

**PENGEMBANGAN APLIKASI WORD MATCHING BAHASA INDONESIA  
MENGUNAKAN METODE ALGORITMA GENETIKA**



**Disusun Oleh :  
ROHMAD HANNY AFANDY  
08.18.027**

**PRODI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2013**

**LEMBAR PERSETUJUAN**

**PENGEMBANGAN APLIKASI WORD MATCHING BAHASA  
INDONESIA MENGGUNAKAN METODE ALGORITMA GENETIKA**

**SKRIPSI**

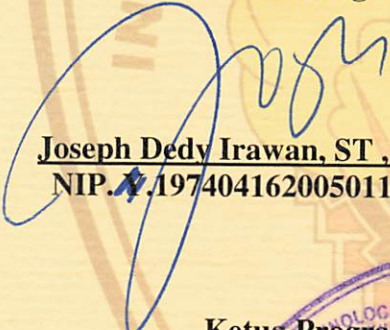
*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Informatika Strata Satu (S-1)*

**Disusun Oleh :**

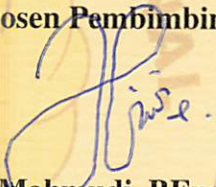
**ROHMAD HANNY AFANDY  
NIM : 08.18.027**

**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

  
**Joseph Dedy Irawan, ST, MT**  
**NIP. 197404162005011002**

**Dosen Pembimbing II**

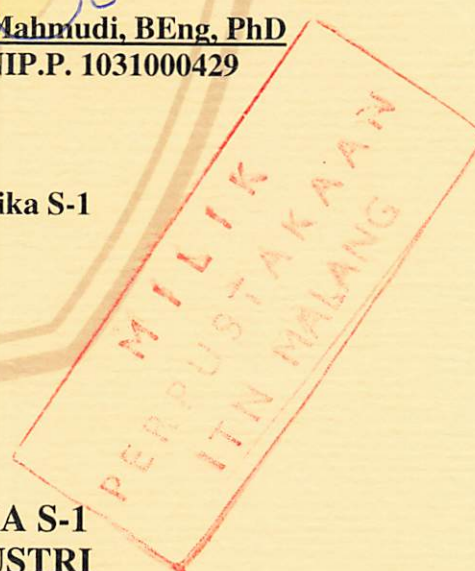
  
**Ali Mahmudi, BEng, PhD**  
**NIP.P. 1031000429**

**Mengetahui**

**Ketua Program Studi Teknik Informatika S-1**

  
**Joseph Dedy Irawan, ST, MT**  
**NIP. 197404162005011002**

**PRODI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2013**



# **PENGEMBANGAN APLIKASI WORD MATCHING BAHASA INDONESIA MENGGUNAKAN METODE ALGORITMA GENETIKA**

**Rohmad Hanny Afandy (08.18.027)**

**Program Studi Teknik Informatika S-1,  
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang  
Email: hanny\_afandy@ymail.com**

**Dosen Pembimbing: I. Joseph Dedi Irawan, ST , MT.  
II. Ali Mahmudi, BEng, PhD**

## **Abstrak**

*Pemanfaatan teknologi kecerdasan buatan (artificial intelligence) sebagai sistem yang cerdas atau pintar pun juga sudah semakin banyak digunakan dalam pemecahan suatu permasalahan. Tidak sedikit pula metode-metode yang dapat diimplementasikan didalam pemanfaatan teknologi tersebut, salah satu nya yaitu metode Algoritma Genetika (AG). Dalam pengembangannya metode algoritma genetika sendiripun juga dapat di buat dan di terapkan kedalam sebuah program komputer. dalam skripsi ini penulis ingin mengembangkan metode Algoritma Genetika (AG) itu sendiri ke dalam suatu perancangan dan pembuatan aplikasi yang disebut dengan Word Matching. Dengan berbasis kecerdasan buatan (artificial intelligence) yang di implementasikan ke dalam aplikasi tersebut, maka aplikasi tersebut dapat mampu memanggil secara otomatis beberapa kata kembali yang telah diinputkan ke dalam nya dengan baik dan benar.*

*Hasil pengujian sistem dapat disimpulkan bahwa dari 10 kata yang diinputkan secara acak 8 diantaranya benar dan dapat diketahui hasilnya dengan perbandingan kata yang terdapat di dalam sistem database nya dan memperoleh nilai fitness yaitu 0, sedangkan 2 diantaranya masih belum dapat diketahui kebenarannya yaitu dengan memperoleh nilai fitness masing-masing 2 dan 1. Hasil pengujian kepada user, menurut 25% user aplikasi ini belum cukup baik untuk mengetahui ejaan kata yang benar. Namun menurut 100% user menu-menu pada aplikasi ini cukup jelas.*

**Kata kunci:** *artificial intelligence, algoritma genetika, word matching.*

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : ROHMAD HANNY AFANDY

NIM : 08.18.027

Program Studi : Teknik Informatika S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Februari 2013

Yang membuat Pernyataan,



**Rohmad Hanny Afandy**

NIM. 08.18.027

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan skripsi dengan judul **“PENGEMBANGAN APLIKASI WORD MATCHING BAHASA INDONESIA MENGGUNAKAN METODE ALGORITMA GENETIKA”** dengan lancar. Skripsi merupakan persyaratan kelulusan di Program Studi Teknik Informatika S-1 ITN Malang dan untuk mencapai gelar Sarjana Komputer.

Keberhasilan penyelesaian laporan skripsi tidak lepas dari dukungan dan bantuan berbagai pihak, untuk itu penyusun menyampaikan terima kasih kepada:

1. Bapak Ir. Soeparno Djiwo, MT. selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Joseph Dedi Irawan, ST , MT. selaku Ketua Program Studi Teknik Informatika S-1, sekaligus Dosen Pembimbing I
4. Bapak Ali Mahmudi, BEng, PhD. selaku Dosen Pembimbing II
5. Orangtua yang selalu memberikan semangat dan motivasi selama menyusun skripsi
6. Teman-teman dan semua yang telah membantu dalam menyelesaikan penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Februari 2013

**Penulis**

## DAFTAR ISI

<b>LEMBAR PERSETUJUAN</b> .....	<b>i</b>
<b>ABSTRAK</b> .....	<b>ii</b>
<b>SURAT PERNYATAAN ORISINALITAS</b> .....	<b>iii</b>
<b>KATA PENGANTAR</b> .....	<b>iv</b>
<b>DAFTAR ISI</b> .....	<b>v</b>
<b>DAFTAR TABEL</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>ix</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xi</b>

### **BAB I PENDAHULUAN**

1.1.Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penulisan .....	2
1.4. Manfaat Penulisan .....	2
1.5. Batasan Masalah .....	2
1.6. Metodologi Penelitian .....	3
1.7. Sistematika Penulisan .....	4

### **BAB II LANDASAN TEORI**

2.1. Kecerdasan Buatan (Artificial Intelligence) .....	6
2.1.1. Pengertian Kecerdasan Buatan (Artificial Intelligence) .....	6
2.1.2. Perkembangan AI Pada Saat Ini .....	9
2.2. Metode Algoritma Genetika .....	10
2.2.1. Pengertian Algoritma Genetika .....	10
2.2.2. Beberapa Definisi Penting Dalam Algoritma Genetika .....	11
2.2.3. Hal-Hal Yang Harus Dilakukan Dalam Menggunakan Algoritma Genetika .....	12
2.2.4. Pengertian Individu .....	12

2.2.5.	Nilai Fitness .....	13
2.2.6.	Siklus Algoritma Genetika .....	13
2.3.	Pengertian dan Pengenalan Delphi .....	17
2.3.1.	Pengertian Delphi .....	17
2.3.2.	Keunggulan Delphi .....	18
2.3.3.	Komponen-Komponen Pada Delphi .....	20
2.4.	Microsoft Access .....	24

### **BAB III ANALISA DAN PERANCANGAN**

3.1.	Diagram Alir Perancangan Sistem Aplikasi .....	30
3.2.	Perancangan Sistem Aplikasi .....	31
3.2.1.	Deskripsi Aplikasi .....	31
3.2.2.	Identifikasi Masalah .....	31
3.3.	Perancangan Flowchart Sistem .....	32
3.3.1.	Gambaran Proses Eksekusi Data .....	33
3.4.	Desain Tampilan Aplikasi .....	35
3.4.1.	Desain Tampilan Form Awal .....	35
3.4.2.	Desain Tampilan Form Utama .....	35
3.4.3.	Desain Tampilan Form Daftar Kata .....	36
3.4.4.	Desain Tampilan Data Base .....	37

### **BAB IV IMPLEMENTASI DAN PENGUJIAN**

4.1.	Kebutuhan Sistem .....	38
4.2.	Implementasi Aplikasi .....	38
4.2.1.	Pembuatan Form Awal .....	26
4.2.2.	Pembuatan Form Utama .....	42
4.2.3.	Pembuatan Sistem Data Base .....	44
4.2.4.	Implementasi Algoritma Genetika Ke Program .....	44
4.3.	Pengujian Aplikasi .....	55
4.3.1.	Pengujian Halaman Form Awal .....	55

4.3.2. Pengujian Halaman Form Utama .....	55
4.3.3. Pengujian Sistem Aplikasi .....	56
4.3.4. Pengujian User .....	58

**BAB V PENUTUP**

5.1. Kesimpulan .....	59
5.2. Saran .....	59

<b>DAFTAR PUSTAKA .....</b>	<b>60</b>
-----------------------------	-----------



## **DAFTAR TABEL**

Tabel 4.1. Script Tombol Pada Form Awal .....	41
Tabel 4.2. Definisi Variabel Utama Program .....	45
Tabel 4.3. Definisi Private Declaration Function .....	46
Tabel 4.4. Code Program Function DecToBinStr.....	47
Tabel 4.5. Code Program Function BinStrToDec.....	48
Tabel 4.6. Code Program Function HtgFitnes .....	48
Tabel 4.7. Code Program Function HtgSelisihFitnes .....	49
Tabel 4.8. Code Program Proses Pertama .....	50
Tabel 4.9. Code Program Proses Kedua .....	51
Tabel 4.10. Code Program Proses Ketiga .....	53
Tabel 4.11. Code Program Proses Keempat .....	53
Tabel 4.12. Code Program Proses Kelima .....	54

## DAFTAR GAMBAR

Gambar 2.1. Bidang-Bidang Tugas (Task Domains) Dari AI .....	7
Gambar 2.2. Penerapan Konsep Kecerdasan Buatan di Komputer .....	8
Gambar 2.3. Contoh individu dalam TSP .....	12
Gambar 2.4. Contoh Individu Dalam TSP (Travelling Salesman Problem) .....	13
Gambar 2.5. Siklus Algoritma Genetika .....	13
Gambar 2.6. Pertukaran Gen 1 titik .....	15
Gambar 2.7. Pertukaran Gen 2 titik .....	15
Gambar 2.8. Mutasi Tingkat Kromosom .....	16
Gambar 2.9. Mutasi Gen .....	16
Gambar 2.10. Mutasi Tingkat Bit .....	17
Gambar 2.11. IDE Delphi .....	20
Gambar 2.12. Menu Bar dan Tool Bar .....	21
Gambar 2.13. Component Pallete .....	21
Gambar 2.14. Form Designer .....	22
Gambar 2.15. Code Explorer .....	23
Gambar 2.16. Object Treeview .....	23
Gambar 2.17. Object Inspector .....	24
Gambar 3.1. Diagram Alir Perancangan Sistem Aplikasi. ....	30
Gambar 3.2. Flowchart Aplikasi Secara Umum .....	32
Gambar 3.3. Gambaran Proses Eksekusi Data .....	34
Gambar 3.4. Desain Tampilan Form Awal .....	35
Gambar 3.5. Desain Tampilan Form Utama .....	36
Gambar 3.6. Desain Tampilan Form Daftar Kata. ....	37
Gambar 3.7. Desain Tampilan Data Base Kata .....	37
Gambar 4.1. Tampilan Form dan Unit .....	39
Gambar 4.2. Proses Memasukkan Image Pada Delphi7 .....	40

Gambar 4.3. Tampilan Form Awal. ....	41
Gambar 4.4. Tampilan Form Utama Sebelum Panel-Panel Utama Program Dibuat. ....	43
Gambar 4.5. Tampilan Panel-Panel Utama Program.....	44
Gambar 4.6. Tampilan Sistem Data Base. ....	44
Gambar 4.7. Hasil Pengujian Form Awal.....	55
Gambar 4.8. Hasil Pengujian Halaman Form Utama. ....	56

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang berhubungan dengan pengautomatisasian tingkah laku cerdas. Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada prinsip-prinsip aplikasi dari bidangnya. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya.

Dengan adanya perkembangan teknologi yang semakin pesat, Pemanfaatan teknologi kecerdasan buatan (*artificial intelligence*) sebagai sistem yang cerdas atau pintar pun juga sudah semakin banyak digunakan dalam pemecahan suatu permasalahan. Tidak sedikit pula metode-metode yang dapat diimplementasikan didalam pemanfaatan teknologi tersebut, salah satu nya yaitu metode Algoritma Genetika (AG). Dalam pengembangannya metode algoritma genetika sendiripun juga dapat di buat dan di terapkan kedalam sebuah program komputer.

Berdasarkan uraian diatas maka penulis ingin mengembangkan metode Algoritma Genetika (AG) itu sendiri ke dalam suatu perancangan dan pembuatan aplikasi yang disebut dengan Word Matching. Dengan berbasis kecerdasan buatan (*artificial intelligence*) yang di implementasikan ke dalam aplikasi tersebut, maka aplikasi tersebut nantinya dapat mampu memanggil secara otomatis beberapa kata kembali yang telah diinputkan ke dalam aplikasi tersebut dengan baik dan benar. Dengan didukung pemanfaatan bahasa pemrograman Delphi yang digunakan sebagai penulisan code programnya serta Microsoft Access sebagai pemnyimpanan data basenya. Maka dari itu penulis mengambil topik skripsi dengan judul “PENGEMBANGAN APLIKASI WORD MATCHING BAHASA INDONESIA MENGGUNAKAN METODE ALGORITMA GENETIKA”.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang pemilihan judul diatas, maka yang menjadi permasalahannya adalah bagaimana mengembangkan metode Algoritma Genetika (AG) tersebut ke dalam sebuah aplikasi dengan melakukan perancangan dan pembuatan aplikasi Word Matching yang nantinya dapat menampilkan kembali secara otomatis (heuristik) kata yang telah diinputkan di dalam nya dengan baik dan benar, serta dapat memanfaatkan bahasa pemrograman dan database yang ada di dalam aplikasi tersebut.

## **1.3 Tujuan Penelitian**

Tujuan dari penyusunan skripsi ini adalah :

1. Mengembangkan Metode Algoritma Genetika dalam bidang Optimasi
2. Merancang dan membuat program aplikasi Word Matching untuk menampilkan kembali kata yang telah diinputkan dengan secara otomatis (Heuristik) sesuai dengan metode Algoritma Genetika
3. Mengaplikasikan database ke dalam program guna sebagai kata saran yang ingin diinputkan
4. Mengaplikasikan bahasa pemrograman Delphi untuk mendukung sistem pembuatan aplikasi Word Matcing yang akan dibuat.

## **1.4 Manfaat Penelitian**

Manfaat dari penyusunan skripsi ini adalah :

1. Membantu mengembangkan metode Algoritma Genetika dengan cara mengimplementasikannya ke dalam sebuah aplikasi.
2. Pemanfaatan kecerdasan buatan (Artificial Intelligence) guna memberikan solusi dalam membantu pemecahan masalah secara otomatis.

## **1.5 Batasan Masalah**

Mengingat begitu luasnya masalah yang akan dibahas maka perlu adanya batasan masalah agar langkah-langkah dalam pemecahan masalah tidak

menyimpang dari tujuan yang hendak dicapai. Adapun batasan-batasan masalah tersebut adalah :

1. Penerapan Aplikasi Word Matcing yang dibuat bisa digunakan untuk menyeleksi kata dari  $\leq 10$  huruf saja.
2. Baik tidaknya hasil susunan huruf dari kata yang dipanggil kembali masih berdasarkan dengan besar kecilnya nilai fitness yang dihasilkan.
3. Susunan huruf dari kata yang bisa diinputkan hanya yang berbahasa Indonesia saja.
4. Kata yang di inputkan ke dalam database program masih terdiri dari beberapa kata saja.

## **1.6 Metodologi Penelitian**

Adapun metode penelitian yang digunakan adalah sebagai berikut:

### **1. Studi Literatur**

Studi literatur yang dilakukan yaitu dengan mencari data dan informasi dari berbagai sumber baik berupa buku, paper maupun dari situs – situs pada internet. Secara lebih spesifik kajian yang dipelajari pada tiap tahap penulisan skripsi ini adalah sebagai berikut :

- a. Kajian mengenai Algoritma Genetika (AG) sebagai metode dasar penyusunan pembuatan Word Matcing
- b. Kajian mengenai bahasa pemrograman Delphi sebagai media penulisan dan perancangan code programnya
- c. Kajian mengenai Microsoft Office Access sebagai media penulisan dan penyimpanan databasenya

### **2. Desain**

Dirancang dan dibuat guna untuk langkah awal dalam mengidentifikasi komponen-komponen sistem cerdas yang akan di buat secara rinci nantinya. Tahap ini dilakukan setelah tahap analisis sistem selesai dilaksanakan, hal itu dikarenakan agar hasil desain seseuai dengan apa yang diinginkan. Desain aplikasi ini meliputi flowchart ataupun teknik lain seperti sketsa dan lain-lain dalam mendukung desain sistem.

### 3. Implementasi

Secara umum pengimplementasian pembuatan aplikasi ini menggunakan bahasa pemrograman Delphi, dimana aplikasi tersebut dibuat dengan tujuan mampu memperbaiki susunan huruf dari kata tertentu sesuai target yang telah kita tentukan secara otomatis (heuristik). Titik utama dari aplikasi ini adalah terdapat pada heuristiknya tersebut, heuristik merupakan sistem cerdas yang akan memecahkan permasalahan sesuai dengan tingkat kecerdasannya. Dan Tingkat kecerdasan system ini dibangun dengan menggunakan sebuah metode yaitu metode Algoritma Genetika (AG). Jadi setelah diterapkannya algoritmanya maka sistem yang dibuat akan dapat bekerja sendiri untuk mencari solusinya (solusi=target).

### 4. Testing

Dalam tahap ini akan dilakukan uji coba sistem untuk mencari kesalahan atau kekurangan dari program atau aplikasi yang dibuat. Apakah sistemnya sudah dapat berjalan dengan baik, semua fungsi harus di uji cobakan, hal ini dilakukan agar dapat memenuhi kebutuhan yang sudah di definisikan sebelumnya.

## 1.7 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan tugas akhir ini, makasistematika penulisan disusun sebagai berikut :

### BAB I : PENDAHULUAN

Berisi latar belakang, Perumusan masalah, tujuan dan manfaat, metodologi dan sistematika penulisan.

### BAB II : LANDASAN TEORI

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang di lakukan.

### BAB III : ANALISIS DAN PERANCANGAN

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

#### **BAB IV : IMPLEMENTASI DAN PENGUJIAN**

**Bab ini berisi tentang pengujian aplikasi, struktur, dan tampilan aplikasi.**

#### **BAB V : PENUTUP**

**Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.**



## **BAB II**

### **LANDASAN TEORI**

Dalam Bab II ini akan di bahas dasar teori mengenai permasalahan yang berhubungan dengan perancangan dan pembuatan *Aplikasi Word Matching* . Dasar teori yang akan dibahas meliputi Kecerdasan Buatan (*Artificial Intelligence*), Metode Algoritma Genetika, *Delphi* dan *Microsoft Acces*. Kecerdasan Buatan (*Artificial Intelligence*). Metode Algoritma Genetika akan membahas tentang pengertian dasar, definisi dan siklus-siklus dari metode tersebut. *Delphi* akan membahas bahasa pemrograman yang digunakan untuk membuat *Aplikasi Word Matching*. *Microsoft Acces* akan membahas tentang database yang digunakan untuk proses penyimpanan data di dalam program.

#### **2.1 Kecerdasan Buatan (Artificial Intelligence)**

##### **2.1.1 Pengertian kecerdasan Buatan (Artificial Intelligence)**

*Artificial Intelligence* (AI) atau kecerdasan buatan merupakan cabang dari ilmu komputer yang berhubungan dengan pengautomatisasian tingkah laku cerdas. Pernyataan tersebut juga dapat dijadikan definisi dari AI. Definisi ini menunjukkan bahwa AI adalah bagian dari komputer sehingga harus didasarkan pada prinsip-prinsip aplikasi dari bidangnya. Prinsip-prinsip ini meliputi struktur data yang digunakan dalam representasi pengetahuan, algoritma yang diperlukan untuk mengaplikasikan pengetahuan tersebut, serta bahasa dan teknik pemrograman yang digunakan dalam mengimplementasikannya. [5]

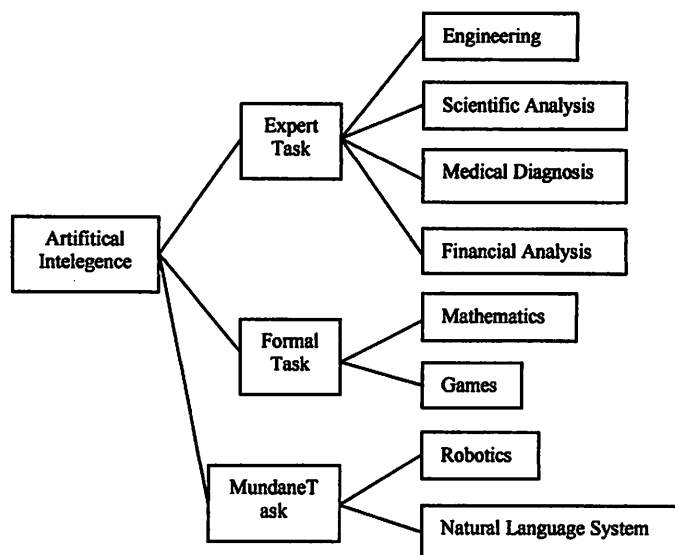
Dari beberapa perspektif, AI dapat dipandang sebagai:

1. Dari perspektif kecerdasan, AI adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan manusia
2. Dari perspektif bisnis, AI adalah sekelompok alat bantu (*tools*) yang berdayaguna dan metodologi yang menggunakan alat-alat bantu tersebut untuk menyelesaikan masalah- masalah bisnis.

3. Dari perspektif pemrograman, AI meliputi studi tentang pemrograman simbolik, pemecahan masalah, dan proses pencarian (*search*).
4. Dari perspektif penelitian :
  - a. Riset tentang AI dimulai pada awal tahun 1960-an, percobaan pertama adalah membuat program permainan catur, membuktikan teori, dan *general problem solving*.
  - b. *Artificial intelligence* adalah nama pada akar dari studi area.

Ada dua hal yang sangat mendasar mengenai penelitian-penelitian AI, yaitu *knowledge representation* (representasi pengetahuan) dan *search* (pelacakan). Para peneliti AI terus mengembangkan berbagai jenis teknik baru dalam menangani sejumlah permasalahan yang tergolong ke dalam AI seperti *vision* dan percakapan, pemrosesan bahasa alami, dan permasalahan khusus seperti diagnosa medis. [5]

AI seperti bidang ilmu lainnya juga memiliki sejumlah sub-disiplin ilmu yang sering digunakan untuk pendekatan yang esensial bagi penyelesaian suatu masalah dan dengan aplikasi bidang AI yang berbeda. Gambar 2.1 merupakan sejumlah bidang-bidang tugas (*task domains*) dari AI.



Gambar 2.1 Bidang-Bidang Tugas (Task Domains) Dari AI

Aplikasi penggunaan AI dapat dibagi ke dalam tiga kelompok, yaitu :

- a. *Mundane task*

Secara harfiah, arti *mundane* adalah keduniaan. Di sini, AI digunakan untuk melakukan hal-hal yang sifatnya duniawi atau melakukan kegiatan yang dapat membantu manusia. Contohnya :

1. Persepsi (*vision & speech*).
2. Bahasa alami (*understanding, generation & translation*).
3. Pemikiran yang bersifat *commonsense*.
4. *Robot control*.

**b. Format Task**

AI digunakan untuk melakukan tugas-tugas formal yang selama ini manusia biasa lakukan dengan lebih baik. Contohnya :

1. Permainan games
2. Matematika (geometri, logika, kalkulus, integral, pembuktian).

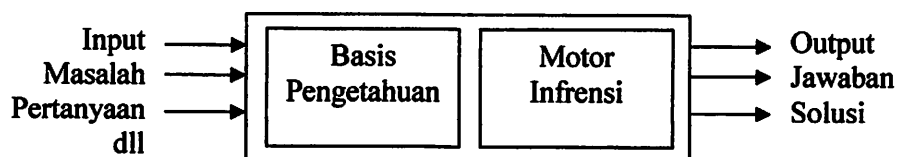
**c. Expert task**

AI dibentuk berdasarkan pengalaman dan pengetahuan yang dimiliki oleh para ahli. Penggunaan ini dapat membantu para ahli untuk menyampaikan ilmu-ilmu yang mereka miliki. Contohnya :

1. Analisis financial
2. Analisis medical
3. Analisis ilmu pengetahuan
4. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur)

Aplikasi *Artificial Inteligent* memiliki dua bagian utama, yaitu seperti pada Gambar 2.2 :

- a. Basis Pengetahuan (*Knowledge Base*) : berisi fakta - fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor Inferensi (*Inference Engine*) : kemampuan menarik kesimpulan berdasarkan pengalaman . [2]



*Gambar 2.2 Penerapan Konsep Kecerdasan Buatan di Ccomputer*

### 2.1.2. Perkembangan AI Pada Saat Ini

Bagaimana kondisi AI pada saat ini? Dengan semakin cepatnya perkembangan *hardware* dan *software*, berbagai produk AI telah berhasil dibangun dan digunakan dalam kehidupan sehari-hari. Produk-produk tersebut dikelompokkan ke dalam empat teknik yang ada di AI, yaitu *searching*, *reasoning*, *planning* dan *learning*. [6]

Teknik *searching* digunakan untuk pencarian rute optimum untuk memandu seseorang di perjalanan. Di kota Goteborg, Swedia, setiap taksi dilengkapi dengan sebuah Komputer dan *global positioning system* (GPS). Ketika penumpang menyebutkan suatu tempat tujuan di jalan *X*, maka sang sopir tinggal memasukkan tempat tujuan tersebut ke komputer. Di layar komputer akan terlihat rute jalan yang harus dilalui untuk menuju ke tempat tersebut. Sang sopir tinggal mengikuti panduan computer kapan belok kiri dan kapan harus belok kanan. Penumpang tidak merasa takut ditipu, dan sang sopir tidak perlu menghafal nama-nama jalan. Semuanya dipandu secara otomatis melalui program AI yang ada di komputer dan posisi taksi selalu dipantau oleh satelit GPS yang bias menentukan posisi taksi berdasarkan koordinat bumidengan akurasi tinggi. Produk AI lainnya adalah TomTom, yaitu sebuah *software* yang berfungsi sebagai *Tourguide* yang dapat di-*instal* di *handphone* ataupun PDA (*Personal Digital Assistant*). TomTom dapat membantu wisatawan untuk menjelajahi tempat-tempat menarik di berbagai kota di Eropa, Amerika dan Australia. [6]

*Software* permainan catur dan disebut *HITECH* adalah system AI pertama yang berhasil mengalahkan *grandmaster* dunia. *HITECH* adalah sistem berbasis pengetahuan yang menggunakan teknik *reasoning*. Dalam dunia kedokteran, telah berhasil dibangun sebuah *software* yang disebut *MedicWare* digunakan untuk merekam catatan medis pasien. Dokter dan pasien bias selalu berhubungan melalui *MedicWare* yang dapat di-*instal* di PDA. *MedicWare* dilengkapi dengan ribuan pengetahuan tentang jenis, merek, efek samping, dan interaksi berbagai jenis obat-obatan. Pasien maupun dokter bias melakukan pengecekan apakah suatu obat berbahaya jika si pasien memiliki sejarah medis tertentu. Misalnya, jika pasien sering sakit maag, maka

pasien atau dokter bias mencari obat yang aman bagi lambung menggunakan bantuan *software* tersebut.

Didunia manufaktur dan robotik, teknik *planning* memainkan peranan sangat penting. Optimum-AIV adalah suatu *planner (software)* yang menggunakan teknik *planning*) yang digunakan oleh *European Space Agency* untuk *Assembly* atau perakitan, *integration* atau penggabungan, dan *Verivication* (AIV) pesawat terbang. Software tersebut digunakan untuk membuat perancangan dan untuk memonitoring eksekusi terhadap perancangan-perancangan tersebut. [6]

Teknik *learning* telah digunakan pada berbagai bidang seperti transportasi, *speech processing*, *computer vision*, *robotics*, dan sebagainya. Sebuah system *auto driver* yang disebut ALVINN telah berhasil dibuat oleh Dean Pomerleau dari Carnegie Mellon University, USA. Dengan sistem ALVINN, sebuah mobil bias berjalan sendiri tanpa disetir oleh manusia. Sistem tersebut dibuat menggunakan Jaringan Syaraf Tiruan (JST) yang dilatih dengan berbagai gambar kondisi jalan raya yang ditangkap melalui kamera yang diletakkan di mobil. JST dilatih sedemikian rupa sehingga dapat menjadi lebih pintar untuk mengetahui kapan harus belok kiri, lurus atau belok kanan, dan seberapa tajam sudut beloknya. [6]

## **2.2 Metode Algoritma Genetika**

### **2.2.1. Pengertian Algoritma Genetika**

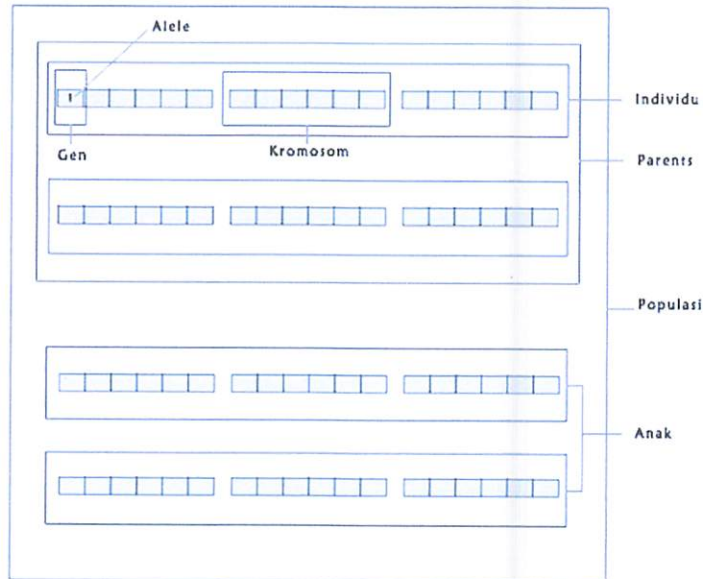
Algoritma genetika merupakan metode pencarian yang disesuaikan dengan proses genetika dari organisme-organisme biologi yang berdasarkan pada teori evolusi Charles Darwin. (Algoritma genetic pertama kali ditemukan oleh John Holland, itu dapat dilihat dalam bukunya yang berjudul *Adaption in Natural and Artificial Systems* pada tahun 1960-an dan kemudian dikembangkan bersama murid dan rekan kerjanya di Universitas Michigan pada tahun 1960-an sampai 1970-an. Tujuan Holland mengembangkan Algoritma Genetika saat itu bukan untuk mendesain suatu algoritma yang dapat memecahkan suatu masalah, namun lebih mengarah ke studi mengenai

fenomena adaptasi di alam dan mencoba menerapkan mekanisme adaptasi alam tersebut ke dalam sistem komputer. [9]

Algoritma Genetika yang dibuat Holland merupakan sebuah metode untuk memisahkan satu populasi kromosom (terdiri dari bit-bit 1 dan 0) ke populasi baru dengan menggunakan “seleksi alam” dan operator genetik seperti *crossover*, *mutation*, *inversion*. *Crossover* menukar bagian kecil dari dua kromosom, *mutation* mengganti secara acak nilai gen di beberapa lokasi pada kromosom, *inversion* membalikkan urutan beberapa gen yang berurutan dalam kromosom. Dasar teori inilah yang menjadi dasar kebanyakan program yang menggunakan algoritma genetika pada saat ini. [11]

### **2.2.2. Beberapa Definisi Penting Dalam Algoritma Genetika :**

1. Genotype (Gen), sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam algoritma genetika, gen ini bisa berupa nilai biner, float, integer maupun karakter, atau kombinatorial.
2. Allele, nilai dari gen.
3. Kromosom, gabungan gen-gen yang membentuk nilai tertentu.
4. Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat
5. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi.
6. Generasi, menyatakan satu-satuan siklus proses evolusi.
7. Nilai Fitness, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan. [1]



Gambar 2.3 Contoh individu dalam TSP

### 2.2.3. Hal-hal Yang Harus Dilakukan Dalam Menggunakan Algoritma Genetika

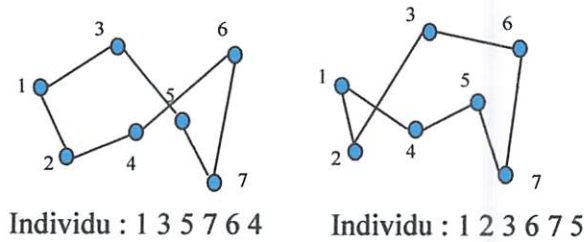
1. Mendefinisikan individu, dimana individu menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan yang diangkat
2. Mendefinisikan nilai *fitness*, yang merupakan ukuran baik-tidaknya sebuah individu atau baik-tidaknya solusi yang didapatkan.
3. Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak seperti *random-walk*.
4. Menentukan proses seleksi yang akan digunakan.
5. Menentukan proses perkawinan silang (*cross-over*).
6. Mutasi gen yang akan digunakan. [1]

### 2.2.4. Pengertian Individu

Individu bisa dikatakan sama dengan kromosom, yang merupakan kumpulan gen. Gen ini bisa berupa biner, float, dan kombinatorial.

Pada Gambar 2.3 berikut dijelaskan bahwa Individu menyatakan salah satu solusi yang mungkin. Misalkan dalam TSP (Travelling Salesman

Problem) individu menyatakan jalur yang ditempuh, dalam penentuan nilai maksimal dari  $F(x,y)$  individu menyatakan nilai  $(x,y)$ . [1]



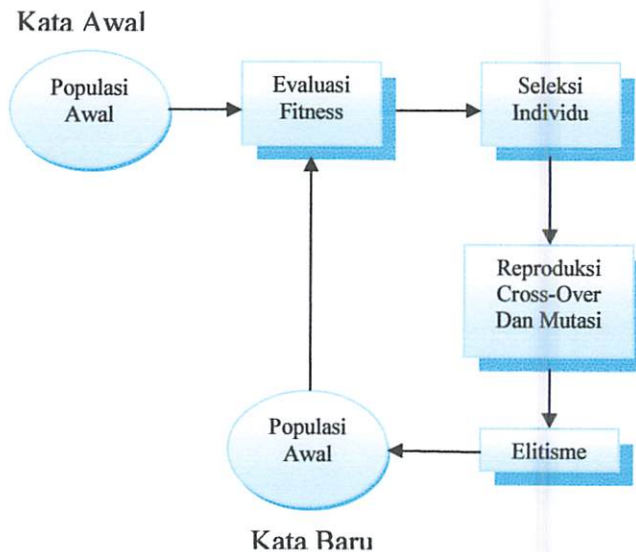
Gambar 2.4 Contoh individu dalam TSP (Travelling Salesman Problem)

### 2.2.5. Nilai Fitness

Nilai fitness adalah nilai yang menyatakan baik tidaknya suatu solusi (individu). Nilai fitness inilah yang dijadikan acuan dalam mencapai nilai optimal dalam algoritma genetika. Algoritma genetika bertujuan mencari individu dengan nilai fitness yang paling tinggi. Dalam TSP, karena TSP bertujuan meminimalkan jarak, maka nilai fitnessnya adalah inversi dari jarak. [5]

### 2.2.6. Siklus Algoritma Genetika

Berikut merupakan diagram siklus Algoritma Genetika yang tertera pada Gambar 2.5 :



Gambar 2.5 Siklus Algoritma Genetika.



### **1. Membangkitkan Populasi Awal**

Membangkitkan populasi awal adalah proses membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap individunya. Dalam TSP, populasi menyatakan sejumlah solusi (jalur) yang dicari secara acak. [1]

### **2. Seleksi**

Seleksi dilakukan untuk mendapatkan calon induk yang baik. “Induk yang baik akan menghasilkan keturunan yang baik”. Semakin tinggi nilai fitness suatu individu semakin besar kemungkinannya untuk terpilih. Seleksi dapat dilakukan dengan menggunakan dua macam teknik, yaitu mesin roulette, dan turnamen. Seleksi dilakukan dengan menggunakan prosentasi fitness setiap individu, dimana setiap individu mendapatkan luas bagian sesuai dengan prosentase nilai fitnessnya. [1]

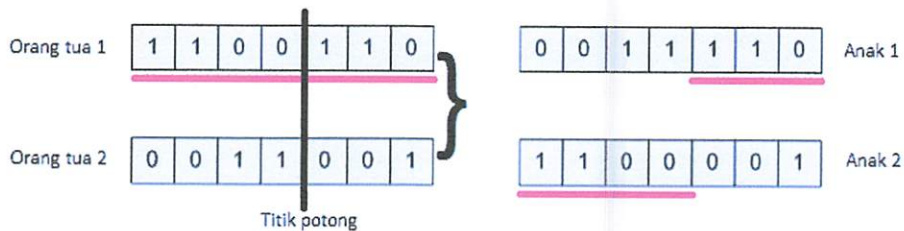
### **3. Cross Over**

Cross Over (Pindah Silang) merupakan salah satu operator dalam algoritma genetika yang melibatkan dua induk untuk menghasilkan keturunan yang baru. Cross over dilakukan dengan melakukan pertukaran gen dari dua induk secara acak. Proses cross over dilakukan pada setiap individu dengan probabilitas cross-over yang ditentukan. [8]

Crossover adalah operator Algoritma Genetika yang utama karena beroperasi pada dua kromosom pada suatu waktu dan membentuk offspring dengan mengkombinasikan dua bentuk kromosom. Cara sederhana untuk memperoleh crossover adalah dengan memilih suatu titik yang dipisahkan secara random dan kemudian membentuk offspring dengan cara mengkombinasikan segmen dari satu induk ke sebelah kiri dari titik yang dipisahkan dengan segmen dari induk yang lain ke sebelah kanan dari titik yang dipisahkan. Metode ini akan berjalan normal dengan representasi bit string. Performa dari Algoritma Genetika bergantung pada performa dari operator crossover yang digunakan.

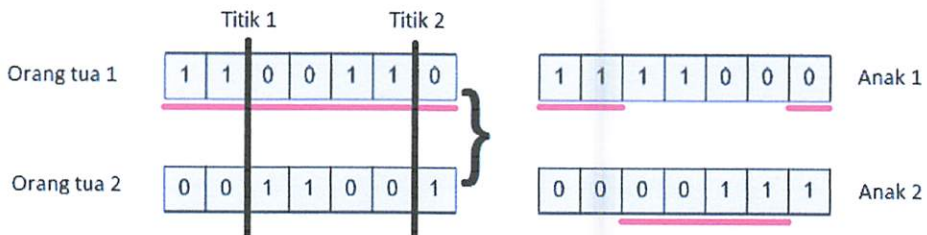
Crossover rate merupakan rasio antara jumlah offspring yang dihasilkan pada setiap generasi terhadap luas populasinya. Semakin tinggi crossover rate akan memungkinkan eksplorasi ruang solusi yang lebih luas dan mereduksi kemungkinan jatuh pada kondisi optimum yang salah. Namun memberikan rate yang memberikan konsekuensi makin lamanya waktu perhitungan yang diperlukan sebagai akibat eksplorasi pada luas populasi yang ada. Macam-macam Cross-Over yang banyak digunakan antara lain: [10]

- a. Pertukaran gen 1 titik, berikut tertera pada Gambar 2.6 :



Gambar 2.6 Pertukaran Gen 1 titik

- b. Pertukaran gen 2 titik, berikut tertera pada Gambar 2.7 :



Gambar 2.7. Pertukaran Gen 2 titik

#### 4. Mutasi Gen

pada Gambar 2.8 dijelaskan bahwa mutasi Gen merupakan operator yang menukar nilai gen dengan nilai inversinya, mialnya gennya bernilai 0 menjadi 1. Setiap individu mengalami mutasi gen dengan probabilitas mutasi yang ditentukan. Mutasi dilakukan dengan

memberikan nilai inversi atau menggeser nilai gen pada gen yang terpilih untuk dimutasikan.

Mutasi dapat dikatakan sebagai operasi pendukung yang menghasilkan perubahan secara acak dan seketika pada berbagai jenis kromosom. Cara mudah untuk mendapatkan mutasi dengan mengubah satu atau lebih genes. Pada Algoritma Genetika, mutasi memainkan peran penting, yaitu pertama, menggantikan genes yang hilang dari populasi selama proses seleksi, sehingga dapat diujikan pada suatu kondisi yang baru. Kedua, menyediakan genes yang tidak ditampilkan pada populasi awal. [1]

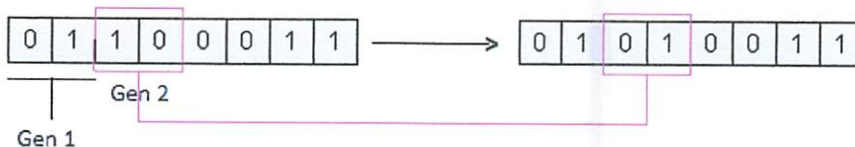
Mutation rate menyatakan presentase dari total jumlah genes dalam populasi. Mutation rate ini melakukan kontrol dimana genes baru dalam populasi dapat diuji seleksi. Jika rate terlalu kecil akan banyak genes yang sebenarnya bermanfaat tetapi tidak pernah diuji seleksi. Namun jika rate terlalu tinggi akan terjadi random perturbation, yang berakibat offspring mulai kehilangan kemiripan dengan induknya dan Algoritma Genetika akan kehilangan kemampuan untuk melihat urutan langkah observasinya. [10]

1. Mutasi pada tingkat kromosom: semua gen dalam kromosom berubah. Contoh:



Gambar 2.8 Mutasi Tingkat Kromosom

2. Mutasi pada tingkat gen: semua bit dalam satu gen akan berubah. Misal gen 2 yang akan mengalami mutasi maka semua bit yang ada dalam gen 2 tersebut akan berubah. Contoh:



Gambar 2.9 Mutasi Gen

### 3. Mutasi pada tingkat: Hanya satu bit yang berubah



Gambar 2.10 Mutasi Tingkat Bit.

## 5. Elitisme

Proses seleksi dilakukan secara random sehingga tidak ada jaminan bahwa suatu individu yang bernilai fitness tertinggi akan selalu terpilih. Walaupun individu bernilai fitness tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai fitnessnya menurun) karena proses pindah silang. Oleh karena itu, untuk menjaga agar individu bernilai fitness tertinggi tersebut tidak hilang selama evolusi, maka perlu dibuat satu atau beberapa kopinya. Prosedure ini dikenal sebagai elitisme. [11]

## 2.3 Pengertian dan Pengenalan Delphi

### 2.3.1. Pengertian Delphi

Delphi adalah suatu bahasa pemrograman (development language) yang digunakan untuk merancang suatu aplikasi program Delphi termasuk dalam pemrograman bahasa tingkat tinggi (*high level language*). Maksud dari bahasa tingkat tinggi yaitu perintah-perintah programnya menggunakan bahasa yang mudah dipahami oleh manusia. Bahasa pemrograman Delphi disebut bahasa prosedural artinya mengikuti urutan tertentu. Dalam membuat aplikasi perintah-perintah, Delphi menggunakan lingkungan pemrograman visual. [1]

Delphi merupakan generasi penerus dari Turbo Pascal. Pemrograman Delphi dirancang untuk beroperasi dibawah sistem operasi Windows. Program ini mempunyai beberapa keunggulan, yaitu produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuan dengan bahasa perogramma yang terstruktur dalam struktur bahasa perogramman *Object Pascal*.

Sebagian besar pengembang Delphi menuliskan dan mengkompilasi kode program di dalam lingkungan pengembang aplikasi atau *Integrated Development Environment* (IDE). Lingkungan kerja IDE ini menyediakan

sarana yang diperlukan untuk merancang, membangun, mencoba, mencari atau melacak kesalahan, serta mendistribusikan aplikasi. Sarana-sarana inilah yang memungkinkan pembuatan prototipe aplikasi menjadi lebih mudah dan waktu yang diperlukan untuk mengembangkan aplikasi menjadi lebih singkat. [10]

### 2.3.2. Keunggulan Delphi

Dewasa ini perkembangan *Software Development Tools* sangat pesat dengan persaingan yang sangat ketat. Masing-masing perangkat memiliki kelebihan dan kekurangan yang sangat tipis bila dibandingkan satu dengan yang lainnya. [4]

#### 1. Kemudahan Penyusunan User Interface

Kelebihan Delphi yang pertama adalah kemudahan penyusunan *User Interface*. Sejak awal dirilis, Delphi berkomitmen untuk menjadi *Rapid Application Development (RAD) tool*. Maksudnya adalah bagaimana menjadi perangkat yang mempercepat pengembangan aplikasi. Untuk RAD ini, Delphi telah melakukan dengan sangat baik, dimulai dari kemudahan penyusunan tampilan program.

Selain itu, dengan konsep komponen yang sangat terbuka, Delphi telah memungkinkan banyak pihak untuk mengembangkan berbagai macam komponen untuk berbagai macam kebutuhan. Sebutlah misalnya komponen TComPort yang banyak sekali digunakan untuk berkomunikasi dengan perangkat keras lain melalui *serial port*. Atau sekumpulan komponen dengan brand DevExpress yang digunakan untuk membuat tampilan program yang sangat indah dan *powerfull*. Bahkan, sekumpulan pengembang dari berbagai penjuru dunia telah berkumpul dan menghasilkan sejumlah besar komponen dengan nama JCL (*JEDI Component Library*) dan JVCL (*JEDI Visual Component Library*) yang *open source* dan dapat digunakan dengan gratis. Tapi, salah satu komponen favorit penulis adalah VisualTreeView, komponen gabungan antara TreeView dengan *Grid*. [4]

## 2. Bahasa Object Pascal

Bahasa yang digunakan dalam Delphi adalah Object Pascal, yang merupakan salah satu varian dari bahasa pascal dengan sejumlah penambahan, terutama terkait dengan konsep *Object Oriented Programming* (OOP).

Salah satu kelebihan bahasa pascal adalah mudah untuk dipelajari. Sejak awal dibuat, bahasa ini sudah digunakan untuk pengajaran bahasa pemrograman di banyak perguruan tinggi di seluruh dunia. Delphi Objected Pascal adalah salah satu versi Pascal yang sangat *powerful*, tetapi tidak menjadikannya terlalu kompleks.

Dengan kelebihan ini, Delphi bisa menjadi alat yang cukup baik untuk memahami konsep-konsep penting dalam pemrograman, seperti Algoritma dan Struktur Data, serta konsep Object Oriented Programming. Pemahaman yang baik terhadap Algoritma dan Struktur Data serta konsep OOP yang telah kita pelajari akan tetap melekat, dan tetap dapat digunakan ketika kita menggunakan bahasa pemrograman lain. [4]

## 3. Native Code

Hasil *compile* Delphi adalah kode *native* untuk Win 32. Ini berarti *file exe* yang dihasilkan oleh *compiler* akan langsung dijalankan oleh mesin tanpa melalui *software* lain seperti *virtual machine* (VM). Secara umum, *native code* tentu saja lebih cepat dari pada yang dijalankan melalui VM. Kelebihan lainnya adalah kemudahan *deployment*. Hasil dari Delphi adalah *file exe* tunggal, tanpa perlu *file-file* lainnya (kecuali menggunakan *library tertentu*).

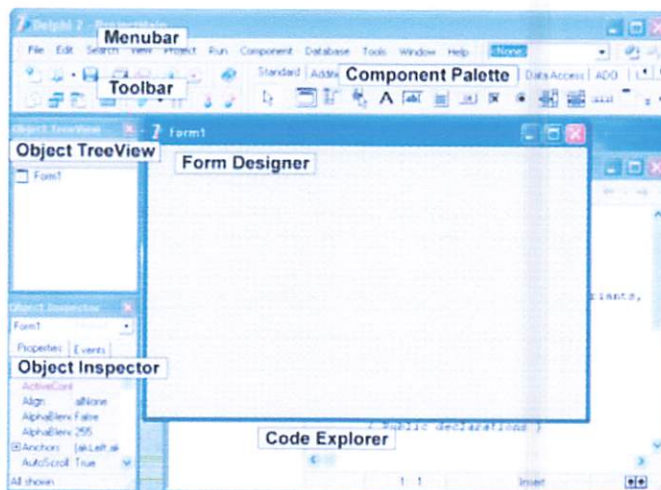
*Native code* adalah kemungkinan yang besar. Kita bisa membuat program yang sangat ringan, bisa menggunakan *installer* sederhana atau bahkan tanpa *installer* sama sekali. Karena ukurannya yang kecil, distribusinya akan jauh lebih mudah. [4]

### 2.3.3. Komponen–Komponen Pada Delphi

Semua *user interface* seperti *form*, tombol (*button*), dan objek *list-list* telah disertakan dalam Delphi dalam bentuk komponen atau *control*. Pengembang dapat dengan mudah menempatkan komponen-komponen tersebut ke dalam *form*. Pengembang dapat juga menempatkan *control ActiveX* pada *form* untuk membuat program-program khusus seperti *Browser Web* dalam waktu yang cepat. Delphi memungkinkan pengembang untuk merancang keseluruhan *interface* secara visual, dan dengan cepat dapat diimplementasikan sebuah kode perintah berbasis *event* (*event driven*) dengan mengklik mouse. Dengan IDE Delphi, pengembang perangkat lunak dapat membuat program *windows* dengan lebih cepat dan lebih mudah dari sebelumnya. [2]

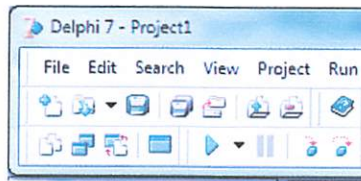
IDE adalah sebuah singkatan dari *Integrated Development Environment* yaitu sebuah lingkungan pengembangan yang terintegrasi, istilah IDE populer untuk menyebut *software* bahasa pemrograman dimana proses pengembangan programnya mulai dari *coding*, *designing* dan *debugging* dilakukan pada satu *framework* atau pada satu aplikasi yang terintegrasi.

IDE delphi pada Gambar 2.11 berikut dijelaskan yaitu terbagi menjadi tujuh bagian utama, yaitu : *menubar*, *toolbar*, *component palette*, *form designer*, *code explorer*, *object treeview* dan *object inspector*. [5]



Gambar 2.11 IDE Delphi

*Menubar* dan *toolbar* pada Gambar 2.12 berikut merupakan dua bagian yang biasanya terdapat pada aplikasi-aplikasi *windows* lain. Fungsi dari *menubar* dan *toolbar* ini relatif hampir sama dengan aplikasi *windows* lain, melainkan hanya *menubar* yang biasa digunakan dalam pembuatan program. [2]



Gambar 2.12 Menu Bar dan Tool Bar

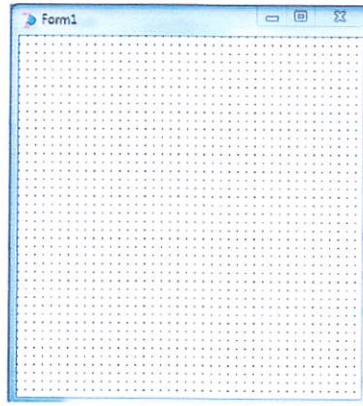
*Component Palette* pada Gambar 2.13 berisi kumpulan VCL (*Visual Component Library*) yang berguna dalam desain aplikasi. VCL merupakan pustaka untuk komponen visual, dimana dalam *component palette* dilambangkan dengan ikon yang merepresentasikan komponen tersebut. Komponen-komponen VCL pada *component palette* dikelompokkan ke dalam *tab*, sesuai dengan fungsinya, dengan maksud untuk memudahkan *programmer* dalam memilih komponen yang diinginkannya. Disamping VCL ada juga CLX (*Component Library for Cross Platform*) dalam desain aplikasi pemrograman Delphi. [2]



Gambar 2.13 Component Palette

Setiap aplikasi biasanya memiliki jendela atau *background interface*, yang dalam bahasa pemrograman Delphi atau bahkan dalam bahasa pemrograman lain yang berbasis visual, biasa disebut dengan *form*. *Form Designer* pada Gambar 2.14 berikut berfungsi sebagai tempat untuk mendesain *form* untuk aplikasi yang akan dibuat, dan juga sebagai tempat untuk meletakkan komponen-komponen yang kita ambil dari *component palette*. [2]

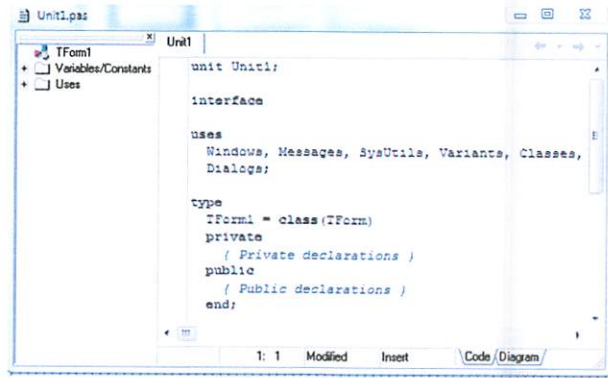




Gambar 2.14 Form Designer

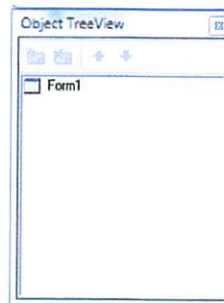
*Code Explorer* pada Gambar 2.15 merupakan area di mana kita menuliskan kode program, posisinya secara default terletak dibelakang *form*. Untuk menampilkan *code explorer* di depan form, bisa digunakan tombol F12 pada *keyboard*. Pada *code explorer* tersebut, akan terlihat kode-kode dalam bahasa pemrograman Delphi yang secara otomatis di'*generate*' oleh Delphi, hal ini jelas akan memudahkan atau mempercepat kita dalam menulis program. Pada *code explorer* Delphi, ada sebuah fitur yang disebut dengan *code completion*. Sesuai dengan namanya, *code completion* berfungsi melengkapi kode yang kita tulis dalam bentuk pilihan/*list* dari *code-code* yang bisa kita gunakan, hal ini akan sangat membantu apabila kita lupa terhadap kode tertentu. [2]

*Code completion* ini secara otomatis akan muncul ketika anda menekan tombol titik pada *keyboard*, selain itu untuk menampilkan *code completion* ini, anda juga bisa melakukannya dengan menekan kombinasi tombol Ctrl + *space* secara bersamaan. Disamping *code completion*, *code explorer* pada Delphi juga dilengkapi dengan *hint* (*layer* berwarna kuning yang muncul ketika mouse didekatkan pada komponen tertentu). *Hint* ini, muncul ketika kita menggunakan *procedure* atau fungsi dan menekan tombol pada *keyboard*, pada *hint* tersebut akan muncul nama dan tipe data dari parameter yang digunakan atau informasi lain yang diperlukan. [5]



Gambar 2.15 Code Explorer

*Object TreeView* pada Gambar 2.16 adalah bagian yang berisi daftar komponen yang digunakan dalam *form designer*. Dengan model tampilan bercabang, akan memudahkan kita dalam menunjuk komponen tertentu terutama jika aplikasi yang dibuat menggunakan banyak komponen. [5]



Gambar 2.16 Object Treeview

*Object Inspector* pada Gambar 2.17 adalah bagian yang digunakan untuk memanipulasi sifat atau karakteristik dan *event* dari komponen yang kita gunakan dalam *form designer*. Jendela *object inspector* terbagi menjadi dua bagian tab, yaitu *tab property* dan *tab event*. *Tab property* digunakan untuk memanipulasi *property* yang dimiliki oleh komponen tertentu, misalkan ukuran, warna dan *caption* dari komponen, sedangkan *tab event* digunakan untuk menangani pemasukan kode pada kejadian tertentu dari suatu komponen, misalnya : kejadian ketika komponen *button* (tombol) diklik atau *onClick*. [5]

Sebagaimana yang telah dijelaskan di atas, salah satu fungsi dari *object inspector* ini adalah untuk mengatur *property* dari komponen, pengaturan

tersebut tidak akan disimpan pada *file* unit, melainkan akan disimpan pada *file form* (\*.dfm). [5]



Gambar 2.17 Object Inspector

## 2.4 Microsoft Access

Microsoft Access (atau Microsoft Office Access) adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna. Versi terakhir adalah Microsoft Office Access 2007 yang termasuk ke dalam Microsoft Office System 2007. [11]

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik

pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek. [10]

Microsoft merilis Microsoft Access 1.0 pada bulan November 1992 dan dilanjutkan dengan merilis versi 2.0 pada tahun 1993. Microsoft menentukan spesifikasi minimum untuk menjalankan Microsoft Access 2.0 adalah sebuah komputer dengan sistem operasi Microsoft Windows 3.0, RAM berkapasitas 4 *megabyte* (6 *megabyte* lebih disarankan) dan ruangan kosong hard disk yang dibutuhkan 8 *megabyte* (14 *megabyte* lebih disarankan). Versi 2.0 dari Microsoft Access ini datang dengan tujuh buah disket *floppy* 3½ inci berukuran 1.44 *megabyte*. Perangkat lunak tersebut bekerja dengan sangat baik pada sebuah basis data dengan banyak *record* tapi terdapat beberapa kasus di mana *data* mengalami kerusakan. Sebagai contoh, pada ukuran basis data melebihi 700 *megabyte* sering mengalami masalah seperti ini (pada saat itu, memang *hard disk* yang beredar masih berada di bawah 700 *megabyte*). Buku manual yang dibawanya memperingatkan bahwa beberapa kasus tersebut disebabkan oleh *driver* perangkat yang kuno atau konfigurasi yang tidak benar. [10]

Nama kode (*codename*) yang digunakan oleh Access pertama kali adalah Cirrus yang dikembangkan sebelum Microsoft mengembangkan Microsoft Visual Basic, sementara mesin pembuat *form* antarmuka yang digunakannya dinamakan dengan Ruby. Bill Gates melihat purwarupa (*prototype*) tersebut dan memutuskan bahwa komponen bahasa pemrograman BASIC harus dikembangkan secara bersamaan sebagai sebuah aplikasi terpisah tapi dapat diperluas. Proyek ini dinamakan dengan Thunder. Kedua proyek tersebut dikembangkan secara terpisah, dan mesin pembuat *form* yang digunakan oleh keduanya tidak saling cocok satu sama lainnya. Hal tersebut berakhir saat Microsoft merilis Visual Basic for Applications (VBA). [10]

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan Microsoft

Internet Information Services (IIS) dan menggunakan Microsoft Active Server Pages (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada Microsoft SQL Server yang memiliki kemampuan yang lebih tinggi. Beberapa pengembang aplikasi profesional menggunakan Microsoft Access untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid Application Development/RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para *salesman*.

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka Access (*form, report, query*, dan kode Visual Basic) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya Microsoft Jet Database Engine (yang secara *default* digunakan oleh Microsoft Access), Microsoft SQL Server, Oracle Database, dan beberapa produk lainnya yang mendukung ODBC. [10]

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statement SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine. [10]

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, Microsoft JET Database Engine tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari Microsoft Access 2000 yang menggunakan Microsoft Jet Database

Engine versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperti halnya sebuah *stored procedure*, meskipun prosedurnya terbatas hanya untuk sebuah pernyataan tiap prosedurnya. Access juga mengizinkan *form* untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan form tersebut, dan merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*passthrough* dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya.

Dalam berkas Access Database Project (ADP) yang didukung oleh Microsoft Access 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (\*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau Microsoft SQL Server, ketimbang menggunakan Microsoft JET Database Engine. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampir semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints* dan *trigger*, *view*, *stored procedure*, dan *UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah *form*, *report*, *macro*, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut. [10]

Access mengizinkan pengembangan yang relatif cepat karena semua table basis data, kueri, form, dan report disimpan di dalam berkas basis data miliknya (\*.MDB). Untuk membuat Query, Access menggunakan Query Design Grid, sebuah program berbasis grafis yang mengizinkan para penggunanya untuk membuat query tanpa harus mengetahui bahasa pemrograman SQL. Di dalam Query Design Grid, para pengguna dapat memperlihatkan tabel basis data sumber dari *query*, dan memilih *field-field* mana yang hendak dikembalikan oleh proses dengan mengklik dan menyeretnya ke dalam *grid*. *Join* juga dapat dibuat dengan cara mengklik dan menyeret *field-field* dalam tabel ke dalam *field* dalam tabel lainnya. Access juga mengizinkan pengguna untuk melihat dan memanipulasi kode SQL jika memang diperlukan.

Bahasa pemrograman yang tersedia di dalam Access adalah Microsoft Visual Basic for Applications (VBA), seperti halnya dalam beberapa aplikasi Microsoft Office. Dua buah pustaka komponen Component Object Model (COM) untuk mengakses basis data pun disediakan, yakni Data Access Object (DAO), yang hanya terdapat di dalam Access 97, dan ActiveX Data Objects (ADO) yang tersedia dalam versi-versi Access terbaru. [10]

Komponen Utama (*Object*)

1. Table 

Table adalah objek utama dalam database yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek.

Table terdiri atas :


- a. Field Name : atribut dari sebuah table yang menempati bagian kolom.
- b. Record : Isi dari field atau atribut yang saling berhubungan yang menempati bagian baris.

2. Query ( SQL / Structured Query Language ) 


Query adalah bahasa untuk melakukan manipulasi terhadap database. Digunakan untuk menampilkan, mengubah, dan menganalisa sekumpulan data.

Query dibedakan menjadi 2, yaitu :

- a. DDL ( *Data Definition Language* ) digunakan untuk membuat atau mendefinisikan obyek-obyek database seperti membuat tabel, relasi antar tabel dan sebagainya.
- b. DML ( *Data Manipulation Language* ) digunakan untuk manipulasi database, seperti : menambah, mengubah atau menghapus data serta mengambil informasi yang diperlukan dari database.

3. Form 

Form digunakan untuk mengontrol proses masukan data (*input*), menampilkan data (*output*), memeriksa dan memperbaharui data.

4. Report 

Form digunakan untuk menampilkan data yang sudah dirangkum dan mencetak data secara efektif. [12]

Tipe data field-field dalam sebuah tabel harus ditentukan tipe datanya. Ada beberapa tipe data dalam Access, yaitu :

**1. Text**

Text digunakan untuk field alfanumeric (misal : nama, alamat, kode pos, telp), sekitar 255 karakter tiap fieldnya.

**2. Memo**

Memo dapat menampung 64000 karakter untuk tiap fieldnya, tapi tidak bisa diurutkan/diindeks.

**3. Number**

Number digunakan untuk menyimpan data numeric yang akan digunakan untuk proses perhitungan matematis.

**4. Date/Time**

**5. Currency**

**6. Auto Number**

**7. Yes/No**

**8. OLE Object**

OLE Object digunakan untuk eksternal objek, seperti bitmap atau file suara.

**9. Hyperlink**

**10. Lookup Wizard**

Jika menggunakan tipe data ini untuk sebuah field, maka bisa memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam combo box. [13]



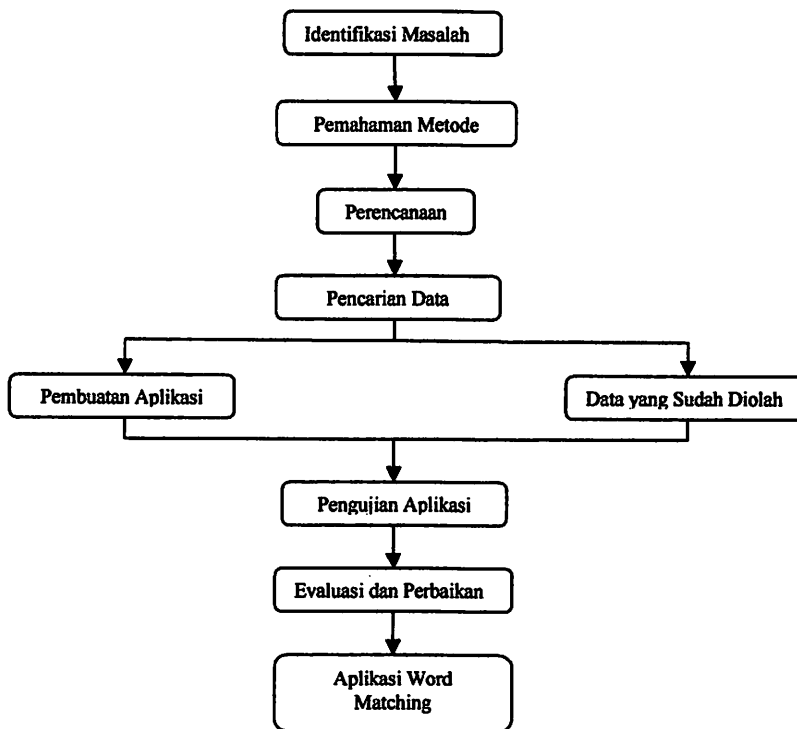
## BAB III

### ANALISIS DAN PERANCANGAN

Di dalam Bab III ini akan di bahas perancangan dan pembuatan sistem pada Aplikasi Word Matching. Perancangan sistem akan meliputi Diagram Alir Perancangan Sistem Aplikasi, Perancangan Sistem Aplikasi yang terdiri dari deskripsi aplikasi dan Identifikasi Masalah, Perancangan Flowchart Sistem Aplikasi, Gambaran Proses eksekusi Data, Desain Tampilan Aplikasi yang terdiri dari Desain Tampilan Form Awal, Desain Tampilan Form Utama, Desain Tampilan Form Daftar Kata dan Desain Tampilan data Base.

#### 3.1 Diagram Alir Perancangan Sistem Aplikasi

Diagram alir perancangan sistem aplikasi pada aplikasi word matching secara keseluruhan dapat dilihat pada Gambar 3.1.



*Gambar 3.1 Diagram Alir Perancangan Sistem Aplikasi .*

Diagram alir Perancangan Sistem Aplikasi menjelaskan tentang alir dari awal pembuatan aplikasi Word Matching yang dimulai dari tahap identifikasi masalah, pemahaman metode, perencanaan dan pencarian data. Selanjutnya proses dilakukan mulai dari penerapan metode ke dalam program sampai dengan pengolahan data yang akan digunakan, sehingga dapat menghasilkan program aplikasi. Sampai pada tahap pengujian aplikasi dapat disimpulkan melalui evaluasi dan perbaikan sehingga sampai mencapai tahap akhir dari pembuatan aplikasi Word Matching.

## **4.2. Perancangan Sistem Aplikasi**

### **4.2.1. Deskripsi Aplikasi**

Aplikasi Word Matching adalah merupakan sebuah aplikasi yang dirancang dan dibuat untuk pengembangan Algoritma Genetika (AG) yang dapat mampu memanggil kembali secara otomatis beberapa kata yang telah diinputkan ke dalam nya dengan baik dan benar.

Aplikasi Word Matching dirancang dengan menghasilkan output tampilan yang menarik dan sesederhana mungkin, hal itu dilakukan agar dapat mudah digunakan oleh penggunanya secara umum tanpa harus adanya tutorial dari penggunaan aplikasi tersebut.

### **4.2.2. Identifikasi Masalah**

Identifikasi masalah merupakan penjelasan mengenai beberapa alasan mengapa aplikasi ini dibangun. Tahapan perencanaan pembangunan aplikasi Word Matching, dikarenakan terdapat beberapa masalah yang teridentifikasi. Dan permasalahan yang teridentifikasi antara lain :

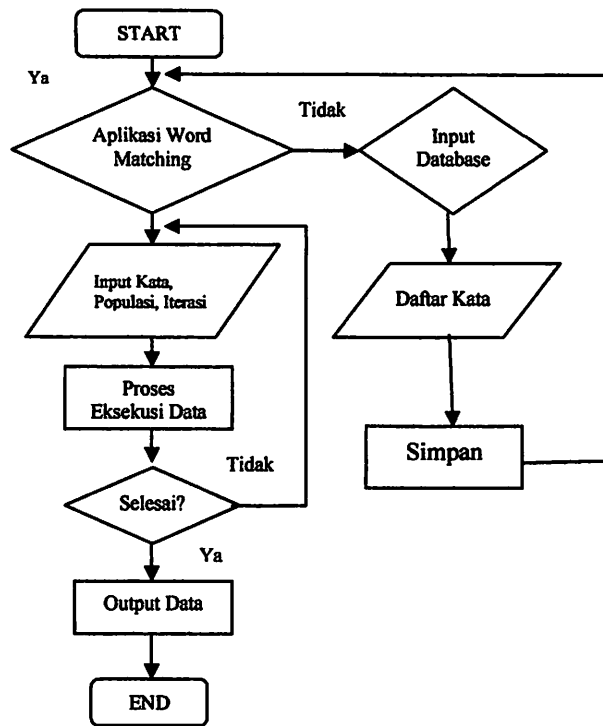
1. Masih minimnya pengembangan Metode Algoritma Genetika dalam bidang optimasi, yang seharusnya dapat dimanfaatkan untuk membangun sistem yang cerdas.
2. Bagaimana mengembangkan metode Algoritma Genetika (AG) dalam bidang optimasi dengan hasil sebuah aplikasi yang dapat berfungsi untuk menampilkan kembali susunan huruf dalam sebuah kata yang telah diinputkan di dalamnya dengan secara otomatis (Heuristik)

sesuai dengan metode Algoritma Genetika (AG) yang digunakan dan didasarkan pada kamus bahasa Indonesia yang baku dan benar

### 2.3. Perancangan Flowchart Sistem Aplikasi

*Flowchart* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *Flowchart* menjelaskan tentang urutan-urutan dari prosedur yang ada di dalam sistem dengan menggunakan simbol-simbol.

Secara umum Flowchart pembuatan Aplikasi Word Matching seperti pada gambar 3.2 :



Gambar 3.2 Flowchart Aplikasi Secara Umum

Keterangan Flowchart :

1. Start

Start menerangkan bahwa suatu sistem dimulai. Sebagai langkah awal sistem akan dimulai.

2. Aplikasi Word Matching

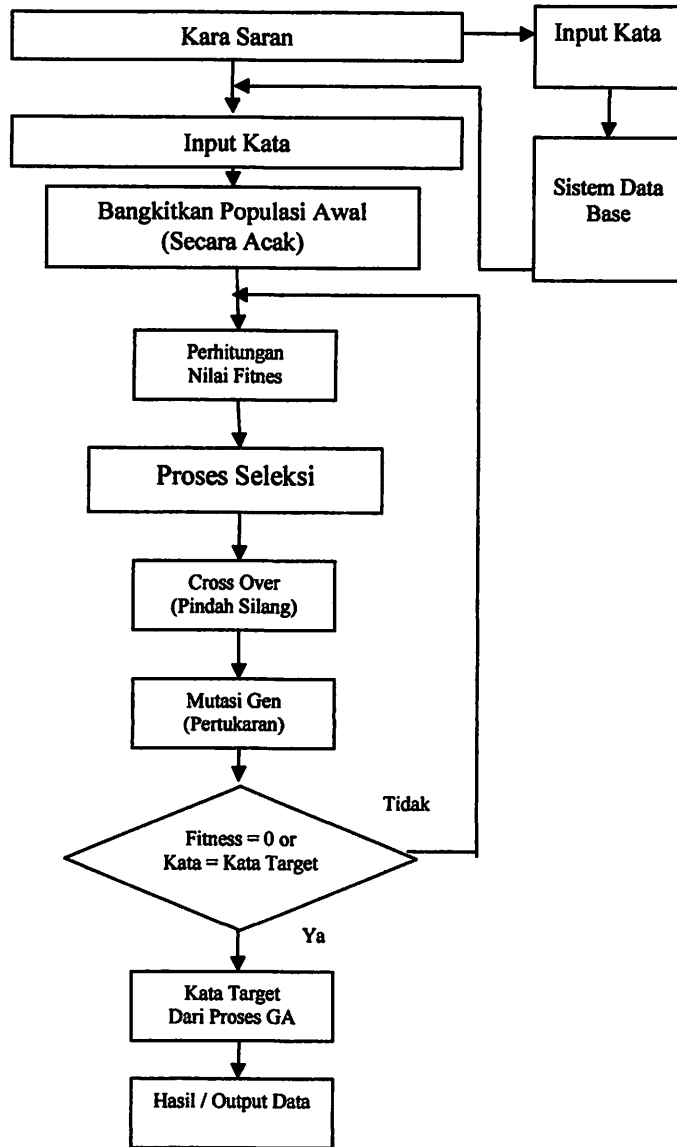
Masuk Ke Menu Aplikasi Word Matching

3. **Input Kata, Populasi Iterasi**  
Merupakan proses untuk penginputan data, inputan berupa kata / kalimat pendek, input populasi dan iterasi
4. **Proses Eksekusi Data**  
Dilakukan proses Eksekusi dari data inputan menggunakan sistem cerdas yang sudah dibangun yaitu dengan menerapkan metode algoritma genetika dan dibantu dengan database yang ada didalam programnya.
5. **Output Data**  
Merupakan outputan dari hasil eksekusi yang telah dilakukan
6. **Input Data**  
Menginputkan kata kedalam Sistem database berupa kata baru
7. **Daftar Kata**  
Proses Penginputan kata kedalam sistem database
8. **Simpan**  
Proses Simpan kata baru kedalam sistem database
9. **End**  
Menerangkan bahwa suatu proses sistem telah selesai

### **3.3.1. Gambaran Proses Eksekusi Data**

Proses dimulai dengan memasukkan kata saran kedalam kolom kata saran yang diperoleh dari sistem database nya, selanjutnya yaitu memasukkan susunan huruf dari kata saran ke dalam kolom input kata. Kemudian dilakukan pembangkitan populasi awal yang dilakukan secara acak yang nantinya akan digunakan untuk penghitungan selisih nilai fitness dari huruf kata tersebut. proses selanjutnya yaitu dilakukan adalah proses seleksi, poses seleksi dalam Algoritma genetika ada dua yaitu dengan cara *Cross Over* (Pindah Silang) dan *Mutasi Gen* (Pertukaran), setelah proses seleksi yang dilakukan akan dilakukan proses penghitungan nilai fitness kembali dari huruf kata yang sudah diseleksi, apakah nilai fitness itu = 0 atau kata = kata target, kalau tidak maka proses akan kembali lagi ke pembangkitan populasi awal yang dilakukan secara acak lagi, dan proses ini akan dilakukan berulang-ulang sampai memperoleh nilai fitness = 0 atau kata = kata target. Setelah itu proses akan berhenti dan memperoleh kata target yang telah dihasilkan dari proses Algoritma Genetika.

Pada Gambar 3.3 berikut ini adalah merupakan penggambaran Proses Eksekusi Data yang ada didalam program :



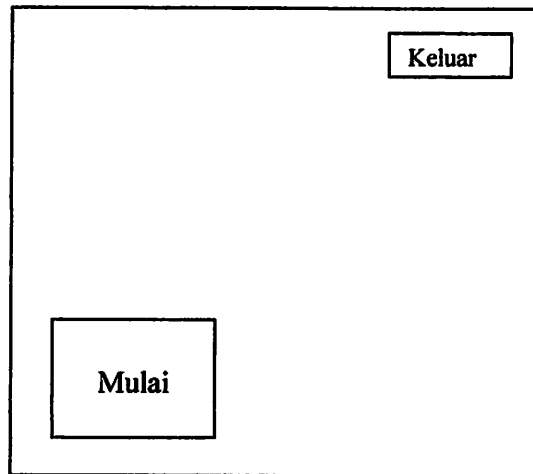
Gambar 3.3 Gambaran Proses Eksekusi Data.

### 3.4 Desain Tampilan Aplikasi

#### 3.4.1. Desain Tampilan Form Awal

Di dalam desain tampilan form awal, terdiri dari button mulai yang berfungsi untuk tombol masuk kedalam form utama, dan button keluar yaitu tombol yang berfungsi untuk keluar atau mengakhiri program.

Desain Tampilan Form Awal seperti pada Gambar 3.4 :



*Gambar 3.4 Desain Tampilan Form Awal.*

#### 3.4.2. Desain Tampilan Form Utama

Di dalam desain tampilan form utama, terdiri dari tombol data kata yang berfungsi untuk masuk kedalam form daftar kata, kolom 1 yang disediakan untuk menginputkan kata yang akan di tampilkan kembali, kolom 2 yang disediakan untuk menginputkan kata saran / kata yang terdapat di dalam database nya, kolom 3 yang disediakan untuk menginputkan berapa jumlah populasi yang akan dimasukkan, kolom 4 yang disediakan untuk menginputkan berapa jumlah iterasi yang akan dimasukkan, kolom ke 5 yang disediakan untuk menampilkan jumlah perulangan yang dilakukan oleh program, button proses tombol yang berfungsi untuk memulai eksekusi kata yang sudah diinputkan di dalam kolom input kata, kolom individu untuk menampilkan jumlah individu

yang terseleksi oleh program, kolom hasil untuk menampilkan nilai fitness dan hasil kata yang sudah di eksekusi oleh program, kolom 6 untuk menampilkan proses aplikasi yang sedang berjalan saat mengeksekusi program dan button kembali tombol yang berfungsi untuk kembali ke form awal.

Desain Tampilan Form Utama seperti pada Gambar 3.5 :

Gambar 3.5 Desain Tampilan Form Utama

### 3.4.3. Desain Tampilan Form Daftar Kata

Di dalam desain tampilan Form Daftar Kata, terdiri dari kolom 1 yang disediakan untuk menginputkan kata yang nantinya akan dimasukkan kedalam data base, Button baru yang berfungsi untuk mengganti kata inputan baru yang ada di dalam kolom 1, button simpan yang berfungsi untuk menyimpan kata yang sudah inputkan dari kolom 1 ke dalam

database, button hapus yang berfungsi untuk menghapus kata yang ada di dalam database, dan kolom kata untuk menampilkan kata yang ada di dalam database.

Desain Tampilan Form Daftar Kata seperti pada Gambar 3.6 :

Gambar 3.6 Desain Tampilan Form Daftar Kata

#### 3.4.4. Desain Tampilan Database

Didalam Database kata terdapat dua kolom yaitu kolom kata dan kolom selisih, dimana kolom kata digunakan untuk menyimpan kata yang telah diinputkan dari form daftar kata, kolom nilai untuk menginputkan berapa jumlah nilai keseluruhan huruf dari kata tersebut, dan kolom panjang yang digunakan untuk menuliskan berapa jumlah huruf dari kata di dalam kolom kata.

Desain Tampilan Database Kata seperti pada Gambar 3.7 :

Kata	Nilai	Panjang

Gambar 3.7 Desain Tampilan Database Kata



## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

Dalam Bab IV ini akan di bahas pembuatan sistem pada Aplikasi Word Matching. Pembuatan sistem dilakukan dengan melihat dari rancangan sistem yang sudah dibuat sebelumnya, dan dengan didukung oleh software-software yang akan digunakan sesuai kebutuhannya sehingga nantinya mampu menghasilkan aplikasi yang bisa di jalankan dengan baik pada tahapan pengujian.

#### **4.1 Kebutuhan Sistem**

Kebutuhan yang digunakan untuk membuat Aplikasi Word Matching, yaitu meliputi perangkat lunak dan perangkat keras yang dihubungkan dengan sistem operasi. Perangkat lunak yang digunakan untuk membuat Aplikasi Word Matching , meliputi :

1. Perangkat lunak *Delphi 7*, berfungsi sebagai perangkat utama, yaitu untuk perancangan form tampilan aplikasi dan penulisan *code program* yang akan disesuaikan dengan tahapan dari metode Algoritma Genetika yang akan digunakan.
2. Perangkat lunak *Microsoft Office Acces 2007*, berfungsi untuk penulisan dan penyimpanan database, yang nantinya akan digunakan sebagai dasar dari pembenahan huruf dari suku kata di dalam Aplikasi Word Matching
3. Notepad, sebagai editor

Sedangkan perangkat keras yang digunakan dalam pembuatan Aplikasi Word Matching, meliputi :

1. Mainboard Intel Centrino 2.
2. Processor Intel Centrino 2.4 GHz.
3. Memory 2048 MB.
4. VGA NVIDIA GeForce 9600M GS 512 MB.
5. Mouse dan Keyboard.

Sistem operasi yang digunakan sebagai penghubung antara perangkat lunak dan perangkat keras dalam pembuatan Aplikasi Word Matching, yaitu menggunakan sistem operasi Windows 7 Ultimate.

## 4.2 Implementasi Aplikasi

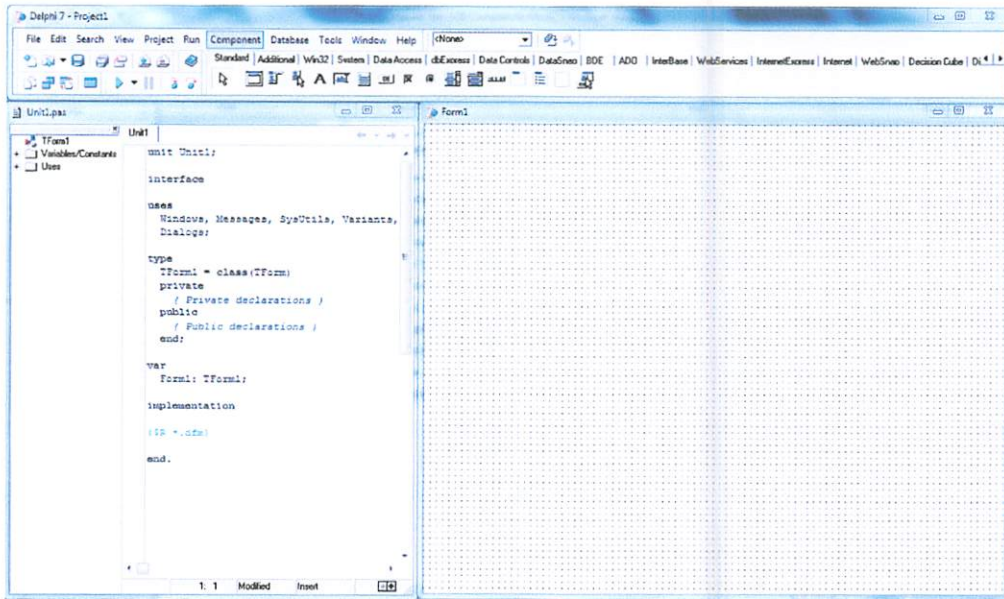
Pada proses implementasi ke dalam Aplikasi Word Matching, membutuhkan beberapa tahapan-tahapan, yang meliputi :

### 4.2.1. Pembuatan Form Awal

Pembuatan form awal, adalah halaman yang digunakan sebagai awalan sebelum masuk ke dalam Aplikasi Word Matching. Tahapan-tahapan yang diperlukan dalam pembuatan form awal, antara lain :

1. Menjalankan perangkat lunak *Delphi 7*.
2. Membuat aplikasi baru dengan *form* sebagai media pembuatan desain dan *unit* untuk penulisan *code program*.

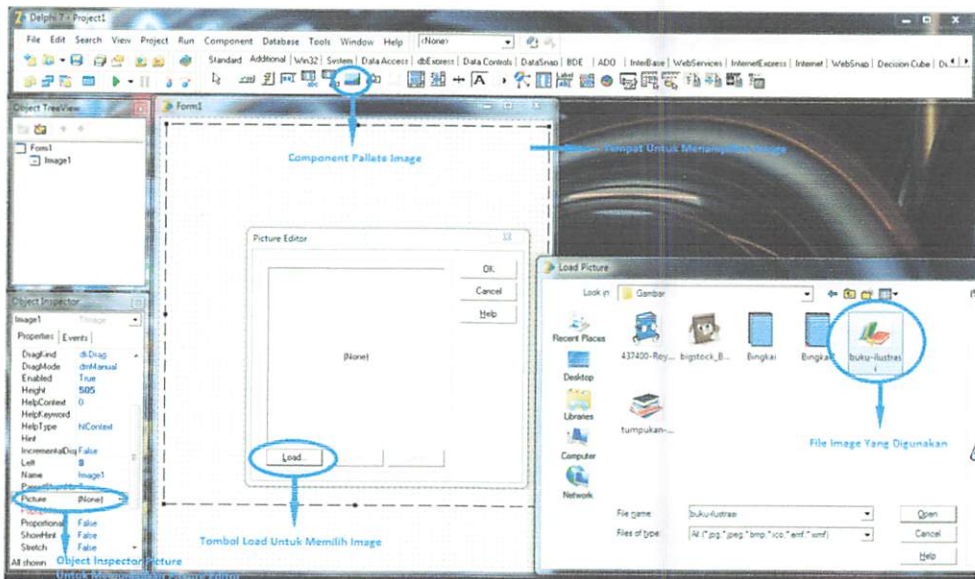
Tampilan *form* dan *unit* yang akan digunakan dapat dilihat pada gambar 4.1 berikut :



Gambar 4.1 Tampilan Form dan Unit

Setelah *form* dan *unit* tersedia, maka langkah selanjutnya yaitu memasukkan gambar ke dalam *form1* yang sudah disediakan sebelumnya. Tahapan-tahapan yang dilakukan yaitu :

1. Dengan cara memilih *component palette standard, image* dan *click* kiri dengan *mouse* pada *form1* yang telah disediakan.
2. Setelah *component palette image* tersedia, langkah selanjutnya yaitu memasukkan image tersebut kedalamnya, yaitu dengan cara di dalam *object inspector* pilih *properties picture* maka akan muncul *picture editor* dan pilih *load* selanjutnya cari di dalam folder penyimpanan *image* yang akan dimasukkan. Proses tahapan tersebut dapat dilihat pada gambar 4.2 berikut :



Gambar 4.2 Proses Memasukkan Image Pada Delphi7

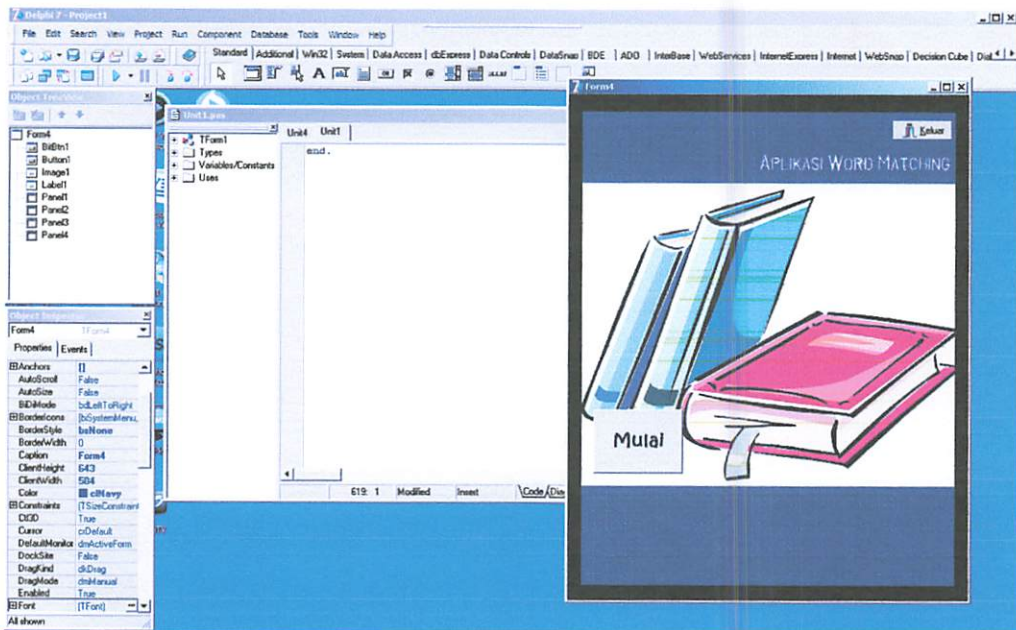
3. Proses selanjutnya yaitu menambahkan *component palette label* kedalam form sebagai tempat untuk menampilkan tulisan, memberi warna *form* dan meletakkan *component palette panel 1,2,3* dan *4* sebagai garis pinggir tampilan.
4. Memasukkan *component palette button* kedalam form, yaitu *button* masuk yang berfungsi untuk masuk kedalam halaman utama Aplikasi Word Matching nantinya dan diisi dengan *script*, dan *button* keluar yang berfungsi untuk tombol keluar program dan diisi dengan *script*. Dapat dilihat dalam tabel 4.1 berikut :

Tabel 4.1 Script Tombol pada Form Awal

Tombol	Script
Masuk	<pre>procedure TForm1.Button1Click(Sender: TObject); begin   Form2.Show; end;</pre>
Keluar	<pre>procedure TForm1.BitBtn1Click(Sender: TObject); begin   end;</pre>

5. Tahap selanjutnya yaitu, pengaturan pada *object inspector*, hal ini dilakukan agar pada saat program running, ukuran tampilan akan sesuai dengan ukuran yang sebelumnya sudah di setting, yaitu dengan cara melakukan pengaturan pada *anchors* dan pada *bordesStyle* nya.

Pada Gambar 4.3 berikut adalah merupakan tampilan pada form awal yang sudah selesai dibuat sesuai dengan proses yang telah dijabarkan diatas :

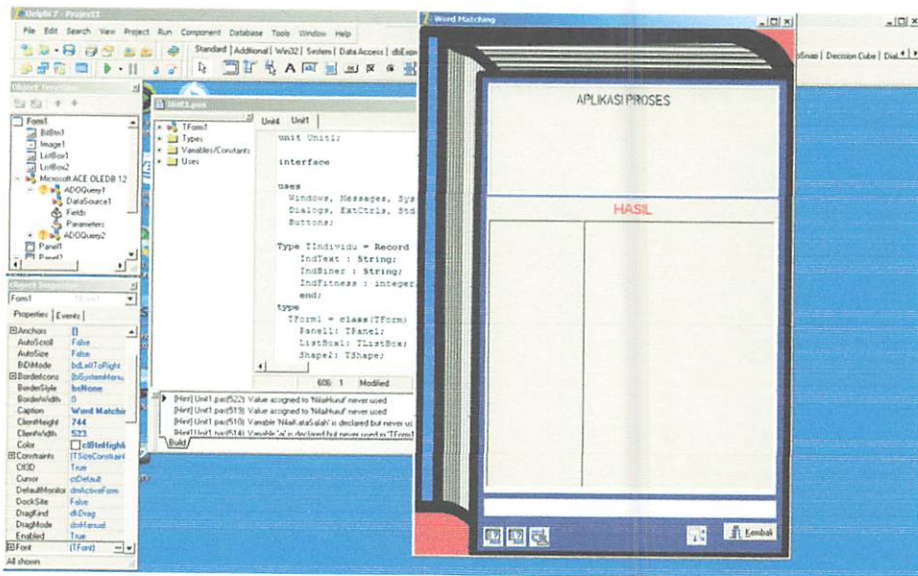


Gambar 4.3 Tampilan Form Awal

#### 4.2.2. Pembuatan Form Utama

Form utama adalah merupakan halaman dimana aplikasi Word Matching ini akan dijalankan, di halaman inilah nantinya akan dilakukan pengimplementasian dari metode Algoritma Genetika kedalam bentuk *code program* di *unit delphi7*. Namun sebelum melakukan tahapan tersebut, maka langkah pertama yang harus dilakukan terlebih dahulu yaitu pembuatan tampilan dari form utama, dan berikut adalah merupakan tahapan pengerjaannya :

1. Sama seperti pembuatan form awal yaitu membuat aplikasi baru dengan *form* sebagai media pembuatan desain dan *unit* untuk penulisan *code program*. Jika di dalam form awal disimpan dengan nama *form1* dan *unit1* maka dalam form utama disimpan dengan nama *form2* dan *unit2*, hal ini dilakukan agar nanti pada saat proses penggabungan project tidak akan muncul nama *form* dan *unit* yang sama.
2. Memasukkan image yang akan digunakan kedalam *form* yang sudah disediakan, langkah-langkah nya sama dengan proses pemasukan image pada form awal.
3. Langkah berikutnya yaitu memasukkan *component pallete* kedalam form sesuai dengan kebutuhannya, yaitu antara lain *label judul* dan *label hasil* yang berfungsi untuk menampilkan tulisan, *ListBox1* yang berfungsi untuk menampilkan hasil nilai individu yang nantinya akan dijalankan oleh program, *ListBox2* yang berfungsi untuk menampilkan hasil dari kata dan jumlah nilai *fitness* yang dihasilkan oleh program dan *progress bar* untuk menampilkan animasi sederhana dari proses aplikasi yang sedang berjalan saat mengeksekusi program. Gambar 4.4 berikut adalah merupakan tampilan dari form utama Aplikasi Word Matching sebelum panel-panel utama program dibuat :



Gambar 4.4 Tampilan Form Utama Sebelum Panel-Panel Utama Program Dibuat.

4. Tahapan berikutnya yaitu, membuat panel-panel utama program Aplikasi Word Matching, *component pallette* yang akan dimasukkan kedalam form yaitu antara lain *panel* yang terdiri dari *LabelInputKata*, *LabelKataSaran*, *LabelPopulasi*, *LabelIterasi*, *LabelJumlahPerulangan*, *EditInputKata* yang berfungsi untuk menginputkan kata, *LabelKataSaran* yang berfungsi untuk menginputkan kata saran yang diambil dari sistem databasenya, *EditPopulasi* yang berfungsi untuk menginputkan berapa jumlah menginputkan kata salah yang akan diinputkan, populasi yang akan dimasukkan, dalam program ini jumlah populasi yaitu kurang lebih 200 angka populasi, *EditIterasi* menginputkan berapa jumlah iterasi yang akan dimasukkan, iterasi yang dimaksudkan adalah proses jumlah perulangan dari individu, dalam program ini jumlah iterasinya yaitu kurang lebih 100, *EditJumlahPerulangan* yang berfungsi untuk menampilkan jumlah perulangan yang dilakukan oleh program dan *ButtonProses* yaitu tombol yang berfungsi untuk memulai eksekusi kata target yang di inputkan dan *ButtonDataKata* yang berfungsi sebagai button untuk masuk kedalam menu Daftar Kata.

Gambar 4.5 berikut adalah merupakan tampilan panel-panel utama program

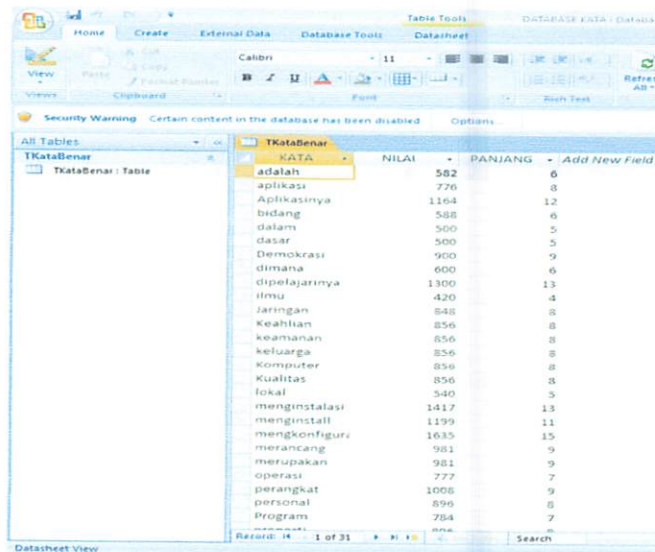


Gambar 4.5 Tampilan Panel-Panel Utama Program

#### 4.2.3. Pembuatan Sistem Data Base

Sistem data base merupakan tempat untuk penyimpanan kata yang ejaan katanya sudah disesuaikan dengan kamus bahasa Indonesia yang baku dan benar, didalam program ini fungsi kata yang ada di dalam database tersebut adalah sebagai kata saran yang akan diinputkan kedalam kolom input kata yang nantinya akan dieksekusi oleh program.

Gambar 4.6 berikut adalah merupakan tampilan sistem database yang ada di dalam Microsoft Access.



Gambar 4.6 Tampilan Sistem Data Base

#### 4.2.4. Implementasi Metode Algoritma Genetika ke Program

Implementasi metode Algoritma Genetika ke program adalah merupakan tahapan penerapan siklus metode Algoritma genetika kedalam program. disinilah fungsi metode tersebut digunakan di dalam aplikasi

Word Matching. Tahapan siklus dari metode Algoritma Genetika akan dituliskan kedalam *code program*. Penulisan akan dilakukan kedalam *unit* didalam *delphi7*. Berikut adalah merupakan langkah-langkah pengerjaannya :

1. Langkah pertama yaitu pendefinisian variabel utama program didalam *unit* yang nantinya akan digunakan sebagai penulisan *code program* pada tahapan proses. Tabel 4.2 berikut adalah merupakan devinisi variabel utama program yang digunakan :

*Tabel 4.2 Definisi Variabel Utama Program*

Script
Var Form2: TForm2; Indiv11,Indiv12,Indiv13,Indiv14,Indiv15 : array of TIndividu; Indiv21,Indiv22,Indiv23,Indiv24,Indiv25 : array of TIndividu; Indiv31,IndivTemp : array of TIndividu;  targetHuruf, AwalKata : String; Populasi, jmlIterasi : integer;

2. Tahapan Berikut adalah merupakan penulisan *private declaration*. yaitu pendefinisian dari tahapan penulisan *function* dalam aplikasi Word Matching. di dalam aplikasi Word Matching *Function* adalah *code program* utama yang digunakan untuk pengubahan bentuk desimal ke biner, pengubahan bentuk biner ke desimal, penghitungan nilai fitness dan penghitungan selisih dari nilai fitness. Di dalam *unit*, tahapan *code program function* dituliskan sebagai *DecToBinStr*, *BinStrToDec*, *keBiner*, *keString*, *HtgFitness*, *HtgSelisihFitness*. Tabel 4.3 berikut adalah merupakan penulisan devinisi *private declaration function* di dalam *unit* dalam aplikasi Word Matching :



Tabel 4.3 Definisi private declaration function

Script
<pre> Private   { Private declarations }   function DecToBinStr(bil: integer): string;   function BinStrToDec(binStr: string): integer;   function keBiner(hrfInput:String): String;   function keString(binernya:string):String;   function HtgFitnes(Huruf:String):integer;   function HtgSelisihFitnes(HurufBanding, HurufTarget:String):Integer;    procedure buatIndivBaru(populasi:integer;trgtHrf:String;var indivhsl:array of TIndividu);   procedure UrutIndividu(populasi:integer;var indivhsl:array of TIndividu); public   { Public declarations }   end; </pre>

3. Adalah merupakan penulisan *code program function* didalam *unit* dalam aplikasi Word Matching. Tabel 4.4 berikut adalah merupakan *code program function* yang pertama, yaitu *function DecToBinStr* yang digunakan untuk merubah bentuk desimal ke dalam bentuk biner :

*Tabel 4.4 Code Program Function DecToBinStr*

Script
<pre>function TForm1.DecToBinStr(bil: integer): string; var   str : string;   i : integer; begin   bil:= abs(bil);   for i:= 1 to SizeOf(bil) * 8 do   begin     if bil &lt; 0 then       str := str + '1'     else       str := str + '0';     bil := bil shl 1;   end;   result := RightStr(str,5);   end;</pre>

Tabel 4.5 Adalah merupakan *code program function* yang kedua, yaitu *code program function BinStrToDec* yang digunakan untuk merubah bentuk biner ke dalam bentuk desimal :

*Tabel 4.5 Code Program Function BinStrToDec*

Script
<pre> function TForm1.BinStrToDec(binStr: string): integer; var   i,j : integer;   bil : extended; begin   bil := 0;   j := 0;   for i := Length(binStr) downto 1 do   begin     bil := strtoint(MidStr(binStr,i,1))* power(2,j)+ bil;     j := j + 1;   end;   result := strtoint(currtostr(bil));   end; </pre>

Tabel 4.6 Adalah merupakan *code program function* yang ketiga, yaitu *code program function HtgFitnes* yang digunakan untuk menghitung jumlah dari nilai fitness :

*Tabel 4.6 Code Program Function HtgFitness*

Script
<pre> function TForm1.HtgFitnes(Huruf:String):integer; var   i,pjgHuruf : integer;   huruf1 : string[1];   nilaiHuruf1 : integer; begin   nilaiHuruf1 := 0;   pjgHuruf := length(Huruf);   For i := 1 to pjgHuruf do   begin     huruf1 := midstr(Huruf,i,1);     if (ord(huruf1[1])-96) &gt; 0 then     begin       //listbox2.Items.Add(inttostr(ord(huruf1[1])-96)); </pre>

```

    nilaiHuruf1 := nilaiHuruf1 + (ord(huruf1[1])-96)
end else
    nilaiHuruf1 := nilaiHuruf1 + 0;
end;
HtgFitnes := nilaiHuruf1;
end;

```

Tabel 4.7 Adalah merupakan *code program function* yang keempat, yaitu *code program function HtgSelisihFitnes* yang digunakan untuk menghitung selisih dari nilai fitness :

*Tabel 4.7 Code Program Function HtgSelisihFitness*

Script
<pre> function TForm1.HtgSelisihFitnes(HurufBanding, HurufTarget:String):Integer; var     i,pjgHuruf : integer;     huruf1,huruf2 : string[1];     nilaiHuruf1,nilaiHuruf2,hasil1, hasil2 : integer; begin     hasil2 := 0;     pjgHuruf := length(HurufTarget);     For i := 1 to pjgHuruf do     begin         huruf1 := midstr(HurufTarget,i,1);         huruf2 := midstr(HurufBanding,i,1);         if (ord(huruf1[1])-96) &gt; 0 then             nilaiHuruf1 := (ord(huruf1[1])-96)         else             nilaiHuruf1 := 0;          if (ord(huruf2[1])-96) &gt; 0 then             nilaiHuruf2 := (ord(huruf2[1])-96)         else             nilaiHuruf2 := 0;          hasil1 := abs(nilaiHuruf2 - nilaiHuruf1);         hasil2 := hasil2 + hasil1     end;     HtgSelisihFitnes := hasil2; end; </pre>

4. Pada tahapan yang selanjutnya adalah proses. Yaitu penerapan siklus Algoritma Genetika kedalam sebuah *code program* di dalam *unit*. *Code program* tersebut akan dituliskan di dalam *ButtonProses* yang sebelumnya sudah dibuat pada tahapan pembuatan form utama. Tabel 4.8 adalah merupakan *code program* proses yang pertama, yaitu *code program* proses untuk mengeksekusi kata target yang di inputkan dan menghitung nilai fitness nya :

*Tabel 4.8 Code Program Proses Pertama*

<b>Script</b>
<pre> procedure TForm1.Button1Click(Sender: TObject); var      i,j, pilihInduk,k : integer;     rand1,rand2,crossingbitawal, crossingbitakhir :integer;     induk1,induk2,crossText1, crossText2 : string;     jmlInduk : integer; begin     {Memasukkan target dan menghitung fitnes target}     targetHuruf := Edit1.Text;     Populasi := StrToInt(Edit2.Text);     jmlIterasi := StrToInt(Edit3.Text);      setlength(Indiv11,Populasi);     setlength(Indiv12,Populasi);     setlength(Indiv13,Populasi);     setlength(Indiv14,Populasi);      setlength(Indiv21,Populasi);      setlength(Indiv31,Populasi);     setlength(IndivTemp,Populasi);      ProgressBar1.Max := jmliterasi;     ProgressBar1.Refresh;     listBox1.Clear;     listBox2.Clear;     For i := 0 to populasi-1 do     begin         Indiv11[i].IndText :=";         Indiv11[i].IndBiner := ";         Indiv11[i].IndFitness := 1000000;          Indiv21[i].IndText :=";         Indiv21[i].IndBiner := ";         Indiv21[i].IndFitness := 1000000;     end; </pre>

Tabel 4.9 Adalah merupakan *code program* proses yang kedua, yaitu *code program* proses untuk membuat individu baru secara acak / random, menghitung nilai fitness dari individu baru tersebut dan mengurutkannya :

*Tabel 4.9 Code Program Proses Kedua*

<b>Script</b>
<pre> For k:=1 to jmlIterasi do begin   ProgressBar1.StepIt;   {Membuat random individu baru,menghitung fitnessnya dan mengurutkan}   For i := 0 to populasi-1 do   begin     Individ11[i].IndText := "";     Individ11[i].IndBiner := "";     Individ11[i].IndFitness := 1000000;      Individ12[i].IndText := "";     Individ12[i].IndBiner := "";     Individ12[i].IndFitness := 1000000;      Individ13[i].IndText := "";     Individ13[i].IndBiner := "";     Individ13[i].IndFitness := 1000000;      Individ14[i].IndText := "";     Individ14[i].IndBiner := "";     Individ14[i].IndFitness := 1000000;    end;    buatIndivBaru(Populasi,targetHuruf,indiv11);   buatIndivBaru(Populasi,targetHuruf,indiv12);   buatIndivBaru(Populasi,targetHuruf,indiv13);   buatIndivBaru(Populasi,targetHuruf,indiv14);    UrutIndividu(populasi, indiv12);   UrutIndividu(populasi, indiv13);   UrutIndividu(populasi, indiv14);    For i:=0 to round((populasi-1)/4) do   begin     Individ11[i].IndText := indiv12[i].IndText;     Individ11[i].IndBiner := Individ12[i].IndBiner;     Individ11[i].IndFitness := Individ12[i].IndFitness;      Individ11[i+round((populasi-1)/4)].IndText := indiv13[i].IndText;     Individ11[i+round((populasi-1)/4)].IndBiner := Individ13[i].IndBiner; </pre>

```

Indiv11[i+round((populasi-1)/4)].IndFitness := Indiv13[i].IndFitness;

Indiv11[i+round((populasi-1)/4*2)].IndText := indiv14[i].IndText;
Indiv11[i+round((populasi-1)/4*2)].IndBiner := Indiv14[i].IndBiner;
Indiv11[i+round((populasi-1)/4*2)].IndFitness :=
Indiv14[i].IndFitness;
end;

UrutIndividu(populasi, indiv11);

UrutIndividu(populasi, indiv21);

pilihInduk := round(populasi * 0.2);

for i:=0 to pilihInduk do
begin
  indiv31[100+i] := indiv11[i];
  indiv31[pilihInduk+101+i] := indiv21[i];
end;

{
  for i:= 0 to populasi-2 do
  begin
    for j:= i+1 to populasi-1 do
    begin
      if indiv31[i].IndFitness = indiv31[j].IndFitness then
        indiv31[j].IndFitness := 2000;
      end;
    end;
  }
  UrutIndividu(populasi, indiv31);

  edit4.text := inttostr(k);
  if indiv31[0].IndFitness = 0 then
  begin
    break;
  end;

```

Tabel 4.10 Adalah merupakan *code program* proses yang ketiga, yaitu *code program* proses untuk memilih induk baru dari nilai fitness yang terbaik atau proses ini disebut dengan elitism :

Tabel 4.10 Code Program Proses Ketiga

Script
<pre> {memilih induk baru dari fitness terbaik/elitism} For j:=0 to pilihinduk do begin   indiv21[j] := indiv31[j];   //indiv21[j+pilihinduk] := indiv21[j]; end; </pre>

Tabel 4.11 Adalah merupakan *code program* proses yang keempat, yaitu *code program* proses untuk melakukan perpindahan silang dari induk baru yang sudah di dapat atau proses ini disebut dengan crossover :

Tabel 4.11 Code Program Proses Keempat

Script
<pre> i:=pilihinduk;  {crossover dari induk baru} //if indiv21[0].IndFitness &gt; 3 then //begin  while i &lt;= populasi-2 do begin   randomize;   rand1 := random(pilihinduk);   rand2 := random(pilihInduk);    induk1 := indiv21[rand1].IndBiner;   induk2 := indiv21[rand2].IndBiner;    crossingbitawal := random(length(induk1))+1;   crossingbitakhir := random(length(induk1)-crossingbitawal)+1;   crossText1 := midstr(induk1,crossingbitawal,crossingbitakhir);   crossText2 := midstr(induk2,crossingbitawal,crossingbitakhir);    delete(induk1,crossingbitawal,crossingbitakhir);   delete(induk2,crossingbitawal,crossingbitakhir);   insert(crossText2,induk1,crossingbitawal);   insert(crossText1,induk2,crossingbitawal);    indiv21[i].IndText := keString(Induk1);   indiv21[i+1].IndText := keString(induk2);   indiv21[i].IndBiner := induk1;   indiv21[i+1].IndBiner := induk2;   indiv21[i].IndFitness := </pre>



```

HtgSelisihFitnes(indiv21[i].IndText,targethuruf);
    indiv21[i+1].IndFitness :=
HtgSelisihFitnes(indiv21[i+1].IndText,targethuruf);

    i:=i+2;
    end;

```

Tabel 4.12 Adalah merupakan *code program* proses yang kelima, yaitu *code program* proses untuk melakukan pertukaran gen dari hasil atau proses ini disebut dengan mutasi gen :

Tabel 4.12 Code Program Proses Kelima

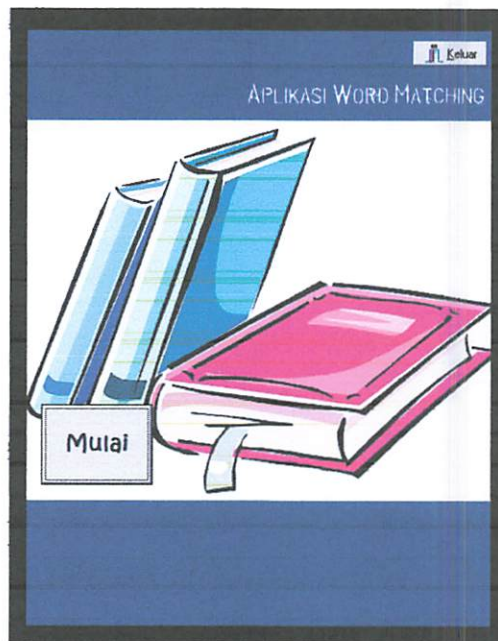
Script
<pre> {Mutasi dari hasil} // For i:=pilihinduk to populasi-1 do For i:=0 to populasi-1 do begin     randomize;     induk1 := indiv21[i].IndBiner;     crossingbitawal := random(length(induk1)+1);     crossText1 := midstr(induk1,crossingbitawal,1);      if crosstext1 = '0' then         crosstext2 := '1'     else         crosstext2 := '0';      insert(crossText2,induk1,crossingbitawal);     delete(induk1,crossingbitawal+1,1);      indiv21[i].IndText := keString(induk1);     indiv21[i].IndBiner := (Induk1);     indiv21[i].IndFitness := HtgSelisihFitnes(indiv21[i].IndText,targethuruf);     end;     UrutIndividu(populasi, indiv21); end; //UrutIndividu(populasi, indiv21); ListBox1.clear; For i:= 0 to Populasi-1 do begin     //ListBox1.Items.add(Indiv21[i].IndText + ' --FITNESS-&gt; ' +     inttostr(Indiv21[i].IndFitness)+' --INDIV-&gt; '+inttostr(i));     //ListBox2.Items.add(Indiv31[i].IndText + '/' +     inttostr(Indiv31[i].IndFitness)+'/'+inttostr(i)); </pre>

```
ListBox2.Items.add(Indiv21[i].IndText + ' --FITNESS->' +  
inttostr(Indiv21[i].IndFitness));  
ListBox1.Items.add('INDIVIDU KE--> '+inttostr(i+1));  
end;
```

### 4.3 Pengujian Aplikasi

#### 4.3.1. Pengujian Halaman Form Awal

Pada saat pertama kali aplikasi Word Matching dijalankan, halaman yang pertama kali ditampilkan adalah form awal, pada halaman form awal akan muncul teks Aplikasi Word Matching dan terdapat dua buah tombol yaitu *Tombol Masuk* yang digunakan untuk memulai aplikasi dan *Tombol Keluar* yang digunakan untuk mengakhiri aplikasi. Pengujian halaman form awal dapat dilihat pada gambar 4.7 :

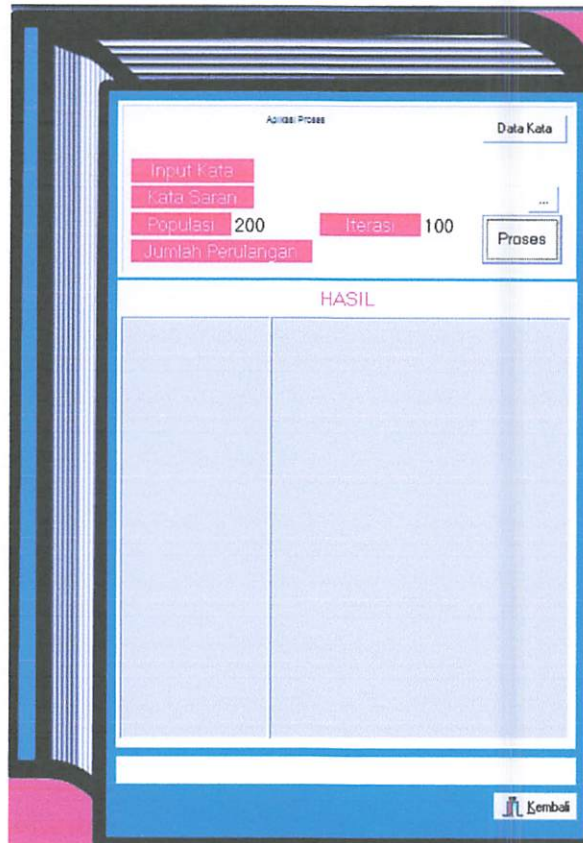


Gambar 4.7 Hasil Pengujian Form Awal

#### 4.3.2. Pengujian Halaman Form Utama

Pada halaman form utama aplikasi Word Matching terdapat teks Aplikasi Proses, *Label Input Kata* dan kolom input kata untuk mengisi kata yang akan dieksekusi, *Label Kata Saran* dan kolom kata saran untuk mengisikan kata saran yang diambil dari sistem database nya, *Label Populasi*

dan kolom populasi untuk menginputkan jumlah populasi, *Label Iterasi* dan kolom iterasi untuk menginputkan jumlah iterasi, *Label Perulangan* dan kolom perulangan untuk menampilkan jumlah perulangan yang dilakukan oleh program, *Button Proses* untuk memulai eksekusi program, terdapat teks Hasil dan *List Box Individu* untuk menampilkan jumlah nilai individu yang sudah dieksekusi oleh program, *List Box Hasil* untuk menampilkan hasil kata dan nilai fitness yang dieksekusi oleh program, progress bar dan *button Kembali* untuk kembali ke halaman form awal, button Data Kata untuk masuk kedalam menu data kata. Pengujian halaman form utama dapat dilihat pada gambar 4.8 :



*Gambar 4.8 Hasil Pengujian Halaman Form Utama*

### **4.3.3. Pengujian Sistem Aplikasi**

Setelah dilakukan pengujian pada halaman aplikasi, proses selanjutnya yaitu dilakukan pengujian pada eksekusi programnya. Eksekusi program dilakukan dengan tahapan :

1. Menginputkan kata saran ke dalam kolom kata saran yang di ambil dari sistem database nya
2. Menginputkan susunan huruf dari kata saran yang akan di eksekusi / ditampilkan kembali kedalam kolom input kata
3. Menginputkan jumlah populasi ke dalam kolom populasi
4. Menginputkan jumlah iterasi ke dalam kolom iterasi
5. Klik tombol proses untuk mulai eksekusi program.

Dari beberapa kali pengujian sistem aplikasi yang dilakukan secara langsung dapat disimpulkan sebagai berikut:

*Tabel 4.13 Hasil Pengujian Sistem Aplikasi*

Uji Coba Ke-	Input Kata		Populasi	Iterasi	Hasil	Total Nilai Fitness
	Input Kata	Kata Saran				
1.	Teknik	Teknik	200	100	Teknik	0
2.	Sistem	Sistem	200	100	Sistem	0
3.	Revisi	Revisi	200	100	Revisi	0
4.	Kualitas	Kualitas	200	100	Kualitas	0
5.	Probabilitas	Probabilitas	200	100	Probabilitat	2
6.	Operasi	Operasi	200	100	Operasi	0
7.	Variabel	Variabel	200	100	Variabel	0
8.	Birokrasi	Birokrasi	200	100	Birokrash	1
9.	Ejaan	Ejaan	200	100	Ejaan	0
10.	Properti	Properti	200	100	Properti	0

Dari hasil pengujian sistem dapat disimpulkan bahwa dari 10 kata yang diinputkan secara acak 8 diantaranya benar dan dapat diketahui hasilnya dengan

perbandingan dari kata saran yang terdapat di dalam sistem database nya dan memperoleh nilai fitness yaitu 0, sedangkan 2 diantaranya masih belum dapat diketahui kebenarannya yaitu dengan memperoleh nilai fitness masing- masing 2 dan 1.

#### 4.3.4. Pengujian User

Pengujian *user* dilakukan kepada *user* untuk mengetahui seberapa besar program ini dapat dipahami oleh *user* dengan memeberikan beberapa pertanyaan berupa kuesioner yang terkait dengan penilaian program aplikasi.

*Tabel 4.14 Hasil Pengujian Sistem Aplikasi*

No.	Pertanyaan	Jawaban	
		Ya	Tidak
1.	Apakah aplikasi ini mudah untuk digunakan ?	80%	20%
2.	Apakah desain aplikasi ini menarik ?	95%	5%
3.	Apakah aplikasi ini cukup membantu anda ?	75%	25%
4.	Apakah menu – menu pada aplikasi cukup lengkap?	80%	20%
5.	Apakah menu - menu pada aplikasi cukup jelas ?	100%	0%
6.	Apakah aplikasi ini berfungsi secara keseluruhan ?	90%	10%
7.	Apakah aplikasi ini sudah cukup baik untuk menapilkan kembali kata sesuai dengan metode AG?	75%	25%

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Kesimpulan yang diperoleh dalam pembuatan Aplikasi Word Matching Bahasa Indonesia meliputi:

1. Dari hasil keseluruhan pengujian aplikasi Word Matching dari halaman form awal dan halaman form utama berjalan cukup baik.
2. Dari hasil pengujian sistem dapat disimpulkan bahwa dari 10 kata yang diinputkan secara acak 8 diantaranya benar dan dapat diketahui hasilnya dengan perbandingan dari kata saran yang terdapat di dalam sistem database nya dengan memasukkan jumlah yang sama yaitu populasi 200 dan jumlah iterasi 100 dengan hasil yang memperoleh nilai fitness yaitu 0, sedangkan 2 diantaranya masih belum dapat diketahui kebenarannya yaitu dengan memperoleh nilai fitness masing-masing 2 dan 1. Hasil pengujian kepada *user*, menurut 25% *user* aplikasi ini belum cukup baik untuk menampilkan kembali kata sesuai dengan metode Algoritma Genetika. Namun menurut 100% *user* menu-menu pada aplikasi ini cukup jelas.

#### **5.2 Saran**

Aplikasi Word Matching masih memiliki banyak keterbatasan yang nantinya diharapkan dapat dikembangkan. Berikut ini adalah saran yang diberikan untuk pengembangan aplikasi selanjutnya, antara lain:




1. Aplikasi dapat dikembangkan lagi untuk menampilkan kembali kata yang terdiri dari lebih dari 10 huruf atau bahkan satu kalimat.
2. Untuk pengembangan aplikasi dapat ditambahkan halaman form untuk petunjuk penggunaan aplikasi.

## DAFTAR PUSTAKA

- [1] Basuki, Ahmad. 2003, *Sistem penjadwalan Perkuliahan Menggunakan Algoritma Genetika*. Skripsi: Semarang: Jurusan Teknik Informatika UNDIP.
- [2] Jamaludin Malik, Jaja. (2006). *Kumpulan Latihan Pemrograman Delphi*. Cirebon: Andi.
- [3] Kadir, Abdul. (2010). *Mudah\_Mempelajari Database ACCESS*. Yogyakarta: Andi.
- [4] Kusnassriyanto (2011). *Belajar Pemrograman Delphi*. Bandung:Modula.
- [5] Nurul Rahayu, Dewi. 2004, *Artificial Intelegent*, Bandung: Jurusan Teknik Informatika UNIKOM.
- [6] Suyanto.ST.Msc (2009). *Artifical Intelegent*. Bandung; Informatika.
- [7] T.Sutojo, S.Si.,M.kom, Edy Mulyanto,S.Si., M.kom dan Dr.Vincent Suhartono (2011). *Kecerdasan Buatan*. Semarang: Andi.
- [8] Wahyu,G. Dan Andi,K. (2011). *Otomatisasi Sistem Penjadwalan Dengan Dekomposisi Attribute Basis Data Menggunakan Algoritma Genetika*, Mataram: SNTEI Fakultas Teknik UNRAM.
- [9] <http://elib.unikom.ac.id/download.php?id=143046> , diakses 23 April 2012 pukul 13.00 wib.
- [10] <http://mnrizal.files.wordpress.com/2008/09/bab-2.pdf> , diakses 19 November 2012 pukul 20.00 wib.
- [11] <http://repository.usu.ac.id/bitstream/123456789/21086/Chapter%20II.pdf> , 23 April 2012 pukul 12.35 wib.
- [12] <http://sanjoyo55.files.wordpress.com/2008/11/non-linier-gen-algol.pdf> , diakses 20 April 2012 pukul 15.00 wib.
- [13] <http://www2.ukdw.ac.id/kuliahinfoTI2023Modul07A.pdf> , diakses 19 November 2012 pukul 20.15 wib.

## FORMULIR BIMBINGAN SKRIPSI

Nama : Rohmad Hanny Afandy  
NIM : 0818027  
Masa Bimbingan : 29 Oktober 2012 s/d 29 April 2013  
Judul Skripsi : Pengembangan Aplikasi Word Matching Bahasa Indonesia  
Menggunakan Metode Algoritma Genetika

NO	TANGGAL	URAIAN	PARAF
1	22/11/2012	Demo, konsultasi Bab I & II	
2	28/11/2012	Konsultasi Bab I, II, dan III Persiapan Makalah Seminar Hasil	
3	20/12/2012	Makalah Seminar Hasil Konsultasi Skripsi Bab I sampai IV	
4	23/1/2013	Makalah Seminar Hasil	

Malang, 12 Februari 2013

Dosen Pembimbing I


  
**Joseph Dedy Irawan, ST, MT**

NIP. ~~19~~197404162005011002



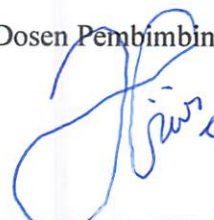
## FORMULIR BIMBINGAN SKRIPSI

Nama : Rohmad Hanny Afandy  
NIM : 0818027  
Masa Bimbingan : 29 Oktober 2012 s/d 29 April 2013  
Judul Skripsi : Pengembangan Aplikasi Word Matching Bahasa Indonesia  
Menggunakan Metode Algoritma Genetika

NO	TANGGAL	URAIAN	PARAF
1	22/11/2012	Demo, konsultasi Bab I & II	
2	28/11/2012	Konsultasi Bab I, II, dan III Persiapan Makalah Seminar Hasil	
3	20/12/2012	Makalah Seminar Hasil Konsultasi Skripsi Bab I sampai IV	
4	23/1/2013	Makalah Seminar Hasil	

Malang, 12 Februari 2013

Dosen Pembimbing II



**Ali Mahmudi, BEng, PhD**

NIP. P. 1031000429



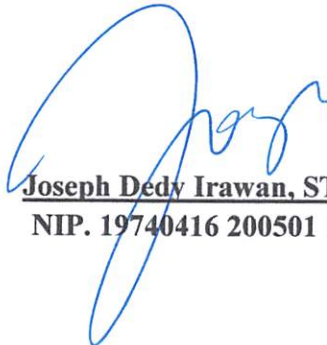
INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
PROGRAM STUDI TEKNIK INFORMATIKA S-1  
Jl. Karanglo, KM 2 Malang

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : ROHMAD HANNY AFANDY  
NIM : 0818027  
JURUSAN : Teknik Informatika S-1  
JUDUL : **PENGEMBANGAN APLIKASI WORD MATCHING  
BAHASA INDONESIA MENGGUNAKAN METODE  
ALGORITMA GENETIKA**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :  
Hari : Rabu  
Tanggal : 20 Februari 2013  
Nilai : A

Panitia Ujian Skripsi :  
**Ketua Majelis Penguji**

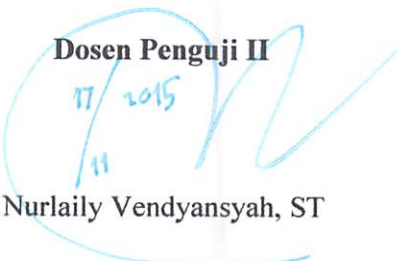
  
**Joseph Dedy Irawan, ST, MT**  
NIP. 19740416 200501 1 002

Anggota Penguji:

**Dosen Pnguji I**

  
Michael Ardita ST, MT  
NIP. P 1031000434

**Dosen Penguji II**

  
17/2015  
11  
Nurlaily Vendyansyah, ST



**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**PROGRAM STUDI TEKNIK INFORMATIKA S-1**  
**Jl. Karanglo, KM 2 Malang**


**FORMULIR PERBAIKAN SKRIPSI**

Dalam pelaksanaan ujian skripsi jenjang strata 1 program studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

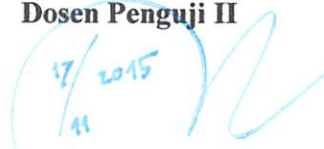
NAMA : ROHMAD HANNY AFANDY  
NIM : 0818027  
JURUSAN : Teknik Informatika S-1  
JUDUL : **PENGEMBANGAN APLIKASI WORD MATCHING  
BAHASA INDONESIA MENGGUNAKAN METODE  
ALGORITMA GENETIKA**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	20 Februari 2013	Perbaikan Program Perbaikan Laporan Bab I, IV,V, dan Abstrak	<i>Ar</i>
2.	Penguji II	20 Februari 2013	Perbaikan Program Perbaikan Bab I, III dan Daftar Pustaka	

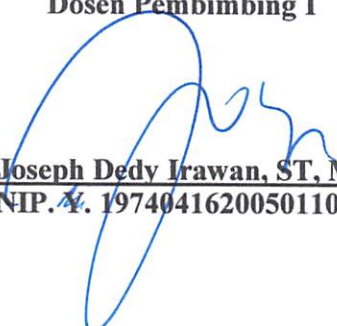
**Dosen Pnguji I**

  
**Michael Ardita, ST, MT**  
NIP. P 1031000434

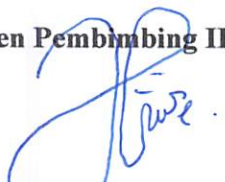
**Dosen Penguji II**

  
**Nurlaily Vendyansyah, ST**

**Dosen Pembimbing I**

  
**Joseph Dedy Irawan, ST, MT**  
NIP. P. 197404162005011002

**Dosen Pembimbing II**

  
**Ali Mahmudi, BEng, PhD**  
NIP. P. 1031000429