sampai tujuan, jumlah pergantian bis, jarak dan rute yang dilewati. Hasil pencarian lima terbaik pada rute tercepat akan mengukur semua kemungkinan rute dengan waktu tercepat dimulai dari jam keberangkatan yang ditentukan sebelum pencarian (bukan jam keberangkatan pada setiap hasil pencarian).

Tampilan detil untuk hasil paling atas dari setiap jenis pengurutan ditunjukkan pada Gambar 12. Tampilan rute dalam Google Map untuk ketiga jenis pengurutan adalah sama untuk kasus yang dicontohkan dalam Gambar 11.

3. Hasil dan Pembahasan

Setelah semua rancangan selesai disusun dan aplikasi telah dibangun, maka tahap selanjutnya adalah pengujian aplikasi tersebut. Pengujian aplikasi meliputi perbandingan antara pencarian rute tercepat berdasarkan jadwal dengan pencarian rute terdekat dalam hal waktu komputasi. Waktu komputasi dihitung pada tipe jaringan dan waktu pencarian yang berbeda. Pengujian selanjutnya adalah mendapatkan waktu tempuh pada jam keberangkatan yang telah didefinisikan pada rute tertentu.

3.1 Data Pengujian

Data yang akan digunakan untuk pengujian adalah data asli jalur Trans-Jogja yang dikeluarkan pada tanggal 20 Mei 2009 oleh Dinas Perhubungan Daerah Istimewa Yogyakarta. Data *shelter* yang digunakan adalah sebanyak 74 *record* dimana data tersebut terhubung pada 6 jalur. Terdapat 134 koneksi antar *shelter* pada semua jalur dengan total 7805 record untuk jadwal keberangkatan pada setiap shelter. Selain data utama, dapat juga ditambahkan data pendukung berupa data-data *Point Of Interest* (POI).

3.2 Pengujian Untuk Waktu Tempuh

Pengujian pertama adalah pengujian waktu tempuh pada algoritma pencarian rute tercepat berdasarkan jadwal. Terdapat 3 pengujian pada tahap ini, yaitu pengujian pada shelter awal dan shelter tujuan dengan jarak yang pendek, menengah dan panjang.

Tabel 1 Pengujian Jalur Terminal Condong Catur – Bandar Udara Adisucipto

Waktu Pencarian	Jumlah shelter yang dilewati	Jumlah Perpindahan Bis	Waktu Tempuh
5:00	4	0	15
6:00	4	0	35
7:00	4	0	15
8:00	4	0	15
11:00	4	0	15
12:00	4	0	15
13:00	4	0	15
16:00	4	0	15
17:00	4	0	15
18:00	4	0	15

Pada semua pengujian akan digunakan waktu keberangkatan yang sama yaitu jam 05:00, 06:00, 07:00, 08:00, 11:00, 12:00, 13:00, 16:00, 17:00 dan 18:00. Penentuan waktu tersebut berdasarkan estimasi penggunaan armada Trans-Jogja terpadat dikarenakan jam-jam tersebut adalah waktunya berangkat sekolah/kerja, waktu makan siang/pulang sekolah dan waktu pulang kerja.

Shelter awal Terminal Condong Catur dan *shelter* tujuan Bandar Udara Adisucipto digunakan untuk menguji waktu tempuh jarak pendek. Hasil pengujian ditunjukkan pada Tabel 1. Terdapat 4 shelter yang dilewati dan tidak ada perpindahan bis untuk menuju Bandar Udara Adisucipto dari Terminal Condong Catur. Waktu tempuh untuk mencapai tujuan relatif sama yaitu 15 menit. Hanya terdapat satu data yang berbeda pada jam 6:00 dengan waktu tempuh 35 menit.

Pengujian pada jarak menengah menggunakan Terminal Jombor sebagai *shelter* awal dan Terminal Giwangan sebagai *shelter* tujuan. Hasil pengujian ditunjukkan pada Tabel 2. Data hasil pengujian memberikan gambaran bahwa aplikasi memilih untuk menggunakan urut-urutan rute yang sama hampir pada semua keberangkatan (Tabel 2). Hanya terdapat satu keberangkatan yaitu jam 13:00 dimana aplikasi menggunakan urut-urutan rute yang berbeda. Jumlah *shelter* yang dilewati sama, yaitu 12 *shelter*. Jumlah perpindahan bis bervariasi tetapi hanya berkisar pada satu kali atau dua kali berganti bis.

Waktu tempuh tercepat yaitu 44 menit terjadi pada keberangkatan jam 5:00, disusul jam 7:00 dengan waktu 51 menit, kemudian jam 11:00, 16:00 dan 18:00 mempunyai waktu tempuh yang sama yaitu 53 menit. Jam 6:00 dan 8:00 memperoleh waktu tempuh 55 menit. Waktu tempuh terlama terjadi pada jam 12:00, 13:00 dan 17:00 yaitu 57 menit.

Tabel 2. Pengujian Jalur Terminal Jombor – Terminal Giwangan

Waktu Pencarian	Jumlah shelter yang dilewati	Jumlah Perpindahan Bis	Waktu Tempuh
5:00	12	1	44
6:00	12	2	55
7:00	12	1	51
8:00	12	1	55
11:00	12	1	53
12:00	12	1	57
13:00	12*	1	57
16:00	12	2	53
17:00	12	2	57
18:00	12	1	53

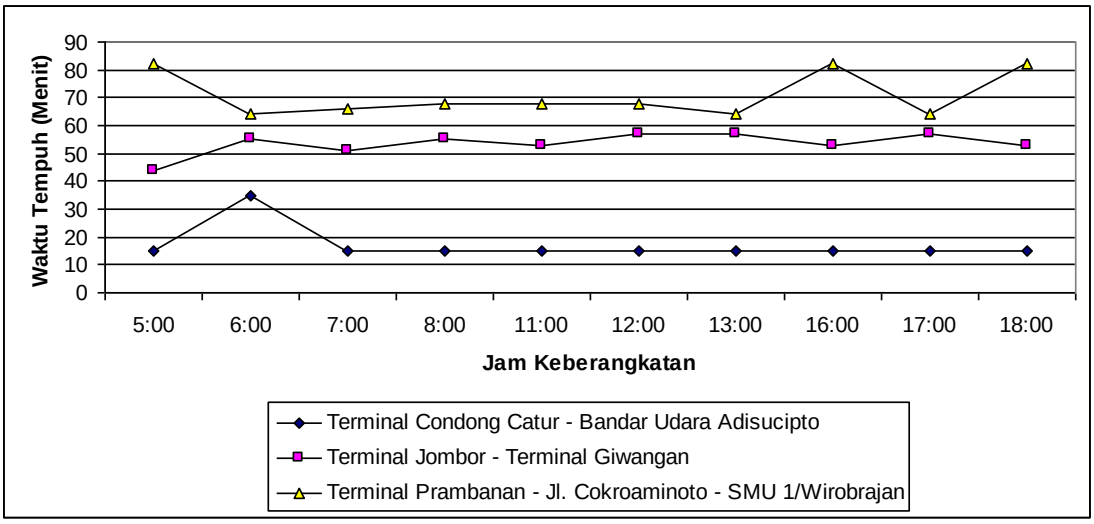
Pengujian untuk jarak tempuh terjauh menggunakan Terminal Prambanan sebagai *shelter* awal dan Jalan Cokroaminoto – SMU 1/Wirobrajan sebagai *shelter* tujuan. Hasil pengujian ditunjukkan pada Tabel 3. Data hasil pengujian pada Tabel 3 memperlihatkan bahwa terdapat tiga variasi urut-urutan rute yang menghasilkan waktu tercepat. Urut-urutan pertama melewati 15 *shelter* dengan jumlah perpindahan sebanyak 2 kali, urut-urutan kedua melewati 11 *shelter* dengan jumlah perpindahan 4 kali dan urut-urutan terakhir melewati 11 shelter dengan jumlah perpindahan bis sebanyak 3 kali.

Waktu tempuh tercepat terjadi pada jam keberangkatan 6:00, 13:00 dan 17:00 dengan waktu 64 menit. Waktu tempuh selanjutnya adalah 66 menit pada jam 7:00. Waktu tempuh selama 68 menit terjadi pada jam 8:00, 11:00, dan 12:00. Jam keberangkatan 5:00, 16:00, dan 18:00 menghasilkan waktu tempuh terlama yaitu 82 menit.

Grafik dari ketiga pengujian tersebut ditampilkan dalam Gambar 13. Dari grafik tersebut terlihat bahwa pada jam-jam tertentu waktu tempuh mengalami peningkatan yang signifikan pada pengujian dengan jarak tempuh jauh. Sedangkan waktu tempuh tidak mengalami perbedaan secara signifikan untuk jarah tempuh menengah maupun jarak pendek. Hal ini dimungkinkan karena tidak banyaknya alternatif jalur yang dilewati pada jarak pendek dan menengah dibanding pada jarak jauh.

Tabel 3. Pengujian Jalur Terminal Prambanan - SMU 1/Wirobrajan

Waktu Pencarian	Jumlah shelter yang dilewati	Jumlah Perpindahan Bis	Waktu Tempuh
5:00	15	2	82
6:00	11*	3	64
7:00	15	2	66
8:00	11**	4	68
11:00	15	2	68
12:00	15	2	68
13:00	11*	3	64
16:00	15	2	82
17:00	11*	3	64
18:00	15	2	82



Gambar 13. Pengujian Waktu Tempuh Pencarian Rute Tercepat

3.3 Waktu Komputasi Pencarian Rute Terbaik

Pengujian pada aplikasi berdasarkan waktu komputasi dilakukan pada tiga *shelter* awal dan *shelter* tujuan yang sama dengan pengujian pada sub-bab 3.2. Untuk setiap pengujian, terdapat empat media internet yang digunakan, yaitu Wifi, Edge, Evdo dan Hsdpa. Untuk tipe Wifi menggunakan koneksi yang sangat stabil dan sangat berdekatan dengan *server* aplikasi. Tipe Edge, menggunakan koneksi internet dengan provider IM3. Evdo menggunakan koneksi internet dari smartfren. Terakhir HSDPA menggunakan koneksi internet dari 3.

Penggunaan tipe koneksi internet yang berbeda-beda bertujuan untuk menguji kecepatan komputasi dari aplikasi ke *server* sampai dengan mendapatkan respon kembalian. Tentu saja kecepatan komputasi pada pengujian ini tidak akan sama setiap saat karena kecepatan koneksi internet akan berbeda-beda pada setiap *provider* pada waktu tertentu.

Pengujian pertama menggunakan Terminal Condong Catur sebagai *shelter* awal dan Bandar Udara Adisucipto sebagai *shelter* tujuan. Tabel 4 menampilkan hasil pengujian waktu komputasi untuk algoritma pencarian rute tercepat.

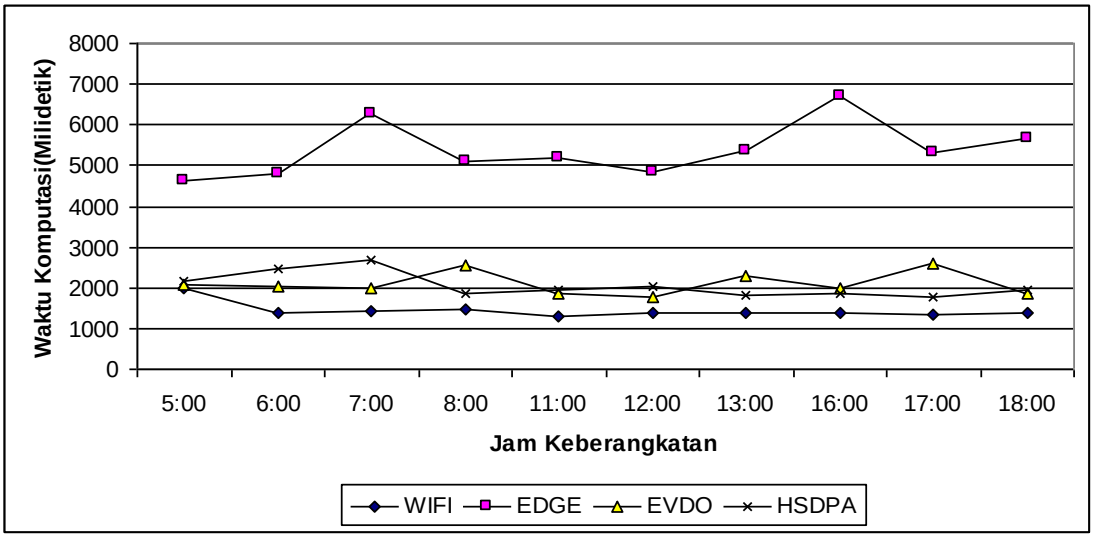
Waktu komputasi menggunakan Wifi secara rata-rata paling cepat dibandingkan dengan menggunakan koneksi yang lain. Waktu tercepatnya adalah 1288 milidetik pada jam 11:00, sedangkan waktu terlama adalah 2008 milidetik pada jam 5:00.

Kecepatan komputasi pada jaringan Edge yang tercepat adalah 4643 milidetik pada jam 5:00, kecepatan ini naik 260.48% dari waktu komputasi tercepat pada jaringan Wifi. Waktu komputasi terlama pada jaringan Edge adalah 6687 milidetik, atau naik sebesar 233.01% dari waktu komputasi terlama pada jaringan Wifi.

Tabel 4. Waktu Komputasi Pengujian Pertama Algoritma Pencarian Rute Tercepat

Jam	WIFI	EDGE	EVDO	HSDPA
5:00	2008	4643	2063	2150
6:00	1402	4792	2037	2458
7:00	1440	6263	1980	2702
8:00	1457	5109	2568	1867
11:00	1288	5188	1851	1944
12:00	1383	4856	1794	2030
13:00	1405	5368	2311	1833
16:00	1390	6687	1979	1854
17:00	1336	5314	2606	1793
18:00	1377	5677	1865	1926

Pengujian menggunakan jaringan Evdo mempunyai waktu komputasi tercepat sebesar 1794 milidetik, naik sebesar 41.15% dari waktu komputasi tercepat para jaringan Wifi. Waktu komputasi terlama sebesar 2568 milidetik, naik sebesar 27.89% dari waktu komputasi terlama pada jaringan Wifi.



Gambar 14. Waktu Komputasi Pengujian Pertama

Gambar 14 menunjukkan bahwa waktu komputasi pada semua jam keberangkatan hampir sama pada jaringan Wifi. Jaringan Evdo maupun Hsdpa mempunyai kecenderungan yang sama yaitu sedikit terdapat fluktuasi waktu komputasi tetapi tidak terlalu jauh simpangannya. Sedangkan jaringan Edge sangat tidak stabil dan waktu komputasi yang lambat apabila dibandingkan dengan ketiga jarigan lainnya.

Pengujian kedua untuk *shelter* awal Terminal Jombor dan *shelter* tujuan Terminal Giwangan. Tabel 5 menampilkan hasil pengujian waktu komputasi untuk algoritma pencarian rute tercepat. Waktu komputasi menggunakan koneksi Wifi terlihat paling cepat,

waktu terlama adalah 2283 milidetik atau setara dengan 2.3 detik terjadi pada jam 5:00. Waktu komputasi tercepat terjadi untuk pencarian jam 16:00 sebesar 1661 milidetik.

Kecepatan melakukan komputasi pada jaringan internet Edge sangat jauh berbeda dari Wifi. Waktu komputasi tercepat adalah 7586 milidetik atau naik sebesar 356.7% dari kecepatan menggunakan Wifi. Sedangkan waktu terlamanya adalah 17065 milidetik atau naik sebesar 647.5% dari waktu terlama pada jaringan Wifi. Hal ini terjadi karena faktor kedekatan server pada jaringan Wifi dan Edge berbeda. Jaringan Wifi yang penulis gunakan posisinya sangat dekat (terdapat dalam lokasi yang sama) sedangkat posisi *server* apabila diakses menggunakan jaringan Edge akan jauh. Koneksi internet menggunakan Edge terlihat tidak cukup stabil apabila dilihat dari waktu komputasi yang jarak nilai antar jam pengujian yang cukup tinggi.

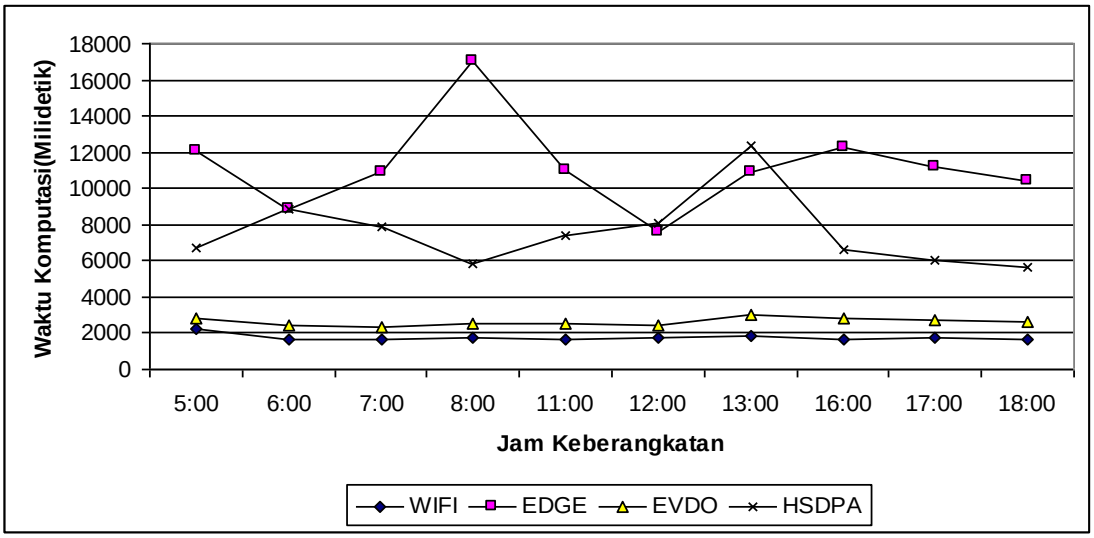
Tabel 5. Waktu Komputasi Pengujian Kedua Algoritma Pencarian Rute Tercepat

Jam	WIFI	EDGE	EVDO	HSDPA
5:00	2283	12087	2794	6702
6:00	1701	8893	2464	8900
7:00	1662	10863	2325	7895
8:00	1769	17065	2492	5884
11:00	1689	10990	2514	7443
12:00	1716	7586	2471	8100
13:00	1813	10916	3033	12356
16:00	1661	12241	2800	6577
17:00	1737	11202	2720	6030
18:00	1670	10366	2598	5613

Koneksi internet Evdo pada pengujian ini terlihat jauh lebih baik dari koneksi internet Edge. Waktu komputasi tercepat yang diperoleh adalah 2325 milidetik atau naik 39.9% dari koneksi Wifi. Waktu terlama yang diperlukan untuk mendapatkan hasil perhitungan pada koneksi Evdo adalah 3033 milidetik atau naik sebesar 32.8% dari waktu terlama pada koneksi Wifi. Koneksi internet Evdo terlihat cukup stabil dengan selisih waktu komputasi yang kurang dari satu detik untuk tiap-tiap waktu pengujian.

Waktu komputasi tercepat pada koneksi internet Hsdpa adalah 5613 milidetik atau naik sebesar 237.9% dari waktu tercepat pada koneksi Wifi. Sedangkan waktu terlama untuk melakukan komputasi adalah 12356 milidetik atau naik sebesar 441.2% dari waktu terlama pada jaringan Wifi. Waktu komputasi pada jaringan Hsdpa terlihat tidak cukup stabil dikarenakan perbedaan yang lebih dari 2 detik untuk tiap-tiap waktu pengujian.

Gambar 15 menunjukkan perbandingan waktu komputasi pada pengujian kedua pada empat jenis jaringan yang berbeda. Waktu komputasi untuk jaringan Wifi dan Evdo mempunyai kesamaan pola yaitu membentuk garis yang mendatar. Selisih waktu komputasi antara jaringan Wifi dan Evdo tidak lebih dari 1000 milidetik, atau satu detik. Waktu komputasi menggunakan jaringan Hsdpa jauh diatas Evdo, yaitu berkisar antara 5631 milidetik sampai 12356 milidetik. Grafik yang dibentuk cukup fluktuatif. Waktu komputasi menggunakan Edge terlihat paling tidak stabil dan paling lama.



Gambar 15. Waktu Komputasi Pengujian Kedua

Pengujian ketiga menghitung waktu komputasi pada *shelter* awal Terminal Prambanan dan *shelter* tujuan adalah Jl Cokroaminoto SMU 1/Wirobrajan. Tabel 6 menampilkan hasil pengujian tersebut.

Waktu komputasi menggunakan Wifi jelas sekali terlihat paling cepat dibandingkan dengan lainnya. Waktu tercepat diperoleh pada jam 17:00 yaitu sebesar 2150 milidetik. Sedangkan waktu terlama diperoleh pada jam 5:00 sebesar 2356 milidetik.

Pengujian pada jaringan Edge menghasilkan waktu komputasi tercepat sebesar 9521 milidetik pada jam 6:00, atau naik 342.83% dari waktu komputasi pada jaringan Wifi. Sedangkan waktu komputasi terlama terjadi pada jam 17:00 sebesar 22015 milidetik, atau 834.42%.

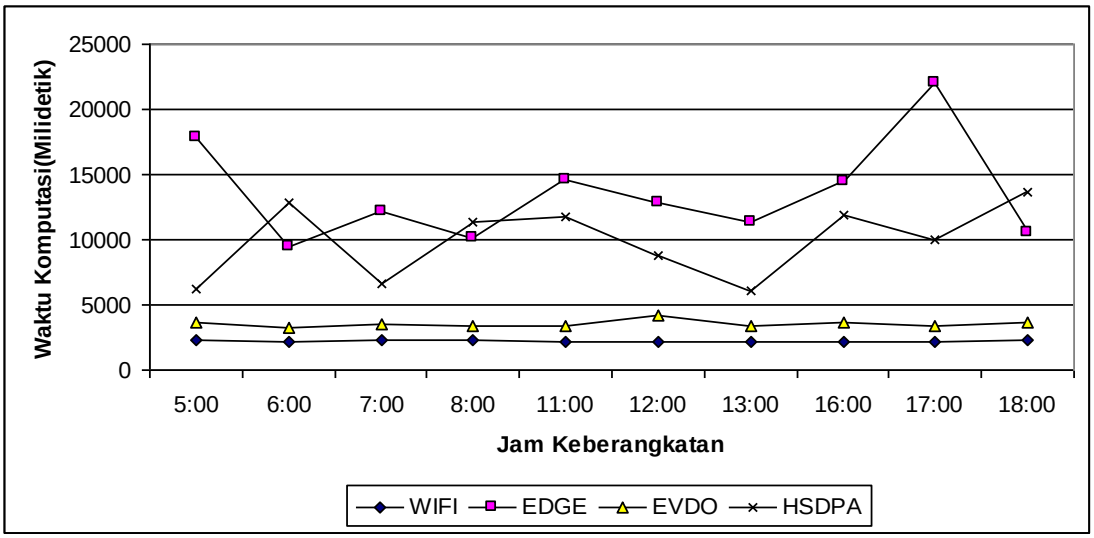
Pengujian pada jaringan Evdo menghasilkan waktu komputasi yang lebih baik, bahkan tidak berbeda jauh dengan pengujian pada jaringan Wifi. Waktu komputasi tercepat terjadi pada jam 6:00 sebesar 3224 milidetik, naik sebesar 49.95% dari jaringan Wifi. Sedangkan waktu komputasi terlama pada pengujian ini adalah 4185 milidetik, naik sebesar 77.63%.

Tabel 6. Waktu Komputasi Pengujian Ketiga Algoritma Pencarian Rute Tercepat

Jam	WIFI	EDGE	EVDO	HSDPA
5:00	2356	17869	3682	6178
6:00	2153	9521	3224	12901
7:00	2343	12130	3492	6685
8:00	2250	10149	3381	11406
11:00	2203	14528	3357	11735
12:00	2163	12856	4185	8843
13:00	2223	11313	3341	6070
16:00	2164	14522	3634	11855
17:00	2150	22015	3333	10053
18:00	2269	10552	3710	13702

Waktu komputasi tercepat pada jaringan Hsdpa adalah 6070 milidetik pada jam 13:00. Waktu komputasi ini naik sebesar 182.32% dari waktu komputasi tercepat pada jaringan Wifi. Jam 6:00 menghasilkan waktu komputasi terlama yaitu sebesar 12901 milidetik, atau naik sebesar 447.58% dari waktu komputasi terlama pada jaringan Wifi.

Grafik untuk menunjukkan perbandingan antara empat jenis jaringan internet untuk pengujian waktu komputasi ditampilkan pada Gambar 16.



Gambar 16. Waktu Komputasi Pengujian Ketiga

Perbandingan waktu komputasi pada pengujian ketiga menghasilkan pola yang tidak jauh berbeda dengan waktu komputasi pada pengujian kedua. Pengujian menggunakan koneksi Wifi mempunyai waktu komputasi yang paling cepat, disusul waktu komputasi menggunakan koneksi Evdo. Sedangkan waktu komputasi menggunakan koneksi Hsdpa dan Edge terlihat tidak stabil tetapi Hsdpa lebih cepat secara rata-rata daripada menggunakan koneksi Edge.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka dapat disimpulkan hal-hal sebagai berikut:

1. Aplikasi pencarian rute terbaik berhasil dikembangkan di sistem operasi Android.
2. Aplikasi pencarian rute terbaik menyediakan tiga jenis pengurutan, yaitu pengurutan berdasarkan rute tercepat sesuai jadwal, rute terdekat dan pergantian bis.
3. Waktu yang diperlukan untuk merespon suatu pencarian sangat bergantung dengan kualitas koneksi internet yang digunakan.

Daftar Pustaka

Muslim, A., 2006: Sistem Penentuan Rute Terbaik Berbasis Sistem Informasi Geografis, *Tesis*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta

Friedrich, M., Hofsaess, I., dan Wekeck, S., 2001, *Timetable-based Transit Assignment Using Branch & Bound*, *Transportation Research Record*, National Academy Press, Washington D.C.

Purnaadi, C., W., 2010, Aplikasi Peta Mobile Untuk Pencarian Jalur Terpendek Pada Sistem Operasi Android, *Skripsi*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada, Yogyakarta