

**PROGRAM BANTU PENGENALAN MACAM MACAM  
CACING SERTA PENGARUHNYA TERHADAP  
KESEHATAN**

**Skripsi**



**Disusun Oleh :**

**Benediktus Budi Tasanto**

**NIM : 975314013**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2005**

**PROGRAM BANTU PENGENALAN MACAM MACAM  
CACING SERTA PENGARUHNYA TERHADAP  
KESEHATAN**

**Skripsi**



**Disusun Oleh :**

**Benediktus Budi Tasanto**

**NIM : 975314013**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2005**

**PROGRAM BANTU PENGENALAN MACAM MACAM  
CACING SERTA PENGARUHNYA TERHADAP  
KESEHATAN**

Skripsi

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
Jurusan Teknik Informatika



Disusun Oleh :

Benediktus Budi Trasanto

NIM : 975314013

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
2005**

**AID PROGRAM TO INTRODUCE WORM CLASSIFICATION  
AND ITS EFFECTS TO HUMAN HEALT**

**A Thesis**

**Presented as Partial Fulfilment of the Requirements  
to Obtain the Sarjana Teknik Degree  
in Informatics Engineering Department**



Beneditus Budi Trasanto

Student Number : 975314013

**INFORMATICS ENGINEERING DEPARTMENT  
FACULTY OF ENGINEERING  
SANATA DHARMA UNIVERSITY  
YOGYAKARTA**

2005

**SKRIPSI**

**PROGRAM BANTU PENGENALAN MACAM MACAM  
CACING SERTA PENGARUHNYA TERHADAP  
KESEHATAN**

Disusun Oleh :

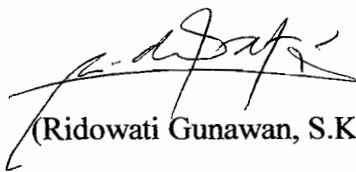
**Benediktus Budi Trasanto**

**NIM : 975314013**

**NIRM : 970051123113120013**

Telah disetujui oleh :

Dosen Pembimbing I



(Ridowati Gunawan, S.Kom., M.T.)

tanggal 19 Agustus 2005

**SKRIPSI**

**PROGRAM BANTU PENGENALAN MACAM MACAM  
CACING SERTA PENGARUHNYA TERHADAP  
KESEHATAN**

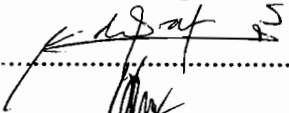
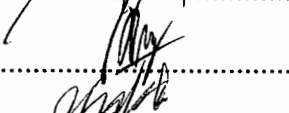
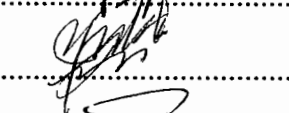

Dipersiapkan dan ditulis oleh

Benediktus Budi Trasanto

NIM : 975314013

Telah dipertahankan di depan Panitia Penguji  
pada tanggal 19 Agustus 2005  
dan dinyatakan memenuhi syarat

Susunan Panitia Penguji

	Nama Lengkap	Tanda tangan
Ketua	Ridowati Gunawan, S.Kom., M.T.	
Sekretaris	H. Agung Hernawan, S.T.	
Anggota	St. Yudianto Asmoro, S.T.	
Anggota	Alb. Agung Hadiatma, S.T., M.T.	

Yogyakarta, Agustus 2005

Fakultas Teknik

Universitas Sanata Dharma

Dekan,



(Ir. Greg. Heliarko, S.J., S.S., B.S.T., M.A., M.Sc.)

## **PERNYATAAN KEASLIAN KARYA**

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah disebutkan dalam kutipan dan daftar pustaka, sebagai layaknya karya ilmiah.

Yogyakarta, Agustus 2005

Penulis

## **HALAMAN PERSEMBAHAN**

### **Karya ini aku persembahkan kepada:**

Tuhan Yesus Kristus dan Bunda Maria yang senantiasa memberikan berkat dan selalu menjagaku setiap saat.

Kedua orang tua yang selalu mendoakan aku, sabar dan sayang serta mendorong untuk semangat terus.

Simbahku tercinta yang selalu merawatku sampai saat ini

Adik Edi & Wahyu yang selalu mendoakan dan memberi semangat selama ini.



## HALAMAN MOTTO

*Hidup bukan milik kita tetapi kita dapat mengisi hidup karena hidup kita milik*

*Dia dan kita tidak tahu kapan Dia akan datang*

*Orang bijak adalah orang yang mampu mengukur kelebihan dan  
kekurangannya sendiri*

*Kasih bukan segala galanya tetapi segala galanya tanpa kasih adalah hampa*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yesus Kristus yang maha pengasih dan pengampun atas segala rahmat dan karunia yang telah dicurahkan kepada penulis, sehingga dapat menyelesaikan tugas akhir ini dengan sebaik baiknya.

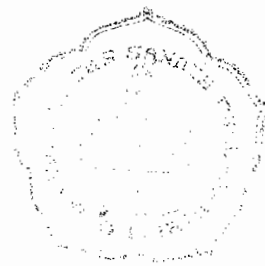
Penulis menyadari sepenuhnya mampu menyelesaikan tugas akhir ini tidak lain karena dukungan, bantuan dan bimbingan dari berbagai pihak. Oleh sebab itu pada kesempatan ini perkenankanlah penulis mengucapkan terima kasih kepada:

1. Ibu A.M Polina S.Kom, MSc. sebagai Ketua Jurusan Teknik Informatika yang selalu memberi kesempatan yang terbaik untuk kami.
2. Ibu Ridowati Gunawan S.Kom, M.T. sebagai dosen pembimbing I yang dengan sabar telah banyak memberi tuntunan, masukan, arahan selama ini.
3. Seluruh dosen TI yang telah memberikan pengetahuannya selama ini
4. Pak Bele, Mas Diar ( trima kasih semuanya selama pelatihan )Mas Danang, Mbak Sri (Lab Komputer), Mbak Marni, Mas Tri(Sekretariat Fakultas Teknik) atas bantuan serta keramahannya.
5. Kedua orang tuaku yang tercinta Al. Ngatimin & P Suyatmi yang selalu menyayangiku, mendoakan aku serta penyemangatku.
6. Simbahku tercinta yang telah mengasuhku selama ini.
7. Adikku Edi & Wahyu yang selalu memberi doa dan dorongan selama ini “ayo ndang lulus ?”

8. Teman teman TI Angkatan 97 semua khususnya Franz, Aan, Fajar, Widi, Wiwik (Nostro Signore), Mr WaOne, Kuntadi, Rosna, Oncu, Alva, Acun, Fery, Arti, Vido, Wondo, kalianlah sebagai inispirasi dan rekan seperjuangan selama ini.
9. Teman Mudika St. Timotius sebagai tempat untuk bertukar pikiran dan dorongannya.

Agustus 2005

Penulis



## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN JUDUL BAHASA INDONESIA.....	ii
HALAMAN JUDUL BAHASA INGGRIS.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
PERNYATAAN KEASLIAN KARYA.....	vi
HALAMAN PERSEMBAHAN.....	vii
HALAMAN MOTTO.....	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xv
INTISARI.....	xvi
ABSTRACTION.....	xvii
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penulisan Tugas Akhir.....	3
1.5. Metodologi Penulisan.....	3
1.6. Sistematika Penulisan.....	4

## BAB II LANDASAN TEORI

2.1. Definisi Program Bantu.....	5
2.2. Antar Muka Manusia-Komputer.....	6
2.3. Basis Data.....	6
2.4. Sistem Informasi.....	7
2.4.1. Pengertian Sistem Informasi.....	7
2.4.2. Siklus Hidup Pengembangan Sistem.....	8
2.5. Cacing.....	10
2.5.1. Jenis Cacing Gilig.....	10
2.5.2. Jenis Cacing Pipih.....	11
2.5.3. Jenis Cacing Bersegmen.....	11

## BAB III ANALISIS DAN PERANCANGAN

3.1. Analisis Sistem.....	13
3.1.1. Diagram Konteks.....	13
3.1.2. Diagram Berjenjang.....	14
3.1.3. Diagram Arus Data Level 0.....	15
3.1.3. ER Diagram.....	16
3.2. Perancangan Basis Data.....	17
3.2.1. <i>Form</i> Utama.....	20
3.2.2. <i>Form</i> Status.....	21
3.2.3. <i>Form</i> Petugas.....	22
3.2.4. <i>Form</i> Pengguna.....	28

3.2.5. Arsitektur Hardware Software.....	29
<b>BAB IV IMPLEMENTASI</b>	
4.1. Hasil Program.....	31
4.1.1. Menu Selamat datang.....	31
4.1.2. Menu Pilihan Status.....	32
4.1.3. Menu Registrasi Pasword.....	32
4.2. Menu Input Program.....	33
4.2.1. Menu Input Cacing.....	33
4.2.2. Menu Input Penyakit.....	35
4.2.3. Menu Input Obat.....	36
4.2.4. Menu Input Kategori.....	37
4.3. Menu Edit.....	37
4.3.1. Menu Edit Cacing.....	38
4.3.2. Menu Edit Obat.....	39
4.3.3. Menu Edit Penyakit.....	40
4.3.4. Menu Edit Kategori.....	41
4.4. Menu Output Pengguna.....	42
<b>BAB V ANALISA HASIL</b>	
5.1. Analisa Perangkat Lunak.....	44
5.2. Analisa Hasil Program.....	44

## BAB VI KESIMPULAN DAN SARAN

6.1. Kesimpulan.....	46
6.2. Saran.....	46

## DAFTAR PUSTAKA

## LAMPIRAN

## DAFTAR TABEL

Tabel 3.1. Tabel Pasword.....	16
Tabel 3.2. Tabel Cacing.....	16
Tabel 3.3. Tabel Penyakit.....	17
Tabel 3.4. Tabel Obat.....	17
Tabel 3.5. Tabel Kategori.....	17



## DAFTAR GAMBAR

Gambar 3.1. Diagram Konteks.....	13
Gambar 3.2. Diagram Berjenjang.....	14
Gambar 3.3. Diagram Arus Data.....	15
Gambar 3.4. ER Diagram.....	16
Gambar 3.5. Desain Basis Data.....	17
Gambar 3.6. <i>Form</i> Utama.....	20
Gambar 3.7. <i>Form</i> Status.....	21
Gambar 3.8. <i>Form</i> Registrasi password.....	22
Gambar 3.9. <i>Form</i> Menu Petugas.....	22
Gambar 3.10. <i>Form</i> Input Cacing.....	23
Gambar 3.11. <i>Form</i> Input Penyakit.....	23
Gambar 3.12. <i>Form</i> Input Kategori.....	24
Gambar 3.13. <i>Form</i> Input Obat.....	24
Gambar 3.14. <i>Form</i> Edit Cacing.....	25
Gambar 3.15. <i>Form</i> Edit Kategori.....	25
Gambar 3.16. <i>Form</i> Edit Obat.....	26
Gambar 3.17. <i>Form</i> Edit Penyakit.....	26
Gambar 3.18. <i>Form</i> Edit Petugas.....	27
Gambar 3.19 <i>Form Help</i> .....	27
Gambar 3.20. <i>Form</i> Informasi Pengguna.....	28
Gambar 3.21. <i>Form</i> Abaut.....	28

Gambar 4.1. Menu Selamat Datang.....	31
Gambar 4.2. Menu Pilihan Status.....	31
Gambar 4.3. Menu Registrasi Pasword.....	32
Gambar 4.4. Menu Input Cacing.....	33
Gambar 4.5. Menu Input Penyakit.....	35
Gambar 4.6. Menu Input Obat.....	36
Gambar 4.7. Menu Input Kategor.i.....	37
Gambar 4.8. Menu Edit Cacing.....	38
Gambar 4.9. Menu Edit Obat.....	39
Gambar 4.10. Menu Edit Penyakit.....	41
Gambar 4.11. Menu Edit Kategori.....	41
Gambar 4.12. Menu Output Pengguna.....	42

## **INTISARI**

Program bantu pengenalam macam-macam cacing serta pengaruhnya terhadap kesehatan bertujuan memberikan informasi kepada user mengenai macam-macam jenis cacing, penyakit yang ditimbulkan serta bagaimana obat yang sesuai untuk membunuh cacing tersebut. Di dalam program bantu ini juga disertai dengan gambar cacing dan siklus hidupnya. Dalam siklus hidup cacing dilengkapi dengan media suara. Program bantu ini dibuat menggunakan bahasa pemrograman Delphi 7 dan databasenya menggunakan Microsoft Access 2000.

## **ABSTRACT**

This programme attend to give an information to user about worm classification, the disease effect0ed by worm and how to find the best medicine to kill those worm. This programme also including worm pictures and it's life cycles. It's life cycles include with voice media. Programme made by Delphi 7 software and the database with Microsoft Acces 2000.

# **BAB I.**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Seseorang dapat mengetahui tanda - tanda awal menderita suatu penyakit akibat cacing sering tidak dirasakan. Hewan ini dapat hidup dan berkembang biak di dalam tubuh manusia maupun hewan. Didalam tubuh inangnya cacing bersifat merugikan bagi kesehatan tubuh, sebab hidupnya menghisap sari sari makanan dari tempatnya berkembang biak yang disebut inang. Dalam berkembang biak cacing dapat menghasilkan ratusan bahkan ribuan buah telur. Cacing termasuk salah satu hewan yang kecil dan berbahaya, akan tetapi sering tidak dianggap oleh masyarakat. Pada hal menurut data yang ada lebih dari 50 % jumlah penduduk pernah menderita penyakit akibat jenis cacing tertentu. Dari sekian jenis cacing yang ada terdapat juga jenis cacing yang mengakibatkan seorang penderita tidak tertolong apabila terlambat menanganinya.

Dengan program bantu pengenalan macam macam jenis cacing serta pengaruhnya terhadap kesehatan diharapkan dapat menambah wawasan serta memberi sedikit sumbangan dalam membantu masyarakat dalam memutus siklus hidup cacing. Mengenal jenis cacing lebih dekat dan mengetahui bahayanya serta bagaimana cara mencegah suatu penyakit yang disebabkan oleh cacing diharapkan akan dapat mengurangi resiko terserang penyakit akibat cacing.

Penggunaan teknologi informasi yang tepat tentunya sangat membantu dalam menyelesaikan suatu masalah yang membutuhkan kecepatan akses data dan informasi maupun keakuratan hasilnya. Didalam bidang kesehatan, penggunaan

teknologi informasi sangat besar manfaatnya, mengingat di dalam bidang kesehatan sering mengenal suatu istilah mencegah lebih bijaksana daripada mengobati.

Siklus hidup cacing sangat dekat dengan lingkungan kehidupan manusia akan tetapi kita sering tidak mengerti secara detail. Untuk itulah program bantu pengenalan macam macam cacing serta pengaruhnya terhadap kesehatan dapat bermanfaat.

### **1.2. Rumusan Masalah**

Dari hasil pengamatan, maka dapat dibuat suatu rumusan masalah sebagai berikut:

Bagaimana membuat analisis, desain dan mengimplementasikan ke dalam program bantu pengenalan macam-macam cacing, penyakit yang ditimbulkannya, jenis obat yang cocok serta bagaimana solusi pencegahannya.

### **1.3. Batasan Masalah**

Pembuatan program bantu ini hanya terbatas pada:

1. Program bantu ini dibuat untuk pengguna tunggal.
2. Program bantu ini hanya mengambil jenis jenis cacing bersegmen, cacing gilig dan cacing pipih yang sering kali menjadi penyebab penyakit.
3. Cacing yang diambil memiliki sifat parasit terhadap manusia.
4. Penggunaan program bantu, untuk rumah sakit yang menangani masalah kesehatan masyarakat.

#### **1.4. Tujuan Penulisan Tugas Akhir**

Tujuan penulisan tugas akhir ini adalah :

Bagaimana menerapkan langkah-langkah metodologi pengembangan sistem sehingga berguna untuk menampilkan macam-macam cacing, jenis penyakit yang ditimbulkan serta jenis obat yang disertai solusi pencegahannya.

#### **1.5 Metodologi Penulisan**

Pengumpulan data data penulisan tugas akhir ini dengan cara studi pustaka serta konsultasi dengan seorang ahli kesehatan mengenai jenis-jenis cacing yang bersifat parasit serta jenis obat yang dapat digunakan untuk memberantas jenis cacing tersebut. Pembuatan perancangan sistem ini menggunakan model *classic life cycle (waterfall)*. Tahapan tahapan yang digunakan pada model ini adalah sebagai berikut:

1. Analisa kebutuhan sistem

Tahap analisa kebutuhan sangat penting dalam rekayasa perangkat lunak. Perlu diketahui sifat program yang dibuat dengan piranti perangkat lunak baik yang bersifat prosedur maupun fungsi.

2. Rancangan

Rancangan perangkat lunak melibatkan langkah langkah multi proses yang memfokuskan pada struktur data, arsitektur perangkat lunak, prosedur detil dan karakter antar muka.

3. Pengkodean

Rancangan yang telah dibuat diterjemahkan kedalam bentuk yang dapat dibaca oleh mesin pembaca atau *kompailler*.

## **1.6 Sistematika Penulisan**

Sistematika penulisan tugas akhir ini adalah sebagai berikut:

### **Bab I. Pendahuluan**

Pada bab ini dibahas mengenai latar belakang masalah , rumusan masalah, batasan masalah, tujuan dan manfaat penulisan tugas akhir, sistematika serta metodologi penulisan yang digunakan.

### **Bab II. Landasan Teori**

Pada bab ini dibahas mengenai dasar-dasar ilmu yang menunjang dalam membangun program bantu ini, baik mengenai software yang digunakan serta pengetahuan mengenai jenis jenis cacing yang bersifat parasit dan yang tidak.

### **Bab III. Analisa dan Perancangan Sistem**

Pada bagian ini penulis akan membahas serta menjelaskan mengenai proses analisa serta perancangan program yang akan dibuat.

### **Bab IV. Implementasi**

Pada bagian ini penulis akan membahas mengenai detail antar muka pengguna serta perintah yang digunakan untuk membangun *form - form* yang ada.

### **Bab V. Analisa Hasil**

Pada bagian ini berisi analisa tentang hasil program yang telah dibuat.

### **Bab VI. Kesimpulan dan Saran**

Berisi kesimpulan akhir program dan saran saran yang dapat dianjurkan.



## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Definisi Program Bantu**

Fungsi dari program bantu sama dengan sistem pakar dimana masing masing berguna untuk membantu manusia dalam memecahkan suatu masalah. Program Bantu sendiri dapat diartikan bahwa komputer sebagai alat bantu bagi manusia dalam mengatasi suatu hal. Sumber ilmu dari program bantu merupakan kecerdasan buatan yang secara khusus adalah sistem pakar. Walaupun kegunaan dari program bantu mirip dengan sistem pakar akan tetapi dalam hal konsep pemecahan masalahnya sedikit berbeda. Dalam sistem pakar dapat diartikan bahwa program komputer dapat bertindak sebagai penasehat / konsultan yang pintar yang menghasilkan dorongan kepada pengguna untuk melakukan suatu tindakan. Dengan pengetahuan yang telah diprogram, seorang pemakai yang tidak atau belum berpengalaman dapat memecahkan masalah dengan sistem pakar.

Konsep program bantu sedikit lebih luas dari sistem pakar, dikarenakan konsep program bantu ditujukan dan dikembangkan ke arah interaksi manusia dan komputer. Sistem pakar tidak dapat memberikan penilaian terhadap keputusan yang diambil akan tetapi yang dapat diberikan adalah informasi atau saran sehingga pengguna dapat menentukan suatu keputusan. Pada program bantu segala inputan yang digunakan menggunakan bentuk baku, misalnya selalu dalam model matematika

## **2.2. Antar Muka Manusia – Komputer**

Interaksi antara manusia dengan komputer dapat dilakukan dengan cara-cara tertentu, seperti pengguna memberikan perintah tertentu kepada komputer kemudian komputer menanggapi dengan mencetak atau menuliskan tanggapan itu di layar tampilan atau monitor.

Di dalam antar muka grafis, grafika komputer menyediakan suatu bahasa antar muka yang baru hal ini dikarenakan interaksi antara manusia dan komputer tidak lagi melalui suatu perintah yang harus diketikkan di papan ketik, akan tetapi cukup dengan menekan tombol atau simbol tertentu yang terlihat dilayar guna mengerjakan perintah selanjutnya (Insap Santoso,1997). GUI (*Graphical User Interface*) merupakan tipe format program antar muka yang berbasis grafis. Dengan GUI ini semua program dapat dijalankan dengan mouse atau papan ketik.

## **2.3. Basis Data**

Basis data merupakan kumpulan informasi yang diatur dan diorganisasikan dalam suatu cara yang sistematis guna mengurangi duplikasi data. Metode perancangan basis data dengan pendekatan desain database terstruktur memiliki beberapa metodologi perancangan.

### **1. Perancangan Konseptual**

Model konseptual menggambarkan pengorganisasian data secara detail. Dari tahap ini perancang memberikan kebutuhan informatif dari sebuah aplikasi dengan penggambaran yang formal dan lengkap tanpa harus mempertimbangkan cara implementasi informasi tersebut dalam sistem yang sesungguhnya.

## 2. Perancangan Logical

Tahap ini merupakan penterjemahan skema konseptual pada tahap sebelumnya. Hasil penterjemahan ini mengacu pada model data logika. Model data logika memberikan data yang tidak tergantung dengan detail fisik, meskipun DBMS yang digunakan pada implementasi harus mendukung model data. Pada tahap ini perancang harus memperhatikan optimasi berdasar pada operasi yang diterapkan pada data. Teknik untuk menguji kualitas skema logika biasanya menggunakan teknik normalisasi.

## 3. Perancangan Fisik

Pada perancangan fisik dilengkapi dengan detail implementasi fisik (organisasi *file* dan indek) pada DBMS yang diberikan. Tahap ini disebut skema fisik dan mengacu pada model data fisik. Model ini tergantung pada sistem manajemen database yang dipilih, serta memperhatikan kriteria pengorganisasian data fisik pada sistem.

### 2.4 Sistem Informasi

Informasi yang akurat baik dan mudah dipakai sangat dibutuhkan untuk proses pengambilan keputusan bagi suatu manajemen. Informasi tersebut dapat terwujud apabila informasi disusun dengan baik terpadu dan terjaga validitasnya.

#### 2.4.1 Pengertian Sistem Informasi

Sistem merupakan kumpulan dari subsistem yang saling dihubungkan untuk membentuk suatu kesatuan yang utuh. Informasi dapat diartikan sebuah data yang telah diolah menjadi bentuk yang lebih berguna bagi yang menerima sebagai dasar pengambilan keputusan.

Sistem Informasi dapat diartikan sistem pengolah data untuk menghasilkan informasi kepada pihak yang membutuhkan. Suatu sistem informasi dapat dikatakan handal apabila informasi yang dihasilkan bersifat akurat, terpercaya, detail, cepat dan relevan.

#### **2.4.2 Siklus Hidup Pengembangan Sistem Informasi**

Pengembangan sistem informasi mempunyai siklus hidup yang terdiri dari :

##### **1. Tahap Perencanaan**

Dalam tahap perencanaan perlu diadakan identifikasi masalah tentang sistem yang akan dibangun. Hal ini berguna untuk mencapai tujuan atau target yang harus dicapai.

##### **2. Tahap Analisis**

Tahap analisis diperlukan karena berguna untuk menganalisa sistem yang sudah ada, kemudian menganalisa kebutuhan sistem serta bertujuan untuk menyiapkan alternative pemecahan masalah.

##### **3. Tahap Desain**

Pada tahap ini dilakukan evaluasi dari alternatif desain yang ada kemudian dilanjutkan persiapan perancangan desain.

##### **4. Tahap Konstruksi**

Setelah perancangan desain selesai dibuat maka konstruksi pengembangan sistem dapat mulai dibangun. Pada tahap ini biasanya akan banyak melibatkan sumber daya manusia, biaya dan waktu yang tidak sedikit

5. Tahap Implementasi

Tahap implementasi merupakan tahap dimana pertama kali sistem informasi telah dibangun mulai dipergunakan secara nyata. Pada tahap ini biasanya dilakukan pula pendokumentasian perangkat lunak, pengujian serta perawatan sistem dan evaluasi sistem yang telah baru saja digunakan.

6. Tahap *Pasca* Implementasi

Pada tahap ini dibutuhkan personel atau team yang dapat melakukan perubahan terhadap sistem informasi sesuai dengan kebutuhan.

## 2.5 Cacing

Dalam kehidupan sehari-hari kita mengetahui bahwa cacing merupakan hewan kecil lunak bertubuh memanjang, tidak berangka dan tidak mempunyai kaki. Akan tetapi lebih jauh cacing sendiri oleh para ilmuwan diklasifikasikan kedalam tiga kelompok yaitu *platyhelmyntes* (sering juga disebut cacing pipih), *nemanthelmyntes* (sering juga disebut cacing gilig) dan *anelida* (cacing bersegmen). Dari ketiga kelompok cacing tersebut ada yang bersifat parasit dan ada yang tidak. Dari ketiga kelompok tersebut sebenarnya yang paling sering bersifat parasit adalah kelompok cacing pipih dan cacing gilig. Cacing yang bersifat parasit sangat merugikan bagi mahluk yang ditempatinya sebagai tempat tinggalnya. Karena disana cacing akan menghisap sari makanan yang seharusnya digunakan oleh mahluk tersebut untuk bertahan hidup.

### 2.5.1. Jenis Cacing Gilig

Disebut cacing gilig dikeranakan dari bentuk tubuhnya yang memang gilig bulat memanjang dan simetris bilateral. Ciri yang lain tubuhnya tidak bersegmen dan memiliki saluran pencernaan yang jelas. Contoh dari jenis cacing gilig adalah cacing tambang, cacing perut, cacing wucheria bancrofti, cacing kremi, cacing loa atau cacing mata. Hewan perantara yang biasanya dapat menularkan siklus hidup cacing gilig seperti, babi, nyamuk. Sedang melalui tanaman biasanya melalui akar tanaman.

### **2.5.2. Jenis Cacing Pipih**

Untuk mengenali cirri-ciri dari kelompok cacing pipih juga dapat dilihat dari bentuk tubuhnya yang pipih bilateral simetris. Cacing kelompok ini tidak memiliki sistem peredaran darah yang jelas, tidak memiliki anus juga tidak berongga badan. Dari jenis cacing pipih ini yang bersifat parasit adalah *fasciola hepatica* (cacing hati), *taenia solium* (cacing pita), *clonorchis sinensis*, *schistosoma haematobium* (cacing darah). Nama dari kelompok cacing pipih kebanyakan berdasarkan tempat dimana ia hidup dan berkembang biak didalam tubuh inangnya. Misalnya saja cacing hati ini berkembang biak didalam hati hewan ataupun manusia yang terinfeksi cacing hati ini. Contoh yang lainnya cacing yang hidup didalam paru-paru adalah *paragonimus wetermanii*. *Schistosoma* sering disebut juga cacing darah karena berkembang biak didalam pembuluh vena, saluran getah bening. Perantara penyebaran cacing pipih bias any melalui binatang ternak ataupun hewan peliharaan yang kurang dijaga kebersihannya. Hewan yang sering sebagai perantara misalnya babi, siput, sapi, kerbau, kambing, ikan dapat juga melalui tanaman (tanaman yang biasanya untuk sayuran).

### **2.5.3 Jenis Cacing Bersegmen**

Kebanyakan dari cacing bersegmen bersifat tidak parasit karena dapat dimanfaatkan bagi kepentingan manusia apabila dapat mengelolanya. Seperti cacing tanah dapat membantu proses penyuburkan tanah contohnya. Contoh jenis cacing yang bersifat parasit adalah lintah dan pacet. Meskipun lintah dan pacet merugikan manusia karena menghisap darah akan tetapi masih ada yang

bermanfaat sebab cairan yang dikeluarkan dari mulutnya dalam dunia kedokteran dahulu dapat digunakan sebagai zat antibodi. Untuk jenis cacing wawo dan cacing palolo sering digunakan untuk sumber bahan makanan.



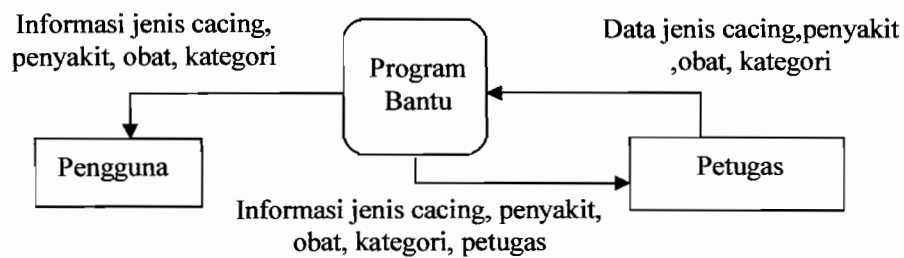
## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

Proses analisis dan perancangan sistem dapat digunakan untuk menggambarkan bagaimana jalannya sistem secara nyata. Untuk pemodelan sistem menggunakan beberapa alat bantu seperti diagram arus data (DAD) yang terdiri atas diagram konteks, diagram berjenjang, dan diagram arus data level 0.

##### 3.1.1 Diagram Kontek



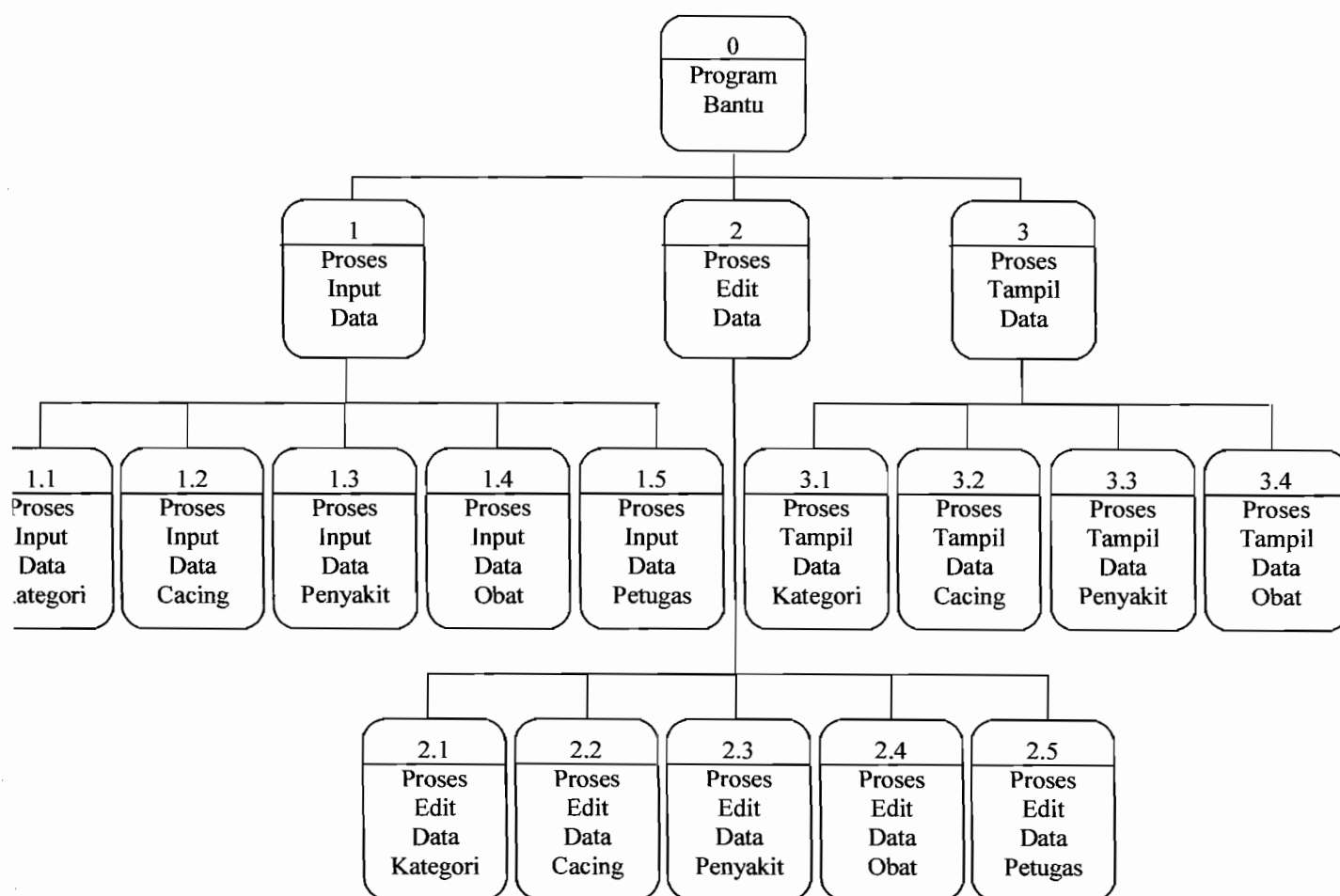
Gambar 3.1 Diagram konteks

Diagram konteks diatas dapat dijelaskan sebagai berikut:

Program bantu akan mendapat masukan dari administrator yang bertugas untuk mengupdate data berupa informasi yang berkaitan dengan jenis cacing serta pengaruhnya terhadap kesehatan. Pengguna akan memperoleh data yang diinginkan dari program bantu ini.

### 3.1.2 Diagram Berjenjang

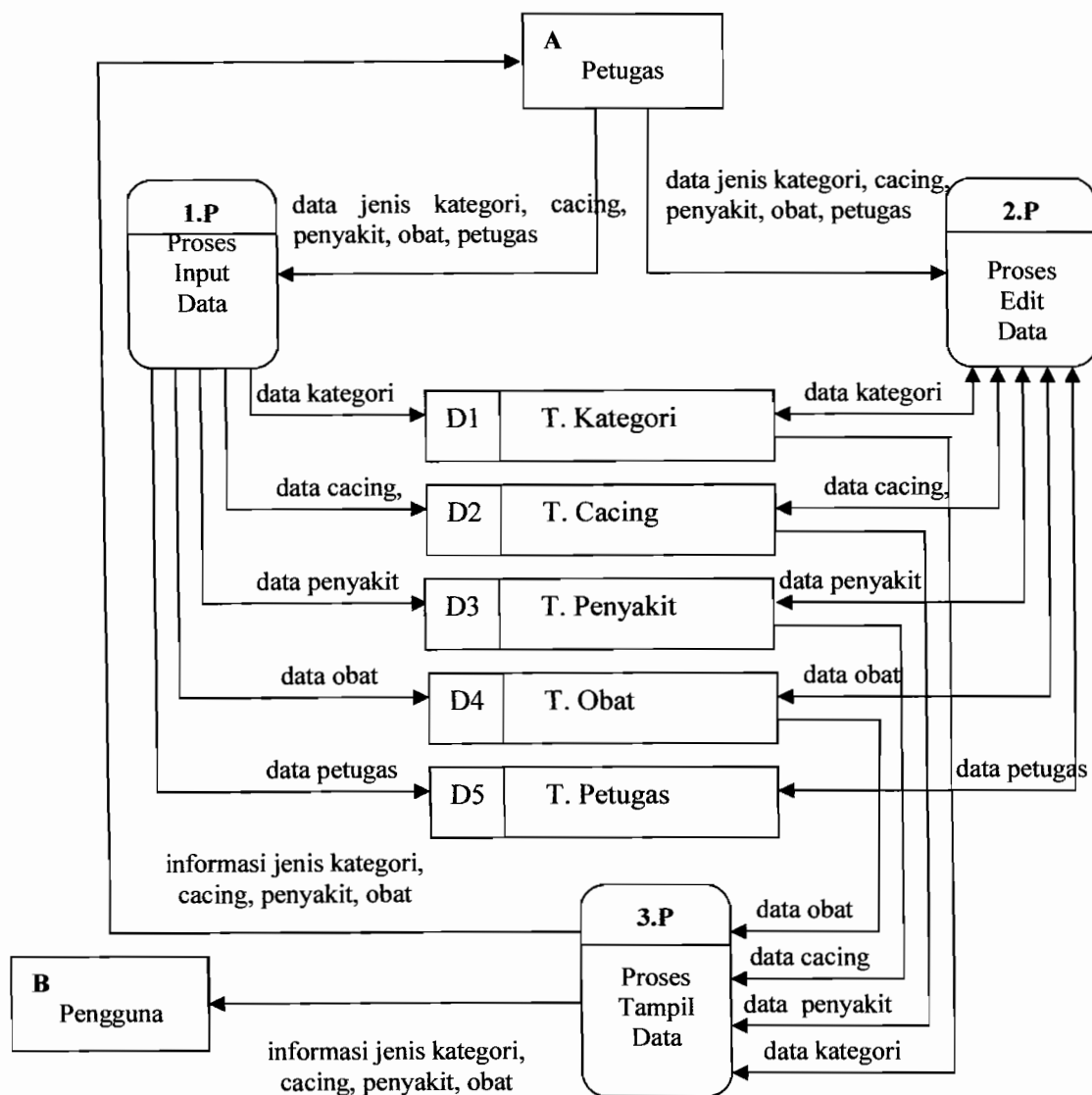
Didalam diagram berjenjang ini terdapat dua proses utama yang terjadi di dalam program bantu yaitu pilih status, input dan edit data. Gambar diagram berjenjang dapat kita lihat seperti pada gambar 3.2:



Gambar 3.2 Diagram berjenjang

### 3.1.4 Diagram Arus Data Level 0 (Overview Diagram)

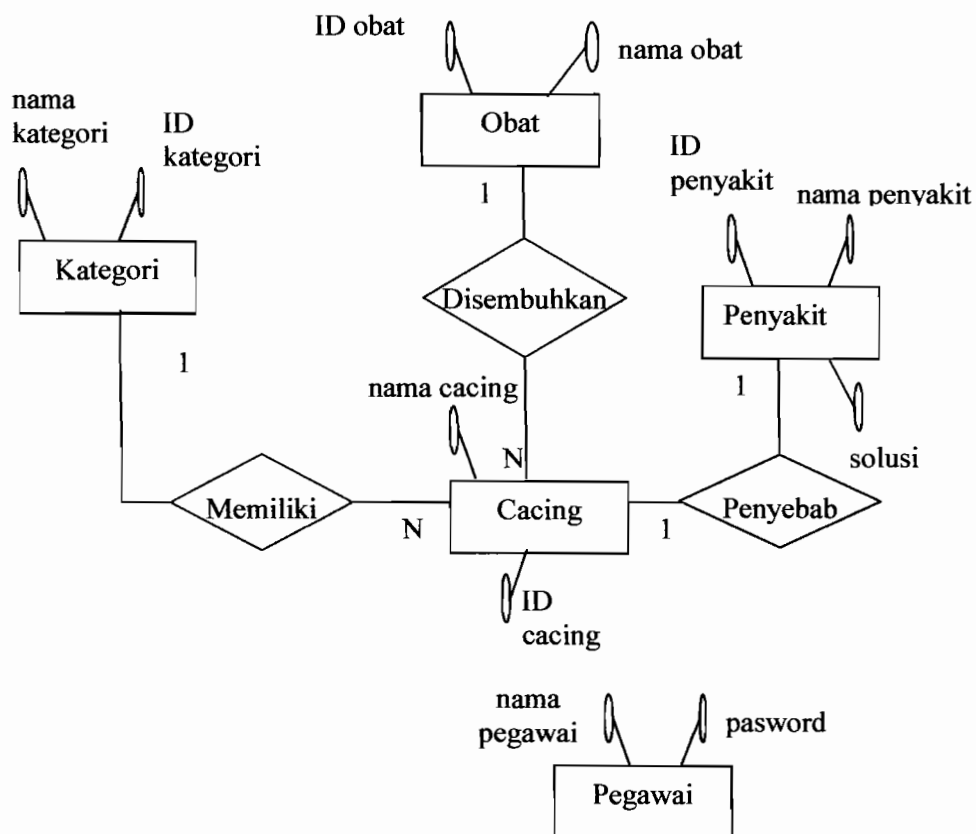
Diagram Arus Data digunakan untuk menggambarkan sistem sebagai satu kesatuan unit kerja antar fungsi yang saling berhubungan satu dengan yang lain. Aliran data dari masing masing entitas yang berhubungan dapat kita lihat seperti pada gambar 3.3 berikut ini.



Gambar 3.3 Diagram arus data level 0

### 3.1.5. ER Diagram

ER Diagram digunakan untuk menggambarkan hubungan antara entitas yang satu dengan yang lain serta atribut-atribut yang mengikuti entitas yang ada. Sebuah entitas cacing mempunyai atribut ID cacing dan nama cacing. Entitas penyakit memiliki atribut nama penyakit, ID penyakit, solusi sementara entitas pegawai memiliki atribut nama pegawai, password. Entitas kategori memiliki atribut ID kategori, nama kategori. Entitas obat memiliki atribut ID obat dan nama obat. Diagramnya dapat kita lihat seperti pada gambar 3.4 di bawah ini.

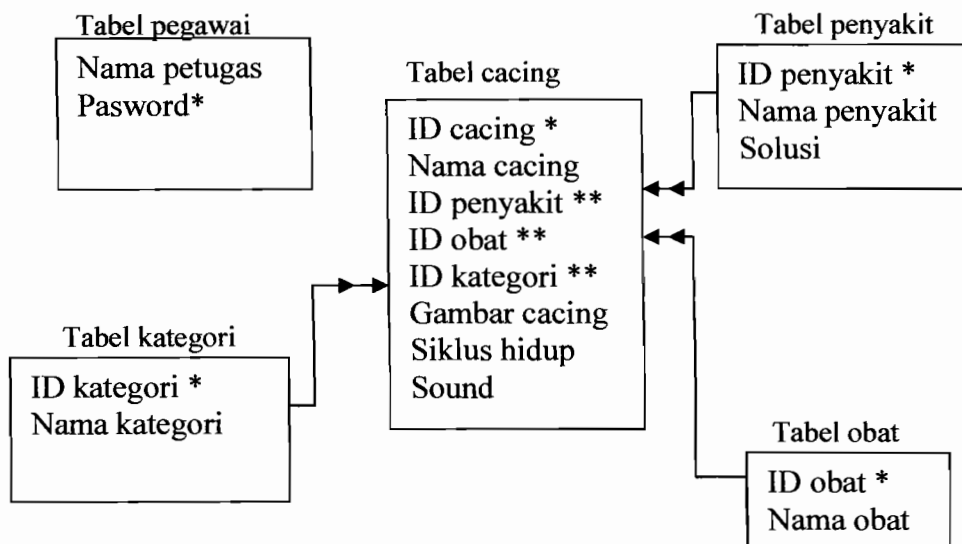


Gambar 3.4 ER Diagram

### 3.2 Perancangan Basisdata

Guna memudahkan dalam menghubungkan beberapa data yang ada maka perancangan basisdata ini diperlukan. Perancangan basisdata ini terdiri atas beberapa tabel yang saling dihubungkan satu dengan yang lain sehingga membantu dalam pemrosesan data.

Desain basisdatanya seperti terlihat pada gambar di bawah ini:



Gambar 3.5 Gambar desain basisdata

Dari ER Diagram diperoleh beberapa bentuk tabel yang akan diolah oleh sistem

No	Nama Item Data	Tipe	Lebar
1.	Nama petugas	Text	20
2.	Pasword	Text	5

Tabel 3.1 Tabel pegawai

Tabel pegawai digunakan untuk menyimpan pasword petugas yang boleh mengupdate data. Tabelnya seperti pada tabel 3.1.

No	Nama Item Data	Tipe	Lebar
1	ID cacing	Text	4
2.	Nama cacing	Text	25
3.	ID penyakit	Text	4
4.	ID obat	Text	4
5.	ID kategori	Text r	3
6.	Gambar cacing	Text	
7.	Siklus hidup	Text	
8.	Sound	Text	

Tabel 3.2 tabel cacing

Tabel cacing digunakan untuk menyimpan jenis jenis cacing yang ada, gambar cacing serta siklus hidup cacing. Tabelnya seperti tabel 3.2 diatas.

No	Nama Item Data	Tipe	Lebar
1.	ID penyakit	Text	4
2.	Nama Penyakit	Text	25
3.	Solusi	Text	100

Tabel 3.3 tabel penyakit

Tabel penyakit ini digunakan untuk menyimpan nama nama penyakit yang ditimbulkan serta solusi pencegahannya. Tabelnya seperti pada tabel 3.3 di atas.

No	Nama Item Data	Tipe	Lebar
1.	ID obat	Text	4
2.	Nama obat	Text	10

Tabel 3.4 tabel obat

Tabel obat digunakan untuk menyimpan nama obat yang dapat digunakan untuk membunuh cacing. Tabelnya seperti pada tabel 3.4 di atas.

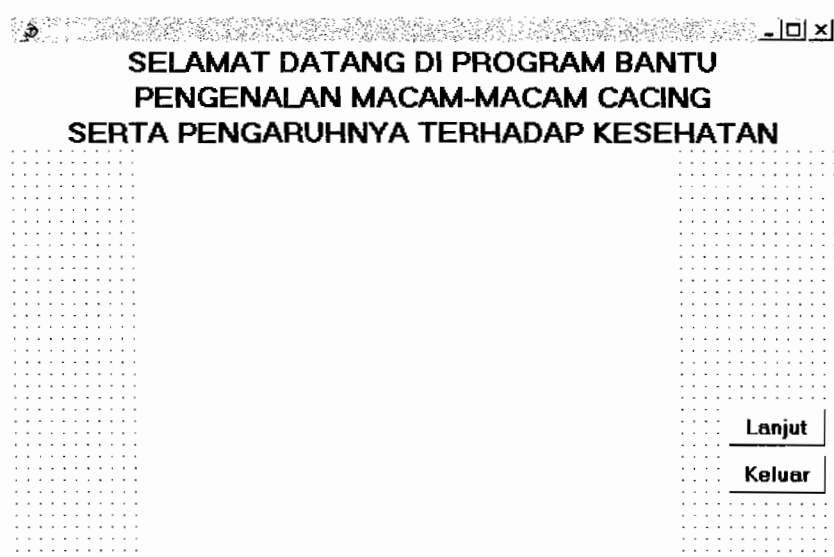
No	Nama Item Data	Tipe	Lebar
1.	ID kategori	Text	3
2.	Nama kategori	Text	15

Tabel 3.5 tabel kategori

Tabel kategori ini digunakan untuk menyimpan kategori cacing yaitu cacing gilig, pipih, bersegmen. Tabelnya seperti pada tabel 3.5 di atas.

### 3.2.1 *Form* Utama

*Form* utama merupakan langkah awal yang harus dilakukan oleh pengguna sehingga dapat menggunakan program ini. Di dalam *form* utama ini terdapat menu menu yang dapat dijadikan petunjuk bagi pengguna untuk mendapatkan informasi yang ada di dalam program.

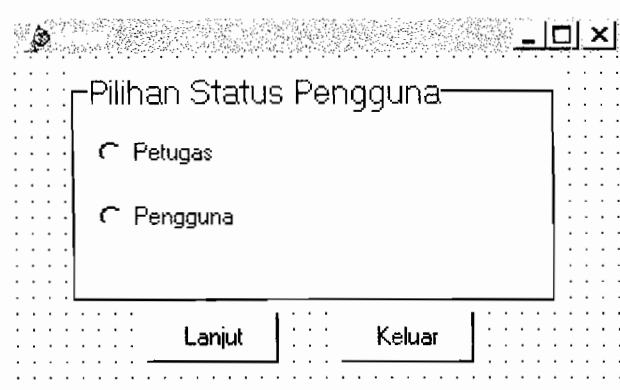


Gambar 3.6 *Form* utama



### 3.2.2 Form Status

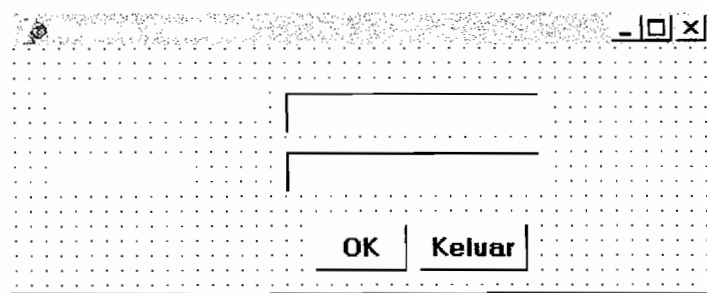
Setelah memilih menu utama pengguna akan dihadapkan pada menu status dimana program akan memberikan izin kepada pengguna batas yang dapat diakses apabila selain administrator tidak di ijinakan untuk mengupdate data. Sehingga hanya petugas yang dapat masuk ke pilihan petugas.



Gambar 3.7 Form Status

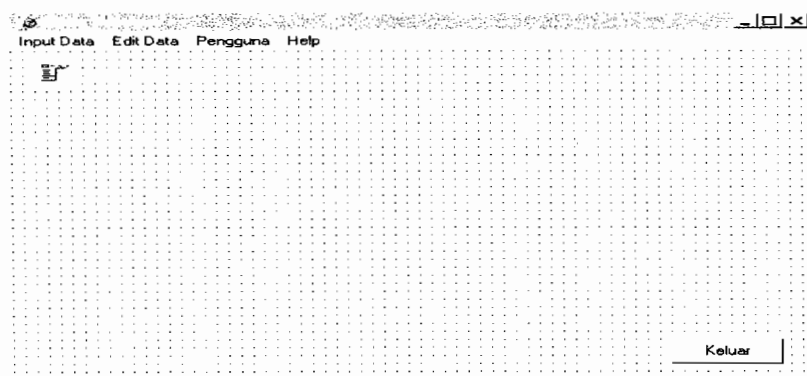
### 3.2.3 Form Petugas

Untuk dapat masuk ke dalam *form* menu petugas, petugas terlebih dahulu harus mengisikan password agar dapat merubah sistem. Rancangannya seperti terlihat pada gambar 3.8 berikut.

A screenshot of a graphical user interface window titled "Form registrasi password". The window has a standard title bar with minimize, maximize, and close buttons. The main area contains two empty text input fields stacked vertically. Below the input fields are two buttons: "OK" and "Keluar". The background of the window is a light gray grid pattern.

Gambar 3.8 Form registrasi password

Form berikutnya yang akan terlihat adalah form menu petugas. Pada *form* ini administrator atau petugas yang telah diberi wewenang dapat melakukan input serta edit data mengenai jenis cacing, jenis penyakit, nama obat serta solusinya. Kemudian mengupdate petugas yang sudah terdaftar maupun menambah baru. Bentuk dari rancangan *form* seperti pada gambar 3.9 berikut ini.

A screenshot of a graphical user interface window titled "Form menu petugas". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with the following items: "Input Data", "Edit Data", "Pegguna", and "Help". The main area of the window is a large, empty grid pattern. In the bottom right corner, there is a button labeled "Keluar".

Gambar 3.9 Form menu petugas

**INPUT CACING**

Kode Cacing:

Nama Cacing:

Nama Penyakit:

Nama Obat:

Nama Kategori:

Gambar Cacing:  Cari...

Siklus Hidup:  Cari...

Tutorial Suara:  Cari...

Tambah  
Simpan  
Batal  
Hapus  
Keluar

ID_Cacing	Nama Cacing	ID_Penyakit	ID_Obat	ID_Kategori	Gambar Cacing
C001	Cacing Kremi	P002	B002	K002	C:\SADAR\IMAGES\Cisap.jpg
C002	Cacing Tambang	P004	B005	K002	C:\SADAR\IMAGES\Ctambang.jpg

Gambar 3.10 Form input cacing

**INPUT PENYAKIT**

Kode Penyakit:

Nama Penyakit:

Solusi:

Tambah  
Simpan  
Batal  
Hapus  
Keluar

ID_Penyakit	Nama Penyakit	Solusi
P002	ascariasis	Jangan makan kalo tangannya kotor
P003	schistosomiasis	Minum jamu anti cacing kremi
P004	Weteng Mengkiing	Makan makanan yang steril

Gambar 3.11 Form input penyakit

**INPUT KATEGORI CACING**

Kode Kategori:

Nama Kategori:

Tambah  
Simpan  
Batal  
Hapus  
Keluar

ID_Kategori	Nama Kategori
K.001	Cacing Pipih
K.002	Cacing Gilig
K.003	Cacing Bersegmen

Navigation arrows: [Left] [Left] [Right] [Right]

Gambar 3.12 Form input kategori

**INPUT OBAT**

Kode Obat:

Nama Obat:

Tambah  
Simpan  
Batal  
Hapus  
Keluar

ID_obat	Nama Obat
B002	fenbendazole
B003	praziquantel
B004	pyrantel pamoat

Navigation arrows: [Left] [Left] [Right] [Right]

Gambar 3.13 Form input obat

Sub menu *form* selanjutnya adalah form edit data obat, cacing, penyakit, input petugas serta *form help*. Gambar *form* formnya terlihat seperti di bawah ini.

ID_Cacing	Nama Cacing	ID_Penyakit	ID_Obat	ID_Kategori	Gambar Cacing
C001	Cacing Kremi	P002	B002	K002	C:\SADAR\IMAGES\Ccacing.jpg
C002	Cacing Tambang	P004	B005	K002	C:\SADAR\IMAGES\Ccacing.jpg
C003	Cacing Perut	P005	B006	K001	C:\SADAR\IMAGES\Ccacing.jpg

Gambar 3.14 *Form* edit cacing

ID_Kategori	Nama Kategori
K001	Cacing Pipih
K002	Cacing Gilig
K003	Cacing Bersegmen

Gambar 3.15 *Form* edit kategori

Kategori Obat | Penyakit | Cacing

### EDIT OBAT

Kode Obat: B002  
 Nama Obat: Ienbendazole

Edt  
 Simpan  
 Batal  
 Hapus

ID_obat	Nama Obat
B001	pyrantel
B002	Ienbendazole
B003	praziquantel
B004	pyrantel pamoat
B005	Obat "Smack Down" Cacing Tambang
B006	piperazine

Keluar

Gambar 3.16 Form edit obat

Kategori | Obat | Penyakit | Cacing

### EDIT PENYAKIT

Kode Penyakit: P002  
 Nama Penyakit: ascariasis  
 Solusi: Jangan makan kalo tangannya kotor habis nyangkut diawah

Edit  
 Simpan  
 Batal  
 Hapus

ID_Penyakit	Nama Penyakit	Solusi
P002	ascariasis	Jangan makan kalo tangannya kotor habis nyangkut diawah
P003	schistosomiasis	Minum jamu anti cacing krema
P004	Weteng Mengkitig	Makan makanan yang steil
P005	ascariasis	menghindari makanan setengah matang

Keluar

Gambar 3.17 Form edit obat

INPUT PENGGUNA (USER)

User name:

Password:

Nama Pegawai:

Tambah  
Simpan  
Batal  
Edit  
Hapus

Nama Pegawai	Username	Password
▶ didik	didik	didik
□ budi	budi	budi

<< < > >> Keluar

Gambar 3.18 Form edit petugas

BANTUAN PENGGUNAAN PROGRAM

Program dijalankan melalui pemilihan menu.

Exit

Gambar 3.19 Form help

### 3.2.5 Form Pengguna

Rancangan *form* pengguna akan terlihat seperti pada gambar berikut ini. Di dalam *form* terdapat dua buah menu informasi pengguna dan abaut. Dari menu informasi pengguna dapat mengetahui nama cacung, kategorinya, jenis penyakit yang ditimbulkan, obat untuk mengatasi dan solusi seperti pada gambar 3.20. Menu *help* dapat dilihat bagaimana langkah langkah mengoperasikan program bantu ini, gambar 3.21.

ID_Cacing	Nama Cacing	Nama Penyakit	Nama Obat
C001	Cacing Kremi	ascariasis	fenbendazole
C002	Cacing Tambang	Weling Mengliang	Obat "Smack Down"
C003	Cacing Perut	ascariasis	piperazone

Gambar 3.20 *Form* informasi pengguna

Aplikasi Informasi Cacing dan Kesehatan Manusia

Version 1.0

OK

Gambar 3.21 *Form* about



### 3.2.6 Arsitektur *Hardware Software*

Sistem tidak akan terbentuk apabila tidak ada kesatuan antara hardware dan software sebagai sarana pembangun sistem.

#### 1. *Software*

Borland Delphi ver 5.0, 6.0, 7.0 merupakan *software* yang dapat digunakan untuk menjalankan program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan. Sistem operasi yang digunakan adalah Windows 98. Untuk media penyimpanan database menggunakan Microsoft Access 2000.

#### 2. *Hardware*

*Hardware* terdiri dari beberapa bagian yang membentuk sebuah PC yang merupakan peralatan fisik dari komputer. Spesifikasi *hardware* yang digunakan adalah sebagai berikut:

PC 233 MMX

*Memory* 64 Mb

*VGA Card*

*Harddisk* 10 GB

*Floopydisk* 1.44 Mb

*Monitor* 14"

Spiker, *keyboard*, *mouse*

## **BAB IV**

### **IMPLEMENTASI**

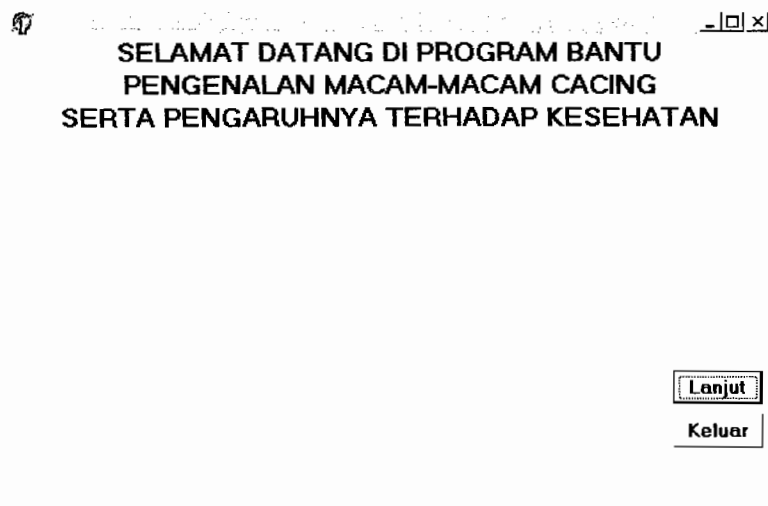
Program bantu pengenalan macam macam cacing serta pengaruhnya terhadap kesehatan terdiri dari beberapa modul-modul database yang terhubung antara yang satu dengan yang lainnya memnjadi suatu aplikasi yang tergabung menjadi satu dan terdiri dari beberapa *form-form* sebagai berikut:

1. *Form* Selamat Datang
2. *Form* Menu Petugas
3. *Form* Input Data
4. *Form* Edit Data
5. *Form* Pengguna (Output Program)

#### 4.1. Hasil Program

Dari hasil perancangan yang telah terbentuk maka proses selanjutnya adalah pengujian dengan menggunakan sebuah software program.

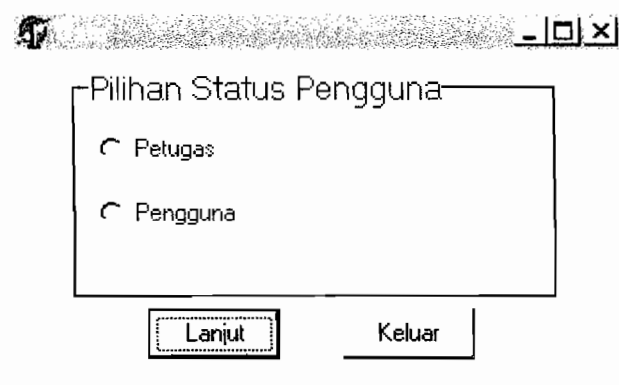
##### 4.1.1. Menu Selamat Datang



Gambar 4.1 Menu Selamat Datang

Menu selamat datang merupakan menu awal dari program bantu pengenalan macam macam cacing serta pengaruhnya terhadap kesehatan, apabila tombol lanjut ditekan maka akan masuk ke dalam menu berikutnya

#### 4.1.2 Menu Pilihan Status



Pilihan Status Pengguna

Petugas

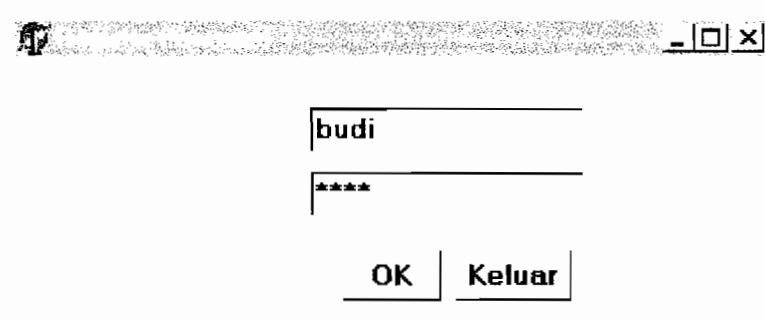
Pengguna

Lanjut      Keluar

Gambar 4.2 Menu Pilihan Status

Form pilihan status pengguna berfungsi untuk mengarahkan *user* tingkat penggunaan *system*. Apabila *user* memilih *button* petugas maka akan dibawa ke *form* registrasi *password* seperti pada gambar 4.3. Akan tetapi apabila *user* memilih *button* pengguna maka akan ditampilkan *form* informasi pengguna 4.11.

#### 4.1.3 Menu Registrasi Pasword



budi

\*\*\*\*

OK      Keluar

Gambar 4.3 Menu registrasi pasword

## 4.2. Menu Input Program

Menu input program berfungsi untuk menginputkan data-data yang masih belum ada di dalam program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan. Menu input program ini terdiri dari input cacing, input penyakit, input obat.

### 4.2.1. Menu Input Cacing

ID_Cacing	Nama Cacing	ID_Penyakit	ID_Obat	ID_Kategori	Gambar Cacing
C001	Cacing Kremi	P002	B002	K002	C:\SADAR\IMAGES\Cisap.jpg
C002	Cacing Tambang	P004	B005	K002	C:\SADAR\IMAGES\Ctambang.jpg
C003	Cacing Perut	P005	B006	K001	C:\SADAR\IMAGES\Cperut.jpg

Gambar 4.4 Menu input cacing

Kode cacing berfungsi untuk membedakan nama cacing. Gambar cacing merupakan gambar dari nama cacing, siklus hidup juga merupakan siklus hidup dari cacing. Kode obat, kode penyakit, kode kategori dalam combobox berfungsi untuk menampilkan pilihan kode. Tombol batal berfungsi untuk membatalkan penambahan data. Tombol simpan berfungsi untuk menyimpan data. Tombol keluar berfungsi untuk kembali ke menu petugas.

Berikut potongan program penambahan data cacing

```
procedure TFormInputCacing.tambahClick(Sender: TObject);
var nomor : string;

begin
q1.Close;
q1.sql.Clear;
q1.SQL.add('select max(id_cacing) from cacing');
q1.open;
if q1.Fields[0].asString="" then nomor:='C001'
else
begin
nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
if length(nomor)=1 then
nomor:='C00'+nomor;
if length(nomor)=2 then
nomor:='C0'+nomor;
if length(nomor)=3 then
nomor:='C'+nomor;
end;
```

#### 4.2.2. Menu Input Penyakit

ID_Penyakit	Nama Penyakit	Solusi
P002	ascariasis	Jangan makan kalo tangannya kotor
P003	schistosomiasis	Minum jamu anti caceng kremi
P004	Weteng Mengkiri	Makan makanan yang steril
P005	trascioliasis	menghindari makanan setengah matang

Gambar 4.5 Menu input penyakit

Menu input penyakit terdiri dari kode penyakit, nama penyakit, solusi. Tombol tambah dan simpan untuk membatalkan dan menyimpan penambahan. Tombol hapus berfungsi menghapus record yang tidak benar dengan mengarahkan tombol *navigator*. Listing programnya penambahan penyakit sebagai berikut.

```

procedure TFormInputPenyakit.tambahClick(Sender: TObject);
var nomor : string;

begin
q1.Close;
q1.sql.Clear;
q1.SQL.add('select max(id_penyakit) from penyakit');
q1.open;
if q1.Fields[0].AsString="" then nomor:='P001'
else
begin
nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
if length(nomor)=1 then
nomor:='P00'+nomor;
if length(nomor)=2 then
nomor:='P0'+nomor;
if length(nomor)=3 then
nomor:='P'+nomor;

```

### 4.2.3. Menu Input Obat

ID_obat	Nama Obat
B002	fenberdazole
B003	praziquantel
B004	pyrantel pamoat
B005	titrazine
B006	piperazine

Gambar 4.6 Menu Input Obat

Menu input obat terdiri dari kode obat, nama obat. Tombol batal dan simpan untuk membatalkan dan menyimpan penambahan. Tombol hapus berfungsi untuk menghapus record tertentu yang tidak benar. Tombol *navigator* berfungsi untuk mengarahkan record ke awal atau akhir. Tombol keluar untuk kembali ke menu petugas. Listing programnya penambahan penyakit sebagai berikut

```

procedure TFormInputObat.tambahClick(Sender: TObject);
var nomor : string;

begin
q1.Close;
q1.sql.Clear;
q1.SQL.add('select max(id_obat) from obat');
q1.open;
if q1.Fields[0].asString="" then nomor:='B001'
else
begin
nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
if length(nomor)=1 then

```



#### 4.2.4 Input Kategori

Gambar 4.7 Menu Input Kategori

Menu input kategori terdiri dari kode kategori, nama kategori. Tombol batal dan simpan untuk membatalkan dan menyimpan penambahan. Tombol keluar untuk kembali ke menu petugas. Listing programnya penambahan penyakit sebagai berikut

```

begin
q1.Close;
q1.sql.Clear;
q1.SQL.add('select max(id_kategori) from kategori');
q1.open;
if q1.Fields[0].asString="" then nomor:='K001'
else
begin
nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
if length(nomor)=1 then
    nomor:='K00'+nomor;
if length(nomor)=2 then
    nomor:='K0'+nomor;
if length(nomor)=3 then
    nomor:='K'+nomor;

```

### 4.3. Menu Edit

Menu edit berguna untuk melakukan perubahan data yang telah tersimpan sebelumnya.

#### 4.3.1. Menu Edit Cacing

Kategori | Obat | Penyakit | Cacing |

**EDIT KATEGORI CACING**

Kode Kategori:

Nama Kategori:

ID_Kategori	Nama Kategori
K001	Cacing Pipih
K002	Cacing Gilig
K003	Cacing Berregmen

Gambar 4.8 Menu Edit Cacing

Untuk pengeditan cacing kode cacing ditampilkan dalam combobox. Tombol buka berfungsi menampilkan nama file gambar cacing dan siklus cacing. Tombol batal untuk membatalkan pengeditan, tombol hapus untuk menghapus record, tombol simpan untuk menyimpan data hasil pengeditan. Listing programnya adalah

```

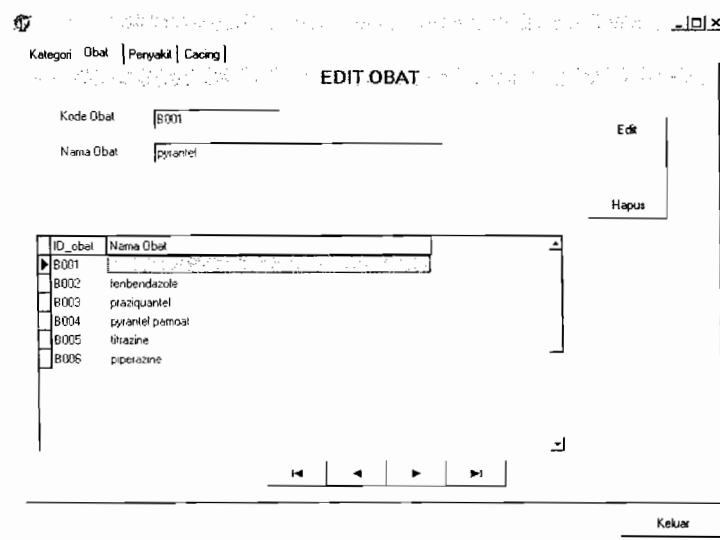
procedure TFormEdit.EditCClick(Sender: TObject);
var nomor : string;

begin
dm.tcacing.edit;

editC.enabled:=false;
simpanC.enabled:=true;
batalC.enabled:=true;

```

### 4.3.2. Menu Edit Obat



Gambar 4.9 Menu Edit Obat

Pada menu edit obat terdapat *field* ID\_Obat dan Nama\_Obat didalam DBGrid atau terlihat seperti tabel. Tombol hapus untuk menghapus record, simpan untuk menyimpan hasil pengeditan, tombol navigator berfungsi untuk mengarahkan record yang akan diedit. Tombol keluar untuk kembali ke menu petugas. Potongan programnya adalah

```

procedure TFormEdit.SimpanOClick(Sender: TObject);
begin
dm.tobat.post;
edito.enabled:=true;
simpano.enabled:=false;
batalo.enabled:=false;

end;

```

### 4.3.3. Menu edit Penyakit

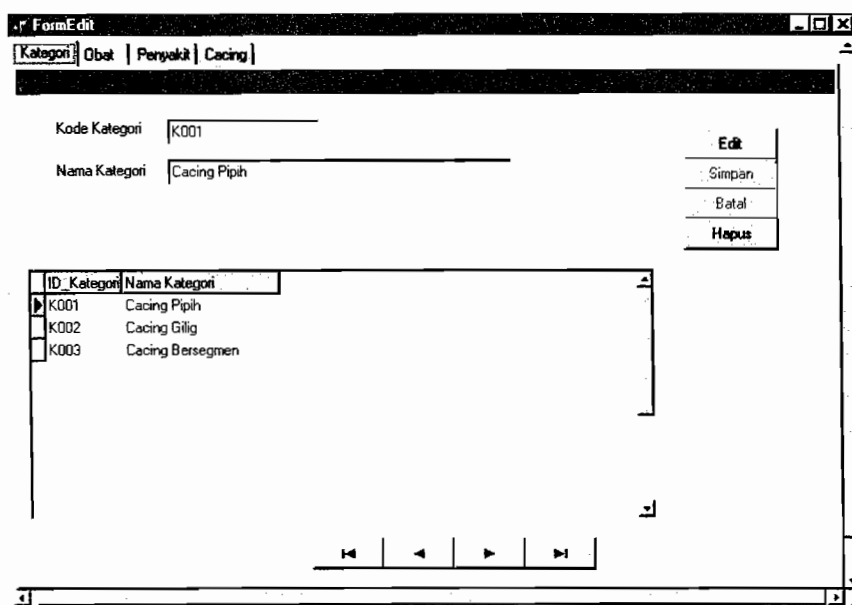
Gambar 4.10 Menu Edit Penyakit

Pada menu edit penyakit berguna untuk mengedit data penyakit akibat cacing . Tombol hapus untuk menghapus record, simpan untuk menyimpan hasil pengeditan, keluar untuk kembali ke menu petugas. Potongan programnya adalah

```
procedure TFormEdit.SimpanOClick(Sender: TObject);
begin
  dm.tobat.post;
  edito.enabled:=true;
  simpano.enabled:=false;
  batalo.enabled:=false;

end;
```

#### 4.3.4 Menu Edit Kategori



Gambar 4.11 Menu Edit Kategori

Pada menu edit kategori terdapat beberapa tombol. Tombol hapus untuk menghapus record, simpan untuk menyimpan hasil pengeditan, keluar untuk kembali ke menu petugas. Potongan programnya adalah

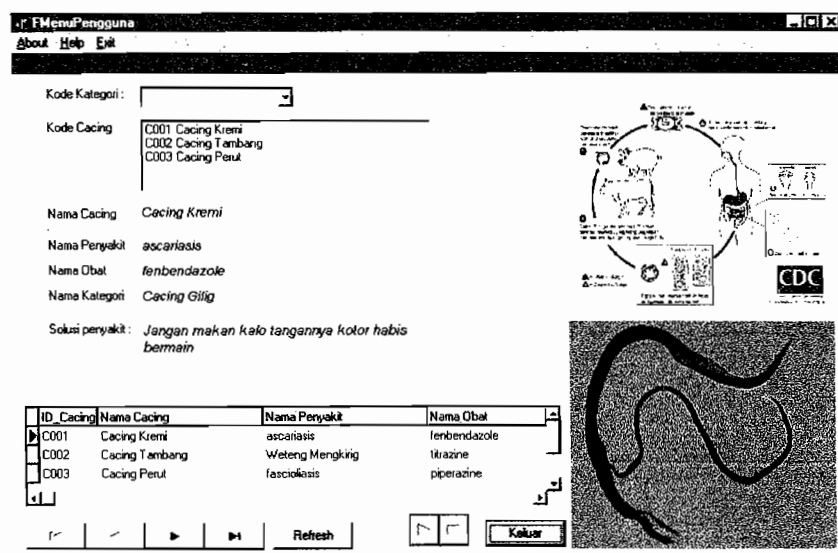
```

procedure TFormEdit.SimpanKClick(Sender: TObject);
begin
  dm.tkategori.post;
  editk.enabled:=true;
  simpank.enabled:=false;
  batalok.enabled:=false;

end;

```

#### 4.4. Menu Output Pengguna



Gambar 4.11 Menu Output pengguna

Pada menu pengguna tombol keluar untuk keluar dari program. Tombol media player digunakan untuk mendengarkan suara dari uraian siklus hidup cacing yang terpilih dari nama cacing berdasarkan pilihan kategori. Dari nama cacing yang terpilih akan menampilkan gambar cacing penyakit yang bisa

ditimbulkan, nama obat yang dapat membunuh cacing tersebut serta gambar siklus hidup cacing.

Potongan programnya adalah

```
procedure TFMenuPegguna.DBLookupComboBox1Click(Sender:
TObject);
var strsql : string;
begin
q2.Close;
if dblookupcombobox1.Text ='' then
strsql:='SELECT Cacing.ID_Cacing, Cacing.[Nama Cacing],
Cacing.ID_Penyakit, Penyakit.[Nama Penyakit], Cacing.ID_Obat,
Obat.[Nama Obat], Cacing.ID_Kategori, Kategori.[Nama Kategori],
Cacing.[Gambar Cacing], Cacing.[Siklus Hidup], Cacing.[Sound
Tutorial]'
+' ,penyakit.solusi FROM Penyakit INNER JOIN (Obat INNER JOIN
(Kategori INNER JOIN Cacing ON Kategori.ID_Kategori =
Cacing.ID_Kategori) ON Obat.ID_obat = Cacing.ID_Obat) ON
Penyakit.ID_Penyakit = Cacing.ID_Penyakit'
else
strsql:='SELECT Cacing.ID_Cacing, Cacing.[Nama Cacing],
```

## **BAB V**

### **ANALISA HASIL**

#### **5.1. Analisis Perangkat Lunak**

Pembuatan program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan menggunakan software Borland Delphi 7.0. Untuk pengembangan sistem yang lebih besar lagi Borland Delphi 7.0 cocok dan kompatibel karena memiliki banyak fasilitas yang lengkap. Borland Delphi 7.0 juga dapat berjalan dengan menggunakan Microsoft Access 2000 sebagai media penyimpanan databasenya.

#### **5.2 Analisis Hasil Program**

Sistem yang dibuat dapat menampilkan informasi tentang nama cacing, jenis penyakit yang ditimbulkannya, serta bagaimana pengobatan maupun solusi untuk mencegah terkena infeksi akibat cacing. Sistem juga menyediakan gambar cacing serta siklus hidupnya walau hanya gambar statis akan tetapi dilengkapi dengan media suara, yaitu penjelasan atau gambaran siklus hidup cacing dapat didengarkan melalui media speaker. Dapat digunakan juga untuk menambah data baru.

Kelemahan yang masih dirasa adalah tampilan yang masih sangat sederhana. Pengguna hanya bertindak sebagai user pasif dalam hal ini tidak melakukan inputan ke sistem.



### **5.3 Analisis Metodologi**

Metode yang digunakan dalam pengembangan sistem adalah teknik classic life cycle. Metode ini menggambarkan secara jelas proses perancangan program dari awal hingga akhir sampai dapat diimplementasikan. Untuk yang sedang dalam belajar mengembangkan sistem metode ini dapat dipergunakan karena mudah dipelajari dan dapat menghasilkan dokumentasi yang baik pula.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1. Kesimpulan

Kesimpulan yang dapat diambil dari program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan adalah :

1. Sitem secara keseluruhan dapat berfungsi dengan baik apabila diterapkan didalam rumah sakit yang menyediakan fasilitas kesehatan masyarakat secara umum.
2. Borland Delphi 7 dan Microsoft Access 2000 sangat mendukung untuk pengembangan sistem yang dibuat.
3. Metode *classic life cycle* dapat digunakan untuk pengembangan program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan.

#### 6.2. Saran

Program bantu pengenalan macam-macam cacing serta pengaruhnya terhadap kesehatan dapat dikembangkan lagi seperti:

1. Untuk program pembelajaran siswa tentang bidang studi Biologi bagian klasifikasi cacing.
2. Bagian tampilan gambar dapat diperbaharui dengan *software* yang lebih baik / menarik lagi dengan adanya fasilitas inputan dari pengguna.

## DAFTAR PUSTAKA

- M. Agus J. Alam, @2000 Mengolah Database dengan Borland Delphi 7. Elex Media Komputindo, Jakarta.
- Ir. Inge Martina, @2004. Pemrograman Visual Borland Delphi 7. Elex Media Komputindo, Jakarta.
- Wahana Komputer, @2003. Pemrograman Borland Delphi 7. Andi Offset, Yogyakarta.
- MADCOM, @2003. Panduan Pemrograman Borland Delphi 7. Andi Offset, Yogyakarta.
- Bambang Robiin, @2000. Mengolah Database Menggunakan Access 2000. Andi Offset, Yogyakarta.
- Slamet Prawirohartono, @2003. Sains Biologi I. Bumi Aksara, Jakarta.
- Yudha Setiawan, @2002. Obat Tradisional Berbagai Penyakit, Gramedia, Jakarta.
- Klasifikasi Phylum Nematelmyntes. <http://www.praweda.co.id/>
- Siklus Hidup Cacing .  
[http://www.biosci.edu/~parasite/lifecycles/strongyloides\\_lifecycle.html](http://www.biosci.edu/~parasite/lifecycles/strongyloides_lifecycle.html).

## **LAMPIRAN**

```
unit Helppengguna;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls;
```

```
type
```

```
TBantuanPengguna = class(TForm)  
    Button1: TButton;  
    Memo1: TMemo;  
    procedure Button1Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    { Public declarations }  
end;
```

```
var
```

```
BantuanPengguna: TBantuanPengguna;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TBantuanPengguna.Button1Click(Sender: TObject);
```

```
begin  
    close;  
end;
```

```
end.
```

```
unit Menupengguna;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, StdCtrls, MPlayer, ExtCtrls, DB, ADODB, DBCtrls, Mask,  
Grids, DBGrids, jpeg;
```

```
type
```

```
TMainMenuPengguna = class(TForm)  
    MainMenu1: TMainMenu;
```

About1: TMenuItem;  
Help1: TMenuItem;  
Exit1: TMenuItem;  
Button1: TButton;  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Label6: TLabel;  
Bevel1: TBevel;  
Bevel2: TBevel;  
DBGrid1: TDBGrid;  
DBNavigator1: TDBNavigator;  
MediaPlayer1: TMediaPlayer;  
DataSource1: TDataSource;  
DBText1: TDBText;  
DBText2: TDBText;  
DBText3: TDBText;  
DBText4: TDBText;  
DBText5: TDBText;  
q2: TADOQuery;  
q2ID\_Cacing: TWideStringField;  
q2NamaCacing: TWideStringField;  
q2ID\_Penyakit: TWideStringField;  
q2NamaPenyakit: TWideStringField;  
q2ID\_Obat: TWideStringField;  
q2NamaObat: TWideStringField;  
q2ID\_Kategori: TWideStringField;  
q2NamaKategori: TWideStringField;  
q2GambarCacing: TWideStringField;  
q2SiklusHidup: TWideStringField;  
q2SoundTutorial: TWideStringField;  
Image2: TImage;  
Button3: TButton;  
Label7: TLabel;  
q2Solusi: TWideStringField;  
DBText6: TDBText;  
Label8: TLabel;  
ADOTable1: TADOTable;  
ADOTable1ID\_Kategori: TWideStringField;  
ADOTable1NamaKategori: TWideStringField;  
DataSource2: TDataSource;  
DBLookupComboBox1: TDBLookupComboBox;  
DBLookupListBox1: TDBLookupListBox;  
Image1: TImage;



```

procedure Button1Click(Sender: TObject);
procedure Edit1Change(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure q2AfterScroll(DataSet: TDataSet);
procedure Exit1Click(Sender: TObject);
procedure Help1Click(Sender: TObject);
procedure About1Click(Sender: TObject);
procedure DBLookupComboBox1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FMenuPengguna: TFMMenuPengguna;

implementation

uses Uabout, Helppengguna;

{$R *.dfm}

procedure TFMMenuPengguna.Button1Click(Sender: TObject);
begin
close;
end;

procedure TFMMenuPengguna.Edit1Change(Sender: TObject);
var strsql : string;
begin
//if edit1.Text ='' then
strsql:='SELECT      Cacing.ID_Cacing,      Cacing.[Nama      Cacing],
Cacing.ID_Penyakit, Penyakit.[Nama Penyakit], Cacing.ID_Obat, Obat.[Nama
Obat], Cacing.ID_Kategori, Kategori.[Nama Kategori], Cacing.[Gambar Cacing],
Cacing.[Siklus Hidup], Cacing.[Sound Tutorial]'
+' FROM Penyakit INNER JOIN (Obat INNER JOIN (Kategori INNER JOIN
Cacing ON Kategori.ID_Kategori = Cacing.ID_Kategori) ON Obat.ID_obat =
Cacing.ID_Obat) ON Penyakit.ID_Penyakit = Cacing.ID_Penyakit'
//else
//strsql:='SELECT      Cacing.ID_Cacing,      Cacing.[Nama      Cacing],
Cacing.ID_Penyakit, Penyakit.[Nama Penyakit], Cacing.ID_Obat, Obat.[Nama
Obat], Cacing.ID_Kategori, Kategori.[Nama Kategori], Cacing.[Gambar Cacing],
Cacing.[Siklus Hidup], Cacing.[Sound Tutorial]'

```

```
//+' FROM Penyakit INNER JOIN (Obat INNER JOIN (Kategori INNER JOIN  
Cacing ON Kategori.ID_Kategori = Cacing.ID_Kategori) ON Obat.ID_obat =  
Cacing.ID_Obat) ON Penyakit.ID_Penyakit = Cacing.ID_Penyakit'  
//+' WHERE Cacing.[Nama Cacing]= :nm';
```

```
////q2.Close;  
//q2.SQL.clear;  
//q2.sql.Add(strsql);  
//if edit1.Text <> "" then  
//begin  
//q2.Prepared;  
//q2.Parameters.ParamByName('nm').Value:=edit1.Text;  
//end;  
//q2.Open;  
end;
```

```
procedure TFMenuPengguna.Button3Click(Sender: TObject);  
begin  
q2.Close;  
q2.Open;  
end;
```

```
procedure TFMenuPengguna.q2AfterScroll(DataSet: TDataSet);  
begin  
if q2.Fields[10].AsString <> "" then  
Begin  
mediaplayer1.Close;  
mediaplayer1.FileName:=q2.Fields[10].AsString;  
mediaplayer1.open;  
end;  
//showmessage(q2.Fields[9].AsString);  
if q2.Fields[9].AsString <> "" then  
image1.Picture.LoadFromFile(q2.Fields[9].AsString);  
if q2.Fields[8].AsString <> "" then  
image2.Picture.loadfromfile(q2.Fields[8].AsString);  
  
end;
```

```
procedure TFMenuPengguna.Exit1Click(Sender: TObject);  
begin  
application.Terminate;  
end;
```

```
procedure TFMenuPengguna.Help1Click(Sender: TObject);
```



```

begin
bantuanpengguna.show;
end;

procedure TFMenuPengguna.About1Click(Sender: TObject);
begin
aboutbox.show;
end;

procedure TFMenuPengguna.DBLookupComboBox1Click(Sender: TObject);
var strsql : string;
begin
q2.Close;
if dblookupcombobox1.Text ='' then
strsql:='SELECT      Cacing.ID_Cacing,      Cacing.[Nama      Cacing],
Cacing.ID_Penyakit, Penyakit.[Nama Penyakit], Cacing.ID_Obat, Obat.[Nama
Obat], Cacing.ID_Kategori, Kategori.[Nama Kategori], Cacing.[Gambar Cacing],
Cacing.[Siklus Hidup], Cacing.[Sound Tutorial]'
+' ,penyakit.solusi FROM Penyakit INNER JOIN (Obat INNER JOIN (Kategori
INNER JOIN Cacing ON Kategori.ID_Kategori = Cacing.ID_Kategori) ON
Obat.ID_obat = Cacing.ID_Obat) ON Penyakit.ID_Penyakit =
Cacing.ID_Penyakit'
else
strsql:='SELECT      Cacing.ID_Cacing,      Cacing.[Nama      Cacing],
Cacing.ID_Penyakit, Penyakit.[Nama Penyakit], Cacing.ID_Obat, Obat.[Nama
Obat], Cacing.ID_Kategori, Kategori.[Nama Kategori], Cacing.[Gambar Cacing],
Cacing.[Siklus Hidup], Cacing.[Sound Tutorial]'
+' ,penyakit.solusi FROM Penyakit INNER JOIN (Obat INNER JOIN (Kategori
INNER JOIN Cacing ON Kategori.ID_Kategori = Cacing.ID_Kategori) ON
Obat.ID_obat = Cacing.ID_Obat) ON Penyakit.ID_Penyakit =
Cacing.ID_Penyakit'
+' WHERE Cacing.[id_kategori]= :nm';

q2.Close;
q2.SQL.clear;
q2.sql.Add(strsql);
if dblookupcombobox1.Text <>'' then
begin
q2.Prepared;
q2.Parameters.ParamByName('nm').Value:=dblookupcombobox1.Text;
end;
q2.Open;
end;

end.

```

```

unit UMenuPetugas;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls;

type
  TFMenuPetugas = class(TForm)
    Button1: TButton;
    MainMenu1: TMainMenu;
    InputData1: TMenuItem;
    Cacing1: TMenuItem;
    Penyakit1: TMenuItem;
    Obat1: TMenuItem;
    Kategori1: TMenuItem;
    EditData1: TMenuItem;
    EntryPassword1: TMenuItem;
    Help: TMenuItem;
    procedure Penyakit1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Kategori1Click(Sender: TObject);
    procedure Obat1Click(Sender: TObject);
    procedure Cacing1Click(Sender: TObject);
    procedure EditData1Click(Sender: TObject);
    procedure EntryPassword1Click(Sender: TObject);
    procedure HelpClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FMenuPetugas: TFMenuPetugas;

implementation

uses UDM, Ukategoricacing, UObat, Ucacing, Ueditcacing, UPengguna, HALp;

{$R *.dfm}

procedure TFMenuPetugas.Penyakit1Click(Sender: TObject);
begin
  forminputpenyakit.show;

```

end;

```
procedure TFMenuPetugas.Button1Click(Sender: TObject);  
begin  
close;  
end;
```

```
procedure TFMenuPetugas.Kategori1Click(Sender: TObject);  
begin  
forminputkategori.show;  
end;
```

```
procedure TFMenuPetugas.Obat1Click(Sender: TObject);  
begin  
Forminputobat.show;  
end;
```

```
procedure TFMenuPetugas.Cacing1Click(Sender: TObject);  
begin  
forminputcacing.show;  
end;
```

```
procedure TFMenuPetugas.EditData1Click(Sender: TObject);  
begin  
formedit.show;  
end;
```

```
procedure TFMenuPetugas.EntryPassword1Click(Sender: TObject);  
begin  
forminputpengguna.show;  
end;
```

```
procedure TFMenuPetugas.HelpClick(Sender: TObject);  
begin  
bantuan.show;  
end;
```

end.

unit Ueditcacing;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, strutils,DBCtrls, DB, ADODB, StdCtrls, Mask, ExtCtrls, Grids,  
DBGrids,  
ComCtrls;

type

```
TFormEdit = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  TabSheet4: TTabSheet;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Bevel1: TBevel;
  Bevel2: TBevel;
  DBGrid1: TDBGrid;
  DBNavigator1: TDBNavigator;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  EditC: TButton;
  SimpanC: TButton;
  BatalC: TButton;
  HapusC: TButton;
  DBEdit6: TDBEdit;
  DBEdit7: TDBEdit;
  DBLookupComboBox1: TDBLookupComboBox;
  DBLookupComboBox2: TDBLookupComboBox;
  DBLookupComboBox3: TDBLookupComboBox;
  Button2: TButton;
  Button3: TButton;
  DBEdit8: TDBEdit;
  Button4: TButton;
  OpenFileDialog1: TOpenDialog;
  OpenFileDialog3: TOpenDialog;
  OpenFileDialog2: TOpenDialog;
  Label10: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  DBGrid2: TDBGrid;
```

```
DBNavigator2: TDBNavigator;
DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
EditK: TButton;
SimpanK: TButton;
BatalK: TButton;
HapusK: TButton;
Q1: TADOQuery;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
DBGrid3: TDBGrid;
DBNavigator3: TDBNavigator;
DBEdit5: TDBEdit;
DBEdit9: TDBEdit;
EditO: TButton;
SimpanO: TButton;
BatalO: TButton;
HapusO: TButton;
ADOQuery1: TADOQuery;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
DBGrid4: TDBGrid;
DBNavigator4: TDBNavigator;
DBEdit10: TDBEdit;
DBEdit11: TDBEdit;
DBMemo1: TDBMemo;
EditP: TButton;
SimpanP: TButton;
BatalP: TButton;
HapusP: TButton;
ADOQuery2: TADOQuery;
Button5: TButton;
procedure EditCClick(Sender: TObject);
procedure SimpanCClick(Sender: TObject);
procedure BatalCClick(Sender: TObject);
procedure keluarClick(Sender: TObject);
procedure HapusCClick(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure EditKClick(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure EditOClick(Sender: TObject);
```

```

procedure EditPClick(Sender: TObject);
procedure SimpanPClick(Sender: TObject);
procedure SimpanOClick(Sender: TObject);
procedure SimpanKClick(Sender: TObject);
procedure BatalPClick(Sender: TObject);
procedure BatalOClick(Sender: TObject);
procedure BatalKClick(Sender: TObject);
procedure HapusPClick(Sender: TObject);
procedure HapusOClick(Sender: TObject);
procedure HapusKClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormEdit: TFormEdit;

implementation

uses Udatamodul;

{$R *.dfm}

procedure TFormEdit.EditCClick(Sender: TObject);
var nomor : string;

begin
dm.tcacing.edit;

editC.enabled:=false;
simpanC.enabled:=true;
batalC.enabled:=true;

end;

procedure TFormEdit.SimpanCClick(Sender: TObject);
begin
dm.tcacing.post;
editC.enabled:=true;
simpanC.enabled:=false;
batalC.enabled:=false;

end;

```

```
procedure TFormEdit.BatalCClick(Sender: TObject);
begin
dm.tcacing.cancel;
editC.enabled:=true;
simpanC.enabled:=false;
batalC.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.keluarClick(Sender: TObject);
begin
close
end;
```

```
procedure TFormEdit.HapusCClick(Sender: TObject);
begin
dm.tcacing.delete;
end;
```

```
procedure TFormEdit.Button2Click(Sender: TObject);
begin
if opendirialog1.Execute then
dbedit6.Text:=opendirialog1.FileName;
end;
```

```
procedure TFormEdit.Button3Click(Sender: TObject);
begin
if opendirialog2.Execute then
dbedit7.Text:=opendirialog2.FileName;
end;
```

```
procedure TFormEdit.Button4Click(Sender: TObject);
begin
if opendirialog3.Execute then
dbedit8.Text:=opendirialog3.FileName;
end;
```

```
procedure TFormEdit.EditKClick(Sender: TObject);
begin
dm.tkategori.edit;
```

```
editK.enabled:=false;
simpanK.enabled:=true;
batalK.enabled:=true;
```

```
end;
```

```
procedure TFormEdit.Button5Click(Sender: TObject);  
begin  
close;  
end;
```

```
procedure TFormEdit.EditOClick(Sender: TObject);  
begin  
dm.tobat.edit;
```

```
edito.enabled:=false;  
simpano.enabled:=true;  
batalo.enabled:=true;
```

```
end;
```

```
procedure TFormEdit.EditPClick(Sender: TObject);  
begin  
dm.tpenyakit.edit;  
editP.enabled:=false;  
simpanP.enabled:=true;  
batalP.enabled:=true;
```

```
end;
```

```
procedure TFormEdit.SimpanPClick(Sender: TObject);  
begin  
dm.tpenyakit.post;  
editp.enabled:=true;  
simpanp.enabled:=false;  
batalp.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.SimpanOClick(Sender: TObject);  
begin  
dm.tobat.post;  
edito.enabled:=true;  
simpano.enabled:=false;  
batalo.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.SimpanKClick(Sender: TObject);  
begin  
dm.tkategori.post;
```



```
editk.enabled:=true;  
simpank.enabled:=false;  
bataalk.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.BatalPClick(Sender: TObject);  
begin  
dm.tpenyakit.cancel;  
editp.enabled:=true;  
simpanp.enabled:=false;  
batalp.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.BatalOClick(Sender: TObject);  
begin  
dm.tobat.cancel;  
edito.enabled:=true;  
simpano.enabled:=false;  
batalo.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.BatalKClick(Sender: TObject);  
begin  
dm.tkategori.cancel;  
editk.enabled:=true;  
simpank.enabled:=false;  
bataalk.enabled:=false;
```

```
end;
```

```
procedure TFormEdit.HapusPClick(Sender: TObject);  
begin  
dm.tpenyakit.delete;  
end;
```

```
procedure TFormEdit.HapusOClick(Sender: TObject);  
begin  
dm.tobat.delete;  
end;
```

```
procedure TFormEdit.HapusKClick(Sender: TObject);  
begin  
dm.tkategori.delete;
```

end;

end.

unit Ucacing;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, strutils, DBCtrls, DB, ADODB, StdCtrls, Mask, ExtCtrls, Grids,  
DBGrids;

type

TFormInputCacing = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

DBGrid1: TDBGrid;

DBNavigator1: TDBNavigator;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

tambah: TButton;

simpan: TButton;

batal: TButton;

keluar: TButton;

Q1: TADOQuery;

Button1: TButton;

DBEdit6: TDBEdit;

DBEdit7: TDBEdit;

DBLookupComboBox1: TDBLookupComboBox;

DBLookupComboBox2: TDBLookupComboBox;

DBLookupComboBox3: TDBLookupComboBox;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Button2: TButton;

Button3: TButton;

DBEdit8: TDBEdit;

Button4: TButton;

Label9: TLabel;

OpenDialog1: TOpenDialog;

OpenDialog3: TOpenDialog;

```

OpenDialog2: TOpenDialog;
Bevel1: TBevel;
Bevel2: TBevel;
procedure tambahClick(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure batalClick(Sender: TObject);
procedure keluarClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormInputCacing: TFormInputCacing;

implementation

uses Udatamodul;

{$R *.dfm}

procedure TFormInputCacing.tambahClick(Sender: TObject);
var nomor : string;

begin
  q1.Close;
  q1.sql.Clear;
  q1.SQL.add('select max(id_cacing) from cacing');
  q1.open;
  if q1.Fields[0].asstring="" then nomor:='C001'
  else
  begin
    nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
    if length(nomor)=1 then
      nomor:='C00'+nomor;
    if length(nomor)=2 then
      nomor:='C0'+nomor;
    if length(nomor)=3 then
      nomor:='C'+nomor;
  end;
end;

```

```
dm.tcacing.insert;
dbedit1.Text:=nomor;
tambah.enabled:=false;
simpan.enabled:=true;
batal.enabled:=true;
keluar.enabled:=false;
end;
```

```
procedure TFormInputCacing.simpanClick(Sender: TObject);
begin
dm.tcacing.post;
tambah.enabled:=true;
simpan.enabled:=false;
batal.enabled:=false;
keluar.enabled:=true;
end;
```

```
procedure TFormInputCacing.batalClick(Sender: TObject);
begin
dm.tcacing.cancel;
tambah.enabled:=true;
simpan.enabled:=false;
batal.enabled:=false;
keluar.enabled:=true;
end;
```

```
procedure TFormInputCacing.keluarClick(Sender: TObject);
begin
close
end;
```

```
procedure TFormInputCacing.Button1Click(Sender: TObject);
begin
dm.tcacing.delete;
end;
```

```
procedure TFormInputCacing.Button2Click(Sender: TObject);
begin
if opendirialog1.Execute then
dbedit6.Text:=opendirialog1.FileName;
end;
```

```
procedure TFormInputCacing.Button3Click(Sender: TObject);
begin
if opendirialog2.Execute then
dbedit7.Text:=opendirialog2.FileName;
```

end;

```
procedure TFormInputCacing.Button4Click(Sender: TObject);  
begin  
if opendialog3.Execute then  
dbedit8.Text:=opendialog3.FileName;  
end;
```

end.

unit Ukategoricacing;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, strutils, StdCtrls, DBCtrls, Mask, ExtCtrls, Grids, DBGrids, DB,  
ADODB;

type

```
TFormInputKategori = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
DBGrid1: TDBGrid;  
DBNavigator1: TDBNavigator;  
DBEdit1: TDBEdit;  
DBEdit2: TDBEdit;  
tambah: TButton;  
simpan: TButton;  
batal: TButton;  
keluar: TButton;  
Q1: TADOQuery;  
Button1: TButton;  
procedure tambahClick(Sender: TObject);  
procedure keluarClick(Sender: TObject);  
procedure simpanClick(Sender: TObject);  
procedure batalClick(Sender: TObject);  
procedure Button1Click(Sender: TObject);  
private  
{ Private declarations }  
public  
{ Public declarations }  
end;
```

```

var
  FormInputKategori: TFormInputKategori;

implementation

uses Udatamodul;

{$R *.dfm}

procedure TFormInputKategori.tambahClick(Sender: TObject);
var nomor : string;

begin
  q1.Close;
  q1.sql.Clear;
  q1.SQL.add('select max(id_kategori) from kategori');
  q1.open;
  if q1.Fields[0].AsString="" then nomor:='K001'
  else
  begin
    nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
    if length(nomor)=1 then
      nomor:='K00'+nomor;
    if length(nomor)=2 then
      nomor:='K0'+nomor;
    if length(nomor)=3 then
      nomor:='K'+nomor;
    end;

    dm.tkategori.insert;
    dbedit1.Text:=nomor;
    tambah.enabled:=false;
    simpan.enabled:=true;
    batal.enabled:=true;
    keluar.enabled:=false;
  end;

  procedure TFormInputKategori.keluarClick(Sender: TObject);
  begin
    close;
  end;

  procedure TFormInputKategori.simpanClick(Sender: TObject);
  begin
    dm.tkategori.post;
    tambah.enabled:=true;

```

```
simpan.enabled:=false;  
batal.enabled:=false;  
keluar.enabled:=true;  
end;
```

```
procedure TFormInputKategori.batalClick(Sender: TObject);  
begin  
dm.tkategori.cancel;  
tambah.enabled:=true;  
simpan.enabled:=false;  
batal.enabled:=false;  
keluar.enabled:=true;  
end;
```

```
procedure TFormInputKategori.Button1Click(Sender: TObject);  
begin  
dm.Tkategori.Delete;  
end;  
  
end.
```

```
unit UObat;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, strutils, StdCtrls, DBCtrls, Mask, ExtCtrls, Grids, DBGrids, DB,  
ADOODB;
```

```
type
```

```
TFormInputObat = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
DBGrid1: TDBGrid;  
DBNavigator1: TDBNavigator;  
DBEdit1: TDBEdit;  
DBEdit2: TDBEdit;  
tambah: TButton;  
simpan: TButton;  
batal: TButton;  
keluar: TButton;  
Q1: TADOQuery;  
Button1: TButton;
```

```

procedure tambahClick(Sender: TObject);
procedure keluarClick(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure batalClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormInputObat: TFormInputObat;

implementation

uses Udatamodul;

{$R *.dfm}

procedure TFormInputObat.tambahClick(Sender: TObject);
var nomor : string;

begin
  q1.Close;
  q1.sql.Clear;
  q1.SQL.add('select max(id_obat) from obat');
  q1.open;
  if q1.Fields[0].AsString="" then nomor:='B001'
  else
  begin
    nomor:=inttostr(strtoint(midstr(q1.Fields[0].AsString,2,3))+1);
    if length(nomor)=1 then
      nomor:='B00'+nomor;
    if length(nomor)=2 then
      nomor:='B0'+nomor;
    if length(nomor)=3 then
      nomor:='B'+nomor;
  end;

  dm.tobat.insert;
  dbedit1.Text:=nomor;
  tambah.enabled:=false;
  simpan.enabled:=true;
  batal.enabled:=true;
  keluar.enabled:=false;

```



```

end;

procedure TFormInputObat.keluarClick(Sender: TObject);
begin
close;
end;

procedure TFormInputObat.simpanClick(Sender: TObject);
begin
dm.tobat.post;
tambah.enabled:=true;
simpan.enabled:=false;
batal.enabled:=false;
keluar.enabled:=true;
end;

procedure TFormInputObat.batalClick(Sender: TObject);
begin
dm.tobat.cancel;
tambah.enabled:=true;
simpan.enabled:=false;
batal.enabled:=false;
keluar.enabled:=true;
end;

procedure TFormInputObat.Button1Click(Sender: TObject);
begin
dm.Tobat.Delete;
end;

end.

```

```

unit UPengguna;

```

```

interface

```

```

uses

```

```

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, strutils, StdCtrls, DBCtrls, Mask, ExtCtrls, Grids, DBGrids, DB,
  ADODB;

```

```

type

```

```

  TFormInputPengguna = class(TForm)
    Label1: TLabel;

```

```

Label2: TLabel;
Label3: TLabel;
DBGrid1: TDBGrid;
DBNavigator1: TDBNavigator;
DBEdit1: TDBEdit;
DBEdit2: TDBEdit;
tambah: TButton;
simpan: TButton;
batal: TButton;
keluar: TButton;
edit: TButton;
DBEdit3: TDBEdit;
Label4: TLabel;
Button2: TButton;
procedure tambahClick(Sender: TObject);
procedure keluarClick(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure batalClick(Sender: TObject);
procedure editClick(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormInputPengguna: TFormInputPengguna;

implementation

uses Udatamodul;

{$R *.dfm}

procedure TFormInputPengguna.tambahClick(Sender: TObject);
var nomor : string;

begin

dm.tpengguna.insert;

tambah.enabled:=false;
simpan.enabled:=true;
batal.enabled:=true;
keluar.enabled:=false;

```

```
edit.enabled:=false;  
end;
```

```
procedure TFormInputPegguna.keluarClick(Sender: TObject);  
begin  
close;  
end;
```

```
procedure TFormInputPegguna.simpanClick(Sender: TObject);  
begin  
dm.tpegguna.post;  
tambah.enabled:=true;  
simpan.enabled:=false;  
batal.enabled:=false;  
keluar.enabled:=true;  
edit.enabled:=true;  
end;
```

```
procedure TFormInputPegguna.batalClick(Sender: TObject);  
begin  
dm.tobat.cancel;  
tambah.enabled:=true;  
simpan.enabled:=false;  
batal.enabled:=false;  
keluar.enabled:=true;  
edit.enabled:=true;  
end;
```

```
procedure TFormInputPegguna.editClick(Sender: TObject);  
begin  
    dm.Tobat.edit;  
    tambah.enabled:=false;  
simpan.enabled:=true;  
batal.enabled:=true;  
keluar.enabled:=false;  
edit.enabled:=false;  
end;
```

```
procedure TFormInputPegguna.Button2Click(Sender: TObject);  
begin  
    dm.Tobat.Delete;  
end;
```

```
end.
```

```

unit Uregister;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB;

type
  TFRegister = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Button2: TButton;
    q1: TADOQuery;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FRegister: TFRegister;

implementation

uses UMenuPetugas;

{$R *.dfm}

procedure TFRegister.Button1Click(Sender: TObject);
begin
  if (edit1.text<>") and (edit2.text<>") then
  begin
    q1.Close;
    q1.sql.Clear;
    q1.SQL.add('select count(username) from pengguna where username=:us and
    password=:pw');
    q1.Prepared;
    q1.Parameters.ParamByName('us').Value:=edit1.Text;
    q1.Parameters.parambyname('pw').Value:=edit2.Text;
    q1.Open;
    if q1.Fields[0].asinteger>0 then

```

```
begin
showmessage('Selamat memasuki sistem');
fregister.Close;
fmenupetugas.show;
end
else
showmessage('Username/password salah');
end
else
showmessage('Isian username/password kosong');
end;

end.
```

```
unit User;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ExtCtrls;
```

```
type
```

```
TFuser = class(TForm)
RadioGroup1: TRadioGroup;
Button1: TButton;
Button2: TButton;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
```

```
var
```

```
Fuser: TFuser;
```

```
implementation
```

```
uses UMenuPetugas, Menupengguna, Uregister;
```

```
{ $R *.dfm }
```

```
procedure TFuser.Button1Click(Sender: TObject);  
begin  
if radiobutton1.Checked then fregister.show;  
if radiobutton2.Checked then fmenupengguna.show;
```

```
end;
```

```
procedure TFuser.Button2Click(Sender: TObject);  
begin  
application.Terminate;  
end;
```

```
end.
```

```
unit Unitutama;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls, jpeg;
```

```
type
```

```
Tsplash = class(TForm)  
Label1: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Button1: TButton;  
Button2: TButton;  
Image1: TImage;  
procedure Button2Click(Sender: TObject);  
procedure Button1Click(Sender: TObject);  
private  
{ Private declarations }  
public  
{ Public declarations }  
end;
```

```
var
```

```
splash: Tsplash;
```

```
implementation
```

```
uses Umenu, UMenuPetugas, User;
```

```
{$R *.dfm}
```

```
procedure Tsplash.Button2Click(Sender: TObject);  
begin  
  application.Terminate;  
end;
```

```
procedure Tsplash.Button1Click(Sender: TObject);  
begin  
  fuser.showmodal;  
  splash.Close;  
end;
```

```
end.
```

