

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

**ANALISIS RUTE PENDISTRIBUSIAN DENGAN MENGGUNAKAN METODE
GENETIC ALGORITHM PERSOALAN THE VEHICLE ROUTING PROBLEM
WITH TIME WINDOWS (VRPTW)
(STUDI KASUS KORAN HARIAN PAGI TRIBUN JABAR)**



**Disusun oleh:
Kelompok E Kelas C**

Diah Anggraeni Pitaloka (125150209111003)

Kiki Aprilia Puspitaningrum (125150209111001)

Ika Retno Putri (125150209111006)

Dody Yudha Prawira (125150209111022)

Dosen Pengajar: Wayan Firdaus Mahmudy, Ph.D.

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

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya kepada seluruh umat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir Mata Kuliah Sistem Pendukung Keputusan yang berjudul **“ANALISIS RUTE PENDISTRIBUSIAN DENGAN MENGGUNAKAN METODE GENETIC ALGORITM PERSOALAN THE VEHICLE ROUTING PROBLEM WITH TIME WINDOWS (VRPTW) PADA STUDI KASUS KORAN HARIAN PAGI TRIBUN JABAR”** dengan baik. Sholawat serta salam kami haturkan kepada junjungan kami Nabi Muhammad SAW, semoga kami selalu istiqomah dalam meneladaninya. Tugas Akhir ini merupakan syarat penilaian akhir yang harus ditempuh oleh mahasiswa sebagai pengaplikasian dari mata kuliah algoritma evolusi yang telah di pada semester ini.

Tidak terlupakan penulis juga mengucapkan terimakasih yang sebesar besarnya kepada pihak-pihak yang membantu terselesaikannya laporan Tugas Akhir ini , ucapan tersebut kami berikan kepada :

1. Syukur Alhamdulillah kepada Allah SWT, atas syukur dan nikmat kesehatan yang telah diberikan.
2. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., selaku dosen mata kuliah Algoritma Evolusi.
3. Seluruh Anggota Kelompok E yaitu Diah Anggraeni Pitaloka, Kiki Aprilia Puspitaningrum, Ika Retno Putri, Dody Yudha Prawira yang telah bekerja sama dengan baik dalam penyelesaian tugas akhir mata kuliah ini.
4. Teman-teman Kelas C mata kuliah Algoritma Evolusi, terima kasih atas dukungannya dan berbagai pihak yang tidak dapat disebut satu persatu yang telah memberikan bantuan dan dukungannya.

Penulis yakin bahwa Tugas Akhir ini masih sangat jauh dari sempurna. Harapannya Tugas Akhir ini dapat memberikan manfaat bagi penulis dan para pembaca.

Malang, 11 Januari 2014

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1 Travelling Salesman Problem (TSP).....	5
2.2 Metode Genetic Algorithms (GA).....	6
2.3 Vehicle Routing Problem (VRP).....	6
2.4 Vehicle Routing Problem with Time Windows (VRPTW).....	7
2.5 Flowchart.....	7
2.6 Bahasa Pemrograman C#	10
BAB III PERANCANGAN SISTEM	11
3.1 Analisis Kebutuhan Sistem	11
3.1.1 Analisis User	11
3.1.2 Analisis Data	11
3.1.3 Analisis Kebutuhan Fungsional.....	11
3.1.4 Analisis Kebutuhan Non-Fungsional	12
3.2 Deskripsi Sistem.....	12
3.3 Flowchart.....	14
3.3 Desain Interface	16
3.4 Spesifikasi Sistem	17
3.4.1 Spesifikasi Perangkat Keras	17
3.4.2 Spesifikasi Perangkat Lunak	17
BAB IV IMPLEMENTASI.....	19

4.1 Tahapan Metode Genetic Algoritma persoalan VRPTW (Vehicle Routing Problem with Time Windows)	19
4.2 Implementasi Algoritma.....	24
4.2.1 Implementasi Algoritma Proses Load Data.....	24
4.2.2 Implementasi Algoritma Proses Pemilihan Rute.....	26
4.3 Pembuatan User Interface	31
BAB V PENGUJIAN.....	33
5.1 Hasil Pengujian	33
5.2 Analisa Pengujian.....	34
BAB VI PENUTUP	36
6.1 Kesimpulan	36
6.2 Saran.....	36
DAFTAR PUSTAKA	37

DAFTAR GAMBAR

Gambar 2.1 Diagram Alur (<i>Flowchart</i>)	8
Gambar 3.1 Alur Diagram Flowchart.....	14
Gambar 3.2 Alur Diagram Flowchart Genetic Algoritma.....	15
Gambar 3.3 Desain Data Uji	16
Gambar 3.4 Desain Form Hasil.....	17
Gambar 4.1 Implementasi Algoritma Load Data	25
Gambar 4.2 Implementasi Algoritma Perhitungan Rute	26
Gambar 4.3 Form Data Uji.....	32
Gambar 4.4 Form Hasil.....	32
Gambar 5.1 Hasil Load Data Uji.....	33
Gambar 5.2 Hasil Data Uji.....	33

DAFTAR TABEL

Tabel 2.1 Simbol-Simbol Flowchart Standart.....	9
Tabel 3.1 Spesifikasi Perangkat Keras Komputer.....	17
Tabel 3.2 Spesifikasi Perangkat Lunak	18
Tabel 4.1 Jadwal Tiap Agen.....	19
Tabel 4.2 Waktu Tempuh.....	19
Tabel 4.3 Solusi.....	20
Tabel 4.4 Inisialisasi Chromosom	21
Tabel 4.5 Nilai <i>offspring</i>	22
Tabel 4.6 Individu hasil proses evaluasi	23
Tabel 5.1 Hasil Pengujian	34

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kegiatan pendistribusian adalah proses penyaluran barang kepada pihak tertentu. Pendistribusian produk yang baik adalah dapat dilakukan secara efisien dengan memanfaatkan waktu yang tepat agar proses pendistribusian sampai kepada pelanggan sesuai dengan waktu yang ditentukan. Hal ini merupakan suatu kunci suatu perusahaan dalam memberikan pelayanan yang maksimal kepada pelanggan, sehingga dapat bersaing dengan para kompetitor yang ada. Hambatan yang sering muncul dalam pendistribusian suatu barang adalah masalah transportasi dan keterbatasan waktu yang tersedia. Masalah transportasi ini sering terjadi khususnya di kota besar yang memiliki jalan yang sangat banyak. Untuk mengatasi permasalahan ini dibutuhkan suatu penentuan rute yang efektif sehingga dapat meminimalkan jarak tempuh dari tempat asal ke tempat tujuan agar permintaan konsumen dapat dilayani dengan baik.

Harian pagi tribun jabar adalah salah satu koran yang terbit di Kota Bandung yang akan didistribusikan ke wilayah Jawa Barat. Dalam penelitian ini hanya dibatasi untuk pengiriman agen distributor untuk wilayah Bandung. Proses pendistribusian ini dilakukan agar dapat meminimalkan biaya perjalanan dan melayani permintaan kebutuhan koran di Kota Bandung. Permintaan banyaknya jumlah koran dari setiap agen berbeda, sesuai dengan kebutuhan dan banyaknya permintaan dari para konsumen. Setiap agen memiliki jadwal pengiriman barang yang berbeda-beda, dan tidak menerima pengiriman barang diluar jam yang telah dijadwalkan sebelumnya. Selain itu kapasitas pengangkutan dalam pendistribusian koran juga terbatas. Jarak tempuh dan biaya yang dibutuhkan dalam pendistribusian barang akan meningkat seiring dengan banyaknya permintaan dan kapasitas angkut yang terbatas dan jadwal penerimaan barang yang berbeda memungkinkan rute perjalanan dilewati lebih dari sekali. Apabila kendaraan pengirim barang sampai ditempat lebih dari waktu pelayanan maka tidak akan dilayani. Namun jika kendaraan pengiriman tiba sebelum waktu pelayanan, maka kendaraan harus menunggu sampai waktu yang telah ditetapkan. Dari permasalahan tersebut dibutuhkan solusi agar setiap agen dapat memberikan pelayanan dalam distribusi koran dengan meminimalkan biaya dan waktu transportasi serta menemukan rute yang efektif untuk sampai ke tempat tujuan.

Banyak metode yang dapat digunakan untuk meminimalkan jarak tempuh dari semua tujuan. Salah satu metode yang dapat digunakan sebagai solusi dari permasalahan ini adalah *Vehicle Routing Problem With Time Windows* (VRPTW) dengan *Genetic Algorithm*. Metode ini adalah metode yang menggunakan algoritma untuk menentukan titik tujuan tertentu dari tempat yang paling dekat. VRPTW atau *Vehicle Routing Problem With Time Windows* adalah sebuah perencanaan suatu kendaraan yang dari suatu tempat ke tempat tujuan dalam kurung waktu tertentu dan berbeda-beda tiap tujuan. Dari permasalahan diatas diharapkan dengan menggunakan metode ini dapat memberikan solusi untuk pendistribusian koran khususnya harian tribun jabar di Kota Bandung agar diperoleh rute dan biaya minimum dengan kapastias pengangkutan yang terbatas serta jadwal yang telah ditentukan tiap agen.

1.2 Perumusan Masalah

Berdasarkan uraian diatas, perumusan yang dapat di bahas dalam pengembangan aplikasi ini adalah sebagai berikut.

1. Bagaimana membuat aplikasi *Vehicle Routing Problem With Time Windows* (VRPTW) agar dapat memberikan solusi dalam pendistribusian koran Harian Pagi Tribun Jabar untuk tujuan tertentu berdasarkan jadwal yang ada?
2. Bagaimana sistem yang dibuat tersebut dapat memberikan informasi pengiriman koran pada setiap tempat sehingga dapat meminimalkan waktu pengiriman dan dapat memberikan pelayan dengan baik?

1.3 Batasan Masalah

Untuk menghindari penyimpangan dalam perancangan sistem ini sehingga dapat mencapai tujuan yang diharapkan, maka penulis membatasi permasalahan yang akan dibahas adalah sebagai berikut ini.

1. Aplikasi ini dibangun hanya terbatas memberikan hasil perhitungan dari solusi yang ada sehingga dapat melayani pendistribusian koran harian pagi tribun Jabar berdasarkan waktu tertentu.
2. Pengolahan data yang akan diproses adalah berupa data yang langsung diambil dari file excel.
3. Perangkat Lunak yang dikembangkan ini berbasis dektop sehingga dapat digunakan oleh para distributor koran di berbagai wilayah di Indonesia.

1.4 Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah sebagai berikut ini.

1. Membuat aplikasi *Vehicle Routing Problem With Time Windows* (VRPTW) agar dapat memberikan solusi dalam pendistribusian koran Harian Pagi Tribun Jabar untuk tujuan tertentu berdasarkan jadwal yang ada.
2. Dari aplikasi yang dikembangkan dapat memberikan informasi pengiriman koran pada setiap tempat sehingga dapat meminimalkan waktu pengiriman dan dapat memberikan pelayanan dengan baik.

1.5 Metodologi Penelitian

Dalam penulisan Tugas Akhir ini penulis menggunakan beberapa metode dalam melakukan penelitian adalah sebagai berikut ini.

1. Studi Lapangan

Studi lapangan dilakukan dengan melakukan penelitian secara langsung pada distributor koran harian pagi tribun Jabar di Kota Bandung serta melakukan proses pengumpulan data yang diperlukan.

2. Studi Pustaka

Studi pustaka dilakukan dengan cara mempelajari referensi, buku dan literatur yang sesuai dengan objek penelitian.

3. Interview

Interview dilakukan dengan melakukan wawancara secara langsung kepada pihak terkait untuk memperoleh data yang lebih akurat.

4. Observasi

Observasi dilakukan dengan mengamati data secara langsung yang diperoleh pada distributor koran harian pagi tribun Jabar.

1.6 Sistematika Penulisan

Pada penulisan Tugas akhir ini terdiri dari beberapa bab dan sub bagian. Dengan menggunakan sistematika penulisan diharapkan agar memudahkan dalam pembahasan masing-masing bab sesuai dengan rumusan masalah yang ada. Adapun sistematika penulisan yang digunakan adalah sebagai berikut ini.

BAB 1 : PENDAHULUAN

Pada bab ini menjelaskan secara garis besar tentang permasalahan yang akan dibahas, mencakup tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

BAB 2: LANDASAN TEORI

Dalam bab ini penulis menjelaskan teori-teori yang akan dibahas untuk menyelesaikan permasalahan, misalnya pengertian *Traveling Salesman Problem* (TSP), Metode *Genetic Algorithm* , *Vehicle Routing Problem* (VRP), *Vehicle Routing Problem Time Windows* (VRPTW), penjelasan tentang *Flowchart*, *Data Flow Diagram* (DFD), dan penjelasan singkat tentang pemrograman C Sharp (C#).

BAB 3: RANCANGAN SISTEM

Bab ini menguraikan tentang gambaran rancangan sistem yang akan dibuat, meliputi analisa kebutuhan sistem, deskripsi sistem, mind mapping sistem, perancangan interface sistem, desain sistem, dan perhitungan manual.

BAB 4: IMPLEMENTASI

Bab ini menguraikan tentang spesifikasi kebutuhan sistem, data awal sebelum diproses dan menjelaskan tentang proses perhitungan VRPTW pada aplikasi yang dikembangkan.

BAB 5: PENGUJIAN DAN HASIL

Bab ini menguraikan tentang proses pengujian pada aplikasi yang telah dikembangkan dengan data yang ada sehingga dapat menghasilkan output berupa hasil dari proses *Vehicle Routing Problem With Time Windows* (VRPTW) dalam studi kasus pendistribusian koran harian pagi.

BAB 6: KESIMPULAN

Bab ini menguraikan tentang kesimpulan pada uraian tugas akhir ini, sehingga dari kesimpulan tersebut penulis mencoba untuk memberikan saran sehingga dapat menyempurnakan pengembangan sistem yang akan datang agar lebih baik.

BAB II

LANDASAN TEORI

2.1 Travelling Salesman Problem (TSP)

Travelling Salesman Problem (TSP) merupakan suatu permasalahan yang sering dihadapi dalam kehidupan sehari-hari khususnya seorang *salesman* dalam mencari rute atau jarak terpendek dengan biaya paling murah. TSP adalah masalah untuk menentukan urutan dari sejumlah kota yang harus dilalui oleh *salesman*, setiap kota hanya boleh dilalui satu kali dalam perjalanannya, dan perjalanan tersebut harus berakhir pada kota keberangkatannya dimana *salesman* tersebut memulai perjalanannya, dengan jarak antara setiap kota satu dengan kota lainnya sudah diketahui. *Salesman* tersebut harus meminimalkan pengeluaran biaya, dan jarak yang harus ditempuh untuk perjalanannya tersebut. Jika terdapat lebih dari 1 *salesman* maka disebut *multi Travelling Salesman Problem* (m-TSP) selain itu juga ada TSP dengan tambahan waktu disetiap kota yang disebut dengan *Traveling Salesman Problem with Time Windows*. TSP diformulasikan sebagai masalah minimasi biaya perjalanan, rumus matematisnya sebagai berikut:

$$Z = \min \{ \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} \} \quad 2.1$$

Dengan kendalanya (constraint):

$$\sum_{i=1}^n x_{ij} = 1, \text{ untuk } j = 1, 2, 3, \dots, n - 1 \quad 2.2$$

$$\sum_{j=1}^n x_{ij} = 1, \text{ untuk } i = 1, 2, 3, \dots, n - 1 \quad 2.3$$

Dimana:

n = banyak kota (disebut simpul/node)

$x_{ij} = 1$ jika ada perjalanan *salesman* dari simpul i ke j jika tidak ada perjalanan maka nilainya 0

c_{ij} = biaya atau jarak (tergantung tujuan minimasi dari simpul i ke j)

Untuk persamaan 2.2 dan 2.3 untuk menjamin bahwa *salesman* mengunjungi sekali setiap simpul.

2.2 Metode Genetic Algorithms (GA)

Metode *Genetic Algorithms* (GA) atau bisa juga disebut dengan metode *Cheapest Insertion Heuristics* (CIH) dimana proses penyisipan yang menyebabkan jarak terendah dalam keseluruhan tur diantara semua simpul yang dimasukkan tidak terlalu jauh. Dibawah ini adalah urutan algoritma GA atau CIH yaitu:

1. Memilih lokasi terdekat untuk dikunjungi terlebih dahulu atau disebut juga dengan proses pengelompokan jarak lokasi terpendek.
2. Dari lokasi yang pertama terpilih lanjutkan ke lokasi lain yang terdekat dengan pertimbangan kapasitas kendaraan bisa disebut dengan pembentukan rute terpendek.
3. Jika semua lokasi belum tercantum pada salah satu rute maka ulangi langkah pertama, perulangan akan berhenti hanya jika semua lokasi telah masuk dalam rute atau proses pengurutan rute.

2.3 Vehicle Routing Problem (VRP)

VRP pertama kali dikenalkan oleh Damtzig dan Ramser pada tahun 1959. VRP ini merupakan perkembangan dari TSP. VRP merupakan permasalahan yang berkaitan dengan pencarian rute yang optimal untuk kendaraan yang melayani beberapa pelanggan. Pada kenyataannya VRP banyak digunakan pada masalah distribusi logistic. Setiap kendaraan memiliki kapasitas angkut, dan setiap pelanggan memiliki *demand*. Tiap pelanggan dikunjungi tepat satu kali dan total *demand* tiap rute tidak boleh melebihi kapasitas angkut kendaraan. Dalam VRP sendiri dikenal pula istilah *depot*, dimana tiap kendaraan harus berangkat dan kembali ke *depot* itu. Hal tersebutlah yang menyebabkan VRP sering disebut sebagai permasalahan m-TSP. Sehingga VRP juga disebut sebagai sebuah problem yang terletak pada irisan dua masalah, yaitu TSP dan BPP. Dimana kedua masalah tersebut termasuk dalam kategori *NP-Hard Problem*, yang berarti waktu komputasi yang digunakan akan semakin sulit dan banyak seiring dengan meningkatnya ruang lingkup masalah. VRP bertujuan untuk menentukan sejumlah rute untuk melakukan pengiriman pada setiap lokasi dengan ketentuan setiap rute berawal dan berakhir di depot, setiap lokasi atau node dikunjungi satu kali dengan satu kendaraan, jumlah permintaan tiap rute tidak melebihi kapasitas kendaraan, dan meminimumkan biaya perjalanan. secara singkatnya tujuan yang ingin dicapai adalah meminimalkan total jarak tempuh dan meminimalkan jumlah kendaraan yang digunakan.

Adapun beberapa jenis VRP diantaranya adalah:

1. Capacitated VRP (CVRP)

Faktor : setiap kendaraan mempunyai kapasitas yang terbatas.

2. VRP With Time Windows (VRPTW)

Faktor : pelanggan harus dilayani dengan waktu tertentu

3. Multiple Depot VRP (MDVRP)

Faktor : distributor memiliki banyak depot

4. VRP With Pick-Up and Delivering (VRPPD)

Faktor : pelanggan diperbolehkan mengembalikan barang ke depot asal

5. Split Delivery VRP (SDVRP)

Faktor : pelanggan dilayani dengan kendaraan berbeda

6. Stochastic VRP (SVRP)

Faktor : munculnya *random values* (seperti jumlah pelanggan, jumlah permintaan, waktu perjalanan atau waktu pelayanan)

7. Periodic VRP

Faktor : pengantaran hanya dilakukan di hari tertentu

2.4 Vehicle Routing Problem with Time Windows (VRPTW)

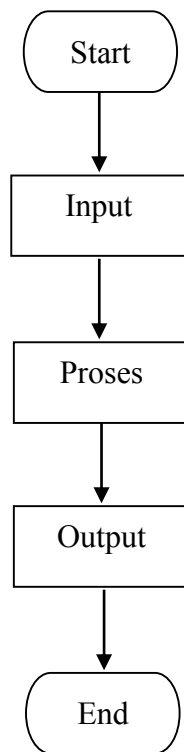
VRPTW merupakan salah satu variasi dari VRP dimana pada satu node, kota, lokasi, atau konsumen hanya dapat dilayani setelah waktu awal yang ditentukan dan tidak dapat dilayani lagi setelah waktu akhir yang ditentukan. Dengan kata lain bila vehicle datang pada suatu node sebelum waktu awal yang ditentukan maka vehicle harus menunggu sampai waktu awal yang ditentukan, namun jika vehicle datang setelah waktu akhir yang ditentukan maka vehicle tersebut tidak diperbolehkan untuk melayani node tersebut. interval waktu antara waktu awal dan waktu akhir inilah yang disebut dengan “Time Windows”. VRPTW ini terbagi menjadi dua kasus yaitu hard time windows dan soft time windows. Untuk kasus hard time windows pengiriman akan ditolak apabila tidak sesuai dengan waktu yang telah ditentukan oleh node, sedangkan kasus soft time windows node akan menerima pengiriman walaupun tidak sesuai dengan waktu yang ditentukan namun diberikan pinalti atau biaya tambahan atas keterlambatannya.

2.5 Flowchart

Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan prosedur-prosedur dari suatu sistem atau program. Flowchart dibuat untuk membantu analisis dan

programmer dalam memecahkan masalah yang perlu dipelajari dan dievaluasi lebih lanjut kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Adapun pedoman-pedoman dalam membuat flowchart yang harus diperhatikan yaitu:





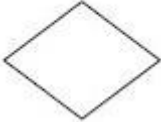



1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan.
2. Harus berhati-hati dalam menggambar dan mendefinisikannya, agar pembaca mudah dalam memahaminya.
3. Kapan aktifitas dimulai dan berakhir harus jelas.
4. Pada setiap langkah dari aktifitas harus diuraikan dengan deskripsi kata kerja.
5. Setiap langkah dari aktifitas harus berada pada urutan yang benar.
6. Kaidah secara garis besar terdiri dari 3 bagian utama yaitu input, proses pengolahan, dan output. Seperti pada gambar dibawah ini.



Gambar 2.1 Diagram Alur (*Flowchart*)

7. Menggunakan simbol-simbol flowchart yang standart dikeluarkan oleh ANSI dan ISO.

Tabel 2.1 Simbol-Simbol Flowchart Standart

Simbol	Arti
<p>Input / Output</p> 	Menunjukkan input data dan output data yang diproses
<p>Proses</p> 	Menunjukkan operasi atau kegiatan
<p>Penghubung</p> 	Keluar dan masuk dari bagian lain halaman yang sama
<p>Anak Panah</p> 	Menunjukkan alur kerja
<p>Keputusan</p> 	Keputusan dalam operasi atau kegiatan
<p>Terminal Points</p> 	Awal dan akhir flowchart
<p>Dokumen</p> 	Dokumen yang dihasilkan
<p>Store data</p> 	Store database atau penampung data

2.6 Bahasa Pemrograman C#

Bahasa Pemrograman C# adalah sebuah bahasa yang dikembangkan oleh Microsoft Corporation sebagai alat bantu pemrograman (*Rapid Application Development Tool*) yang telah dimasukkan kedalam produk Microsoft Visual Studio. Bahasa Pemrograman ini digunakan untuk membuat program berbasis grafis karena bersifat OOP (Object Oriented Programing) sehingga aplikasi yang dibuat lebih dinamis. Bahasa C# merupakan bagian dari inisiatif kerangka .NET Framework yang dibuat berbasiskan bahasa C++ dan dipengaruhi oleh fitur-fitur yang ada pada bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain. C# juga dapat digunakan untuk mengembangkan aplikasi seperti mobile application, aplikasi berbasis web, serta aplikasi-aplikasi berskala besar. Secara umum bahasa C# memiliki karakteristik seperti tertulis dibawah ini.

1. Tidak ada alokasi memori secara manual menggunakan pointer yang mirip dengan bahasa pemrograman Java.
2. Manajemen memori otomatis menggunakan salah satu fiturnya yang dinamakan garbage collection yang mirip dengan bahasa pemrograman Java.
3. Mendukung konstruksi kelas, antarmuka, struktur dan enumerasi seperti bahasa-bahasa pemrograman berorientasi objek lainnya misalnya C++ atau Java.
4. Mendukung pemrograman berbasis atribut
5. Mendukung LINQ (Language Integreted Query) yang memungkinkan aplikasi yang ditulis menggunakan bahasa pemrograman C# mampu berinteraksi dengan berbagai jenis format data dimana aplikasi dengan bahasa C# dapat mengakses sistem basis data relational (RDBMS – *Relational Database Management System*).
6. Mendukung tipe data dan kelas generic mirip bahasa C++ atau Java.
7. Mendukung operator delegasi ($=>$).

BAB III

PERANCANGAN SISTEM

3.1 Analisis Kebutuhan Sistem

Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat di dalamnya. Berikut merupakan penjelasan analisis user, analisis data serta analisis kebutuhan yang digunakan dalam sistem ini.

3.1.1 Analisis User

Pada tahapan ini dilakukan analisis user yang akan menggunakan sistem ini, dimana user itu sendiri adalah agen distribusi koran Harian Pagi Tribun Jabar. Para agen distribusi koran disini akan bertindak sebagai aktor utama yang memiliki hak untuk melakukan semua proses pengelolaan data yang ada pada sistem ini. Pengolahan data yang dilakukan berupa data tempat tujuan distribusi, data jadwal buka dan tutup toko serta data tempuh perjalanan serta jarak tempuh.

3.1.2 Analisis Data

Data yang diperlukan dalam perancangan aplikasi VRPTW pada studi kasus pendistribusian Koran Harian Pagi Tribun Jabar berdasarkan paper yang dianalisis adalah sebagai berikut.

- a. Data Jadwal Harian Agen
- b. Data Waktu Tempuh
- c. Data Jarak Tempuh

3.1.3 Analisis Kebutuhan Fungsional

Pada tahapan ini dilakukan analisis kebutuhan fungsional dari sistem, sehingga dapat diuraikan sebagai berikut :

- a. Membuat aplikasi *Vehicle Routing Problem With Time Windows* (VRPTW) pendistribusian koran pagi.
- b. Membuat aplikasi VRPTW yang dapat digunakan oleh agen distribusi koran bukan hanya pada Koran Harian Pagi Tribun Jabar, namun juga dapat digunakan oleh para agen koran lainnya.
- c. Membuat aplikasi VRPTW yang dapat membantu setiap agen untuk memberikan pelayanan dalam distribusi koran dengan meminimalkan biaya dan waktu transportasi serta menemukan rute yang efektif untuk sampai ke tempat tujuan.

- d. Membuat aplikasi VRPTW yang dapat memberikan solusi yang berdasarkan data yang ada pada sistem.

3.1.4 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional yang dilakukan sistem dapat diuraikan sebagai berikut :

- a. Aplikasi ini harus userfriendly atau dapat digunakan dengan mudah oleh user.
- b. Aplikasi ini harus dapat beroperasi secara terus – menerus.
- c. Fungsi-fungsi semua operasi yang ada di dalam aplikasi ini harus dapat berjalan dengan baik, terutama pada pengolahan datanya.

3.2 Deskripsi Sistem

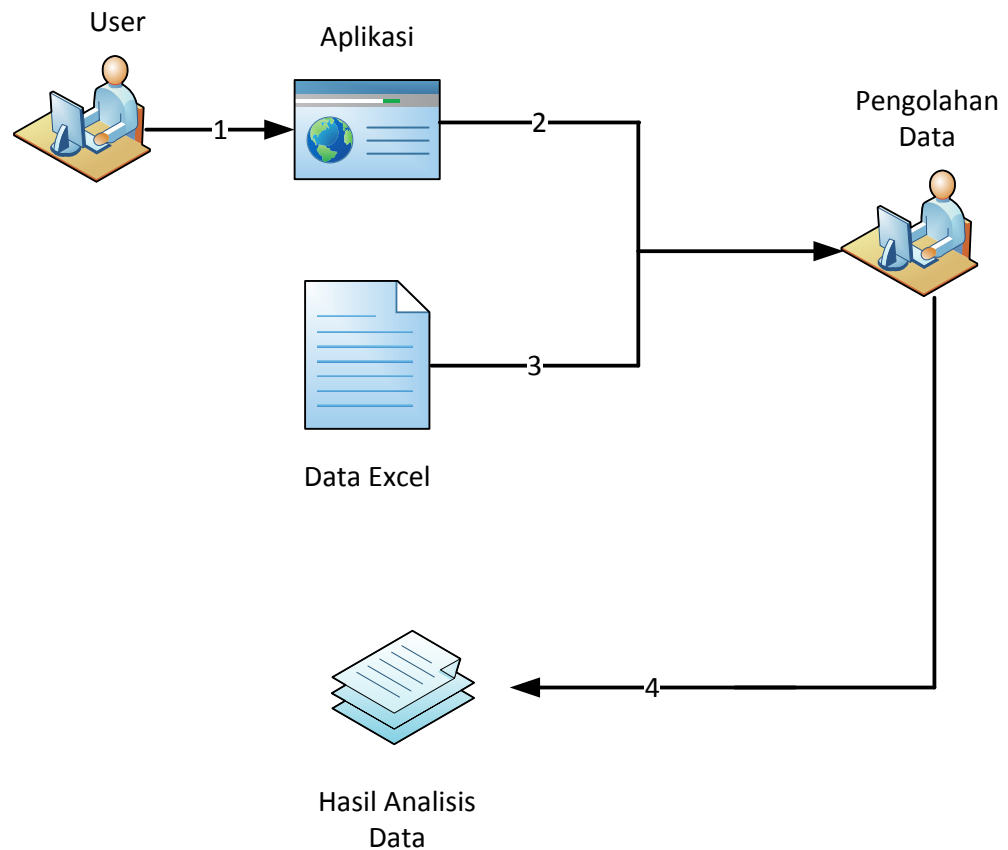
Aplikasi ini merupakan aplikasi untuk menganalisis rute pendistribusian dengan menggunakan metode *Genetic Algorithm* dengan tujuan agar setiap agen dapat memberikan pelayanan dalam distribusi koran dengan meminimalkan biaya dan waktu transportasi serta menemukan rute yang efektif untuk sampai ke tempat tujuan. Pada aplikasi ini tidak menggunakan database, akan tetapi menggunakan sebuah inputan data yaitu berupa file excel yang sudah dibuat sebelumnya dimana di dalamnya sudah terdapat beberapa perhitungan.

Pada step pertama aplikasi ini akan menampilkan data yaitu jadwal harian agen dan juga waktu tempuh. Kemudian selanjutnya dilakukan perhitungan dengan menginputkan beberapa nilai data diantaranya *Pop size*, *Crossover Rate*, *Mutation Rate*, dan *Maximum Generasi*. Dari perhitungan yang dilakukan oleh sistem tersebut nantinya akan diperoleh hasil yaitu berupa generasi terbaik yang didapatkan, lalu individu terbaik / kromosom, besaran pinalti, nilai fitness dan yang terakhir adalah nilai rata-rata.

Dalam perhitungan ini dilakuakn dengan proses iterasi sampai menemukan generasi terbaik melalui data yang telah diinputkan sebelumnya. Dari proses iterasi tersebut dihasilkan sebuah kromosom yang nantinya dapat digunakan untuk proses pendistribusian Koran Harian Tribun Jabar. Kromosom yang diperoleh tersebut diinisialisasikan sebagai tujuan agen pendistribusian koran dan waktu tempuh yang dipelukan agar proses pengirimam koran tepat sesuai jadwal dari masing-masing agen tujuan.

3.2.1 Workflow

Workflow (Alur Kerja) dari “Aplikasi VRPTW Dengan Menggunakan Metode Genetic Algoritm Pada Studi Kasus Koran Harian Pagi Tribun Jabar” dapat digambarkan sebagai berikut :



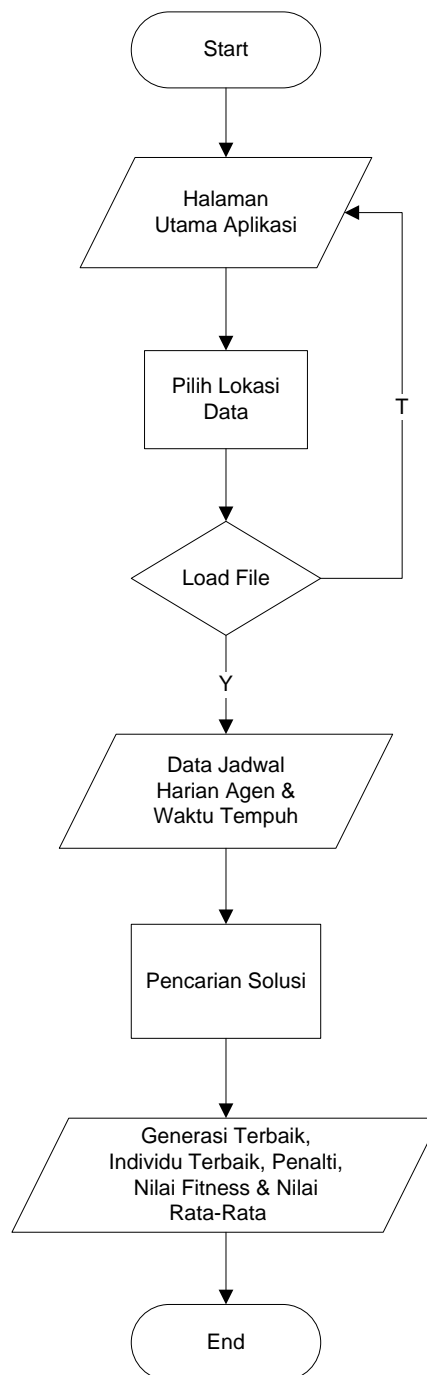
Gambar 3.1 Alur Kerja Aplikasi

Penjelasan Alur Kerja pada gambar diatas adalah sebagai berikut:

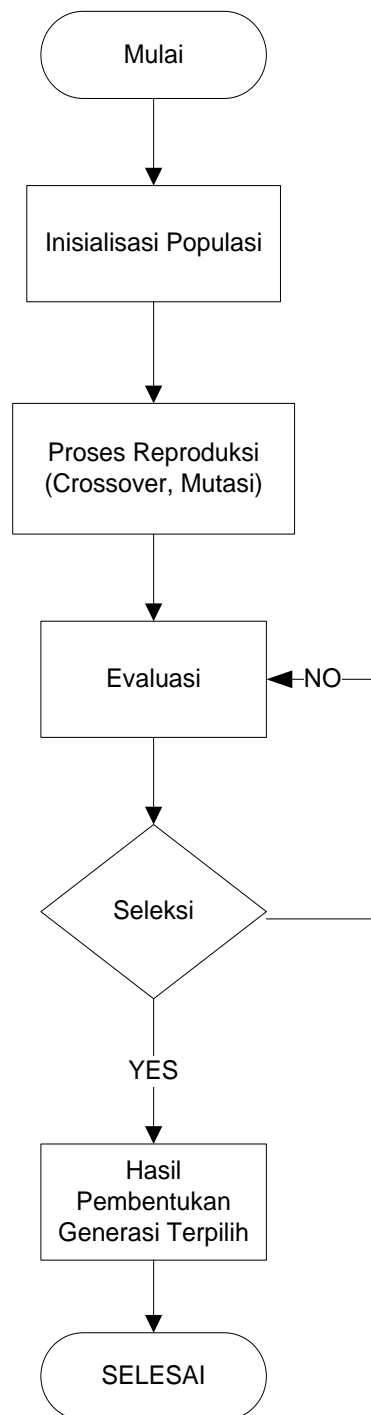
1. User mengakses aplikasi VRPTW.
2. Kemudian user melakukan pengolahan data yang terdapat dalam aplikasi tersebut.
3. Data yang akan diolah disimpan dalam bentuk file excel, yang kemudian dijadikan sebagai inputan sistem, dengan melakukan load data.
4. User melakukan pengolahan data terhadap aplikasi, kemudian dari pengolahan data tersebut diperoleh suatu hasil yaitu hasil analisis data yang akan ditampilkan didalam aplikasi tersebut.
5. Data yang dihasilkan berupa data hasil penjadwalan pendistribusian koran pada beberapa agen dengan memperhitungkan jadwal dan waktu tempuh.

3.3 Flowchart

Flowchart untuk “Aplikasi VRPTW Dengan Menggunakan Metode Genetic Algoritma Pada Studi Kasus Koran Harian Pagi Tribun Jabar” dapat digambarkan sebagai berikut :



Gambar 3.1 Alur Diagram Flowchart



Gambar 3.2 Alur Diagram Flowchart Genetic Algoritma

Berikut merupakan perencanaan *flowchart* umum dari aplikasi vrptw, pertama *user* akan masuk ke dalam halaman utama aplikasi, setelah itu *user* diharuskan untuk melakukan pemilihan file terlebih dahulu sebagai data inputan di dalam aplikasi. File yang dipilih adalah file dengan format excel. Kemudian *user* melakukan load file agar data dapat ditampilkan, jika tidak melakukannya maka *user* akan tetap berada di halaman utama aplikasi tersebut. Setelah melakukan load file, maka data akan ditampilkan yaitu berupa data jadwal harian agen dan data waktu tempuh.

Selanjutnya *user* melakukan pencarian solusi dengan menginputkan beberapa data diantaranya *Pop size*, *Crossover Rate*, *Mutation Rate* dan *Maximum Generasi* untuk kemudian akan menghasilkan sebuah data keluaran yaitu Generasi Terbaik, Individu Terbaik, Pinalti, Nilai Fitness dan Nilai Rata-Rata. Dalam program juga telah dijelaskan informasi tentang hasil dari proses inisialisasi, proses mutasi dari beberapa induk, proses *crossover* dengan membangkitkan beberapa induk untuk menghasilkan individu baru (*children*), hasil reproduksi yang diketahui nilai pinalti dan fitness, serta proses seleksi individu menggunakan *roulette wheel*. Hingga akhirnya diperoleh individu terpilih pada generasi tersebut. Selanjutnya dilakukan iterasi sampai menemukan hasil pinalti dengan nilai minimum, hal ini berlanjut sesuai dengan maximum generasi yang telah diinputkan sebelumnya. *Output* yang dihasilkan dalam aplikasi ini adalah letak generasi terbaik, kromosom, nilai pinalti, nilai fitness, dan nilai rata-rata.

3.3 Desain Interface

Berikut ini adalah desain interface pada “*Aplikasi VRPTW Dengan Menggunakan Metode Genetic Algoritm Pada Studi Kasus Koran Harian Pagi Tribun Jabar*” yang dapat digambarkan sebagai berikut.

a. Desain Form Data Uji

Gambar 3.3 Desain Data Uji

b. Desain Form Hasil

The image shows a software window with three tabs: "Data Uji", "Hasil", and "tabPage3". The "Hasil" tab is selected. The interface includes several input fields for parameters: "Pop Size", "Cross Over Rate", "Mutation Rate", and "Maximum Generasi", each followed by a colon and a text box. A "Run" button is positioned to the right of these inputs. To the right of the inputs is a table with five columns: "Gen", "Individu Terbaik", "Pikalti", "Fitness", and "Rata-rata". Below the table is a large empty rectangular area labeled "Solusi".

Gambar 3.4 Desain Form Hasil

3.4 Spesifikasi Sistem

Spesifikasi sistem ini terdiri dari spesifikasi kebutuhan perangkat lunak dan perangkat keras. Berikut ini merupakan spesifikasi sistem untuk *Aplikasi VRPTW Dengan Menggunakan Metode Genetic Algoritm Pada Studi Kasus Koran Harian Pagi Tribun Jabar*.

3.4.1 Spesifikasi Perangkat Keras

Dalam pembuatan aplikasi ini digunakan sebuah komputer dengan spesifikasi perangkat keras seperti yang ditunjukkan pada tabel 4.1.

Tabel 3.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Processor	2.5 GHz Intel Core i5
Memory	4 GB
Harddisk	750GB
VGA Card	NVIDIA GEFORCE GT 635 – 2GB

3.4.2 Spesifikasi Perangkat Lunak

Dalam pembuatan aplikasi ini digunakan sebuah komputer dengan spesifikasi perangkat lunak seperti yang ditunjukkan pada tabel 4.2.

Tabel 3.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 8
Bahasa Pemrograman C#	Microsoft Visual Studio 2010 Ultimate
Microsoft Excel	Microsoft Excel 2010

BAB IV

IMPLEMENTASI

4.1 Tahapan Metode *Genetic Algorithm* persoalan VRPTW (*Vehicle Routing Problem with Time Windows*)

Berikut ini adalah perhitungan manual aplikasi VRPTW Dengan Menggunakan Metode *Genetic Algorithm* Pada Studi Kasus Koran Harian Pagi Tribun Jabar.

- a. Inisialisasi data uji yang digunakan diambil dari paper penelitian yang dilakukan oleh Agus Purnomo dengan data sebagai berikut dibawah ini:
 1. Data uji Inisialisasi Jadwal dimana telah ditentukan waktu pelayanan 5 menit seperti tabel dibawah ini:

Tabel 4.1 Jadwal Tiap Agen

No	Kode	Agen	Jadwal Buka	Jadwal Tutup
0	A	DEPOT(Rancaekek)		
1	B	Jalan Leuwipanjang	2:30	4:20
2	C	Jalan Terusan Pasirkoja	2:00	5:00
3	D	Jalan Antanan	2:30	5:00
4	E	Jalan Baladewa Indah	2:00	5:00
5	F	Jalan Dokter Otten	2:00	6:00
6	G	Jalan Cihampelas	2:30	6:00
7	H	Jalan Sukajadi	2:00	5:00
8	I	Jalan Sirnagalih	2:00	4:45
9	J	Jalan Listrik 2	2:10	2:35
10	K	Jalan Jenderal Abdul Haris Nasution	2:00	4:09
11	L	Jalan Caringin	2:20	5:09
12	M	Jalan Babakan Ciparay	2:30	5:11
13	N	Jalan Holis	2:30	5:00
14	O	Jalan Raya Cibeureum	2:00	4:19
15	P	Jalan Kebon Kawung	2:00	3:13
16	Q	Jalan Buah Batu	2:05	5:14

2. Data Uji waktu yang di tempuh dari kota satu ke kota lainnya seperti tabel dibawah ini:

Tabel 4.2 Waktu Tempuh

Kode	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
A	0	44	45	34	48	45	43	46	47	16	19	47	48	51	53	44	40
B	44	0	5	18	10	10	15	12	15	33	31	2	3	9	13	8	5

C	45	5	0	15	5	4	5	6	9	32	30	5	4	5	8	3	4
D	34	18	15	0	16	13	11	12	13	18	16	19	19	20	20	12	12
E	48	10	5	16	0	3	5	4	6	34	32	8	7	4	9	4	9
F	45	10	4	13	3	0	1	2	5	31	28	9	8	7	16	1	9
G	43	15	5	11	5	1	0	3	5	29	26	10	9	9	20	2	9
H	46	12	6	12	4	2	3	0	2	31	28	11	10	8	16	4	9
I	47	15	9	13	6	5	5	2	0	31	29	10	12	10	9	6	14
J	16	33	32	18	34	31	29	31	31	0	2	35	35	37	39	30	28
K	19	31	30	16	32	28	26	28	29	2	0	33	33	35	37	28	25
L	47	2	5	19	8	9	10	11	10	35	33	0	1	6	11	8	7
M	48	3	4	19	7	8	9	10	12	35	33	1	0	5	9	7	7
N	51	9	5	20	4	7	9	8	10	37	35	6	5	0	4	7	10
O	53	13	8	20	9	16	20	16	9	39	37	11	9	4	0	8	13
P	44	8	3	12	4	1	2	4	6	30	28	8	7	7	8	0	7
Q	40	40	4	12	9	9	9	9	14	28	25	7	7	10	13	7	0

- b. Sehingga diperoleh solusi dimana diambil dari generasi individu terbaik dari generasi 1 sampai generasi ke 500 dimana generasi terakhir merupakan generasi terbaik dengan menggunakan reproduksi crossover dan mutasi serta seleksi individu menggunakan roulette wheel. Generasi individu terbaik dipilih berdasarkan nilai total penalti terkecil. Dimana nilai pinalti diperoleh dari waktu berangkat dikurangi jadwal tutup apabila waktu datang tidak dalam waktu antara jadwal buka dengan jadwal tutup, namun apabila waktu datang masih dalam waktu antara jadwal buka dan jadwal tutup maka dianggap 0 untuk nilai pinaltinya, dimana nilai pinalti ini dalam hitungan menit. Seperti terlihat pada tabel dibawah ini:

Tabel 4.3 Solusi

Kota	Permasalahan			Solusi			Penalti
	Jadwal Buka	Jadwal tutup	Waktu Tempuh	Waktu datang	Pelayanan	Waktu Berangkat	
A	-	-	-	-		2:00	
J	2:10	2:35	16	2:16	0:05	2:21	0:00
K	2:00	4:09	2	2:23	0:05	2:28	0:00
P	2:00	3:13	28	2:56	0:05	3:01	0:00
C	2:00	5:00	3	3:04	0:05	3:09	0:00
L	2:20	5:09	5	3:14	0:05	3:19	0:00
B	2:30	4:20	2	3:21	0:05	3:26	0:00
M	2:30	5:11	3	3:29	0:05	3:34	0:00

O	2:00	4:19	9	3:43	0:05	3:48	0:00
N	2:30	5:00	4	3:52	0:05	3:57	0:00
E	2:00	5:00	4	4:01	0:05	4:06	0:00
F	2:00	6:00	3	4:09	0:05	4:14	0:00
I	2:00	4:45	5	4:19	0:05	4:24	0:00
H	2:00	5:00	2	4:26	0:05	4:31	0:00
D	2:30	5:00	12	4:43	0:05	4:48	0:00
Q	2:05	5:14	12	5:00	0:05	5:05	0:00
G	2:30	6:00	9	5:14	0:05	5:19	0:00
Total Pinalti							0:00

c. Proses Algoritma Genetika

1. Iterasi 1 Generasi 0

Dimana nilai pinalti di dapatkan dari waktu berangkat dikurangi jadwal tutup apabila waktu datang tidak dalam waktu antara jadwal buka dengan jadwal tutup, namun apabila waktu datang masih dalam waktu antara jadwal buka dan jadwal tutup maka dianggap 0 untuk nilai pinaltinya, dimana nilai pinalti ini dalam hitungan menit seperti pada table hasil diatas. Sedangkan nilai fitness di dapatkan dari rumus dibawah ini.

$$\frac{1}{(1+pinalti)/60}$$

Misalnya pada individu 1 $\text{Fitness} = \frac{1}{(1+pinalti)/60} = \frac{1}{(1+543)/60} = 0.1103$

a) Inisialisasi Populasi

Populasi inisialisasi ini dibangkitkan secara random, dalam hal ini ditentukan nilai $popsiz = 10$. Maka populasi yang dihasilkan adalah sebagai berikut.

Tabel 4.4 Inisialisasi Choromosom

Chromosom	
P1	0 15 5 3 4 2 6 7 8 9 10 11 12 13 14 1 16
P2	0 1 2 3 4 8 6 7 12 9 10 11 5 13 14 15 16
P3	0 1 11 3 4 12 6 7 8 9 10 2 5 13 14 15 16
P4	0 1 2 3 4 5 15 7 8 9 10 12 11 13 14 6 16
P5	0 1 12 3 4 5 6 7 8 15 10 11 2 13 14 9 16
P6	0 1 2 3 4 5 6 7 8 14 10 11 12 9 13 15 16

P7	0	1	2	3	4	5	6	7	11	10	9	8	12	13	14	15	16
P8	0	1	2	3	4	5	6	7	12	9	10	11	8	13	14	15	16
P9	0	1	2	3	4	12	16	7	8	9	10	11	5	13	14	15	6
P10	0	1	2	3	4	11	5	7	8	9	10	6	12	13	14	15	16

b) Reproduksi

1) Crossover

Proses crossover dalam aplikasi ini menggunakan metode *one cut point*, yaitu dengan memilih salah satu titik potong secara random. Dengan menggunakan nilai $pc=0,4$ dikali nilai $popsiz=10$ maka nilai *offspring* yang dihasilkan adalah 4. Berikut ini salah satu proses crossover pada parent 4 dan parent 1.

	Cut point ↓																
P1	0	15	5	3	4	2	6	7	8	9	10	11	12	13	14	1	16
P3	0	1	11	3	4	12	6	7	8	9	10	2	5	13	14	15	16
Child	0	15	5	3	4	2	6	7	8	9	10	11	12	13	1	14	16

2) Mutasi

Proses mutasi dalam aplikasi ini menggunakan metode *replace exchange mutation*, yaitu dengan memilih dua titik potong secara random, kemudian menukar nilai pada titik tersebut. Dengan menggunakan nilai $pm=0,2$ dikali nilai $popsiz=10$ maka nilai *offspring* yang dihasilkan adalah 2. Berikut ini salah satu proses mutasi pada parent 1.

	XP1 ↓ XP2 ↓																
P1	0	15	5	3	4	2	6	7	8	9	10	11	12	13	14	1	16
Child	0	15	5	14	4	2	6	7	8	9	10	11	12	13	3	1	16

3) Hasil Reproduksi

Hasil reproduksi dari proses *crossover* dan mutasi diperoleh data *offspring* seperti pada table berikut ini.

Tabel 4.5 Nilai *offspring*

	Chromosom
C1	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16
C2	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16
C3	0 1 12 3 4 5 6 7 8 15 10 11 2 13 9 14 16

C4	0 1 11 3 4 12 6 7 8 9 10 2 5 13 14 15 16
C5	0 15 5 14 4 2 6 7 8 9 10 11 12 13 3 1 16
C6	0 1 2 14 4 5 6 7 12 9 10 11 8 13 3 15 16

c) Seleksi

Proses seleksi ini dilakukan dengan menggabungkan individu dan *offspring*. Kemudian dipilih 10 terbaik dari 16 individu yang ada. Metode yang digunakan dalam proses seleksi ini adalah *roulette wheel*. Metode ini dengan menghitung nilai probabilitas kumulatif (*probCum*) pada setiap individu. ProbCum didapatkan dari nilai probabilitas kumulatif dari fitness di bagi total fitness.

$$\text{Misanya ProbCum individu 1} = \frac{0.1103}{1.4536} = 0.759$$

Tabel 4.6 Individu hasil proses evaluasi

	Chromosome	Fitness	ProbCum
P1	0 15 5 3 4 2 6 7 8 9 10 11 12 13 14 1 16	0.1103	0.0759
P2	0 1 2 3 4 8 6 7 12 9 10 11 5 13 14 15 16	0.0799	0.1308
P3	0 1 11 3 4 12 6 7 8 9 10 2 5 13 14 15 16	0.0878	0.1912
P4	0 1 2 3 4 5 15 7 8 9 10 12 11 13 14 6 16	0.1165	0.2714
P5	0 1 12 3 4 5 6 7 8 15 10 11 2 13 14 9 16	0.0839	0.3291
P6	0 1 2 3 4 5 6 7 8 14 10 11 12 9 13 15 16	0.0678	0.3758
P7	0 1 2 3 4 5 6 7 11 10 9 8 12 13 14 15 16	0.0792	0.4302
P8	0 1 2 3 4 5 6 7 12 9 10 11 8 13 14 15 16	0.0759	0.4825
P9	0 1 2 3 4 12 16 7 8 9 10 11 5 13 14 15 6	0.0884	0.5433
P10	0 1 2 3 4 11 5 7 8 9 10 6 12 13 14 15 16	0.0894	0.6048
C1	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16	0.1058	0.6776
C2	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16	0.1058	0.7503
C3	0 1 12 3 4 5 6 7 8 15 10 11 2 13 9 14 16	0.0751	0.802
C4	0 1 11 3 4 12 6 7 8 9 10 2 5 13 14 15 16	0.0878	0.8624
C5	0 15 5 14 4 2 6 7 8 9 10 11 12 13 3 1 16	0.1134	0.9404
C6	0 1 2 14 4 5 6 7 12 9 10 11 8 13 3 15 16	0.0866	1
TOTAL		1.4536	

d) Individu Terpilih

Individu terpilih diperoleh dari hasil perhitungan *roulette wheel* dengan langkah sebagai berikut ini:

1. Membangkitkan nilai r secara random
2. Pilih nilai k mulai dari 2 sampai $popsiz$ sehingga $r \leq probCum$

P(t+1)	Random	asal	Chromosome	Fitness
P1	0.6452	11	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16	0.1058
P2	0.37	6	0 1 2 3 4 5 6 7 8 14 10 11 12 9 13 15 16	0.0678
P3	0.8893	15	0 15 5 14 4 2 6 7 8 9 10 11 12 13 3 1 16	0.1134
P4	0.7217	12	0 15 5 3 4 2 6 7 8 9 10 11 12 13 1 14 16	0.1058
P5	0.1036	2	0 1 2 3 4 8 6 7 12 9 10 11 5 13 14 15 16	0.0799
P6	0.893	15	0 15 5 14 4 2 6 7 8 9 10 11 12 13 3 1 16	0.1134
P7	0.5247	9	0 1 2 3 4 12 16 7 8 9 10 11 5 13 14 15 6	0.0884
P8	0.8391	14	0 1 11 3 4 12 6 7 8 9 10 2 5 13 14 15 16	0.0878
P9	0.5105	9	0 1 2 3 4 12 16 7 8 9 10 11 5 13 14 15 6	0.0884
P10	0.7972	13	0 1 12 3 4 5 6 7 8 15 10 11 2 13 9 14 16	0.0751

Nilai Rata-rata fitness = 0.09258

Individu terbaik adalah : 0 15 5 14 4 2 6 7 8 9 10 11 12 13 3 1 16 Fitness = 0.1134

1. Iterasi 2 Generasi 1 sampai dengan iterasi generasi terakhir proses nya tetap hanya inisialisasi populasi dipilih dari individu terpilih pada iterasi sebelumnya.

4.2 Implementasi Algoritma

Aplikasi ini mempunyai beberapa proses utama yaitu proses load data yang diambil dari *Microsoft Excel* dan perhitungan rute untuk permasalahan *Vehicle Routing Problem With Time Windows* (VRPTW).

4.2.1 Implementasi Algoritma Proses Load Data

Pada proses ini, user diminta untuk load data dari excel dengan memilih file yang akan di load. Selanjutnya user diminta menginputkan nama sheet jadwal serta waktu tempuh sesuai dengan sheet yang digunakan pada file excel. Proses ini menyimpan jadwal harian agen serta matrik waktu tempuh setiap node. Berikut merupakan implementasi algoritma pada proses load data.

```

1 private void btnLoadFile_Click(object sender, EventArgs e)
2 {
3     try
4     {
5         string pathConn =
6         "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + txtPath.Text + ";Extended
7         Properties=\\Excel 8.0;HDR=Yes;\\";
8         OleDbConnection conn = new OleDbConnection(pathConn);
9
10        OleDbDataAdapter myDataAdapter = new OleDbDataAdapter("select *
11        from [" + txtSheetJadwal.Text + "$]", conn);
12        DataTable dt = new DataTable();
13        myDataAdapter.Fill(dt);
14        dgvJadwal.DataSource = dt;
15        myDataAdapter = new OleDbDataAdapter("select * from [" +
16        txtSheetWaktu.Text + "$]", conn);
17        dt = new DataTable();
18        myDataAdapter.Fill(dt);
19        dgvWaktu.DataSource = dt;
20    }
21    catch (Exception ex)
22    {
23        MessageBox.Show("Error : " + ex.ToString());
24    }
25    private void btnGo_Click(object sender, EventArgs e)
26    {
27        try
28        {
29            totData = dgvJadwal.Rows.Count - 1;
30            agen = new string[totData];
31            jadwalBuka = new string[totData];
32            jadwalTutup = new string[totData];
33            int x = 0;
34            while (x < totData)
35            {
36                agen[x] = dgvJadwal["Agen", x].Value.ToString();
37                jadwalBuka[x] = dgvJadwal["Jadwal Buka", x].Value.ToString();
38                jadwalTutup[x] = dgvJadwal["Jadwal Tutup",
39                x].Value.ToString();
40                x++;
41            }
42            matrixWaktu = new int[totData, totData];
43            for (int i = 0; i < totData; i++)
44            {
45                for (int j = 0; j < totData; j++)
46                {
47                    matrixWaktu[i,j] = Convert.ToInt32(dgvWaktu[j, i].Value);
48                }
49                Console.WriteLine();
50            }
51            tabControl1.SelectedIndex = 1;
52        }
53        catch (Exception ex)
54        {
55            MessageBox.Show("Error : " + ex.ToString());
56        }
57    }
58 }

```

Gambar 4.1 Implementasi Algoritma Load Data
Sumber : Implementasi

Penjelasan algoritma proses *login* pada Gambar 4.1 yaitu :

1. Baris 1-26 menjelaskan program membuka koneksi untuk dilakukan load data dari file excel ke datagridview.
2. Baris 27-63 menjelaskan ketika button diklik, maka data jadwal agen dan matrik waktu tempuh disimpan kedalam variabel masing-masing.

4.2.2 Implementasi Algoritma Proses Pemilihan Rute

Pada proses ini dilakukan pemilihan rute terbaik dengan menggunakan metode algoritma genetika. Berikut ini merupakan implementasi proses pemilihan rute terbaik.

1	<code>private void btnRun_Click(object sender, EventArgs e)</code>
2	<code>{</code>
3	<code>try</code>
4	<code>{</code>
5	<code>valPopSize = Convert.ToInt32(txtPopSize.Text);</code>
6	<code>valCr = Convert.ToDouble(txtCr.Text);</code>
7	<code>valMr = Convert.ToDouble(txtMr.Text);</code>
8	
9	<code>totChildCrossOver = Convert.ToInt32(valCr * valPopSize);</code>
10	<code>totChildMutation = Convert.ToInt32(valMr * valPopSize);</code>
11	<code>totChild = totChildCrossOver + totChildMutation;</code>
12	
13	<code>rtbSolusi.Text = "";</code>
14	<code>dgvResult.Rows.Clear();</code>
15	
16	<code>int z = 0;</code>
17	<code>r = new Random();</code>
18	<code>while (z < generasi)</code>
19	<code>{</code>
20	<code>totFitness = 0;</code>
21	<code>fitnessParents = new double[valPopSize];</code>
22	<code>fitness = new double[valPopSize + totChild];</code>
23	<code>pinalti = new double[valPopSize + totChild];</code>
24	
25	<code>strIndividu = new string[valPopSize + totChild];</code>
26	<code>newParents = new string[valPopSize + totChild, totData];</code>
27	
28	<code>for (int i = 0; i < valPopSize; i++)</code>
29	<code>{</code>
30	<code>if (z < 3)</code>
31	<code>{</code>
32	<code>rtbSolusi.Text += (i + 1).ToString() + "\t";</code>
33	<code>}</code>
34	<code>tempParents1 = new string[totData];</code>
35	
36	<code>tempIndividu = "";</code>
37	<code>for (int j = 0; j < totData; j++)</code>
38	<code>{</code>
39	<code>if (z < 3)</code>
40	<code>{</code>
41	<code>rtbSolusi.Text += parents[i, j].ToString() + " ";</code>
42	<code>}</code>
43	<code>tempIndividu += parents[i, j].ToString() + " ";</code>
44	<code>tempParents1[j] = parents[i, j].ToString();</code>


```

45         newParents[i, j] = parents[i, j];
46     }
47     strIndividu[i] = tempIndividu;
48     seleksi s = new seleksi(tempParents1, matrixWaktu, jadwalBuka,
49 jadwalTutup);
50     s.cariNilaiFitness();
51     pinalti[i] = s.fitness;
52
53     fitnessParents[i] =
54     Math.Round(1/((Convert.ToDouble(s.fitness+1) / 60)), 4);
55     fitness[i] = fitnessParents[i];
56     totFitness += fitnessParents[i];
57 }
58 childs = new string[totChild, totData];
59 reproduksi rep;
60
61 x = 0;
62
63 valRandom1 = r.Next(3, totData-2);
64 while (x < totChildCrossOver)
65 {
66     noParent1 = r.Next(0, valPopSize);
67     noParent2 = r.Next(0, valPopSize);
68
69     tempParents1 = new string[totData];
70     tempParents2 = new string[totData];
71     tempChild = new string[totData];
72
73
74     if (noParent1 != noParent2)
75     {
76
77         for (int i = 0; i < totData; i++)
78         {
79             tempParents1[i] = parents[noParent1, i];
80             tempParents2[i] = parents[noParent2, i];
81         }
82
83         rep = new reproduksi(tempParents1, tempParents2,
84 valRandom1);
85         rep.reproduksiCrossover();
86         tempChild = rep.child;
87
88         tempIndividu = "";
89         for (int i = 0; i < totData; i++)
90         {
91             childs[x, i] = tempChild[i];
92             if (z < 3)
93             {
94                 rtbSolusi.Text += tempChild[i].ToString() + " ";
95             }
96             tempIndividu += tempChild[i].ToString() + " ";
97         }
98         strIndividu[valPopSize + x] = tempIndividu;
99     }
100     else
101     {
102         x--;
103     }
104
105     x++;
106 }

```

```

107
108         x = 0;
109
110         valRandom1 = r.Next(1, totData/2);
111         valRandom2 = r.Next(totData/2, totData);
112         while (x < totChildMutation)
113         {
114             noParent1 = r.Next(0, valPopSize);
115             tempParents1 = new string[totData];
116             tempChild = new string[totData];
117
118             for (int i = 0; i < totData; i++)
119             {
120                 tempParents1[i] = parents[noParent1, i];
121             }
122
123             rep = new reproduksi(tempParents1, valRandom1, valRandom2);
124             rep.reproduksiMutasi();
125             tempChild = rep.child;
126
127             tempIndividu = "";
128             for (int i = 0; i < totData; i++)
129             {
130                 childs[x + totChildCrossOver, i] = tempChild[i];
131                 if (z < 3)
132                 {
133                     rtbSolusi.Text += tempChild[i].ToString() + " ";
134                 }
135                 tempIndividu += tempChild[i].ToString() + " ";
136             }
137             strIndividu[valPopSize + totChildCrossOver + x] =
138 tempIndividu;
139
140             x++;
141         }
142
143         fitnessChild = new double[totChild];
144
145         for (int i = 0; i < totChild; i++)
146         {
147             if (z < 3)
148             {
149                 rtbSolusi.Text += (i + 1).ToString() + "\t";
150             }
151
152             tempChild = new string[totData];
153
154             for (int j = 0; j < totData; j++)
155             {
156                 if (z < 3)
157                 {
158                     rtbSolusi.Text += childs[i, j].ToString() + " ";
159                 }
160                 tempChild[j] = childs[i, j].ToString();
161                 newParents[valPopSize + i, j] = childs[i, j];
162             }
163             seleksi s = new seleksi(tempChild, matrixWaktu, jadwalBuka,
164 jadwalTutup);
165             s.cariNilaiFitness();
166
167             pinalti[i+valPopSize] = s.fitness;
168

```

```

169         fitnessChild[i] = Math.Round(1/((Convert.ToDouble(s.fitness+1)
170 / 60)), 4);
171         totFitness += fitnessChild[i];
172         fitness[valPopSize + i] = fitnessChild[i];
173     }
174
175     prob = 0;
176     totProbCum = 0;
177
178     totGenerasi = valPopSize + totChild;
179     probCum = new double[totGenerasi];
180
181     for (int i = 0; i < totGenerasi; i++)
182     {
183         prob = fitness[i] / totFitness;
184         totProbCum += prob;
185         probCum[i] = Math.Round(totProbCum, 4);
186     }
187     tempDouble = 0;
188     individuTerpilih = new string[valPopSize];
189     fitnessTerpilih = new double[valPopSize];
190     penaltiTerpilih = new double[valPopSize];
191
192     tempRata = 0;
193     int terbaik = 0;
194
195     for (int i = 0; i < valPopSize; i++)
196     {
197         tempDouble = r.NextDouble();
198         x = 0;
199         int terpilih = 0;
200         while (probCum[x] < tempDouble)
201         {
202             terpilih = x + 1;
203             x++;
204         }
205
206         tempRata += fitness[terpilih];
207         individuTerpilih[i] = strIndividu[terpilih];
208         fitnessTerpilih[i] = fitness[terpilih];
209         penaltiTerpilih[i] = pinalti[terpilih];
210
211         if (fitnessTerpilih[terbaik] > fitnessTerpilih[i])
212         {
213             terbaik = terpilih;
214         }
215         else
216         {
217             terbaik = i;
218         }
219
220         for (int j = 0; j < totData; j++)
221         {
222             parents[i,j] = newParents[terpilih,j];
223         }
224     }
225
226     tempRata = tempRata / valPopSize;
227
228     generasiBestIndividu[z] = individuTerpilih[terbaik].ToString();
229     generasiPinalti[z] = penaltiTerpilih[terbaik];
230     generasiRataRata[z] = tempRata;

```

```

231         generasiFitnessTerbaik[z] = fitnessTerpilih[terbaik];
232         z++;
233     }
234
235     int tempTerbaik = 0;
236     for (int i = 0; i < generasi; i++)
237     {
238         if (generasiFitnessTerbaik[tempTerbaik] >
239 generasiFitnessTerbaik[i])
240         {
241             tempTerbaik = tempTerbaik;
242         }
243         else
244         {
245             tempTerbaik = i;
246         }
247         row = new string[] { (i+1).ToString(),
248 generasiBestIndividu[i].ToString(), generasiPinalti[i].ToString(),
249 generasiFitnessTerbaik[i].ToString(), generasiRataRata[i].ToString()};
250         dgvResult.Rows.Add(row);
251     }
252
253     rtbSolusi.Text +=
254     "\n\n=====
255     =====\n\n";
256     rtbSolusi.Text += "Generasi terbaik didapatkan pada" +
257     "\ngenerasi = " + (tempTerbaik + 1).ToString() +
258     "\nKromosom = " + generasiBestIndividu[tempTerbaik].ToString() +
259     "\nPinalti = " + generasiPinalti[tempTerbaik].ToString() +
260     "\nFitness = " + generasiFitnessTerbaik[tempTerbaik].ToString() +
261     "\nRata-rata = " + generasiRataRata[tempTerbaik].ToString();
262     rtbSolusi.Text += "\nDetail Solusi";
263     rtbSolusi.Text +=
264     "\n=====
265     =====\n\n";
266
267
268     string[] strind = generasiBestIndividu[tempTerbaik].Split(' ');
269     string[] newStrind = new string[totData];
270
271     for (int i = 0; i < totData; i++)
272     {
273         newStrind[i] = strind[i];
274     }
275
276
277     tempWaktuDatang = new string[totData];
278     tempWaktuBerangkat = new string[totData];
279     penalti = new int[totData];
280     tempWaktuTempuh = new int[totData];
281
282     seleksi sel = new
283 seleksi(newStrind,matrixWaktu,jadwalBuka,jadwalTutup);
284     sel.cariNilaiFitness();
285     tempWaktuDatang = sel.tempWaktuDatang;
286     tempWaktuBerangkat = sel.tempWaktuBerangkat;
287     penalti = sel.penalti;
288     tempWaktuTempuh = sel.waktuTempuh;
289
290     rtbSolusi.Text += "Berangkat dari node 0 pada pukul 2.00\n";
291
292     for (int i = 1; i < newStrind.Length; i++)

```

293	{
294	node = Convert.ToInt32(newStrind[i].ToString());
295	
296	rtbSolusi.Text += node.ToString() + "\tBuka = " +
297	(jadwalBuka[node].ToString()).Substring(10,5) + "\tTutup = " +
298	(jadwalTutup[node].ToString()).Substring(10,5) + "\tWaktu Tempuh = "+
299	tempWaktuTempuh[i].ToString()+
300	"\tDatang = "+ (tempWaktuDatang[i].ToString()).Substring(0,5)
301	+"\tPelayanan = 5" +"\tWaktu Berangkat =
302	+(tempWaktuBerangkat[i].ToString()).Substring(0,5)+"\tPenalti = "+
303	penalti[i].ToString()+"\n";
304	}
305	
306	}

Gambar 4.2 Implementasi Algoritma Perhitungan Rule
Sumber : Implementasi

Penjelasan algoritma proses pembentukan rule pada gambar 4.2 yaitu :

1. Baris 5-11 menjelaskan proses penyimpana nilai pop size, crossover rate, mutation rate, dan jumlah generasi dari textbox masing-masing.
2. Baris 13-59 menjelaskan proses pembangkitan generasi awal sebagai parents dari node yang ada.
3. Baris 61-106 menjelaskan proses crossover. Pada proses ini, parents diambil secara acak untuk kemudian dilakukan proses crossover dengan metode one-cut-point.
4. Baris 110-141 menjelaskan proses mutasi.pada proses ini, parents diambil secara acak untuk kemudian dilakukan proses mutasi dengan metode exchange mutation.
5. Baris 143-172 menjelaskan proses perhitungan nilai fitness untuk semua individu baik itu parents maupun child.
6. Baris 173-185 menjelaskan proses perhitungan nilai probabilitas dan probabilitas kumulatif untuk setiap individu.
7. Baris 186-233 menjelaskan proses pemilihan individu terbaik sebanyak pop size berdasarkan metode roulette wheel. Nilai diambil secara acak antara 0 sampai dengan 1 kemudian memilih peluang berdasarkan nilai probabilitas kumulatifnya.
8. Baris 235-251 menjelaskan proses pemilihan individu terbaik dari generasi tersebut.
9. Baris 253-306 menjelaskan proses menampilkan detail perhitungan untuk menghasilkan nilai pinalti dari individu terbaik.

4.3 Pembuatan User Interface

Berikut ini adalah gambar form-form yang ada dalam aplikasi berdasarkan desain yang dibuat sebelumnya.

a. Form Data Uji

Gambar 4.3 Form Data Uji

b. Form Hasil

Gen	Individu Terbaik	Pinalti	Fitness	Rata-Rata
*				

Gambar 4.4 Form Hasil

BAB V

PENGUJIAN

5.1 Hasil Pengujian

Berikut ini adalah gambar hasil pengujian yang telah dilakukan dalam aplikasi.

1. Load data uji

The screenshot shows the 'Form1' application window with the 'Data Uji' tab selected. It includes input fields for 'Pilih Lokasi Data.xls', 'Inputkan Nama Sheet Jadwal', and 'Inputkan Nama Sheet Waktu Tempuh'. Below these are two data tables: 'Jadwal Harian Agen' and 'Waktu Tempuh'.

No	Agen	Jadwal Buka	Jadwal Tutup
0	DEPOT(Rancaek...	12/30/1899 2...	12/3
1	Jalan Leuwipanj...	12/30/1899 2...	12/3
2	Jalan Terusan P...	12/30/1899 2...	12/3
3	Jalan Antanan	12/30/1899 2...	12/3
4	Jalan Baladewa I...	12/30/1899 2...	12/3
5	Jalan Dokter Ott...	12/30/1899 2...	12/3
6	Jalan Cihampelas	12/30/1899 2...	12/3
7	Jalan Sukajadi	12/30/1899 2...	12/3
8	Jalan Sirnagalih	12/30/1899 2...	12/3
9	Jalan Listrik 2	12/30/1899 2...	12/3
10	Jalan Jenderal A...	12/30/1899 2...	12/3
11	Jalan Caringin	12/30/1899 2...	12/3
12	Jalan Babakan C...	12/30/1899 2...	12/3
13	Jalan Holis	12/30/1899 2...	12/3

A	B	C	D	E
0	44	45	34	48
44	0	5	18	10
45	5	0	15	5
34	18	15	0	16
48	10	5	16	0
45	10	4	13	3
43	15	5	11	5
46	12	6	12	4
47	15	9	13	6
16	33	32	18	34
19	31	30	16	32
47	2	5	19	8
48	3	4	19	7
51	9	5	20	4

Gambar 5.1 Hasil Load Data Uji

2. Hasil data uji

The screenshot shows the 'Form1' application window with the 'Hasil' tab selected. It displays the results of a genetic algorithm, including parameters like Pop Size, Cross Over Rate, and Mutation Rate, and a table of the best individual's performance.

Gen	Individu Terbaik	Pinalti	Fitness	Rata-Rata
497	0 9 10 15 12 11 1 14 13 4 2 5 8 7 3 16 6	0	60	60
498	0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6	0	60	60
499	0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6	0	60	60
500	0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6	0	60	60

Generasi terbaik didapatkan pada generasi = 500
Kromosom = 0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6
Pinalti = 0
Fitness = 60
Rata-rata = 60

Detail Solusi

Berangkat dari node 0 pada pukul 2.00

No	Buka	Tutup	Waktu Tempuh	Datang	Pelayanan	Waktu Berangkat	Penalti
9	Buka = 2:10	Tutup = 2:35	Waktu Tempuh = 16	Datang = 02:16	Pelayanan = 5	Waktu Berangkat = 02:21	Penalti = 0
10	Buka = 2:00	Tutup = 4:09	Waktu Tempuh = 2	Datang = 02:23	Pelayanan = 5	Waktu Berangkat = 02:28	Penalti = 0
15	Buka = 2:00	Tutup = 3:13	Waktu Tempuh = 28	Datang = 02:56	Pelayanan = 5	Waktu Berangkat = 03:01	Penalti = 0
2	Buka = 2:00	Tutup = 5:00	Waktu Tempuh = 3	Datang = 03:04	Pelayanan = 5	Waktu Berangkat = 03:09	Penalti = 0
11	Buka = 2:20	Tutup = 5:09	Waktu Tempuh = 5	Datang = 03:14	Pelayanan = 5	Waktu Berangkat = 03:19	Penalti = 0
1	Buka = 2:30	Tutup = 4:20	Waktu Tempuh = 2	Datang = 03:21	Pelayanan = 5	Waktu Berangkat = 03:26	Penalti = 0
12	Buka = 2:30	Tutup = 5:11	Waktu Tempuh = 3	Datang = 03:29	Pelayanan = 5	Waktu Berangkat = 03:34	Penalti = 0
14	Buka = 2:00	Tutup = 4:19	Waktu Tempuh = 9	Datang = 03:43	Pelayanan = 5	Waktu Berangkat = 03:48	Penalti = 0
13	Buka = 2:30	Tutup = 5:00	Waktu Tempuh = 4	Datang = 03:52	Pelayanan = 5	Waktu Berangkat = 03:57	Penalti = 0
4	Buka = 2:00	Tutup = 5:00	Waktu Tempuh = 4	Datang = 04:01	Pelayanan = 5	Waktu Berangkat = 04:06	Penalti = 0
5	Buka = 2:00	Tutup = 6:00	Waktu Tempuh = 3	Datang = 04:09	Pelayanan = 5	Waktu Berangkat = 04:14	Penalti = 0
8	Buka = 2:00	Tutup = 4:45	Waktu Tempuh = 5	Datang = 04:19	Pelayanan = 5	Waktu Berangkat = 04:24	Penalti = 0
7	Buka = 2:00	Tutup = 5:00	Waktu Tempuh = 2	Datang = 04:26	Pelayanan = 5	Waktu Berangkat = 04:31	Penalti = 0
3	Buka = 2:30	Tutup = 5:00	Waktu Tempuh = 12	Datang = 04:43	Pelayanan = 5	Waktu Berangkat = 04:48	Penalti = 0
16	Buka = 2:05	Tutup = 5:14	Waktu Tempuh = 12	Datang = 05:00	Pelayanan = 5	Waktu Berangkat = 05:05	Penalti = 0
6	Buka = 2:30	Tutup = 6:00	Waktu Tempuh = 9	Datang = 05:14	Pelayanan = 5	Waktu Berangkat = 05:19	Penalti = 0

Gambar 5.2 Hasil Data Uji

5.2 Analisa Pengujian

Hasil Pengujian di peroleh dari beberapa kali iterasi, dimana iterasi terjadi berdasarkan banyaknya generasi yang diminta. Sehingga diperoleh generasi individu terbaik dengan menggunakan reproduksi crossover dan mutasi serta menggunakan proses seleksi roulette wheel. Individu terbaik ini didapatkan dari perhitungan nilai total pinalti terkecil, dimana nilai pinalti diperoleh dari waktu berangkat dikurangi jadwal tutup apabila waktu datang tidak dalam waktu antara jadwal buka dengan jadwal tutup, namun apabila waktu datang masih dalam waktu antara jadwal buka dan jadwal tutup maka dianggap 0 untuk nilai pinaltinya, dimana nilai pinalti ini dalam hitungan menit. Nama kota diinisialisasikan dari tempat pertama hingga tempat terakhir yaitu node 0 sampai 16.

Dimana hasil individu terbaik di dapatkan dari generasi terakhir dimana nilai pinaltinya paling kecil seperti dibawah ini:

Generasi terbaik didapatkan pada generasi = 500

Kromosom = 0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6

Pinalti = 0

Fitness = 60

Rata-rata = 60

Detail Solusi

=====

Tabel 5.1 Hasil Pengujian

Berangkat dari node 0 pada pukul 2.00

Node	Buka	Tutup	Waktu Tempuh	Datang	Pelayanan	Waktu Berangkat	Penalti
9	2:10	2:35	16	02:16	5	02:21	0
10	2:00	4:09	2	02:23	5	02:28	0
15	2:00	3:13	28	02:56	5	03:01	0
2	2:00	5:00	3	03:04	5	03:09	0
11	2:20	5:09	5	03:14	5	03:19	0
1	2:30	4:20	2	03:21	5	03:26	0
12	2:30	5:11	3	03:29	5	03:34	0
14	2:00	4:19	9	03:43	5	03:48	0
13	2:30	5:00	4	03:52	5	03:57	0
4	2:00	5:00	4	04:01	5	04:06	0

5	2:00	6:00	3	04:09	5	04:14	0
8	2:00	4:45	5	04:19	5	04:24	0
7	2:00	5:00	2	04:26	5	04:31	0
3	2:30	5:00	12	04:43	5	04:48	0
16	2:05	5:14	12	05:00	5	05:05	0
6	2:30	6:00	9	05:14	5	05:19	0

BAB VI

PENUTUP

6.1 Kesimpulan

Persoalan VRPTW pada studi kasus pendistribusian koran Harian Pagi Tribun Jabar dapat diselesaikan dengan menggunakan beberapa metode antara lain *Nearest Insertion Heuristic*, selain itu dapat diselesaikan dengan metode *Genetic Algorithm*. Dalam hal ini metode yang digunakan adalah *Genetic Algorithm*. Data yang digunakan dalam aplikasi ini adalah berupa data jadwal agen koran, dan waktu tempuh. Hal ini dilakukan untuk meminimalkan biaya dan menghemat waktu agar pengirim dapat mengirimkan koran ke beberapa tujuan sesuai dengan jadwal harian tiap agen tujuan.

Proses penyelesaian masalah ini Metode *Genetic Algorithm* (GA) dimulai dengan inisialisasi populasi, proses reproduksi yang terdiri dari *crossover* dan mutasi, selanjutnya adalah proses seleksi, dan yang terakhir adalah *output* berupa hasil generasi yang terpilih. Proses iterasi berlangsung sesuai dengan banyaknya Maximum Generation yang diinputkan. Di akhir iterasi dihasilkan generasi yang terpilih yang menghasilkan nilai $pinalti=0$. Berdasarkan hasil pengujian diatas, generasi terpilih adalah Kromosom = 0 9 10 15 2 11 1 12 14 13 4 5 8 7 3 16 6. Dari hasil akhir tersebut dapat diketahui rute mana sesuai dengan node atau tujuan yang harus diprioritaskan untuk pelayanan terlebih dahulu.

6.2 Saran

Aplikasi ini dapat dikembangkan dengan menggunakan metode lain yang lebih efektif, sehingga dapat menghasilkan data yang lebih akurat. Selain itu untuk desain interface dalam aplikasi ini dapat dikembangkan sesuai dengan kebutuhan pengguna untuk kedepannya.

DAFTAR PUSTAKA

- Adi Nugroho. 2010. *Mengembangkan Basis Data Menggunakan C# dan SQL Server*. Yogyakarta: AndiOFFSET
- Agia Angga. *Rancang Bangun Sistem Informasi Distribusi Air Minum Dalam Kemasan dengan Model Arus Jaringan Pada CV. Sumber Nadi Jaya*. [online] <http://dazzdays.wordpress.com/tag/metode-heuristik/>
- Agus Purnomo. 2010. *Analisis Rute Pendistribusian dengan Menggunakan Metode Nearest Insertion Heuristic Persoalan Vehicle Routing Problem with Time Windows (VRPTW) (Studi Kasus di Koran Harian Pagi Tribun Jabar)*. Jurusan Teknik Industri. Fakultas Teknik Universitas Pasundan.
- Aji Raditya. *Penggunaan Metode Heuristik Dalam Permasalahan Vehicle Routing Problem dan Implementasinya di PT Nippon Industri Corbindo*. Institut Pertanian Bogor. Bandung
- Andiwayudin. 2011. *Standart Simbol Flowchart*. <http://abunajmu.wordpress.com/2011/07/07/standar-simbol-flowchart/>
- Ferry. 2011. *Pengenalan dan Definisi Flowchart*. <http://twokanggrafis.blogspot.com/2011/12/pengenalan-dan-definisi-flowchart.html>
- Hafizh, Muhammad. 2010. *Perancangan Sistem Informasi Pemesanan Barang Online pada Distro Black Metal Menggunakan ASP.Net dan SQL Server 2000*. Tugas Akhir. Universitas Brawijaya Malang
- Primasari, Donna. 2011. *Perancangan Sistem Informasi pada Toko Sahaaba*. Skripsi. Universitas Diponegoro. <http://eprints.undip.ac.id/27077/1/repositori.pdf>
- Wayan FM. 2013. *Travelling Salesman Problem*. Modul Matakuliah Algoritma Evolusi. PTIIK Universitas Brawijaya Malang
- Wiliam Tanujaya dkk. *Penerapan Algoritma Genetik Untuk Penyelesaian Masalah Vehicle Routing Di PT.MIF*. Journal Mahasiswa Fakultas Teknik Industri. Universitas Katolik Widya Mandala Surabaya
- _____, *Microsoft Visual C#*. [online] http://id.wikipedia.org/wiki/Microsoft_Visual_C_Sharp
- _____, *CSharp*. [online] http://id.wikipedia.org/wiki/C_sharp
- _____, *Tutorial Travelling Salesman Problem*. [online] <http://www.metode-algoritma.com/2013/06/tutorial-travelling-salesman-problem-tsp.html>
- _____, *Vehicle Routing Problem*. [online] <http://digilib.itelkom.ac.id/index.php?view=article&catid=20:informatika&id=607:vehicleroutingproblemvrp&option=comcontent&Itemid=14>