IMPLEMENTASI *CLARKE-WRIGHT SAVING METHOD* PADA LAYANAN TAKSI WISATA BERBASIS VOIP

Yusiana Kartikasari[1],Mike Yuliana, ST, MT[1], Ira Prasetyaningrum, S.Si, MT [1]
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember,Kampus ITS,Surabaya 60111
e-mail : yusiana.kartikasari@yahoo.com

Abstrak

Perkembangan teknologi telekomunikasi dan informatika saat ini semakin mempermudah manusia dalam mengakses informasi. Termasuk layanan hiburan untuk masyarakat seperti perencanaan wisata. Pada tugas akhir ini akan dibuat sistim layanan call centre taksi wisata dengan implementasi Metode Penghematan Clarke-wright untuk mempermudah masayarakat dalam berwisata.

Dari hasil pengujian diperoleh waktu komputasi algoritma saving adalah semakin banyak node tujuan maka waktu komputasi yang dibutuhkan juga semakin banyak, misal node tujuan sebanyak 5, waktu komputasi adalah 0.186 s dan untuk 10 node waktunya adalah 0.332 s, hal ini dikarenakan proses perulangan juga semakin banyak. Dan dibandingkan dengan algoritma greedy waktu komputasi hanya 0.034 s dan 0.075 s untuk 5 dan 10 titik. Sehingga dapat dikatakan waktu komputasi algoritma saving lambat. Sementara untuk jarak yang ditempuh saving untuk 5 titik dalam kota ialah 48 Km dan greedy sejauh 54 Km, jadi jarak tempuh saving lebih optimal. Rata-rata persentasi pengguna diatas 50% menganggap sistem ini membantu dalam merencanakan wisata terutama dalam hal biaya.

Kata Kunci − *Clarke-wright Saving Method*, UMS *(Unified Message System)*, VoIP *(Voice over Internet Protocol)*, IVR *(Interaktive Voice Response*),

1. Pendahuluan

Era perkembangan teknologi komputer dan telekomunikasi yang sudah melambung pesat dan cepat saat ini bermanfaat memberikan kemudahan dalam pengaksesan suatu layanan. Kemudahan informasi ini hampir terdapat pada semua layanan. Termasuk layanan hiburan untuk masyarakat seperti wisata. Saat ini telah tersedia layanan *call centre* taksi wisata yaitu transmojo yang melayani untuk wisata di Jogjakarta. Layanan taksi ini pemesanan melalui *call centre* dengan memberikan informasi lokasi penjemputan serta jam penjemputan. Layanan transmojo menggunakan tarif tetap dengan ketentuan sewa 8 jam atau 16 jam dan biaya tersebut hanya mencakup BBM dan tarif supir dan tidak termasuk biaya tiket masuk, dll. Dan layanan *call centre* ini tidak memberikan informasi mengenai tempat tempat wisata yang ada di Jogjakarta.

Pada tugas akhir ini akan dibuat sistim layanan *call centre* taksi wisata dengan implementasi Metode Penghematan *Clarke-wright*. Layanan *Call Centre* 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 jarak, berapa kilometer yang ditempuh. Metode penghematan *Clarke-wright* pada layanan taksi wisata ini akan dapat membantu untuk menginformasikan mengenai rute yang optimal kepada supir sehingga jarak yang ditempuh ke antar tempat wisata lebih optimal. Dengan adanya layanan *call centre* ini diharapkan masyarakat akan lebih mudah dalam merencanakan liburan mereka dan mudahnya mendapat informasi mengenai tempat tujuan wisata tersebut seperti biaya dan fasilitas yang ada di tempat wisata. Dan implementasi *Clarke-wright* algoritma dapat memberi kemudahan bagi pelanggan dan pemilik dalam mengatur jadwal, optimasi waktu serta biaya. Sehingga liburan yang direncanakan dapat optimal dan memberi keuntungan baik bagi wisatawan maupun pengusaha.

2. Penelitian Terdahulu

Eko Sugiarto telah mengembangkan aplikasi UMS (*Unified Messaging System*) berbasis VoIP. Aplikasi [3] ini bekerja saat terdapat panggilan selama 20 detik dan tidak ada respon maka pemanggil dapat meninggalkan pesan ke mailbox dan selanjutnya record dari mailbox akan dikirimkan ke email pemilik telepon. Anita Christine Sembiring menggunakan algoritma penghematan *Clarke-wright* untuk menentukan rute optimal pada pendistribusian produk Coca-Cola di Medan. Tujuan pembuatan aplikasi [4] ini untuk penghematan waktu pendistribusian produk ke setiap lokasi outlet serta meningkatkan kemampuan perusahaan untuk dapat memenuhi permintaan produk secara lebih cepat sehingga kepercayaan dan kepuasan konsumen meningkat.

Pada Tugas akhir ini merupakan pengembangan dari dua aplikasi diatas. Tugas akhir kali ini akan dibuat suatu layanan call centre taksi wisata berbasis VoIP. Cara kerja dari tugas akhir ini adalah pemesanan layanan taksi wisata melalui operator, jika operator sedang sibuk maka pemanggil dapat meninggalkan pesan berupa rekaman suara yang nantinya rekaman ini akan dikirimkan ke operator berupa voice mail. Pengiriman voice record ke email operator inilah yang menggunakan aplikasi Unified Messaging System

(UMS). Serta pada web server terdapat perhitungan untuk mencari jarak yang optimal serta info biaya dengan mengaplikasikan metode penghematan *Clarke – Wright*.

3. Dasar Teori Sistem

3.1 Clarke-wright Saving Method

Metode Penghematan *Clarke-wright* (*Clarke-wright Savings Method*) merupakan suatu prosedur pertukaran, dimana sekumpulan rute pada setiap langkah ditukar untuk mendapatkan sekumpulan rute yang lebih baik. Langkah-langkah pada metode ini adalah sebagai berikut:

- Menentukan *node* sebagai *node central* atau disebut *depot* dan *node node* tujuan.
- Membuat matriks jarak yaitu matriks jarak antara *depot* dengan *node* dan jarak antar *node*. Pada tugas akhir ini akan dibuat matrik jarak yang simetris.
- Membuat matriks penghematan.
- Nilai saving tertinggi merupakan rute awal.
- Pada tahap selanjutnya proses berulang itu digerakkan dari yang matrik terbesar ke matriks yang bernilai kecil, sampai masing-masing matriks penghematan itu dievaluasi untuk perbaikan rute lebih lanjut.

3.2 Asterisk

Asterisk, yang merupakan salah satu sistem server PBX open source, saat ini juga mendukung jangkauan yang luas dari protokol VOIP mencakup SIP, MGCP dan H.323. Asterisk dapat beroperasi dengan kebanyakan telepon SIP, seolah-olah sebagai gateway antara IP telepon dan PSTN. Developer Asterisk juga telah mendesain protokol baru, yaitu Inter-Asterisk eXchange, untuk melakukan efisiensi panggilan trunking antara banyak Asterisk PBX. Beberapa telepon memberi dukungan terhadap protokol IAX, yaitu protokol yang secara langsung berkomunikasi dengan server Asterisk.

3.3 PHP-AGI

AGI atau Asterisk Gateway Interface itu ada 4 macam: AGI, EAGI, FastAGI dan DeadAGI, yang pemakaiannya tergantung pada keperluan. Tapi intinya sama, yaitu sebagai *interface* komunikasi antara aplikasi. PHPAGI adalah salah satu kelas dari PHP untuk Asterisk Gateway Interface (AGI). PHP-AGI termasuk class untuk menulis script php berdasarkan pada standart interface AGI dengan berdasarkan pada perform dari fungsi asterisk manager.

3.4 Postfix Mail Server

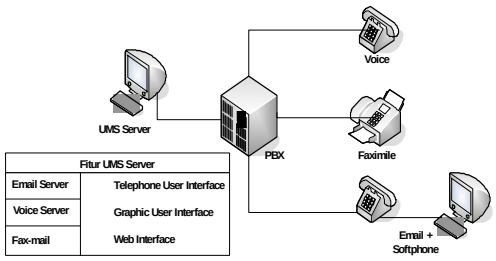
Server email Postfix dirancang dalam beberapa program kecil. Proses-proses yang ada bersifat semi-tetap. Proses-proses saling bekerja sama dalam melakukan task dalam sebuah kerja sama sejajar, bukan dalam bentuk hubungan *parent-child*. Di samping itu, Postfix mempunyai pelayanan untuk setiap program kecilnya sehingga tidak perlu mengeluarkan biaya untuk membentuk pelayanan itu.Email terdiri dari dua bagian yang terpisah baris kosong :

a. Header → sender, recipient, date, subject, delivery path,…
b. Body → Isi pesan

Software atau paket –paket yang mendukung email antara lain:

a. Webmin → untuk administrasi mail server.
b. Courier – imap → untuk POP email dari smtp ke webmail.
c. Webmail

3.5 Unified Messaging System

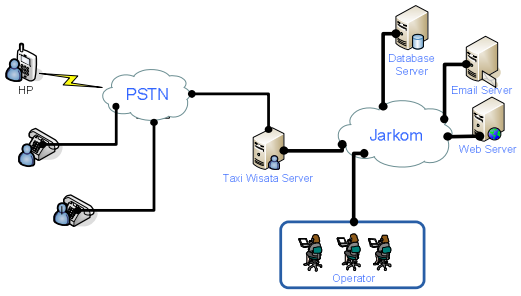


Gambar 1. Blok Diagram Sistem UMS

Unified Message merupakan gabungan dari dua kata yaitu *unified* yang berarti menyatukan dan *message* yang berarti pesan, dari dua arti etimologi *unified message* berarti Integrasi dari beberapa media komunikasi seperti seorang user yang dapat menerima dan mengirim suara , fax, email dengan satu interface , baik itu berupa jalur telephone, wireless phone, PC, internet-enabled PC.

4. Implementasi dan Hasil Pengujian

Pada tahap ini, *Clarke-wright* Saving *Method* akan diterapkan pada web admin untuk menangani pemesanan. Serta asterisk akan diterapkan untuk membangun layanan *call center* berbasis VoIP. Blok Diagram sistem dapat dilihat pada Gambar 2.



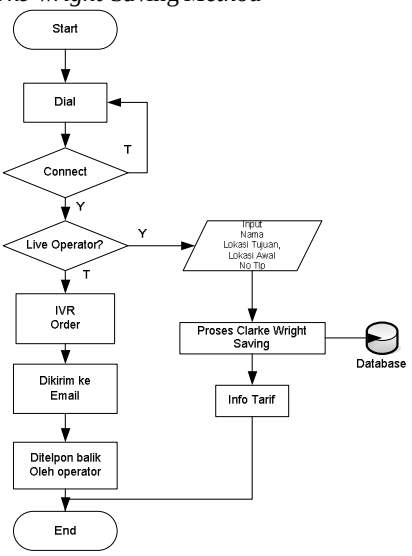
Gambar 2. Blok Diagram Sistem

4.1. Implementasi Sistem

Untuk menyelesaikan layanan *call center* taksi wisata ini dilakukan beberapa tahap meliputi:

1. Konfigurasi asterisk
2. Pembuatan UMS

3. Pembuatan web admin dengan implementasi *Clarke-wright* Saving *Method*



Gambar 2a. Flowchart Sistem

4.1.1. Konfigurasi Asterisk

Konfigurasi asterisk ini untuk membangun layanan *call center*. Ada dua macam file yang harus dikonfigurasi. Untuk konfigurasi /etc/asterisk/sip.conf berisi tentang inisialisasi ekstensi yang akan digunakan. Berikut contoh konfigurasinya:

```
[102]
type=friend
username=102
secret=102
host=dynamic
nat=no
dtmfmode=rfc2833
allow=all
callerid="SIP102"
context=taksi
canreinvite=no
mailbox=102@taksi
```

Selanjutnya adalah konfigurasi pada file /etc/asterisk/extensions.conf .File konfigurasi disini berisi dial plan. Contoh konfigurasinya sebagai berikut:

```
exten => 102,1,dial(sip/102,5)
exten => 102,2,AGI(kirim.php)
exten => 102,3,hangup
```

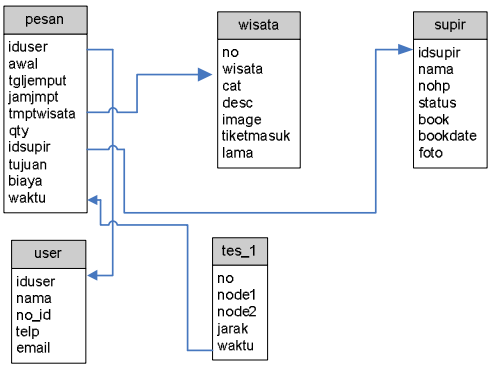
4.1.2. Pembuatan UMS

Untuk pembuatan aplikasi UMS ini digunakan bahasa pemrograman PHP. Yang nantinya akan dijalankan apabila operator sedang sibuk. Berikut potongan program yang digunakan untuk mengirim email:

```
<?php
$to="someone@example.com";
$subject="Testmail";
$message = "Hello! This is a simple email
message.";
$from = "someonelse@example.com";
$headers = "From: $from";
mail($to,$subject,$message,$headers);
echo "Mail Sent.";
?>
```

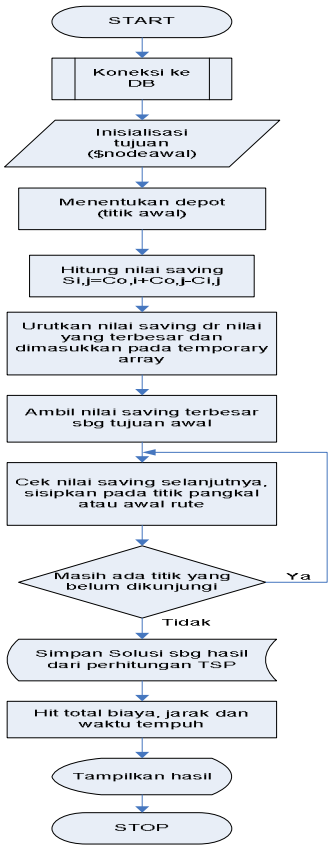
4.1.3. Pembuatan web admin dengan Implementasi *Clarke-wright* Saving *Method*

Untuk pembuatan web admin, dibutuhkan database yang akan digunakan untuk membantu admin dalam memberikan informasi aupun untuk melakukan pemesanan. Pada tugas akhir ini dibuat lima buah tabel dalam satu database. Berikut relasi antar tabel:



Gambar 3. Relasi Database

Untuk Implementasi *Clarke-wright* Saving *Method* diletakkan setelah pengguna melakukan pemesanan. Berikut flowchart dari *Clarke-wright* Saving *Method*:



Gambar 4. Flowchart *Clarke-wright* Saving

4.2. Hasil Pengujian Sistem

Pengujian merupakan salah satu langkah penting yang harus dilakukan untuk mengetahui apakah sistem yang dibuat telah sesuai dengan apa yang direncanakan. Pada bagian ini akan dilakukan pengujian dan analisa sistem meliputi:

1. Pengujian Sistem meliputi pengujian:
   - Hunting
   - Deteksi Digit
   - Waktu Eksekusi UMS
2. Pengujian Algoritma meliputi:
   - Waktu Komputasi
   - Perbandingan dengan Algoritma Greedy
3. Respondensi Terhadap Sistem

4.2.1. Pengujian Sistem

Pengujian ini bertujuan untuk mengetahui apakah layanan *call center* yang dibuat telah bekerja dengan baik atau tidak.

4.2.1.1. Pengujian Hunting

Untuk pengujian pertama akan dilakukan pengujian hunting, dimana pada *call center* ini dibuat apabila line sedang sibuk akan mencari line yang kosong dan apabila seluruh line operator penuh maka akan diterima olhe IVR. Pada pengujian ini akan dihitung waktu yang dibutuhkan untuk melakukan sambung ke operator layanan taksi wisata. Hasil dari pengujian direpresentasikan dalam bentuk tabel dibawah:

Tabel 1. Pengujian Hunting

Operator/Kondisi	101	102	103	yang menerima	Waktu (s)
1	on-hook	on-hook	on-hook	101	2.6
2	off-hook	on-hook	on-hook	102	4.6
3	off-hook	off-hook	on-hook	103	7.1
4	off-hook	off-hook	off-hook	IVR	7.4

Dari tabel diatas, dapat dilihat waktu yang dibutuhkan untuk tersambung ke line operator bergantung pula pada ada tidaknya operator yang sibuk. Makin banyak operator yang sibuk maka waktu yang dibutuhkan juga bertambah. Hal ini disebabkan butuh waktu untuk berpindah line dan mencari line yang kosong apabila yang line dituju sedang sibuk.

4.2.1.2. Pengujian Deteksi Digit

Pada pengujian deteksi digit ini, akan dilakukan pengujian sebanyak 10 kali untuk memasukkan digit dan akan diamati apakah digit yang dikirimkan ke *email* sesuai dengan digit yang telah dimasukkan pada saat proses pemesanan. Berikut hasil pengujian untuk deteksi digit, direpresentasikan dalam bentuk tabel.

Tabel 2. Pengujian Deteksi Digit

Percobaan	Jumlah Digit	Digit yang Ditekan	Digit yang Dikenali	Cocok	
				Y	T
1	5	12345	12345	√	
2	6	123456	123456	√	
3	7	1234567	1234567	√	
4	8	12345678	12345678	√	
5	9	123456789	123456789	√	
6	10	1234567890	1234567890	√	
7	11	12345678901	12345678901	√	
8	12	123456789012	123456789012	√	
9	13	1234567890123	1234567890123	√	
10	14	12345678901234	12345678901234	√	

Dari tabel diatas untuk deteksi digit hasilnya cocok atau sesuai dengan digit yang ditekan. Dari hasil ini dapat diketahui bahwa program deteksi digit sudah bekerja dengan baik.

4.2.1.3. Pengujian Waktu Eksekusi UMS

Pengujian waktu eksekusi UMS ini dilakukan dengan cara menghitung waktu eksekusi dari pelanggan memasukkan nomor telepon hingga operator menerima *email* yang berisi nomor telepon pelanggan tersebut. Pada pengujian ini akan dilakukan pengujian sebanyak 10 kali.

Tabel 3. Pengujian waktu eksekusi UMS

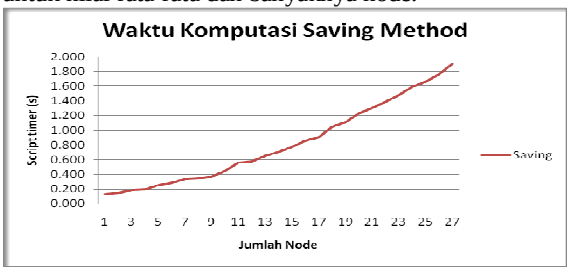
Percobaan	Waktu Komputasi (s)
1	43.18
2	40.08
3	40.63
4	43.76
5	37.81
6	40.16
7	38.35
8	39.64
9	41.14
10	41.3
average	**40.605**

Dari tabel hasil pengujian diatas, dapat diketahui bahwa rata-rata waktu yang diperlukan dari pelanggan memasukkan nomor telepon hingga *email* sampai ke customer adalah 40 detik. Dari hasil pengujian dapat dikatakan bahwa program UMS dapat bekerja dengan baik dan waktu yang dibutuhkan tidak terlalu lama untuk mengirim *email* ke operator.

4.2.2. Pengujian Algoritma

4.2.2.1. 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 untuk nilai rata-rata dan banyaknya node.



Gambar 5. Waktu Komputasi Algoritma Saving

Dari Gambar 5 diketahui bahwa untuk 10 titik tujuan waktu komputasi rata-rata adalah 0.346 detik

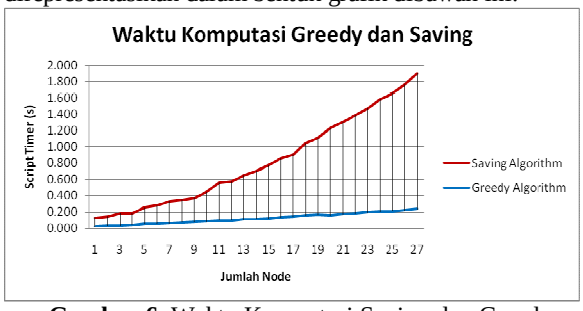
dan untuk 15 titik tujuan waktu komputasi adalah 0.648 detik. Hal ini menunjukkan bahwa program membutuhkan waktu komputasi yang lebih lama untuk node yang lebih banyak. Dan kenaikan waktu untuk algoritma *saving* ini agak tajam. Sehingga dapat dikatakan bahwa makin banyak node yg diproses makin lama pula waktu komputasi yang dibutuhkan.

.

4.2.2.2. Perbandingan Algoritma Saving dengan Algoritma Greedy

- **Perbandingan Waktu Komputasi**

Waktu komputasi disini adalah waktu yang dibutuhkan oleh program untuk melakukan perhitungan dari memperoleh masukan data hingga menghasilkan rute. Hasil pengujian menunjukkan semakin banyak node yang digunakan waktu komputasi *saving* dan greedy juga makin lama. Tetapi waktu komputasi Greedy jauh lebih cepat dibandingkan komputasi *saving*. Dari hasil pengujian direpresentasikan dalam bentuk grafik dibawah ini:

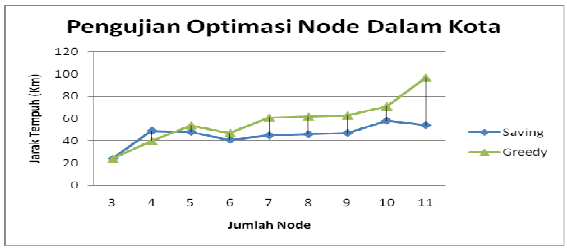


Gambar 6. Waktu Komputasi Saving dan Greedy

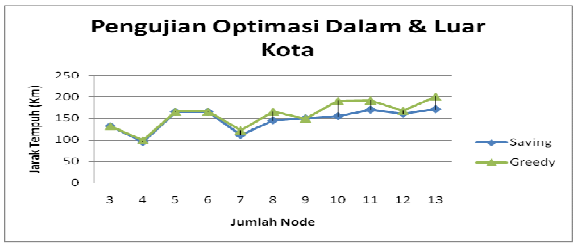
Dari gambar 6 dapat diketahui waktu komputasi algoritma greedy dengan 10 titik tujuan adalah 0.075 detik sementara untuk algortima *saving* adalah 0.346 detik. Sehingga dapat dikatakan waktu komputasi algoritma *saving* lebih baik dari algoritma greedy. Hal ini disebabkan pada algoritma *saving* pemrosesan data lebih panjang dibanding algoritma greedy. Pada algoritma *saving* data jarak yang diperoleh akan dihitung nilai *saving* lalu data diurutkan dan untuk kemudian baru proses penentuan rute. Sementara pada algoritma greedy data jarak yang diperoleh tidak dihitung melainkan langsung ke proses penentuan rute. Hal inilah yang menyebabkan waktu komputasi algoritma *saving* lebih lama dibandingkan algoritma greedy.

- **Perbandingan Optimasi Rute**

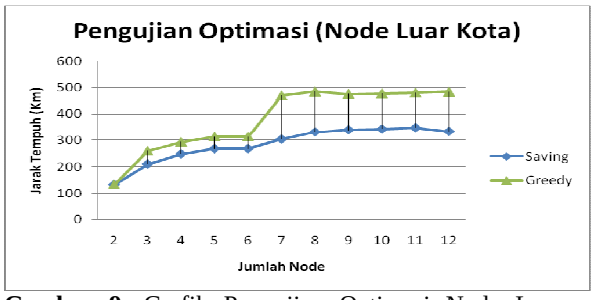
Pengujian selanjutnya yaitu perbandingan optimasi rute, untuk pengujian ini akan dilakukan pengujian panjang jarak yang ditempuh dari hasil proses perhitungan algoritma. Makin dekat jarak yang tempuh yang dihasilkan, maka algoritma tersebut dikatakan makin optimal dalam menghasilkan rute. Pada pengujian dilakukan tiga kali pengujian untuk node dalam kota, dalam dan luar kota serta node luar kota. Hasil dari ketiga pengujian tersebut direpresentasikan dalam bentuk grafik sebagai berikut:



Gambar 7. Grafik Pengujian Optimasi Node Dalam Kota



Gambar 8. Grafik Pengujian Optimasi Node Dalam & Luar Kota



Gambar 9. Grafik Pengujian Optimasi Node Luar Kota

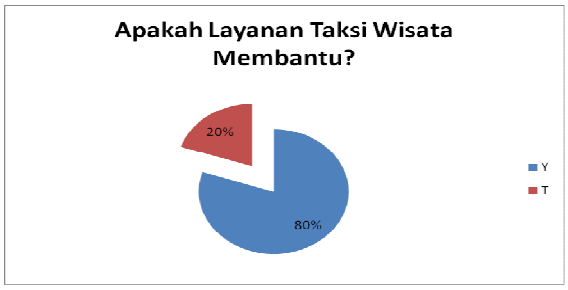
Dari semua pengujian diatas dapat ditarik kesimpulan bahwa untuk penentuan rute algoritma *saving* lebih baik dibandingkan dengan algoritma greedy, karena hasil perhitungan algoritma *saving* lebih banyak menghasilkan rute yang lebih optimal. Dikatakan lebih optimal karena total jarak tempuh hasil proses algoritma *saving* lebih sedikit. Contoh untuk jumlah node 3, hasil perhitungan algoritma *saving* menunjukkan total jarak tempuh 210 Km dan biaya sebesar Rp. 818.000, untuk algoritma greedy dihasilkan total jarak tempuh sejauh 294 Km dan biaya sebesar Rp. 1.114.000. Dari hasil perhitungan tersebut tampak bahwa jarak tempuh dan biaya hasil perhitungan algoritma *saving* lebih optimal.

.

4.2.3. Respondensi Terhadap Sistem

4.2.3.1. Respondensi Pengguna

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:

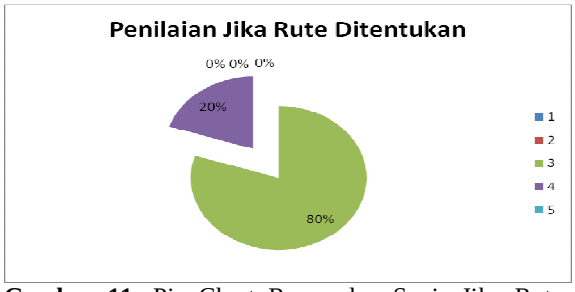


Gambar 10. Pie Chart Respondensi Terhadap Layanan Taksi Wisata

Dari gambar 10, 80% 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.

4.2.3.2. Respondensi Pengendara

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 11. Pie Chart 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 diimplemetasikan ke dalam sistem. Dan dari keseluruhan hasil respondensi dapat dikatakan bahwa sistem ini cukup membantu para supir travel.

5. Kesimpulan

a. Dibutuhkan waktu sekitar tiga detik untuk tersambung ke operator dan waktu yang dibutuhkan makin banyak bergantung dari operator yang sibuk saat itu.
b. Waktu rata-rata yang dibutuhkan untuk pengiriman *email* ke operator dari pengguna memasukkan nomor handphone hingga email diterima oleh operator adalah 40 detik.
c. Waktu Komputasi *Clarke-Wright Saving Method* makin lama seiring dengan banyaknya titik tujuan. Untuk 10 titik tujuan waktu komputasi rata-rata adalah 0.35 detik, sementara untuk 15 titik tujuan waktu komputasi menjadi 0.65 detik. Hal ini disebabkan proses komputasi *Clarke-Wright Saving Method* menggunakan proses perulangan ke-n, makin banyak nilai n (data) maka proses perulangan akan semakin lama.
d. Dibandingkan dengan Algoritma Greedy, waktu komputasi algoritma *saving* jauh lebih lama. Untuk 10 titik tujuan algoritma greedy memiliki waktu komputasi 0.075 detik, sementara algoritma *saving* memiliki nilai 0.35 detik.
e. Untuk perbandingan optimasi rute, hasil perhitungan algoritma saving lebih optimal dibandingkan dengan algoritma greedy, dimana untuk 10 titik kunjungan di dalam kota, hasil dari algoritma greedy jarak yang harus ditempuh sejauh 71 Km, sementara untuk algoritma saving total jarak yang ditempuh adalah 58 Km. Dikatakan optimal disini karena total jarak yang ditempuh lebih sedikit bila menggunakan hasil perhitungan algoritma saving.
f. Hasil dari respondensi, mayoritas mengatakan bahwa sistem ini cukup membantu dibuktikan dari 20 orang responden pengguna, 80% mengatakan bahwa sistem ini cukup membantu untuk berwisata. Sementara dari responden masayarakat yg berprofesi supir sebanyak 10 orang, 60% mengatakan bahwa sistem ini bagus.

6. Referensi

[1] Bunafit Nugroho. “Database Relasional dengan MySQL”, Andi Offset, Yogyakarta, 2005.
[2] Abdul Kadir, “Dasar Pemrograman Web Dinamis menggunakan PHP”, Andi Offset, Yogyakarta, 2002.
[3] Eko Sugiarto, “Pembuatan Aplikasi UMS Berbasis Jaringan VoIP”, Surabaya,2010.
[4] Christine Anita Sembiring, “Penentuan Rute Distribusi Produk yang Optimal dengan menggunakan algoritma heuristik pada PT. Coca Cola Bottling Indonesia Medan”, Medan, 2008.
[5] Jim van Meggelen, Leif Madsen dan Jared Smith, “Asterisk The Future of Telephony 2nd Edition”, O’Reilly Media, United States of America, 2005.
[6] Simionovich Nir, “ Asterisk Gateway Interface 1.4 and 1.6 Programming”, 2009.
[7] http://digilib.petra.ac.id/
[8] Simon de Givry,” A Brief Introduction to combinatorial optimization” The Travelling Salesman Problem”, Thales Research & Technology, France, 2001.
[9] Mathirajan.M , “Logistics Planning”, Indian Institute of Science, Bangalore, India,2000.
[10] Johnson David S. & McGeoch Lyle A. , “The Travelling Saleman Problem : A case study in Local Optimization”, AT&T Labs, Florham Park, Department of Mathematics and Computer Science, Amherst College, 1995.