

**Penyelesaian *Vehicle Routing Problem with Time Windows* (VRPTW)
Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)**

SKRIPSI

**Untuk Memenuhi Sebagian Persyaratan Mencapai Gelar Sarjana Komputer
Pada Program Teknologi Informasi Dan Ilmu Komputer**

UNIVERSITAS BRAWIJAYA



Oleh:

I Made Riko Adi Saputra

115060807111087

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

MALANG

2015

LEMBAR PERSETUJUAN

Penyelesaian *Vehicle Routing Problem with Time Windows* (VRPTW)

Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar sarjana komputer



Disusun Oleh:

I MADE RIKO ADI SAPUTRA

115060807111087

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si., M.Kom

NIK. 850719 16 1 1 0422

Edy Santoso, S.Si., M.Kom

NIP. 19740414 200312 1 004

LEMBAR PENGESAHAN

Penyelesaian *Vehicle Routing Problem with Time Windows* (VRPTW)

Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)

SKRIPSI

KONSENTRASI KOMPUTASI CERDAS DAN VISUALISASI

Untuk memenuhi sebagian persyaratan mencapai gelar sarjana komputer

Disusun Oleh:

I MADE RIKO ADI SAPUTRA

115060807111087

Skripsi ini telah diuji dan dinyatakan lulus

Tanggal 7 Agustus 2015

Dosen Penguji I

Dosen Penguji II

Dr. Eng. Fitri Utamaningrum., S.T,M.T

Ir. Heru Nurwasito, M.Kom.

NIP. 19820710 200812 2 001

NIP. 19650402 199002 1 001

Dosen Penguji III

M. Ali Fauzi S.Kom.,M.Kom.

NIK. 2015028901011000

Mengethui,

Ketua Program Studi Teknik Informatika/Illmu Komputer

Drs. Marji, MT

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan di bawah ini,

Nama : I Made Riko Adi Saputra
NIM : 115060807111087
Jurusan/Program studi : Informatika/Ilmu Komputer
Fakultas/Program : Fakultas Ilmu Komputer/Strata-1

Menyatakan dengan sungguh-sungguh bahwa skripsi yang saya tulis ini benar-benar tulisan saya, dan bukan merupakan plagiat baik sebagian atau seluruhnya. Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil plagiat, baik sebagian atau seluruhnya, maka saya bersedia menerima sanksi atas perbuatan tersebut sesuai dengan ketentuan yang berlaku.

Malang, 7 Agustus 2015

Mahasiswa,

I Made Riko Adi Saputra

NIM 115060807111087

KATA PENGANTAR

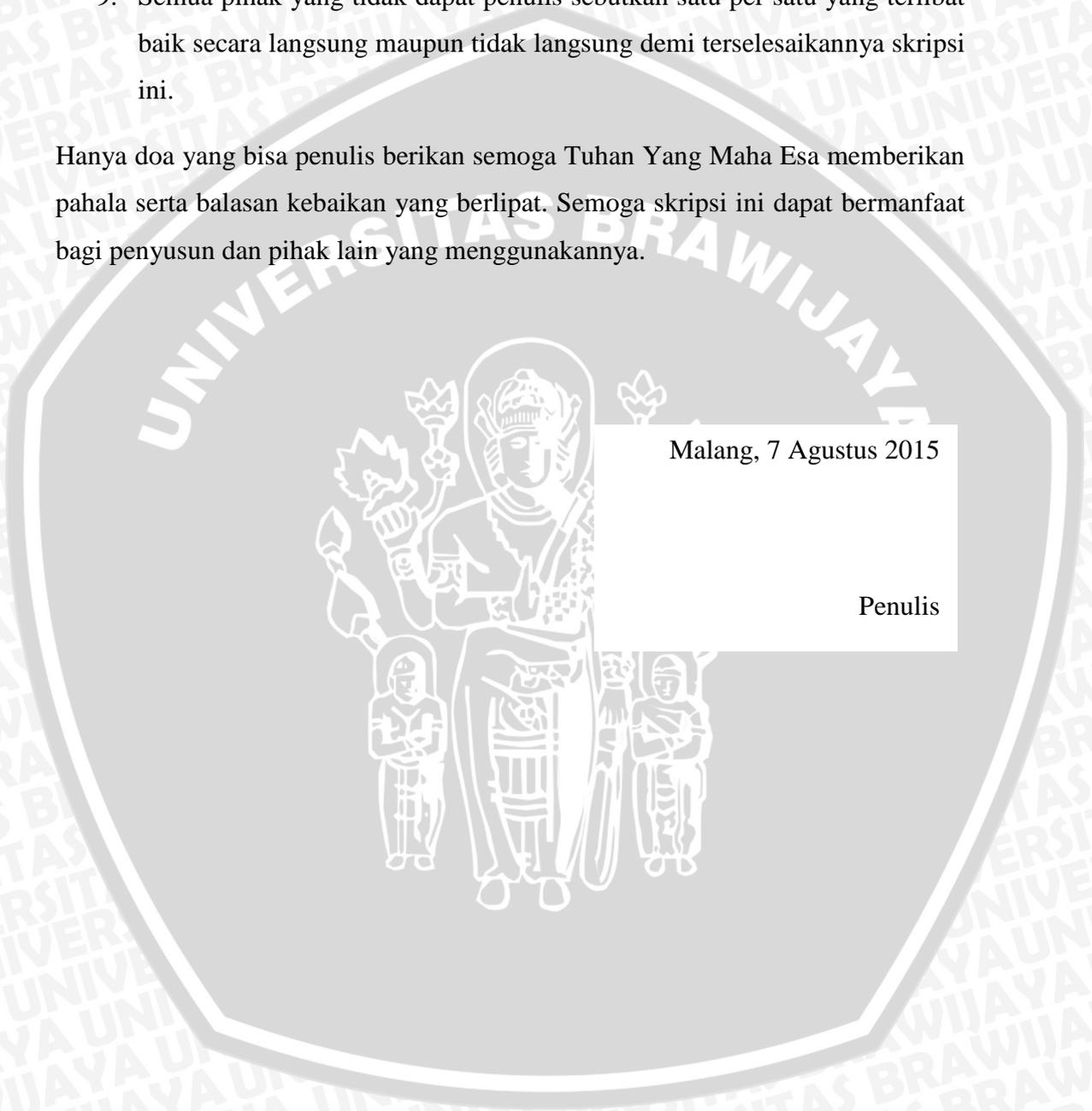
Dengan mengucapkan puji dan syukur kepada Tuhan Yang Maha Kuasa, karena atas segala rahmat kasih dan karunia-Nya-lah penulis mampu menempuh dan menyelesaikan skripsi dengan judul: “Penyelesaian Vehicle Routing Problem with Time Windows (VRPTW) Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)”.

Rasa syukur dan terima kasih bahwa beberapa kendala dan hambatan yang dijumpai dalam penulisan skripsi ini telah dapat diatasi baik, disamping itu penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna dan masih banyak kekurangan-kekurangan lainnya, maka dari itu saran dan kritik yang membangun dari semua pihak akan menjadi masukan yang sangat diharapkan. Penulis mengucapkan terima kasih kepada :

1. Imam Cholissodin, S.Si., M.Kom selaku dosen pembimbing skripsi pertama yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
2. Edy Santoso, S.Si., M.Kom selaku dosen pembimbing skripsi kedua yang telah meluangkan waktu dan juga memberikan pengarahan bagi penulis.
3. Ir. Sutrisno, MT., selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Drs. Marji, MT., selaku Ketua Program Studi Informatika/Ilmu Komputer Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak I Wayan Sukadana dan Ibu Ni Made Darmini selaku orang tua yang selalu memberikan semangat dan doa hingga terselesaikannya skripsi ini.
6. Seluruh Civitas Akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian skripsi ini.
7. Teman-teman dan sahabat-sahabat Informatika/Ilmu Komputer angkatan 2011 khususnya dari kelas TIF E. Teman-teman GGS tercinta, Sena, Rivi, Wina, Made Erwin dan Bagus. Serta Frida Baskarani atas bantuan, semangat dan dukungannya ketika penulis mengalami kesulitan.

8. Keluarga besar UNIKAHIDHA Universitas Brawijaya, Tim Asisten Laboratorium Sistem Komputer dan Robotika, HMIF dan kontrakan candi mendut, yang telah memberikan banyak pengalaman yang berharga selama menjalani studi di Universitas Brawijaya.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Tuhan Yang Maha Esa memberikan pahala serta balasan kebaikan yang berlipat. Semoga skripsi ini dapat bermanfaat bagi penyusun dan pihak lain yang menggunakannya.



Malang, 7 Agustus 2015

Penulis

ABSTRAK

I Made Riko Adi Saputra. 2015 : Penyelesaian *Vehicle Routing Problem with Time Windows* (VRPTW) Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)

Dosen Pembimbing: Imam Cholissodin, S.Si., M.Kom dan Edy Santoso, S.Si., M.Kom

Bali merupakan daerah tujuan wisata yang memiliki banyak lokasi wisata yang indah dan beranekaragam. Banyak pilihan wisata tersedia, namun jarak tempuh selalu menjadi pertimbangan yang utama. Apabila jarak tempuh yang dipilih salah atau tidak diperhitungkan, maka waktu perjalanan akan menjadi lebih lama. Semakin lamanya waktu dalam perjalanan, maka akan menambah biaya yang dihabiskan. Oleh karena itu perlu dibuat suatu perhitungan untuk mengoptimalkan penentuan rute wisata. Penentuan rute ini dapat dimodelkan menjadi permasalahan *Vehicle Routing Problem with Time windows*. Salah satu metode yang dapat digunakan untuk menyelesaikan masalah ini adalah Algoritma Genetika. Algoritma genetika dapat digunakan untuk menyelesaikan bentuk-bentuk persamaan matematika yang kompleks dan sulit dibangun. Dari hasil pengujian solusi optimal diperoleh pada nilai *fitness* yang dihasilkan setelah terjadinya konvergensi. Solusi optimal diperoleh pada nilai *fitness* yang dihasilkan setelah terjadinya konvergensi. Nilai optimal yang dihasilkan dari penelitian ini terdapat pada populasi: 30, generasi: 250, kombinasi Cr: 0.6, dan Mr: 0.4 dengan rata-rata *fitness* 0.00307 pada panjang kromosom 5. Nilai efisiensi dari algoritma genetika adalah 16.22%. Kondisi ini menunjukkan algoritma genetika menghasilkan solusi yang lebih baik dan dapat menyelesaikan permasalahan VRPPTW dengan studi kasus pariwisata Bali.

Kata kunci: Pariwisata, penentuan rute, *Vehicle Routing Problem with Time windows*, algoritma genetika, *fitness*, optimal.

ABSTRACT

I Made Riko Adi Saputra. 2015 : *The Solution of Vehicle Routing Problem with Time Windows (VRPTW) Using Genetic Algorithm (Case Study: Tourism Bali)*

Advisor: Imam Cholissodin, S.Si., M.Kom and Edy Santoso, S.Si., M.Kom

Bali is one of tourist destination that has many wonderful tourist sites and diverse. Many tour options available, but the distance has always been a major consideration. If the mileage is chosen incorrectly or not taken into account, then the travel time will be longer. The length of time in transit, it will add to the cost spent. It is therefore necessary to make a calculation to optimize the travel route determination. The determination of these problems can be modeled into a Vehicle Routing Problem with Time Windows. One method that can be used to solve this problem is a Genetic Algorithm. Genetic algorithms can be used to solve a complex mathematical equations forms and difficult to build. From the test results the optimal solution obtained at the fitness value generated after the occurrence of convergence. Optimal solution obtained at the fitness value generated after the occurrence of convergence. The optimal value resulting from this study are the population: 30, generation: 250, a combination of Cr: 0.6, and Mr: 0.4 with average fitness 0.00307 on the length of chromosome is 5 Genetic algorithms generate the efficiency of 16.22%. This condition shows the genetic algorithm produces better solutions and can solve the problems VRPPTW with case studies of tourism in Bali.

Keywords: *Tourism, route determination, Vehicle Routing Problem with Time windows, genetic algorithm, fitness, optimal*

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN.....	iii
ORISINALITAS SKRIPSI	iii
KATA PENGANTAR	iv
ABSTRAK	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR SOURCE CODE	xiii
BAB I.....	1
PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan masalah.....	3
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	4
1.5. Manfaat.....	4
1.6. Sistematika penulisan.....	5
BAB II.....	7
KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1. Kajian Pustaka	7
2.2. Pariwisata	8
2.3. Pemilihan Rute	8
2.4. Vehicle Routing Problem	8
2.5. Algoritma Genetika	10
2.5.1. Komponen-komponen Algoritma Genetika.....	12
2.6. Kecepatan Rata-Rata Kendaraan di Bali	16
BAB III	18
METODOLOGI DAN PERANCANGAN SISTEM.....	18
3.1. Metode Penelitian.....	18
3.2. Analisa Permasalahan.....	19



3.3.	Pengumpulan Data	20
3.3.1.	Studi literatur	20
3.3.2.	Dialog diskusi dan konsultasi	20
3.3.3.	Observasi	21
3.4.	Perancangan sistem	21
3.5.	Proses Algoritma Genetika.....	23
3.5.1.	Membuat Populasi Awal.....	24
3.5.2.	Proses <i>Crossover</i> (Pindah Silang).....	26
3.5.3.	Mutasi	28
3.5.4.	Menghitung nilai <i>Fitness</i>	30
3.5.5.	Seleksi.....	32
3.6.	Perhitungan Manual	33
3.6.1.	Membuat populasi awal.....	34
3.6.2.	Menghitung Nilai <i>Fitness</i>	35
3.6.3.	<i>Crossover</i>	38
3.6.4.	Mutasi	39
3.6.5.	Seleksi.....	39
3.7.	Perancangan <i>User interface</i>	41
3.8.	Perancangan Uji Coba dan Evaluasi.....	43
3.8.1.	Uji Coba Banyaknya Generasi.....	43
3.8.2.	Uji Coba Banyaknya Populasi	44
3.8.3.	Uji Coba Kombinasi <i>Crossover Rate</i> dan <i>Mutation Rate</i>	45
3.8.4.	Uji Coba Panjang Kromosom.....	46
3.8.5.	Uji Coba Perbandingan Algoritma Genetika dan Algoritma <i>Greedy</i>	46
BAB IV	47
IMPLEMENTASI	47
4.1.	Lingkungan Implementasi	47
4.1.1.	Lingkungan Perangkat Keras	47
4.1.2.	Lingkungan Perangkat Lunak.....	47
4.2.	Implementasi Program	47
4.2.1.	Membangkitkan Populasi Awal.....	48
4.2.2.	Proses <i>Crossover</i>	49
4.2.3.	Proses Mutasi.....	50
4.2.4.	Perhitungan <i>Fitness</i>	50

4.2.5. Proses Seleksi	52
4.3. Implementasi User interface.....	53
BAB V.....	57
PENGUJIAN DAN ANALISA.....	57
5.1. Pengujian dan Analisa Banyaknya Generasi.....	57
5.2. Pengujian dan Analisa Ukuran Populasi	58
5.3. Pengujian dan Analisa Kombinasi <i>Cr</i> dan <i>Mr</i>	61
5.4. Pengujian dan Analisa Panjang Kromosom	62
5.5. Pengujian dan Analisa Algoritma Genetika dengan <i>Greedy</i>	64
BAB VI.....	66
KESIMPULAN DAN SARAN.....	66
6.1. Kesimpulan.....	66
6.2. Saran.....	66
Daftar Pustaka	68
Lampiran	70



DAFTAR GAMBAR

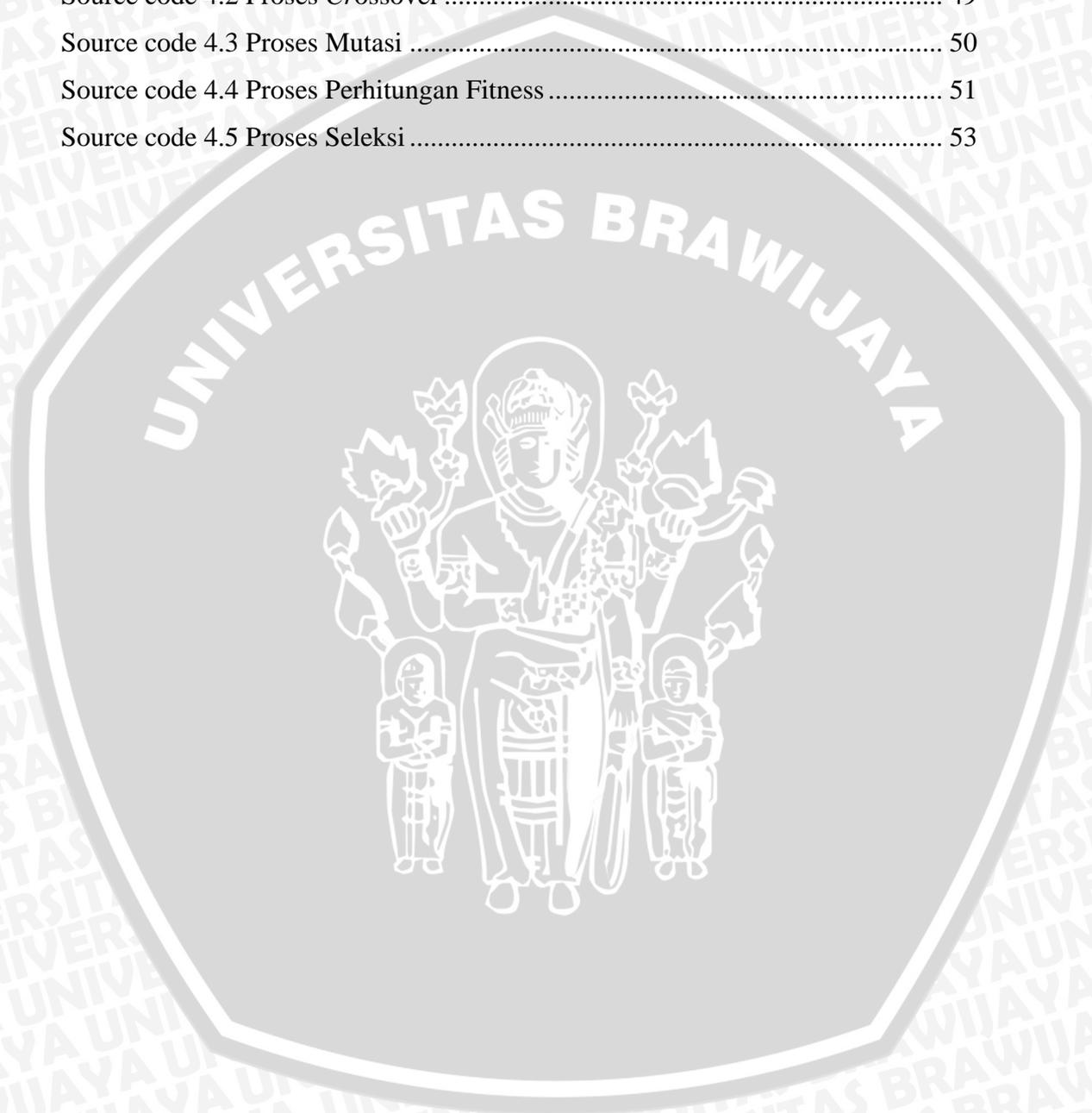
Gambar 2.1 Binary Encoding.....	12
Gambar 2.2 Permutation Encoding.....	12
Gambar 2.3 Value Encoding.....	13
Gambar 2.4 Tree Encoding.....	13
Gambar 2.5 Ilustrasi Exchange Mutation.....	16
Gambar 3.1 Metode Penelitian.....	18
Gambar 3.2 Flowchart Algoritma Genetika Pada Sistem.....	22
Gambar 3.3 Flowchart Membangkitkan Populasi Awal.....	25
Gambar 3.4 Proses Crossover.....	27
Gambar 3.5 Proses Mutasi.....	29
Gambar 3.6 Proses Perhitungan Fitness.....	31
Gambar 3.7 Proses Seleksi.....	32
Gambar 3.8 Proses Crossover.....	38
Gambar 3.9 Proses Mutasi.....	39
Gambar 3.10 Halaman Utama Perancangan User interface.....	41
Gambar 3.11 Halaman Kedua Perancangan User interface.....	42
Gambar 4.1 Implementasi User interface Halaman Pertama.....	53
Gambar 4.2 Tampilan Halaman Pertama Setelah Load Data.....	54
Gambar 4.3 Halaman Perhitungan.....	54
Gambar 4.4 Kotak Dialog Menentukan Krosomom.....	55
Gambar 4.5 Kotak Dialog Menentukan Parameter Crossover dan Mutasi.....	55
Gambar 4.6 Tampilan Detail Hasil Perhitungan.....	56
Gambar 5.1 Grafik Hasil Uji Coba Banyak Generasi.....	60
Gambar 5.2 Grafik Uji Coba Banyak Populasi.....	61
Gambar 5.3 Grafik Waktu Komputasi.....	62
Gambar 5.4 Grafik Hasil Uji Coba Kombinasi Cr dan Mr.....	64
Gambar 5.5 Grafik Hasil Uji Coba Panjang Kromosom.....	65

DAFTAR TABEL

Tabel 3.1 Data Objek Wisata	33
Tabel 3.2 Data Jarak Objek Wisata.....	33
Tabel 3.3 Data Waktu Tempuh Objek Wisata	34
Tabel 3.4 Populasi Awal	34
Tabel 3.5 Perhitungan Pinalti P1.....	35
Tabel 3.6 Perhitungan Pinalti P2.....	36
Tabel 3.7 Perhitungan Pinalti P3.....	36
Tabel 3.8 Perhitungan Fitness.....	37
Tabel 3.9 Perhitungan Seleksi.....	39
Tabel 3.10 Perhitungan Roulette Wheel	40
Tabel 3.11 Skenario Uji Coba Banyak Generasi	43
Tabel 3.12 Skenario Uji Coba Banyaknya Populasi	44
Tabel 3.13 Skenario Uji Coba Nilai Kombinasi Cr dan Mr.....	45
Tabel 3.14 Skenario Uji Coba Panjang Kromosom.....	46
Tabel 5.1 Hasil Uji Coba Banyak Generasi	59
Tabel 5.2 Hasil Uji Coba Banyak Populasi	61
Tabel 5.3 Hasil Uji Coba Kombinasi Cr dan Mr.....	63
Tabel 5.4 Hasil Uji Coba Panjang Kromosom.....	65
Tabel 5.5 Hasil Uji Coba Perbandingan Genetika dan <i>Greedy</i>	66

DAFTAR SOURCE CODE

Source code 4.1 Membangkitkan Kromosom Awal	48
Source code 4.2 Proses <i>Crossover</i>	49
Source code 4.3 Proses <i>Mutasi</i>	50
Source code 4.4 Proses <i>Perhitungan Fitness</i>	51
Source code 4.5 Proses <i>Seleksi</i>	53



BAB I

PENDAHULUAN

1.1. Latar Belakang

Salah satu kegiatan yang sangat digemari masyarakat Indonesia adalah berwisata. Statistik jumlah perjalanan wisatawan nusantara menyebutkan setiap tahunnya selalu mengalami peningkatan. Pada tahun 2013 terdapat 250.036.000 perjalanan wisatawan nusantara [1]. Berwisata merupakan salah satu kebutuhan, hal ini membuat wisatawan berusaha mencari tempat-tempat menarik dan terjangkau untuk dikunjungi. Bali merupakan daerah tujuan wisata yang memiliki banyak lokasi wisata yang indah dan beranekaragam. Jumlah wisatawan yang berkunjung ke Bali pada tahun 2014 adalah 6.392.460 wisatawan domestik dan 3.278.598 wisatawan mancanegara [2].

Para wisatawan terkadang bingung menentukan lokasi wisata yang ada di Bali. Setelah menentukan objek wisata, mereka harus menentukan obyek wisata mana yang harus dikunjungi terlebih dahulu. Banyak pilihan wisata tersedia, namun jarak tempuh selalu menjadi pertimbangan yang utama. Apabila jarak tempuh yang dipilih salah atau tidak diperhitungkan, maka waktu perjalanan akan menjadi lebih lama. Semakin lamanya waktu dalam perjalanan, maka akan menambah biaya yang dihabiskan. Selain itu waktu kunjungan yang sesuai juga menjadi pertimbangan, karena dapat mempengaruhi kenyamanan saat melakukan kunjungan wisata. Beberapa teknologi telah tersedia misalnya GPS. Walaupun GPS mampu menentukan rute dari satu tempat ke satu tempat yang lainnya, namun alat tersebut belum mampu menentukan waktu kunjungan yang sesuai bagi para wisatawan.

Dari uraian permasalahan tersebut, maka akan memunculkan pilihan alternatif solusi yang sesuai dengan kebutuhan dan kondisi wisatawan. Masalah seperti ini merupakan permasalahan umum tentang *Vehicle Routing Problem* (VRP). VRP merupakan permasalahan rute minimum dengan kendaraan untuk melayani beberapa pelanggan dengan kebutuhan tertentu. Dimana awal dan akhir keberangkatan kendaraan berada pada satu titik yang sama, yang harus dirancang sedemikian rupa sehingga setiap pelanggan dilayani hanya sekali dan hanya dengan satu kendaraan [3].

VRP memiliki tujuan untuk meminimalkan biaya yang direpresentasikan oleh total jarak tempuh dan jumlah kendaraan yang digunakan atau pun total jarak tempuh saja atau jumlah kendaraan yang digunakan saja. Penelitian tentang VRP telah banyak dilakukan, dimulai sejak akhir tahun 50-an setelah adanya penjelasan masalah tersebut merupakan generalisasi dari *Travelling Salesman Problem* (TSP) oleh Dantzig dan Ramser[4]. Masalah VRP kemudian dikembangkan menjadi berbagai macam diantaranya *Vehicle Routing Problem with Pick-Up and Delivery* (VRPPD), *Capacitated Vehicle Routing Problem* (CVRP), *Dynamic Vehicle Routing Problem* (DVRP), dan *Vehicle Routing Problem with Time Windows* (VRPTW).

Dalam pemilihan rute transportasi dengan hasil optimum dipengaruhi oleh jarak dan waktu kunjungan yang telah ditentukan atau *time window* dari objek wisata yang akan dikunjungi. Masing-masing objek wisata mempunyai waktu kunjungan ideal yang telah ditetapkan. Apabila kedatangan wisatawan dilakukan sebelum *time window* dari idealnya waktu kunjungan, maka akan dikenakan waktu tunggu. Sebaliknya apabila kedatangan wisatawan setelah *time window* dari idealnya waktu kunjungan, maka akan dikenakan waktu *penalty*.

Telah terdapat banyak penelitian mengenai pencarian rute transportasi optimum dengan kendala *time window* diantaranya, penyelesaian permasalahan *travelling salesman problem* dengan kendala *time window* pada perusahaan pengantar barang [5]. Permasalahan multi obyektif seperti uraian tersebut banyak diselesaikan dengan algoritma genetika. Perkembangan teknologi informasi yang sangat pesat membuat algoritma genetika semakin berkembang. Algoritma ini banyak digunakan dan telah terbukti menyelesaikan masalah optimasi dalam bidang fisika, biologi, ekonomi, sosiologi, industri, wisata dan lain-lain yang biasanya menggunakan model matematika yang kompleks atau bahkan sulit dibangun [6]. Contoh dalam bidang industri, algoritma genetika memberikan solusi yang lebih layak dan dapat mengurangi biaya dalam penyelesaian kasus pendistribusian logistik pada suatu perusahaan [7]. Algoritma genetika juga digunakan untuk menyelesaikan permasalahan *vehicle routing problem* dengan kendala *time windows* untuk kasus transportasi dengan *multi-depot*. Algoritma genetika memberikan solusi yang optimum dan layak untuk menyelesaikan permasalahan

tersebut [8]. Sehingga dengan menggunakan algoritma genetika diharapkan dapat memberikan solusi yang lebih optimum kepada para wisatawan dalam melakukan kunjungan wisata berdasarkan *time window*.

Berdasarkan latar belakang di atas dijelaskan bahwa algoritma genetika terbukti dapat digunakan untuk menyelesaikan masalah multi obyektif. Maka dalam tugas akhir ini akan dibangun suatu sistem cerdas untuk pencarian rute kunjungan wisata yang optimum dengan kendala *time window* berdasarkan urutan kunjungan ke masing-masing objek wisata. Untuk itu judul yang diambil pada tugas akhir ini adalah “**Penyelesaian Vehicle Routing Problem with Time Windows (VRPTW) Menggunakan Algoritma Genetika (Studi Kasus: Pariwisata Bali)**”.

1.2. Rumusan masalah

Adapun Rumusan Masalah yang ada pada penelitian ini berdasarkan latar belakang diatas adalah:

1. Bagaimana menerapkan algoritma genetika untuk menyelesaikan *Vehicle Routing Problem with Time Windows (VRPTW)* dengan studi kasus pariwisata Bali.
2. Bagaimana nilai evaluasi dari hasil implementasi algoritma genetika untuk menyelesaikan *Vehicle Routing Problem with Time Windows (VRPTW)* dengan studi kasus pariwisata Bali.

1.3. Batasan Masalah

Dalam menghindari pembahasan yang tidak terfokus pada tujuan maka penulis membuat batasan-batasan ruang lingkup sebagai berikut:

1. Perjalanan dimulai pukul 07.00 dan dilakukan dalam waktu satu hari dengan mengunjungi objek wisata tepat satu kali.
2. Susunan gen dalam kromosom diasumsikan sebagai objek wisata.
3. Data wisata didapat dari dinas pariwisata provinsi Bali dan data jarak didapat dari *Google Maps*.
4. Wisatawan mengunjungi objek wisata sesuai dengan waktu ideal kunjungan (*time-windows*) yang sudah ditentukan pada masing-masing objek wisata.

5. Kedatangan sebelum *time-windows* dikenakan waktu tunggu, sedangkan kedatangan setelah *time-windows* dikenakan waktu *penalty*.
6. Objek kendaraan wisata yang digunakan adalah mobil.
7. Kondisi jalan diasumsikan lancar pada dua arah dengan kecepatan rata-rata kendaraan di jalan raya dalam kota di Bali yaitu 40 km/jam.

1.4. Tujuan

Adapun tujuan yang ada pada penelitian ini dilihat dari rumusan masalahnya adalah:

1. Menerapkan algoritma genetika untuk menyelesaikan *Vehicle Routing Problem With Time Windows* (VRPTW) dengan studi kasus pariwisata Bali
2. Mengetahui nilai evaluasi dari hasil implementasi algoritma genetika untuk menyelesaikan *Vehicle Routing Problem with Time Windows* (VRPTW) dengan studi kasus pariwisata Bali.

1.5. Manfaat

Harapannya dalam penelitian ini dapat memberikan manfaat bagi semua pihak yang terkait. Adapun manfaat yang diperoleh dari pengimplementasian Algoritma genetika untuk menyelesaikan *Vehicle Routing Problem With Time Windows* ini adalah:

1. Memberikan solusi optimum bagi wisatawan dalam menentukan rute perjalanan, sehingga waktu yang tersedia dapat dimaksimalkan pada saat melakukan kunjungan wisata di Bali.
2. Meminimalkan biaya perjalanan pada saat melakukan kunjungan wisata di Bali.
3. Membantu wisatawan dalam menyesuaikan waktu kunjungan yang tepat ke beberapa tempat wisata di Bali, sehingga saat berkunjung menjadi lebih nyaman.
4. Mempercepat dalam menemukan solusi optimum bagi wisatawan secara efektif dan efisien untuk menyusun rencana kunjungan di Bali.

1.6. Sistematika penulisan

Dalam penulisan skripsi ini terdiri dari 6 bab dan beberapa sub bagian. Untuk memudahkan dalam penulisan dan pembahasan pada tiap-tiap bab agar sesuai dengan rumusan masalah yang ada maka digunakan sistematika penulisan seperti dibawah ini.

BAB 1: PENDAHULUAN

Didalam bab pendahuluan berisi pembahasan secara umum mengenai permasalahan yang akan diangkat mencakup latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB 2: KAJIAN PUSTAKA DAN DASAR TEORI

Didalam bab kajian pustaka dan dasar teori berisi mengenai penjelasan-penjelasan teori-teori dan konsep dasar yang berkaitan dengan Penyelesaian *Vehicle Routing Problem With Time Windows* (VRPTW) Menggunakan Algoritma Genetika.

BAB 3: METODOLOGI PENELITIAN DAN PERANCANGAN

Didalam bab metodologi penelitian dan perancangan berisi mengenai penjelasan metode yang digunakan untuk menyelesaikan permasalahan yang diangkat dalam penelitian skripsi ini.

BAB 4: IMPLEMENTASI

Didalam bab implementasi menguraikan tentang Gambaran serta proses implementasi metode Algoritma genetika untuk menyelesaikan *Vehicle Routing Problem With Time Windows* dengan studi kasus pariwisata Bali, dimana pada implementasi ini juga disertai dengan source code dan screenshot aplikasi yang dibuat.

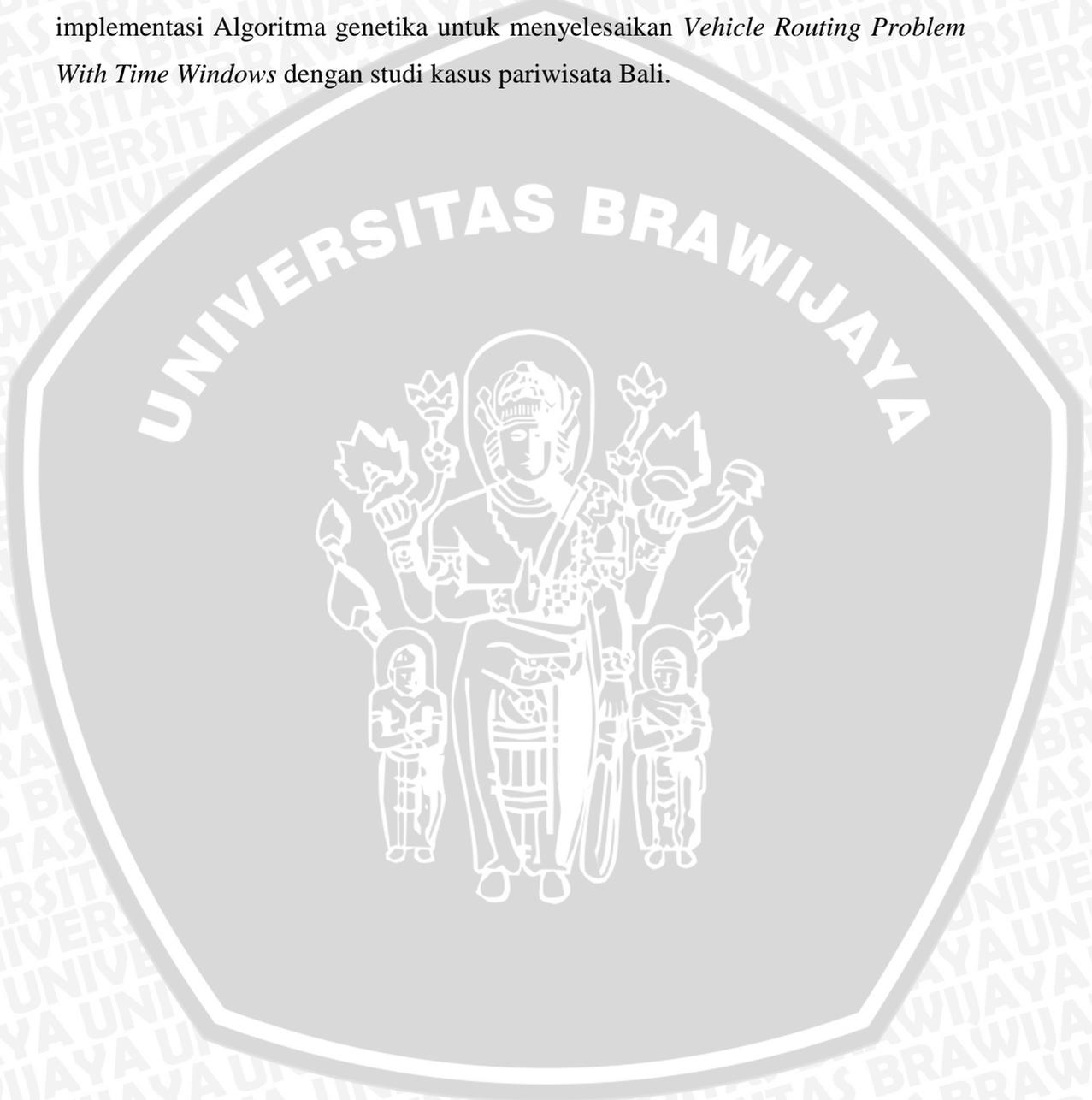
BAB 5: PENGUJIAN DAN ANALISIS

Didalam bab pengujian berisi mengenai hasil pengujian dari implementasi metode Algoritma genetika untuk menyelesaikan *Vehicle Routing Problem With Time Windows* dengan studi kasus pariwisata Bali yang telah di buat. Selanjutnya

menganalisa dari hasil pengujian tersebut untuk mengetahui tingkat akurasi penerapan metodenya.

BAB 6: PENUTUP

Didalam bab penutup berisi mengenai uraian tentang kesimpulan dan saran dari implementasi Algoritma genetika untuk menyelesaikan *Vehicle Routing Problem With Time Windows* dengan studi kasus pariwisata Bali.



BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini berisi kajian pustaka dan dasar teori yang berhubungan dengan penyelesaian *Vehicle Routing Problem with Time Windows* (VRPTW) menggunakan algoritma genetika pada studi kasus pariwisata Bali.

2.1. Kajian Pustaka

Dalam penulisan skripsi ini, dilakukan kajian terhadap penelitian-penelitian sebelumnya. Penelitian tersebut meliputi penyelesaian *vehicle routing problem* dan pemanfaatan algoritma genetika.

Pada penelitian yang dilakukan oleh Abdul Kadar tentang *vehicle routing problem* (VRP) dalam pelayanan pelanggan, dikemukakan bahwa kasus VRP merupakan permasalahan kombinasi yang kompleks. VRP sulit diselesaikan dengan metode yang tepat pada contoh kasus dengan data yang besar. Algoritma genetika merupakan salah satu teknik pencarian solusi yang mendekati optimum. Dalam kasus tersebut algoritma genetika dapat dibuktikan untuk menyelesaikan masalah VRP dengan waktu yang cepat untuk menghasilkan solusi yang mendekati optimum [8].

Kasus VRP lain dengan tambahan yaitu penelitian tentang pendistribusian logistic yang dilakukan oleh Chen dan Zhou. Dalam penelitian ini VRP dibuat lebih kompleks dengan menambahkan *time-windows*. Untuk menyelesaikan kasus ini digunakan algoritma genetika, karena algoritma genetika dapat digunakan menyelesaikan kasus multi objektif dengan persamaan matematika yang kompleks. Penelitian ini menunjukkan bahwa algoritma genetika memberikan hasil yang efektif dan mungkin untuk diterapkan [7].

Beberapa kelebihan yang dimiliki algoritma genetika yaitu, Algoritma genetik tidak memerlukan banyak perhitungan matematik dari masalah optimasi yang dihadapi. Algoritma genetik mampu menangani berbagai jenis fungsi objektif dengan batasan-batasan yang diberikan pada ruang pencarian yang diskrit, kontinyu, maupun campuran. Operasi evolusi seleksi yang berkerja secara

probabilistik terhadap beberapa kemungkinan solusi sekaligus membuat algoritma genetik sangat efektif dalam melakukan pencarian global. Algoritma genetika memiliki fleksibel yang cukup baik dan cukup memungkinkan untuk digabungkan dengan metode lain dan membuat implementasi yang lebih efisien dalam masalah tertentu [16].

2.2. Pariwisata

Menurut Undang-Undang No. 10 Tahun 2009 tentang Kepariwisataaan, Pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, pemerintah, dan pemerintah daerah. Pariwisata berfungsi untuk memenuhi kebutuhan jasmani, rohani dan intelektual setiap wisatawan dengan rekreasi dan perjalanan serta meningkatkan pendapatan negara untuk mewujudkan kesejahteraan rakyat. [9]

2.3. Pemilihan Rute

Pemilihan rute pada setiap proses pengantaran barang merupakan faktor penting yang perlu diperhatikan. Pada proses pengiriman barang diperlukan optimasi rute jarak tempuh sehingga akan meminimalkan waktu tempuh perjalanan dan meminimalkan biaya transportasi yang digunakan saat proses distribusi berlangsung. Proses distribusi dilakukan dalam sehari sesuai dengan jam kerja pelaksanaan pengiriman beras ke beberapa tujuan. Keberhasilan dalam suatu proses distribusi diukur dari optimasi waktu jarak tempuh, waktu pembongkaran barang, waktu tunggu buka pelayanan dengan kemungkinan akan mengakibatkan keterlambatan kendaraan kembali. Agar proses distribusi tetap berjalan lancar perlu adanya solusi untuk optimasi beberapa faktor yang dapat mempengaruhi penggunaan waktu dan biaya yang lebih. Pemilihan rute dengan perhitungan jarak tempuh merupakan faktor utama yang mempengaruhi penggunaan waktu dan biaya dalam proses distribusi. Dengan menghitung jarak tempuh kemudian memutuskan urutan tujuan sesuai rute optimum yang terpilih maka dapat mempengaruhi waktu minimum [10].

2.4. Vehicle Routing Problem

Pada tahun 1959, Dantzing dan Ramser menemukan *Vehicle Routing Problem* (VRP) pertama kali. VRP merupakan program non-linier untuk pencarian

sebuah solusi dalam memecahkan suatu masalah. Bentuk permasalahan VRP adalah salah satu masalah optimasi kombinatorial yang paling banyak dipelajari dan memperhatikan Gambaran rute optimal yang dapat dilalui kendaraan untuk melayani pelanggan. Setiap kendaraan memiliki kapasitas angkut dan setiap pelanggan memiliki permintaan (*demand*). Setiap pelanggan dikunjungi kendaraan tepat satu kali dengan total *demand* tiap rute tidak boleh melebihi kapasitas angkut kendaraan. Dalam VRP kendaraan harus berangkat dan kembali pada tempat distribusi yang sama, tempat tersebut biasa dikenal dengan istilah depot. Faktor yang sering muncul pada VRP adalah masalah kapasitas yang dikenal dengan nama *Capacitated Vehicle Routing Problem* (CVRP) serta permasalahan pada batasan waktu yang dikenal dengan nama *Vehicle Routing Problem with Time Windows* (VRPTW). Kedua permasalahan ini dapat digabungkan dengan prioritas utama yaitu semua permintaan terpenuhi ada batas waktu tertentu [4].

2.4.1. Vehicle Routing Problem With Time Windows

Vehicle Routing Problem with Time Windows (VRPTW) merupakan bentuk metode perluasan dari VRP dalam kendala penentuan kapasitas sesuai kebutuhan yang terkait dengan kendala pengantaran pada masing-masing pelanggan yang memiliki interval waktu atau biasa disebut *time windows*. Pada VRPTW permasalahan yang dapat dipecahkan adalah masalah kapasitas angkut kendaraan dalam *timeframe* dan kendaraan harus melayani setiap konsumen pada *timeframe* tertentu. VRPTW dapat digambarkan sebagai masalah perancangan optimasi rute dengan pengeluaran biaya yang minimal dari suatu depot beberapa titik tujuan. Rute harus dirancang sesuai perhitungan jarak ke setiap tujuan agar setiap titik hanya dikunjungi sekali saja oleh kendaraan. Semua rute yang dilalui harus berangkat dan berakhir di depot yang sama dengan total *demand* semua titik pada rute tersebut tidak boleh melebihi kapasitas angkut kendaraan. VRPTW dibagi menjadi dua bagian yaitu *Vehicle Routing Problem with Hard Time Windows* (VRPHTW) dan *Vehicle Routing Problem with Soft Time Windows* (VRPSTW). Dimana pada VRPHTW konsumen hanya dapat dilayani selama selang waktu yang telah ditentukan, sedangkan pada VRPSTW konsumen dapat dilayani setiap saat dengan [11].

2.5. Algoritma Genetika

Alogaritma genetika merupakan alogaritma pencarian yang berdasarkan ilmu genetika dan seleksi alam pada mekanisme alam. Pada aplikasi Genetika, variabel solusi tersebut dibuat kode dalam bentuk *string* yang mempresentasikan barisan-barisan gen, yang merupakan karakteristik dari solusi permasalahan.

Umumnya Alogaritma genetika berawal dari himpunan solusi yang dihasilkan secara acak, dari himpunan acak tersebut dihasilkan sebuah populasi. Didalam populasi terdapat Individu yang disebut *chromosome*, *chromosome* merupakan representasi dari solusi tersebut. *Chromosome* tersebut berevolusi dalam proses Iterasi yang berkelanjutan, sehingga dihasilkan generasi. Setiap generasi, *chromosome* akan dievaluasi berdasarkan suatu fungsi evaluasi. Setelah dihasilkan beberapa generasi maka alogaritma genetika akan konvergen pada *chromosome* terbaik yang merupakan solusi optimal [12].

Alogaritma genetika banyak digunakan untuk aplikasi bisnis teknik, teknik serta keilmuan lainnya. Alogaritma ini dipakai untuk mendapatkan solusi yang tepat untuk masalah optimal dari satu variable maupun multi variable. Sebelum alogaritma ini diterapkan, harus dinyatakan dalam sebuah fungsi tujuan terlebih dahulu, yang disebut dengan fungsi *fitness*. Jika nilai yang didapatkan semakin besar, maka solusi yang dihasilkan semakin baik. Meskipun awalnya semua nilai *fitness* yang dihasilkan kemungkinan sangat kecil, hal ini bisa terjadi karena alogaritma genetika menghasilkan nilai secara random, namun sebagian nilai *fitness* tersebut lebih tinggi dari nilai yang lain. *Chromosome* dengan nilai *fitness* tinggi akan memberikan nilai probabilitas yang cukup tinggi untuk melakukan reproduksi serta menyeleksi individu pada generasi selanjutnya. Alogaritma genetika sangat tepat digunakan untuk menyelesaikan masalah optimasi yang kompleks dan sulit dipecahkan dalam matematika [6].

Terdapat dua variasi dalam alogaritma genetika yaitu Steady State dan Generational Replacement. Alogaritma genetika steady state untuk proses replacement dilakukan setiap kali telah dilakukan dua *offspring* dari hasil *crossover*. *Offspring* ini akan menggantikan *chromosome* yang memiliki nilai *fitness* terkecil, sehingga populasi baru akan memiliki individu-individu yang lebih baik dibandingkan dengan populasi lama. Sedangkan pada jenis *generational*

replacement, proses *replacement* dilakukan pada saat dihasilkan satu populasi baru [13].

Adapun ciri-ciri bahwa suatu permasalahan dapat diselesaikan dengan menggunakan algoritma genetika adalah [6]:

- Menggunakan fungsi tujuan optimasi non linear dengan banyak kendala yang juga dalam bentuk non linier
- Mempunyai kemungkinan solusi yang jumlahnya tak terhingga.
- Membutuhkan solusi “*real-time*” dalam arti solusi tersebut bisa didapatkan dengan cepat sehingga dapat diimplementasikan untuk permasalahan yang mempunyai perubahan yang cepat.
- Mempunyai *multi-objective* dan *multi-criteria*, sehingga diperlukan solusi yang dapat diterima oleh semua pihak.

Secara umum Algoritma genetika dapat dijelaskan dengan algoritma berikut[13]:

1. [Mulai] Membangun populasi awal secara random sebanyak n *chromosome* (sesuai dengan permasalahannya)
2. [Fitness] Evaluasi setiap *fitness* $f(x)$ dari setiap *chromosome* x pada populasi
3. [Populasi baru] membuat populasi baru hingga didapatkan populasi baru yang lengkap dengan melakukan langkah-langkah berikut
 - a. [Seleksi] Pilih dua *chromosome* induk dari populasi berdasarkan *fitness*-nya (semakin besar *fitness*nya semakin besar kemungkinan untuk dipilih sebagai induk)
 - b. [Crossover] Sesuai dengan besarnya kemungkinan *crossover*, induk yang dipilih disilangkan untuk membentuk anak. Jika tidak ada *crossover*, maka anak merupakan Salinan dari induknya.
 - c. [Mutasi] Sesuai dengan besarnya kemungkinan mutasi, anak akan dimutasi pada setiap posisi pada *chromosome*
 - d. [Penerimaan] Penempatan anak baru pada populasi baru
4. [Ganti] Menggunakan populasi yang baru dibentuk untuk proses algoritma selanjutnya
5. [Tes] Jika kondisi akhir terpenuhi, berhenti, dan hasilnya merupakan solusi terbaik dari populasi saat itu.

6. [Ulangi] Kembali ke nomer 2

2.5.1. Komponen-komponen Algoritma Genetika

Pada dasarnya algoritma genetika memiliki 7 komponen, antara lain [14]:

a. Pendefinisian Individu

Pendefinisian individu adalah langkah pertama yang harus dilakukan dalam GA. menyatakan salah satu solusi yang mungkin dari suatu permasalahan yang diangkat. Pendefinisian individu dilakukan dengan mendefinisikan jumlah dan tipe dari gen yang digunakan dan tentunya dapat mewakili solusi permasalahan yang diangkat. Sebuah kromosom dibentuk dari komponen-komponen penyusun yang disebut sebagai gen dan nilainya dapat berupa bilangan numerik, biner, simbol ataupun karakter tergantung dari permasalahan yang ingin diselesaikan. Menurut Marek Obitko terdapat beberapa macam teknik pengkodean antara lain [14]:

1. *Binary Encoding* (Pengkodean Biner)

Pada teknik ini, setiap kromosom terdiri dari susunan bilangan biner, sehingga setiap kromosom memiliki 2 nilai yaitu 0 dan 1. Contohnya pada Gambar 2.1:

Kromosom 1	110011010101010001
Kromosom 2	101010010101010011

Gambar 2.0.1 *Binary Encoding*

2. *Permutation Encoding* (Pengkodean Permutasi)

Pada teknik ini, setiap kromosom tersusun dari deretan angka yang mendefinisikan posisi dalam suatu urutan. Nilai dalam satu deret kromosom tidak boleh sama satu dengan yang lain. Teknik ini banyak digunakan untuk penyelesaian masalah TSP. Contoh pada Gambar 2.2.

Kromosom 1	13245678
Kromosom 2	54321678

Gambar 2.0.2 *Permutation Encoding*

Pada Tabel di atas diasumsikan seorang sales harus mengantar barang melewati titik-titik tertentu, dengan syarat tidak boleh melewati titik yang sama.

3. *Value Encoding* (Pengkodean Nilai)

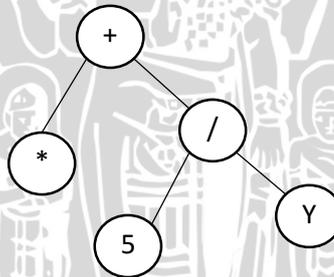
Pada teknik ini, setiap kromosom tersusun dari suatu nilai yang bisa berupa macam-macam nilai sesuai dengan permasalahan yang dihadapi. Seperti bilangan real, char dan lain-lain. Contohnya pada Gambar 2.3:

Kromosom 1	ABCDAECC
Kromosom 2	EEDEWWE

Gambar 2.0.3 *Value Encoding*

4. *Tree Encoding* (Pengkodean Pohon)

Pada teknik ini, tiap kromosom adalah pohon dari objek-objek seperti fungsi atau perintah dalam bahasa pemrograman. Contohnya: $(+ x (/5 y))$ dapat dilihat pada Gambar 2.4



Gambar 2.0.4 *Tree Encoding*

b. Nilai *Fitness*

Fitness merupakan fungsi yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin di capai. Individu tersebut dievaluasi berdasarkan fungsi matematika sebagai ukuran performasinya. Dalam evolusi natural individu dengan nilai *fitness* tinggi akan bertahan hidup, sedangkan untuk individu yang mempunyai nilai *fitness* rendah akan mati [8].

Dalam permasalahan optimasi, fungsi *fitness* dapat berupa solusi dengan memaksimalkan atau meminimalkan suatu fungsi misalnya x , kedua solusi tersebut adalah sama dan bersifat kebalikan. Akan tetapi untuk permasalahan dengan solusi meminimalkan fungsi x , tidak bisa menggunakan fungsi x secara langsung sebagai fungsi *fitness*, sebab hal tersebut bertentangan dengan konsep evolusi. Dimana individu yang mampu bertahan hidup adalah individu yang memiliki nilai *fitness* tinggi. Persamaan yang digunakan untuk mencari fungsi *fitness* [14].

$$fitness = \frac{1}{fx} \quad (2.1)$$

Dimana :

$$fx = \sum(\alpha D_{ij}) + \sum(\alpha P) \quad (2.2)$$

Dengan keterangan:

- fx adalah fungsi *fitness* individu x
- D_{ij} adalah jarak tempuh dari titik i ke titik j .
- P merupakan Penalty apabila customer dilayani diluar jadwal

c. Seleksi

Untuk melihat kualitas dari setiap kromosom dalam populasi, maka dilakukan proses seleksi. Hal ini ditujukan untuk menghasilkan ranking calon solusi. Seleksi bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling *fit*. Nilai *fitness* yang ada digunakan sebagai acuan dalam menentukan ranking dari individu, yang selanjutnya dipilih secara acak, kromosom yang mengalami rekombinasi. Kromosom dengan kualitas yang lebih baik akan memiliki peluang lebih besar untuk terpilih sebagai calon kromosom generasi berikutnya [15].

Metode yang digunakan pada proses seleksi adalah *roulette wheel*. Perhitungan nilai *fitness*, probabilitas, dan probabilitas kumulatif dinyatakan pada persamaan 2.3 dan persamaan 2.4 [14].

$$prob_k = \frac{fitness(P_k)}{total_fitness} \quad (2.3)$$

$$probCum_k = \sum_{j=1}^k prob_j \quad (2.4)$$

Dengan Keterangan :

1. $prob$ = Probabilitas individu
2. $fitness(P_k)$ = *Fitness* individu
3. $total_fitness$ = Total *fitness* dalam satu populasi
4. $probCum_k$ = Probabilitas kumulatif

d. Perkawinan Silang (*Crossover*)

Perkawinan silang atau *Crossover* merupakan bagian paling penting dari GA. *Crossover* bertujuan untuk menambah keanekaragaman kromosom pada generasi berikutnya berdasar kromosom generasi sebelumnya. *Crossover* dilakukan dengan melakukan pertukaran gen dari dua induk secara acak. Kromosom baru yang terbentuk akan mewarisi sebagian dari sifat kromosom induknya. Perkawinan Silang (*Crossover*) juga dapat berakibat buruk jika ukuran populasinya sangat kecil. Untuk mengatasi masalah ini digunakan suatu aturan bahwa perkawinan silang hanya bisa dilakukan dengan suatu probabilitas tertentu (Cr). Probabilitas *crossover* atau *crossover rate* (Cr) itu sendiri merupakan nilai perbandingan jumlah kromosom yang diharapkan akan mengalami perkawinan silang terhadap jumlah kromosom dalam suatu populasi [14].

e. Mutasi

Setelah mengalami proses *crossover*, pada *offspring* dapat dilakukan mutasi. Mutasi dilakukan dengan cara melakukan perubahan pada satu gen atau lebih dari individu. Namun tidak semua keturunan dari *crossover* mengalami mutasi. Banyaknya parent yang mengalami mutasi akan bergantung pada *mutation rate* (Mr) yang di tentukan. Parent yang tidak mengalami mutasi akan secara langsung disalin ke *chromosome* keturunan.

Beberapa operator mutasi yang dapat digunakan adalah penukaran gen pada *chromosome*, penukaran barisan gen pada *chromosome*, penggantian nilai gen dengan inversinya. Pemilihan operator yang digunakan bergantung pad acara pengkodean masalah. Operator mutasi yang dapat digunakan untuk masalah yang menggunakan pengkodean

dengan bilangan biner adalah metode mengganti nilai gen dengan inversinya. Sedangkan untuk masalah yang menggunakan pengkodean permutasi, operator mutasi yang dapat digunakan adalah menukar gen dan menukar barisan gen pada *chromosome* [14].

Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama seleksi serta menyediakan gen yang tidak ada dalam populasi awal. Sehingga mutasi akan meningkatkan variasi populasi. Mutasi sangat berperan penting ketika populasi awal hanya ada sedikit solusi yang mungkin terpilih. Sehingga operasi itu sangat berguna dalam keanekaragaman individu dalam populasi, meskipun dengan mutasi tidak dapat diketahui apa yang terjadi pada individu terbaru.

Salah satu metode mutasi yang dapat digunakan adalah *Reciprocal Exchange Mutation*. Metode ini bekerja dengan memilih dua porsi secara acak, kemudian menukar dua gen dalam posisi tersebut.

Pertukaran mutasi dapat dilakukan dengan cara:

1. Menentukan dua *site* secara random
2. *Site* pertama ditempatkan ke *site* kedua, untuk selanjutnya digeser ke kiri seperti pada Gambar 2.5 berikut.



Gambar 2.0.5 Ilustrasi *Exchange Mutation*

2.6. Kecepatan Rata-Rata Kendaraan di Bali

Bali memiliki beberapa jenis jalan diantaranya, jalan tol, jalan *by pass*, jalan provinsi, jalan kabupaten, jalan raya dalam kota dan jalan di pedesaan. Dari beberapa jenis jalan tersebut kecepatan kendaraan yang melintasi jalan tersebut berbeda beda. Berdasarkan pantauan *Area Traffic Control* Provinsi Bali, kecepatan rata-rata kendaraan yang melintas di jalan tol adalah 70 km/jam, kecepatan rata-rata di jalan *by pass* adalah 50 km/jam, dan di jalan raya dalam kota kecepatan rata-rata

kendaraan adalah 40 km/jam [17]. Untuk menghitung kecepatan dan waktu tempuh kendaraan digunakan rumus 2.5 dan rumus 2.6.

$$V = \frac{s}{t} \quad (2.5)$$

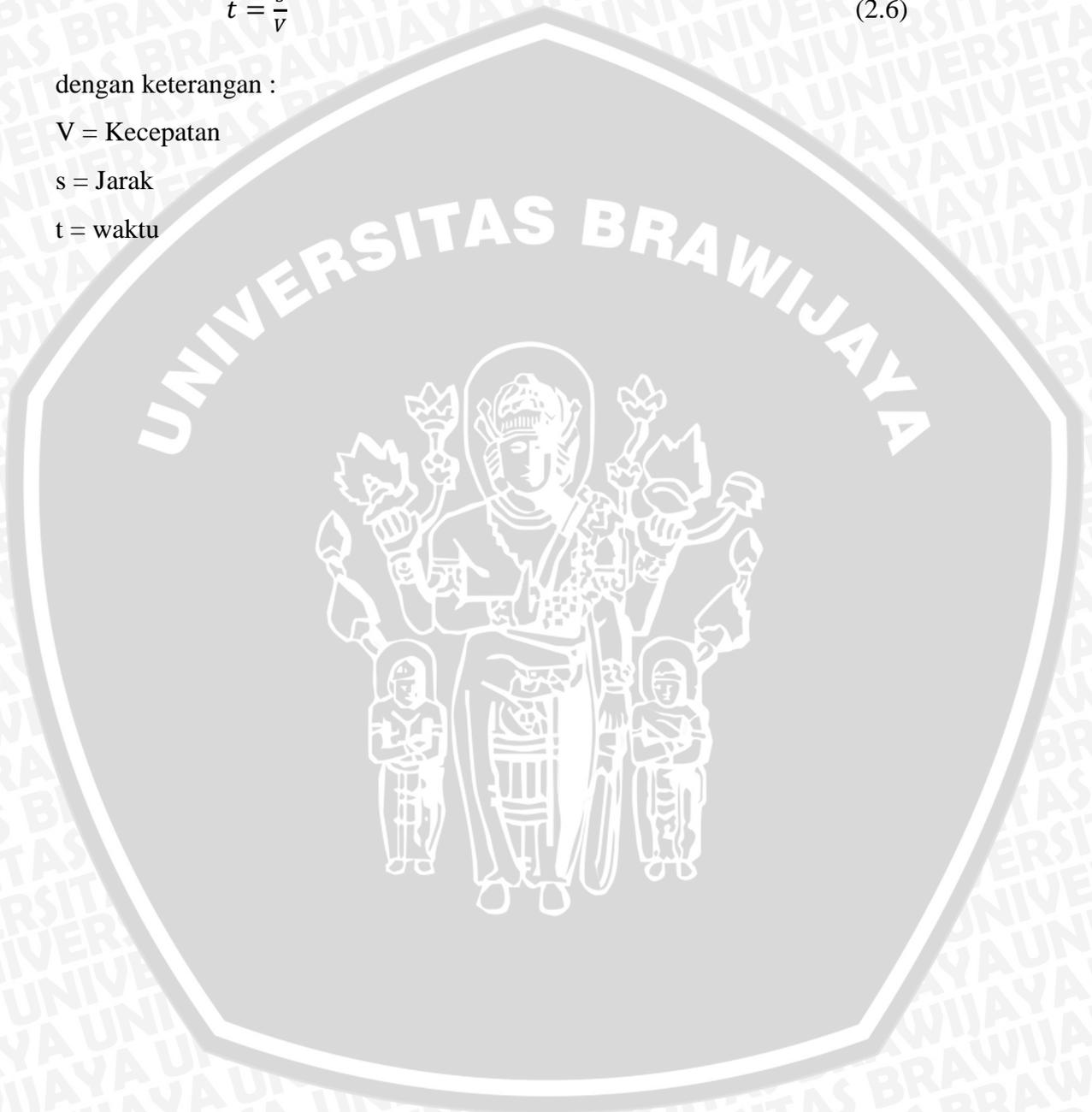
$$t = \frac{s}{v} \quad (2.6)$$

dengan keterangan :

V = Kecepatan

s = Jarak

t = waktu

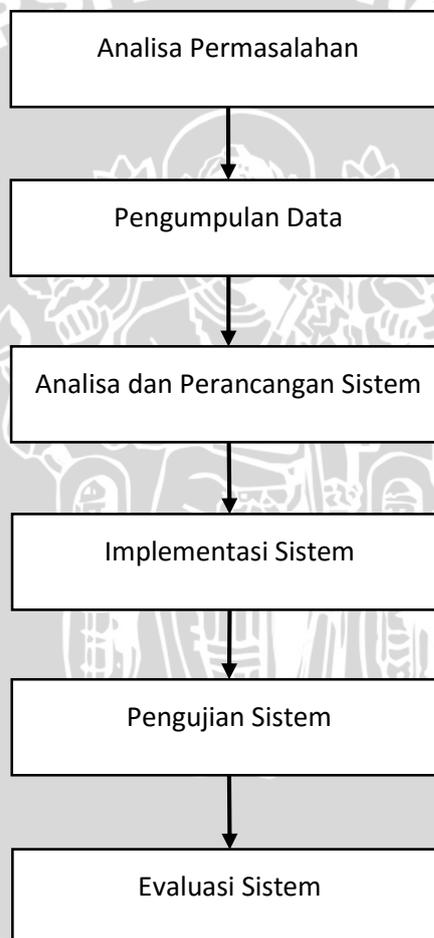


BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1. Metode Penelitian

Pada bab ini dibahas mengenai metode yang digunakan dalam penyelesaian *Vehicle Routing Problem With Time Windows (VRPTW)* dengan studi kasus pariwisata Bali. Metode penelitiannya dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metode Penelitian

Adapun tahapan tahapan dalam metode penelitiannya adalah sebagai berikut:

1. Analisa permasalahan yang akan dibahas dengan mendeskripsikan masalah yang akan diselesaikan

2. Pengumpulan data dilakukan secara sekunder, dimana data diambil dari beberapa sampel yang ada dilapangan. Untuk data jarak dibangkitkan secara random.
3. Analisa dan perancangan sistem berdasarkan deskripsi masalah.
4. Membuat sistem berdasarkan analisa permasalahan serta perancangan sistem yang telah dilakukan.
5. Melakukan uji coba terhadap sistem untuk mengetahui nilai parameter Algoritma genetika secara optimal untuk permasalahan *Vehicle Routing Problem*.
6. Melakukan evaluasi yang diperoleh dari uji coba sistem.

3.2. Analisa Permasalahan

Sistem penentuan rute kunjungan pariwisata di Bali saat ini masih menggunakan sistem manual untuk mencapai semua objek wisata tujuan yang telah ditentukan. Dalam penentuan rute tidak diketahui rute tersebut termasuk rute optimal atau tidak. Sehingga memakan banyak biaya dan juga waktu perjalanan.

Sistem yang akan dibangun ini merupakan pencarian rute optimum dengan kendala time window, dimana setiap objek wisata telah memiliki waktu kunjungan yang telah disesuaikan dengan kondisi objek wisata tersebut.

Sistem ini dibuat untuk memudahkan wisatawan dalam hal pengambilan keputusan untuk menentukan rute kunjungan objek wisata yang telah ditentukan. Dengan adanya sistem ini diharapkan dapat membantu wisatawan dalam memecahkan masalah optimasi penentuan rute dalam hal biaya perjalanan dan waktu, sehingga segala upaya yang akan dilakukan lebih efektif dan efisien. Dengan menggunakan *interface* yang dapat mempermudah pengguna karena akan ditampilkan cukup detail pada layar komputer dan pengguna dapat mengetahui lama perjalanan ke setiap objek wisata sehingga dapat sampai tujuan pada waktu yang telah ditentukan. Dalam hal ini pengguna adalah seorang wisatawan yang ingin melakukan kunjungan wisata ke Bali.

Algoritma yang digunakan untuk pembuatan aplikasi ini adalah algoritma genetika, dimana algoritma genetika dapat digunakan untuk menyelesaikan

masalah optimasi yang kompleks, menemukan solusi terbaik dalam pencarian rute yang memiliki waktu tempuh tercepat dan tepat untuk masing-masing tujuan. Dalam tugas ini akan dijelaskan mengenai tentang penerapan algoritma genetika untuk pencarian rute optimum kunjungan ke beberapa objek wisata dari pusat wisatawan berada ke setiap objek wisata tujuan dengan memperhatikan time windows. Untuk mencapai semua objek wisata tujuan, wisatawan akan mencari rute terbaik.

3.3. Pengumpulan Data

Pengumpulan data merupakan usaha awal yang dilakukan untuk memperoleh data dari beberapa sumber yang dibutuhkan untuk menyelesaikan tugas akhir ini. Kemudian data tersebut akan diproses sesuai dengan kebutuhan.

Data yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Data mengenai jarak tempuh antar objek wisata ditentukan menggunakan *Google Maps*.
2. Data spesifikasi objek wisata yang terdiri dari waktu kunjungan yang sesuai dan akhir waktu kunjungan sesuai dengan melakukan *survey* langsung ke lapangan dan informasi dari situs resmi objek wisata.
3. Data lama waktu kunjungan ditentukan dari *survey* di beberapa objek wisata.
4. Data yang dipakai adalah kunjungan ke suatu objek wisata dalam satu hari yang sama.

3.3.1. Studi literatur

Studi literatur adalah salah satu pengumpulan data yang diperoleh dengan mengumpulkan beberapa sumber pustaka berupa buku-buku, jurnal, laporan penelitian dan lain sebagainya untuk ditelaah lebih lanjut sebagai materi pendukung untuk tugas akhir ini

3.3.2. Dialog diskusi dan konsultasi

Dialog, diskusi serta konsultasi dengan pembimbing tugas akhir dan berdiskusi dengan rekan-rekan mahasiswa dengan tujuan untuk memperoleh solusi dalam hal pengambilan metode pengujian yang efektif.

3.3.3. Observasi

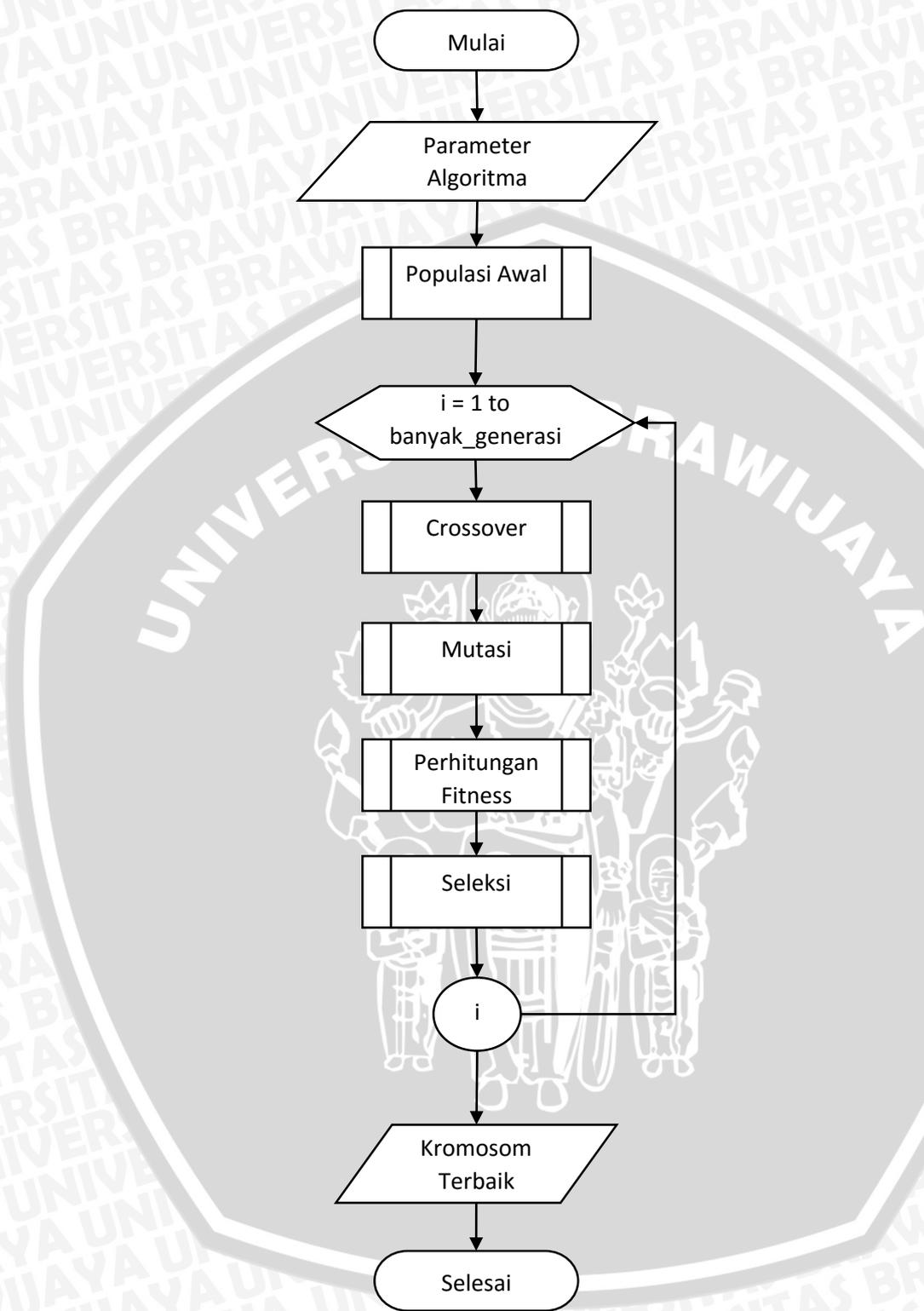
Observasi ini dilakukan dengan mencari data secara langsung dari beberapa narasumber untuk mendapatkan data yang sesuai agar bisa dilakukan penelitian lebih lanjut.

3.4. Perancangan sistem

Dari beberapa analisis yang telah dilakukan, subbab ini akan membahas mengenai proses algoritma genetika pada perangkat lunak ini. Proses yang akan dibahas dalam algoritma genetika terdiri dari beberapa fungsi yaitu fungsi representasi kromosom, *crossover*, mutasi, dan menghitung nilai *fitness*. Pada prosesnya terdapat beberapa parameter yang dibutuhkan dalam sistem ini yaitu:

1. Parameter Genetika yang terdiri dari:
 - Nilai ukuran populasi
 - Banyaknya Generasi
 - Probabilitas *Crossover* dan mutasi
2. Parameter VRPTW yang terdiri dari:
 - Jumlah objek wisata
 - Jarak antar objek wisata
 - Waktu tempuh antar objek wisata
 - Waktu kunjungan yang ideal untuk setiap objek wisata
 - Interval atau durasi kunjungan setiap objek wisata

Gambar 3.2 berikut adalah *Flowchart* yang digunakan untuk proses algoritma genetika.



Gambar 3.2 Flowchart Algoritma Genetika Pada Sistem

3.5. Proses Algoritma Genetika

Algoritma genetika merupakan algoritma pencarian solusi terbaik atau splusi yang mendekati hasil terbaik dari beberapa solusi yang ada. Algoritma genetika bekerja dengan membuat beberapa solusi yang dilakukan secara acak. Dimana solusi tersebut akan mengalami proses evolusi secara terus menerus hingga menghasilkan suatu solusi yang lebih baik. Setiap solusi yang terentuk mewakili satu kromosom dan satu individu terdiri dari satu kromosom. Kumpulan dari beberapa individu akan membentuk suatu populasi. Kemudian dari populasi tersebut akan dihasilkan individu-individu baru sampai dengan sejumlah generasi yang ditentukan.

Penyelesaian masalah dalam penelitian ini dibagi menjadi beberapa tahap yaitu:

1. Membangkitkan urutan perjalanan secara random sehingga membentuk susunan kromosom yang merepresentasikan urutan perjalanan.
2. Pembagian *time window* pada masing-masing objek wisata.
3. Pembagian durasi waktu kunjungan pada setiap objek wisata.
4. Mengolah data-data yang ada dengan menggunakan Algoritma genetika untuk mendapatkan solusi terbaik yaitu rute optimum.

Langkah-langkah proses pencarian rute terbaik adalah sebagai berikut:

1. Menentukan nilai awal
Langkah awal yang dilakukan adalah menentukan nilai awal dari panjang kromosom, penentuan cara teknik *decode* dan juga penentuan fungsi objektif.
2. Inisialisasi parameter
 - Menentukan jumlah individu pada setiap populasi
 - Jumlah generasi
 - Menentukan jumlah tujuan dan waktu pada setiap pelanggan
 - *Crossover rate* (Cr)
 - *Mutation rate* (Mr)

Langkah awal yang dilakukan adalah menentukan nilai awal berupa inisialiasi parameter, menentukan jumlah individu pada setiap populasi,

jumlah generasi, menentukan tujuan dan waktu untuk masing-masing tujuan, *crossover rate* (Cr), *Mutation rate* (Mr).

3. Populasi awal

Membangkitkan populasi awal sebanyak individu yang telah di Inisialiasai. Populasi awal yang dipakai ini dapat bermacam-macam dan besarnya juga dapat disesuaikan dengan kebutuhan. Misalnya jumlah kromosom pada setiap populasi mengikuti nilai populasi awal.

4. Membuat populasi baru dengan langkah yang telah ditentukan sebanyak jumlah generasi yang ditentukan.

5. Melakukan proses *crossover* pada dua parent yang telah dipilih, berdasarkan Cr yang telah ditentukan untuk menghasilkan *child*.

6. Melakukan proses mutasi pada dua parent yang telah dipilih berdsarkan Mr yang ditentukan.

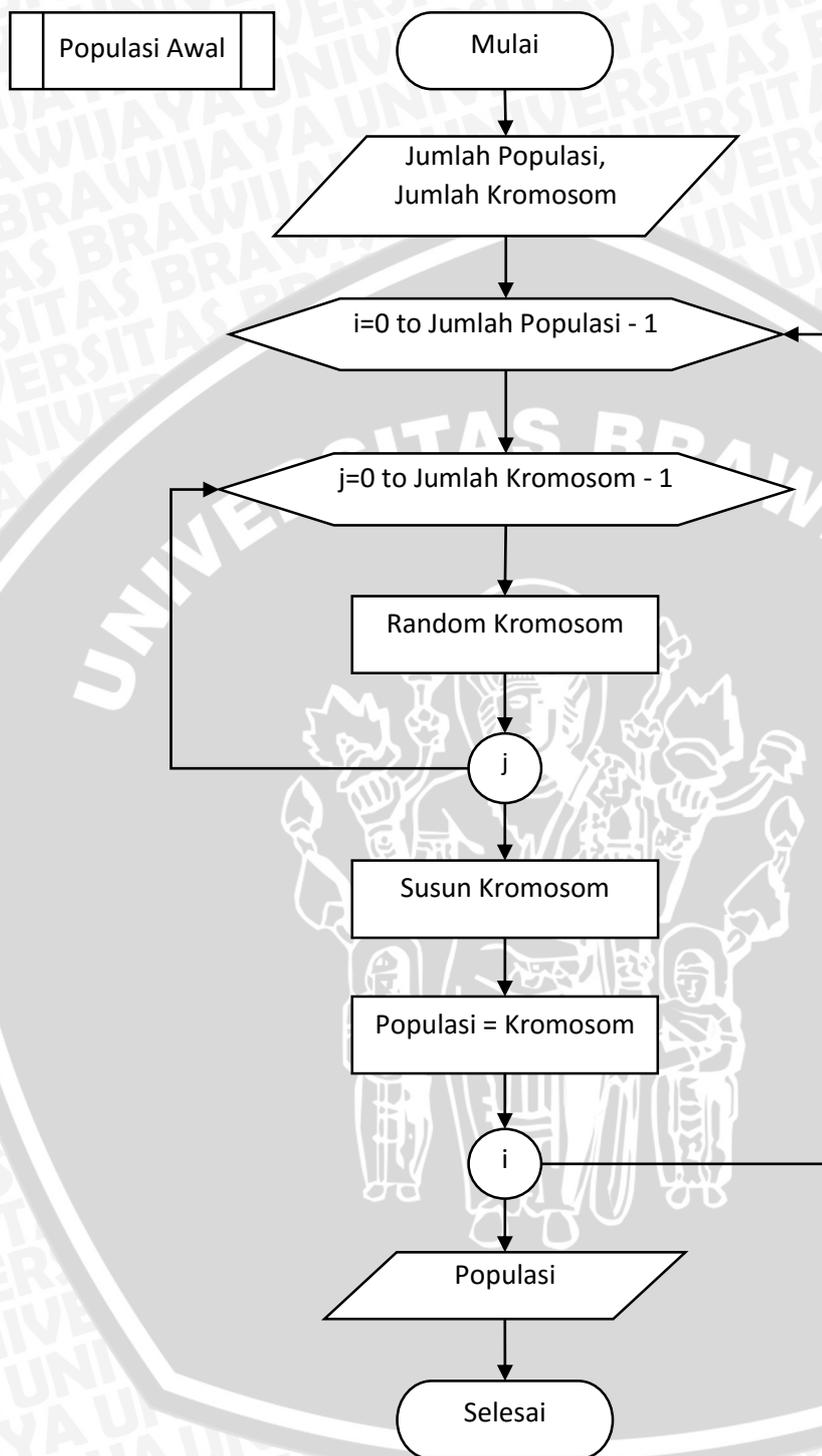
7. Menghitung nilai *fitness* pada masing-masing kromosom

8. Melakukan proses seleksi pada populasi dengan menggunakan metode *Roulete Wheel* untuk menentukan individu baru pada generasi berikutnya.

9. Jika pada kondisi akhir telah memenuhi kriteria, berhenti dan menghasilkan solusi terbaik pada seluruh generasi.

3.5.1. Membuat Populasi Awal

Tahap pertama dilakukan proses pembuatan Individu dan Populasi awal sebanyak *Popsiz*e yang telah ditentukan dengan panjang kromosom yang sama dengan jumlah objek wisata yang akan dikunjungi. Representasi yang digunakan adalah representasi permutasi dimana dalam satu individu terdapat gen. Gen tersebut merupakan objek wisata yang akan dikunjungi. Gambar 3.3 Menunjukkan proses pembuatan populasi awal.



Gambar 3.3 Flowchart Membangkitkan Populasi Awal

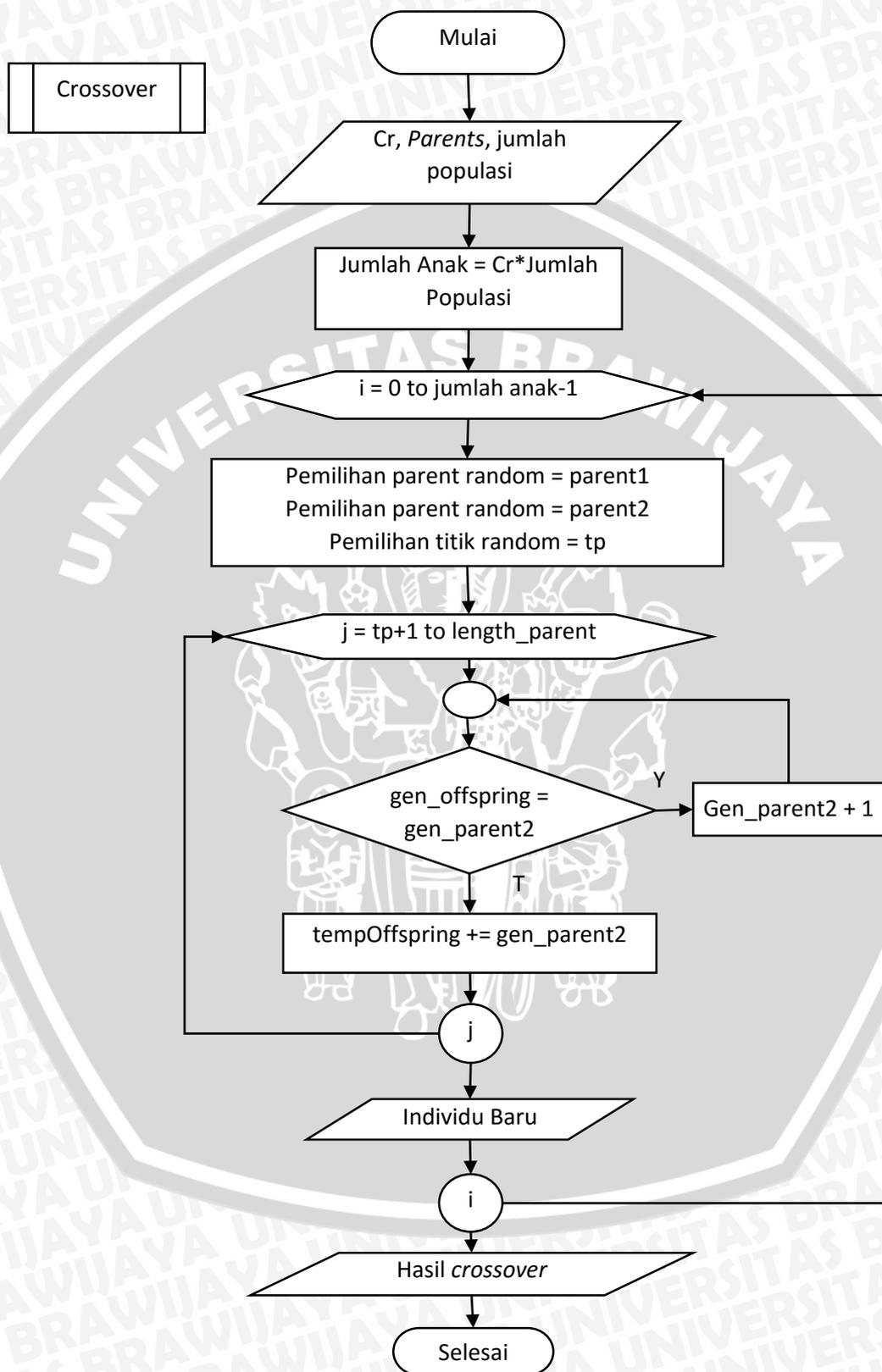


3.5.2. Proses *Crossover* (Pindah Silang)

Setelah tahap membentuk populasi awal dengan beberapa Individu maka tahap selanjutnya adalah memproduksi populasi baru dengan melakukan proses *crossover*. Pada tahap *crossover* ini populasi dari generasi yang dihasilkan sebelumnya, diambil untuk dijadikan sebagai calon *parent*. Banyaknya child yang akan terbentuk ditentukan berdasarkan nilai *crossover rate*. *Crossover* yang digunakan yaitu *crossover* satu titik atau *one cut point crossover*. Dimana titik tersebut dipilih secara acak sebagai titik potong untuk kemudian saling dipertukarkan antara 2 induk. Pada *crossover* ini dilakukan dengan cara menukar nilai gen pada posisi gen yang sama dari kedua *parent*. *Flowchart* metode *crossover* yang akan digunakan dapat dilihat pada Gambar 3.4.

Adapun langkah langkah proses *crossover* pada sistem ini berdasarkan Gambar 3.4 adalah:

1. Inisialisasi jumlah populasi dan nilai Cr
2. Menghitung jumlah anak yang dihasilkan sebanyak Cr dikali jumlah populasi
3. Memilih 2 *parents* secara acak dalam satu populasi
4. Menentukan titik potong secara random antara indeks ke-2 sampai dengan $length_parent-2$.
5. Melakukan proses *crossover* dengan metode *one-cut point* yaitu dengan mengambil potongan dari induk pertama. Potongan induk pertama kemudian disimpan sebagai calon *offspring* atau anak.
6. Gen anak yang diisi dengan potongan induk pertama kemudian ditambahkan dengan gen pada induk 2 yang belum terkandung pada gen anak.
7. Proses ini akan menghasilkan anak baru.



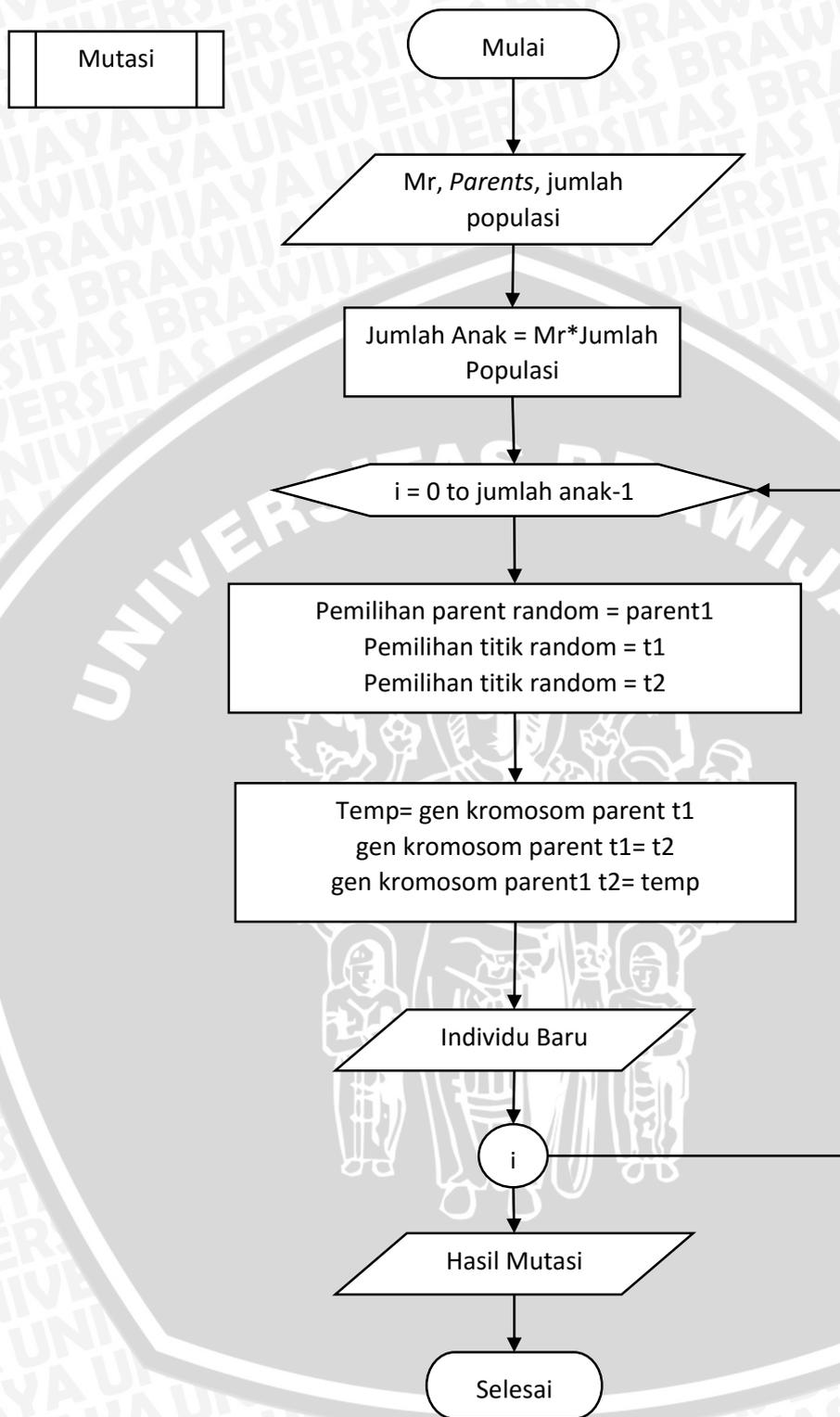
Gambar 3.4 Proses Crossover

3.5.3. Mutasi

Proses dalam membuat generasi baru yang kedua adalah proses mutasi. Parent diambil dari kromosom dari hasil proses berikutnya. Mutasi menciptakan individu baru dengan melakukan modifikasi satu atau lebih gen dalam individu yang sama. Mutasi berfungsi untuk menggantikan gen yang hilang dari populasi selama proses seleksi serta menyediakan gen yang tidak ada dalam populasi awal, sehingga mutasi akan meningkatkan variasi populasi. Dimana jumlah banyaknya child yang dihasilkan ditentukan dengan *mutation rate* yang sebelumnya telah diinisialisasi. Pada mutasi ada parameter yang penting yaitu *mutation rate*. Dengan adanya *mutation rate* diharapkan terdapat banyak individu yang akan mengalami mutasi. Metode mutasi yang digunakan adalah *Reciprocal Exchange mutation*. Metode ini bekerja dengan memilih dua porsi secara acak, kemudian menukar dua gen dalam posisi tersebut. Proses mutasi dapat dilihat pada Gambar 3.5.

Adapun langkah-langkah proses mutasi yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Inisiasi jumlah populasi dan nilai Mr .
2. Menghitung jumlah *Offspring* yang dihasilkan sebanyak Mr dikalikan dengan Jumlah Populasi
3. Melakukan proses *Exchange Mutation*
4. Memilih *parent* secara acak dari populasi. Jumlah *parent* yang dipilih disesuaikan dengan nilai Mr .
5. Random *exchange point* 1 (t_1) dan *exchange point* (t_2) untuk dilakukan proses pertukaran posisi gen *parent*. Titik ini dapat diambil secara bebas dari urutan gen manapun.
6. Tukarkan kedua posisi gen pada *parent*. Gen yang ditukarkan yaitu gen yang terpilih sebagai *exchange point 1* dan *exchange point 2*. Hal ini akan membentuk susunan kromosom baru.
7. Akan dihasilkan individu baru dari proses ini.



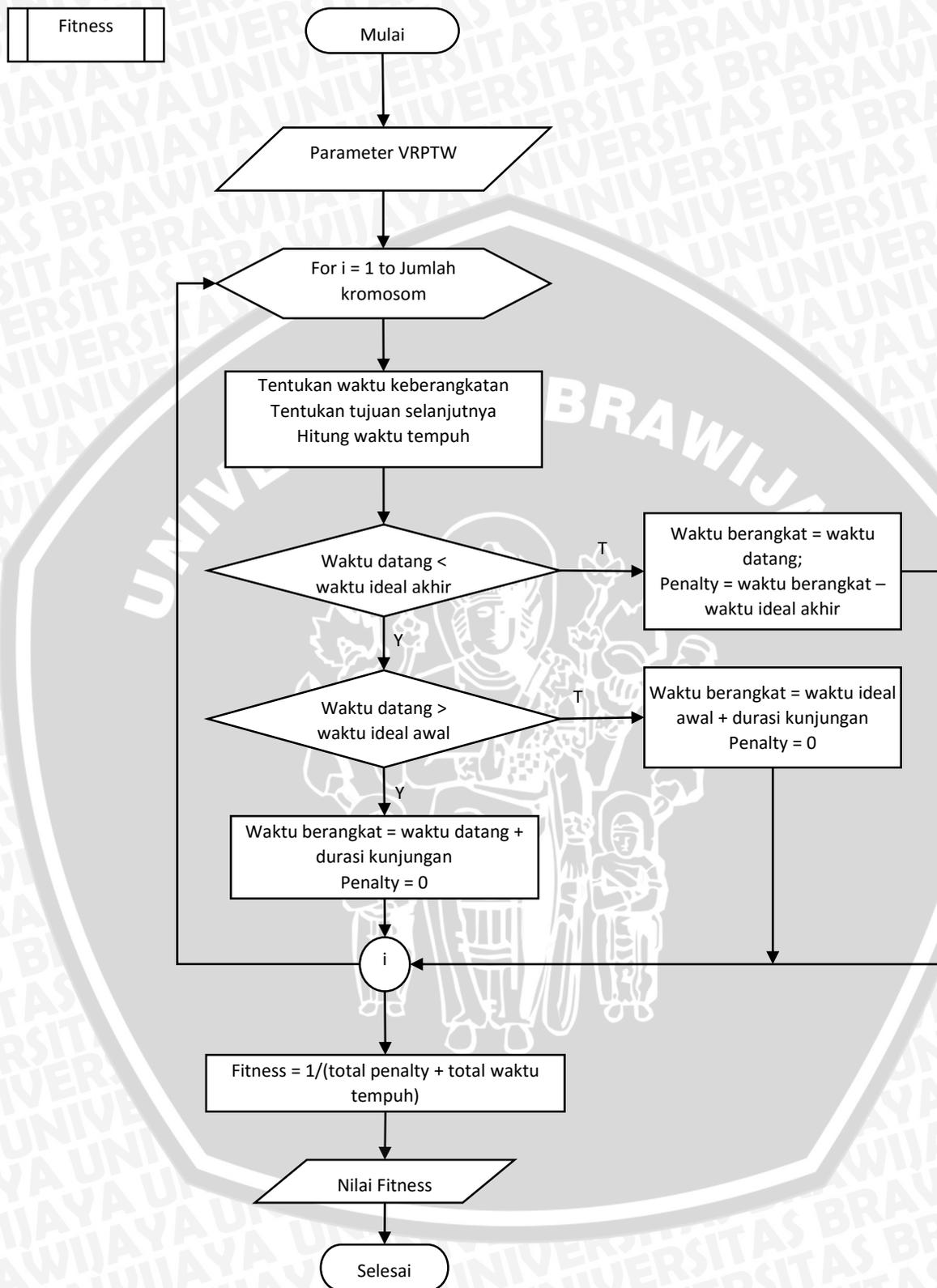
Gambar 3.5 Proses Mutasi

3.5.4. Menghitung nilai *Fitness*

Sebelum menuju tahap proses seleksi maka sebelumnya dilakukan proses perhitungan *fitness* terlebih dahulu. Nilai *fitness* dihitung dari hasil semua kromosom dari proses sebelumnya ditambah dengan kromosom hasil generasi dari proses *crossover* dan mutasi. Dari masing-masing kromosom akan dihasilkan nilai *Penalty* dan total waktu tempuh. *Penalty* ini didapat dari keterlambatan kedatangan kendaraan ketika sampai di lokasi objek wisata. Total waktu tempuh dihitung dengan menjumlahkan setiap waktu yang dihabiskan dalam perjalanan dari satu objek wisata ke objek wisata lainnya. *Penalty* dan total waktu tempuh ini kemudian dihitung berdasarkan rumus *fitness* sehingga menghasilkan nilai *fitness* dari masing-masing kromosom. Flowchart untuk perhitungan nilai *fitness* bisa dilihat pada Gambar 3.6.

Adapun langkah-langkah menghitung *fitness* adalah sebagai berikut :

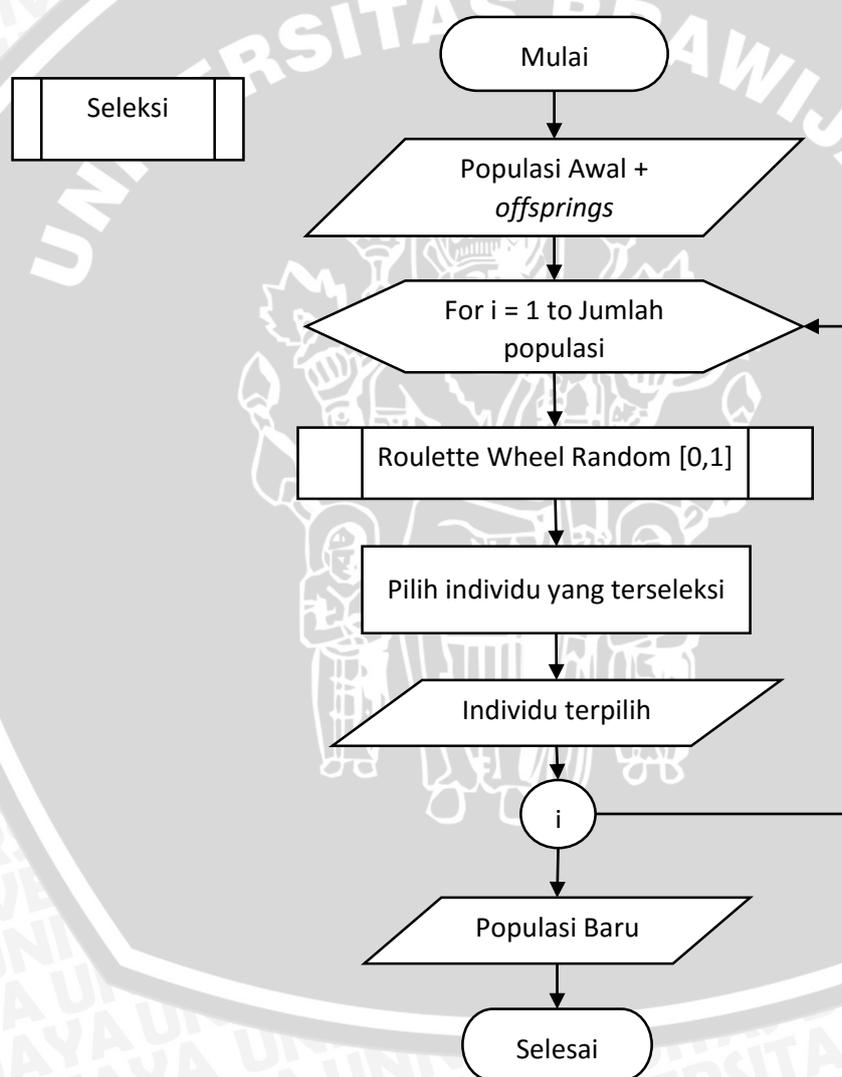
1. Inputkan waktu buka objek wisata
2. Menentukan waktu keberangkatan
3. Menentukan tujuan dari titik terakhir kunjungan ke titik kunjungan selanjutnya.
4. Hitung waktu tempuh semua perjalanan dari masing-masing objek wisata yang dikunjungi
5. Apabila kedatangan kendaraan sebelum waktu ideal awal kunjungan maka hitung waktu tunggu
6. Apabila kedatangan kendaraan setelah waktu ideal akhir kunjungan maka hitung waktu pinalti
7. Apabila tidak memenuhi keduanya maka waktu sekarang sama dengan waktu kedatangan ditambahkan dengan durasi kunjungan
8. Hitung *fitness* dengan menggunakan rumus *fitness* yaitu, satu dibagi dengan total *penalty* ditambah total waktu tempuh
9. Maka akan dihasilkan nilai *fitness*



Gambar 3.6 Proses Perhitungan *Fitness*

3.5.5. Seleksi

Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses seleksi. Proses seleksi dilakukan dengan menggunakan metode *Roulette wheel*. Ide dasar dari *Roulette Wheel Selection* untuk menentukan probabilitas seleksi untuk setiap *chromosome* sebanding pada nilai *fitness*. Semakin besar nilai *fitness* maka semakin besar kemungkinan individu tersebut terpilih sebagai *parent* pada generasi berikutnya. Untuk *flowchart* seleksi *Roulette Wheel* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Proses Seleksi

Adapun langkah-langkah seleksi menggunakan metode *roulette wheel* adalah sebagai berikut:

1. Masukkan populasi awal dan *offspring*
2. Individu dipilih sebanyak jumlah populasi dengan menggunakan metode *roulette wheel*. Metode ini memilih secara random nilai antara 0 dan 1. Kemudian dipilih individu berdasarkan probabilitas kumulatif yang dihasilkan.

3.6. Perhitungan Manual

Dalam penerapan algoritma genetika pada permasalahan VRPTW untuk penentuan rute pariwisata Bali dilakukan contoh perhitungan manual. Misalkan terdapat 8 objek wisata tujuan dengan *time windows*, dan jarak dalam waktu tempuh antara depot ke objek wisata dan objek wisata ke objek wisata lain yang berbeda-beda. Berikut adalah data objek wisata pada Tabel 3.1.

Tabel 3.1 Data Objek Wisata

Node	Nama Objek Wisata	Waktu ideal kunjungan (wita)		Durasi Kunjungan (Menit)
		Awal	Akhir	
1	Pantai Kuta	16:00	18:00	60
2	Bedugul	10:00	14:00	60
3	Ubud	8:00	11:00	60
4	jati luwih	11:00	16:00	60
5	Tirta Empul	13:00	16:00	60
6	taman ujung	14:00	17:00	60
7	Sangeh	10:00	15:00	60
8	Taman Budaya Arda Candra	8:00	20:00	60

Dalam perhitungan manual ini jumlah populasi ditetapkan sebagai parameter awal, yaitu 10. Panjang kromosom diasumsikan sebagai banyaknya tujuan yang harus dikunjungi dimulai dari keberangkatan kendaraan pukul 07:00. Kendaraan diasumsikan dengan kecepatan rata-rata 40 km/jam. Pada Tabel 3.2 merupakan data jarak objek wisata.

Tabel 3.2 Data Jarak Objek Wisata

Data Jarak Objek Wisata (km)									
	S	1	2	3	4	5	6	7	8
S	0	4.2	63.8	40.8	60.2	54.3	75.8	36.5	17.2
1	4.2	0	58.4	30.8	54.8	55.6	77.1	30	13.8

2	63.8	58.4	0	43.2	22.3	59.6	98.8	30.5	51.7
3	40.8	30.8	43.2	0	40.5	17.7	56.8	13.4	17.9
4	60.2	54.8	22.3	40.5	0	56	98.4	26.9	48
5	54.3	55.6	59.6	17.7	56	0	65.3	29.8	38.9
6	75.8	77.1	98.8	56.8	98.4	65.3	0	36.5	17.2
7	36.5	30	30.5	13.4	26.9	29.8	36.5	0	21.8
8	17.2	13.8	51.7	17.9	48	38.9	17.2	21.8	0

Dari data jarak objek wisata berdasarkan kecepatan rata-rata 40 km/jam, maka didapat waktu tempuh yang dapat dilihat pada Tabel 3.3. Contoh perhitungan node 1 ke node 2 adalah sebagai berikut.

$$\begin{aligned}\text{Waktu (1,2)} &= (\text{jarak/kecepatan rata-rata}) \times 60 \text{ menit} \\ &= (58.4/40) \times 60 \\ &= 88 \text{ menit}\end{aligned}$$

Tabel 3.3 Data Waktu Tempuh Objek Wisata

Data Waktu Tempuh Objek Wisata (menit)									
	t	1	2	3	4	5	6	7	8
T	0	6	96	61	90	81	114	55	26
1	6	0	88	46	82	83	116	45	21
2	96	88	0	65	33	89	148	46	78
3	61	46	65	0	61	27	85	20	27
4	90	82	33	61	0	84	148	40	72
5	81	83	89	27	84	0	98	45	58
6	114	116	148	85	148	98	0	109	91
7	55	45	46	20	40	45	109	0	33
8	26	21	78	27	72	58	91	33	0

3.6.1. Membuat populasi awal

Tahap awal yang dilakukan dalam membuat populasi awal adalah membangkitkan individu secara random dari 1-8 sebanyak jumlah populasi yang ditentukan. Representasi yang digunakan adalah representasi permutasi dimana kromosom yang dibangkitkan berupa rute tujuan objek wisata. Contoh populasi awal yang dibangkitkan dapat dilihat pada Tabel 3.4.

Tabel 3.4 Populasi Awal

Parent	Kromosom							
P1	7	6	1	2	3	4	5	8
P2	8	7	4	5	6	1	2	3

P3	1	2	3	4	5	6	7	8
P4	5	6	7	8	1	2	3	4
P5	8	2	3	4	5	6	1	7
P6	2	2	4	5	6	7	8	1
P7	5	6	7	8	1	2	3	4
P8	3	4	5	6	1	2	7	8
P9	4	5	6	7	8	1	2	3
P10	6	7	8	1	2	3	4	5

3.6.2. Menghitung Nilai *Fitness*

Nilai *fitness* yang digunakan dalam permasalahan VRPTW pada algoritma genetika adalah optimasi rute dengan meminimalkan waktu tunggu dan total jarak tempuh, maka persamaan yang digunakan adalah persamaan (2.1). Dalam perhitungan nilai *fitness* terdapat total jarak (km) yang diakumulasikan dalam satuan waktu (menit) berdasarkan Tabel 3.3.

Setiap perjalanan memiliki waktu tempuh yang berbeda-beda tergantung rute yang dilewati. Perjalanan kendaraan akan mendapat *penalty* apabila waktu sampai kendaraan lebih dari waktu kunjungan akhir objek wisata. Waktu *penalty* dihitung dengan cara waktu sampai dikurangi akhir waktu ideal kunjungan. Contoh perhitungan dapat dilihat pada Tabel 3.5 – Tabel 3.7.

Tabel 3.5 Perhitungan Pinalti P1

Node	Waktu Ideal		Perjalanan (km)	Waktu Tempuh	Durasi	Sampai	Tunggu	Mulai	selesai	Pinalty
	Awal	Akhir								
7	10:00	15:00	55	0:55:00	1:00:00	7:55	2:05:00	10:00	11:00	0:00:00
6	14:00	17:00	109	1:49:00	1:00:00	12:49	1:11:00	14:00	15:00	0:00:00
1	16:00	18:00	116	1:56:00	1:00:00	16:56	0:00:00	16:56	17:56	0:00:00
2	10:00	14:00	88	1:28:00	1:00:00	19:24	0:00:00	19:24	19:24	5:24:00
3	8:00	11:00	65	1:05:00	1:00:00	20:29	0:00:00	20:29	20:29	9:29:00
4	11:00	16:00	61	1:01:00	1:00:00	21:30	0:00:00	21:30	21:30	5:30:00
5	13:00	16:00	84	1:24:00	1:00:00	22:54	0:00:00	22:54	22:54	6:54:00
8	8:00	20:00	58	0:58:00	1:00:00	23:52	0:00:00	23:52	23:52	3:52:00
Total				10:36:00						31:09:00
Dalam Satuan Menit				636						1869

Tabel 3.6 Perhitungan Pinalti P2

Node	Waktu Ideal		Perjalanan (km)	Waktu Tempuh	Durasi	Sampai	Tunggu	Mulai	selesai	Pinalty
	Awal	Akhir								
8	8:00	20:00	26	0:26:00	1:00:00	7:26	0:34:00	8:00	9:00	0:00:00
7	10:00	15:00	33	0:33:00	1:00:00	9:33	0:27:00	10:00	11:00	0:00:00
4	11:00	16:00	40	0:40:00	1:00:00	11:40	0:00:00	11:40	12:40	0:00:00
5	13:00	16:00	84	1:24:00	1:00:00	14:04	0:00:00	14:04	15:04	0:00:00
6	14:00	17:00	98	1:38:00	1:00:00	16:42	0:00:00	16:42	17:42	0:00:00
1	16:00	18:00	116	1:56:00	1:00:00	19:38	0:00:00	19:38	19:38	1:38:00
2	10:00	14:00	88	1:28:00	1:00:00	21:06	0:00:00	21:06	21:06	7:06:00
3	8:00	11:00	65	1:05:00	1:00:00	22:11	0:00:00	22:11	22:11	11:11:00
Total				9:10:00						19:55:00
Dalam Satuan Menit				550						1195

Tabel 3.7 Perhitungan Pinalti P3

Node	Waktu Ideal		Perjalanan (km)	Waktu Tempuh	Durasi	Sampai	Tunggu	Mulai	selesai	Pinalty
	Awal	Akhir								
1	16:00	18:00	6	0:06:00	1:00:00	7:06	8:54:00	16:00	17:00	0:00:00
2	10:00	14:00	88	1:28:00	1:00:00	18:28	0:00:00	18:28	18:28	4:28:00
3	8:00	11:00	65	1:05:00	1:00:00	19:33	0:00:00	19:33	19:33	8:33:00
4	11:00	16:00	61	1:01:00	1:00:00	20:34	0:00:00	20:34	20:34	4:34:00
5	13:00	16:00	84	1:24:00	1:00:00	21:58	0:00:00	21:58	21:58	5:58:00
6	14:00	17:00	98	1:38:00	1:00:00	23:36	0:00:00	23:36	23:36	6:36:00
7	10:00	15:00	109	1:49:00	1:00:00	1:25	0:00:00	1:25	1:25	10:25:00
8	8:00	20:00	33	0:33:00	1:00:00	1:58	0:00:00	1:58	1:58	5:58:00
Total				9:04:00						46:32:00
Dalam Satuan Menit				544						2792

Tabel 3.5 – Tabel 3.7 merupakan contoh perhitungan beberapa individu dengan urutan kunjungan dari awal hingga kunjungan terakhir. Pada Tabel 3.5, awal keberangkatan kendaraan pada pukul 07:00. Pada node pertama yaitu node 7, waktu tempuh objek wisata dari depot adalah 55 menit, maka waktu sampainya pukul 07:55.

$$\begin{aligned}
 \text{Waktu sampai} &= \text{awal keberangkatan} + \text{waktu tempuh} \\
 &= 7:00 + 55 \text{ menit} \\
 &= 7:55
 \end{aligned}$$

Karena waktu mulainya pukul 10:00 maka dikenakan waktu tunggu yang dihitung dengan cara waktu mulai dikurangi waktu sampai.

$$\text{Waktu tunggu} = \text{waktu mulai} - \text{waktu sampai}$$

$$= 10:00 - 7:55$$

$$= 2:05:00$$

Apabila kedatangan kendaraan setelah waktu akhir maka akan dikenakan waktu *penalty* yang dihitung dengan cara, waktu sampai dikurangi waktu ideal akhir kunjungan.

$$\text{Waktu } \textit{penalty} = \text{waktu sampai} - \text{waktu ideal akhir kunjungan}$$

Waktu selesai dihitung dengan cara, waktu mulai ditambah durasi kunjungan. Apabila kedatangan dikenakan *penalty* maka waktu selesai sama dengan waktu mulai.

$$\begin{aligned} \text{Waktu selesai} &= \text{waktu mulai} + \text{durasi} \\ &= 10:00 + 1:00:00 \\ &= 11:00 \end{aligned}$$

Setelah hasil *penalty* dan hasil total waktu perjalanan diketahui, maka tahap selanjutnya yaitu menghitung *fitness* setiap individu menggunakan persamaan 2.1.

Nilai *fitness* didapat dari perhitungan setiap satu individu. Hasil perhitungan *fitness* dapat dilihat pada Tabel 3.8. Contoh perhitungan *fitness* pada individu P1 adalah sebagai berikut

$$\textit{fitness P1} = \frac{1}{1869 + 636} = 0.000399$$

Tabel 3.8 Perhitungan *Fitness*

Individu	Kromosom	Penalti	tempuh	<i>Fitness</i>
P1	76123458	1869	636	0.000399
P2	87456123	1195	550	0.000573
P3	12345678	2792	544	0.000300
P4	56781234	2006	556	0.000390
P5	82345617	678	573	0.000799
P6	23456781	631	567	0.000835
P7	56781234	2006	556	0.000390

P8	34561278	861	587	0.000691
P9	45678123	1351	588	0.000516
P10	67812345	2365	670	0.000329

3.6.3. Crossover

Proses reproduksi yang pertama dilakukan adalah *crossover* dengan menggunakan metode *one-cut point crossover* untuk membangkitkan keturunan baru (*offspring*) dari hasil persilangan antara dua parent yang diambil secara acak. Tingkat *crossover* (*crossover rate*) merupakan jumlah kromosom dalam suatu populasi yang diharapkan akan mengalami *crossover* hingga menghasilkan keturunan baru yang memiliki perbedaan sifat dari induknya seperti pada proses berikut. Terdapat P3 dan P8 yang merupakan individu terpilih sebagai parent. Kemudian menentukan dua titik potong secara acak sebagai area yang akan disilangkan. Titik potong P3 dan P8 terdapat pada titik 4. Setelah itu lakukan pertukaran area pada parent1 dan parent2. Jika terdapat nilai yang sama dalam satu individu maka nilai yang sama diganti dengan nilai yang belum ada pada individu tersebut. Nilai diganti sesuai dengan ketentuan persilangan.

Proses *crossover* ini memiliki nilai $Cr=0.4$, jadi dilakukan 4 kali *crossover* karena $child = 10 \times 0.4 = 4$, maka keturunan yang dihasilkan dapat dilihat pada Gambar 3.8.

P3	1	2	3	4	5	6	7	8
P8	3	4	5	6	1	2	7	8
Offspring P11	1	2	3	4	5	6	7	8
P4	5	6	7	8	1	2	3	4
p9	4	5	6	7	8	1	2	3
Offspring P12	5	6	7	8	4	1	2	3
P1	7	6	1	2	3	4	5	8
P7	5	6	7	8	1	2	3	4
Offspring P13	7	6	1	2	5	8	3	4
P1	7	6	1	2	3	4	5	8
P9	4	5	6	7	8	1	2	3
Offspring P14	7	6	1	2	4	5	8	3

Gambar 3.8 Proses *Crossover*

3.6.4. Mutasi

Proses reproduksi selanjutnya adalah mutasi. Pada operator mutasi tidak perlu menggunakan dua parent untuk menghasilkan keturunan. Mutasi dilakukan oleh satu individu sebagai induk yang dipilih secara acak kemudian dilakukan penukaran gen dalam individu tersebut hingga menghasilkan susunan gen baru pada individu baru hasil mutasi. Pada proses mutasi ini gen yang akan ditukar sebanyak 2 titik yang diambil secara acak. Nilai Mr adalah 0.2, maka dilakukan proses mutasi pada dua individu. Individu yang terpilih sebagai induk adalah P3 dengan penukaran gen pada titik-2, ditukar dengan gen pada titik-6. Kemudian P8 dengan penukaran gen pada titik-2 ditukar dengan gen pada titik-6. Hal ini dilakukan agar anak yang dihasilkan tidak mengalami banyak kemiripan dengan induk. Contoh proses mutasi dapat dilihat pada Gambar 3.9.

P3	1	2	3	4	5	6	7	8
P15	1	6	3	4	5	2	7	8
P8	3	4	5	6	1	2	7	8
P16	3	2	5	6	1	4	7	8

Gambar 3.9 Proses Mutasi

3.6.5. Seleksi

Setelah dilakukan proses reproduksi *crossover* dan mutasi, langkah selanjutnya adalah proses seleksi. Metode yang digunakan pada proses seleksi adalah *roulette wheel*. Perhitungan nilai *fitness*, probabilitas, dan probabilitas kumulatif dinyatakan pada persamaan 2.2 sampai dengan 2.4. Contoh perhitungan probabilitas dan probabilitas kumulatif pada individu P1 adalah sebagai berikut.

$$prob_{P1} = \frac{0.000399}{0.007797} = 0.051202$$

$$probCum_{P1} = 0 + 0.051202 = 0.051202$$

$$probCum_{P2} = 0.051202 + 0.073501 = 0.124703$$

Tabel 3.9 Perhitungan Seleksi

Individu	Kromosom	Penalti	tempuh	<i>Fitness</i>	Prob	ProbCum
P1	76123458	1869	636	0.000399	0.051202	0.051202

P2	87456123	1195	550	0.000573	0.073501	0.124703
P3	12345678	2792	544	0.000300	0.038447	0.16315
P4	56781234	2006	556	0.000390	0.050062	0.213213
P5	82345617	678	573	0.000799	0.102526	0.315738
P6	23456781	631	567	0.000835	0.107062	0.4228
P7	56781234	2006	556	0.000390	0.050062	0.472862
P8	34561278	861	587	0.000691	0.088577	0.56144
P9	45678123	1351	588	0.000516	0.066147	0.627587
P10	67812345	2365	670	0.000329	0.04226	0.669847
P11	12345678	2792	544	0.000300	0.038447	0.708295
P12	56784123	2242	628	0.000348	0.04469	0.752985
P13	76125834	1845	603	0.000408	0.052394	0.805378
P14	76124583	1727	570	0.000435	0.055838	0.861216
P15	16345278	2955	520	0.000288	0.036909	0.898126
P16	32561478	675	584	0.000794	0.101874	1
		Total		0.007797		

Setelah didapatkan nilai *fitness* dari masing – masing individu kemudian dipilih 10 individu terbaik berdasarkan nilai *fitness* yang tertinggi seperti pada Tabel 3.9. Dengan menggunakan *roulette wheel* membangkitkan nilai random interval [0, 1]. Kemudian pilih nilai k sebagai individu terpilih dimana nilai $probCum_{k-1} < random \leq probCum_k$.

Tabel 3.10 Perhitungan *Roulette Wheel*

Individu	Random	k	Individu	Fitness
P1	0.703993	11	p11	0.000300
P2	0.812046	14	p14	0.000435
P3	0.169125	4	p4	0.000390
P4	0.585947	9	p9	0.000516
P5	0.781151	13	p13	0.000408
P6	0.970429	16	p16	0.000794
P7	0.341632	6	p6	0.000835
P8	0.879543	15	p15	0.000288

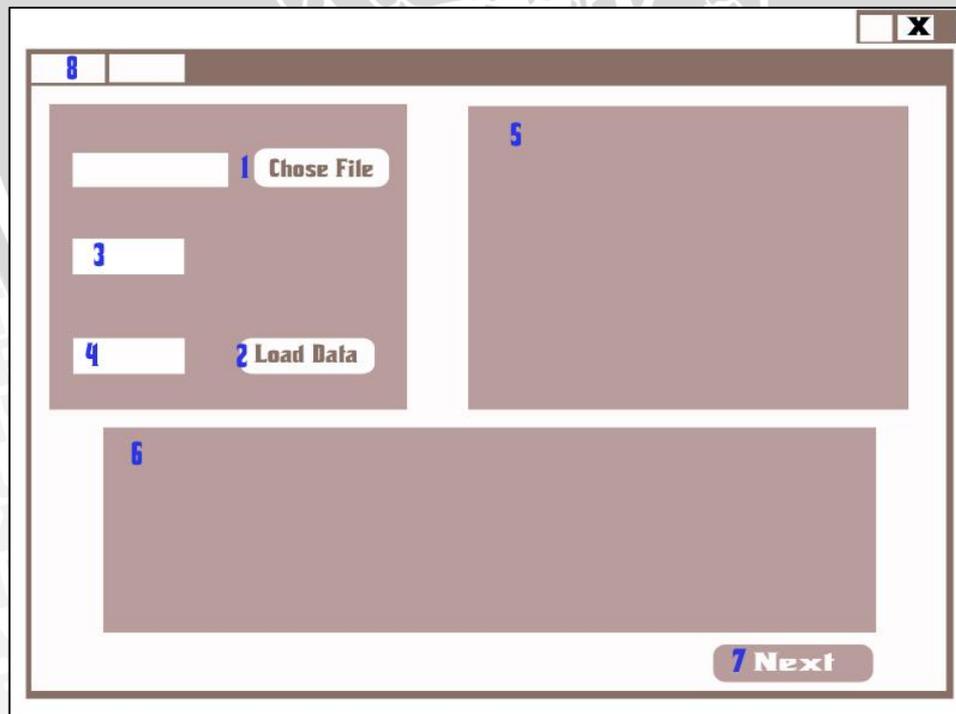
P9	0.131532	3	p3	0.000300
P10	0.585354	9	p9	0.000516

3.7. Perancangan *User interface*

Perancangan *user interface* dari perangkat lunak untuk permasalahan penentuan rute pariwisata di Bali terdiri dari dua halaman. Halaman pertama berisi *import* data dan tampilan data objek wisata yang dapat dilihat pada Gambar 3.10.

Keterangan Gambar 3.10 :

1. Pilih lokasi data yang akan di-*import*
2. *Load* data yang telah dipilih
3. Inputkan nama *sheet* jadwal
4. Inputkan nama *sheet* waktu tempuh
5. Tampilan data jadwal buka objek wisata
6. Tampilan data waktu tempuh objek wisata
7. Beralih ke halaman selanjutnya
8. Tab pindah halaman

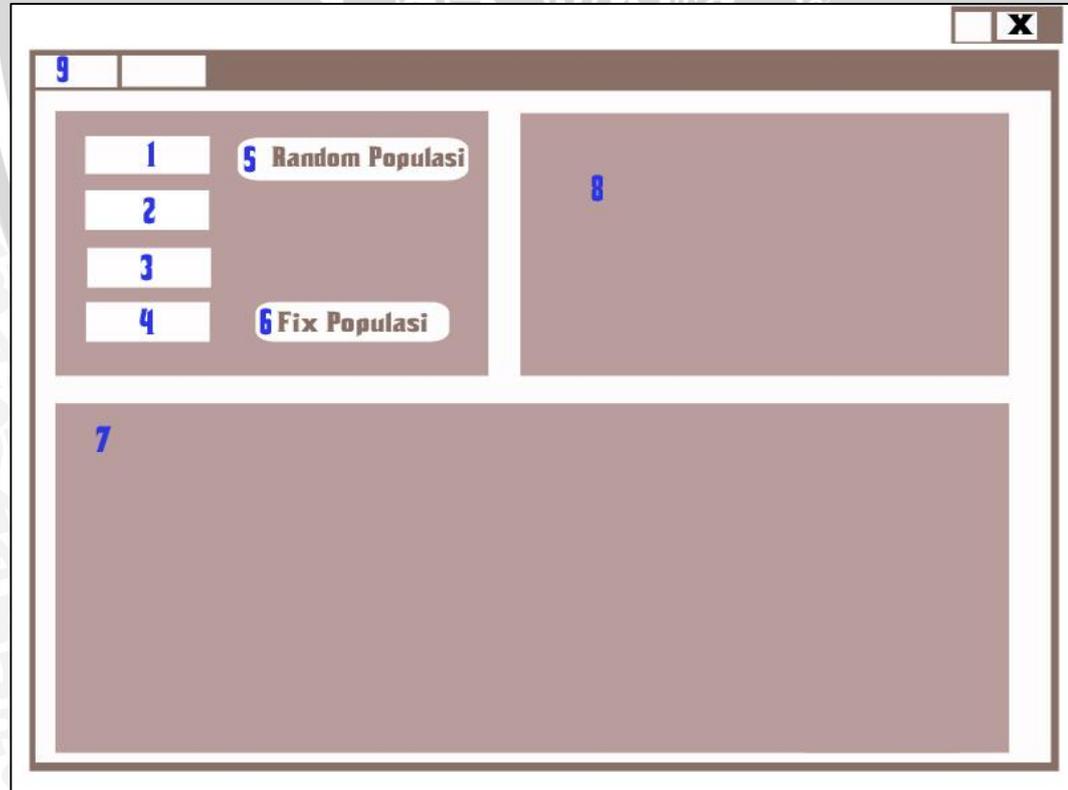


Gambar 3.10 Halaman Utama Perancangan *User interface*

Halaman kedua perancangan *user interface* merupakan halaman yang digunakan untuk memasukkan parameter algoritma genetika dan melakukan perhitungan. Pada halaman ini juga ditampilkan proses perhitungan dan hasil generasi terbaik. Halaman kedua dapat dilihat pada Gambar 3.11.

Keterangan Gambar 3.11:

1. Input Banyaknya populasi
2. Input nilai *crossover rate*
3. Input nilai *mutation rate*
4. Input jumlah generasi
5. Melakukan perhitungan dengan populasi random
6. Melakukan perhitungan sesuai dengan perhitungan manual
7. Tampilan solusi dan perhitungan
8. Tampilan generasi terbaik
9. Tab pindah halaman



Gambar 3.11 Halaman Kedua Perancangan *User interface*

3.8. Perancangan Uji Coba dan Evaluasi

Proses pengujian merupakan evaluasi perangkat lunak yang akan diuji dengan berbagai macam parameter GA. Pengujian bertujuan untuk mengetahui pengaruh GA terhadap hasil optimasi rute distribusi beras bersubsidi. Hasil yang didapatkan berupa individu dengan nilai *fitness* terbaik. Uji coba dilakukan dalam 4 skenario, dimana setiap skenario akan dilakukan pengujian sebanyak 10 kali lalu diambil rata-rata nilai *fitness* terbaik dari setiap skenario uji. Pengujian tersebut antara lain adalah:

1. Uji coba untuk menentukan banyaknya generasi yang optimal untuk proses algoritma genetika pada pemilihan rute terbaik objek wisata Bali.
2. Uji coba untuk menentukan banyaknya populasi yang optimal untuk proses algoritma genetika pada pemilihan rute terbaik objek wisata Bali.
3. Uji coba untuk mencari kombinasi *mutation rate* dan *crossover rate* yang terbaik untuk menyelesaikan permasalahan VRPTW untuk pemilihan rute terbaik objek wisata Bali.
4. Uji coba untuk menentukan panjang kromosom yang terbaik untuk menentukan jumlah objek wisata yang dapat dikunjungi dalam satu hari.

3.8.1. Uji Coba Banyaknya Generasi

Uji coba banyaknya generasi terhadap program dilakukan untuk mengetahui seberapa banyak generasi yang optimal pada permasalahan VRPTW. Uji coba akan dilakukan sebanyak 10 kali dengan generasi kelipatan 50 yang dimulai dari 50 - 500 generasi. Semakin banyak jumlah generasi, maka semakin meningkat kemampuan algoritma genetika dalam mencari solusi terbaik. Jumlah populasi yang digunakan sebanyak 10 dengan Cr dan Mr 0.5 : 0.5 pada metode seleksi *Roulette Wheel*. Tabel uji coba banyaknya generasi dapat dilihat pada Tabel 3.11

Tabel 3.11 Skenario Uji Coba Banyak Generasi

Generasi	Percobaan										Rata Rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50											
100											
150											

200											
250											
300											
350											
400											
450											
500											

3.8.2. Uji Coba Banyaknya Populasi

Uji coba banyaknya populasi pada program dilakukan untuk mengetahui banyaknya populasi dengan solusi optimasi terbaik pada permasalahan VRPTW. Semakin banyak jumlah populasi maka semakin berpengaruh terhadap peningkatan kemampuan algoritma genetika untuk menemukan solusi terbaik. Uji coba banyaknya populasi dilakukan pada generasi terbaik. Skenario uji coba dilakukan sebanyak generasi terbaik yang diperoleh dari percobaan banyak generasi dengan nilai Cr dan Mr 0.5:0.5 dan metode seleksi *Roulette Wheel*. Banyaknya populasi adalah kelipatan 5 yaitu dari 5 populasi hingga 50 populasi. Tabel uji coba banyaknya populasi dapat dilihat pada Tabel 3.12

Tabel 3.12 Skenario Uji Coba Banyaknya Populasi

Banyak Populasi	Nilai <i>Fitness</i>										Rata Rata <i>Fitness</i>
	Percobaan										
	1	2	3	4	5	6	7	8	9	10	
5											
10											
15											
20											
25											
30											
35											
40											
45											

3.8.4. Uji Coba Panjang Kromosom

Uji coba panjang kromosom pada program dilakukan untuk mengetahui panjang kromosom dengan solusi optimasi terbaik pada permasalahan VRPTW. Semakin panjang kromosom yang dihasilkan maka semakin banyak objek wisata yang dapat dilewati dalam satu hari. Skenario uji coba dilakukan dengan mengujikan panjang kromosom antara 5 – 10, dengan masing-masing percobaan sebanyak 10 kali. Skenario uji coba dilakukan sebanyak generasi terbaik yang diperoleh dari uji coba banyak generasi. Jumlah populasi yang digunakan disesuaikan dengan jumlah populasi terbaik yang diperoleh dari uji coba banyak populasi. Dan kombinasi nilai Cr dan Mr yang dipakai berdasarkan hasil terbaik pada uji coba kombinasi Cr dan Mr , dengan metode seleksi *Roulette Wheel*. Tabel uji coba banyaknya populasi dapat dilihat pada Tabel 3.14.

Tabel 3.14 Skenario Uji Coba Panjang Kromosom

Panjang Kromosom	Nilai <i>Fitness</i>										Rata Rata <i>Fitness</i>
	Percobaan										
	1	2	3	4	5	6	7	8	9	10	
5											
6											
7											
8											
9											
10											

3.8.5. Uji Coba Perbandingan Algoritma Genetika dan Algoritma *Greedy*

Uji coba perbandingan algoritma genetika dan algoritma *greedy* dilakukan untuk mengetahui nilai efisiensi yang dihasilkan dari solusi optimasi terbaik pada permasalahan VRPTW. Skenario uji coba dilakukan dengan mengujikan panjang kromosom antara 5 – 10. Skenario uji coba dilakukan sebanyak generasi terbaik yang diperoleh dari uji coba banyak generasi. Jumlah populasi yang digunakan disesuaikan dengan jumlah populasi terbaik yang diperoleh dari uji coba banyak populasi. Dan kombinasi nilai Cr dan Mr yang dipakai berdasarkan hasil terbaik pada uji coba kombinasi Cr dan Mr .

BAB IV

IMPLEMENTASI

Pada bab implementasi ini merupakan tahap penerapan sistem, dimana aplikasi siap dijalankan pada keadaan sebenarnya penerapan sistem yang akan dijelaskan terdiri dari lingkungan implementasi, implementasi aplikasi dan implementasi tampilan *user interface*.

4.1. Lingkungan Implementasi

Lingkungan implementasi merupakan ruang lingkup kebutuhan dalam penerapan sistem Algoritma Genetika pada VRPTW untuk penentuan rute pariwisata Bali. Terdapat dua lingkungan yang digunakan dalam membangun sistem aplikasi tersebut, yaitu lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1. Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam implementasi aplikasi algoritma genetika untuk penentuan rute pariwisata Bali adalah sebagai berikut.

1. Processor 2.3 GHz Intel Core i5
2. Memory 4 GB 1333 MHz DDR3
3. Harddisk dengan kapasitas 500 GB
4. VGA Card Intel HD Graphics 3000 384 MB

4.1.2. Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi aplikasi algoritma genetika untuk penentuan rute pariwisata Bali adalah sebagai berikut.

1. Sistem operasi yang digunakan adalah Windows 8.1 Pro 64 bit.
2. *Editor* yang digunakan adalah Visual Studio 2010
3. Penyimpanan data menggunakan Microsoft Excel 2013

4.2. Implementasi Program

Berdasarkan metodologi dan perancangan sistem yang telah dijelaskan pada Bab 3, maka selanjutnya akan dijelaskan implementasi proses-proses tersebut kedalam bentuk sistem dengan bahasa pemrograman C#.

4.2.1. Membangkitkan Populasi Awal

Membangkitkan populasi awal dimulai dengan membangkitkan individu. Satu individu dibentuk secara random dengan panjang sesuai dengan panjang kromosom yaitu 8. Proses pembangkitan individu dilakukan pengecekan setiap gen agar tidak terdapat nilai gen yang sama dalam satu kromosom. Proses membangkitkan populasi awal dapat dilihat pada *source code* 4.1.

```
1 public string[,] inisiasiPopulasi
2     {
3         get
4         {
5             parents = new string[valPopSize, totKromosom];
6             r = new Random();
7
8             for (int i = 0; i < valPopSize; i++)
9             {
10                for (int j = 0; j < totKromosom; j++)
11                {
12                    parents[i, j] = j.ToString();
13                }
14            }
15
16            for (int i = 0; i < valPopSize; i++)
17            {
18                x = 0;
19                while (x < 2)
20                {
21                    valRandom1 = r.Next(1, 9);
22                    valRandom2 = r.Next(1, 9);
23                    string temp = "";
24                    if (valRandom1 != valRandom2)
25                    {
26                        temp = parents[i, valRandom2];
27                        parents[i, valRandom2] = parents[i,
28 valRandom1];
29                        parents[i, valRandom1] = temp;
30                    }
31                    x++;
32                }
33            }
34            return parents;
35        }
36    }
37 }
```

Source code 4.1 Membangkitkan Kromosom Awal

1. Baris 5-6 merupakan pendefinisian *parent* dan nilai random

2. Baris 23-24 merupakan penentuan rentangan nilai acak untuk kromosom
3. Baris 29-32 menyusun kromosom

4.2.2. Proses Crossover

Metode *crossover* yang digunakan adalah *one-cut point crossover*. Sebelum dilakukan *crossover*, maka dilakukan pemilihan induk yang akan mengalami *crossover* sesuai dengan nilai *Cr* yang dimasukkan. Proses *crossover* dapat dilihat di *source code* 4.2.

```

1  public void reproduksiCrossover()
2      {
3      lengtKromosom = strParent1.Length;
4      r = new Random();
5      valRandom1 = r.Next(3, totData - 2);
6      noParent1 = r.Next(0, valPopSize);
7      noParent2 = r.Next(0, valPopSize);
8      strChild = new string[lengtKromosom];
9
10     x=0;
11     while (x < valRandom1)
12     {
13         strChild[x] = strParent1[x];
14         x++;
15     }
16
17     for (int i = 0; i < lengtKromosom; i++)
18     {
19         tempVal = 0;
20         for (int j = 0; j < lengtKromosom; j++)
21         {
22             if (strParent2[i] == strChild[j])
23             {
24                 j = lengtKromosom;
25                 tempVal = 1;
26             }
27         }
28
29         if (tempVal == 0)
30         {
31             strChild[valRandom1] = strParent2[i];
32             valRandom1 += 1;
33         }
34     }
35 }
36

```

Source code 4.2 Proses Crossover

1. Baris 4 – 7 merupakan penentuan titik potong dan *parent* secara acak
2. Baris 10 – 15 menyusun kromosom *offspring* dari sebelum titik potong
3. Baris 17 – 34 mencocokkan nilai gen *parent1* dan *parent2*, kemudian menyusun kromosom *offspring*.

4.2.3. Proses Mutasi

Metode mutasi yang digunakan adalah metode *reciprocal exchange mutation*. Sebelum dilakukannya proses mutase, maka dilakukan pemilihan induk terlebih dahulu. Dimana induk yang terpilih akan mengalami proses mutasi sesuai dengan nilai *mutation rate* yang telah ditentukan. Prose mutasi dapat dilihat pada *source code* 4.3.

```

1 public void reproduksiMutasi()
2     {
3         lengtKromosom = strParent1.Length;
4         r = new Random();
5         valRandom1 = r.Next(1, totData / 2);
6         valRandom2 = r.Next(totData / 2, totData);
7
8         strChild = new string[lengtKromosom];
9
10        strChild = strParent1;
11        tempString = child[valRandom1];
12        child[valRandom1] = child[valRandom2];
13        child[valRandom2] = tempString;
14
15    }

```

Source code 4.3 Proses Mutasi

1. Baris 4 – 6 menentuka nilai yang ditukarkan secara acak
2. Baris 10 – 13 melakukan pertukaran nilai gen

4.2.4. Perhitungan Fitness

Perhitungan nilai *fitness* diperoleh dari total waktu tempuh kesetiap objek wisata yang dimulai dari titik awal keberangkatan hingga tujuan akhir ditambahkan dengan total *penalty* yang didapat pada setiap objek wisata. Perhitungan nilai *fitness* dilakukan pada setiap individu dalam suatu populasi. Proses perhitungan *fitness* dapat dilihat pada *source code* 4.4.

```

1 public void cariNilaiFitness()
2     {
3
4         waktuBerangkat = Convert.ToDateTime("07:00").TimeOfDay;
5         totalPenalti = 0;
6         totalWaktuTempuh = 0;
7
8         totLength = individu.Length;
9         temp1 = new string[totLength];
10        temp2 = new string[totLength];
11        tempPenalti = new int[totLength];
12        tempWaktuTempuh = new int[totLength];
13
14        for (int i = 1; i < totLength; i++)

```

```

15     {
16         awal = Convert.ToInt32(individu[i - 1]);
17         tujuan = Convert.ToInt32(individu[i]);
18
19         waktuPerjalanan = matrixWaktu[awal, tujuan];
20         tempWaktuTempuh[i] = waktuPerjalanan;
21         waktuDatang = waktuBerangkat.Add(new TimeSpan(0,
22 waktuPerjalanan, 0));
23         temp1[i] = waktuDatang.ToString();
24
25         totalWaktuTempuh+= tempWaktuTempuh[i];
26         tsWaktuidealAwal =
27 Convert.ToDateTime((waktuidealAwal[tujuan]).ToString()).TimeOfDay;
28         tsWaktuidealAkhir =
29 Convert.ToDateTime((waktuidealAkhir[tujuan]).ToString()).TimeOfDay;
30
31         if (waktuDatang < tsWaktuidealAkhir)
32         {
33             if (waktuDatang > tsWaktuidealAwal)
34             {
35                 waktuBerangkat = waktuDatang.Add(new
36 TimeSpan(0, 60, 0));
37                 totalPenalti += 0;
38
39                 temp2[i] = waktuBerangkat.ToString();
40                 tempPenalti[i] = 0;
41             }
42             else
43             {
44                 waktuBerangkat = tsWaktuidealAwal.Add(new
45 TimeSpan(0,60,0));
46                 totalPenalti += 0;
47
48                 temp2[i] = waktuBerangkat.ToString();
49                 tempPenalti[i] = 0;
50             }
51         }
52         else
53         {
54             waktuBerangkat = waktuDatang;
55             temp2[i] = waktuBerangkat.ToString();
56
57             ts = new TimeSpan();
58             ts = waktuBerangkat.Subtract(tsWaktuidealAkhir);
59             tempPenalti[i] = Convert.ToInt32(
60 ts.TotalMinutes);
61
62             totalPenalti +=
63 Convert.ToInt32(ts.TotalMinutes);
64
65         }
66     }
67 }
68 }
69
70 fitnessParents[i] = Math.Round(1 / ((Convert.ToDouble(s.fitness +
71 s.totalWT))), 6);
72

```

Source code 4.4 Proses Perhitungan Fitness

1. Baris 4 penentuan waktu keberangkatan
2. Baris 16 – 29 menentukan waktu tempuh dan kedatangan
3. Baris 35 – 40 Perhitungan kedatangan sebelum waktu ideal akhir kunjungan dan setelah waktu ideal awal kunjungan
4. Baris 44 – 49 Perhitungan kedatangan sebelum waktu ideal akhir kunjungan dan sebelum waktu ideal awal kunjungan
5. Baris 55 – 63 perhitungan kedatangan setelah waktu ideal akhir kunjungan
6. Baris 70 – 71 perhitungan *fitness*

4.2.5. Proses Seleksi

Proses seleksi untuk memilih individu terbaik digunakan metode seleksi *roulette wheel*. Pada seleksi ini individu dipilih berdasarkan kualitasnya. Kualitas individu dihitung dengan probabilitas individu tersebut dan probabilitas kumulatif individu dalam satu generasi. Individu yang terpilih pada seleksi ini akan dijadikan induk pada generasi berikutnya. Jumlah individu yang terpilih sebanyak ukuran populasinya. Proses seleksi *roulette wheel* dapat dilihat pada *source code* 4.5.

```

1  r = new Random();
2
3  individuTerpilih = new string[valPopSize];
4  fitnessTerpilih = new double[valPopSize];
5  penaltiTerpilih = new double[valPopSize];
6  tempRata = 0;
7  int terbaik = 0;
8
9  for (int i = 0; i < valPopSize; i++)
10     {
11         tempDouble = r.NextDouble();
12         x = 0;
13         int terpilih = 0;
14         while (probCum[x] < tempDouble)
15             {
16                 terpilih = x + 1;
17                 x++;
18             }
19
20         tempRata += fitness[terpilih];
21         individuTerpilih[i] = strIndividu[terpilih];
22         fitnessTerpilih[i] = fitness[terpilih];
23         penaltiTerpilih[i] = pinalti[terpilih];
24
25         if (fitnessTerpilih[terbaik] > fitnessTerpilih[i])
26             {
27                 terbaik = terpilih;
28             }
29         else

```

```
30         {
31             terbaik = i;
32         }
33     for (int j = 0; j < totData; j++)
34     {
35         parents[i, j] = newParents[terpilih, j];
36     }
37 }
38 }
39 }
```

Source code 4.5 Proses Seleksi

1. Baris 1 Penentuan nilai random
2. Baris 9 – 23 pemilihan individu dari seleksi
3. Baris 25 – 38 pemilihan individu terbaik dari generasi

4.3. Implementasi User interface

Jumlah halaman pada implementasi *user interface* sistem penentuan rute pariwisata Bali terdiri dari 2 halaman utama. Halaman pertama terdiri berisi input data parameter VRPTW yang berfungsi untuk menentukan objek wisata tujuan beserta *time-windowsnya*. Pada Gambar 4.1 merupakan implementasi *user interface* halaman pertama.

The screenshot shows a Windows-style application window titled 'Form1'. It has a menu bar with 'Data Uji' and 'Hasil'. The main area is divided into two columns. The left column contains three input fields with labels: 'Pilih Lokasi Data .xls' (with a 'Choose File' button), 'Inputkan Nama Sheet Jadwal' (with a 'Jadwal' input and a 'Load File' button), and 'Inputkan Nama Sheet Waktu Tempuh' (with a 'Waktu' input). The right column contains two large empty rectangular areas, the top one labeled 'Jadwal Buka Objek Wisata' and the bottom one labeled 'Waktu Tempuh'. At the bottom right of the window, there is a button labeled 'Selanjutnya >>'.

Gambar 4.1 Implementasi *User interface* Halaman Pertama

Setelah halaman pertama muncul, kemudian tekan tombol *choose file* untuk memilih data yang akan dihitung. Setelah memilih data, tekan tombol *load file*. Maka akan muncul halaman seperti Gambar 4.2

The screenshot shows a software window titled 'Form1' with a 'Data Uji' tab. On the left, there are input fields for 'Pilih Lokasi Data.xls' (set to 'D:\K U L I J'), 'Inputkan Nama Sheet Jadwal' (set to 'Jadwal'), and 'Inputkan Nama Sheet Waktu Tempuh' (set to 'Waktu'). There are 'Choose File' and 'Load File' buttons. On the right, there is a table titled 'Jadwal Buka Objek Wisata' and a spreadsheet titled 'Waktu Tempuh'.

No	ObjekWisata	Jadwal Buka	Jadwal Tutup
0	DEPOT(Hotel K...		
1	Pantai Kuta	12/30/1899 4...	12/30/1899 6...
2	Bedugul	12/30/1899 10...	12/30/1899 2...
3	Ubud	12/30/1899 8...	12/30/1899 11...
4	Jati luwih	12/30/1899 11...	12/30/1899 4...
5	Tirta Empul	12/30/1899 1...	12/30/1899 4...
6	Taman ujung	12/30/1899 2...	12/30/1899 5...
7	Sangeh	12/30/1899 10...	12/30/1899 3...
8	Taman budaya ..	12/30/1899 8...	12/30/1899 8...

A	B	C	D	E	F	G	H	I	J
0	6	96	61	90	81	114	55	2	
6	0	88	46	82	83	116	45	2	
96	88	0	65	33	89	148	46	7	
61	46	65	0	61	27	85	20	2	
90	82	33	61	0	84	148	40	7	
81	83	89	27	84	0	98	45	5	
114	116	148	85	148	98	0	109	9	
55	45	46	20	40	45	109	0	3	
26	21	78	27	72	58	91	33	0	

Gambar 4.2 Tampilan Halaman Pertama Setelah Load Data

Setelah data berhasil dimasukkan, tekan tombol selanjutnya untuk beralih kehalaman perhitungan. Maka akan muncul halaman seperti Gambar 4.3.

The screenshot shows a software window titled 'Form1' with a 'Data Uji' tab. On the left, there are input fields for 'Pop Size' (10), 'Cross Over Rate' (0.4), 'Mutation Rate' (0.2), and 'Maximum Generasi' (100). There are 'Random Populasi' and 'Fix Populasi' buttons. On the right, there is a table titled 'Generasi' with columns for 'Gen', 'Individu Terbaik', 'Pinalti', 'Fitness', and 'Rata-Rata'. Below the table is a large empty area labeled 'Solusi'.

Gen	Individu Terbaik	Pinalti	Fitness	Rata-Rata
*				

Gambar 4.3 Halaman Perhitungan

Untuk melakukan perhitungan menggunakan populasi yang kita tentukan sendiri secara manual maka tekan tombol *fix* populasi. Maka akan muncul kotak

dialog untuk meng-*import* data kromosom dan menentukan *parent* dan titik potong atau titik pertukaran untuk metode *one-cut point crossover* dan *exchange mutation*. Kotak dialog *fix* populasi dapat dilihat pada Gambar 4.4 dan Gambar 4.5.

The screenshot shows a window titled "FormDialog" with a yellow border. At the top, it says "Populasi Awal :". Below this are two buttons: "Import" and "Submit". The main area contains a 10x10 grid of empty rectangular input boxes, numbered 1 through 10 on the left side. The window has standard minimize, maximize, and close buttons in the top right corner.

Gambar 4.4 Kotak Dialog Menentukan Krosomom

The screenshot shows a window titled "FormDialog2" with a yellow border. It is divided into two sections: "Crossover" and "Mutation". Under "Crossover", there are three dropdown menus labeled "valRandom1", "Parent 1", and "Parent 2". Under "Mutation", there are three dropdown menus labeled "valRandom1", "valRandom2", and "Parent 1". At the bottom of the window is a "Submit" button. The window has standard minimize, maximize, and close buttons in the top right corner.

Gambar 4.5 Kotak Dialog Menentukan Parameter *Crossover* dan Mutasi

Setelah tombol submit ditekan maka akan dilakukan perhitungan sesuai dengan populasi awal dan parameter crossover dan mutation yang telah di masukkan. Apabila pada halaman kedua sesuai Gambar 4.3 yang dipilih adalah random

populasi maka akan dilakukan perhitungan dengan populasi acak sesuai dengan jumlah populasi, banyaknya generasi dan kombinasi Cr dan Mr yang telah ditentukan. Hasil perhitungan manual dengan metode GA akan ditampilkan. Kromosom terbaik setiap generasi akan ditampilkan pada halaman kedua. Pada Gambar 4.6 merupakan implementasi *user interface* detail hasil proses algoritma genetika.

Pencarian Rute Pariwisata Bali Menggunakan Algoritma Genetika

Data Uji Hasil

Pop Size : 10

Cross Over Rate : 0.4

Mutation Rate : 0.2

Maximum Generasi : 100

Generasi	Gen	Individu Terbaik	Pinalti	Fitness	Rata-Rata
1	0 4 2 3 1 5 6 7 8		1095	0.000605	0.0004038
2	0 4 2 3 1 5 6 7 8		1095	0.000605	0.0003927
3	0 2 4 3 7 6 5 8 1		534	0.000971	0.0006172
4	0 4 2 3 1 6 5 7 8		1099	0.000615	0.000606
5	0 4 2 3 1 6 5 7 8		1099	0.000615	0.000607

Solusi

Populasi Awal

Generasi 0

1	0 1 2 3 7 6 5 4 8	penalti = 2676	Fitness = 0.000311	TotalWT = 542
2	0 1 2 3 8 4 6 7 5	penalti = 2736	Fitness = 0.000303	TotalWT = 560
3	0 1 2 6 4 7 3 5 8	penalti = 3176	Fitness = 0.000269	TotalWT = 535
4	0 1 2 3 7 5 6 4 8	penalti = 2548	Fitness = 0.000324	TotalWT = 542
5	0 1 3 4 8 6 7 2	penalti = 2473	Fitness = 0.000337	TotalWT = 495
6	0 1 3 4 6 5 7 2	penalti = 2755	Fitness = 0.000312	TotalWT = 452
7	0 1 2 3 4 6 7 5 8	penalti = 2978	Fitness = 0.000281	TotalWT = 580
8	0 4 2 3 1 5 6 7 8	penalti = 1095	Fitness = 0.000605	TotalWT = 557
9	0 1 6 3 4 5 8 7 2	penalti = 2849	Fitness = 0.0003	TotalWT = 489
10	0 6 2 3 4 5 1 7 8	penalti = 2139	Fitness = 0.000361	TotalWT = 633

Create Children From Crossover

Crossover 1: Parents 9 + Parents 6

0 1 6 3 8 4 5 7 2

Crossover 2: Parents 9 + Parents 5

0 1 6 3 4 5 8 7 2

Crossover 3: Parents 9 + Parents 3

0 1 6 3 4 2 7 5 8

Crossover 4: Parents 9 + Parents 6

0 1 6 3 4 8 5 7 2

Gambar 4.6 Tampilan Detail Hasil Perhitungan

BAB V

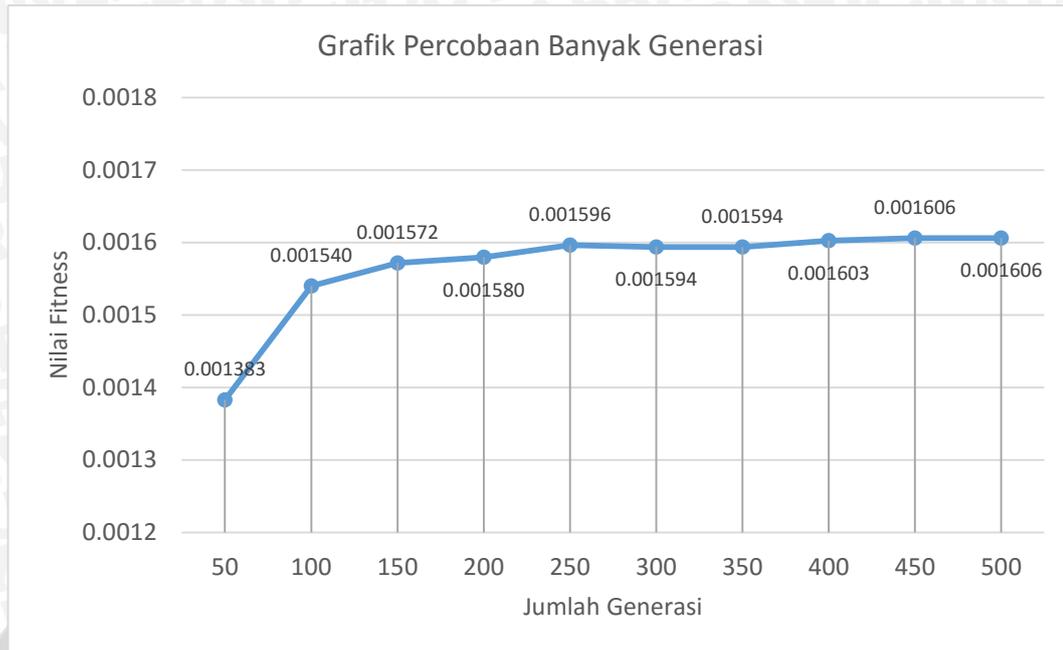
PENGUJIAN DAN ANALISA

5.1. Pengujian dan Analisa Banyaknya Generasi

Pengujian ini dilakukan untuk mengetahui seberapa banyak generasi yang optimal pada permasalahan penentuan rute pariwisata Bali dengan metode VRPTW. Pengujian dilakukan sebanyak 10 kali dengan banyak generasi kelipatan 50 yang dimulai dari 50-500 generasi. Jumlah populasi dalam pengujian sebanyak 10 populasi. Kombinasi Cr dan Mr yang digunakan yaitu 0.5 : 0.5. Proses pengujian menggunakan metode *Roulette Wheel*. Hasil dari setiap percobaan akan didapatkan nilai rata-rata *fitness* untuk mengetahui solusi terbaik dari generasi yang optimal. Pada Tabel 5.1 merupakan Tabel hasil percobaan banyaknya generasi.

Tabel 5.1 Hasil Uji Coba Banyak Generasi

Generasi	Percobaan										Rata Rata <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
50	0.00150	0.00133	0.00144	0.00147	0.00100	0.00138	0.00146	0.00149	0.00119	0.00157	0.00138
100	0.00140	0.00157	0.00157	0.00157	0.00151	0.00162	0.00157	0.00157	0.00149	0.00157	0.00154
150	0.00162	0.00149	0.00151	0.00162	0.00162	0.00162	0.00162	0.00160	0.00162	0.00160	0.00159
200	0.00151	0.00162	0.00162	0.00151	0.00162	0.00162	0.00144	0.00157	0.00157	0.00157	0.00156
250	0.00154	0.00160	0.00162	0.00162	0.00162	0.00162	0.00157	0.00162	0.00162	0.00157	0.00160
300	0.00162	0.00157	0.00162	0.00162	0.00157	0.00162	0.00151	0.00162	0.00157	0.00162	0.00159
350	0.00157	0.00157	0.00162	0.00157	0.00162	0.00160	0.00162	0.00157	0.00162	0.00162	0.00159
400	0.00157	0.00162	0.00162	0.00162	0.00162	0.00160	0.00162	0.00160	0.00157	0.00162	0.00160
450	0.00162	0.00160	0.00162	0.00160	0.00162	0.00162	0.00160	0.00162	0.00157	0.00162	0.00161
500	0.00162	0.00162	0.00160	0.00162	0.00162	0.00162	0.00162	0.00162	0.00162	0.00160	0.00161



Gambar 5.1 Grafik Hasil Uji Coba Banyak Generasi

Gambar 5.1 merupakan grafik hasil uji coba banyak generasi dalam 10 kali percobaan. Pada grafik tersebut perilaku GA terlihat dalam penyelesaian masalah, dimana pada setiap generasi GA selalu memperbaiki nilai *fitness* yang didapat. Dapat dilihat pada generasi 50-500 generasi, nilai *fitness* mengalami peningkatan. Peningkatan yang signifikan berada pada generasi 50-250. Sedangkan untuk generasi 250 ke atas hingga 500 nilai *fitness* yang diperoleh tidak mengalami peningkatan yang signifikan sehingga terjadi konvergensi. Kondisi seperti ini mengakibatkan proses reproduksi menghasilkan *Offspring* yang hampir sama dengan induknya. Dari hasil pengujian ini, terlalu banyak generasi belum tentu menghasilkan solusi yang lebih baik. Solusi yang dihasilkan hampir sama dengan solusi yang diperoleh dari generasi sebelumnya. Maka solusi optimal yang diperoleh pada penelitian ini adalah pada 250 generasi.

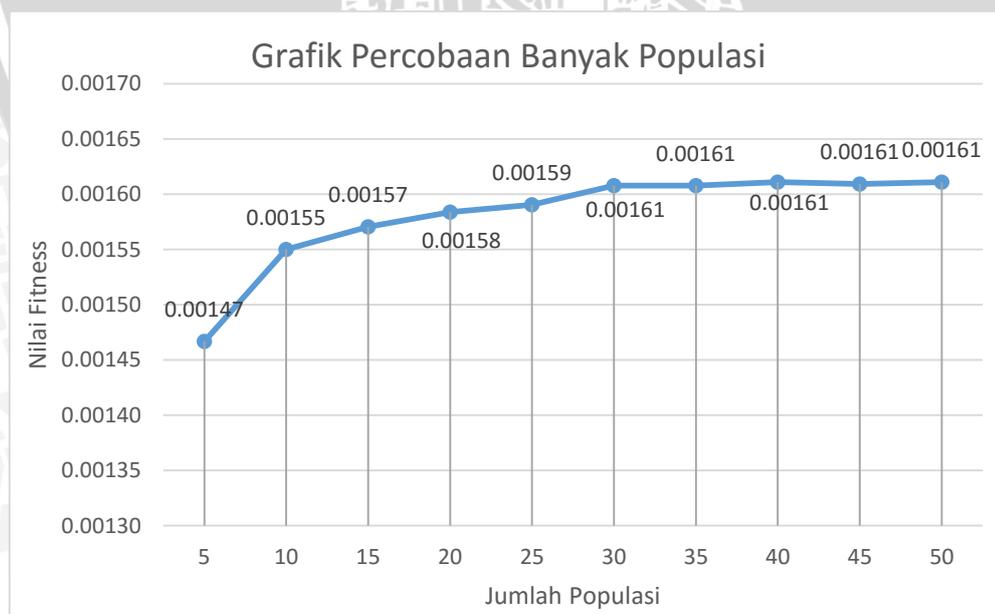
5.2. Pengujian dan Analisa Ukuran Populasi

Pada pengujian ini dilakukan untuk mengetahui banyaknya populasi dengan solusi terbaik pada permasalahan VRPTW untuk penentuan rute pariwisata Bali. Pengujian ini dilakukan sebanyak 10 kali percobaan pada generasi optimal yang diperoleh dari uji coba banyak banyak generasi yaitu 250 generasi. Kombinasi Cr

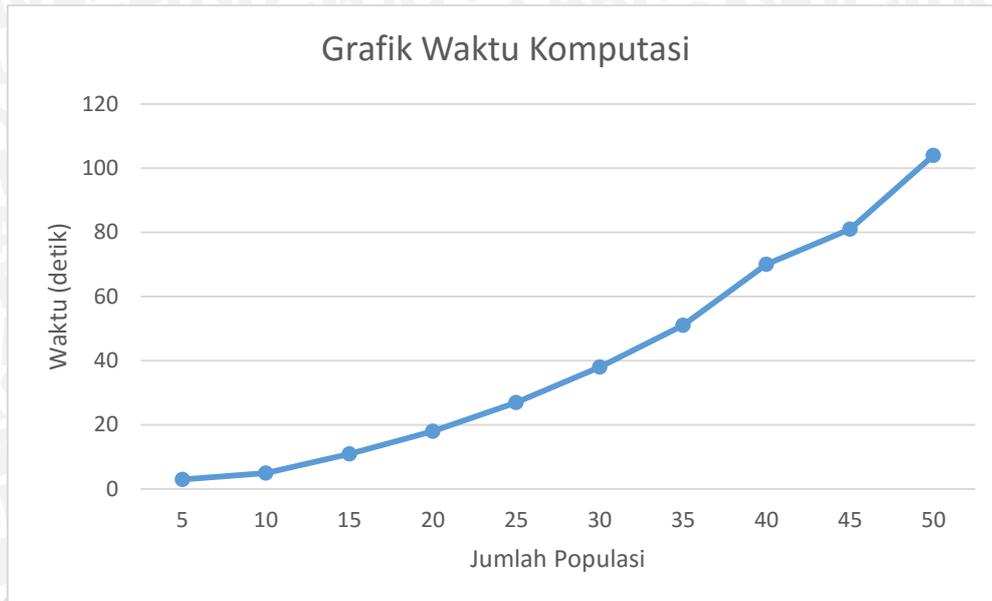
dan Mr yang digunakan yaitu 0.5:0.5. Ukuran populasi yang diujikan adalah kelipatan 5, yaitu dari 5 populasi hingga 50 populasi. Proses pengujian menggunakan metode *Roulette Wheel*. Hasil dari setiap percobaan akan didapatkan nilai rata-rata *fitness* untuk mengetahui solusi terbaik dari ukuran populasi yang optimal. Pada Tabel 5.2 merupakan Tabel hasil percobaan ukuran populasi.

Tabel 5.2 Hasil Uji Coba Banyak Populasi

Banyak Populasi	Nilai <i>Fitness</i>										Rata Rata <i>Fitness</i>	Waktu Komputasi (detik)
	Percobaan											
	1	2	3	4	5	6	7	8	9	10		
5	0.00147	0.00157	0.00151	0.00149	0.00151	0.00112	0.00149	0.00135	0.00150	0.00147	0.00145	3
10	0.00146	0.00160	0.00147	0.00160	0.00144	0.00157	0.00162	0.00150	0.00160	0.00154	0.00154	5
15	0.00157	0.00162	0.00162	0.00160	0.00149	0.00157	0.00157	0.00162	0.00152	0.00160	0.00158	11
20	0.00162	0.00157	0.00162	0.00151	0.00157	0.00160	0.00157	0.00157	0.00157	0.00160	0.00158	18
25	0.00157	0.00162	0.00162	0.00157	0.00160	0.00151	0.00162	0.00162	0.00157	0.00157	0.00158	27
30	0.00157	0.00162	0.00162	0.00162	0.00162	0.00162	0.00157	0.00162	0.00157	0.00157	0.00160	38
35	0.00157	0.00162	0.00162	0.00162	0.00157	0.00162	0.00160	0.00157	0.00162	0.00162	0.00160	51
40	0.00162	0.00162	0.00157	0.00162	0.00157	0.00162	0.00162	0.00162	0.00157	0.00160	0.00160	70
45	0.00162	0.00162	0.00162	0.00162	0.00162	0.00162	0.00157	0.00162	0.00162	0.00162	0.00161	81
50	0.00162	0.00162	0.00162	0.00162	0.00157	0.00162	0.00162	0.00162	0.00162	0.00162	0.00161	104



Gambar 5.2 Grafik Uji Coba Banyak Populasi



Gambar 5.3 Grafik Waktu Komputasi

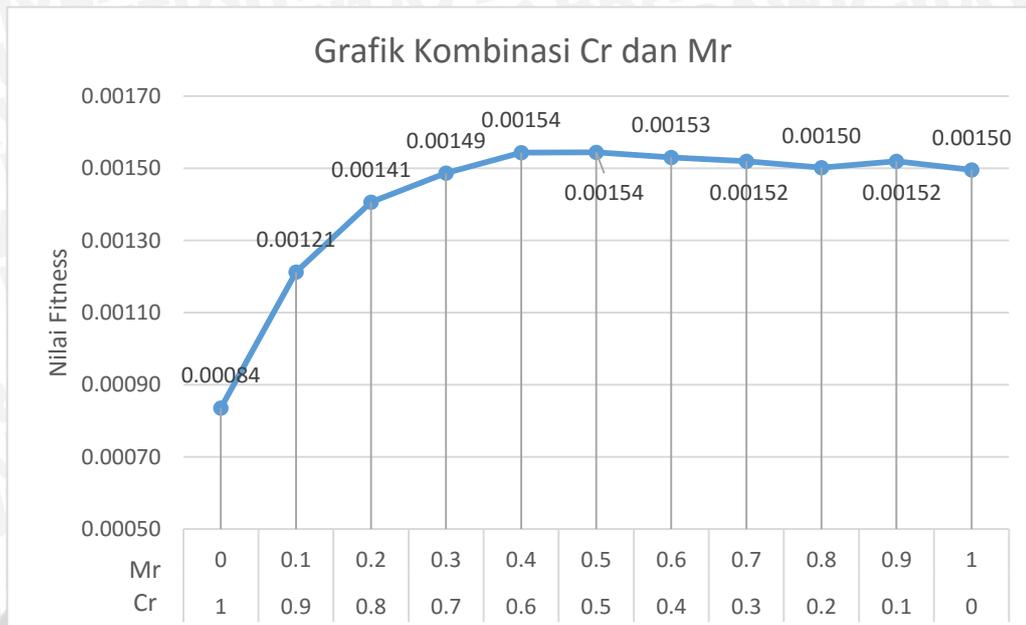
Pada Gambar 5.2 merupakan grafik hasil uji coba banyak populasi dalam 10 kali percobaan. Percobaan pada setiap populasi selalu mengalami peningkatan nilai *fitness*. Kenaikan nilai *fitness* yang signifikan terjadi pada jumlah populasi 5 hingga 30 populasi, sedangkan pada ukuran populasi 30 hingga 50 tidak mengalami kenaikan *fitness* yang signifikan sehingga terjadi konvergensi. Pada kondisi konvergensi, proses eksplorasi tidak berjalan dengan baik sehingga saat reproduksi, *Offspring* yang dihasilkan akan mirip dengan induknya. Semakin banyak jumlah populasinya, maka jumlah anak yang dihasilkan juga semakin banyak, karena hasil reproduksi dipengaruhi oleh jumlah populasi, nilai *Cr* dan *Mr*. Keragaman individu dalam suatu populasi mempengaruhi solusi yang dihasilkan. Semakin beragamnya individu maka cakupan ruang pencarian yang lebih luas dan kompleks [14]. Waktu komputasi yang diperlukan berbanding lurus dengan jumlah populasinya. Semakin banyak jumlah populasi maka waktu komputasi yang diperlukan juga semakin lama, begitu pula sebaliknya. Grafik waktu komputasi dapat dilihat pada Gambar 5.3. Dari hasil percobaan ini, jumlah populasi yang terlalu banyak belum tentu memberikan solusi yang lebih baik, serta membutuhkan waktu komputasi yang lebih lama. Dari hasil grafik permasalahan pencarian rute pariwisata Bali menunjukkan bahwa jumlah populasi yang optimal dihasilkan pada 30 populasi.

5.3. Pengujian dan Analisa Kombinasi Cr dan Mr

Pada pengujian ini dilakukan untuk mengukur kombinasi *Crossover rate* dan *mutation rate* yang paling tepat pada permasalahan VRPTW untuk penentuan rute pariwisata Bali. Pengujian ini dilakukan karena pada permasalahan yang berbeda, kombinasi Cr dan Mr yang diperlukan juga berbeda-beda. Hal ini harus disesuaikan dengan kasus yang ingin diselesaikan. Kombinasi Cr dan Mr yang baik akan menghasilkan nilai rata-rata *fitness* yang tinggi. Pengujian ini dilakukan sebanyak 10 kali percobaan pada generasi optimal yang didapat dari percobaan banyak generasi yaitu 250 generasi. Dan dengan ukuran populasi sebanyak 30 populasi, yaitu populasi optimal yang diperoleh dari percobaan banyak populasi. Nilai Cr dan Mr diujikan berskala 0 hingga 1. Proses pengujian menggunakan metode *Roulette Wheel*. Hasil dari setiap percobaan didapatkan nilai rata-rata *fitness* untuk mengetahui kombinasi Cr dan Mr yang optimal dari masing-masing kombinasi. Pada Tabel 5.3 merupakan tabel hasil percobaan kombinasi Cr dan Mr .

Tabel 5.3 Hasil Uji Coba Kombinasi Cr dan Mr

Nilai Cr	Nilai Mr	Nilai Fitness										Rata Rata Fitness
		Percobaan										
		1	2	3	4	5	6	7	8	9	10	
1	0	0.00114	0.00107	0.00073	0.00064	0.00071	0.00081	0.00047	0.00087	0.00087	0.00107	0.00084
0.9	0.1	0.00144	0.00078	0.00078	0.00110	0.00126	0.00123	0.00123	0.00149	0.00144	0.00136	0.00121
0.8	0.2	0.00143	0.00144	0.00120	0.00120	0.00162	0.00142	0.00126	0.00144	0.00157	0.00151	0.00141
0.7	0.3	0.00142	0.00142	0.00146	0.00144	0.00162	0.00151	0.00151	0.00160	0.00138	0.00151	0.00149
0.6	0.4	0.00160	0.00160	0.00143	0.00162	0.00143	0.00143	0.00151	0.00162	0.00160	0.00160	0.00154
0.5	0.5	0.00151	0.00157	0.00151	0.00150	0.00162	0.00162	0.00157	0.00138	0.00162	0.00157	0.00154
0.4	0.6	0.00144	0.00157	0.00154	0.00154	0.00146	0.00160	0.00151	0.00162	0.00143	0.00160	0.00153
0.3	0.7	0.00157	0.00143	0.00152	0.00157	0.00151	0.00157	0.00157	0.00142	0.00149	0.00157	0.00152
0.2	0.8	0.00162	0.00160	0.00126	0.00160	0.00157	0.00151	0.00146	0.00151	0.00146	0.00143	0.00150
0.1	0.9	0.00146	0.00152	0.00151	0.00157	0.00160	0.00152	0.00147	0.00149	0.00154	0.00151	0.00152
0	1	0.00162	0.00154	0.00132	0.00147	0.00149	0.00162	0.00138	0.00154	0.00138	0.00162	0.00150



Gambar 5.4 Grafik Hasil Uji Coba Kombinasi *Cr* dan *Mr*

Pada Gambar 5.4 merupakan grafik hasil uji coba kombinasi *crossover rate* dan *mutation rate* dalam 10 kali percobaan. Rata-rata nilai *fitness* yang diperoleh dalam penelitian ini sangat bervariasi. Tidak ada suatu ketetapan nilai *Cr* maupun *Mr* yang digunakan untuk memperoleh solusi optimal. Suatu permasalahan sangat mempengaruhi nilai kombinasi yang tepat [14]. Dalam permasalahan VRPTW ini, terjadi peningkatan nilai *fitness* yang signifikan dari kombinasi *Cr* dan *Mr* 1:0 hingga 0.6:0.4. Pada kombinasi *Cr* dan *Mr* 0.6:0.4 hingga 0:1, tidak terjadi perubahan nilai *fitness* yang signifikan, bahkan terjadi penurunan. Kombinasi nilai *Cr* dan *Mr* yang berbeda-beda tidak terlalu mempengaruhi lamanya komputasi, yang dipengaruhi adalah variasi *Offspring* yang dihasilkan. Pada percobaan ini kombinasi dengan hasil *fitness* terendah terdapat pada *crossover rate* 1 dan *mutation rate* 0 dengan rata-rata *fitness* 0.00084. Solusi optimal diperoleh pada kombinasi *crossover rate* 0.6 dan *mutation rate* 0.4 dengan rata-rata *fitness* 0.00154.

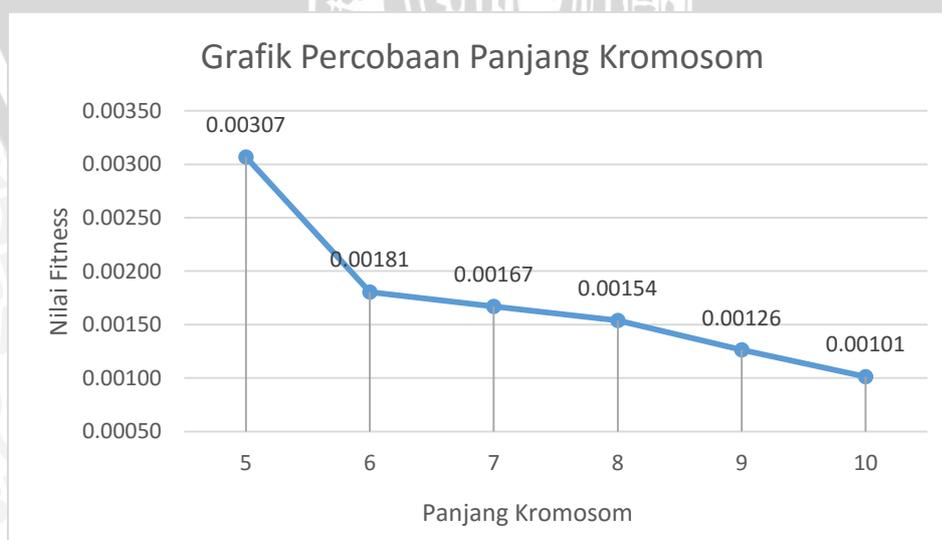
5.4. Pengujian dan Analisa Panjang Kromosom

Pada pengujian ini dilakukan untuk membandingkan hasil rata-rata *fitness* yang dihasilkan untuk beberapa panjang kromosom. Panjang kromosom menunjukkan jumlah objek wisata yang dapat dikunjungi dalam satu hari. Dalam pengujian ini

dilakukan percobaan 10 kali dengan jumlah populasi 10, yaitu populasi optimal yang diperoleh dari hasil percobaan banyak populasi. Jumlah generasi yang digunakan adalah sebanyak jumlah generasi optimal yang diperoleh dari hasil percobaan banyak generasi yaitu 250 generasi. Kombinasi nilai Cr dan Mr yang digunakan adalah 0.6:0.4, merupakan kombinasi terbaik yang diperoleh dari percobaan kombinasi Cr dan Mr . Panjang kromosom yang diujikan yaitu dari 5 hingga 10. Perubahan panjang kromosom ini dilakukan dengan menambahkan atau mengurangi jumlah objek wisata (gen) yang akan dikunjungi. Pemilihan tujuan objek wisata dilakukan secara random. Proses pengujian menggunakan metode *Roulette Wheel*. Hasil dari setiap percobaan didapatkan nilai rata-rata *fitness* untuk mengetahui panjang kromosom yang optimal. Pada Tabel 5.4 merupakan Tabel hasil percobaan panjang kromosom.

Tabel 5.4 Hasil Uji Coba Panjang Kromosom

Panjang Kromosom	Nilai <i>Fitness</i>										Rata Rata <i>Fitness</i>	Node Yang Terlewati
	Percobaan											
	1	2	3	4	5	6	7	8	9	10		
5	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0.00307	0
6	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	0.00181	1
7	0.00167	0.00167	0.00166	0.00167	0.00167	0.00167	0.00167	0.00166	0.00167	0.00167	0.00167	2
8	0.00146	0.00160	0.00147	0.00160	0.00144	0.00157	0.00162	0.00150	0.00160	0.00154	0.00154	3
9	0.00147	0.00110	0.00110	0.00147	0.00123	0.00123	0.00150	0.00145	0.00110	0.00097	0.00126	4
10	0.00082	0.00119	0.00075	0.00128	0.00090	0.00129	0.00091	0.00096	0.00113	0.00091	0.00101	5



Gambar 5.5 Grafik Hasil Uji Coba Panjang Kromosom

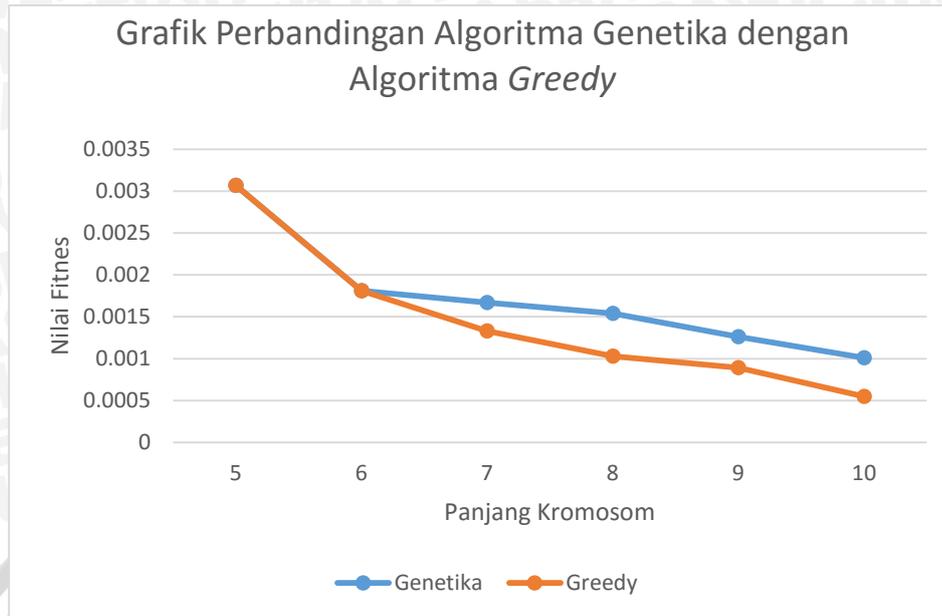
Pada Gambar 5.5 merupakan grafik hasil uji coba panjang kromosom dalam 10 kali percobaan. Rata-rata nilai *fitness* yang diperoleh dalam penelitian ini terjadi penurunan seiring bertambahnya panjang kromosom. Penurunan ini terjadi karena dengan bertambahnya panjang kromosom, maka lebih banyak objek wisata yang harus dikunjungi dalam waktu satu hari. Sedangkan setiap objek wisata memiliki batasan waktu kunjungan ideal, sehingga banyak objek wisata yang tidak dapat dicapai atau dikunjungi dalam waktu satu hari. Hal ini akan memperbesar nilai *penalty* dan memperkecil nilai *fitness* yang dihasilkan. Dalam permasalahan VRPTW ini, nilai rata-rata *fitness* yang paling besar dan optimal yaitu pada panjang kromosom 5 dengan nilai 0.00307 dimana tidak ada node yang terlewat atau semua objek wisata tujuan dapat dikunjungi.

5.5. Pengujian dan Analisa Perbandingan Algoritma Genetika dengan Algoritma *Greedy*

Pada pengujian ini dilakukan untuk membandingkan solusi yang dihasilkan oleh algoritma genetika dengan algoritma *greedy*. Jumlah populasi yang digunakan adalah populasi optimal yang diperoleh dari hasil percobaan banyak populasi. Jumlah generasi yang digunakan adalah sebanyak jumlah generasi optimal yang diperoleh dari hasil percobaan banyak generasi. Kombinasi nilai *Cr* dan *Mr* yang digunakan adalah kombinasi terbaik yang diperoleh dari percobaan kombinasi *Cr* dan *Mr*. Panjang kromosom yang diujikan yaitu dari 5 hingga 10. Hasil dari setiap percobaan didapatkan nilai rata-rata *fitness* dari algoritma genetika dan nilai *fitness* dari algoritma *greedy*. Pada Tabel 5.5 merupakan Tabel hasil percobaan perbandingan algoritma genetika dengan algoritma *greedy*.

Tabel 5.5 Hasil Uji Coba Perbandingan Genetika dan *Greedy*

Panjang Kromosom	Nilai <i>Fitness</i>	
	Genetika	<i>Greedy</i>
5	0.00307	0.00307
6	0.00181	0.00181
7	0.00167	0.00133
8	0.00154	0.00103
9	0.00126	0.00089
10	0.00101	0.00055



Gambar 5.6 Grafik Perbandingan Algoritma Genetika dengan *Greedy*

Solusi yang lebih baik adalah solusi yang memiliki nilai *fitness* lebih besar. Pada Gambar 5.6 dapat dilihat perbandingan solusi yang dihasilkan oleh algoritma Genetika dan algoritma *Greedy*. Pada panjang kromosom 5 dan 6 nilai *fitness* yang dihasilkan oleh kedua algoritma adalah sama. Tetapi pada panjang kromosom 7 sampai dengan 10 nilai *fitness* yang dihasilkan oleh algoritma genetika lebih besar. Dari hasil tersebut dapat dikatakan algoritma genetika lebih efektif digunakan untuk menyelesaikan permasalahan dengan data yang besar. Nilai efisiensi algoritma genetika jika dibandingkan dengan algoritma *greedy* adalah 16.22%. Nilai efisiensi ini menunjukkan bahwa algoritma genetika menghasilkan hasil yang lebih optimal dibandingkan dengan algoritma *greedy*.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil uji coba mengenai penerapan algoritma genetika pada permasalahan VRPTW pada kasus pariwisata Bali, terdapat beberapa kesimpulan yaitu:

1. Algoritma genetika dapat menyelesaikan permasalahan VRPTW pada kasus pariwisata Bali. Implementasi dilakukan dengan langkah pertama membentuk populasi awal menggunakan representasi permutasi dalam bentuk integer sepanjang gen berdasarkan data objek wisata. Langkah berikutnya yaitu melakukan proses reproduksi menggunakan metode *one cut point crossover* dan *Reciprocal Exchange Mutation*. Kemudian dilakukan perhitungan *fitness* dan melakukan seleksi menggunakan metode *Roulette Wheel*. Hasil *fitness* terbaik yang diperoleh yaitu dengan waktu tempuh dan jumlah pinalti yang minimum. Hasil akhirnya berupa kromosom yang menunjukkan rute kunjungan objek wisata. Nilai efisiensi yang dihasilkan oleh algoritma genetika dalam menyelesaikan permasalahan VRPTW ini adalah 16.22%.
2. Perubahan parameter algoritma genetika mempengaruhi rata-rata hasil *fitness*. Pada data berukuran besar (generasi dan populasi), algoritma genetika memperoleh solusi yang optimal dan waktu proses yang sebanding. Solusi optimal diperoleh pada nilai *fitness* yang dihasilkan setelah terjadinya konvergensi. Nilai optimal yang dihasilkan dari penelitian ini terdapat pada jumlah populasi 30 dengan banyak generasi 250, dan nilai $Cr : 0.6$ dan $Mr : 0.4$ dengan rata-rata *fitness* 0.00307 pada panjang kromosom 5. Hal ini dapat dikatakan bahwa jumlah kunjungan ideal di Bali dalam sehari adalah 5 objek wisata.

6.2. Saran

Berdasarkan hasil penelitian mengenai penerapan algoritma genetika pada permasalahan VRPTW pada kasus pariwisata Bali, terdapat beberapa saran yaitu:

1. Pada penelitian lebih lanjut dapat dikembangkan untuk menyelesaikan permasalahan penentuan rute objek wisata dengan menggunakan metode *crossover*, metode mutasi, dan metode seleksi yang lainnya. Sehingga mempengaruhi nilai *fitness* individu untuk menghasilkan hasil yang lebih optimal.
2. Pada penelitian lebih lanjut agar menambahkan jumlah objek wisata yang dapat dikunjungi dan jarak yang bervariasi dengan memperhatikan perhitungan jarak tempuh sesuai kondisi jalan satu arah dan arah berlawanan. Dalam perhitungannya dapat menggunakan matriks jarak atau waktu tempuh yang asimetris.
3. Pada penelitian lebih lanjut dapat dikembangkan dengan memasukkan lama waktu (dalam hari) untuk mengunjungi objek wisata. Hal ini dilakukan untuk menambah jumlah objek wisata yang dapat dikunjungi.



Daftar Pustaka

- [1] Pusdatin Kemenparekraf & BPS. *Perkembangan Wisatawan Nusantara*. 2014. URL: <http://www.parekraf.go.id/asp/detil.asp?c=111&id=1191>, diakses tanggal 23 Januari 2015.
- [2] Dinas Pariwisata Provinsi Bali. *Perkembangan Kunjungan Wisatawan ke Bali*. 2014. URL: <http://www.disparda.baliprov.go.id/id/Statistik2>, diakses tanggal 23 Januari 2015.
- [3] Gambardella, L.M., Taillard, E., & Agazzi, G. 1999. *A Multiple Ant Colony Sistem For Vehicle Routing Problems With Time Windows*. New Ideas in Optimization
- [4] Yeun, L.C., Ismail, R.W., Omar, K & Zirous, M. 2008. *Vehicle Routing Problem : Model and Solution*. Journal of Quality Measurement and Analysis. 205-218.
- [5] Saputro, Rio. 2009. *Travelling Salesman Problem Dengan Kendala Time Window Pada Perusahaan Jasa Pengantar Barang Menggunakan Algoritma Genetika*.
- [6] Rennard JP. Genetic Algorithms Viewer 1.0. 2000. URL: <http://www.rennard.org/alife/english/gavgb.html>, diakses tanggal 29 Januari 2015.
- [7] Chen, Tinggui, & Zhou Guanglan. 2012. *Vehicle Routing Optimization Problem with Time-windows and its Solution by Genetic Algorithm*. Journal of Digital Information Management
- [8] Kadar, Abdul. 2011. *Solving the Vehicle Routing Problem using Genetic Algorithm*. International Journal of Advanced Computer Science and Applications
- [9] Suwantoro, Gamal. 2002. *Dasar-Dasar Pariwisata*. Yogyakarta: Andi. Offset.
- [10] Putri, A. N (2008). *Penentuan Rute Optimal Pengangkutan Sampah Menggunakan Algoritma Genetika*. Malang: Universitas Brawijaya

- [11] Kallehauge, B. L., & Madsen, O.B. (2001). *Lagrangean Duality Applied On Vehicle Routing With Time Windows*. Lyngby: Technical University of Denmark.
- [12] Setiawan, K. (2003). *Pradigma Sistem Cerdas*. Surabaya: Bayumedia.
- [13] Suyanto. (2005). *Algoritma Genetika Dalam Matlab*. Yogyakarta: Andi.
- [14] Mahmudi, W. F. (2013). *Algoritma Evolusi Program Teknologi Informasi dan Ilmu Komputer*. Malang: Universitas Brawijaya.
- [15] Kusumadewi, S. (2003). *Artificial Intelegence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- [16] Syamsi, Badrus. (2002). *Penerapan Algoritma Genetik untuk Meningkatkan Performansi Pengendalian PID pada Sistem Pengendalian Kecepatan Motor Induksi 3 Fase*. ITS. Surabaya.
- [17] Dinas Perhubungan Provinsi Bali. *Area Traffic Control*. 2015. URL : <http://www.atcs.baliprov.go.id>, diakses tanggal 25 Juni 2015.



Lampiran

Data objek wisata

No	Nama Objek Wisata	Waktu Ideal Kunjungan		Durasi Kunjungan
		Awal	Akhir	
1	Pantai Kuta	16:00	18:00	1:00:00
2	Bedugul	10:00	14:00	1:00:00
3	Ubud	8:00	11:00	1:00:00
4	jati luwih	11:00	16:00	1:00:00
5	Tirta Empul	13:00	16:00	1:00:00
6	taman ujung	14:00	17:00	1:00:00
7	penglipuran	15:00	17:00	1:00:00
8	Sanur	5:00	7:00	1:00:00
9	Pantai Pandawa	17:00	18:00	1:00:00
10	GWK	8:00	14:00	1:00:00
11	Sangeh	10:00	15:00	1:00:00
12	Kintamani	5:00	16:00	1:00:00
13	Batur	14:00	16:00	1:00:00
14	Lovina	5:00	7:00	1:00:00
15	Tanah Lot	16:00	18:00	1:00:00
16	Taman Budaya Arda Candra	8:00	20:00	1:00:00

Tabel Data Jarak Antar Objek Wisata

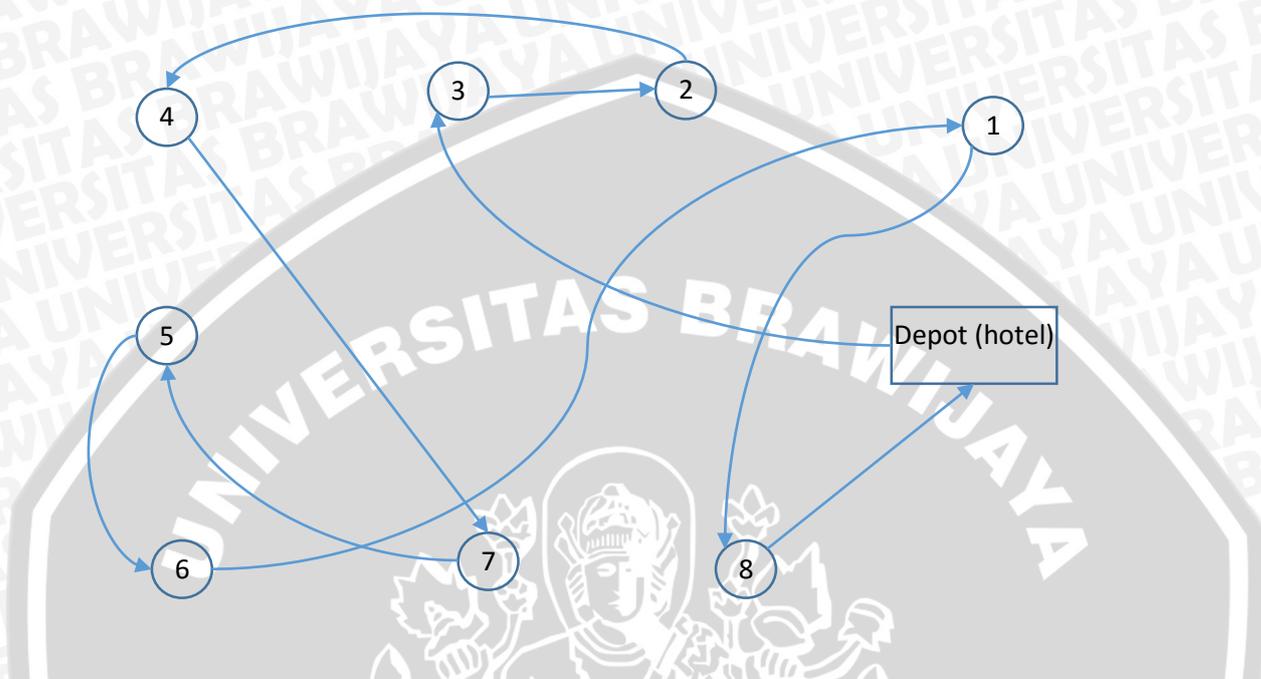
No	Nama Objek Wisata	Jarak (km)	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		S	0	4.2	63.8	40.8	60.2	54.3	75.8	51.3	15.7	17.9	10.8	36.5	67.4	73.8	89.9	26.2	17.2
1	Pantai Kuta	1	4.2	0	58.4	30.8	54.8	55.6	77.1	57.5	17	24	16.9	30	64.2	81	86.3	21.1	13.8
2	Bedugul	2	63.8	58.4	0	22.3	59.6	98.8	68.2	68.2	57.9	77.1	70	30.5	58.4	64.8	26.6	50.8	51.7
3	Ubud	3	40.8	30.8	43.2	0	40.5	17.7	56.8	26.2	21.4	48	41.7	13.4	35.5	39.3	69.2	31.4	17.9
4	ijati luwih	4	60.2	54.8	22.3	40.5	0	56	98.4	64.5	54.3	80.8	67.1	26.9	66.4	72.8	48.3	42.4	48
5	Tirta Empul	5	54.3	55.6	59.6	17.7	56	0	65.3	15.8	39.9	66.4	60.1	29.8	21.6	25.5	77.7	47.9	38.9
6	taman ujung	6	75.8	77.1	98.8	56.8	98.4	65.3	0	47.3	61.4	87.9	81.6	72.9	55.7	63	102	79.6	60.4
7	penglipuran	7	51.3	57.5	68.2	26.2	64.5	15.8	47.3	0	41.6	68.1	61.8	38.4	20.3	24.2	78.4	55.2	40.6
8	Sanur	8	15.7	17	57.9	21.4	54.3	39.9	61.4	41.6	0	27.6	21.4	27.9	62.2	66.1	83.9	27.1	6.8
9	Pantai Pandawa	9	17.9	24	77.1	48	80.8	66.4	87.9	68.1	27.6	0	9.7	49.1	88.7	92.6	104	39.6	33.3
10	GWK	10	10.8	16.9	70	41.7	67.1	60.1	81.6	61.8	21.4	9.7	0	42.7	83	86.9	97.3	33.5	27.6
11	Sangeh	11	36.5	30	30.5	13.4	26.9	29.8	72.9	38.4	27.9	49.1	42.7	0	46.1	62.3	57	28	21.8
12	Kintamani	12	67.4	64.2	58.4	35.5	66.4	21.6	55.7	20.3	62.2	88.7	83	46.1	0	9	60.4	65.1	60.1
13	Batur	13	73.8	81	64.8	39.3	72.8	25.5	63	24.2	66.1	92.6	86.9	62.3	9	0	68.7	71.9	64
14	Lovina	14	89.9	86.3	26.6	69.2	48.3	77.7	102	78.4	83.9	104	97.3	57	60.4	68.7	0	76.8	78.4
15	Tanah Lot	15	26.2	21.1	50.8	31.4	42.4	47.9	79.6	55.2	27.1	39.6	33.5	28	65.1	71.9	76.8	0	21
16	Taman Budaya Arda Cand	16	17.2	13.8	51.7	17.9	48	38.9	60.4	40.6	6.8	33.3	27.6	21.8	60.1	64	78.4	20.7	0

Tabel Data Waktu Tempuh Antar Objek Wisata

Nama Objek Wisata	Waktu (menit)	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Pantai Kuta	1	6	0	96	61	90	81	114	77	24	27	16	55	101	111	135	39	26
Bedugul	2	96	88	0	65	33	89	148	102	87	116	105	46	96	122	129	32	21
Ubud	3	61	46	65	0	61	27	85	39	32	72	63	20	53	59	104	47	27
Jati luwih	4	90	82	33	61	0	84	148	97	81	121	101	40	100	109	72	64	72
Tirta Empul	5	81	83	89	27	84	0	98	24	60	100	90	45	32	38	117	72	58
taman ujung penglipuran	6	114	116	148	85	148	98	0	71	92	132	122	109	84	95	153	119	91
Sanur	7	77	86	102	39	97	24	71	0	62	102	93	58	30	36	118	83	61
Pantai Pandawa	8	24	26	87	32	81	60	92	62	0	41	32	42	93	99	126	41	10
GWK	9	27	36	116	72	121	100	132	102	41	0	15	74	133	139	156	59	50
Sangeh	10	16	25	105	63	101	90	122	93	32	15	0	64	125	130	146	50	41
Kintamani	11	55	45	46	20	40	45	109	58	42	74	64	0	69	93	86	42	33
Batur	12	101	96	88	53	100	32	84	30	93	133	125	69	0	14	91	98	90
Tanah Lot	13	111	122	97	59	109	38	95	36	99	139	130	93	14	0	103	108	96
Taman Budaya Arda Candra	15	39	32	76	47	64	72	119	83	41	59	50	42	98	108	115	0	32
	16	26	21	78	27	72	58	91	61	10	50	41	33	90	96	118	31	0

Peta Hasil Uji Coba

1. Hasil Optimal Uji Coba Banyak Generasi, Jumlah Populasi dan Kombnasi Cr dan Mr . Jalur yang dihasilkan sama yaitu dari depot 0 – 3 – 2 – 4 – 7 – 5 – 6 – 1 – 8



2. Hasil Optimal Uji Coba Panjang Kromosom, yaitu 5 kromosom. Dimulai dari depot 0 – 3 – 2 – 4 – 5 – 1

