

IMPLEMENTASI ALGORITMA *GREEDY* PADA LAYANAN TAKSI WISATA BERBASIS WEB

Adi Cahyo Purnomo¹, Mike Yuliana, ST. MT.¹, Ira Prasetyaningrum, S.Si. MT.¹
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya 60111
email : adi_cahyo_purnomo@yahoo.com

ABSTRAK

Perkembangan teknologi telekomunikasi dan informatika saat ini semakin mempermudah setiap penggunanya dalam mengakses layanan informasi. Layanan informasi ini memberikan kemudahan bagi pengguna dalam memperoleh informasi dengan cepat, serta dapat diakses dari mana saja dan kapan saja. Termasuk layanan hiburan untuk masyarakat seperti perencanaan wisata.

Pada tugas akhir ini akan dibuat sistem layanan taksi wisata dengan implementasi Algoritma *Greedy*. Layanan yang akan dibuat pada tugas akhir ini akan menawarkan informasi tempat wisata yang dapat dikunjungi dan menginformasikan biaya pada customer dimana biaya ini mencakup biaya tiket masuk, biaya parkir, termasuk biaya makan serta tarif untuk taksi berdasarkan seberapa jauh jarak yang ditempuh. Algoritma *Greedy* pada layanan taksi wisata ini dapat membantu untuk menginformasikan mengenai urutan tempat wisata yang akan dikunjungi selama melakukan wisata.

Hasil dari tugas akhir ini adalah sebuah layanan taksi wisata berbasis web, dimana pelanggan dapat melakukan pemesanan melalui *website* tempat wisata mana yang akan dituju. Dengan mengimplementasikan algoritma *greedy* di dalamnya untuk menghitung optimasi kunjungan tempat wisata, hal tersebut dapat memberikan pengguna informasi yang optimal mengenai jarak tempuh serta biaya yang dikeluarkan. Algoritma *greedy* ini lebih bagus apabila digunakan pada kurang dari 10 tempat tujuan. Karena hasil dari algoritma *greedy* dan dibanding dengan algoritma lain, algoritma *greedy* memiliki hasil optimasi yang lebih bagus dan tentu saja dengan waktu komputasi yang lebih cepat.

1. Pendahuluan

Waktu dan biaya akan menjadi isu paling penting yang sangat diperhitungkan apabila seseorang akan melakukan perjalanan, baik itu wisata atau perjalanan dinas. Karena tidak ada seorang pun yang mengenal semua wilayah yang akan dia datangi. Saat ini sistem informasi mengenai suatu tempat wisata sudah banyak bermunculan. Tetapi hampir semuanya belum ada

yang bisa memberikan informasi lengkap apabila hendak melakukan wisata ke beberapa tempat wisata dalam satu waktu, serta adanya perhitungan secara detail tentang akomodasi dalam berwisata.

Dalam menentukan biaya serta informasi tempat wisata secara detail, digunakan algoritma *Greedy* untuk optimasi kunjungan tempat wisata. Algoritma ini bekerja dengan mencari jalur terpendek antara setiap node dan menempatkannya pada solusi yang disediakan atau bisa dituliskan dengan notasi *big-O*, $O(n^2 \log_2(n))$. Dalam kasus untuk proyek akhir ini, akan diimplementasikan TSP (*Travelling Salesman Problem*) didalam algoritma *Greedy*. Dimana untuk kasus TSP memiliki persyaratan bahwa semua tujuan wisata akan dikunjungi semuanya, dan setiap tempat wisata hanya sekali dikunjungi.

Pada pembuatan tugas akhir ini, informasi yang diperlukan sistem sebagai input adalah tempat wisata. Setiap pengguna dapat memasukkan tempat wisata yang akan dia kunjungi. Kemudian sistem akan menerima inputan tersebut dan menghitung setiap waktu dan biaya yang dikeluarkan oleh pengguna selama berwisata. Dan keluaran yang diterima oleh pengguna dapat berupa *time management* selama berwisata. Di dalam *time management* disebutkan tempat wisata, waktu yang diberikan dalam setiap tempat wisata serta estimasi perjalanan ke setiap tempat wisata. Sehingga diharapkan, setiap pengguna dapat mengetahui informasi secara detail pada setiap kunjungan wisatanya.

2. Penelitian Sebelumnya

Pada *paper* yang dibuat oleh Wiradeva Arif Kristawarman [9]. Dilakukan perbandingan kompleksitas penerapan algoritma *Greedy* pada beberapa masalah, diantaranya adalah TSP (*Traveling Salesman Problem*), *Minimum Spanning Tree*, dan Minimasi waktu pada sistem (Penjadwalan). Untuk permasalahan kasus TSP, seseorang harus mengunjungi sebanyak n kota dengan aturan harus mengunjungi setiap kota hanya sebanyak satu kali, harus meminimalisasi total jarak perjalanan, serta pada akhirnya akan kembali ke kota asalnya. Dengan begitu, pa yang telah dilakukan adalah sebuah tour. Dilakukan pemetaan

untuk n kota tersebut untuk memudahkan penyelesaian, dimana jumlah *vertice* dan *edge* terbatas (sebuah *vertice* mewakili sebuah kota dan *edge* mewakili jarak antara dua kota). Penanganan problem TSP ini sama dengan pencarian sirkuit *Hamiltonian* terpendek. Metode penelitian beliau menggunakan pemrograman bahasa C untuk menyelesaikan permasalahan tersebut.

3. Dasar Teori Sistem

3.1 Algoritma Greedy

Konsep permasalahan TSP (*Traveling Salesman Problem*) memiliki aturan (i) harus mengunjungi setiap kota sebanyak satu kali, (ii) semua kota harus dikunjungi dalam satu kali *tour*, (iii) dan harus kembali ke kota asalnya. Dengan demikian, apa yang telah dilakukan adalah membentuk sebuah *tour*. Penanganan problem TSP ini ekuivalen dengan mencari sirkuit *Hamiltonian* terpendek dari serangkaian solusi yang ada. Mengacu pada konsep *greedy* yang menganggap bahwa pada setiap langkah akan dipilih tempat yang belum pernah dikunjungi, dimana tempat tersebut memiliki jarak terdekat dari tempat yang sebelumnya. Bahwa *greedy* tidak mempertimbangkan nilai *heuristic*, dalam hal ini bisa berupa jarak langsung antar dua tempat.

Algoritma Greedy merupakan sebuah algoritma yang dapat menentukan sebuah jalur terpendek antara node-node yang akan digunakan dengan mengambil secara terus-menerus dan menambahkannya ke dalam jalur yang akan dilewati. Atau pada notasi big-O dituliskan $O(n^2 \log_2(n))$. Berikut algoritmanya,

1. Kelompokkan semua jalur (edge).
2. Pilih jalur yang terpendek kemudian masukkan ke dalam himpunan solusi.
3. Apakah sudah ada N jalur pada solusi? Jika tidak, ulangi langkah 2.

Dalam membentuk solusi, algoritma *greedy* pada setiap langkahnya akan mengambil pilihan yang merupakan optimum lokal atau pilihan yang sesuai dengan spesifikasi pembuat algoritma. Dengan pengambilan pilihan yang sesuai pada setiap langkah ini diharapkan solusi yang didapat optimum global atau sesuai keinginan pembuat algoritma. Pada setiap langkah algoritma *greedy*, kita akan mendapat optimum lokal. Bila algoritma berakhir maka diharapkan optimum lokal ini akan menjadi optimum global. Sehingga sebenarnya algoritma *greedy* mengasumsikan bahwa optimum lokal ini merupakan bagian dari optimum global.

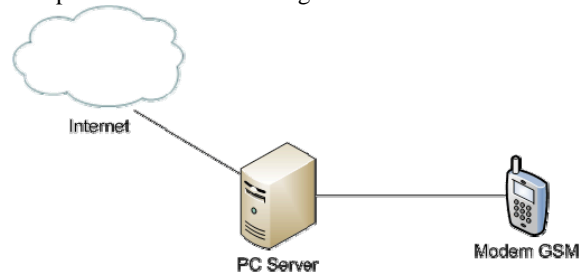
3.2 SMS Gateway

SMS (*Short Messages Services*) merupakan salah satu layanan dari berbagai operator GSM. Teknologi SMS memungkinkan kita mengirim pesan alphanumeric singkat dari sebuah *handphone* ke *handphone* yang lain. Ada sebuah ide menarik

yang kemudian diadopsi oleh berbagai portal besar saat ini, yakni menyediakan layanan pengiriman SMS dari website ke sebuah nomor *handphone* baik gratis ataupun dipungut bayaran. Mungkin anda pun tertarik tentang bagaimana membangun infrastruktur SMS Gateway yang memungkinkan kita mengirim SMS melalui jaringan TCP/IP.

Linux merupakan salah satu sistem operasi terbuka yang popularitasnya semakin menanjak. Untuk berbagai keperluan komputasi baik itu pengolahan database, *internetworking*, atau keperluan yang lain, sistem operasi Linux diakui sangat handal. Kali ini akan coba dibahas penggunaan Linux untuk keperluan SMS Gateway.

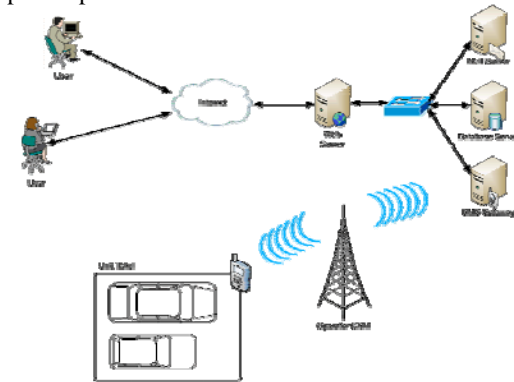
Salah satu bagan GSM SMS Gateway yang cukup sederhana adalah sebagai berikut :



Gambar 1. Sistem SMS Gateway pada tugas akhir ini.

4. Implementasi dan Hasil Pengujian

Pada tahap ini, algoritma *greedy* akan diterapkan pada *web interface* untuk dapat menangani pemesanan serta mencari optimasi dari tempat-tempat wisata yang dipilih pengguna. Serta adanya SMS gateway yang digunakan sebagai sistem notifikasi ke pengguna dan komunikasi ke para supir.



Gambar 2. Blok Diagram Sistem

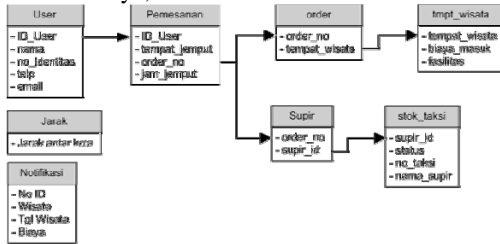
4.1. Implementasi Sistem

Untuk dapat menyelesaikan sistem layanan call center taksi wisata ini dilakukan beberapa tahapan, meliputi :

4.1.1. Pembuatan *web interface*

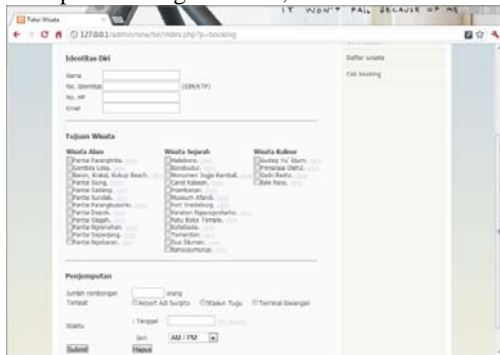
Untuk menangani interaksi aliran informasi data pada web, diperlukan adanya *database* untuk

memudahkan pengguna mendapatkan informasi dan melakukan pemesanan. Berikut relasi *database* antar tabelnya,



Gambar 3. Relasi Database

Sehingga untuk *interface web* dapat ditampilkan sebagai berikut,



Gambar 4. Halaman Pemesanan

4.1.2. Penerapan Algoritma Greedy

Setelah pengguna melakukan selesai memilih tujuan wisata, kemudian algoritma *Greedy* bekerja untuk mencari hasil optimasi dari tujuan-tujuan wisata yang telah dipilih. Berikut *flowchart* dari algoritma *Greedy*.



Gambar 5. Flowchart Algoritma Greedy

4.1.3. Pembuatan SMS Gateway

SMS *gateway* diimplementasikan sebagai sistem notifikasi serta komunikasi antara *server* dengan para supir taksi yang sedang tidak bertugas. SMS *gateway* yang digunakan adalah Gammu. Sedangkan media yang menjadi penghubung antara pemrograman PHP dengan SMS *gateway* menggunakan *database* MySQL. Dimana setiap data yang akan dikirim harus dimasukkan ke tabel *outbox*.

4.2. Hasil Pengujian

Pengujian ini merupakan sesuatu hal yang dibutuhkan untuk mengetahui performa dari sistem yang telah dibuat. Pengujian tersebut meliputi,

4.2.1. Pengujian SMS Gateway

Yang diuji pada SMS *gateway* kali ini adalah waktu pengiriman. Waktu pengiriman ini dihitung mulai dari program PHP yang membaca tabel *history* pada *database* yang kemudian memasukkannya pada tabel *outbox* pada *database*

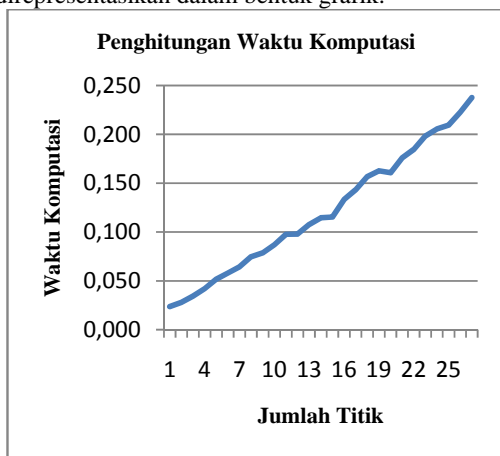
SMS. Pengujian ini melibatkan dua provider yaitu Indosat dan Telkomsel. Berikut hasil pengujiannya.

Tabel 1. Pengujian SMS

Percobaan	Provider		Waktu pengiriman (s)
	Pengirim	Penerima	
1	Indosat	Indosat	2,3
2	Indosat	Indosat	2,5
3	Indosat	Indosat	3,2
4	Indosat	Indosat	2,6
5	Indosat	Indosat	2,5
6	Indosat	Indosat	1,9
7	Indosat	Indosat	2,7
8	Indosat	Indosat	2,6
9	Indosat	Indosat	2,8
10	Indosat	Indosat	2,7

4.2.3. Pengujian Waktu Komputasi

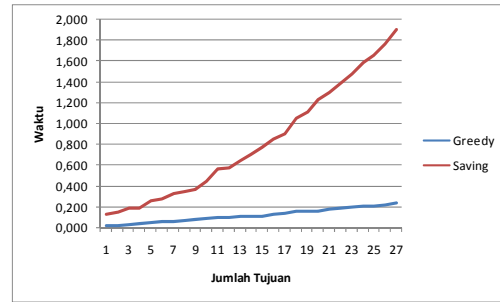
Pada pengujian ini dilakukan pengamatan antara banyak node dan waktu komputasi dari program. Parameter yang digunakan adalah banyaknya masukan dari user saat memilih tujuan wisata dan waktu yang dibutuhkan oleh program untuk memperoleh rute paling optimal dari masukan tempat wisata tersebut. Hasil pengujian direpresentasikan dalam bentuk grafik.



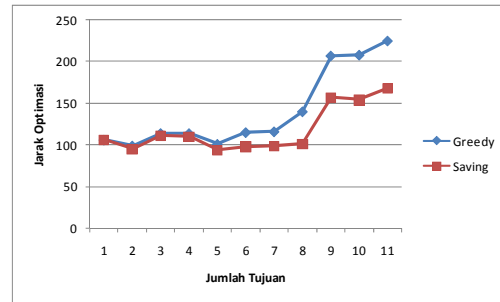
Gambar 6. Waktu Komputasi

4.2.4. Perbandingan dengan Metode Saving

Pada pengujian ini, algoritma *Saving* digunakan sebagai algoritma pembanding dalam mengeksekusi kasus TSP. dalam perbandingan ini digunakan dua parameter, yaitu waktu komputasi dan optimasi rute.



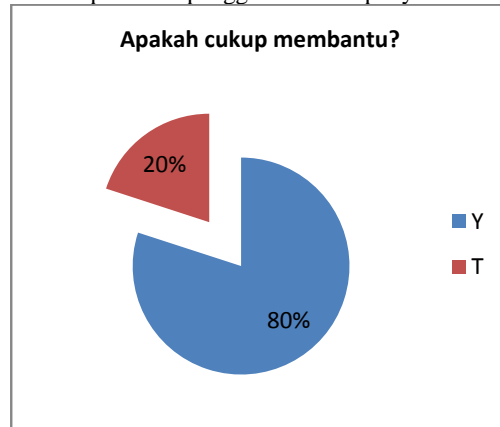
Gambar 7. Perbandingan waktu komputasi



Gambar 8. perbandingan optimasi rute

4.2.5. Hasil Respondensi

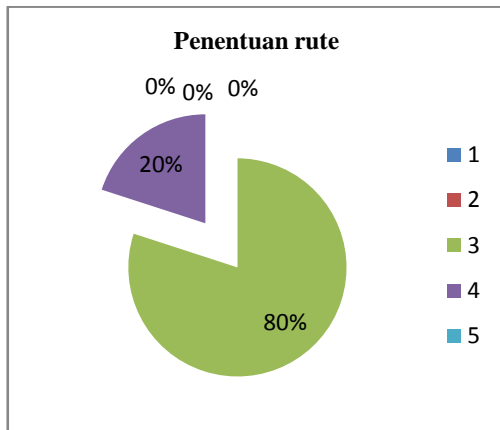
Pada pengujian ini, menggunakan opini masyarakat umum tentang layanan taksi wisata yang telah dibuat. Hasil dari respondensi ini, mayoritas masyarakat cukup terbantu dengan adanya layanan *call center* taksi wisata ini. Berikut hasil respondensi pengguna terhadap layanan ini:



Gambar 9. Respondensi Terhadap Layanan

Dari gambar 10, 85% responden mengatakan bahwa sistem layanan taksi wisata ini sudah cukup membantu. Namun, hasil dari responden tidak dapat mengatakan mutlak sistem ini benar-benar bagus karena jumlah responden hanya 20 orang.

Untuk pengujian ini, menggunakan pendapat dari supir travel pada khususnya untuk mengetahui dengan adanya penentuan rute apakah mereka cukup terbantu. Hasil dari respondensi ditunjukkan grafik dibawah ini:



Gambar 10. Responden Supir Jika Rute Ditentukan

Dengan range jawaban sebagai berikut:

1: Jelek Sekali, 2: Jelek, 3: Cukup Bagus, 4: Bagus, 5: Bagus Sekali

Dari hasil respondensi diatas, sebanyak 80% juga menyatakan bahwa penentuan rute adalah bagus jika diimplementasikan ke dalam sistem. Dan dari keseluruhan hasil respondensi dapat dikatakan bahwa sistem ini cukup membantu para supir travel.

5. Kesimpulan

Dari hasil pengujian dan analisa pada bab sebelumnya, maka pada bagian ini bisa ditarik beberapa kesimpulan sebagai berikut,

1. Algoritma *Greedy* dapat diimplementasikan pada kasus TSP yang memiliki jarak antar node-nodenya pendek. Sebagai contoh untuk kasus TSP pada tempat yang berada di dalam kota.
2. Metode yang digunakan algoritma *greedy*, adalah mencari solusi-solusi optimum lokal dan menjadikannya solusi optimum global.
3. Dibandingkan dengan algoritma lain, waktu komputasi yang dibutuhkan oleh algoritma *greedy* dalam penyelesaian kasus TSP lebih cepat.
4. Setiap metode optimasi memiliki kelebihan dan kekurangan masing-masing. Suatu metode yang performansinya sangat bagus untuk suatu masalah, bisa jadi sangat buruk atau bahkan tidak mungkin diterapkan untuk masalah lain.

6. Referansi

1. Suyanto, "Algoritma Optimasi: Deterministik atau Probabilistik", Graha Ilmu, Yogyakarta, 2010.
2. Thomas H Cormen, "Introduction to Algorithms, Second Edition", The MIT Press, London, 2001.
3. Christian Nilsson, "Heuristic for Travelling Salesman Problem", Linköping University, 2002.
4. Simon Harris, "Beginning Algorithms", Wiley Publishing Inc., Indianapolis, 2006.
5. Aulia Fitrah, "Penerapan Algoritma Genetika Pada Persoalan Pedagang Keliling (TSP)", Institut Teknologi Bandung, 2006.
6. Muhadkly, "SMS Gateway Menggunakan Gammu", Ilmu Komputer, 2007.
7. Gregory Gutin, "The Greedy Algorithm for the Symmetric TSP", Royal Holloway, University of London, 2002.
8. William, "Penjadwalan Pertandingan Liga dengan Algoritma Greedy", Institut Teknologi Bandung, 2008.
9. Wiradeva Arief K., "Perbandingan Kompleksitas Penerapan Algoritma Greedy untuk Beberapa Masalah", Institut Teknologi Bandung, 2007.