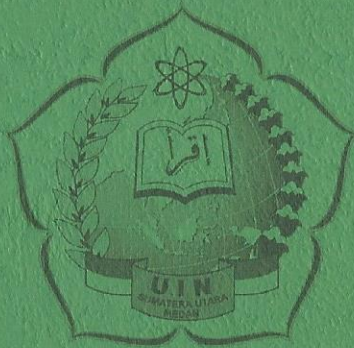


12 / LP / FST / 04 / 2018

Penelitian

**APLIKASI PENDUKUNG PENCARIAN RUTE
TERPENDEK KOTA MEDAN DENGAN
ALGORITMA GENETIKA**



Oleh

**ABDUL HALIM HASUGIAN, M.Kom
NIDN : 0127038801**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2017**

Penelitian

**APLIKASI PENDUKUNG PENCARIAN RUTE
TERPENDEK KOTA MEDAN DENGAN
ALGORITMA GENETIKA**



Oleh

**ABDUL HALIM HASUGIAN, M.Kom
NIDN : 0127038801**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2017**

REKOMENDASI

Setelah membaca dan menelaah hasil penelitian yang berjudul "Aplikasi Pendukung pencarian Rute Terpendek Kota Medan Dengan Algoritma Genetika". Yang dilakukan oleh Abdul Halim Hasugian, M.Kom maka saya berkesimpulan bahwa hasil penelitian ini dapat diterima sebagai karya tulis berupa hasil penelitian. Demikianlah rekomendasi diberikan kepada yang bersangkutan untuk dapat dipergunakan sebagaimana mestinya.

Medan, April 2017
Konsultan



Mhd. Furqan, S.Si., M.Comp.Sc
NIP.19800806 200604 1 003

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan segala kerendahan hati, penulis sampaikan puji syukur kepada Allah SWT berkat Rahmat dan Hidayah-Nya memberi kesehatan, pengetahuan dan kesempatan kepada penulis sehinggadapat menyelesaikan penelitianini yang berjudul“**Aplikasi Pendukung pencarian Rute Terpendek Kota Medan Dengan Algoritma Genetika**”.

Dalam menyelesaikan penelitian ini banyak bantuan bimbingan dari berbagaipihak, baik berupa materil, spiritual, maupun informasi. Sehingga penelitian ini dapat diselesaikan. Maka selayaknya penulis mengucapkan terima kasih kepada :

1. Bapak Dr. H.M. Jamil, MA selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
2. Ibu Dr. Rina Filia Sari, M.Si selaku Wakil Dekan I Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
3. Bapak MHD.Furqan S.Si, M.Comp.Sc selaku Kaprodi Ilmu Komputer Fakultas Sains danTeknologi UIN Sumatera Utara Medan sekaligus Konsultan pada penelitian ini.
4. Bapak/ibu rekan rekan dosen tetap Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.

Menyadari kekurangan dan keterbatasan pada penelitian ini, maka penulis tetap mengharapkan kritik dan saran dari berbagai pihak agar penelitian ini bisa dikembangkan dikemudian hari. Akhir kata semoga penelitian ini bermanfaat bagi kita semua dan Semoga Allah SWT berkenan memberikan berkahnya sehingga semua harapan dan cita-cita penulis dapat terkabulkan. Amin

Medan, April 2017

Abdul Halim Hasugian, M.Kom

2.3	Microsoft Visual Basic.....	30
2.3.1	Menu Bar.....	30
2.3.2	Standard Bar.....	30
2.3.3	ToolBox.....	31
2.3.4	Form.....	31
2.3.5	Project Window.....	32
BAB III	ANALISA DAN PERANCANGAN.....	37
3.1	Analisa Sistem.....	37
3.1.1.	Gambar Peta Kota Medan.....	37
3.1.2	Algoritma Penyelesaian.....	38
3.1.3	Contoh Lain Pencarian Jalur Pendek.....	44
3.2	Perancangan.....	49
3.2.1.	Menu Utama Sistem.....	49
3.2.2	Form Jarak dan Peta.....	50
3.2.3	Form Koordinat Peta.....	52
3.2.4	Form Peta.....	53
3.2.5	Perancangan Basis Data.....	53
3.2.6	Flowchart.....	54
BAB IV	Algoritma dan Implemenstasi.....	56
4.1	Algoritma.....	56
4.1.1	Algoritma Pada Pencarian Jalur Terpendek.....	56
4.1.2	Algoritma Pada Form Jarak dan Jalan.....	56
4.1.3	Algoritma Pada Form Koordinat Peta.....	57
4.2	Implementasi.....	58
4.2.1	Perangkat Keras.....	58
4.2.2	<i>Perangkat Lunak</i>	59
4.2.4	Form Jarak dan Peta.....	60
4.2.5	Form Koordinat.....	60
4.2.6	Form Peta.....	61
4.2.7	Form Pencarian Jalur Terpendek.....	62

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan.....	63
5.2	Rekomendasi.....	63

DAFTAR PUSTAKA

DAFTAR TABEL

Table 2.1 AI Vs Pemrograman Konvensional.....	18
Table 3.1 Jarak Antar Objek.....	40
Table 3.2 Arc Penambah Subtour ke 1	41
Table 3.3 Arc Penambah Subtour ke 2	30
Table 3.4 Arc Penambah Subtour ke 3	35
Table 3.5 Jarak.....	53
Tabel 3.6 Proses.....	42

DAFTAR GAMBAR

Gambar 2.1	Sistem yang Menggunakan AI.....	11
Gambar 2.2	Teknik Pencarian BFS.....	19
Gambar 2.3	Contoh Graf yang akan ditelusuri.....	21
Gambar 2.4	Prosedur Pencarian Mendalam Pertama.....	24
Gambar 2.5	Menu Bar.....	30
Gambar 2.6	Standard Bar.....	31
Gambar 2.7	ToolBox.....	31
Gambar 2.8	Form.....	31
Gambar 2.9	Project windows.....	32
Gambar 2.10	Propertis.....	33
Gambar 3.1	Peta Kota Medan.....	37
Gambar 3.2	Subtour.....	39
Gambar 3.3	Contoh Keterhubungan antar titik dalam algoritma genetik.....	45
Gambar 3.4	Contoh Kasus Genetika.....	45
Gambar 3.5	Contoh Kasus Genetika langkah 2	46
Gambar 3.6	Contoh Kasus Genetika langkah 3	46
Gambar 3.7	Contoh Kasus Genetika langkah 4	47
Gambar 3.8	Contoh Kasus Genetika langkah 5.....	48
Gambar 3.9	Menu Utama Sistem	50
Gambar 3.10	Form Jarak dan Peta	51
Gambar 3.11	Form Koordinat Peta	52
Gambar 3.12	Form Peta	53
Gambar 3.13	Flowchart	55
Gambar 4.1	Menu Utama.....	59
Gambar 4.2	Jarak Peta.....	60
Gambar 4.3	Jarak Koordinat.....	61
Gambar 4.4	Form Koordinat Peta.....	62
Gambar 4.5	Pencarian Jalur Terpendek.....	62

Abstrak

Pencarian rute terpendek merupakan permasalahan untuk mencari rute minimum dari titik (node) awal ke titik (node) tujuan. Salah satu algoritma yang dapat digunakan untuk menyelesaikan masalah pencarian rute terpendek adalah Algoritma Genetika. Pencarian rute terpendek dilakukan berdasarkan titik awal, titik tujuan dan waktu keberangkatan.

Dalam penyelesaian rute terpendek tersebut sistem memerlukan suatu metode yang dapat digunakan untuk membantu pencarian rute terpendek, metode yang digunakan untuk pencarian rute terpendek tersebut adalah algoritma genetika, karena algoritma genetika merupakan sebuah metode yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi yang memiliki fungsi rekombinasi, mutasi, maupun perbaikan individu, dan yang menggunakan proses-proses yang sebagian besar dilakukan secara acak yang mempertimbangkan kecepatan di setiap jarak jalannya yang berdampak mempengaruhi waktu.

Berdasarkan hasil penelitian, rute terpendek merupakan rute tercepat tetapi jarak terpendek tidak berarti rute terpendek karena memperhitungkan faktor nilai bobot kemacetan jalan.

Kata Kunci: Pencarian Rute, algoritma genetika, Visual Basic

Abstract

The shortest route search is the problem of finding the minimum route from the starting point (node) to the destination point (node). One of the algorithms that can be used to solve the problem of shortest route search is Genetic Algorithm. The shortest route search is done based on starting point, destination point and departure time.

In the solution of the shortest route the system requires a method that can be used to help the shortest route search, the method used to search the shortest route is a genetic algorithm, because the genetic algorithm is a method that utilizes a natural selection process known as the evolutionary process that has the function of recombination, mutations, and individual improvements, and which use processes that are largely randomly assigned to take into consideration the speed at each distance that impact time.

Based on the results of research, the shortest route is the fastest route but the shortest distance does not mean the shortest route because it takes into account the value factor of the weight of road congestion.

Keywords: Route Search, Genetic Algorithm, Visual Basic

DAFTAR ISI

KATA PENGANTAR	ii
REKOMENDASI.....	iii
DAFTAR ISI	iv
DAFTAR TABEL.....	vi
DAFTAR GAMBAR	vii
ABSTRAK	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan dan Penelitian.....	3
1.3.1 Tujuan Penelitian.....	3
1.3.2 Manfaat Penelitian.....	3
1.4 Batasan Masalah.....	4
1.5 Metode Penelitian.....	5
1.6 Sistematika Tulisan.....	7
BAB II TINJAUAN PUSTAKA.....	9
2.1 Artificial Intelligenci	9
2.1.1 Penegrtian AI	9
2.1.2 Sejarah AI	12
2.1.3 Perbedaan antara AI dan Kecerdasan alami.....	13
2.1.4 Ruang Lingkup AI pada aplikasi Komersial.....	15
2.1.5 Komputasi AI dan Komputasi Konvensional.....	17
2.2 Strategi Pencarian Lintasan.....	19
2.2.1 Breadth First Search.....	19
2.2.2 Pencarian Mendalam Pertama (Depth-First Search).....	19
2.2.2 Pencarian Mendalam pertama (Depth-First Search).....	22
2.2.3 Pencarian Dengan Algoritma Genetika.....	24
2.2.4 Struktur Umum Algoritma Genetika.....	27
2.2.5 Aplikasi Algoritma Genetika.....	28

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah

Medan dikenal sebagai kota wisata dan merupakan salah satu kota besar di Indonesia. Objek religi dan lainnya yang terkenal dengan keindahan alamnya, dan perkembangan pusat pembelanjaan khususnya dalam bidang market seperti Mall yang pesat beberapa tahun terakhir ini membuat Medan menjadi tujuan wisata yang layak untuk dikunjungi.

Berdasarkan hasil pengamatan di lapangan dan jalan-jalan di kota besar sangat membingungkan karena Medan adalah salah satu kota besar di Indonesia, walau pemetaan kota Medan yang cukup lumayan baik, tetapi pendatang masih dihadapkan kebingungan dengan jalan- jalan kota Medan, dengan itu pendatang menjadi kesulitan untuk mencari jalan yang dituju, dan pendatang pun kebingungan untuk mencari rute menuju tujuan serta memilih jenis transportasi.

Karena permasalahan di atas maka di perlukan sebuah sistem yang membantu informasikan rute jalan yang harus dilalui menuju jalan tujuan dengan waktu yang efisien untuk mendukung kemajuan pariwisata kota Medan, dengan dukungan informasi dari Dinas Perhubungan Provinsi Sumatera Utara. Agar pemilihan rute jalan bisa sesuai dengan kriteria yang diharapkan, perlu dilakukan proses penyeleksian rute jalan terlebih dahulu. Namun dikarenakan

penyeleksian rute jalan yang masih manual tidak mungkin dilakukan, maka dibutuhkan waktu lebih dan kecermatan dalam proses untuk menentukan rute terpendek.

Dalam penyelesaian rute terpendek tersebut sistem memerlukan suatu metode yang dapat digunakan untuk membantu pencarian rute terpendek, metode yang digunakan untuk pencarian rute terpendek tersebut adalah algoritma genetika, karena algoritma genetika merupakan sebuah metode yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi yang memiliki fungsi rekombinasi, mutasi, maupun perbaikan individu, dan yang menggunakan proses-proses yang sebagian besar dilakukan secara acak yang mempertimbangkan kecepatan di setiap jarak jalannya yang berdampak mempengaruhi waktu tempuh di setiap jalan, algoritma genetika bisa menghasilkan solusi yang “bagus” dengan kecepatan dapat diterima.

Maka dengan menggunakan metode algoritma genetika akan membantu pencarian waktu yang efisien yang menggunakan proses-proses yang sebagian besar dilakukan secara acak, dan menghasilkan solusi yang bagus dengan kecepatan dapat diterima. Berdasarkan permasalahan diatas maka diusulkan aplikasi untuk menunjang untuk pencariin rute kota Medan. Atas dasar inilah gagasan membuat suatu sistem pendukung keputusan yang penulis tuangkan dalam judul **“Aplikasi Pendukung pencarian Rute Terpendek Kota Medan Dengan Algoritma Genetika”**.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, didapatkan masalah pokok yang perlu diselesaikan yaitu :

1. Bagaimana menentukan rute terpendek pada kota Medan
2. Bagaimana menerapkan metode Algoritma Genetika untuk mencari rute terpendek kota Medan.
3. Bagaimana Membangun Aplikasi Pendukung Pencarian Rute Terpendek Kota Medan Dengan Algoritma Genetika.

1.3. Tujuan dan Manfaat Penelitian

1.3.1. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Untuk mengetahui rute terpendek kota Medan agar masyarakat/pendatang tidak Kewalahan dalam mencari objek yang akan mereka kunjungi.
2. Untuk menerapkan algoritma genetika untuk menentukan rute terpendek kota Medan.
3. Untuk merancang suatu aplikasi untuk menentukan rute terpendek kota Medan.

1.3.2. Manfaat Penelitian

Manfaat dari penelitian membangun aplikasi pendukung pencarian rute terpendek kota Medan dengan algoritma genetika yaitu:

1. Membantu pendatang, menuju jalan yang dituju di kota Medan.
2. Membantu pencarian rute menuju jalan tujuan di kota Medan.
3. Membantu pencarian rute terpendek menuju jalan tujuan di kota Medan.
4. Membantu meningkatkan efektifitas dan efisiensi menuju jalan tujuan di kota Medan.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini mencakup:

1. Pada pencarian jalur jalan, diasumsikan setiap jalur jalan hanya dapat dilalui sebanyak satu kali.
2. Diasumsikan pengguna menggunakan kendaraan beroda empat atau beroda dua.
3. Jalan yang digunakan dapat berupa jalan satu arah atau jalan dua arah.
4. Rute yang tersedia hanya untuk kota Medan dengan sarana dan prasarana.
5. Dalam penelitian ini, rute yang dibahas yaitu Medan Tuntungan, Medan Johor, Medan Selayang, Medan Baru, Medan Area, Medan Kota dan Istana Maimun.
6. Dalam penelitian ini, resiko penghambat seperti lampu merah di persimpangan jalan, pedagang kaki lima, gang kecil dan faktor penghambat lainnya tidak diperhitungkan.

7. Dalam penelitian ini peranan algoritma genetika lebih dominan.
8. Kecepatan yang digunakan untuk menempuh satu jalan adalah konstan.
9. Pemodelan data yang digunakan adalah pemodelan analisis berbasis objek oriented dimana *tolls* yang digunakan adalah *usecase diagram*, *activity diagram*, *class diagram* dan *sequence diagram*.
10. Aplikasi ini dibangun dengan menggunakan bahasa pemograman *Visual Basic* dengan databasenya *SQL Server*.

1.5. Metode Penelitian

Dalam penulisan penelitian ini, dilakukan beberapa hal untuk mendapatkan data yang diperlukan, antara lain:

1. Metode Pengumpulan Data

Beberapa metode pengumpulan data yang dilakukan yaitu:

a. Metode Observasi

Metode ini dilakukan oleh penulis dengan meneliti obyek serupa yang sudah jadi dan masih aktif di internet.

b. Metode Kepustakaan

Metode keputakaan dilakukan oleh penulis dengan mencari sumber yang terkait dengan Sistem yang akan dibangun.

2. Metode Perancangan Sistem

Metode Perancangan yang digunakan dalam pembuatan penelitian ini adalah dengan metode Waterfall yang tahap-tahapnya dilakukan seperti terlihat dibawah ini :

a. Analisis Kebutuhan

Analisis kebutuhan adalah yaitu analisa tentang metode genetika

b. Analisa dan Perancangan Sistem

Perancangan sistem merupakan tahapan yang dilakukan untuk membuat sebuah rancangan program berdasarkan input dan output yang diinginkan.

c. Implementasi Sistem

Setelah pembuatan perancangan sistem maka langkah selanjutnya adalah mengimplementasi hasil perancangan ke dalam program

d. Evaluasi Sistem

Evaluasi merupakan langkah setelah program dimplementasikan untuk mengetahui kesalahan atau trouble yang mungkin terjadi, sampai dipastikan sistem dapat berjalan dengan sempurna.

e. Penulisan laporan penelitian

Ini adalah tahap akhir dari penelitian .

1.6. Sistematika Penulisan

Sistematika penulisan penelitian ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini akan menjelaskan tentang Latar Belakang Masalah, Rumusan Masalah, Manfaat Penelitian, Tujuan Penelitian, Batasan Masalah, Metode Penelitian, Sistematika Penulisan.

BAB II : LANDASAN TEORI

Dalam bab ini diuraikan tentang teori-teori yang digunakan sebagai dasar untuk melakukan pemecahan masalah yang telah dirumuskan.

BAB III : ANALISA DAN PERANCANGAN

Bab ini menguraikan seluruh uraian mengenai informasi dan data yang telah dikumpulkan dan dikaitkan dengan cara berfikir guna mendapatkan pemecahan masalah.

BAB IV : ALGORITMA DAN IMPEMENTASI

Bab ini berisi tentang pembahasan program aplikasi menurut hak akses dan fasilitas yang digunakan masing-masing pengguna, kelebihan dan kekurangan program aplikasi dan perbandingan dengan program yang ada dipasaran.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi tentang berisi kesimpulan dari pembahasan dan saran-saran yang diharapkan dapat memberikan masukan.

Lampiran

BAB II

TINJAUAN PUSTAKA

2.1. Artificial Intelligence (AI)

2.1.1. Pengertian AI

Artificial Intelligence (AI) disebut juga dengan kecerdasan buatan. AI merupakan salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia.

Tujuan dari AI adalah untuk memecahkan persoalan dunia nyata (bersifat praktis) dan memahami inteligensia (bersifat memahami). AI merupakan salah satu bagian ilmu komputer yang mempelajari tentang bagaimana cara membuat agar komputer dapat melakukan pekerjaan seperti yang dilakukan oleh manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. Manusia bisa menjadi pandai dalam menyelesaikan segala permasalahan di dunia ini karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki oleh seseorang tentu saja diharapkan akan lebih mampu dalam menyelesaikan

permasalahan. Namun bekal pengetahuan saja tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil kesimpulan berdasarkan pengetahuan dan pengalaman yang mereka miliki. Tanpa memiliki kemampuan menalar yang baik, manusia dengan segudang pengalaman dan pengetahuan tidak akan dapat menyelesaikan masalah dengan baik. Demikian pula, dengan kemampuan menalar yang sangat baik, namun tanpa bekal pengetahuan dan pengalaman yang memadai, manusia juga tidak akan bisa menyelesaikan masalah dengan baik.

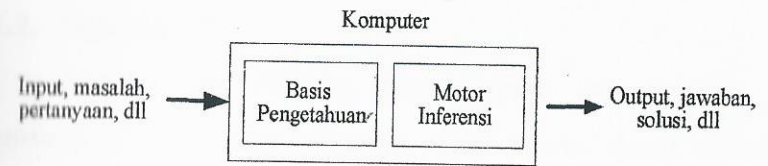
Komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar, agar komputer bisa bertindak seperti dan sebaik manusia. Untuk itu pada *artificial intelligence*, akan mencoba untuk memberikan beberapa metode untuk membekali komputer dengan kedua komponen tersebut agar komputer bisa menjadi mesin yang pintar. Untuk menciptakan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu:

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.

2. Motor Inferensi (*Inference Engine*)

Motor Inferensi merupakan kemampuan menarik kesimpulan berdasarkan pengalaman. Atau dapat juga disebut dengan penalaran.



Gambar 2.1 Sistem yang menggunakan AI

Definisi AI dapat dipandang dari berbagai sudut pandang, yaitu:

1. Sisi kecerdasan.

AI akan membuat mesin menjadi 'cerdas' (mampu berbuat seperti apa yang dilakukan oleh manusia).

2. Sisi penelitian.

AI adalah suatu studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dikerjakan oleh manusia.

3. Sisi bisnis.

AI adalah kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4. Sisi pemrograman.

AI meliputi studi tentang pemrograman simbolik, penyelesaian masalah (*problem solving*) dan pencarian (*searching*).

2.1.2. Sejarah AI

Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya tes untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil tes tersebut kemudian dikenal dengan Turing Test, dimana si mesin tersebut menyamar seolah-oleh sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat dikatakan bahwa mesin tersebut cerdas, seperti layaknya manusia.

AI termasuk bidang ilmu yang relatif muda. AI itu sendiri dimunculkan oleh seorang profesor dari Massachusetts Institute of Technology yang bernama John McCarthy pada tahun 1956 pada Darmouth Conference yang dihadiri oleh para peneliti AI. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan mesin tersebut.¹

¹ KusumaDewi, S, 2003, *Artificial Intelligence*, penerbit graham ilmu, Yogyakarta

Beberapa program AI yang mulai dibuat pada tahun 1956-1966, antara lain:

1. *Logic Theorist*, diperkenalkan pada Dartmouth Conference, program ini dapat membuktikan teorema-teorema matematika.
2. *Sad Sam*, diprogram oleh *Robert K. Lindsay* (1960). Program ini dapat mengetahui kalimat-kalimat sederhana yang ditulis dalam bahasa Inggris dan mampu memberikan jawaban dari fakta-fakta yang didengar dalam sebuah percakapan.
3. *ELIZA*, diprogram oleh *Joseph Weizenbaum* (1967). Program ini mampu melakukan terapi terhadap pasien dengan memberikan beberapa pertanyaan dan jawaban.

2.1.3. Perbedaan antara AI dan Kecerdasan Alami

Dibandingkan dengan kecerdasan kecerdasan yang dimiliki oleh manusia (kecerdasan alami), AI memiliki beberapa keuntungan secara komersial antara lain:

1. AI lebih bersifat permanen.

Kecerdasan alami akan cepat mengalami perubahan. Hal ini dimungkinkan karena sifat manusia yang pelupa. AI tidak akan berubah sepanjang sistem komputer dan program tidak diubah.

2. AI lebih mudah diduplikasi dan disebar.

Mentransfer pengetahuan manusia dari satu orang ke orang lain membutuhkan proses yang sangat lama, dan keahlian itu juga tidak akan pernah dapat diduplikasi dengan lengkap. Oleh karena itu, jika pengetahuan terletak pada suatu sistem komputer, pengetahuan tersebut dapat disalin dari komputer

tersebut dan dapat dipindahkan dengan mudah ke komputer yang lain.

3. AI lebih murah dibanding dengan kecerdasan alami.

Menyediakan layanan komputer akan lebih mudah dan lebih murah dibandingkan dengan harus mendatangkan seseorang untuk mengerjakan sejumlah pekerjaan dalam jangka waktu yang sangat lama.

4. AI bersifat konsisten.

Hal ini disebabkan karena AI adalah bagian dari teknologi komputer. Sedangkan kecerdasan alami akan senantiasa berubah-ubah.

5. AI dapat didokumentasi.

Keputusan yang dibuat oleh komputer dapat didokumentasi dengan mudah dengan cara melacak setiap aktivitas dari sistem tersebut. Kecerdasan alami sangat sulit untuk direproduksi.

6. AI dapat mengerjakan pekerjaan jauh lebih cepat dibanding dengan kecerdasan alami.

7. AI dapat mengerjakan pekerjaan lebih baik dibanding dengan kecerdasan alami. Sedangkan keuntungan dari kecerdasan alami adalah:

1. Kreatif.

Kemampuan untuk menambah ataupun memenuhi pengetahuan itu sangat melekat pada jiwa manusia. Pada kecerdasan buatan, untuk menambah pengetahuan harus dilakukan melalui sistem yang dibangun.

2. Kecerdasan alami memungkinkan orang untuk menggunakan pengalaman secara langsung. Sedangkan pada kecerdasan buatan harus berkerja dengan input-input simbolik.
3. Pemikiran manusia dapat digunakan secara luas, sedangkan AI sangat terbatas.

2.1.4. Ruang Lingkup AI pada Aplikasi Komersial

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran AI. Karakteristik 'cerdas' sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. AI tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan AI melahirkan sebuah area yang dikenal dengan nama *cognition & psycholinguistics*. Irisan antara teknik elektro dengan AI melahirkan berbagai ilmu, seperti: pengolahan citra, teori kendali, pengenalan pola dan robotika.

Dewasa ini, AI juga memberikan kontribusi yang cukup besar di bidang manajemen. Adanya sistem pendukung keputusan dan Sistem Informasi Manajemen juga tidak terlepas dari andil AI.

Adanya irisan penggunaan AI di berbagai disiplin ilmu tersebut menyebabkan cukup rumitnya untuk mengklasifikasikan AI menurut disiplin ilmu yang menggunakannya. Untuk memudahkan hal tersebut, maka pengklasifikasian lingkup AI

didasarkan pada *output* yang diberikan, yaitu pada aplikasi komersial (meskipun sebenarnya AI itu sendiri bukan merupakan medan komersial).Lingkup utama dalam kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*). Disini, komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian, komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami (*Natural Language Processing*). Dengan pengolahan bahasa alami ini diharapkan *user* dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (*Speech Recognition*). Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.
4. Robotika & Sistem Sensor (*Robotics & Sensory Systems*).
5. *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game Playing*.

Beberapa karakteristik yang ada pada sistem yang menggunakan AI adalah pemogramannya yang cenderung bersifat simbolik ketimbang algoritmik, bisa mengakomodasi *input* yang

tidak lengkap, bisa melakukan inferensi dan adanya pemisahan antara kontrol dengan pengetahuan.

Namun, seiring dengan perkembangan teknologi, muncul beberapa teknologi yang juga bertujuan untuk membuat agar komputer menjadi cerdas sehingga dapat menirukan kerja manusia sehari-hari.

Teknologi ini juga mampu mengakomodasi adanya ketidakpastian dan ketidaktepatan data *input*. Dengan didasari pada teori himpunan, maka pada tahun 1965 muncul Logika Fuzzy. Kemudian pada tahun 1975, *John Holland* mengatakan bahwa setiap *problem* berbentuk adaptasi (alami maupun buatan) secara umum dapat diformulasikan dalam terminologi genetika. Algoritma genetika ini merupakan simulasi proses evolusi Darwin dan operasi genetika atas kromosom.²

2.1.5. Komputasi AI dan Komputasi Konvensional

Pada awal diciptakannya, komputer hanya diperuntukkan sebagai alat hitung atau komputasi konvensional. Untuk itu, ada beberapa perbedaan antara komputasi yang dilakukan pada AI dengan komputasi konvensional tersebut. Perbedaan komputasi AI dengan komputasi konvensional dapat dilihat pada tabel 2.1.

² KusumaDewi, S, 2003, *Artificial Intelligence*, penerbit graham ilmu, Yogyakarta

Tabel 2.1 AI Vs Pemograman Konvensional

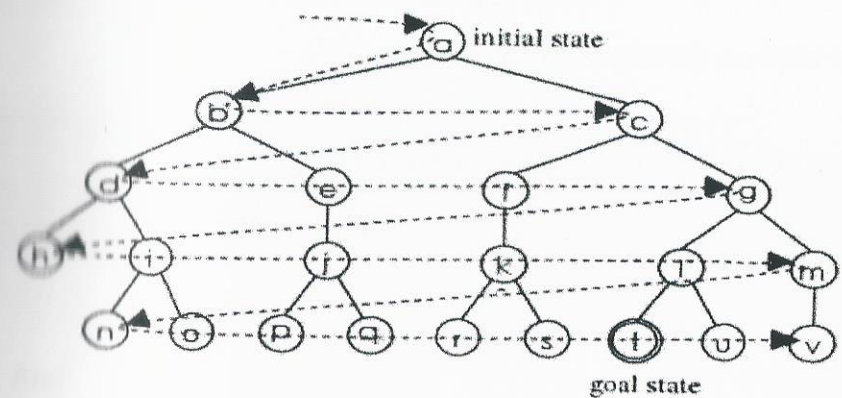
Dimensi	Komputasi AI	Komputasi Konvensional
Pemrosesan	Mengandung konsep-konsep simbolik	Algoritmik
Sifat Input	Bisa tidak lengkap	Harus lengkap
Pencarian	Kebanyakan bersifat Genetika	Biasanya didasarkan pada algoritma
Keterangan	Disediakan	Biasanya tidak disediakan
Fokus	Pengetahuan	Data dan informasi
Struktur	Kontrol dipisahkan dari pengetahuan	Kontrol terintegrasi dengan informasi (data)
Sifat output	Kuantitatif	Kualitatif
Pemeliharaan & update	Relatif mudah	Sulit
Kemampuan menalar	Ya	Tidak

2.2. Strategi Pencarian Lintasan

Pencarian boleh jadi merupakan hasil dari suatu solusi ruang keadaan yang mungkin telah terkunjungi semua, tetapi tanpa penyelesaian. Pencarian yang mendalam (*Exhausting search Strategy*) mungkin dilakukan dengan menggunakan strategi *Breadth First search* atau *Depth First search (Iterative Deepening)*. Kedua pencarian ini merupakan pencarian buta (*blind search*).

2.2.1. Breadth First Search

Prosedur *Breadth First search* merupakan pencarian yang dilakukan dengan mengunjungi tiap-tiap *node* secara sistematis pada setiap level hingga keadaan tujuan (*goal state*) ditemukan. Atau dengan kata lain, penelusuran yang dilakukan adalah dengan mengunjungi tiap-tiap *node* pada level yang sama hingga ditemukan *goal state*-nya. Untuk lebih jelasnya, perhatikan ilustrasi dari *Breadth First search* pada gambar dibawah ini.



Gambar 2.2. Teknik pencarian Breadth First search

Pengimplementasian *Breadth First search* dapat ditelusuri Dengan menggunakan daftar (*list*), *open*, dan *closed*; untuk menelusuri gerakan pencarian di dalam ruang keadaan. Prosedur untuk *Breadth First search* dapat dituliskan sebagai berikut :

Procedure Breadth First search

Begin

Open := [start]

Closed := []

While open = [] do

Begin

Remove leftmost state from open, call it x

If x is goal then return SUCCESS

Else

Begin

Generate children of x;

Put x on closed;

discard children of x is already on open or closed;

put remaining children on right end of open;

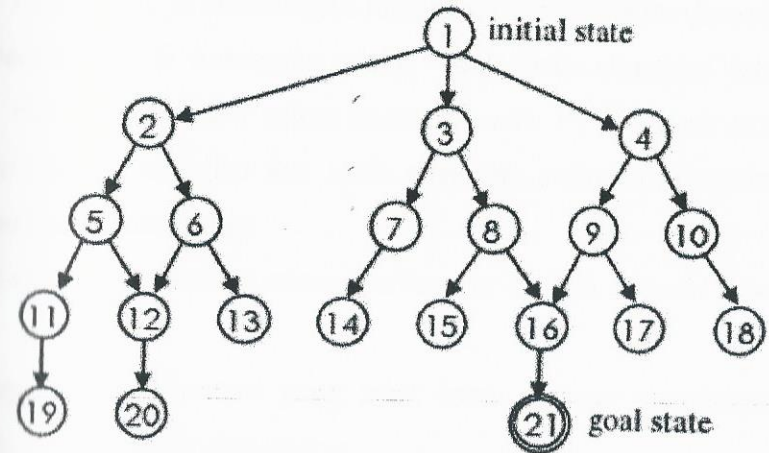
End;

End

return FAIL

End

Untuk lebih jelasnya, simak contoh graf pada gambar berikut ini :



Gambar 2.3. Contoh graf yang akan ditelusuri

Pada gambar 2.3, *state* 21 merupakan tujuannya (*goal*) sehingga bila ditelusuri menggunakan prosedur *Breadth First search*, diperoleh :

1. Open = [1]; closed = [].
2. Open = [2, 3, 4]; closed = [1].
3. Open = [3, 4, 5, 6]; closed = [2, 1].
4. Open = [4, 5, 6, 7, 8]; closed = [3, 2, 1].
5. Open = [5, 6, 7, 8, 9, 10]; closed = [4, 3, 2, 1].
6. Open = [6, 7, 8, 9, 10, 11, 12]; closed = [5, 4, 3, 2, 1].
7. Open = [7, 8, 9, 10, 11, 12, 13] (karena 12 telah di-open);
8. closed = [6, 5, 4, 3, 2, 1].
9. Open = [8, 9, 10, 11, 12, 13, 14]; closed = [7, 6, 5, 4, 3, 2, 1].
10. Dan seterusnya sampai *state* 21 diperoleh atau open = []

Ada beberapa keuntungan menggunakan algoritma *Breadth First search* ini, di antaranya adalah tidak akan menemui jalan buntu dan jika ada satu solusi maka *Breadth First search* akan menemukannya, dan jika ada lebih dari satu solusi maka solusi minimum akan ditemukan.

Namun ada tiga persoalan utama berkenaan dengan *Breadth First search* ini yaitu:

Membutuhkan memori yang lebih besar, karena menyimpan semua *node* dalam satu pohon.

Membutuhkan sejumlah besar pekerjaan, khususnya jika lintasan solusi terpendek cukup panjang, karena jumlah *node* yang perlu diperiksa bertambah secara eksponensial terhadap panjang lintasan.

Tidak relevannya operator akan menambah jumlah *node* yang harus diperiksa.

Oleh karena proses *Breadth First search* mengamati *node* di tiap level graf sebelum bergerak menuju ruang yang lebih dalam maka mula-mula semua keadaan akan dicapai lewat lintasan yang terpendek dari keadaan awal. Oleh sebab itu, proses ini menjamin ditemukannya lintasan terpendek dari keadaan awal ke keadaan tujuan (akhir). Lebih jauh karena mula-mula semua keadaan ditemukan melalui lintasan terpendek sehingga setiap keadaan yang ditemui pada kali kedua didapati pada sepanjang sebuah lintasan yang sama atau lebih panjang. Kemudian, jika tidak ada

kesempatan ditemukannya keadaan yang identik pada sepanjang lintasan yang lebih baik maka algoritma akan menghapusnya.

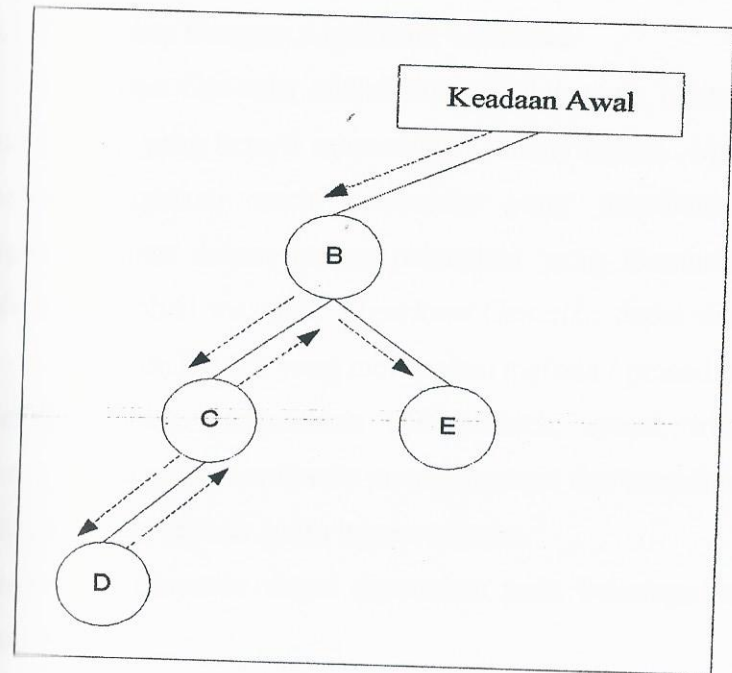
4.3.2. Pencarian Mendalam Pertama (Depth-First Search)

Pencarian mendalam pertama merupakan teknik penelusuran data pada *node-node* secara vertikal dan sudah mendefinisikan secara mendalam. Proses pemeriksaan akan bergerak turun jika *node* yang diperiksa saat ini tidak sesuai dengan tujuan. *Node* tidak akan dikembangkan ke samping walaupun *node* masih mempunyai beberapa *node* anak yang dapat dikembangkan. *Node* anak baru dikembangkan lagi setelah selesainya pencarian mendalam terhadap *node* anak yang sebelumnya telah dikembangkan. Proses ini diulangi terus hingga didapatkan solusi. Berikut prosedur pencarian mendalam pertama.

Keuntungan dari metode ini adalah membutuhkan memori yang relatif kecil, karena hanya *node-node* pada lintasan yang aktif saja yang disimpan. Secara kebetulan, metode *depth-first search* akan menemukan solusi tanpa harus menguji lebih banyak lagi semua ruang keadaan.

Sedangkan kelemahan dari metode ini adalah memungkinkan tidak ditemukannya tujuan yang diharapkan dan hanya akan mendapatkan 1 solusi pada setiap pencarian.²

² Nisiani, A., dan Arhami, M., 2006, *Konsep Kecerdasan Buatan*, Yogyakarta



Gambar 2.4 Prosedur Pencarian Mendalam Pertama

Pada gambar 2.4, pencarian dimulai dari keadaan awal kemudian dilanjutkan dengan mengembangkan *node B*, *node C* dan *node D*. Ketika diketahui *node D* bukan merupakan solusi dan tidak dapat mengembangkan *node* anak lagi, pencarian dikembalikan ke *node parent*, dan mengecek apakah ada *node* anak yang dapat dikembangkan lagi. Apabila ada, maka pencarian dilakukan ke level bawah lagi. Pencarian dilakukan terus hingga menemukan solusi.

2.2.3. Pencarian Dengan Algoritma Genetika

Algoritma Genetika adalah suatu istilah yang berasal dari bahasa Yunani yang berarti menemukan / menyingkap. *Algoritma Genetika* merupakan suatu perbuatan yang membantu kita menemukan jalan dalam pohon pelacakan yang menuntut kita kepada suatu solusi masalah. *Algoritma Genetika* dapat diartikan juga sebagai suatu kaidah yang merupakan metoda / prosedur yang didasarkan kepada pengalaman dan praktek, syarat, trik atau bantuan lainnya yang membantu mempersempit dan memfokuskan proses pelacakan kepada suatu tujuan tertentu.³

Algoritma Genetika dapat digunakan pada beberapa kondisi berikut ini :

1. Mengatasi *combinatorial explosion*.

Ada masalah yang kemungkinan arah penyelesaiannya berkembang pesat (bersifat faktorial) sehingga menimbulkan *combinatorial explosion*. *Algoritma Genetika* merupakan cara untuk menentukan kemungkinan arah penyelesaian masalah secara efisien.

2. Pada umumnya hasilnya cukup baik.

Sekali pun tidak optimal biasanya mendekati optimal.

3. Membantu pemahaman bagi orang yang menyelesaikan persoalan.

Banyak alternatif *Algoritma Genetika* yang dapat diterapkan dalam suatu percobaan. Orang yang menyelesaikan persoalan

³ http://id.wikipedia.org/wiki/Algoritma_genetik

tersebut akan lebih mengerti persoalannya jika mencoba *Algoritma Genetika* yang diterapkannya.

Function GenetikAlgorithm(**population**, *Fitness-FN*)-> an individual

input berupa population, sebuah kumpulan individual dan *Fitness-FN*, sebuah fungsi yang mengukur *fitness* suatu individual

deklarasi

i, x, y : integer

Algoritma

repeat

new_population <- empty set

for *i* = 1 to size(population) **do**

x <- RandomSelection(population, *Fitness-FN*)

y <- RandomSelection(population, *Fitness-FN*)

child <- Reproduce(*x, y*)

if (smallRandomProbability) **then**

child <- mutate(child)

add child to new_population

population <- new_population

dikenal dengan istilah anak (*offspring*) terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*). Selain operator penyilangan, suatu kromosom juga dapat dimodifikasi dengan menggunakan operator mutasi.

Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai fitness dari kromosom induk (*parent*) dan nilai fitness dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

Aplikasi Algoritma Genetika

Sejak dirintis oleh *John holland*, Algoritma Genetika telah dipejari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma Genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter optimal. Hal ini membuat banyak orang mengira bahwa Algoritma Genetika hanya bisa digunakan untuk masalah optimasi. Pada kenyataannya, Algoritma Genetika juga memiliki performansi yang bagus untuk masalah-masalah selain optimasi.

Keuntungan penggunaan Algoritma Genetika terlihat dari kemudahan implementasi dan kemampuan untuk menemukan solusi yang “bagus” (bisa diterima) secara cepat untuk masalah-

masalah berdimensi tinggi. Algoritma Genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut :

1. Ruang masalah sangat besar, kompleks, dan sulit dipahami.
2. Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit.
3. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi.
4. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup "bagus" atau bisa diterima.
5. Terdapat batasan waktu, misalnya dalam *real time sistem* atau sistem waktu nyata.

Algoritma Genetika telah banyak di aplikasikan untuk menyelesaikan masalah dan pemodelan dalam bidang teknologi, bisnis, dan entertainment, seperti :

1. Optimasi

Algoritma Genetika digunakan untuk optimasi numerik dan optimasi kombinatorial seperti Travelling Salesman Problem (TSP), perancangan IC, Job Shop Sceduling, optimasi suara dan video.

2. Pemograman Otomatis

Algoritma genetika digunakan untuk melakukan proses evolusi terhadap program komputer untuk merancang struktur komputasional, seperti cellular automata dan sorting networks

3. Machine Learning

Algoritma genetika berhasil diaplikasikan untuk memprediksi struktur protein. Algoritma genetika juga berhasil dalam perancangan neural networks (jaringan syaraf tiruan).³

2.3. Microsoft Visual Basic

2.3.1 Menu Bar

Menu bar dalam VB seperti yang biasa kita lihat dalam Microsoft Office. Di dalamnya terdapat menu File, Edit, View, Project, Format dan sejenisnya



Gambar 2.5. Menu Bar

2.3.2 Standard Bar

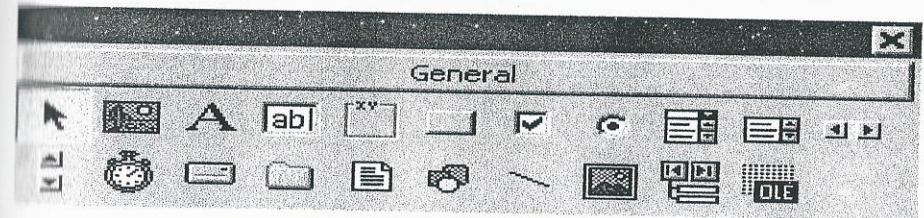
Standard Bar adalah sebuah window yang berisi ikon-ikon yang fungsinya sama dengan menu, tetapi dapat digunakan dengan lebih cepat karena sebuah ikon mewakili satu perintah tertentu. Contohnya adalah ikon Open, Save, Copy, Paste, Undo dan sejenisnya.



Gambar 2.6. Standard Bar

2.3.3 ToolBox

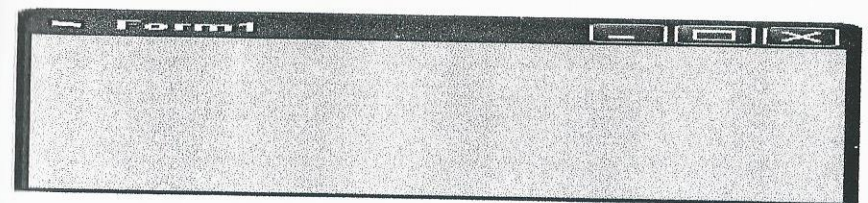
ToolBox adalah window yang berisi objek-objek untuk ditempatkan dalam form. Contohnya objek -Label, TextBox, ListBox, ComboBox, Frame, dan sejenisnya.



Gambar 2.7. Toolbox

2.3.4 Form

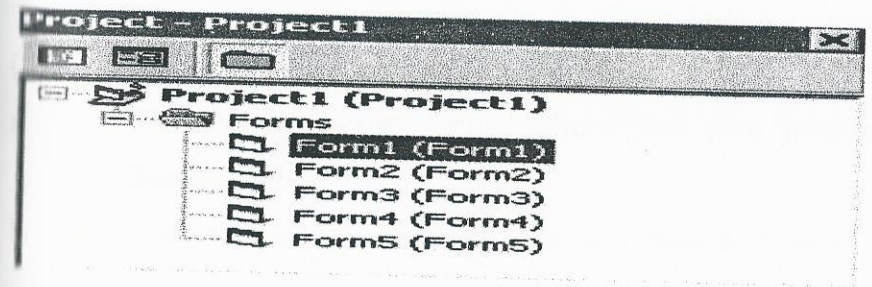
Form sering disebut dengan GUI (Graphical User Interface), adalah sebuah objek yang digunakan untuk menempatkan objek-objek dari Toolbox.



Gambar 2.8. Form

2.3.5 Project Window

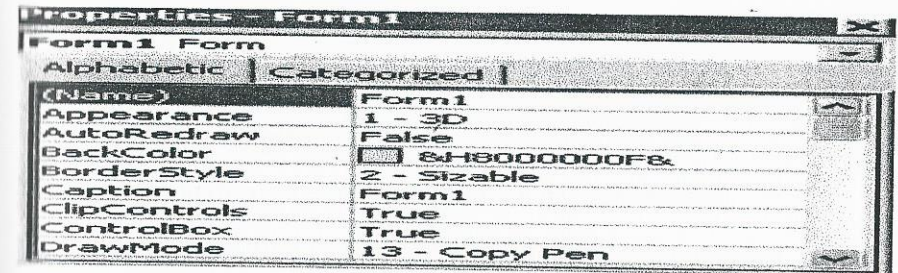
Project Window adalah sebuah window yang menampung project dan Form. Didalamnya juga terdapat ikon View Code (untuk menampilkan area text editor) dan ikon View Object (untuk menampilkan GUI). Dengan menggunakan project window ini kita juga dapat menambah form, menghapus, mengganti nama form, memilih startup project dan lain-lain.



Gambar 2.9. Project windows

2.3.6 Properti Window

Properti window adalah sebuah window yang digunakan untuk memodifikasi objek berupa mengubah Caption, memberi nama objek, mengubah warna, ukuran, model dan sejenisnya.



Gambar 2.10. Propertis

2.4. Database

Menurut Jogianto, "Analisa dan Desain", Andi Yogyakarta (1990), tujuan utama perancangan sistem database yaitu⁵:

- Memiliki kemampuan menyimpan seluruh data yang berguna dalam database.
- Pengamanan terhadap isi dan pengguna data dapat diterapkan

2.4.1. Model Database

Pada dasarnya kekuatan dan kelemahan dari model mencerminkan perbedaan filosofi manajemen database. Beberapa kriteria dalam mengevaluasi model data:

- Kesederhanaan (*Simplicity*)

Sebuah kesederhanaan sedikit mungkin jumlah tiga struktur, knidah dan atribut.

- Pemodelan yang mendekati kenyataan (*Modelling Direction*)

Sebuah model sebaiknya mendekati konsep yang nyata di dunia.

⁵ Jogianto, "Analisa dan Desain", Andi Yogyakarta (1990),

c. Keluwesan

Model ini seharusnya mudah untuk memberikan kemampuan pemodelan langsung

d. Kemampuan menghasilkan gambar (*picturability*)

model seharusnya mudah menampilkan format gambar.

e. Keistimewaan model (*modeling uniqueness*)

Sebuah model seharusnya bebas dari pertimbangan implementasi khusus.

f. Terminologi tidak bertentangan

Sebuah model seharusnya menggambarkan terminologi yang latin.

g. Dekat dengan dasar informasi

Gagasan yang digunakan dalam sebuah model seharusnya dekat dengan basis perangkat keras dimana model ini akan di implementasikan. Model yang di kenal ada tiga yaitu:

a. Model Relasional

Dikembangkan berdasarkan konsep matematika dari hubungan dan kumpulan entity atau beberapa objek dari kumpulan atribut. Model data relational menggunakan table dua dimensi, kolom di dalam table cocok untuk setiap atribut untuk setiap entity atau relationship tipe dan baris (*tuple*) untuk setiap transaksi.

b. Model Hirarki

Sering dikenal dengan *Binery Tree* (Pohon biner atau *Binary Relationship*), merupakan jenis struktur tree yang terbaik dimana data akar (*root*) mempunyai beberapa cabang.

c. Jaringan atau *Network*

Suatu model dimana objek dapat dihubungkan dengan beberapa objek lain yang lebih tinggi atau yang lebih rendah.

2.4.2. Langkah-Langkah Perancangan Database

Dalam perancangan database perlu diperhatikan langkah-langkah sebagai berikut :

1. Menentukan kebutuhan input dari sistem baru

Input yang dirancang dapat ditentukan dari *Data Flow Diagram* (DFD) sistem yang dibuat.

2. Menentukan parameter dari input

Setelah input-input yang akan dirancang telah dapat ditentukan, maka parameter dari input selanjutnya juga ditentukan.

Parameter ini juga meliputi :

a. Bentuk dari input dan dokumen dasar

b. Sumber *input*

c. Volume *input*

d. Periode *input*

Tiga sasaran yang akan digunakan oleh seseorang perancang model fisik dari database:

- a. Pengetahuan tentang fungsi-fungsi database manajemen sistem (DBMS)
- b. Memberikan tentang karakteristik pengaksesan secara langsung
- c. Pengetahuan tentang aplikasi

Model fisik adalah suatu rencana kerja dari database yang disimpan secara fisik

ditempat yang telah ditetapkan.

3.1.2. Algoritma Penyelesaian

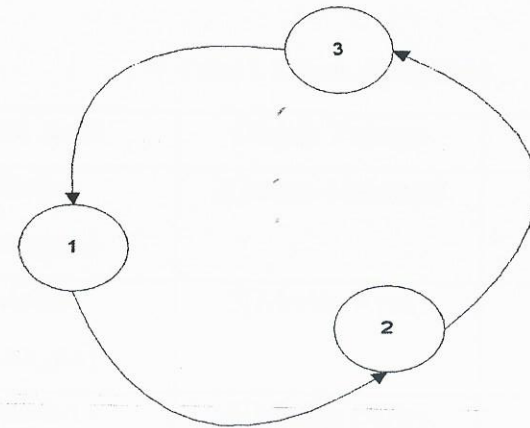
Penyelesaian menggunakan algoritma genetika menghasilkan solusi dengan rute perjalanan yang memiliki jarak terpendek. Penyelesaian perhitungan menggunakan algoritma genetika muncul dua kasus, yaitu kasus khusus solusi tidak tunggal dan kasus khusus satu arah.

Algoritma genetika berhasil dibuat menjadi perangkat lunak, sehingga proses perhitungan dan penentuan rute terdekat akan jauh lebih cepat dibandingkan dengan perhitungan secara manual. Hasil yang didapatkan antara perhitungan dengan program algoritma genetika sama dengan perhitungan secara manual.

Perbedaan hanya dalam perhitungan kasus khusus solusi tidak tunggal. Hal ini dikarenakan dalam perhitungan menggunakan program algoritma genetika, hanya memilih salah satu dari biaya penyisipan objek paling minimal yang ada, sedangkan jika dihitung secara manual, maka semua kemungkinan yang ada dicoba untuk mendapatkan solusi yang optimal.

Berikut ini adalah tata urutan algoritma genetika :

- a. Penelusuran dimulai dari sebuah objek pertama yang dihubungkan dengan sebuah objek terakhir.
- b. Dibuat sebuah hubungan subtour antara 2 objek tersebut. Yang dimaksud subtour adalah perjalanan dari objek pertama dan berakhir di objek pertama, misalnya $(1,3) \rightarrow (3,2) \rightarrow (2,1)$ seperti tergambar dalam gambar 3.2.



Gambar 3.2. Subtour

- c. Ganti salah satu arah hubungan (arc) dari dua objek dengan kombinasi dua arc, yaitu arc (i,j) dengan arc (i,k) dan arc (k,j), dengan k diambil dari objek yang belum masuk subtour dan dengan tambahan jarak terkecil. Jarak diperoleh dari:

$$C_{ik} + C_{kj} - C_{ij}$$

Dimana

C_{ik} = jarak dari objek i ke objek k,

C_{kj} = jarak dari objek k ke objek j dan

C_{ij} = adalah jarak dari objek i ke objek j

- d. Ulangi langkah 3 sampai seluruh objek masuk dalam subtour Sebagai contoh diberikan 5 objek dengan jarak antar objek dan dengan menggunakan Sepeda Motor atau Mobil untuk menempuh Objek tersebut, seperti tertera dalam tabel 3.1 dibawah ini.

Tabel 3.1. Jarak Antar Objek

Objek Asal	Objek Tujuan	Jarak
1(Medan Tuntungan)	2(Istana Maimun)	40
1(Medan Tuntungan)	3(Medan Area)	50
1(Medan Tuntungan)	4(Medan Baru)	30
1(Medan Tuntungan)	5(Medan Johor)	19
2(Istana Maimun)	3(Medan Area)	17
2(Istana Maimun)	4(Medan Kota)	15
2(Istana Maimun)	5(Medan Johor)	29
3(Medan Area)	4(Medan Baru)	37
3(Medan Area)	5(Medan Johor)	39
4(Medan Baru)	5(Medan Johor)	34

Untuk mencari jarak terpendek melalui ke 5 objek tersebut sebagaimana terdapat dalam tabel 3.1, ambil langkah-langkah sebagai berikut:

- a. Ambil perjalanan dari objek 1 ke 5
- b. Buat jalur dari $\rightarrow (1,5) \rightarrow (5,1)$

- c. Buat tabel yang menyimpan objek yang bisa disisipkan dalam jalur beserta tambahan jaraknya, seperti ditampilkan dalam tabel 3.2.

Tabel 3.2. Arc Penambah Subtour ke 1

Arc yang akan diganti	Arc yang ditambahkan ke subtour	Tambahan Jarak
(1,5) (Medan Tuntungan, Medan Johor)	(1,2) – (2,5) (Medan Tuntungan, Maimun) – (Maimun, Medan Johor)	$C_{12} + C_{25} + C_{15} = 88$
(1,5) (Medan Tuntungan, Medan Johor)	(1,3) – (3,5) (Medan Tuntungan, Medan Area) – (Medan Area, Medan Johor)	$C_{13} + C_{35} + C_{15} = 108$
(1,5) (Medan Tuntungan, Medan Johor)	(1,4) – (4,5) (Medan Tuntungan, Medan Baru) – (Medan Baru, Medan Johor)	$C_{14} + C_{45} + C_{15} = 83$
(5,1) (Medan Johor, Medan Tuntungan)	(5,2) – (2,1) (Medan Johor, Maimun) – (Maimun, Medan Tuntungan)	$C_{52} + C_{21} + C_{51} = 88$
(5,1)	(5,3) – (3,1)	$C_{52} + C_{31} +$

(Medan Johor,Medan Tuntungan)	(Medan Johor,Medan Area)(Medan Area,Medan Tuntungan)	$C_{51} = 108$
(5,1)	(5,4) – (4,1)	$C_{54} + C_{41} +$
(Medan Johor,Medan Tuntungan)	(Medan Johor,Medan Baru)(Medan Baru,Medan Tuntungan)	$C_{51} = 83$

Dari tabel 3.2 diperoleh tambahan jarak terkecil apabila arc(1,5) diganti dengan arc(1,4) dan arc(4,5) atau arc(5,1) diganti dengan arc(5,4) dan arc(4,1). dari kemungkinan tersebut,bisa dipilih salah satu.Misal dipilih kemungkinan pertama maka subtour yang baru menjadi $\rightarrow (1,4) \rightarrow (4,5) \rightarrow (5,1)$

- d. Selanjutnya dibuat tabel Arc Penambah Subtour ke 2 menyimpan objek yang bisa disisipkan dalam subtour beserta tambahan jaraknya, seperti ditampilkan dalam tabel 3.3.

Tabel 3.3 Arc Penambah Subtour ke 2

Arc yang akan diganti	Arc yang ditambahkan ke subtour	Tambahan Jarak
(1,2)	(1,3) – (3,2)	$C_{13} + C_{32} + C_{12} =$ 107
(1,2)	(1,4) – (4,2)	$C_{14} + C_{42} + C_{12} =$

		85
(2,5)	(2,3) - (3,5)	$C_{23} + C_{35} + C_{25} =$ 85
(2,5)	(2,4) - (4,5)	$C_{52} + C_{21} + C_{51} =$ 78
(5,1)	(5,3) - (3,1)	$C_{53} + C_{31} + C_{51} =$ 108
(5,1)	(5,4) - (4,1)	$C_{54} + C_{41} + C_{51} =$ 83

Dari tabel 3.3 diperoleh tambahan jarak terkecil adalah 83 dengan menggantikan arc(5,1) dengan arc(5,4) dan arc(4,1), sehingga subtour baru yang dihasilkan adalah: $\rightarrow (5,4) \rightarrow (4,1) \rightarrow (2,5) \rightarrow (1,2)$

Tabel 3.4. Arc Penambah Subtour ke 3

Arc yang akan diganti	Arc yang ditambahkan ke subtour	Tambahan Jarak
(1,4)	(1,3) - (3,4)	$C_{13} + C_{34} + C_{14} =$ 117
(4,2)	(4,3) - (3,2)	$C_{43} + C_{32} + C_{42} =$ 69
(2,5)	(2,3) - (3,5)	$C_{23} + C_{35} + C_{25} =$ 85

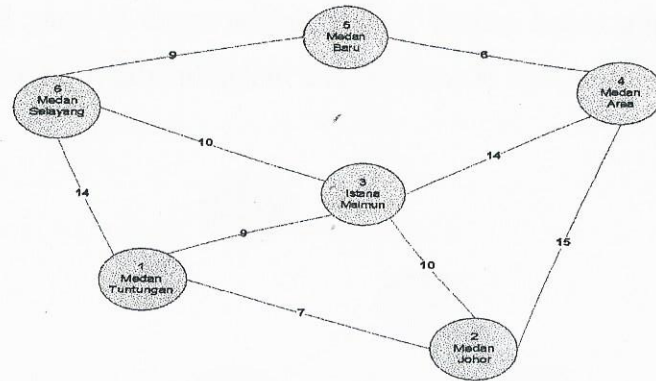
(5,1)	(5,3) - (3,1)	$C_{53} + C_{31} + C_{51} =$ 108
-------	---------------	-------------------------------------

Dari tabel 3.4 diperoleh tambahan jarak terkecil adalah 69 dengan menggantikan arc(4,2) dengan arc(4,3) dan arc(3,2), sehingga subtour baru yang dihasilkan adalah $\rightarrow(4,3) \rightarrow (3,2) \rightarrow (2,5) \rightarrow (5,1) \rightarrow (1,4)$.

Dari langkah-langkah tersebut diatas dapat diperoleh lintasan terpendek untuk mengunjungi 5 objek adalah $\rightarrow (4,3) \rightarrow (3,2) \rightarrow (2,5) \rightarrow (5,1) \rightarrow (1,4)$

3.1.3. Contoh Lain Pencarian Jalur Terpendek

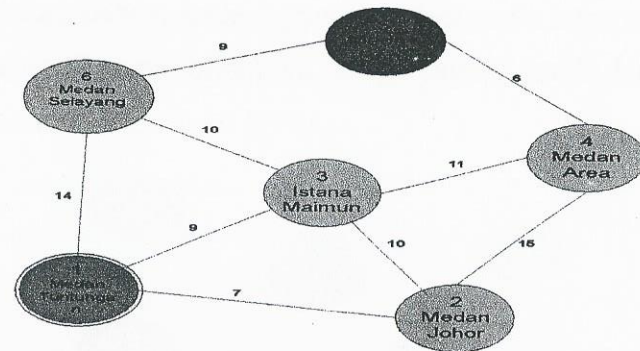
Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik ke titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka algoritma Genetika melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



Gambar 3.3 Contoh keterhubungan antar titik dalam algoritma Genetika

Dibawah ini penjelasan langkah per langkah pencarian jalur terpendek secara rinci dimulai dari node awal sampai node tujuan dengan nilai jarak terkecil.

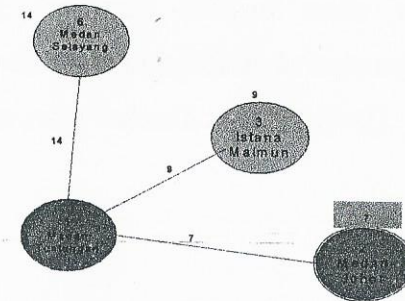
1. Node awal 1, Node tujuan 5. Setiap edge yang terhubung antar node telah diberi nilai



Gambar 3.4 Contoh kasus Genetika - Langkah 1

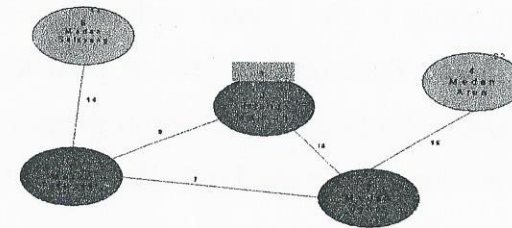
2. Genetika melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node 1), dan

hasil yang di dapat adalah node 2 karena bobot nilai node 2 paling kecil dibandingkan nilai pada node lain, nilai = 7 (0+7).



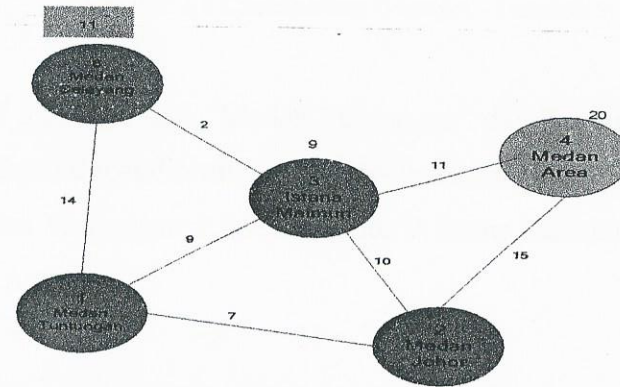
Gambar 3.5 Contoh kasus Genetika - Langkah 2

- Node 2 diset menjadi node keberangkatan dan di tandai sebagai node yang telah terjamah. Genetika melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node yang telah terjamah. Dan kalkulasi Genetika menunjukkan bahwa node 3 yang menjadi node keberangkatan selanjutnya karena bobotnya yang paling kecil dari hasil kalkulasi terakhir, nilai 9 (0+9).



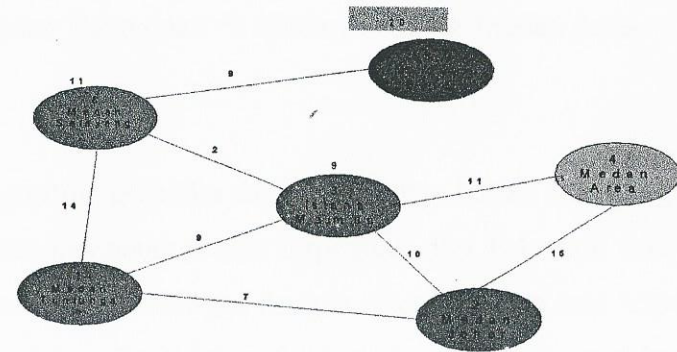
Gambar 3.6 Contoh kasus Genetika - Langkah 3

4. Perhitungan berlanjut dengan node 3 ditandai menjadi node yang telah terjamah. Dari semua node tetangga belum terjamah yang terhubung langsung dengan node terjamah, node selanjutnya yang ditandai menjadi node terjamah adalah node 6 karena nilai bobot yang terkecil, nilai 11 ($9+2$).



Gambar 3.7 Contoh kasus Genetika - Langkah 4

5. Node 6 menjadi node terjamah, Genetika melakukan kalkulasi kembali, dan menemukan bahwa node 5 (node tujuan) telah tercapai lewat node 6. Jalur terpendeknya adalah 1-3-6-5, dan nilai bobot yang di dapat adalah 20 ($11+9$). Bila node tujuan telah tercapai maka kalkulasi Genetika dinyatakan selesai.



Gambar 3.8 Contoh kasus Genetika - Langkah 5

Jadi jika Asal dari “Medan Tuntungan” dan Tujuan ke “Medan Area” dapat dapat dihitung untuk keseluruhan node yaitu :

1. Medan Tuntungan → Medan Johor → Istana Maimum → Medan Area
 $7 \rightarrow 10 \rightarrow 11 = 28$
2. Medan Tuntungan → Medan Johor → Medan Area
 $7 \rightarrow 15 = 22$
3. Medan Tuntungan → Medan Selayang → Medan Baru → Medan Area
 $14 \rightarrow 9 \rightarrow 6 = 29$
4. Medan Tuntungan → Medan Selayang → Istana Maimum → Medan Area
 $14 \rightarrow 2 \rightarrow 15 = 31$
5. Medan Tuntungan → Istana Maimum → Medan Area
 $9 \rightarrow 15 = 24$

Berarti jalur terpendek untuk untuk mencapai Medan Area dari Medan Tuntungan adalah :

Medan Tuntungan → Medan Johor → Medan Area

Catatan :

1. Algoritma genetika dapat digunakan ketika seseorang ingin cepat mengetahui rute terpendek dari beberapa rute yang akan dilalui sehingga dengan ditemukannya rute terpendek tersebut maka tidak perlu lagi menjalani rute tersebut satu persatu sampai ditemukan rute terpendek.
2. Pencarian rute terpendek ini dibuat dengan menggunakan program visual basic 6.0 untuk membantu mensimulasikan rute dari setiap objek yang akan dilalui

3.2. Perancangan

3.2.1. Menu Utama Sistem

Form ini digunakan sebagai menu utama dari program yang akan dibuat dan di dalamnya terdapat pilihan-pilihan untuk mengolah data peta dan jarak.

Pencarian Jalur Terpendek

Cari Jalur Terpendek

Jarak & Jalan

Koordinat Peta

Show Map

Copyright (C) : xxxxxx

Keluar

Gambar 3.9: Menu Utama Sistem

3.2.2. Form Jarak dan Peta

Form ini digunakan sebagai membuat nama jalan/objek baru serta jarak dari objek, yang gambarnya dapat dilihat seperti gambar dibawah ini :

3.2.3. Form Koordinat Peta

Form ini digunakan sebagai membuat dan mengubah koordinat dari sebuah jalur/peta, yang gambarnya dapat dilihat seperti gambar dibawah ini :

The diagram shows a rectangular form with a header section and a large map area below. The header section contains several input fields and buttons. Callout 1 points to the 'Jalan/Objek' text box. Callout 2 points to the 'X:' text box. Callout 3 points to the 'Y:' text box. Callout 4 points to the 'Simpan' button. Callout 5 points to the 'Tampilkan Peta' button. Callout 6 points to the 'Keluar' button. Callout 7 points to the large map area.

Gambar 3.11: Form Koordinat Peta

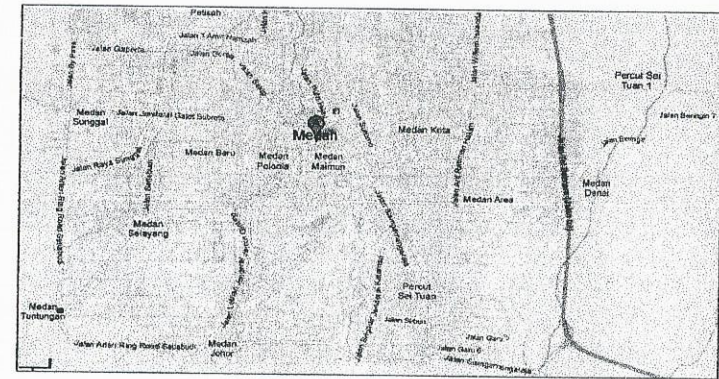
Keterangan

1. Textbok untuk memasukkan nama Objek
2. Textbook untuk memasukkan Koordinat X
3. Textbook untuk memasukkan Koordinat Y

4. Tombol untuk menyimpan data
5. Tombol untuk tampil peta
6. tombol untuk keluar dari form
7. List untuk menampilkan data jalur

3.2.4. Form Peta

Form ini digunakan untuk menampilkan peta kota medan, yang gambarnya dapat dilihat seperti gambar dibawah ini :



Gambar 3.12 : Form Peta

3.2.5 Perancangan Basis Data

Untuk mengimplementasikan algoritma, penulis menggunakan bantuan basis data. Adapun rancangan basis data yang digunakan adalah seperti tertera pada gambar.

Tabel 3.5 Jarak

Tabel jarak digunakan untuk menyimpan data objek-objek yang akan dikunjungi beserta jarak antar objek.

Field Name	Type	Size	Indexed	Decription
Asal	Varchar	50	Yes	Asal
Tujuan	Varchar	50	Yes	Tujuan
Jarak	Int	2	-	Jarak

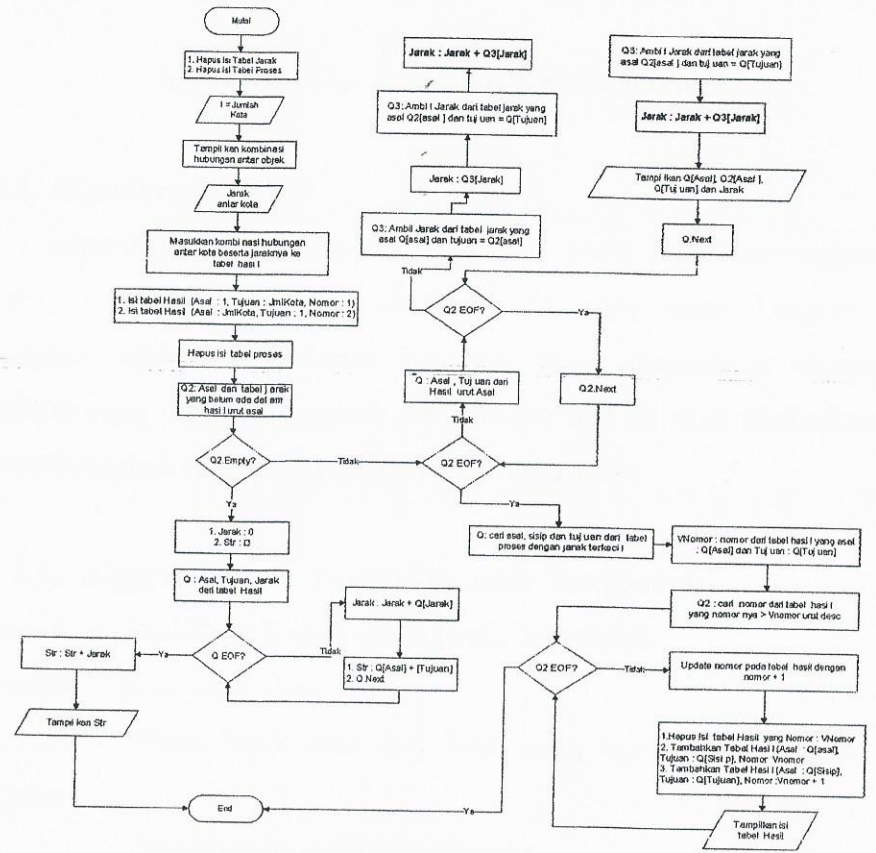
Tabel 3.6 Proses

Tabel proses digunakan sebagai fasilitastemporary dalam penyisipan subtour. Penggunaan tabel ini akan mempermudah dalam pengambilan informasi jarak minimal dari beberapa alternatif yang ada. Dalam pencarian jarak minimal digunakan query dengan fungsi agregasi min.

Field Name	Type	Size	Indexed	Decription
Asal	Varchar	50	Yes	Asal
Tujuan	Varchar	50	Yes	Tujuan
Jarak	Int	2	-	Jarak

3.2.6 Flowchart

Flowchart aplikasi yang akan dibangun dapat dilihat pada gambar dibawah ini.



Gambar 3.13. Flowchart

BAB IV

ALGORITMA DAN IMPLEMENTASI

4.1. Algoritma

Algoritma merupakan suatu metode yang digunakan untuk menyelesaikan suatu masalah. Algoritma merupakan langkah-langkah untuk perancangan program yang dinyatakan dalam bahasa yang dapat dimengerti manusia. Berikut ini akan dijelaskan secara ringkas cara kerja dari sistem yang di buat.

4.1.1. Algoritma Pada Pencarian Jalur Terpendek

Input : Asal Kota, Kota Tujuan, jarak Terpendek

Output : jalur yang akan dilalui

Proses : Baca Jarak Asal dan Jalur yang lain(i) dengan jarak tujuan

Jika $\text{jarak}(i) < \text{jarak}(i+1)$ maka

$\text{Jarak Terpendek} = \text{jarak}(i)$

$\text{Jalur} = \text{jarak Terpendek}$

4.1.2. Algoritma Pada Form Jarak dan Jalan

Algoritma ini adalah cara yang di gunakan untuk memasukkan proses ke dalam tabel **Jarak dan Jalan**

Input : jarak & jalan

Output : informasi jarak dan jalan

Proses :Masukkan Kelas lalu tekan enter, Jika Kelas yang di masukkan sudah ada pada tabel maka akan di tampilkan pesan salah dan setelah itu akan kembali ke awal dan jika “**Jarak & Jalan**” yang di masukkan belum ada pada tabel maka isian selanjutnya dapat di isi, Jika Data sudah selesai diisi maka tekan tombol “**Simpan**” untuk menyimpan data Kelas tersebut ke dalam database,Setelah Data disimpan maka akan kembali ke menu awal Untuk Mengubah Data, Double Klik pada “List Data” maka data yang akan diubah akan di tampilkan ke dalam form isian,Ubah Data **Jarak dan Jalan** yang ada sesuai dengan keinginan penamai Jika Data login sudah selesai di isi maka tekan tombol “Simpan” untuk menyimpan data tersebut ke dalam database.

4.1.3. Algoritma Pada Form Koordinat Peta

Algoritma ini adalah cara yang di gunakan untuk memasukkan proses ke dalam tabel **Koordinat Peta**.

Input : Nilai Koordinat

Output : Informasi Koordinat Peta

Proses : Jika Koordinat Pada Peta yang di masukkan sudah ada pada tabel maka akan di tempilkan pesan salah dan setelah itu akan kembali ke sebelumnya dan jika “**Koordinat Pada Peta**” yang di masukkan belum ada pada tabel maka isian selanjutnya dapat di

isi, Jika Data sudah selesai di isi maka, tekan tombol Simpan untuk menyimpan data Kelas tersebut ke dalam database, Setelah Data disimpan maka akan kembali ke awal pilihan.

4.2. Implementasi

Sistem pengolahan program merupakan suatu kesatuan pengolahan yang terdiri dari prosedur dan pelaksanaan data, Komputer sebagai sarana pengolahan program haruslah menyediakan fasilitas-fasilitas pendukung dalam pengolahan nantinya. Secara proporsional harus memenuhi akses yaitu :

1. Perangkat keras (Hardware)
2. Perangkat Lunak (SoftWare)

4.2.1. Perangkat keras

Spesifikasi perangkat keras (Hardware) yang digunakan untuk implementasi pembuatan dan menjalankan program, agar berjalan dengan baik adalah sebagai berikut :

1. Processor : Minimal Pentium III-800 Mhz
2. Memory : Minimal 512 MB
3. Hardisk : Minimal 160 GB
4. Kartu Grafik : VGA 8 MB dengan monitor SVGA
5. Monitor : SVGA 1924x78 pixel kedalam warna 16 Bit.

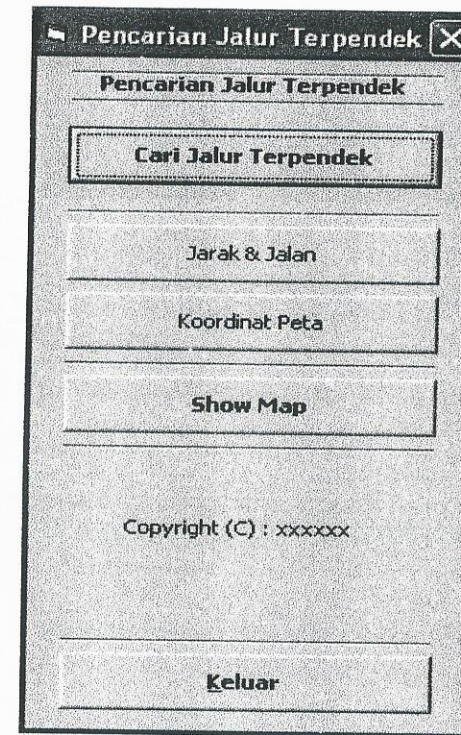
4.2.2. Perangkat Lunak

Dalam pembuatan program ini diperlukan beberapa software yaitu :

1. Sistem Operasi : Windows XP
2. Aplikasi : Visual Basic 6.0

4.2.3. Form Menu Utama

Form ini digunakan sebagai tampilan menu utama, seperti terlihat pada gambar dibawah ini :



Gambar 4.1: Menu Utama

4.2.4. Form Jarak dan Peta

Form ini digunakan sebagai membuat nama jalan/objek baru serta jarak dari objek, yang gambarnya dapat dilihat seperti gambar dibawah ini :

The screenshot shows a window titled "Input Jalan" with a close button in the top right corner. Inside the window, there are three input fields: "Asal" with the value "Johor", "Tujuan" with the value "Polonia", and "Jarak" with the value "60". To the right of these fields are three buttons: "Update Street", "Tampil Peta", and "Keluar". Below the input fields is a table with three columns: "Asal", "Tujuan", and "Jarak". The table contains the following data:

Asal	Tujuan	Jarak
Johor	Polonia	60
Johor	Tuntungan	4
Tuntungan	Selayang	7

Gambar 4.2 : Jarak Peta

4.2.5. Form Koordinat

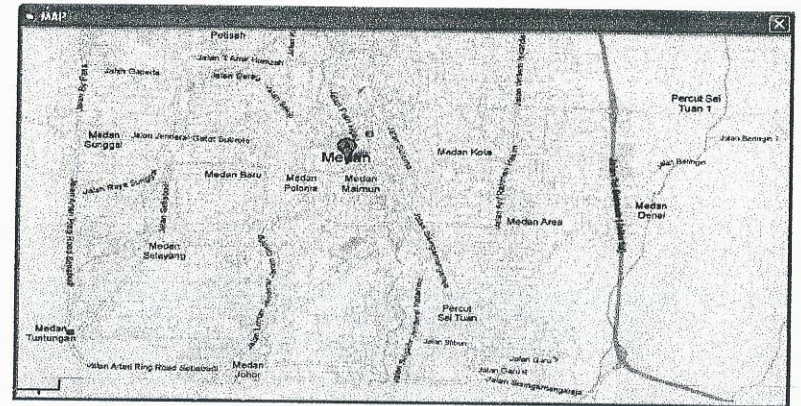
Form ini digunakan untuk membuat dan mengubah koordinat dari sebuah jalur/peta, yang gambarnya dapat dilihat seperti gambar dibawah ini :

Tuntungan	X	Y
Johor	252	426
Polonia	298	193
Selayang	157	281

Gambar 4.3 : Jarak Koordinat

4.2.6. Form Peta

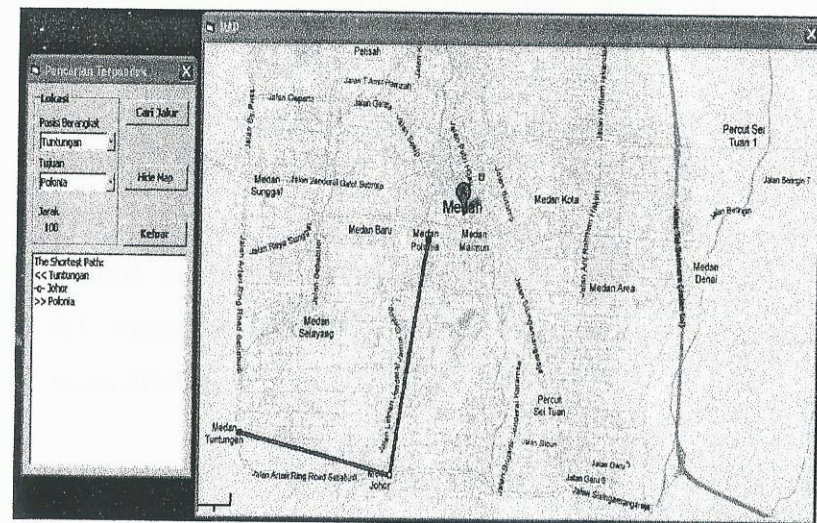
Form ini digunakan untuk menampilkan peta kota medan, yang gambarnya dapat dilihat seperti gambar dibawah ini :



Gambar 4.4 : Form Koordinat Peta

4.2.7. Form Pencarian Jalur Terpendek

Form ini digunakan untuk menampilkan jalur terpendek dari tujuan yang di pilih, yang gambarnya dapat dilihat seperti gambar dibawah ini :



Gambar 4.5 : Pencarian Jalur Terpendek

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Sebagai penutup pembahasan dalam penulisan penelitian, penulis mengambil kesimpulan-kesimpulan sekaligus memberikan saran kepada instansi yang menggunakan aplikasi rute terpendek kota Medan ini.

Dengan adanya kesimpulan dan saran ini dapatlah diambil suatu perbandingan yang akhirnya dapat memberikan perbaikan-perbaikan pada masa yang akan datang.

Adapun kesimpulan yang penulis peroleh adalah sebagai berikut:

1. Cara menentukan jalur terpendek adalah melalui setiap jalur yang ada dan menghitung jarak dari setiap jalur yang akan dilewati.
2. Untuk mencari rute terpendek dipakai algoritma genetika dengan menggunakan program visual Basic 6.0 untuk menghitung jarak dari masing-masing jalur tersebut.
3. Sistem pencarian dengan menggunakan program Visual Basic 6.0 untuk membantu mensimulasikan rute dari setiap jalur yang akan dilalui.

5.2 Saran

Adapun saran-saran yang akan penulis usulkan untuk meningkatkan kinerja instansi tersebut adalah sebagai berikut:

1. Pada sistem komputerisasi yang digunakan dalam pencarian jalur terpendek ini perlu digunakan program *anti virus* yang residen. Hal ini dimaksudkan untuk mencegah terjadinya kerusakan pada program ataupun data-data yang lain.
2. Proses pembuatan *file* cadangan (*back up*) sebaiknya dilakukan secara rutin.
3. Sistem ini masih memerlukan perbaikan-perbaikan yang lain seperti *inter face* sehingga diharapkan ada perbaikan yang berkelanjutan.

DAFTAR PUSTAKA

1. Suyanto (2008), Evolutionary computation. Bandung: Informatika.
2. <http://www.bimacipta.com/ga.htm> diakses jam 01.30 20 maret 2011.
3. <http://id.wikipedia.org/wiki/UML> diakses jam 19.00 21 maret 2011.
4. http://www.informatika.org/-rinaldi/stmik/2006/2007/makalah_2007 Diakses jam 19.00 20 maret 2011.
5. http://id.wikipedia.org/wiki/Algoritma_genetik
6. KusumaDewi, S ,2003, Artificial Intelligence, penerbit graham ilmu, Yogyakarta.
7. Desiani, A ,dan Arhami, M ,2006, Konsep Kecerdasan Buatan, Andi, Yogyakarta.
8. Novian, A ,2004, panduan MS.Visual Basic 6, Andi, Yogyakarta.
9. <http://www.Google.com>
10. Jogianto, "Analisa dan Desain", Andi Yogyakarta (1990),

