

**PROYEK AKHIR
MATA KULIAH ALGORITMA EVOLUSI
SEMESTER GANJIL 2013-2014**

**OPTIMASI RUTE PENGIRIMAN LAUNDRY DENGAN *TIME*
WINDOWS (VRPTW) MENGGUNAKAN ALGORITMA
GENETIKA**



Disusun oleh:

Kelompok B Kelas A

Shinta Ayu Valensia	(105090601111013)
Meitasari Winardi S	(105090603111001)
Anjar Dwi Oktavianing	(105090604111003)
Dita Sundarningsih	(105090607111035)
Eunike Endariahna S	(115060801111081)

Dosen Pengajar: Wayan Firdaus Mahmudy, Ph.D.

**PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG**

ABSTRAK

Pada kehidupan sehari-hari pengiriman barang membutuhkan jalur yang optimal agar barang-barang tersebut bisa sampai ditempat tujuan sesuai dengan permintaan dan waktu tempat tujuan tersebut. Dalam kasus ini adalah pengiriman laundry ke customer. Setiap customer memiliki kriteria tersendiri dalam pengiriman laundry mereka, misalnya customer hanya bisa menerima kiriman laundry pada waktu tertentu. Keterlambatan pengiriman laundry akan mempengaruhi kredibilitas dari pemilik laundry tersebut, dan mungkin bisa mengurangi jumlah customer. Kasus pengiriman laundry ini merupakan contoh penerapan dari Vehicle Routing Problem with Time Windows (VRPTW) pada kehidupan nyata. VRPTW adalah sebuah permasalahan pencarian rute untuk sejumlah kendaraan dari suatu tempat menuju node-node yang tersedia dengan tujuan mengantarkan barang dari tempat menuju tempat asal menuju node-node tujuan dengan batasan Time Windows pada setiap node. Dalam masalah VRPTW ini akan digunakan metode optimasi dari Algoritma Evolusi secara random untuk mendapatkan hasil yang optimal

Kata kunci: Pencarian Rute, Vehicle Routing Problem With Time Windows (VRPTW), Node, Optimasi dan Algoritma Evolusi

DAFTAR ISI

ABSTRAK.....	2
DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	5
2.1. Vehicle Routing Problems.....	5
2.2. Vehicle Routing and Scheduling.....	7
2.3. Penyelesaian Vehicle Routing Problems.....	8
2.3.1. Solusi Eksak.....	8
2.3.2. Heuristik.....	8
2.3.3. Metaheuristik.....	9
BAB III METODE PENELITIAN.....	10
3.1 Identifikasi Masalah.....	10
3.2 Studi Literatur.....	11
3.3 Proses Pengambila Data.....	11
3.4 Pengolahan Data dan Analisis Data.....	11
3.5 Penyelesaian Permasalahan Penentuan Rute Kendaraan Dalam Pemenuhan Permintaan Customer Dengan Kendala <i>Time Window</i>	12
3.6 Proses Manualisasi.....	12
BAB IV ANALISA DA PERANCANGAN.....	20
4.1. Spesifikasi Software dan Hardware.....	20
4.1.1. Spesifikasi Software.....	20
4.1.2. Spesifikasi Hardware.....	20
4.2. Desain Antar Muka.....	21
BAB V IMPPLEMENTASI DAN PEMBAHASAN.....	22

5.1.	Spesifikasi Sistem	22
5.1.1.	Spesifikasi Perangkat Keras	22
5.1.2.	Spesifikasi Perangkat Lunak	23
5.2.	Batasan- Batasan Implementasi	23
5.3.	Implementasi Algoritma	24
5.3.1.	Implementasi Algoritma Proses Pembangkitan Populasi Awal	24
5.3.2.	Implementasi Algoritma Proses Reproduksi	24
5.3.3.	Implementasi Algoritma Proses Penghitungan Nilai Fitness	27
5.3.4.	Implementasi Algoritma Proses Sorting Nilai Fitness	28
5.4.	Implementasi Antarmuka	31
5.4.1.	Implementasi Tampilan Data Awal	32
5.4.2.	Implementasi Inputan Data Set	32
5.4.3.	Implementasi Tampilan Hasil Seleksi Sebelum di Sorting	33
5.4.4.	Implementasi Tampilan Hasil Seleksi Setelah di Sorting	33
BAB VI PENUTUP		35
6.1.	Kesimpulan	35
6.2.	Saran	35
DAFTAR PUSTAKA		36

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pemilik laundry yang memiliki jasa layanan antar, mengharapkan rute optimal baik dalam waktu, ketepatan dan biaya dalam pengiriman setiap laundry. Tidak semua customer memiliki waktu senggang yang banyak. Beberapa customer memiliki rentang waktu tertentu dalam menerima kiriman laundry. Lama perjalanan pengiriman laundry akan mengurangi kredibilitas dari pemilik laundry dan juga mungkin akan mengurangi jumlah customer apabila keterlambatan sering terjadi.

Selain itu ketepatan dalam pengiriman barang akan memengaruhi kepercayaan customer terhadap pemilik laundry. Waktu yang tepat dalam pengiriman laundry juga loyalitas customer kepada pemilik laundry. Mengirim laundry tidak hanya di satu tempat saja, dalam suatu waktu bisa mengantarkan laundry ke beberapa tempat customer. Bagi pemilik laundry yang memiliki layanan antar, diharapkan dapat mencapai waktu yang tepat dengan rute yang terpendek, jika tidak akan menimbulkan kerugian baik bagi pemilik laundry maupun customer. Permasalahan *Vehicle Routing Problem with Time Windows* (VRPTW) pada kasus nyata pengiriman laundry ini mampu mendapatkan rute dengan metode optimasi dari algoritma evolusi. Dengan mencoba secara random setiap kemungkinan rute yang bisa ditempuh diharapkan mendapatkan hasil yang optimal.

1.2. Rumusan Masalah

Dari uraian latar belakang diatas, maka didapatkan rumusan masalah untuk Optimasi Rute Pengiriman Laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika adalah sebagai berikut :

1. Bagaimana penggunaan metode *Vechile Routing Problem with Time Windows*(VRPTW) pada pencarian rute terpendek pengiriman *laundry*?
2. Bagaimanamengukur tingkat *fitness* dari rute yang telah dipilih secara random pada kasus *Vechile Routing Problem with Time Windows*(VRPTW)?
3. Bagaimana menentukan hasil optimasi berdasarkan nilai *fitness* yang telah dicari dari rute yang telah dipilih pada kasus *Vechile Routing Problem with Time Windows*(VRPTW)?

1.3. Batasan Masalah

Dari uraian rumusan masalah diatas, maka batasan masalah yang digunakan masalah untuk Optimasi Rute Pengiriman Laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika adalah sebagai berikut :

1. Software ini dibuat dengan bahasa pemrograman Java.
2. Output yang dikeluarkan adalah solusi terbaik yang sudah melalui proses seleksi Elitism
3. Menggunakan metode *Vechile Routing Problem with Time Windows*(VRPTW).
4. Data dibangkitkan secara random

1.4. Tujuan

Sesuai dengan rumusan dan latar belakang, tujuan dari penelitian ini adalah pencarian rute terpendek untuk pengiriman laundry menggunakan metode *Vehicle Routing Problem with Time Windows*(VRPTW).

1.5. Manfaat

Manfaat yang didapatkan dengan adanya optimasi pencarian rute ini, kita dapat memilih rute secara random dan didapatkan hasil yang optimal atau mendekati optimal. Jasa kiriman laundry dapat mengerjakan tugasnya dengan tepat dan mengalami keuntungan termasuk bagi customer tersebut.

1.6. Sistematika Penulisan

Sistematika penyusunan laporan ini secara garis besar meliputi beberapa bab, yaitu sebagai berikut:

BAB I : Pendahuluan

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah, manfaat dan tujuan dari penelitian dari optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika.

BAB II : Dasar Teori

Menguraikan tentang dasar teori dan referensi yang mendasari penelitian optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika

BAB III: Metodologi Penelitian

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi dari optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika.

BAB IV :Analisa dan Perancangan

Membahas analisis kebutuhan dan perancangan aplikasi optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika.

BAB V : Implementasi dan Pembahasan

Membahas proses penerapan dan tahap-tahap yang dilakukan dalam penelitian optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika.

BAB VI : Kesimpulan dan Saran

Memuat kesimpulan yang diperoleh dari penelitian optimasi rute pengiriman laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika.

BAB II

DASAR TEORI

2.1. Vehicle Routing Problems

Logistik mempunyai pengaruh yang signifikan terhadap biaya dan keputusan suatu perusahaan. Logistik juga berpengaruh untuk menghasilkan level pelayanan kepada konsumen yang berbeda-beda. Tujuan akhir manajemen logistik adalah mendapatkan sejumlah barang atau jasa yang tepat pada tempat dan waktu yang tepat, serta kondisi yang diinginkan dengan memberikan kontribusi terbesar bagi perusahaan. Untuk mencapai tujuan akhir manajemen logistik, diperlukanlah suatu system distribusi produk yang :

- Memastikan bahwa produk yang tersedia pada waktu dan jumlah yang tepat sesuai permintaan konsumen
- Memiliki kualitas yang terjamin
- Memperhatikan tingkat keselamatan dalam pendistribusiannya.

Suatu perusahaan harus dapat mengoptimalkan sistem distribusinya agar dapat bersaing dengan perusahaan sejenis lainnya. Salah satu caranya adalah dengan pengoptimalan transportasi. Salah satu permasalahan dalam transportasi adalah *Vehicle Routing Problems* (VRP), yaitu merancang rute kendaraan dengan biaya rendah dimana tiap kendaraan berawal dan berakhir di depot, setiap konsumen hanya dilayani sekali oleh sebuah kendaraan, serta total permintaan yang dibawa tidak melebihi kapasitas kendaraan. Transportasi ini memberikan kontribusi biaya $1/3$ sampai $2/3$ dari total biaya distribusi. *Vehicle Routing Problems* (VRP), pertama kali dikenalkan oleh Dantzig dan Ramser pada tahun 1959. VRP ini memegang peranan penting pada manajemen distribusi dan telah menjadi salah satu permasalahan dalam optimalisasi kombinasi yang dipelajari secara luas. VRP merupakan manajemen distribusi barang yang memperhatikan pelayanan, periode waktu tertentu, sekelompok konsumen dengan sejumlah kendaraan yang berlokasi pada satu atau lebih depot yang dijalankan oleh sekelompok pengemudi, menggunakan road penentuan rute network yang sesuai.

Solusi dari sebuah VRP yaitu menentukan sejumlah rute, yang masing-masing dilayani oleh suatu kendaraan yang berasal dan berakhir pada depotnya, sehingga kebutuhan pelanggan terpenuhi, semua permasalahan operasional terselesaikan dan biaya transportasi secara umum diminimalkan.

Di bawah ini merupakan karakteristik konsumen dalam *Vehicle Routing Problems*:

- Menempatkan road graph dimanakonsumenberada
- Adanya demand dalam berbagai tipe dan harus diantarkan ketempat konsumen
- Terdapat perio dewaktu (*time window*) dimana konsumen dapat dilayani
- Waktu yang dibutuhkan untuk mengantarkan barang kelokasi konsumen(*loading time*), hal tersebut dapat berhubungan dengan jenis kendaraan
- Sekelompok kendaraan tersedia digunakan untuk melayani konsumen

Di bawah ini merupakan tujuan umum dari *Vehicle Routing Problems*, diantaranya adalah :

- Meminimalkan biaya transportasi global, terkait dengan jarak dan biaya tetap yang berhubungan dengan kendaraan
- Meminimalkan jumlah kendaraan (atau pengemudi) yang dibutuhkan untuk melayani semua konsumen.
- Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan
- Meminimalkan *penalty* akibat servis yang kurang memuaskan dari konsumen

Menurut toth dan vigo (2002) ditemukan variasi permasalahan utama vrp yaitu:

- Kapasitas terbatas dimiliki oleh setiap kendaraan (*capacitatedvrp-cvrp*)
- Barang dikirim untuk periode tertentu pada setiap konsumen (*vrp With Time Windows-vrptw*)
- Vendor menggunakan banyak depot untuk mengirimi konsumen (*multiple depot vrp-mdvrp*)
- Barang dapat dikembalikan ke depot oleh konsumen (*vrp with pick up and delivering-vrppd*)
- Konsumen dilayani dengan menggunakan kendaraan yang berbeda-beda (*split delivery vrp-sdvrp*)
- Beberapa besaran (seperti jumlah konsumen, jumlah permintaan, Waktu layanan dan waktu perjalanan)

2.2. Vehicle Routing and Scheduling

Vehicle Routing and Scheduling merupakan perluasan dari *Vehicle Routing Problem*. Beberapa batasan yang realistis yang termasuk didalamnya adalah sebagai berikut :

1. Dalam setiap titik pemberhentian, ada sejumlah volume yang diambil dan dikirim.
2. Beragam kendaraan kemungkinan digunakan, disebabkan karena beragam batasan kapasitas pengangkutan.
3. Maksimum total waktu kerja operator kendaraan untuk melakukan pengiriman sebelum periode istirahat selama kurang lebih 8 jam.
4. Titik pemberhentian (konsumen) hanya memperbolehkan pengiriman dan/atau pengambilan produk pada waktu tertentu (disebut: *Time Windows*).
5. Pengambilan hanya boleh dilakukan setelah pengiriman.
6. Operator kendaraan diperbolehkan istirahat atau makan siang pada waktu tertentu.

Beberapa batasan diatas menambah kompleksitas masalah routing ini dan mempersulit kita dalam pemilihan solusi yang paling optimal. Solusi yang paling optimal dapat diperoleh dengan cara menerapkan beberapa panduan untuk menghasilkan routing dan scheduling yang baik atau beberapa prosedur *logical Heuristic* dengan pertimbangan kendaraan memulai perjalanan dari pabrik (depot), menuju ke beberapa titik pemberhentian (stop) untuk melakukan pengiriman, dan kembali ke pabrik (depot) pada hari yang sama.

Permasalahan untuk mendapatkan hasil solusi yang optimal dari pemecahan VRP (*Vehicle Routing Problems*) menjadi bertambah jika terdapat penambahan kendala (*constraint*) pada kasus yang harus diselesaikan. Kendala-kendala tersebut antara lain batasan waktu (*time window*), jenis kendaraan angkut yang berbeda-beda kapasitas angkutnya, total waktu maksimum operator kendaraan melakukan pengiriman, hambatan-hambatan yang di perjalanan, waktu istirahat operator kendaraan ketika melakukan pengiriman dan lain sebagainya. Dari banyak pendekatan untuk memecahkan masalah VRP terdapat dua metode yang paling umum digunakan yaitu *sweep method* dan *savings method*. Kedua metode tersebut merupakan tehnik pemecahan VRP secara heuristic.

2.3. Penyelesaian Vehicle Routing Problems

Pada dasarnya terdapat 3 macam penyelesaian *Vehicle Routing Problems*, yaitu Solusi Eksak, Heuristik dan Metaheuristik.

2.3.1. Solusi Eksak

Pada solusi eksak dilakukan pendekatan dengan menghitung setiap solusi yang mungkin sampai satu terbaik dapat diperoleh. *Branch and bound* dan *branch and cut* merupakan contoh dari penyelesaian eksak.

2.3.2. Heuristik

Metode heuristic memberikan suatu cara untuk menyelesaikan permasalahan optimasi yang lebih sulit dan dengan kualitas dan waktu

penyelesaian yang lebih cepat daripada solusi eksak. Contoh metode heuristik antara lain: *saving based*, *matching based*, *multiroute improvement heuristic*, dll.

2.3.3. Metaheuristik

Metaheuristik, adalah suatu metode untuk melakukan eksplorasi yang lebih dalam pada daerah yang menjanjikan dari ruang solusi yang ada. Kualitas solusi yang dihasilkan dari metode ini jauh lebih baik daripada yang didapat heuristik klasik. Contoh *metaheuristik* adalah *genetic algorithm*, *simulated annealing*, *tabu search*, *ant colony system* dsb.

BAB III

METODE PENELITIAN

Pada bab ini akan dijelaskan langkah-langkah yang dilakukan dalam proses pengambilan data, pengolahan data, dan pengkajian pustaka. Dalam hal ini yang dilakukan adalah mengamati proses dari pengiriman barang yang dilakukan pada laundry agar mengetahui cara proses kerjanya, sehingga bisa dilakukan representasi kromosom pada Individu dengan menggunakan metode *Vehicle Routing Problem With Time Windows* (VRPTW).

3.1 Identifikasi Masalah

Untuk pelayanan pengiriman barang kepada *customer*, laundry mengoperasikan sebuah kendaraan dari laundry untuk melayani beberapa *customer* yang berupa pengantaran permintaan barang dalam interval waktu yang telah ditentukan oleh *customer*.

Permasalahan penentuan rute untuk memenuhi permintaan dari customer oleh laundry dapat dideskripsikan sebagai berikut:

1. Dimana terdapat sebuah laundry tunggal yang bisa melayani permintaan beberapa customer
2. Customer hanya akan dilayani oleh kendaraan sekali sesuai dengan *time window*nya.
3. Setiap customer disini mempunyai permintaan, dan waktu servis serta *time window*. *Time window* didefinisikan sebagai interval waktu yang diberikan oleh *customer* kepada laundry untuk dapat mengirim barang, dengan waktu awal berupa jam buka dan waktu akhir berupa jam tutup.

Data mengenai jumlah *customer*, jarak *customer* dari tempat laundry serta jarak *customer* satu menuju *customer*lain diketahui dalam satuan waktu yaitu menit. Permasalahan untuk menentukan rute dalam formulasi VRPTW bertujuan untuk meminimalkan biaya rute dari jarak yang ditempuh dengan memperhatikan waktu tempuh yang disediakan oleh *customer*.

3.2 Studi Literatur

Studi literatur ini bertujuan untuk mempelajari teori-teori yang sesuai dengan masalah yang dibahas untuk membantu memecahkan pemecahan masalah tersebut.

3.3 Proses Pengambila Data

Dalam suatu representasi kromosom pada sejumlah individu, pengiriman barang pada laundry dibutuhkan data yang digunakan sebagai parameter input maupun dalam pemrosesan data. Pada tahap dilakukan pengumpulan data yang akan digunakan untuk menyelesaikan permasalahan rute pengiriman barang pada laundry. Dimana data yang dibutuhkan adalah jumlah data, jarak antar node, jumlah waktu yang dibutuhkan dalam proses pengiriman barang mulai dari buka, tutup dan waktu pelayanan barang. Kemudian dalam akan diberikan beberapa contoh proses pengiriman barang pada kasus yang lain.

3.4 Pengolahan Data dan Analisis Data

Data studi kasus yang diperoleh pada tahap pengumpulan data yang diolah sesuai dengan metode yang digunakan, adapun tahapan pengolahan data antara lain sebagai berikut:

1. Menentukan populasi baru dengan menentukan populasi awal kemudian melakukan evaluasi fitness, seleksi individu, dilakukan reproduksi crossover dan mutasi sehingga dihasilkan populasi baru.
2. Menentukan jarak dan waktu perjalanan dari laundry ke customer dan dari customer ke customer dengan membuat matrik jarak dan waktu.
3. Membuat implementasi program algoritma genetika
4. Kemudian melakukan input data terhadap Program algoritma genetika yang telah dibuat untuk melakukan beberapa kali simulasi untuk mengetahui nilai parameter-parameter algoritma genetika yang paling baik agar mendapatkan hasil yang terbaik.

3.5 Penyelesaian Permasalahan Penentuan Rute Kendaraan Dalam Pemenuhan Permintaan Customer Dengan Kendala *Time Window*.

Proses penentuan rute ini didapat dari biaya minimal perjalanan kunjungan laundry ke seluruh customer dan lama waktu pelayanannya dengan tidak melanggar batasan *time window* pelayanan customer. Jika laundry datang sebelum interval waktu yang diberikan oleh customer maka pihak laundry akan diberikan waktu tunggu kendaraan. Namun apabila laundry datang disaat melebihi interval waktu yang diberikan customer, maka akan diberikan waktu penalty.

3.6 Proses Manualisasi

Di dalam sub bab ini akan dijelaskan secara khusus bagaimana sistem ini bekerja sesuai dengan tinjauan pustaka yang telah dibuat. Salah satu dari tinjauan pustaka tersebut adalah *Time Windows* (VRPTW) menggunakan Algoritma Genetika. Kami akan menjelaskan bagaimana proses manualisasi dari *Time Windows* (VRPTW) menggunakan Algoritma Genetika. Di d bawah ini adalah proses perhitungan manual dari *Time Windows* (VRPTW) menggunakan Algoritma Genetika :

Tabel 3.1. Tabel interval waktu tiap node dan waktu pelayanannya

Node	Buka	Tutup	Layanan (menit)
1	09.00	17.00	35
2	12.00	23.00	40
3	08.00	16.00	55
4	10.00	21.00	50
5	08.00	21.00	30
6	10.00	22.00	60
7	12.00	23.00	45

Pada penelitian ini kami menggunakan 7 node, dan setiap node merepresentasikan rumah customer. Setiap customer memiliki kriteria tersendiri dalam penerimaan pengiriman laundry. Jam buka merupakan waktu dimana customer mulai menerima kiriman laundry, jam tutup

merupakan akhir dari customer menerima kiriman laundry sedangkan layanan merupakan lama waktu pembongkaran laundry.

Tabel 3.2. Tabel Lama Waktu Tempuh Antar Node

Lama Waktu (dalam menit)								
	s	1	2	3	4	5	6	7
s	-	60	180	60	120	60	120	60
1	60	-	60	120	60	120	180	120
2	180	60	-	180	120	60	120	60
3	60	120	180	-	120	60	60	120
4	120	60	120	120	-	120	180	60
5	60	120	60	60	120	-	60	120
6	120	180	120	60	180	60	-	60
7	60	120	60	120	60	120	60	-

Pada tabel 3.2. merupakan lama waktu tempuh dari masing-masing node (customer). Seperti customer, pemilik laundry juga memiliki kriteria tersendiri dalam pengiriman laundry. Pemilik laundry mulai mengirimkan laundry kepada customer mulai dari jam 07.00.

Tabel 3.3. Tabel Inisialisasi Awal dengan Popsiz = 5

Parent ke-	Chromosom						
P1	5	4	3	7	1	6	2
P2	7	4	1	3	2	6	5
P3	1	3	6	7	2	4	5
P4	5	7	1	6	3	2	4
P5	1	6	2	4	7	5	3

Tabel 3.3. menunjukkan parent yang terbentuk pada tahap inisialisasi awal. Inisialisasi awal yang menggunakan popsize=5 akan menghasilkan 5 buah parent. Inisialisasi ini menggunakan permutasi, dimana pada setiap parent, hanya muncul tepat 1 dari masing- masing node. Sehingga pada setiap individu akan memiliki 7 node.

Proses reproduksi menggunakan metode Crossover Permutasi dan Exchange Mutation. Pada proses Crossover akan menghasilkan 2 buah offspring. P1 dan P2 terpilih sebagai parent yang akan di crossover menggunakan metode permutasi. Dari Crossover ini akan menghasilkan offspring yaitu C1

P1	5	4	3	7	1	6	2
P5	1	6	2	4	7	5	3
C1	5	4	3	1	6	2	7

P3 dan P2 terpilih sebagai parent yang akan di crossover menggunakan metode permutasi. Dari Crossover ini akan menghasilkan offspring yaitu C2

P3	1	3	6	7	2	4	5
P2	7	4	1	3	2	6	5
C2	1	3	6	7	4	2	5

P5 dan P4 terpilih sebagai parent yang akan dilakukan mutasi. Mutasi ini menggunakan metode *Exchange Point*. Dimana Exchange Mutation ini merupakan proses mutasi yang dilakukan dengan menukar kromosom satu dengan salah satu kromosom lain. Dari hasil mutasi ini akan dihasilkan 2 buah *offspring*, yaitu C3 dan C4.

P5	1	6	2	4	7	5	3
C3	1	5	2	4	7	6	3
P4	5	7	1	6	3	2	4
C4	5	2	1	6	3	7	4

Di bawah ini merupakan *parent* dan *child* yang terbentuk setelah mengalami proses *crossover* dan *exchange mutation* :

Tabel 3.4. Populasi Baru setelah Proses Reproduksi

Parent ke	Chromosom						
P1	5	4	3	7	1	6	2
P2	7	4	1	3	2	6	5
P3	1	3	6	7	2	4	5
P4	5	7	1	6	3	2	4
P5	1	6	2	4	7	5	3
C1	5	4	3	1	6	2	3
C2	1	3	6	7	4	2	5
C3	1	5	2	4	7	6	3
C4	5	2	1	6	3	7	4

Nilai *fitness* didapatkan dari menjumlah waktu perjalanan yang dilakukan driver laundry pada setiap solusi. Di bawah ini merupakan hasil perhitungan *fitness* dari masing-masing parent :

Tabel 3.4. Hasil perhitungan Fitness yang dihasilkan dari Parent 1

SOLUSI 1						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalty (menit)	Perjalanan
5	08.00	0	08.00	08.30	0	60
4	10.30	0	10.30	11.20	0	60
3	13.20	0	13.20	14.25	0	120
7	16.25	0	16.25	17.10	0	120
1	19.10	0	19.10	19.45	110	120
6	22.45	0	22.45	23.45	45	180
2	01.45	0	01.45	02.24	105	120
					260	780

Tabel 35. Hasil perhitungan Fitness yang dihasilkan dari Parent 2

SOLUSI 2						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
7	08.00	4	12.00	12.45	0	60
4	13.45	0	13.45	14.35	0	60

1	15.35	0	15.35	16.10	0	60
3	18.10	0	18.10	19.05	90	120
2	22.05	0	22.05	22.45	0	180
6	00.45	0	00.45	01.45	165	120
5	02.45	0	02.45	03.15	345	60
					600	660

Tabel 3.6. Hasil perhitungan Fitness yang dihasilkan dari Parent 3

SOLUSI 3						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
1	08.00	1	09.00	09.35	0	12
3	11.35	0	11.35	12.30	0	60
6	13.30	0	13.20	14.30	0	60
7	15.30	0	15.30	16.15	0	60
2	17.15	0	17.15	17.55	0	120
4	19.55	0	19.55	20.45	0	60
5	22.45	0	22.45	23.15	45	120
					45	492

Tabel 3.7. Hasil perhitungan Fitness yang dihasilkan dari Parent 4

SOLUSI 4						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
5	08.00	0	08.00	08.30	0	60
7	10.30	0	10.30	11.15	0	120
1	13.15	0	13.15	13.50	0	120
6	16.50	0	16.50	18.00	0	180
3	19.00	0	19.00	19.55	235	60
2	22.55	0	22.55	23.35	0	180
4	01.35	0	01.35	02.25	275	120
					510	840

Tabel 3.8. Hasil perhitungan Fitness yang dihasilkan dari Parent 5

SOLUSI 5						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
1	08.00	1	09.00	09.35	0	60
6	12.35	0	12.35	13.35	0	180
2	15.35	0	15.35	16.15	0	120
4	18.15	0	18.15	19.05	0	120
7	20.05	0	20.05	20.50	0	60
5	22.50	0	22.50	23.20	110	120
3	24.20	0	24.20	01.15	260	60
					370	720

Tabel 3.9. Hasil perhitungan Fitness yang dihasilkan dari Parent 6

SOLUSI 6						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
5	08.00	0	08.00	08.30	0	60
4	10.30	0	10.30	11.20	0	120
3	13.20	0	13.20	14.15	0	120
1	16.15	0	16.15	16.45	0	120
6	19.45	0	19.45	20.45	0	180
2	22.45	0	22.45	23.25	25	120
3	02.25	0	02.25	03.30	625	180
					650	900

Tabel 3.10. Hasil perhitungan Fitness yang dihasilkan dari Parent 7

SOLUSI 7						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
1	08.00	1	09.00	09.35	0	60
3	11.35	0	11.35	12.30	0	120
6	13.20	0	13.20	14.20	0	60
7	15.20	0	15.20	16.05	0	60
4	17.05	0	17.05	17.55	0	60

2	19.55	0	19.55	20.35	0	120
5	21.35	0	21.35	22.05	35	60
					35	540

Tabel 3.11. Hasil perhitungan Fitness yang dihasilkan dari Parent 8

SOLUSI 8						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
1	08.00	1	09.00	09.35	0	60
5	11.35	0	11.35	12.05	0	120
2	13.05	0	13.05	13.45	0	60
4	15.45	0	15.45	16.35	0	120
7	17.35	0	17.35	18.20	0	120
6	19.20	0	19.20	20.20	0	60
3	21.20	0	21.20	22.15	320	60
					320	600

Tabel 3.12. Hasil perhitungan Fitness yang dihasilkan dari Parent 9

SOLUSI 9						
Node	Sampai	Tunggu	Mulai	Selesai	Pinalti (menit)	Perjalanan
5	08.00	0	08.00	08.30	0	60
2	09.30	2,5	12.00	12.40	0	60
1	13.40	0	13.40	13.15	0	60
6	16.15	0	16.15	17.15	0	180
3	18.15	0	18.15	19.20	0	60
7	22.20	0	22.20	23.05	0	120
4	00.05	0	00.05	00.55	185	60
					185	600

Proses Seleksi menggunakan Elitism. Elitism adalah pemilihan *chromosome* yang memiliki *fitness* yang paling baik (paling besar) di dalam satu populasi. Di bawah ini merupakan hasil seleksi dari satu kali generasi :

Tabel 3.13. Tabel Hasil Seleksi

Solusi ke-i	Pinalti	Perjalanan
3	45	492
7	35	540
8	320	600
9	185	600
2	600	660
5	370	720
1	260	780
4	510	820
6	650	900

BAB IV

ANALISA DA PERANCANGAN

4.1. Spesifikasi Software dan Hardware

Spesifikasi yang kompatibel untuk Optimasi Rute Pengiriman Laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika adalah sebagai berikut :

4.1.1. Spesifikasi Software

4.1.1.1. Manajemen Data

Data yang digunakan di dalam penelitian Optimasi Rute Pengiriman Laundry dengan *Time Windows* (VRPTW) menggunakan Algoritma Genetika adalah data random. Dimana data dibangkitkan secara random pada program.

4.1.1.2. Manajemen Model

Bahasa pemrograman yang digunakan adalah bahasa pemrograman Java. Bahasa pemrograman ini memiliki kelebihan dalam hal multiplatform dan menggunakan konsep OOP (*Object Oriented Programming*). Selain itu juga memiliki library yang lengkap dimana sangat membantu dalam membangun sebuah aplikasi.

Netbeans adalah salah satu aplikasi IDE yang digunakan programmer untuk menulis, meng*compile*, mencari kesalahan, dan menyebarkan program.netbeans ditulis dalam bahasa java namun dapat juga mendukung bahasa pemrograman lain. Fitur yang dimiliki pada Netbeans ini seperti Smart code compilation, menggunakan code generator, error stripe, dan go to commands. Netbeans yang digunakan dalam penelitian ini adalah Netbeans IDE 7.1.

4.1.2. Spesifikasi Hardware

Operating system	: Windows 7 Ultimate 32-bit (6.1, Build 7601)
System model	: ACER 4741
Processor	: Intel Core i ⁵ , 2.53 GHz

Memory : 2048 MB

4.2. Desain Antar Muka

Perancangan tampilan dari sistem Optimasi Rute Pengiriman Laundry dengan *Time Windows* (VRPTW)

Node	Waktu	Tutup	Waktu

Node	Node

Waktu Tempuh dari Node satu menuju Node Lainnya

Banyak Individu

Banyak Iterasi

Set

Hasil fitness

Hasil fitness setelah disorting

Gambar 4.1. Desain Antar Muka

BAB V

IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil. Hasil ini diperoleh dari proses perancangan perangkat lunak yang dibuat. Pembahasan ini terdairi dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi algoritma pada program, implementasi antarmuka dan implementasi metode.

5.1. Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi acuan untuk melakukan implementasi. Hal ini dimaksudkan agar sistem dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan menjadi spesifikasi perangkat keras dan perangkat lunak.

5.1.1. Spesifikasi Perangkat Keras

Pengembangan Sistem Aplikasi Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (VRPTW) Menggunakan Algoritma Genetika dengan spesifikasi perangkat keras seperti yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras SPK Pencairan Kredit

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core (TM) i5-430 M @ 2.26 GHz
Memori (RAM)	2048 mb
Hardisk	500

5.1.2. Spesifikasi Perangkat Lunak

Pengembangan Aplikasi Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (VRPTW) Menggunakan Algoritma Genetika dengan spesifikasi perangkat lunak seperti yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat LunakSPK Pencairan Kredit

Nama	Spesifikasi
Sistem Operasi	Windows 7 Ultimate 32-bit
Bahasa Pemrograman	Java
Tools Pemrograman	Netbeans

5.2. Batasan- Batasan Implementasi

Batasan implementasi adalah batasan proses yang bisa dilakukan sistem sesuai dengan perancangan awal sistem. Batasan implementasi ditampilkan agar penelitian ini memiliki ruang lingkup yang jelas dalam mengimplementasikan sistem. Beberapa batasan dalam mengimplementasikan Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (VRPTW) Menggunakan Algoritma Genetika adalah sebagai berikut :

- Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (VRPTW) Menggunakan Algoritma Genetika dirancang dan dijalankan menggunakan *Netbeans*
- Metode penyelesaian masalah yang digunakan adalah metode Time Window (VRPTW)
- Input yang diterima oleh aplikasi berupa jumlah individu dan jumlah maksimal iterasi
- Output yang dihasilkan adalah hasil fitness dari masing- masing individu yang sudah di sorting dari terkecil ke terbesar menggunakan metode seleksi Elitism.
- Data jarak antarnode dibangkitkan secara random

- Jumlah node diinisialisasikan sebanyak 7

5.3. Implementasi Algoritma

Aplikasi Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (VRPTW) Menggunakan Algoritma Genetika ini mempunyai beberapa proses utama yaitu proses *pembangkitan populasi*, *proses reproduksi* dan *proses seleksi*.

5.3.1. Implementasi Algoritma Proses Pembangkitan Populasi Awal

Proses ini merupakan proses dimana populasi awal dibentuk. Di dalam proses ini, akan terbentuk kromosom yang memiliki sifat permutasi. Jadi di dalam satu kromosom tidak boleh ada nilai yang sama, hanya terdapat tepat satu nilai

```
boolean[] sudahAda = new boolean[x + 1];
Random rnd = new Random();
for (int j = 0; j <= x; j++) {
    sudahAda[j] = false;
}

for (int i = 0; i < x; i++) {
    int tmp;
    do {
        tmp = rnd.nextInt(x) + 1; //1 s/d x
    } while (sudahAda[tmp]);

    kromosom[i] = tmp;
    sudahAda[tmp] = true;
}
```

5.3.2. Implementasi Algoritma Proses Reproduksi

Di dalam aplikasi ini terdapat proses reproduksi, dimana di dalamnya terdapat proses crossover dan mutasi. Proses Crossover menggunakan teknik crossover permutasi sedangkan proses mutasi menggunakan teknik exchange point. Di bawah ini merupakan penggalan source code dari proses Crossover Permutasi :

```
for (int itr=1; itr<=nItr; itr++)
{
    //CROSSOVER
    for (int idCr=1; idCr<=nCr; idCr++)
    {
```

```

        int idChd = (nIdv+idCr)-1; //index kromosom pada
populasi yang akan dijadikan child

        //select parent
        int p1,p2;
        p1 = rnd.nextInt(nIdv); // 1 s/d nIdv
        do {
            p2 = rnd.nextInt(nIdv); // 1 s/d nIdv
        } while (p2==p1);
        str += " Cross"+idCr+": "+(p1+1)+" dengan "+(p2+1);

        //sudahAda: variabel untuk mencegah redundancy pemilihan
gen
        boolean[] sudahAda = new boolean[nGen+1];
        for (int i=0; i<=nGen; i++)
        { sudahAda[i] = false; }

        int cP = nGen / 2; //cP: cut Point
        int idx = 0;
        //copy dari P1 (sebelum cut Point)
        for (int i=0; i<cP; i++)
        {
            int val = pop[p1].kromosom[i];
            pop[idChd].kromosom[idx] = val;
            idx++; sudahAda[val]=true;
        }
        //copy dari P2 (sebelum cut Point)
        for (int i=0; i<cP; i++)
        {
            int val = pop[p2].kromosom[i];
            if (sudahAda[val]==false)
            {
                pop[idChd].kromosom[idx] = val;
                idx++; sudahAda[val]=true;
            }
        }

```

```

    }
    if (idx<nGen)
    {
        //copy dari P1 (setelah cut Point)
        for (int i=cP; i<nGen; i++)
        {
            int val = pop[p1].kromosom[i];
            if (sudahAda[val]==false)
            {
                pop[idChd].kromosom[idx] = val;
                idx++; sudahAda[val]=true;
            }
        }
    }
    if (idx<nGen)
    {
        //copy dari P2 (setelah cut Point)
        for (int i=cP; i<nGen; i++)
        {
            int val = pop[p2].kromosom[i];
            if (sudahAda[val]==false)
            {
                pop[idChd].kromosom[idx] = val;
                idx++; sudahAda[val]=true;
            }
        }
    }
}

```

Di bawah ini merupakan penggalan source code dari proses Mutasi Exchange Point :

```

str = "";
int mP1,mP2,idxM; //mP1,m2: mutation Point 1 & 2, idxM:

```

```

kromosom yang akan dimutasi

    mP1 = rnd.nextInt(nGen); // 0 s/d nGen-1
    do {
        mP2 = rnd.nextInt(nGen); // 0 s/d nGen-1
    } while (mP2==mP1);

    //sudahPilih: variabel untuk mencegah redundancy pemilihan
    kromosom yang akan dimutasi

    boolean[] sudahPilih = new boolean[nIdv+1];
    for (int i=0; i<=nIdv; i++)
        { sudahPilih[i] = false; }

    for (int idMt=1; idMt<=nMt; idMt++)
    {
        //pilih kromosom yang akan dimutasi
        do {
            idxM = rnd.nextInt(nIdv); // 0 s/d nIdv-1
        } while (sudahPilih[idxM]);
        sudahPilih[idxM] = true;

        int idChd = (nIdv+nCr+idMt)-1; //index kromosom pada
        populasi yang akan dijadikan child (hasil mutasi)

        pop[idChd].kromosom[mP1] = pop[idxM].kromosom[mP2];
        pop[idChd].kromosom[mP2] = pop[idxM].kromosom[mP1];

        str += " Mutation"+idMt+": "+(idxM+1)+" pada posisi
        "+(mP1+1)+" & "+(mP2+1);
    }

```

5.3.3. Implementasi Algoritma Proses Penghitungan Nilai Fitness

Pada bagian implementasi perhitungan fitness dilakukan setelah pemilihan chromosome yang akan di mutasi. Perhitungan fitness dilakukan untuk mengetahui nilai hasil fitness yang paling optimal untuk penjadwalan waktu pada studi kasus laundry. Dalam menentukan nilai fitness digunakan metode seleksi elitism yaitu memilih nilai fitness yang mempunyai nilai terkecil dari kumpulan

chromosome yang telah dihitung. Struktur program untuk perhitungan fitness ditunjukkan pada source code :

```

for (int idK=0; idK<(nIdv+nCr+nMt); idK++)
{
    int idA, idB, minAdd, minTotal;
    Date wkt = new
SimpleDateFormat("HH:mm").parse("07:00");
    minTotal = 0; minAdd = 0; idA = 0;
    String wktAkhir = "";
    for (int i=0; i<nGen; i++)
    {
        if (i==0)
        {
            idA = pop[idK].kromosom[i]-1;
            minAdd = costMatrix.awal[idA] +
spot.waktu[idA];
        }
        else if (i>0)
        {
            idB = pop[idK].kromosom[i]-1;
            minAdd = costMatrix.mtr[idA][idB] +
spot.waktu[idB];
            idA = idB;
        }
        minTotal += minAdd;
        wkt.setTime(wkt.getTime()+ minAdd *60000);
        wktAkhir = spot.tutup[idA];
    }
    Date jamAkhir = new
SimpleDateFormat("HH:mm").parse(wktAkhir);
    long selisih = ((wkt.getTime()/60000) -
(jamAkhir.getTime()/60000));
    int penalti = 0;
    if (selisih>0) { penalti = (int) selisih;
minTotal += penalti; }
    pop[idK].fitness = 1/(double) minTotal;
}

```

5.3.4. Implementasi Algoritma Proses Sorting Nilai Fitness

Proses kedua untuk menampilkan populasi setelah perhitungan fitness dilakukan. Hasil populasi dengan fitnessnya akan muncul pada kolom dengan nama tbPop1. Sebelum menampilkan hasil fitness secara keseluruhan, maka dilakukan pengaturan jumlah kolom dan jumlah baris dengan menggunakan method `setColumnCount(nGen+2)` dan `setNumRows(nIdv+nCr+nMt+1)` yang akan digunakan. Untuk mengatur kolom agar rata samping maka menggunakan

DefaultTableCellRenderer. Struktur program untuk menampilkan hasil populasi yang telah dilakukan perhitungan fitnessnya ditunjukkan pada source code di bawah ini :

```

        model3.setColumnCount(nGen+2); //set col count
        model3.setNumRows(nIdv+nCr+nMt+1); //set row count
        DefaultTableCellRenderer centerRenderer = new
DefaultTableCellRenderer();
        centerRenderer.setHorizontalAlignment(SwingConstants.CENTER)
;
        for (int i=0; i<=nGen; i++)
        {

        tbPop1.getColumnModel().getColumn(i).setPreferredWidth(30);

        tbPop1.getColumnModel().getColumn(i).setCellRenderer(centerR
enderer);
        }

        tbPop1.getColumnModel().getColumn(nGen+1).setPreferredWidth(
80);

        tbPop1.getColumnModel().getColumn(nGen+1).setCellRenderer(ce
nterRenderer);

        for (int i=0; i<nGen; i++)
        { model3.setValueAt(i+1, 0, i+1); }
        model3.setValueAt("Fitness", 0, nGen+1);
        for (int i=0; i<nIdv; i++)
        { model3.setValueAt("P"+(i+1), i+1, 0); }
        for (int i=0; i<(nCr+nMt); i++)
        { model3.setValueAt("C"+(i+1), (nIdv+i+1), 0); }

        for (int i=1; i<=nIdv+nCr+nMt; i++)
        {
            for (int j=1; j<=nGen; j++)
            { model3.setValueAt(pop[i-1].kromosom[j-1], i, j);
            }
            model3.setValueAt(String.format("%.5f",      pop[i-
1].fitness), i, nGen+1);
        }
    }

```

Proses pengurutan hasil fitness atau sorting merupakan proses perhitungan yang digunakan untuk mengetahui hasil optimasi dari hasil fitness. Pengurutan dilakukan dari nilai fitness terbesar ke nilai fitness terkecil. Struktur program untuk mengurutkan nilai fitness dari besar ke kecil ditunjukkan pada source code di bawah ini :

```

        Individu tmpIdv = new Individu(nGen);
        for (int i=0; i<((nIdv+nCr+nMt)-1); i++)
            for (int j=i+1; j<(nIdv+nCr+nMt); j++)

```

```

        {
            if (pop[i].fitness < pop[j].fitness)
            {
                tmpIdv = pop[i];
                pop[i] = pop[j];
                pop[j] = tmpIdv;
            }
        }
    }
}

```

Pada proses untuk menampilkan hasil fitness setelah dilakukan pengurutan dari nilai terbesar ke nilai terkecil sama seperti tampilan hasil fitness sebelum diurutkan. Letak label untuk menampilkan hasil pengurutan fitnessnya adalah tbPop2. Struktur program untuk menampilkan nilai fitness dari besar ke kecil ditunjukkan pada source code di bawah ini :

```

model4.setColumnCount(nGen+2); //set col count
model4.setNumRows(nIdv+nCr+nMt+1); //set row count
//set column width and alignment
for (int i=0; i<=nGen; i++)
{
    tbPop2.getColumnModel().getColumn(i).setPreferredWidth(30);

    tbPop2.getColumnModel().getColumn(i).setCellRenderer(centerR
enderer);
}

tbPop2.getColumnModel().getColumn(nGen+1).setPreferredWidth(
80);

tbPop2.getColumnModel().getColumn(nGen+1).setCellRenderer(ce
nterRenderer);

    for (int i=0; i<nGen; i++)
    { model4.setValueAt(i+1, 0, i+1); }
    model4.setValueAt("Fitness", 0, nGen+1);
    for (int i=0; i<(nIdv+nCr+nMt); i++)
    { model4.setValueAt("K"+(i+1), i+1, 0); }

    for (int i=1; i<=nIdv+nCr+nMt; i++)
    {
        for (int j=1; j<=nGen; j++)
        { model4.setValueAt(pop[i-1].kromosom[j-1], i, j);
        }
        model4.setValueAt(String.format("%.5f",      pop[i-
1].fitness), i, nGen+1);
    }
}
} catch (ParseException ex) {

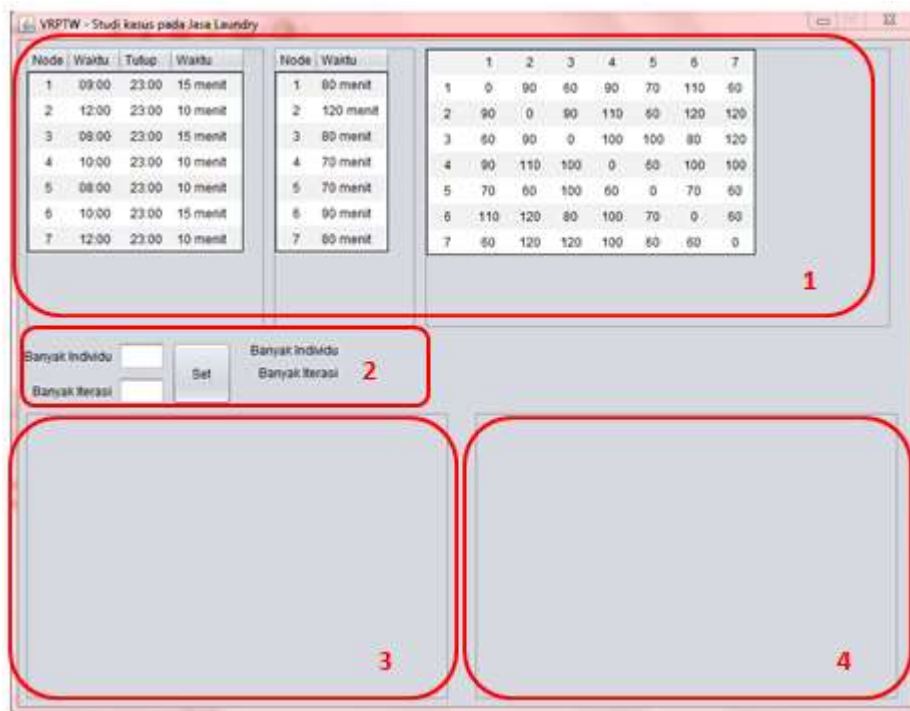
Logger.getLogger(fLaundry.class.getName()).log(Level.SEVERE,
null, ex);
}
}

```

}

5.4. Implementasi Antarmuka

Implementasi antarmuka sistem dibagi menjadi empat bagian, yaitu implementasi tampilan data awal, implementasi inputan data set, implementasi tampilan data hasil fitness, dan implementasi tampilan data fitness yang telah disorting. Implementasi dalam sistem ini diterapkan dalam bahasa pemrograman java.



Gambar 5.1. Implementasi sistem

Keterangan :

- 1 = Implementasi Tampilan Data Awal
- 2 = Implementasi Tampilan Inputan Data Set
- 3 = Implementasi Tampilan Hasil Seleksi Sebelum di Sorting
- 4 = Implementasi Tampilan Hasil Seleksi Setelah di Sorting

5.4.1. Implementasi Tampilan Data Awal

Implementasi tampilan data awal dalam sistem terdiri dari 3 bagian, yaitu data interval waktu buka dan tutup dari tiap node serta waktu pelayanan, data waktu tempuh tiap node dari tempat laundry, dan data waktu tempuh dari node satu menuju node lainnya.

Node	Waktu	Tutup	Waktu
1	09:00	23:00	35 menit
2	12:00	23:00	40 menit
3	08:00	23:00	55 menit
4	10:00	23:00	50 menit
5	08:00	23:00	30 menit
6	10:00	23:00	60 menit
7	12:00	23:00	45 menit

Node	Waktu
1	100 menit
2	100 menit
3	100 menit
4	110 menit
5	100 menit
6	100 menit
7	80 menit

	1	2	3	4	5	6	7
1	0	110	120	120	60	120	70
2	110	0	70	110	120	110	70
3	120	70	0	70	110	60	70
4	120	110	70	0	100	110	60
5	60	120	110	100	0	70	100
6	120	110	60	110	70	0	110
7	70	70	70	60	100	110	0

Gambar 5.2. Tampilan Data Awal

Keterangan :

- 1 = Daftar Node (Customer) yang di dalamnya terdapat waktu buka, tutup dan lama pembongkaran
- 2 = Merupakan lama waktu perjalanan dari node S (node berangkat) ke masing- masing node (Customer) yang terdapat pada daftar
- 3 = Tabel matriks jarak tempuh antarnode dalam menit

5.4.2. Implemetasi Inputan Data Set

Implementasi tampilan inputan data set digunakan untuk insisialisi jumlah individu dan iterasi yang akan di proses. Dari jumlah individu yang dimasukkan nanti akan dijadikan jumlah parent dalam satu populasi. Sedangkan banyak iterasi menunjukkan banyaknya generasi yang ingin diuji,

Gambar 5.3. Tampilan Inputan Data Set

5.4.3. Implementasi Tampilan Hasil Seleksi Sebelum di Sorting

Implementasi tampilan hasil seleksi sebelum di sorting merupakan hasil seleksi dari seluruh iterasi yang telah dilakukan. Hasil fitness di dalam tabel ini belum melalui proses sorting, sehingga belum diketahui solusi terbaik dari pengujian.

	1	2	3	4	5	6	7	Fitness
P1	5	2	2	7	4	5	6	0,00121
P2	4	7	5	7	3	3	6	0,00114
P3	4	2	7	5	1	3	6	0,00110
P4	4	2	7	5	1	3	6	0,00110
P5	7	5	6	3	1	2	4	0,00109
C1	4	2	7	5	1	3	6	0,00110
C2	4	2	7	5	6	1	3	0,00109
C3	2	5	6	4	3	6	5	0,00109
C4	7	5	6	2	3	1	2	0,00106

Gambar 5.4. Tampilan Hasil Seleksi Sebelum di Sorting

5.4.4. Implementasi Tampilan Hasil Seleksi Setelah di Sorting

Implementasi tampilan hasil seleksi sebelum di sorting merupakan hasil seleksi dari seluruh iterasi yang telah dilakukan. Hasil fitness di dalam tabel ini sudah melalui proses sorting, sehingga sudah dapat diketahui solusi terbaik dari pengujian. Di bawah ini merupakan tampilan hasil seleksi setelah di sorting :

	1	2	3	4	5	6	7	Fitness
K1	5	2	2	7	4	5	6	0,00121
K2	4	7	5	7	3	3	6	0,00114
K3	4	2	7	5	1	3	6	0,00110
K4	4	2	7	5	1	3	6	0,00110
K5	4	2	7	5	1	3	6	0,00110
K6	7	5	6	3	1	2	4	0,00109
K7	4	2	7	5	6	1	3	0,00109
K8	2	5	6	4	3	6	5	0,00109
K9	7	5	6	2	3	1	2	0,00106

Gambar 5.5. Tampilan Hasil Seleksi Setelah di Sorting

BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan perancangan dan implementasi Aplikasi Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (Vrptw) Menggunakan Algoritma Genetika, maka didapatkan kesimpulan sebagai berikut:

1. Penentuan jumlah individu dan banyaknya iterasi berpengaruh kepada hasil fitness.
2. Aplikasi Optimasi Rute Pengiriman Laundry Dengan *Time Windows* (Vrptw) Menggunakan Algoritma Genetika telah dibuat sesuai dengan perancangan dan dapat digunakan dalam merekomendasikan rute terdekat dalam pengiriman laundry

6.2. Saran

Saran yang diberikan untuk pengembangan penelitian selanjutnya antara lain:

1. Sebaiknya dilakukan perbaikan pada proses pembangkitan individu karena masih terdapat gen yang sama dalam satu kromosom.
2. Sebaiknya data customer (jumlah node) ditambah untuk mengetahui hasil seleksi yang lebih optimal.

DAFTAR PUSTAKA

[AST-08]	Astelaria, Clarista. 2008. Jurnal Penentuan Rute. http://lontar.ui.ac.id/file?file=digital/116807-T%2024624-Penentuan%20rute-Tinjauan%20literatur.pdf , diakses tanggal 18 Desember 2013
[DON-05]	Donald, H U N. 2005. Studi tentang vehicle routing problem with time windows (VRPTW) dengan menggunakan metode simulated annealing. http://www.citeulike.org/user/puslit/article/4856002 , diakses tanggal 18 Desember 2013
[LUK-11]	Lukitasari, Winda 2011. Capacitated Vehicle Routing Problem Time Windows(CVRPTW) Dengan Menggunakan Algoritma Improved Ant Colony System (IACS) Dan Algoritma Simulated Annealing (SA). http://digilib.itelkom.ac.id/index.php?option=com_repository&Itemid=34&task=detail&nim=113060022 , diakses tanggal 18 Desember 2013