

**PEMBUATAN PROGRAM SISTEM INFORMASI PADA
PETERNAKAN BURUNG PERKUTUT 'HMT FARM'**



Oleh :

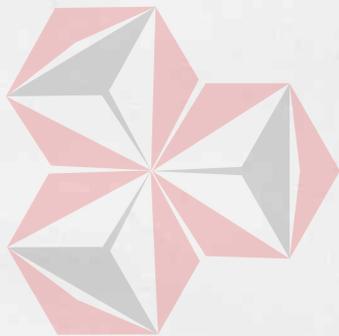
Nama : HENDRY TJANDRA
NIM : 94.41010.6001
Program : S1 (Strata Satu)
Jurusan : Manajemen Informatika

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA
2002**

**PEMBUATAN PROGRAM SISTEM INFORMASI PADA
PETERNAKAN BURUNG PERKUTUT 'HMT FARM'**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Komputer**



**UNIVERSITAS
Dinamika**

Oleh :

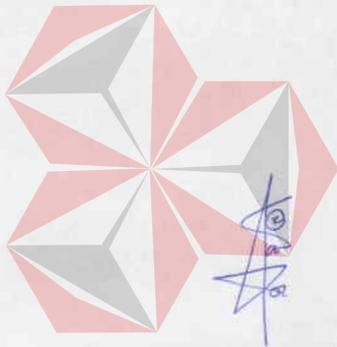
**Nama : HENDRY TJANDRA
NIM : 94.41010.6001
Program : S1 (Strata Satu)
Jurusan : Manajemen Informatika**

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA
2002**

**PEMBUATAN PROGRAM SISTEM INFORMASI PADA
PETERNAKAN BURUNG PERKUTUT 'HMT FARM'**

Telah diperiksa, diuji dan disetujui

Surabaya, Maret 2002



Januar Wibowo, ST, MM
Dosen Pembimbing II

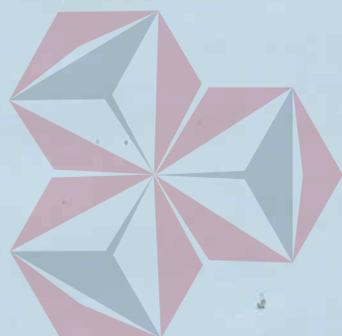
Menyetujui :

Dra. MJ Dewiyani S, MPd
Dosen Pembimbing I

Mengetahui :



Haryanto Tanuwijaya, S.Kom, M.MT
Pembantu Ketua I



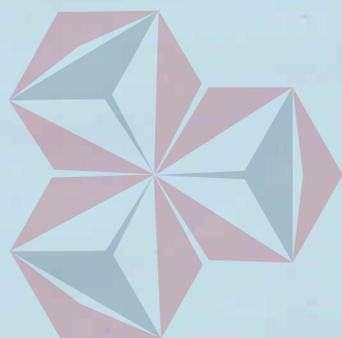
UNIVERSITAS
Dinamika

ABSTRAKSI

ABSTRAKSI

HMT Farm merupakan salah satu peternakan burung perkutut mempunyai nilai ekonomis yang tinggi . Hasil peternakan tersebut merupakan perkembangan dari burung unggulan tersebut yang juga mempunyai nilai ekonomis yang tinggi. Untuk itu setiap peternakan burung akan mempunyai suatu sistem yang menyediakan data dari masing-masing burung perkutut. Selama ini mereka menggunakan kartu untuk masing-masing burung perkutut. Ketika peternakan ini berkembang dengan pesat maka data ini akan berjumlah ribuan sehingga diperlukan sistem informasi yang baru untuk menangani masalah tersebut. Karena sistem dengan menggunakan kartu akan memerlukan waktu yang lama untuk mencari silsilah burung tersebut dalam membuat sebuah sertifikat. Proses penjualan seekor perkutut yang telah menjadi juara akan juga mempengaruhi nilai jual dari perkutut-perkutut lain yang dilahirkan oleh pasangan yang sama sehingga diperlukan informasi tentang banyaknya anak dari sepasang perkutut tersebut.

Dalam pembuatan sertifikat pada sistem masih belum terkomputerisasi dilakukan dengan mencari data dari orang tua burung dengan cara manual. Cara ini sangat memungkinkan jika jumlah masih sedikit tetapi kalau sudah banyak maka akan memerlukan waktu yang sangat lama dan hampir tidak mungkin dilakukan. Selain itu juga proses untuk mencari jumlah anak dari sepasang perkutut akan terasa sulit dan menyita waktu. Untuk memecahkan masalah ini dibuat suatu sistem komputer yang dapat memproses sertifikat dengan cepat dan efisien.



UNIVERSITAS
Dinamika

KATA PENGANTAR

KATA PENGANTAR

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa yang telah memberikan berkat dan rahmatnya sehingga dapat menyelesaikan Tugas Akhir yang merupakan persyaratan yang wajib dipenuhi oleh mahasiswa dan mahasiswi jurusan Manajemen Informatika pada program Strata Satu di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya.

Tugas Akhir ini membahas Masalah Pembuatan Sistem Informasi Pada Peternakan Burung Perkutut 'HMT Farm'. Dalam menyelesaikan Tugas Akhir ini penulis telah berusaha sebaik mungkin. Penulis juga menyadari akan terbatasnya waktu, kemampuan serta pengalaman yang dimiliki. Oleh karena itu, penulis bersedia menerima kritik dan saran yang sangat bermanfaat dalam penyempurnaan Tugas Akhir ini.

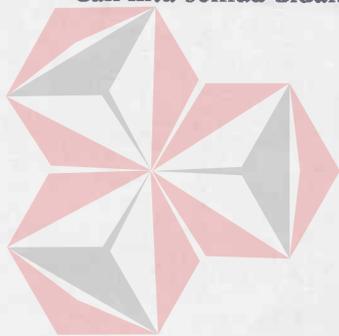
Penulis pada kesempatan ini, menyampaikan ucapan terima kasih kepada yang terhormat :

1. Bapak Ir. Ronny S. Susilo, MM selaku Ketua STIKOM yang telah memberikan kesempatan pada penulis untuk menuntut ilmu di STIKOM.
2. Bapak Haryanto Tanuwijaya, S. Kom, M.MT selaku pembantu Ketua 1 yang telah membantu kelancaran dalam penyusunan tugas akhir ini
3. Bapak Januar Wibowo, ST, MM, selaku dosen pembimbing II, yang telah banyak memberi pengarahan dan petunjuk yang sangat berguna dalam penyusunan Tugas akhir ini.
4. Ibu Dra.M.J Dewiyani S, Mpd selaku dosen pembimbing I, yang telah memberikan pengarahan dan petunjuk dalam penyusunan tugas akhir ini.

5. Bapak Wahyu, selaku koordinator tugas akhir, yang telah banyak membantu dalam peny. Seluruh staff dan karyawan
6. Peternakan HMT Farm, yang telah memberikan waktu dan bantuan dalam menyelesaikan penyusunan tugas akhir ini.
7. Orang Tua saya dan saudara serta teman-teman yang juga memberi dukungan dalam penyelesaian Tugas akhir ini.

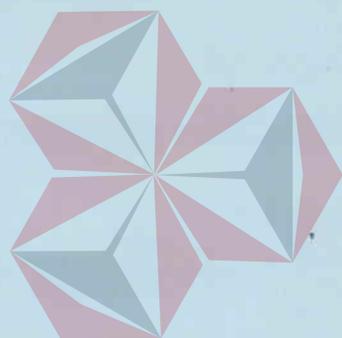
Semoga Tuhan Yang Maha Esa membalas budi baik kepada semua pihak tersebut diatas.

Harapan penulis, semoga Tugas akhir ini dapat berguna bagi para pembaca dan kita semua didalam melaksanakan tugas dimasa yang akan datang.



UNIVERSITAS
Dinamika
Surabaya, Maret 2002

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

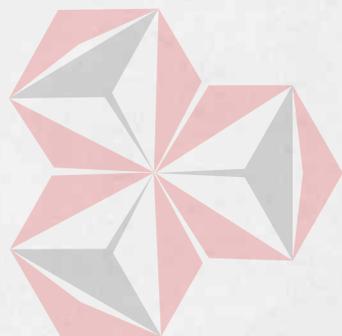
DAFTAR ISI

	Halaman
ABSTRAKSI	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
BAB I. PENDAHULUAN	1
1.1. Latar Belakang Permasalahan	1
1.2. Perumusan Masalah	2
1.3. Pembatasan Masalah	2
1.4. Maksud Dan Tujuan	3
1.5. Metodologi Pemecahan Masalah.....	3
1.6. Sistematika Penulisan Tugas Akhir	4
BAB II. LANDASAN TEORI	5
2.1. Produksi	5
2.2. Penjualan	5
2.3. Kartu Burung.....	6
2.4. Sertifikat Burung.....	6
2.5. Konsep Dasar Informasi.....	6
2.6. Basis Data.....	7
2.6.1. Teori Basis Data.....	7
2.6.2. Kegunaan Database.....	10
2.6.3. Merancang Model Konseptual Database.....	12

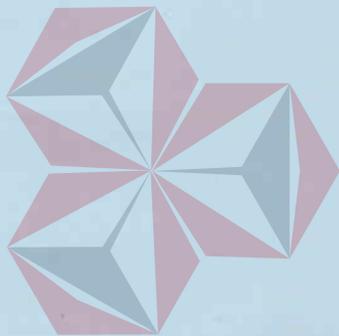
2.6.4. Perancangan Sistem.....	19
2.7. Rational Database.....	20
2.8. Pseudocode.....	21
2.9. Bentuk Program.....	21
2.9.1. Program Primer.....	22
2.9.2. Program Sekunder.....	22
2.10. Konsep Dasar Program.....	24
2.11. Prosedur Dan Fungsi.....	27
2.12. Percabangan Dan Pengulangan.....	28
2.12.1 Percabangan.....	28
2.12.2. Pengulangan.....	29
2.13 Block Dan Scope.....	30
2.14 Penentuan Type Data Baru.....	31
2.15 Type Data Abstrak.....	33
2.16 Type Data Abstrak Dasar.....	34
BAB III. PERMASALAHAN	37
BAB IV. ANALISA PERMASALAHAN	39
4.1. Analisa Permasalahan.....	39
4.2. Data Flow Diagram (DFD).....	40
4.3. Entity Relationship Diagram.....	43
4.4. Pembuatan Database.....	43
4.5. Perancangan Perangkat Lunak.....	44
4.6. Sistem Menu.....	50
4.7. Panduan Untuk Menjalankan Program.....	58

BAB V. PENUTUP	66
5.1. Kesimpulan	66
5.2. Saran	66

DAFTAR PUSTAKA



UNIVERSITAS
Dinamika



UNIVERSITAS
Dinamika

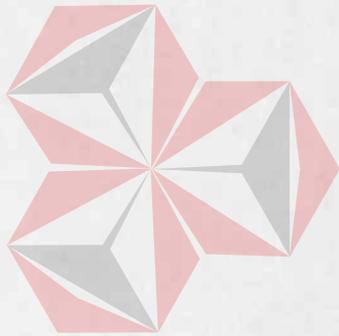


DAFTAR GAMBAR

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Hubungan Satu Lawan Satu	16
Gambar 2.2. Hubungan Satu Lawan Banyak	16
Gambar 2.3. Hubungan Banyak Lawan Banyak	17
Gambar 2.4. Bagan Relasi Satu Lawan Satu	17
Gambar 2.5. Bagan Relasi Banyak Lawan Satu	18
Gambar 2.6. Bagan Relasi Banyak Lawan Banyak	18
Gambar 4.1. Blok Sistem Diagram Data Burung	39
Gambar 4.2. Context Diagram.....	40
Gambar 4.3. DFD Bagan Berjenjang	41
Gambar 4.4. DFD Level 1 Proses 1.....	41
Gambar 4.5. DFD Level 1 Proses 3.....	42
Gambar 4.6. Overview Diagram.....	42
Gambar 4.7. Entity Relationship.....	43
Gambar 4.8. Pohon Biner Silsilah Garis Keturunan Burung Perkutut.....	44
Gambar 4.9. Diagram Alur Program Utama.....	47
Gambar 4.10. Diagram Alur Sub Menu Identitas Burung.....	47
Gambar 4.11. Diagram Alur Sub Menu Silsilah Burung.....	48
Gambar 4.12. Diagram Alur Sub Menu Inventori.....	48
Gambar 4.13. Diagram Alur Sub Menu Laporan Burung.....	49
Gambar 4.14. Diagram Alur Sub Menu Sertifikat	49
Gambar 4.15. Menu Utama.....	51

Gambar 4.16. Menu Identitas Burung.....	52
Gambar 4.17. Menu Silsilah Burung.....	54
Gambar 4.18. Menu Inventori Burung.....	55
Gambar 4.19. Menu Laporan Burung Yang Belum Terjual.....	56
Gambar 4.20. Menu Laporan Burung Yang Terjual.....	56
Gambar 4.21. Hasil Print Sertifikat.....	57
Gambar 4.22. Menu Sertifikat Burung.....	58



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1. Latar Belakang Permasalahan

Menjamurnya lomba-lomba burung memacu perkembangan peternakan burung. Banyak orang berlomba-lomba menernakkan burung yang telah menjadi juara karena burung yang telah juara mempunyai nilai ekonomis yang sangat tinggi, demikian juga dengan anak hasil perkembang biakan dari burung unggulan tersebut juga mempunyai nilai ekonomis yang tinggi.

Berdasarkan hal tersebut maka setiap kelahiran seekor burung perkutut akan dibuatkan sebuah sertifikat yang berisi kode burung kemudian orang tua burung serta kakek nenek dari burung tersebut. Ini sangat perlu karena nilai ekonomis seekor burung perkutut berdasarkan dari garis keturunan dari burung tersebut.

Untuk itu setiap peternakan burung akan mempunyai suatu sistem yang menyediakan data dari masing-masing burung perkutut. Selama ini mereka menggunakan kartu untuk masing-masing burung perkutut. Ketika peternakan ini berkembang dengan pesat maka data ini akan berjumlah ribuan sehingga diperlukan sistem informasi yang baru untuk menangani masalah tersebut. Karena sistem dengan menggunakan kartu akan memerlukan waktu yang lama untuk mencari silsilah burung tersebut dalam membuat sebuah sertifikat.

Proses penjualan seekor perkutut yang telah menjadi juara akan juga mempengaruhi nilai jual dari perkutut-perkutut lain yang dilahirkan oleh pasangan yang sama sehingga diperlukan informasi tentang banyaknya anak dari sepasang perkutut tersebut.

HMT Farm merupakan salah satu peternakan burung perkutut mempunyai nilai ekonomis yang tinggi . Hasil peternakan tersebut merupakan perkembangan dari burung unggulan tersebut yang juga mempunyai nilai ekonomis yang tinggi. Dalam pembuatan sertifikat pada sistem masih belum terkomputerisasi dilakukan dengan mencari data dari orang tua burung dengan cara manual. Cara ini sangat memungkinkan jika jumlah masih sedikit tetapi kalau sudah banyak maka akan memerlukan waktu yang sangat lama dan hampir tidak mungkin dilakukan. Selain itu juga proses untuk mencari jumlah anak dari sepasang perkutut akan terasa sulit dan menyita waktu. Untuk memecahkan masalah ini dibuat suatu sistem komputer yang dapat memproses sertifikat dengan cepat dan efisien.

1.2. Perumusan Masalah

Dari latar belakang permasalahan yang telah diuraikan diatas dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana dapat meningkatkan ketepatan dan kecepatan dalam penyajian informasi tentang data identitas burung yang dibutuhkan ?
2. Bagaimana pemilik peternakan burung dapat membuat sertifikat dengan cepat, tepat dan efisien ?
3. Bagaimana mengetahui inventori jumlah anak dari sepasang burung perkutut yang telah laku terjual atau belum terjual dengan cepat ?

1.3. Pembatasan Masalah

Pembuatan program hanya untuk menyimpan identitas masing-masing burung dan membuat sertifikat dari burung perkutut tersebut. Juga dapat

menyimpan sertifikat yang telah dibuat selain itu menyediakan informasi tentang jalur keturunan dan jumlah anak dari masing-masing pasangan perkutut baik yang sudah laku atau sisa anak dari pasangan burung tersebut.

1.4. Maksud dan Tujuan

Tujuan dari Tugas Akhir ini yaitu:

1. Membuat program sistem informasi pada peternakan burung perkutut.
2. Memberikan informasi tentang garis keturunan dari seekor burung perkutut.
3. Membuat sertifikat burung perkutut dan menyimpannya.
4. Inventori anak burung dari masing-masing pasangan.

1.5. Metode Pemecahan Masalah

Metode yang digunakan untuk menyelesaikan masalah di atas, dengan melakukan studi pustaka dan survey ke lokasi. Dengan cara ini diharapkan akan memperoleh data-data, informasi, konsep-konsep yang bersifat teoritis.

Setelah itu mulai membuat program yang berhubungan dengan tugas akhir ini. Kemudian melakukan pengujian terhadap program tersebut. Terakhir adalah penyusunan dan penulisan buku tugas akhir.

1.6. Sistematika Penulisan Tugas Akhir

Sistematika tugas akhir ini diatur dan disusun dalam 5 (lima) bab, dan tiap-tiap bab terdiri dari beberapa sub bab. Untuk memberikan gambaran tugas akhir ini akan diuraikan secara singkat materi dari bab-bab sebagai berikut :

Bab I Pendahuluan

Dalam bab ini dibahas mengenai latar belakang, permasalahan, pembatasan masalah, maksud dan tujuan serta sistematika tugas akhir.

Bab II Landasan Teori.

Bab ini membahas mengenai teori-teori yang melandasi dalam pembuatan program.

Bab III Permasalahan.

Bab ini membahas mengenai permasalahan pokok dari tugas akhir.

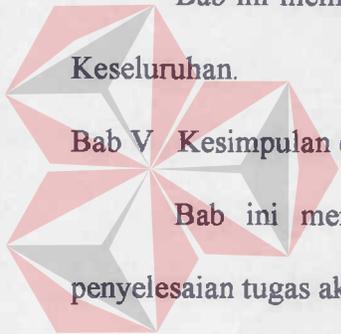
Bab IV Pemecahan Masalah.

Bab ini membahas mengenai pemecahan masalah perangkat lunak secara

Keseluruhan.

Bab V Kesimpulan dan Saran.

Bab ini merupakan kesimpulan dan saran yang dapat diambil dari penyelesaian tugas akhir ini.



UNIVERSITAS

Dinamika

BAB II

LANDASAN TEORI

2.1. Produksi

Produksi adalah kegiatan yang menghasilkan suatu barang. Pada peternakan perkutut barang yang dihasilkan adalah burung perkutut hasil dari penangkaran pada peternakan tersebut. Hasil produksi ini tidak dapat dipastikan nilai ekonomisnya. Biasanya terkait dengan keberhasilan dari orang tua burung tersebut menjuarai berbagai turnamen. Untuk itu perkutut yang akan ditenakkan pasti mempunyai kemampuan dan telah memenangkan beberapa turnamen burung perkutut.



2.2. Penjualan

Penjualan adalah penukaran suatu barang dengan menggunakan alat pembayaran yang sah yang berlaku pada saat itu. Dalam dunia burung perkutut penentuan harga dari seekor burung melalui beberapa cara yaitu mengamati burung tersebut, mendengarkan bunyinya kemudian menentukan penawaran harga dari burung tersebut, biasanya sampai berhari-hari. Cara yang lain adalah menentukan nilai dari burung perkutut berdasarkan kemampuan dari orang tua burung perkutut tersebut dalam memenangkan kejuaraan-kejuaraan yang pernah diikutinya. Yang terakhir dijual dengan masal sesuai dengan harga pada saat itu biasa disebut beli borongan atau partai.

Penjualan burung perkutut ini dilengkapi dengan sertifikat yang diterbitkan dari peternakan tersebut. Guna dari sertifikat ini adalah untuk pedoman silsilah

dari burung perkutut tersebut. Dengan adanya sertifikat ini pembeli akan merasa terlindungi hak-haknya juga digunakan untuk menjualnya lagi.

2.3. Kartu Burung

Kartu burung merupakan kartu yang berisi identitas burung, dimana kartu tersebut ditempatkan pada kandang tiap burung. Kartu burung yang berada di kandang tersebut akan didata ulang.

2.4. Sertifikat Burung

Sertifikat burung merupakan data lampiran yang berisi identitas burung itu sendiri serta data silsilah burung itu sendiri.

2.5. Konsep Dasar Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Pendukung dari informasi adalah siklus informasi dan kualitas informasi. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Kejadian-kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi. Data yang diolah melalui model menjadi informasi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus yang disebut dengan siklus informasi (*information cycle*) atau siklus pengolahan data (*data processing cycle*).

2.6. Basis Data

2.6.1. Teori Basis Data

Basis data adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan dalam batasan tertentu. Dalam satu file terdapat record-record yang sejenis, sama besar, sama bentuk dan merupakan satu kumpulan entity yang seragam. Satu record terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. Untuk menyebut isi dari field maka digunakan attribute atau merupakan judul dari satu kelompok entity tertentu, misalnya attribute alamat menunjukkan entity alamat dari pegawai.

Tujuan dibentuknya suatu basis data adalah untuk mempermudah serta mengeffisienkan pengaksesan data, baik pada saat penyimpanan maupun pengambilan kembali informasi yang telah disimpan dalam basis data. Basis data (database) merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data ini merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan dasar / sumber data dalam menyediakan informasi bagi para pemakai. Basis data merupakan tempat penyimpanan data yang secara umum bersifat integrated (berhubungan) dan shared (bersama / multiuser).

Sistem basis data (database system) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-

macam didalam suatu organisasi. Sistem basis data memang dirancang untuk pengelolaan data dalam jumlah yang sangat besar. Pengelolaan data yang dimaksud disini termasuk pendefinisian struktur guna kemudahan penyimpanan informasi, serta persiapan dari langkah-langkah yang akan dilakukan guna kemudahan dalam memanipulasi data agar dapat dilihat dalam berbagai cara serta membuatnya menjadi informasi yang lebih berarti. Data dikelompokkan menurut jenis dan keperluannya. Ini dilakukan agar tidak terlalu banyak memori yang terbuang dalam pengelolaan data tersebut.

Database sistem informasi adalah suatu kumpulan data item yang berhubungan satu sama lain, yang terorganisasi sehingga memungkinkan sebuah program atau lebih dapat mengolah data tersebut, tanpa perlu memperhatikan penyimpanan secara fisik.

File merupakan kumpulan data yang diatur dengan mengikuti beberapa struktur baku. Desain file yang dirancang harus berorientasi pada masa depan, karena dapat mempengaruhi keandalan sistem. Hal ini berkaitan apabila ada perubahan kecil tidak perlu diadakan perubahan struktur.

DBMS (Data Base Management System) ini merupakan satu sistem software untuk menangani masalah-masalah yang berhubungan dengan jumlah data yang sangat banyak dan dibutuhkan waktu yang singkat untuk memperoleh data yang diinginkan. DBMS disusun sedemikian rupa, sehingga suatu data yang diinginkan dapat dicari / diakses dengan mudah. Kemudahan ini dapat dicapai karena susunan masing-masing file terdiri dari record dan field.

Sifat user friendly yang berarti bahwa user tidak kesulitan dalam langkah-langkah, dapat menuntun pemakai / user mulai proses input data sampai tersajinya

informasi dengan adanya fasilitas help bila tersedia dan komentar-komentar yang ditampilkan pada layar sehingga user mengetahui apa yang harus dilakukan kemudian. Definisi bagian-bagian dari database :

a. Entity

Entity adalah orang, tempat, kejadian atau konsep yang informasinya direkam.

b. Attribute

Setiap entity mempunyai attribute atau sebutan untuk mewakili suatu entity. Seorang karyawan dapat dilihat dari atributnya, misalnya nama karyawan, nomor karyawan, alamat, nomor telepon. Attribute juga disebut sebagai data elemen, data field atau data item.

c. Data value (nilai atau isi data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data elemen atau attribute. Attribute nama karyawan menunjukkan tempat dimana informasi nama karyawan disimpan, sedang data value adalah Yohanes, Yohan, merupakan isi data nama karyawan tersebut.

d. Record

Kumpulan elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap. Satu record mewakili satu data atau informasi tentang seseorang misalnya nomor karyawan, nama, alamat, kota, bagian.

e. File

File adalah kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda data valuenya. Contohnya file master customer.

2.6.2. Kegunaan Database

Penyusunan satu database digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu :

1. Redudansi dan inkonsistensi data.

Jika file-file dan program aplikasi diciptakan oleh programmer yang berbeda pada waktu yang berselang cukup panjang, maka ada beberapa bagian data mengalami penggandaan pada file-file yang berbeda. Penyimpanan di beberapa tempat untuk data yang sama ini disebut sebagai redudansi dan mengakibatkan pemborosan ruang penyimpanan dan juga biaya untuk mengakses jadi lebih tinggi. Penyimpanan data yang berulang-ulang di beberapa file dapat mengakibatkan juga inkonsisten (tidak konsisten).

2. Kesulitan dalam pengaksesan data.

DBMS mampu mengambil data secara langsung dengan bahasa yang familiar dan mudah digunakan.

3. Isolasi data untuk standarisasi

Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data. Maka haruslah data dalam satu database dibuat satu format sehingga mudah dibuat program aplikasinya.

4. Multiple User

Dalam rangka mempercepat semua daya guna sistem dan mendapat responsi waktu yang cepat, beberapa sistem mengijinkan banyak pemakai untuk meng"update" data secara simultan. Salah satu alasan mengapa database dibangun karena nantinya data tersebut digunakan oleh banyak orang dalam

waktu yang berbeda, diakses oleh program yang sama tapi berbeda orang dan waktu. Semua ini memungkinkan terjadi karena data yang diolah tidaklah tergantung dan menyatu dalam program tapi ia terlepas dalam satu kelompok data.

5. Masalah keamanan

Tidak setiap pemakai sistem database diperbolehkan untuk mengakses semua data. Keamanan ini dapat diatur lewat program yang dibuat oleh pemrogram atau fasilitas keamanan dari operating sistem misalnya Novell Netware untuk local area network.

6. Masalah integritas (kesatuan)

Database berisi file-file yang saling berkaitan, masalah utama adalah bagaimana kaitan antara file tersebut terjadi. Meskipun kita mengetahui bahwa file a berkaitan dengan file b, namun secara teknis maka ada field kunci yang mengaitkan kedua file tersebut.

7. Masalah data independence (kebebasan data)

Apabila ada perubahan struktur file sedangkan program telah dibuat maka program tersebut harus diubah. Hal ini disebut bahwa program yang telah dibuat tidak bebas terhadap database yang ada. Lain dengan paket bahasa yang diciptakan dari DBMS, apapun yang terjadi pada struktur file, setiap kali kita hendak melihat data cukuplah dengan utility LIST, hendak menambah data cukup dengan APPEND. Ini berarti perintah dalam paket DBMS bebas terhadap database. Apapun perubahan dalam database, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

2.6.3. Merancang model konseptual database

Pada perancangan model konseptual penekanan tinjauan dilakukan pada struktur data dan relasi antar file. Pendekatan yang dilakukan pada perancangan model konseptual adalah menggunakan model data relational. Terdapat dua teknik yaitu :

1. Teknik Normalisasi
2. Teknik Entity Relationship

1. Teknik Normalisasi

Proses normalisasi merupakan proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Proses normalisasi selalu diuji pada beberapa kondisi, apakah ada kesulitan pada saat menambah atau insert, menghapus atau delete, mengubah atau update, membaca atau retrieve pada satu database. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapat database yang optimal. Ada beberapa konsep yang harus diketahui terlebih dulu, yaitu :

a. Field atau atribut kunci.

Setiap field selalu terdapat kunci dari file berupa satu field atau satu set field yang dapat mewakili record. Misalnya nomor pegawai merupakan kunci dari tabel pegawai suatu perusahaan, setiap pencarian cukup dengan menyebut nomor pegawai tersebut maka dapat diketahui nama, alamat dan attribute lainnya mengenai seorang pegawai tersebut.

a.1. Candidat key atau kunci calon

Adalah satu atribut atau satu set minimal attribute yang

mengidentifikasi secara unik suatu kejadian spesifik dari entity. Satu minimal set dari atribut menyatakan secara tak langsung dimana kita tidak dapat membuang beberapa attribute dalam set tanpa merusak kepemilikan yang unik. Jika satu set kunci kandidat berisi lebih dari satu attribute, biasanya disebut sebagai composit key (kunci campuran atau gabungan).

contoh : File master burung berisi attribute

- kode burung
- kode ayah
- kode ayah

Kunci kandidat disini adalah :

- Kode burung, kode ayah dan kode ibu karena unik tidak mungkin ganda.

a.2. Primary key atau kunci primer

Adalah satu atribut atau satu set minimal attribute yang tidak hanya mengidentifikasi secara unik suatu kejadian spesifik, tapi juga dapat mewakili setiap kejadian dari suatu entity. Setiap kunci kandidat punya peluang menjadi primary key, tetapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap entity yang ada.

a.3 Alternate key atau kunci alternatif

Adalah kunci kandidat yang tidak dipakai sebagai primary key. Kerap kali kunci alternatif dipakai sebagai kunci pengurutan dalam laporan misalnya.

a.4. Foreign Key atau kunci tamu

Adalah satu attribute (atau satu set attribute) yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya. Kunci tamu

ditempatkan pada entity anak dan sama dengan kunci primary induk direlasikan. Hubungan antara entity induk dengan entity anak adalah hubungan satu lawan banyak (one to many relationship).

b. Bentuk-bentuk normalisasi

b.1. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

b.2. Bentuk normal kesatu

Bentuk normal kesatu mempunyai ciri yaitu setiap data dibentuk dalam flat file (file datar / rata), data dibentuk dalam satu record demi satu record dan nilai dari field-field berupa "atomic value". Tidak ada set atribut yang berulang-ulang atau atribut bernilai ganda (multivalue).

Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

b.3. Bentuk normal kedua

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama / primary key, sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

b.4. Bentuk normal ketiga

Untuk menjadi bentuk normal ketiga maka relasi haruslah dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci haruslah bergantung hanya pada primary key dan pada primary key secara menyeluruh.

2. Teknik Entity Relationship

Pada model data relational hubungan antar file direlasikan dengan kunci relasi (relation key), yang merupakan kunci utama dari masing-masing file.

a. Entity Relationship Konsep

Relasi antar dua file atau dua tabel dapat dikategorikan menjadi tiga macam.

◆ One to one relationship 2 file

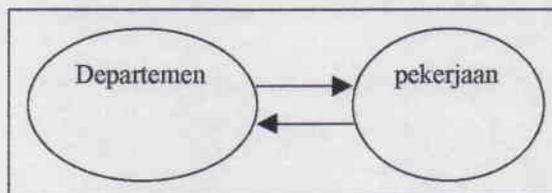
Hubungan antara file pertama dengan file kedua adalah satu berbanding satu.

Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk

menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah

tunggal. Misalnya suatu departemen hanya mengerjakan satu jenis pekerjaan

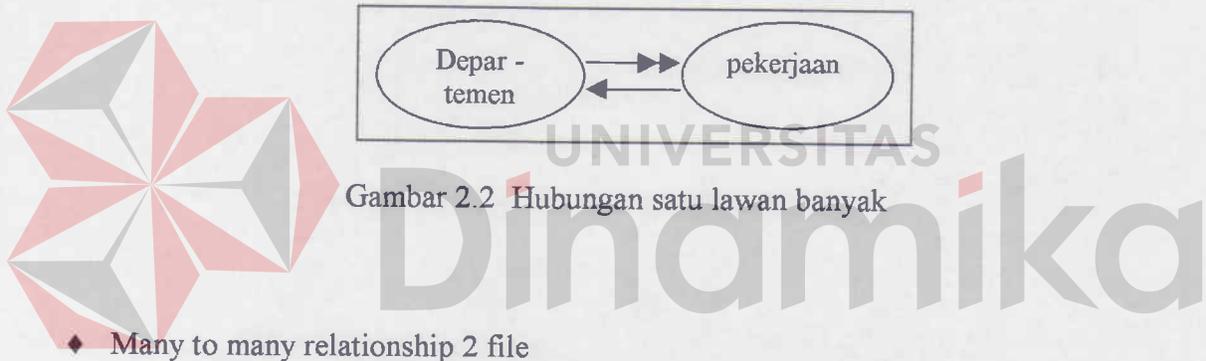
saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.



Gambar 2.1. Hubungan satu lawan satu

◆ One to many relationship 2 file

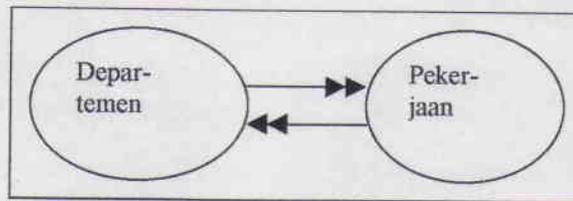
Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut. Misalnya suatu pekerjaan hanya dikerjakan oleh satu departemen saja, tetapi suatu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus. Maka hubungannya digambar menunjukkan bahwa satu departemen menunjuk banyak pekerjaan.



Gambar 2.2 Hubungan satu lawan banyak

◆ Many to many relationship 2 file

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut. Misalnya satu departemen mampu mengerjakan banyak pekerjaan juga satu pekerjaan dapat ditangani oleh banyak departemen.

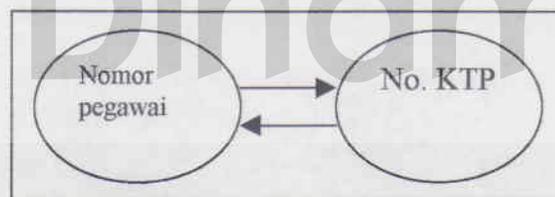


Gambar 2.3 Hubungan banyak lawan banyak

Demikian pula untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu file.

◆ Relasi one to one 2 atribut dalam 1 file

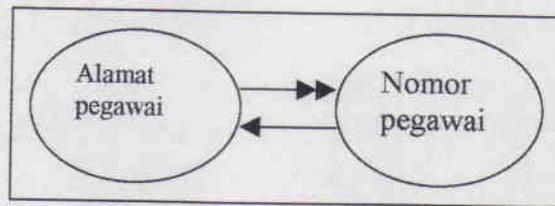
Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan satu. Misalnya atribut nomor pegawai yang unik dan atribut nomor ktp pegawai mempunyai hubungan satu lawan satu. Satu nomor pegawai hanya satu nomor ktp tidak ada yang berganda.



Gambar 2.4 Bagan relasi satu lawan satu

◆ Relasi many to one 2 atribut dalam 1 file

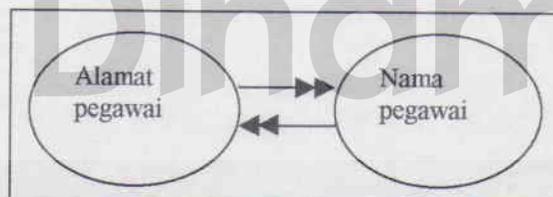
Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut maka hubungan antara atribut alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai.



Gambar 2.5 Bagan relasi banyak lawan satu

◆ Relasi many to many 2 attribute dalam 1 file

Hubungan antara satu attribute dengan attribute yang lain dalam satu file yang sama mempunyai hubungan banyak lawan banyak. Misalnya satu alamat pegawai terdapat puluhan pegawai kantor tersebut dan beberapa pegawai punya nama yang sama mempunyai alamat berbeda. Maka hubungan antara attribute alamat pegawai dengan nama pegawai adalah beberapa alamat menunjukkan nama pegawai yang sama dan sebaliknya.



Gambar 2.6 Bagan relasi banyak lawan banyak

b. Langkah-langkah perancangan basis data adalah :

1. Memilih kunci record.
2. Membentuk entity / tabel berdasarkan kunci record.
3. Mencari relasi antar file dan menggabungkannya.
4. Memasang kunci relasi antar file.
5. Melengkapi isi tiap file.
6. Melakukan pemeriksaan relasi.

2.6.4. Perancangan Sistem

DFD (Data flow diagram) atau DAD (diagram arus data) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur dimana dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas juga merupakan dokumentasi dari sistem yang baik. Komponen yang ada dalam DFD adalah sebagai berikut :

1. Kesatuan luar (external entity),

Notasi atau simbol :



Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (external entity) merupakan kesatuan (entity) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

2. Arus data (data flow)

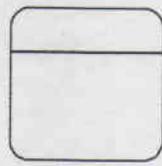
Notasi atau simbol : 

Arus data (data flow) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (process), simpanan (data store) dan kesatuan luar (external entity). Aliran data ini digunakan untuk menjelaskan pergerakan

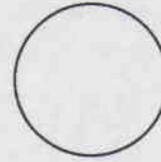
paket informasi dari satu bagian sistem ke bagian lain.

3. Proses (process)

Notasi atau simbol :



atau



Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Proses menggambarkan bagian dari sistem yang mengubah masukan (input) menjadi keluaran (output).

4. Simpanan data (Data store)

Notasi atau simbol :



Simpanan data (data store) merupakan simpanan dari data yang dapat berupa : file atau database di sistem komputer, arsip atau catatan manual,

suatu agenda atau buku. Untuk penyimpanan data sebelum dan sesudah proses lebih lanjut.

2.7. Relational Database

Sistem informasi pada perusahaan ini menggunakan konsep relational database model. Dimana terjadi hubungan antara dua atau lebih tabel yang menghasilkan suatu informasi tertentu. Kemudian relasi dalam database ini dibangun berdasarkan teknik Enforce Referential Integrity, yaitu suatu teknik dimana informasi data disimpan dalam tabel-tabel yang saling berhubungan satu atau lebih field sebagai kunci penghubungnya. Jika pemakai memasukkan atau melakukan kesalahan dalam pemilihan data, maka sistem akan secara otomatis

menolaknyanya. Kemudian jika field yang menjadi key dihapus, maka semua record yang ada hubungannya dengan field tersebut juga akan dihapus oleh sistem secara otomatis, hal ini terjadi pula pada updating data. Maka kelebihan teknik ini adalah menghindari kesalahan memasukkan data atau update data serta menghindari adanya duplikasi data. Data yang ada pasti benar atau dengan kata lain bebas dari kesalahan yang mungkin dilakukan oleh operator.

2.8. Pseudocode

Pseudocode adalah kode yang mirip dengan instruksi kode program yang sebenarnya, sehingga lebih tepat digunakan untuk menggambarkan algoritma yang akan dikomunikasikan oleh programmer. Pseudocode akan memudahkan baik user atau programmer untuk memahami dan menggunakannya. Pseudocode menjelaskan secara terperinci prosedur-prosedur yang akan dijalankan program.

2.9. Bentuk Program

Bahasa pemrograman Pascal terkenal berkat kedekatannya dengan bahasa manusia. Selain itu, Pascal unggul dalam hal kemudahan, kecanggihan dan metode pemrograman terstruktur. Bahasa pemrograman yang dipakai Delphi berdasarkan pada bahasa pemrograman Pascal, yang berarti semua keunggulan yang dimiliki Pascal juga dimiliki Delphi. Delphi sepenuhnya menerapkan metode *Object Oriented Programming*, suatu metode pemrograman yang mampu menangani program berukuran besar karena memiliki sistem organisasi yang bagus.

Delphi memilah program menjadi dua bagian utama yaitu bagian primer dan bagian sekunder. Bagian primer berupa file program yang mengkoordinasi

keseluruhan program, file ini dicirikan dari ekstensinya yaitu: *.dpr. Sedangkan bagian sekunder merupakan bagian yang melaksanakan tugas-tugas tertentu dan lazim disebut unit (ada yang menyebutnya sebagai 'rutin'), file bagian ini dicirikan oleh ekstensi .pas. pada saat menyusun program delphi, yang kita tuliskan hanya code untuk file-file unit (.pas), sedangkan file project akan otomatis terbentuk, misalnya pada project yang telah kita susun sebelumnya. Saat penyusunan project tersebut, kita melakukan proses perancangan tampilan dan penulisan code program. Pada perencanaan tampilan, kita menuliskan suatu tugas yang harus diemban oleh tombol. Semua hasil kerja kita tersebut disimpan dalam file-file tertentu yang terorganisir. Hasil rancangan tampilan disimpan dalam file dengan ekstensi *.dfm, code program disimpan dalam file dengan ekstensi *.pas dan semua file tersebut diorganisir oleh sebuah file project dengan ekstensi *.dpr. File terakhir ini otomatis dibentuk oleh Delphi sendiri.

2.9.1. Program Primer

Program primer adalah bagian program yang disimpan sebagai file dengan ekstensi *.dpr, dimana kode programnya akan otomatis dituliskan oleh Delphi sendiri dan tidak dianjurkan untuk mengeditnya. Untuk itu Delphi sengaja tidak menampilkan langsung pada *Code Editor*. Untuk membukanya lewat perintah Units... dari menu view.

2.9.2. Program Sekunder

Bagian program sekunder yang paling populer adalah unit, yaitu suatu kumpulan konstanta, type-type data, variabel, prosedur maupun fungsi-fungsi. Unit digunakan sebagai modul-modul library yang dapat dipakai berulang-ulang.

Jadi unit dapat dipandang sebagai pemasok konstanta- konstanta, type-type data, variabel, prosedur maupun fungsi / rutin.

Dengan unit, pekerjaan programmer akan sangat terbantu dalam menyusun suatu program yang sangat besar, karena aplikasi Delphi tersusun atas unit-unit yang membentuk rangkaian fungsi baru. Sebuah unit dalam suatu program mirip dengan sebuah himpunan bagian dalam sebuah himpunan besar. Himpunan-himpunan bagian tersebut memiliki fungsinya masing-masing yang dapat berdiri sendiri atau dengan kata lain sudah dapat melakukan suatu tugas program sendiri. Kemandirian ini sangat membantu, terutama dalam penanganan error. Apabila terjadi error pada program maka biasanya error terdapat pada suatu tempat saja, yaitu pada unit-unit tertentu yang tidak akan merambat kepada keseluruhan program. Sehingga koreksi kesalahan dapat dikerjakan hanya pada unit yang bermasalah saja. Tindakan seperti ini dikenal dengan istilah lokalisir gangguan.

Antara unit satu dengan lainnya dalam suatu program dapat saling berkomunikasi dengan relasi pemakaian bersama suatu data, rutin atau kandungan lain dalam unit. Masing-masing komponen Delphi sudah memiliki unitnya sendiri-sendiri. Delphi telah menyediakan banyak unit yang dapat dimanfaatkan setiap saat. Secara garis besar, sebuah unit mempunyai tiga bagian utama, yaitu :

1. *Interface* adalah bagian yang mengandung semua deklarasi yang dapat diakses ke unit lain. Penulisannya dimulai dari kata interface dan terletak setelah baris header atau judul unit dan berakhir pada baris yang mengandung kata implementation. Pada bagian ini dapat mendeklarasikan konstanta-konstanta, variabel, prosedur maupun fungsi tertentu.

2. *Implementation* adalah bagian yang terdiri atas badan prosedur, badan fungsi maupun metode dari semua nama prosedur, fungsi maupun metode yang telah dideklarasikan pada interface. Penulisannya terletak setelah kata *implementation* dan sebelum kata *end* yang terakhir.
3. *Initialization* adalah bagian ini berguna untuk membuat inisialisasi dalam suatu unit. Pada bagian ini dapat menuliskan code tertentu yang diinginkan. Code ini akan dijalankan paling awal oleh Delphi. Lokasi penulisannya ada pada baris sebelum kata *end* yang terakhir.

2.10. Konsep Dasar Program

Secara garis besar, program memiliki beberapa konsep dasar, antara lain:

1. *Statement*

Aspek awal yang perlu diketahui dalam bahasa pemrograman adalah *reserve word* dan *statement*. *Reserve word* harus dikenal bahkan harus dihafalkan layaknya menghafal kata-kata baru apabila mempelajari suatu bahasa pemrograman baru. Contoh *reserve word* adalah *close*, *procedure*, *while*, *for*, *case*, *until* dan sebagainya. *Statement* adalah perintah-perintah dalam code yang akan dijalankan oleh Delphi. *Statement* merupakan bagian yang penting dalam suatu program, karena tanpanya program tidak akan pernah berjalan sesuai dengan yang dikehendaki. *Statement* dituliskan di antara kata *begin* dan *end*. Penulisan diakhiri dengan titik koma (;), misalnya *statement close* yang berarti memerintahkan penutupan program. Penulisannya adalah:

Begin

Close;

End.

Statement penugasan adalah statement yang berfungsi untuk perubahan properti atau variabel saat program berjalan, misalnya: `Edit1.Color:=ClRed` adalah statement penugasan yang memerintahkan agar warna *form* berubah menjadi merah apabila statement ini dijalankan.

2. Komentar Program

Komentar dapat ditambahkan pada suatu baris program dengan menuliskannya di dalam kurung kurawal '{ }'. Dengan penambahan kurung kurawal ini maka Delphi tidak akan menanggapinya sebagai kode sehingga tidak akan dijalankan.

3. Deklarasi *Identifier*.

Identifier suatu pengenalan atau nama-nama yang diberikan sebagai bagian dari suatu aplikasi, diantaranya: variabel, type, prosedur, fungsi, metode dan lain-lain. Identifier ini penting, dan harus dideklarasikan sebelum memakainya pada kode program. Hal ini penting karena akan menaikkan efisiensi kode dan mengurangi error.

4. Deklarasi *variabel*

Variabel adalah nama dalam kode yang mengacu ke alamat memori dimana isinya dapat berubah saat program berjalan. Anda harus mendeklarasikan variabel, karena tanpa deklarasi variabel tidak akan dapat diproses oleh program. Untuk mendeklarasikan variabel ada dua tahap, yaitu : tahap penanaman dan tahap pemberian type.

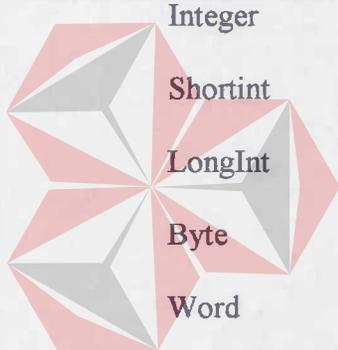
Penanaman variabel, dianjurkan yang mudah diingat, misalnya: Bulan, jumlah dan sebagainya, bukan x,y,z dan seterusnya. Adapun aturannya adalah:

- Panjang maksimum 63 karakter.

- Harus diawali dengan sebuah huruf atau garis bawah (`_`).
- Karakter selanjutnya dapat berupa huruf, garis bawah atau angka (`0..9`).
- Tidak boleh mengandung simbol (`#,$,%,^....`).
- Tidak boleh menggunakan reserved word milik Delphi.

Sedangkan pemberian type variabel sifatnya mutlak, harus dilakukan karena bertujuan mendefinisikan himpunan nilai-nilainya. Ada 15 jenis data yang merupakan nilai variabel antara lain:

1. Kategori *integer*.



Integer	-32768 32768.
Shortint	-128 127.
LongInt	-2147483648 2147483648.
Byte	0...255.
Word	0...65535.

2. Kategori *Real*.

Single	1.59×10^{-45} 3.4×10^{38} .
Double	5.0×10^{-325} 1.7×10^{308} .
Extended	1.9×10^{-4951} 1.1×10^{4932} .
Comp	$-263+1 \dots 263-1$.
Real	2.9×10^{-23} ... 1.7×10^{38} .

3. Kategori *boolean*.

Boolean	Bernilai True atau False.
---------	---------------------------

4. Kategori *Char*.

Char	<i>Sebuah karakter ASCII.</i>
------	-------------------------------

2. Fungsi.

Pemanggilan fungsi pada dasarnya sama dengan pemanggilan prosedur, yaitu penulisan nama fungsinya pada baris code yang terkait

2.12. Percabangan Dan Pengulangan

Dalam menyusun suatu program tentunya mempunyai alur tertentu dari proses yang diinginkan. Untuk melaksanakan pemindahan alur diperlukan percabangan, misalnya ada kondisi A terjadi maka akan mengerjakan pekerjaan B, namun jika kondisi A tidak terjadi maka akan mengerjakan pekerjaan C. Kondisi ini dapat dipandang sebagai hubungan sebab akibat, dimana keadaan A dapat disebut penyebab dan pekerjaan B sebagai akibat 1 serta pekerjaan C sebagai akibat 2 dan seterusnya. Dengan Delphi proses percabangan dengan statement *if* dan *case*. Kadang kala dalam membuat program diperlukan proses pengulangan untuk itu Delphi menyediakan statement tiga jenis yaitu *repeat*, *while* dan *for*.

2.12.1. Percabangan

Ada dua percabangan yang disediakan oleh Delphi adalah :

1. Percabangan dengan *if*.

Percabangan *if* akan mensyaratkan suatu kondisi sebelum melakukan proses selanjutnya. Jika kondisi benar (*true*) maka proses akan berlanjut ke arah jalur tertentu. Namun apabila keadaan salah (*false*) maka proses akan berjalan ke arah lain.

2. Percabangan dengan Case

Statement Case pada dasarnya digunakan juga untuk percabangan yang banyak, hanya saja ia lebih disukai karena penulisannya lebih ringkas dan lebih mudah di mengerti serta prosesnyapun cepat.

2.12.2. Pengulangan

Pengulangan pada Delphi menyediakan tiga macam, yaitu:

1. Pengulangan dengan *statement repeat*. Statement repeat melaksanakan pengulangan proses terhadap statement tunggal ataupun kelompok sampai suatu keadaan dinyatakan true (bernilai benar). Penulisan statement ini diawali dengan kata repeat dan diakhiri dengan kata until. Kondisi yang ingin dicapai selalu suatu ekspresi Boolean. Statement Repeat until digunakan jika tidak diketahui jumlah pengulangan yang akan dilakukan tapi tahu program harus berjalan minimum satu kali.
2. Pengulangan dengan *statement While*. Statement while akan mengevaluasi suatu kondisi pada awal proses, kebalikan dari statement repeat. Penulisan statement ini diawali dengan pasangan kata *While... do* diikuti dengan pasangan kata *begin..end*. Statement While digunakan jika tidak yakin jumlah pengulangannya dan masih ada peluang untuk menjalankan program secara keseluruhan.
3. Pengulangan dengan *statement for*. Statement for digunakan untuk pengulangan proses terhadap statement tunggal ataupun kelompok. Pengulangan dilakukan dalam jumlah tertentu. Penulisan statement ini diawali dengan pasangan kata *for..do* didalam pasangan kata *begin..end*. Statement for

adalah statement perulangan yang sangat cepat dan efisien jika telah diketahui berapa kali harus melakukan pengulangan.

2.13. Blok Dan Scope

Delphi memakai metode Object Oriented Programming (OOP). Object memegang peranan penting dalam OOP, demikian pula dengan blok dalam Object Pascal. Blok berperan besar dalam penyusunan struktur program aplikasi, juga dalam penentuan scope variabel, properti dan lain-lain.

Blok adalah suatu kelompok tertentu dalam program yang mempunyai fungsi tertentu pula, dimana dikenal adanya awalan dan akhiran. Blok dapat berupa sebuah prosedur, sebuah unit bahkan sebuah project. Jadi sekelompok blok-blok prosedur akan membentuk suatu unit, dan sekelompok unit-unit membentuk suatu project.

Blok Pascal mempunyai dua bagian utama, yaitu bagian deklarasi dan bagian statement. Bagian statement akan menggunakan segala sesuatu sesuai dengan yang ada pada bagian deklarasi. Deklarasi yang dikenal dalam Object Pascal antara lain:

- Deklarasi Variabel.
- Deklarasi Konstanta.
- Deklarasi Type.
- Deklarasi Label.
- Deklarasi Prosedure, Fungsi dan Metode.

Sedangkan ruang lingkup antara awalan dan akhiran suatu blok disebut

Scope lokal merupakan ruang kerja setempat, sedangkan scope global lebih luas lagi ruangnya. Misalnya deklarasi identifier pada suatu blok prosedur, berarti identifier tersebut hanya dapat digunakan untuk prosedur itu saja, prosedur lain tidak dapat menggunakannya, apalagi unit dimana prosedur tersebut berada. Agar dapat dipakai oleh prosedur lain maka deklarasi identifier tersebut harus bersifat global. Misalnya deklarasi dilakukan tidak pada prosedur tertentu, tetapi pada suatu unit dimana prosedur tersebut berada (bagian implementation). Ada beberapa penamaan blok sesuai lokasinya yaitu:

- Blok di dalam *event handler* (= prosedur).
- Blok di dalam unit.
- Blok di dalam project.

Blok didalam event handler adalah yang paling sering dibicarakan, karena merupakan kumpulan fungsi terkecil dalam struktur program. Sedangkan blok selanjutnya diawali dengan bagian deklarasi, yaitu pada kata var. Bagian statement dari blok ini hanya dapat menggunakan segala sesuatunya dari yang telah dideklarasikan pada bagian deklarasinya saja (dalam suatu blok). Deklarasi demikian dikenal sebagai deklarasi scope lokal. Artinya deklarasi tersebut tidak dapat digunakan oleh blok lain. Jika anda menginginkan deklarasi yang dapat dipakai oleh banyak blok maka anda harus melakukan deklarasi scope global, dan penulisannya tidak di dalam suatu blok. Namun pada implementasi dari suatu unit.

2.14. Penentuan Type Data Baru

Banyak type data yang telah disediakan namun demikian tidak menutup kemungkinan type-type data tersebut masih belum sesuai dengan data yang

diinginkan. Untuk itu Delphi menyediakan cara untuk menentukan type data tertentu.

1. Type urutan. Type data urutan, sesuai namanya, mengenal adanya urutan dalam kelompok datanya yang disesuaikan dengan letak penulisan masing-masing data. Urutan merupakan nilai (*value*) yang dinyatakan dalam bentuk angka dan dimulai dari kiri ke kanan. Jadi data paling kiri (awal) akan bernilai 0 diikuti data bernilai 1, 2, 3, 4, dan seterusnya.
2. Type Subrange. Type data ini memberlakukan suatu ruang lingkup yang dapat ditentukan batasnya dan berlaku untuk data integer, boolean, char atau urutan. Type ini sangat membantu apabila bekerja dengan data yang memiliki range. Berdasarkan kenyataan, semua peralatan yang ada di dunia ini memiliki range kerja tertentu. Bahkan tidak hanya kapasitas yang mempunyai batas, umur pun ada batasnya
3. Type himpunan (set) adalah koleksi data dengan sebuah type data saja yang dapat berupa integer, boolean, char, urutan atau subrange.
4. Type data *array* adalah suatu kumpulan data dimana masing-masing data ditandai oleh urutan angka yang disebut index. Macam-macam type data array antara lain adalah :
 1. Array multidimensi.

Array dapat menyatakan kumpulan variabel dalam dimensi yang lebih dari satu. Misalnya suatu tabel yang merupakan kumpulan data dalam dua dimensi, yaitu kelompok data baris dan kelompok data kolom.

2. Array karakter.

String sebenarnya suatu array dari karakter satu dimensi. Jika mendeklarasikan suatu variabel bertipe string harus menentukan ukuran string tersebut seperti layaknya pada waktu mendeklarasikan type array.

5. Type Record *Record* adalah kumpulan data yang dapat dipakai secara menyeluruh bagi suatu aplikasi. Tidak seperti type set, type record merupakan gabungan semua type data yang dipakai aplikasi. Jadi di dalamnya ada type array, real, boolean dan sebagainya. Record mengandung field yang dapat menyimpan nilai-nilai data. Masing-masing field memiliki sebuah type datanya sendiri. Field-field dalam sebuah record selain dapat dipakai oleh record itu sendiri dapat juga dipakai oleh record lain.

2.15. Type Data Abstrak

Definisi dari TDA adalah TDA dapat dipandang sebagai suatu model matematika dan sekumpulan operasi yang didefinisikan terhadap model itu.

Konsep prosedur yang menarik untuk dianalogikan dengan konsep TDA adalah:

1. Generalisasi Operator.

Konsep ini memberi kesempatan agar pemakai memperluas penggunaan operator yang ada dengan cara mendefinisikan operator baru. Pendefinisian operator baru (misalnya pengali matriks, penghitung akar, dan lain-lain) didasarkan pada operator-operator dasar (kali, bagi, dan lain-lain).

2. Enkapsulasi bagian-bagian algoritma.

Konsep ini memberi kesempatan untuk memisah-misahkan algoritma menjadi modul-modul kerja berdasarkan aspek-aspek tertentu dari kegiatan

algoritma. Sebagai contohnya, dalam sebuah program kemahasiswaan, dilakukan pembagian secara modular, untuk kegiatan pembacaan data dan pengujian kesahihan data diimplementasikan dalam sebuah prosedur “validasi data”. Dengan demikian jika pada suatu saat diinginkan perubahan pola pengujian data karena perubahan data, perubahan hanya dilakukan terhadap prosedur “validasi data” tersebut. Jadi kegunaan enkapsulasi prosedur ini adalah kemudahan dalam perancangan, implementasi dan pemodifikasian program, karena prosedur yang bersangkutan mudah ditemukan, lalu diperbaiki beberapa baris tanpa banyak mengubah prosedur-prosedur yang lainnya, dan keseluruhan program.

2.16. Type data abstrak dasar

Hampir pada setiap program yang sifatnya agak lanjut untuk penerapan di bidang compiler, sistem operasi maupun struktur data akan mengikutsertakan tiga

TDA dasar yaitu:

- Senarai bertaut (*Linked List*).
- Tumpukan atau *stack*.
- Antrian atau *queue*.
- Pohon atau *Tree*.

Konsep linked list adalah konsep data yang fleksibel. Struktur ini terdiri atas rangkaian elemen yang saling berhubungan, dimana setiap elemen dihubungkan dengan elemen lainnya oleh sebuah pointer. Pointer adalah sel yang nilainya merupakan alamat dari sel lain, sel yang lain itu dapat berupa data atau berupa

pointer juga, Jadi setiap elemen dalam linked list selalu ada pointer. Macam-macam Linked List antara lain :

1. *Single Linked List.*
2. *Doubly Linked List.*
3. *Circular Linked List.*
4. *Multilevel Linked List.*

Konsep Stack sebenarnya juga merupakan list, tapi list yang dikenai proses penghapusan dan penyisipan hanya pada sebuah tempat yaitu di ujung. Dengan demikian jika akan dilakukan penghapusan data, data yang akan dihapus justru data yang paling baru disisipkan. Karena inilah maka nama lain dari stack adalah LIFO (*Last In First Out*).

Konsep antrian (*queue*) atau lebih dikenal dengan FIFO (*first in First out*) merupakan TDA yang sering digunakan dalam program. Antrian juga diimplementasikan dalam struktur list.

Konsep Pohon tree adalah tipe struktur data yang sangat penting dan sangat sering digunakan. Secara informal sebuah pohon adalah struktur dari sekumpulan elemen dengan salah satu elemennya merupakan akarnya atau *root*, dan sisanya yang lain merupakan bagian-bagian pohon yang terorganisasi dalam susunan berhirarki dengan *root* sebagai puncaknya. Secara rekursif suatu pohon dapat didefinisikan sebagai berikut:

1. Sebuah simpul tunggal adalah sebuah pohon.
2. Bila terdapat simpul n , dan beberapa sub- pohon T_1, T_2, \dots, T_k , yang tidak saling berhubungan, yang masing-masing akarnya adalah

n_1, n_2, \dots, n_k , dari simpul / sub pohon ini dapat dibuat sebuah pohon baru dengan n sebagai akar dari simpul-simpul n_1, n_2, \dots, n_k .

Istilah-istilah dasar pada struktur data pohon adalah :

1. Simpul / Node merupakan elemen-elemen pohon yang mengandung informasi atau data dan penunjuk percabangan.
2. Akar / Root adalah relasi antara simpul satu dengan simpul lainnya dianalogikan dengan relasi dalam sebuah garis keturunan. Simpul yang atas merupakan akar dari simpul yang dibawahnya.
3. Tingkat dari simpul adalah tingkat atau level dari akar didefinisikan sebagai 1, sedangkan tingkat dari simpul lainnya didefinisikan sebagai $1 +$ tingkat dari orangtua simpul tersebut. Kedalaman dari suatu pohon dinyatakan oleh tingkat yang tertinggi dari simpul yang terdapat pada pohon tersebut. Jadi jika sebuah pohon dikatakan mempunyai kedalaman n maka terdapat paling sedikit satu simpul dari pohon tersebut yang mempunyai tingkat n .
4. Derajat dari simpul: setiap simpul memiliki derajat keturunan atau order, sebuah simpul dikatakan berderajat n jika memiliki n tahap turunan menuju simpul yang terendah.

Macam-macam bentuk Pohon antara lain :

1. Pohon *Biner* adalah pohon yang setiap simpulnya memiliki paling banyak dua buah cabang / anak.
2. Pohon *Threaded* adalah bentuk pohon yang menyediakan pointer tambahan yang digunakan untuk menunjuk ke orang tua dari node tersebut.

BAB III

PERMASALAHAN

Dalam era informasi ini hampir semua bidang usaha tidak terlepas dari campur tangan komputer yang semakin hari meningkat kemampuannya baik hardware maupun software. Banyak perusahaan-perusahaan maupun instansi baik pemerintah maupun non pemerintah yang mengharapkan adanya otomatisasi kantor dengan menggunakan komputer. Perkembangan ini juga merasuki dunia peternakan terutama peternakan percutut yang dituntut untuk menyediakan informasi yang akurat karena burung percutut mempunyai nilai jual yang tinggi.

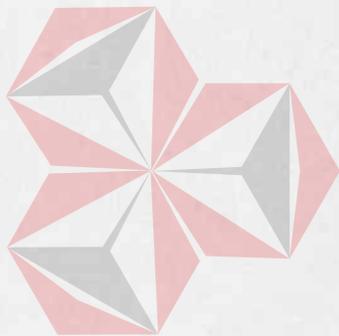
Pada peternakan burung percutut kelahiran seekor burung percutut akan dibuatkan sebuah sertifikat yang berisi kode burung kemudian orang tua burung serta kakek nenek dari burung tersebut. Ini sangat perlu karena nilai ekonomis seekor burung percutut berdasarkan dari garis keturunan dari burung tersebut.

Untuk itu setiap peternakan burung akan mempunyai suatu sistem yang menyediakan data dari masing-masing burung percutut. Selama ini mereka menggunakan kartu untuk masing-masing burung percutut. Ketika peternakan ini berkembang dengan pesat maka data ini akan berjumlah ribuan sehingga diperlukan sistem informasi yang baru untuk menangani masalah tersebut. Karena sistem dengan menggunakan kartu akan memerlukan waktu yang lama untuk mencari silsilah burung tersebut dalam membuat sebuah sertifikat.

Pada saat pembuatan sertifikat pada sistem sebelum komputerisasi dilakukan dengan mencari data dari orang tua burung dengan cara manual. Cara ini sangat memungkinkan jika jumlah masih sedikit tetapi kalau sudah banyak maka akan memerlukan waktu yang sangat lama dan hampir tidak mungkin dilakukan. Selain itu juga proses untuk mencari jumlah anak dari sepasang percutut akan terasa sulit dan

menyita waktu. Untuk memecahkan masalah ini dibuat suatu sistem komputer yang dapat memproses sertifikat dengan cepat dan efisien.

Pembuatan program hanya untuk menyimpan identitas masing-masing burung dan membuat sertifikat dari burung perkutut tersebut. Juga dapat menyimpan sertifikat yang telah dibuat selain itu menyediakan informasi tentang jalur keturunan dan jumlah anak dari masing-masing pasangan perkutut baik yang sudah laku atau sisa anak dari pasangan burung tersebut.



UNIVERSITAS
Dinamika

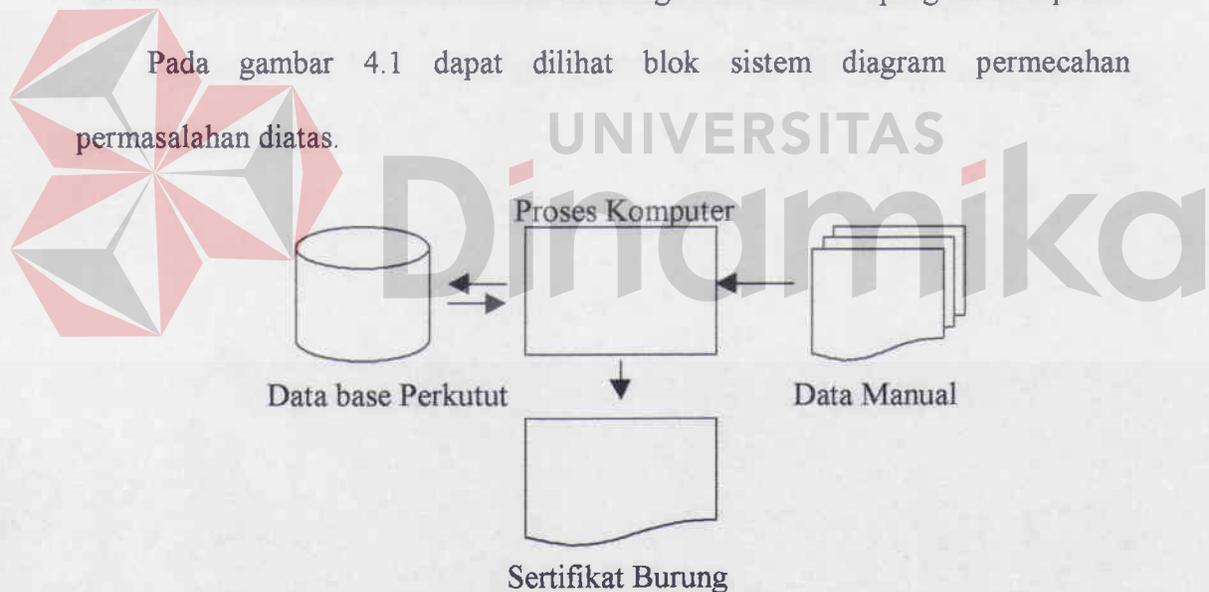
BAB IV

PEMECAHAN MASALAH

4.1. Analisa Permasalahan

Berdasarkan permasalahan yang ada pada peternakan burung perkutut berupa pengarsipan data-data burung perkutut, jika peternakan tersebut mulai berkembang menjadi besar maka data-data burung perkutut tersebut akan bertambah banyak sehingga penggunaan cara manual akan menemui kesulitan, maka untuk memecahkan masalah tersebut digunakan bantuan program komputer.

Pada gambar 4.1 dapat dilihat blok sistem diagram pemecahan permasalahan diatas.

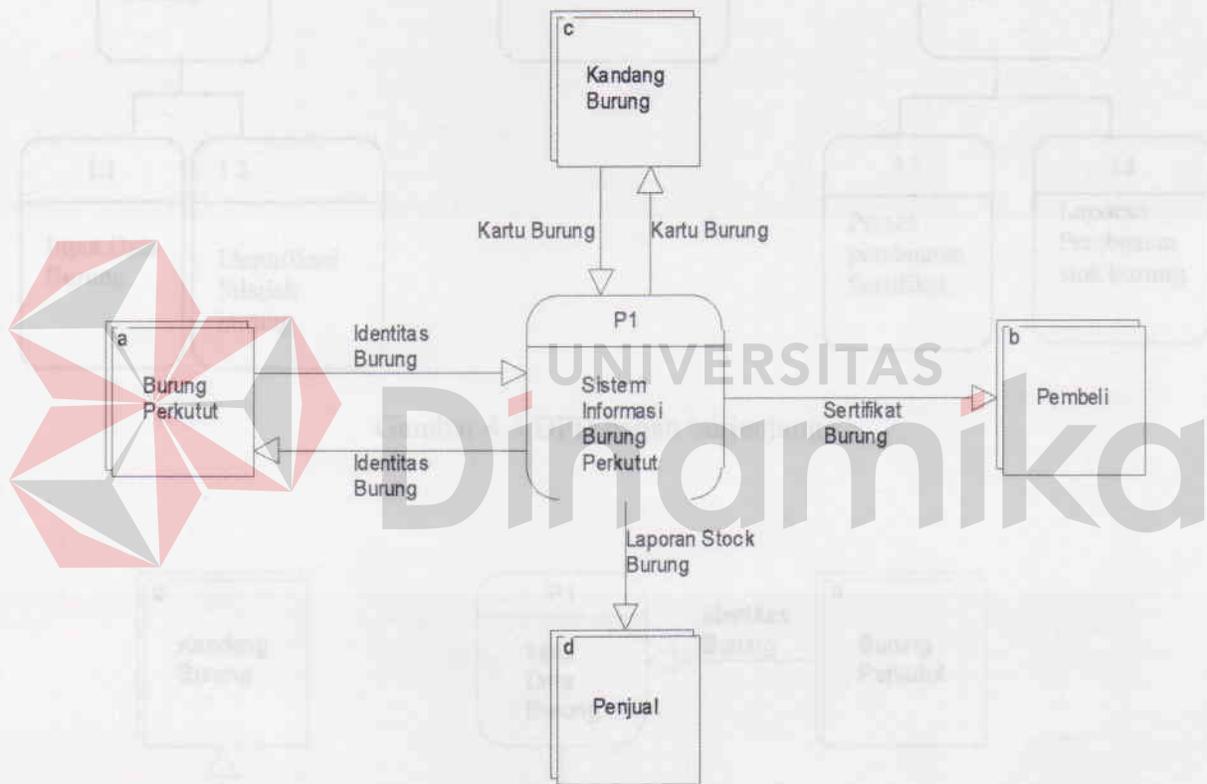


Gambar 4.1. Blok sistem diagram data burung

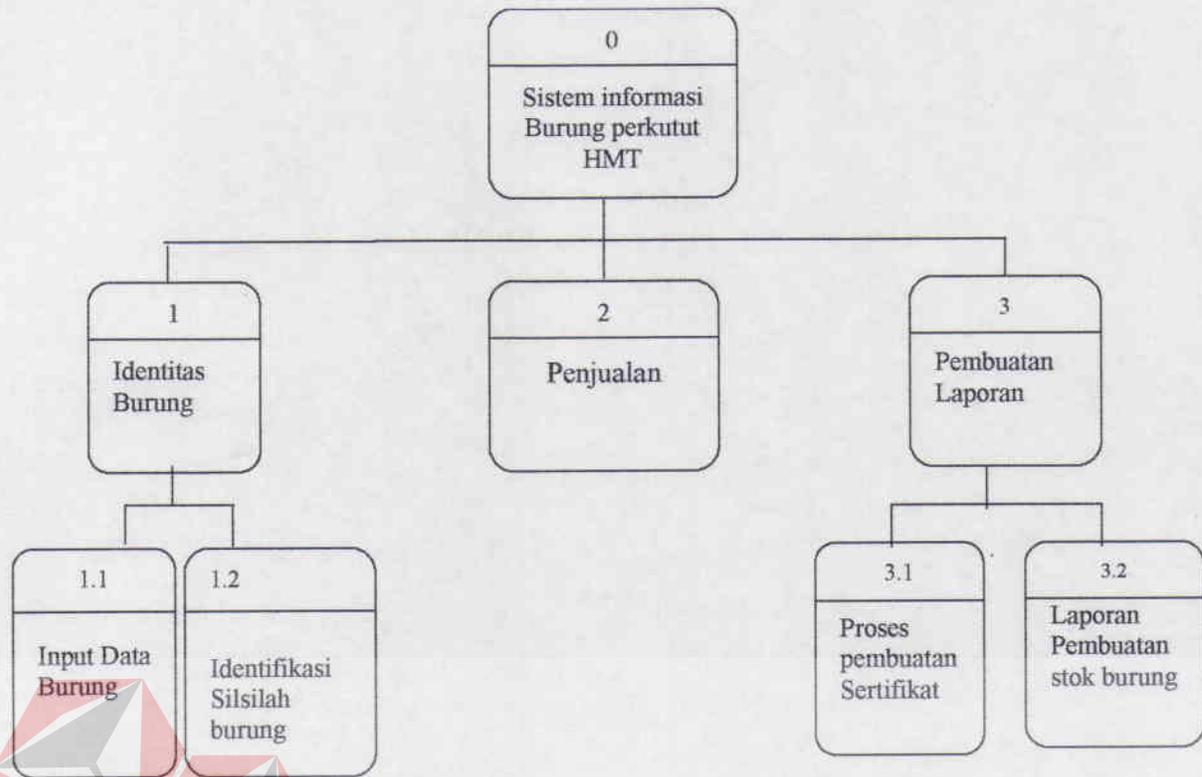
Blok diagram tersebut menggambarkan bahwa data-data burung perkutut dimasukkan ke komputer secara manual dan di simpan dalam perangkat keras komputer harddisk, kemudian diproses untuk menghasilkan sertifikat yang diperlukan pada saat yang diinginkan.

4.2. Data Flow Diagram (DFD)

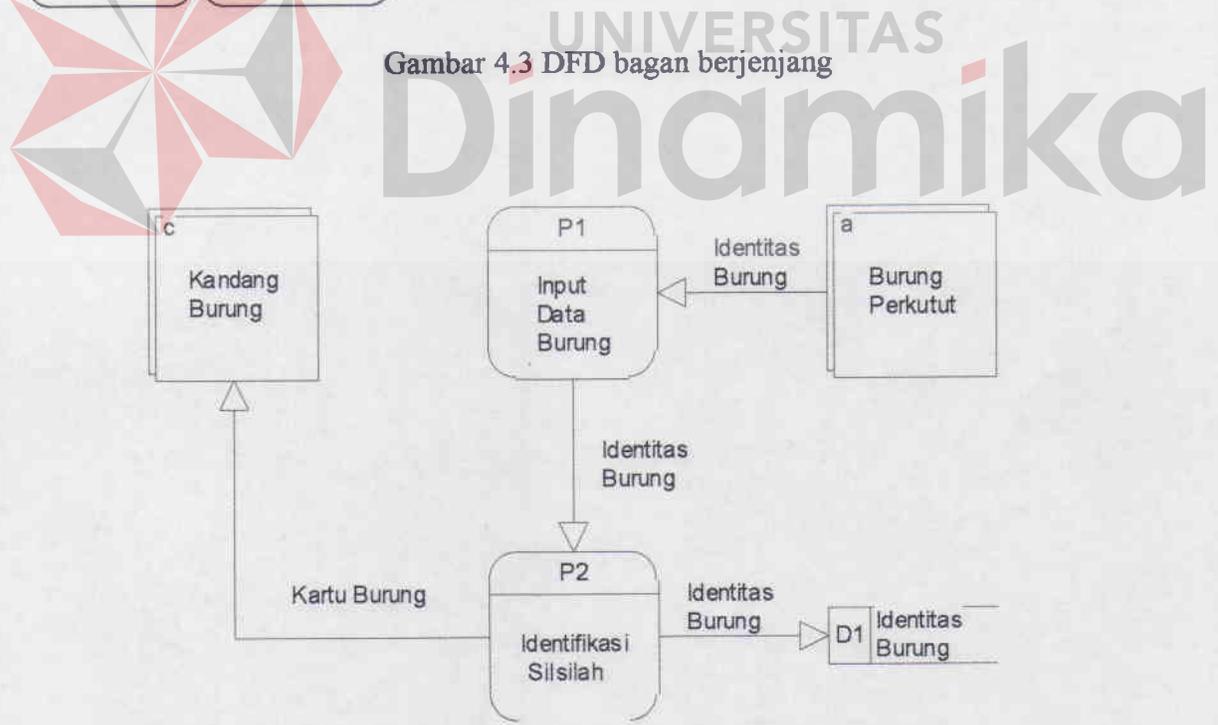
Dari analisis yang ada, sebagai solusi dapat digambarkan DFD tentang proses- proses yang ada, arus data yang mengalir, simpanan- simpanan data yang dibutuhkan beserta eksternal entiti yang berhubungan dengan sistem yang dibahas. Dalam DFD terdapat beberapa tahap yang dibuat. Adapun tahap- tahap tersebut adalah sebagai berikut :



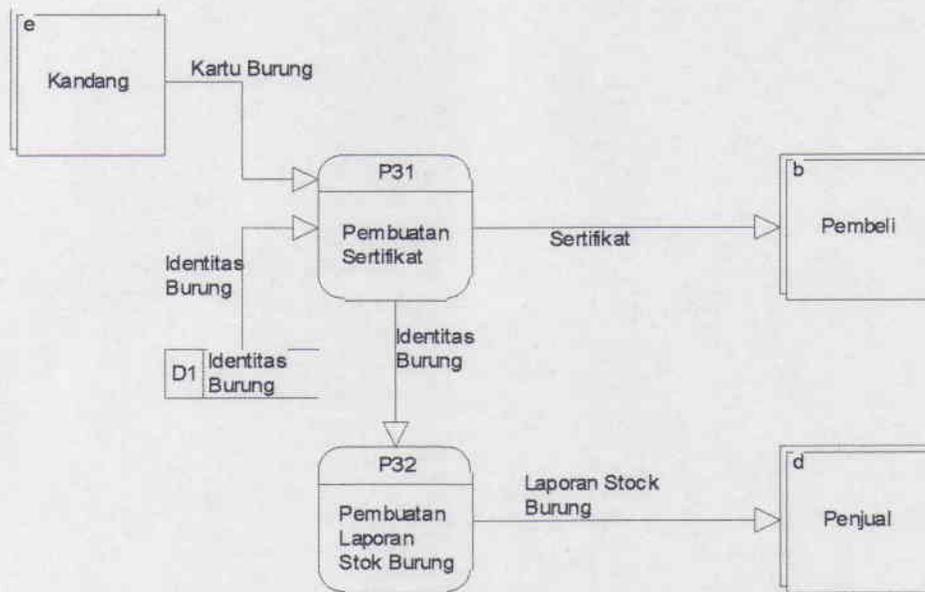
Gambar 4.2 Context Diagram



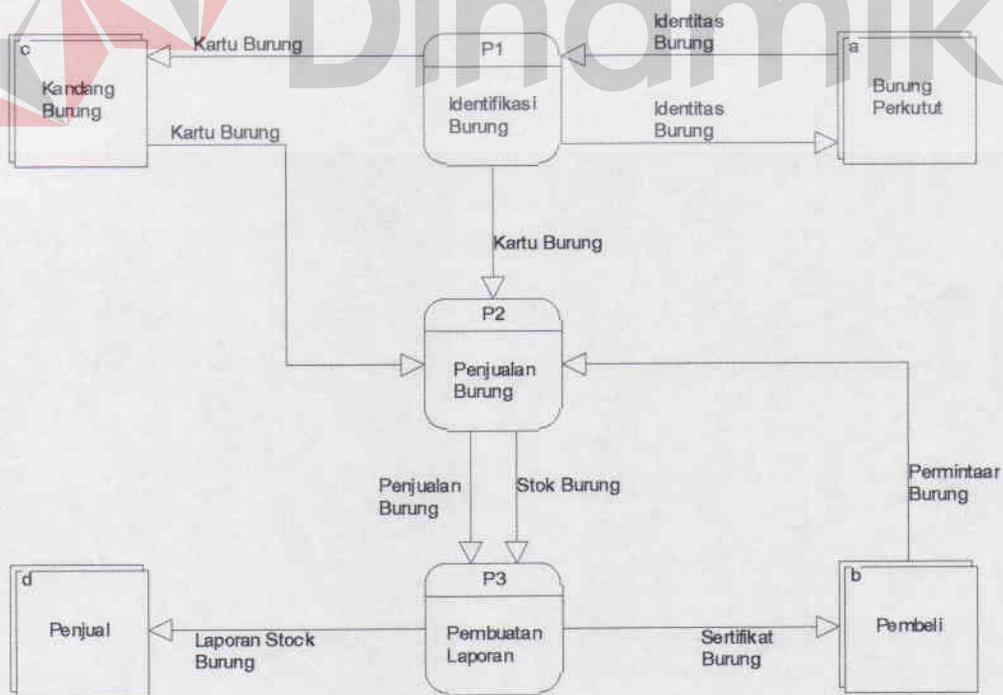
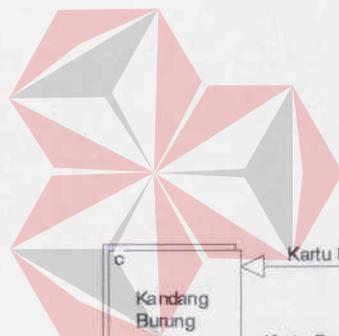
Gambar 4.3 DFD bagan berjenjang



Gambar 4.4 DFD level 1 Proses 1

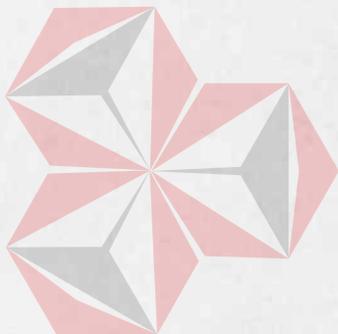
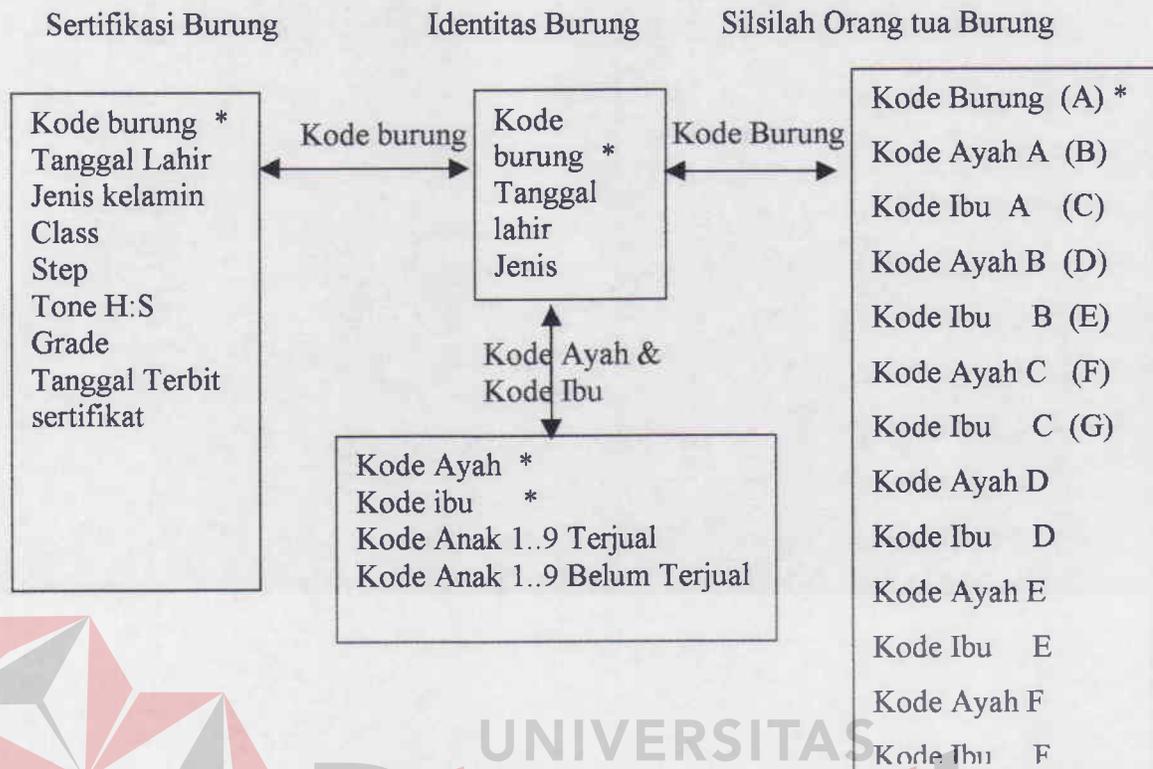


Gambar 4.5 DFD level 1 proses 3



Gambar 4.6 Overview Diagram

4.3. Entity Relationship Diagram



UNIVERSITAS
Dinamika

Gambar 4.7 Entity relationship

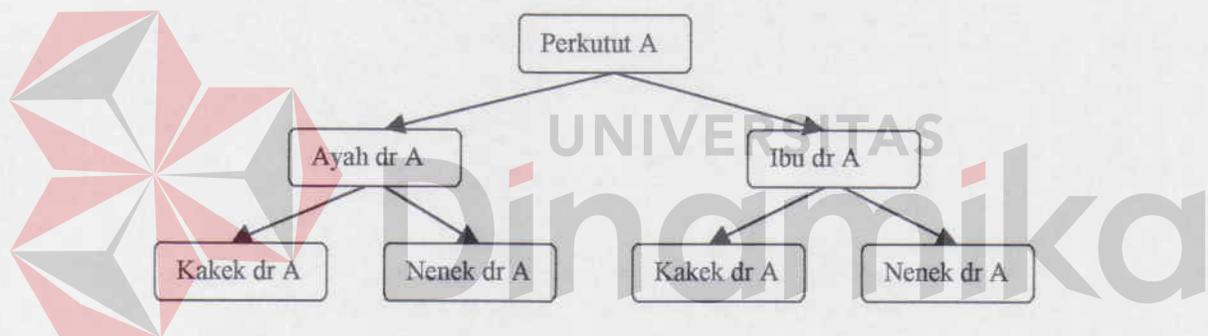
4.4. Pembuatan Data base

Untuk membuat program ini data base yang digunakan dan cocok yaitu model pohon biner. Karena masing-masing burung perkutut hanya memiliki dua buah penghubung yaitu perkutut ayah dan perkutut ibu. Sehingga model ini sangat cocok menggunakan data base berbentuk biner. Pohon biner (binary tree) bisa didefinisikan sebagai suatu kumpulan simpul yang mungkin kosong atau mempunyai akar dua sub pohon yang terpisah yang disebut dengan subpohon kiri, dan subpohon kanan. Subpohon juga disebut dengan cabang. Pohon biner merupakan tipe yang sangat penting dari struktur pohon, dan banyak dijumpai dalam berbagai terapan. Sesuai dengan definisi di atas, maka ada karakteristik

yang dimiliki oleh pohon biner, yaitu bahwa setiap simpul paling banyak hanya memiliki dua buah anak.

Sebuah pohon biner bisa dibuat dengan dua cara. Cara pertama adalah dengan cara non rekursif, dan cara kedua dengan cara rekursif. Untuk program Tugas Akhir ini pohon biner dibuat dengan cara non rekursif.

Data base dari burung perkutut akan selalu memiliki dua buah anak cabang jadi dapat dikatakan pohon biner yang sempurna, karena masing-masing perkutut akan memiliki dua buah anak cabang yang mempunyai arti sebagai ayah dan ibu dari perkutut tersebut. Untuk lebih jelasnya dapat dilihat pada gambar 4.8



Gambar 4.8. Pohon Biner Silsilah garis Keturunan Burung Perkutut

4.5. Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak ini dibagi menjadi tiga bagian yaitu:

1. Struktur File.
2. Sistem program.
3. Pembahasan program.

1. Struktur File

Program menggunakan dua buah file data base yang masing-masing antara

lain : 1. Nama file : Databird

File ini digunakan untuk menyimpan semua data mengenai burung

Keyfield : Kode burung

Struktur File :

Nam Field	Tipe Data	Panjang Field	Keterangan
Rkode	String	4	Kode Burung
Rnama	String	20	Nama Burung
Rtglhr	String	10	Tanggal Lahir
Rjnsklm	String	10	Jenis Kelamin
Rbapak	String	4	Kode Bapak
Ribu	String	4	Kode Ibu
Rkdg	String	6	Kode Kandang
Rtrjl	Boolean		

Kode burung merupakan identitas yang unik (tidak boleh sama) dan digunakan untuk mengenali burung perkutut tersebut. Kode Ayah dan kode ibu merupakan penghubung dengan file berikutnya yang berisikan keterangan dari burung perkutut yang menjadi ibu maupun ayah dari perkutut tersebut.

Sedangkan file berikutnya merupakan file dengan bentuk gambar. Jadi file tersebut akan disimpan dengan format gambar BMP (Bit Map Graphic). Isi dari

file ini adalah bentuk asli dari sertifikat burung perkutut tersebut. Adapun keterangan yang dapat disampaikan adalah sebagai berikut:

1. Header dari peternakan tersebut.
2. Tanggal Kelahiran.
3. Jenis Kelamin.
4. Class.
5. Step.
6. Tone H:S.
7. Grade.
8. Tanda tangan.
9. Tanggal diterbitkannya sertifikat.
10. Susunan hirarki dari burung perkutut tersebut sampai tiga generasi.

2. Sistem Program

Sistem program yang dibuat terdiri dari dua bagian besar yaitu sistem file

data base dan Informasi. Masing-masing terdiri dari :

Bagian File data base berisi :

- Update data.
- Edit data.
- Hapus data.
- Lihat data.

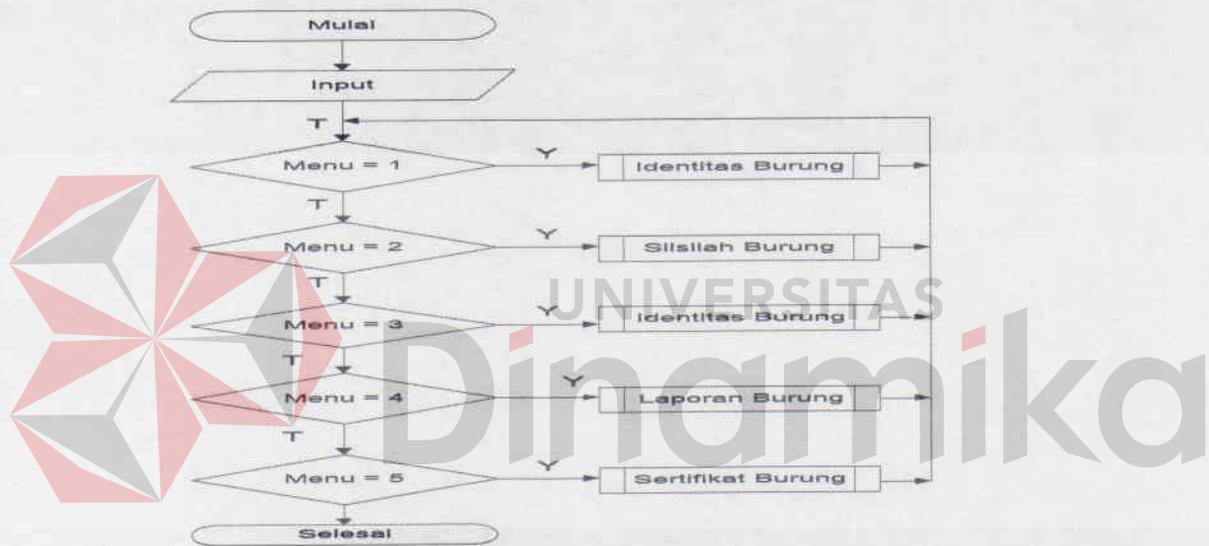
Bagian Informasi :

- Informasi tentang garis keturunan dari seekor burung perkutut.
- Informasi inventori perkutut yang telah terjual dari sepasang induk perkutut.

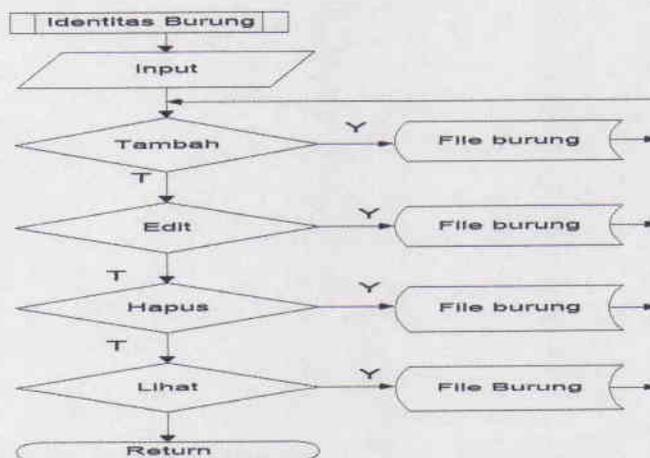
Pembuatan program ini terdiri dari empat bagian yaitu :

1. Identitas Burung.
2. Mencari silsilah orang tua burung.
3. Inventori.
4. Laporan Burung
5. Membuat sertifikat burung.

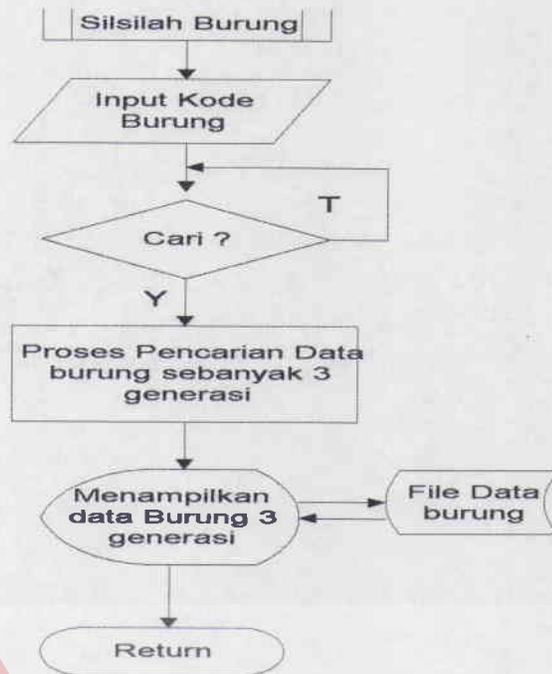
Berikut ini adalah diagram alur dari program yang dibuat :



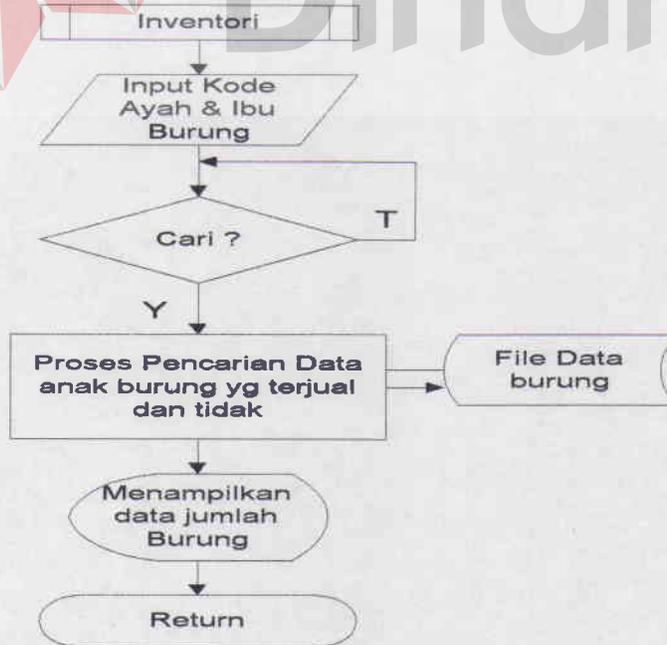
Gambar 4.9 Diagram Alur Program Utama



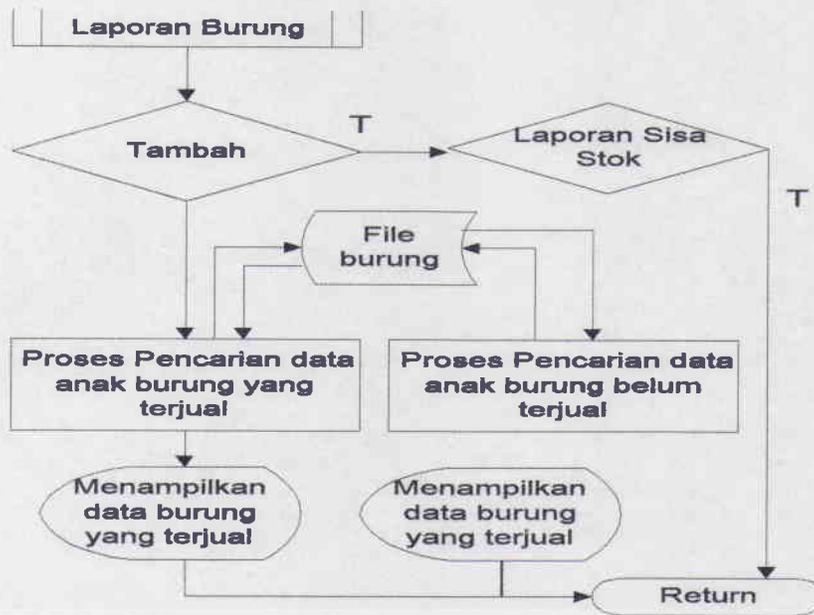
Gambar 4.10 Diagram Alur Sub Menu Identitas Burung.



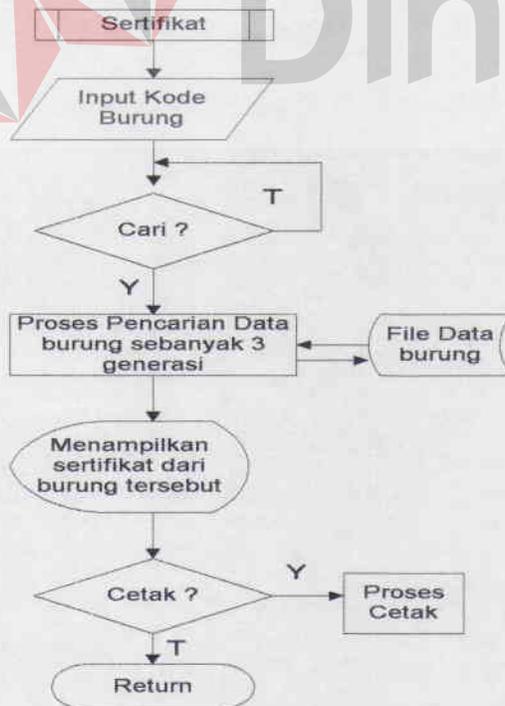
Gambar 4.11 Diagram Alur Sub Menu Silsilah Burung



Gambar 4.12 Diagram Alur Sub menu Inventori



Gambar 4.13 Diagram Alur Sub menu Laporan Burung



Gambar 4.14 Diagram Alur Sub Menu Sertifikat

3. Pembahasan Program

Pembuatan program peternakan burung perkutut ini diletakan pada direktori utama yaitu dengan nama Farmbird kemudian pada direktori FarmBird dibagi menjadi dua bagian sub direktori yaitu subdirektori Databird dan sub direktori DataSertifikat. Sub direktori Databird digunakan untuk menyimpan file data burung sedangkan sub direktori DataSertifikat digunakan untuk menyimpan data-data sertifikat yang pernah dibuat. Dengan peletakan yang terpisah akan memudahkan perawatan dari file-file yang dibuat.

Letak Program.

\FarmBird.

\FarmBird\DataBird.

\FarmBird\DataSertificate.

4.6. Sistem Menu

Hubungan antara komputer dengan pemakai sistem juga programmer terwujud pada desain dialog menu. Hubungan yang dimaksud adalah interface tampilan di layar komputer dari software. Software yang dibangun untuk Sistem informasi ini menggunakan *interface grafis (Graphical User Interface)*. Perancangan sistem menu ini terdiri dari menu utama dengan model *pull-down menu*, untuk kemudahan pengorganisasian proses. Sistem menu ini juga menentukan mudah atau sulitnya pemakaian suatu program.

Pada program aplikasi sistem informasi ini terdiri dari Menu Utama, yang mempunyai beberapa menu dan memiliki sub menu. Menu-menu tersebut terdiri dari sub menu yang terorganisasi sehingga membentuk struktur hirarki. Masing-

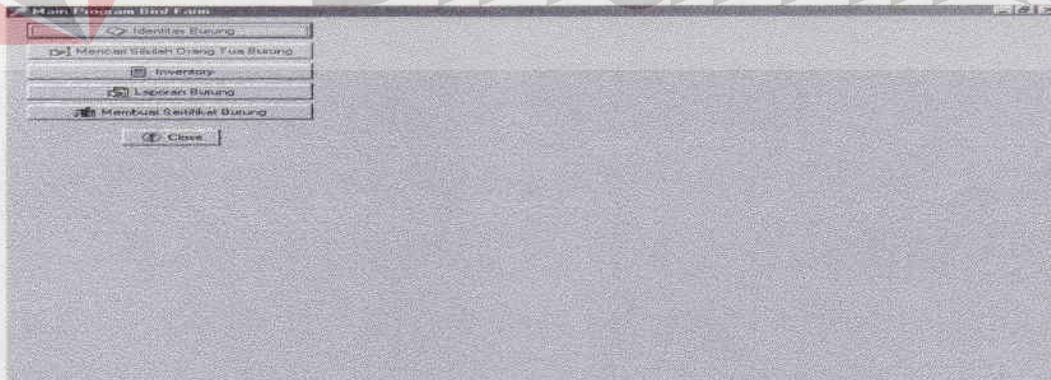
masing menu akan memanggil program-program tertentu untuk melakukan beberapa proses yang membentuk sistem tersebut.

Menu utama program Peternakan Perkutut ini terdiri dari empat bagian :

- 1 Identitas Burung.
- 2 Mencari Silsilah Orang Tua Burung.
- 3 Inventori.
- 4 Laporan Burung.
- 5 Membuat Sertifikat Burung.
- 6 Close.

Menu utama merupakan bagian yang sangat pokok karena semua bermuara pada bagian ini dan untuk keluar dari program juga melalui bagian ini dengan menekan tombol close.

Pada saat program diaktifkan, akan ditampilkan menu utama



Gambar 4.15 Menu Utama

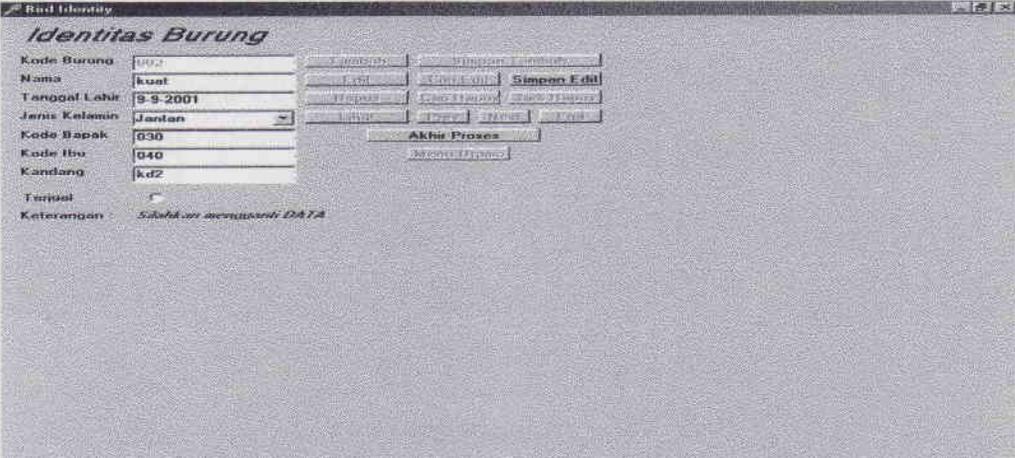
1. Identitas Burung

Pada sub menu Identitas Burung pemakai dapat melangsungkan proses sebagai berikut:

1. Penambahan Data Burung
2. Pembetulan Data Burung
3. Penghapusan Data Burung
4. Melihat Data Burung
5. Kembali ke Menu Utama

Penambahan Data burung merupakan proses untuk menambah record berupa data burung yaitu Kode Burung, Tanggal Lahir, Jenis Kelamin, Kode bapak, Kode Ibu dan Kandang. Data-data ini akan disimpan dalam file dengan nama MasterBird.

Pembetulan Data burung digunakan untuk mengedit record yang telah ada, dikarenakan terjadi perubahan data, atau kesalahan pengetikan pada saat memasukkan data burung tersebut. Penghapusan Data burung digunakan untuk menghilangkan data yang telah ada misalnya karena burung tersebut mati. Melihat data digunakan untuk meneliti ulang data yang telah dimasukkan oleh pengguna sehingga dapat dipastikan kebenarannya.



The screenshot shows a web application window titled "Bird Identity" with a sub-header "Identitas Burung". The form contains the following fields and controls:

Kode Burung	1002	Simpan	Simpan	Simpan
Nama	kuat	Simpan	Simpan	Simpan Edit
Tanggal Lahir	9-9-2001	Simpan	Simpan	Simpan
Jenis Kelamin	Jantan	Simpan	Simpan	Simpan
Kode Bapak	030	Akhir Proses		
Kode Ibu	040	Simpan		
Kandang	kd2	Simpan		
Tanggal				
Keterangan	Silahkan masukkan DATA			

Gambar 4.16 Menu Identitas Burung

2. Mencari Silsilah Orang Tua Burung

Bagian ini digunakan untuk melihat silsilah dari seekor burung perkutut sampai dengan tiga generasi. Untuk menampilkan data tersebut pemakai harus memasukkan kode burung yang akan dilihat silsilahnya.

Program akan mencari data dari kode burung tersebut jika ketemu maka akan dilanjutkan dengan mencari data berikutnya melalui kode ayah dan kode ibu secara terus menerus sampai tiga generasi ditemukan.

Algoritma dari proses ini adalah :

- Langkah 1. Cari Kode Burung Yang dimasukkan oleh pengguna.
- Langkah 2. Jika tidak ditemukan Selesai.



Jika ditemukan lanjut

Vsilsilah[1]=KodeBurung.

Vsilsilah[2]=KodeAyah.

Vsilsilah[3]=KodeIbu.

- Langkah 3. Ulangi dari I:=2 ke 7.

Jika ayah/ibu tidak diketahui maka kosong

Jika Vsilsilah[I] <> " maka

Cari data dengan kode Vsilsilah[I]

Jika diketemukan,

Vsilsilah[I*2]=Kode Bapak

Vsilsilah[I*2+1]=KodeIbu

- Langkah 4. Tampilkan data yang telah ditemukan.
- Langkah 5. Selesai.

The screenshot shows a web application interface with the following elements:

- Title: *Silsilah Orang Tua Burung*
- Form fields:
 - Mak: [] Male
 - Ibu: [] Female
 - Kode Burung: []
 - Mak: [] Male
 - Ibu: [] Female
 - Mak: [] Male
 - Ibu: [] Female
 - Mak: [] Male
 - Ibu: [] Female
- Buttons: and
- Note: *Masukkan Kode Burung Dalam List*

Gambar 4.17 Menu Silsilah Burung



3. Inventori

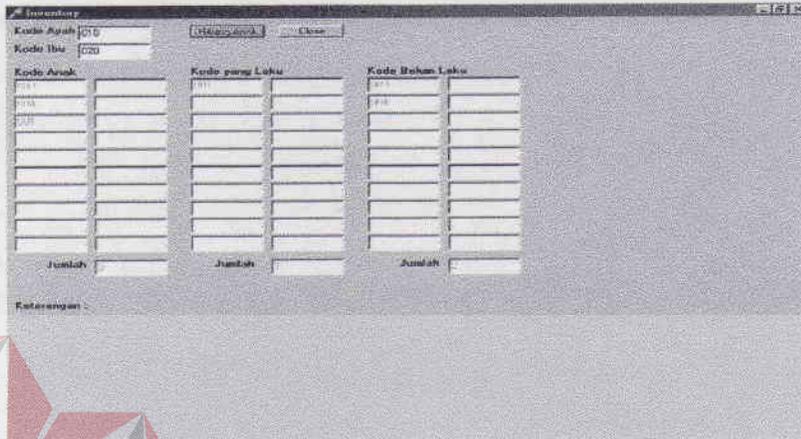
Pada bagian ini pengguna dapat mencari jumlah akan dari sepasang burung perkutut yang telah laku dan yang belum laku dengan memasukkan kode ayah dan kode ibu yang akan dicari.

Program akan mencari data dari burung perkutut yang mempunyai kode ayah dan kode ibu sama dengan yang dimasukkan maka akan mengecek kode terjual dari masing-masing record yang ada kemudian ditampilkan pada layar komputer.

Algoritma dari Program ini adalah :

- Langkah 1. Masukan kode Ayah dan kode Ibu.
- Langkah 2. Cari record burung dengan Kode Ayah dan Kode Ibu yang Sama.
- Langkah 3. Cek field terjual jika ya masukkan kode burung ke bagian t, Jika tidak masukkan kode burung ke bagian tidak terjual.

- Langkah 4. Ulangi langkah 2 sampai data burung habis jika habis Lanjutkan ke langkah 5.
- Langkah 5. Tampilkan data burung terjual dan yang tidak terjual.
- Langkah 6. Selesai.



Gambar 4.18 Menu Inventori Burung

4. Membuat Laporan burung

Pada bagian ini pengguna dapat mencari jumlah keseluruhan dari data burung per kutut yang telah terjual dan yang belum terjual.

Program menyediakan dua menu yaitu terjual dan sisa stock. Menu terjual akan menampilkan seluruh data burung yang telah terjual, sedangkan menu sisa stock akan menampilkan semua data burung yang belum terjual.

Algoritma dari Program ini adalah :

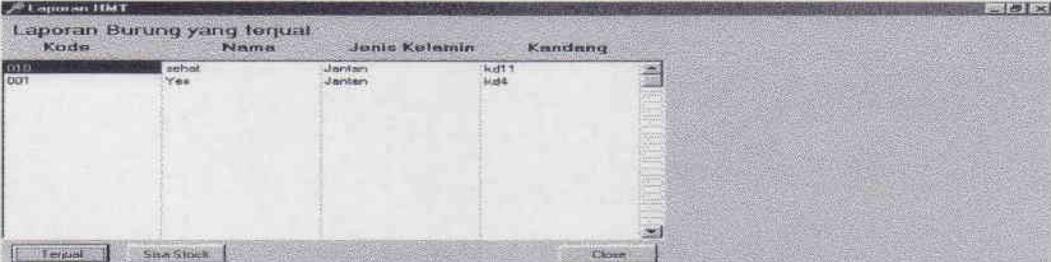
- Langkah 1: Masukan Pilih menu terjual ke langkah 2 sisa stock ke langkah 4.
- Langkah 2. Terjual Cari semua record yang tanda terjual = true.

- Langkah 3. Tampilkan semua data burung yang terjual ke layar. Ke langkah 6.
- Langkah 4. Sisa stock Cari semua record yang tanda terjual = false.
- Langkah 5. Tampilkan semua data burung tidak terjual ke layar. Ke langkah 6.
- Langkah 6. Selesai.



Kode	Nama	Jenis Kelamin	Kandang
002	kuat	Jantan	kd2
030	vovwan	Jantan	kd15
030	han	Jantan	kd9
040	nio	Betina	
050	vinomnt	Jantan	kd9
060	sindy	Jantan	kd1
000	nico	Jantan	
081	han	Jantan	kd9
082	han	Betina	kd10
070	jack	Jantan	
070	pones	Jantan	
090	dinda	Betina	

Gambar 4.19 Menu Laporan Burung yang belum terjual



Kode	Nama	Jenis Kelamin	Kandang
000	sehat	Jantan	kd11
001	Yes	Jantan	kd4

Gambar 4.20 Menu Laporan burung yang terjual

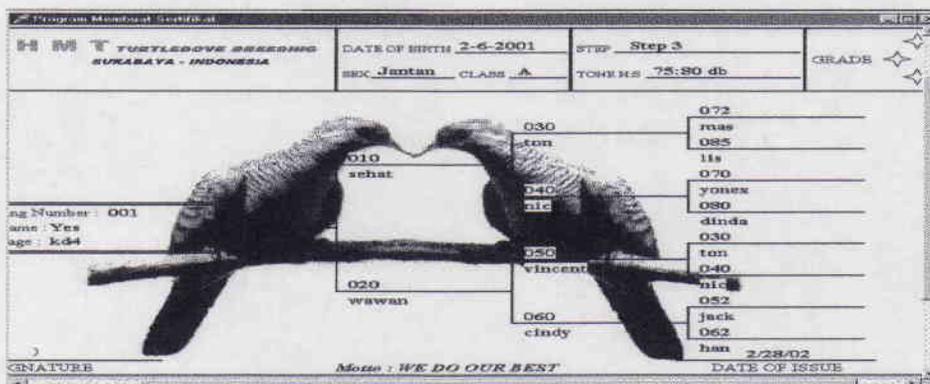
5. Membuat sertifikat burung.

Pada bagian ini yang dapat dilakukan oleh pengguna adalah membuat sertifikat burung perkutut dan dengan mencetak melalui peralatan printer.

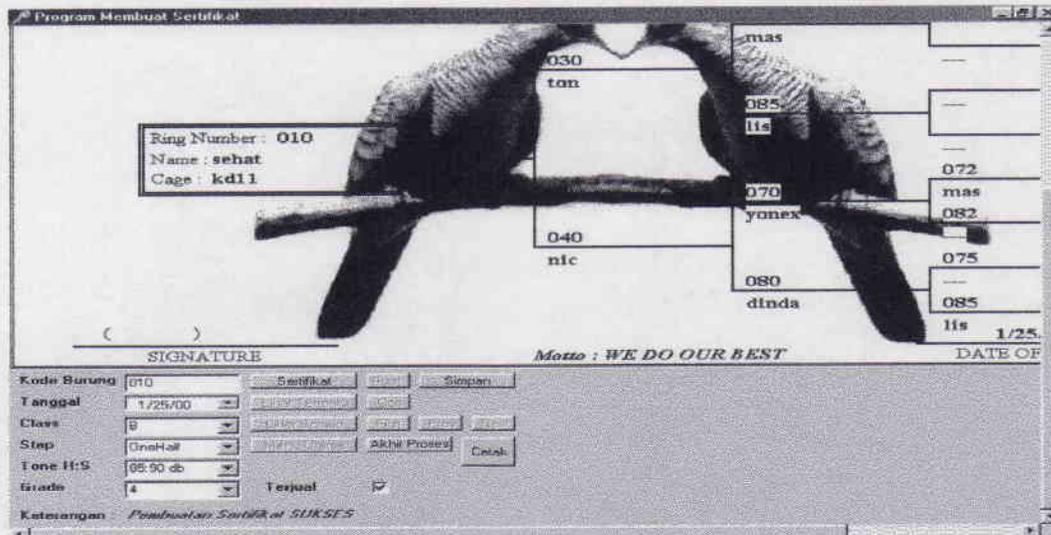
Bagian ini pengguna dapat melakukan :

1. Pembuatan Sertifikat + Cetak.
2. Melihat Sertifikat Tertentu + Cetak.
3. Melihat Semua Sertifikat + Cetak.
4. Kembali ke Menu Utama.

Pembuatan sertifikat pengguna tinggal memasukkan kode burung yang akan di buat sertifikat. Kemudian memasukkan klasifikasi dari burung tersebut melalui pengamatan pada saat terakhir yaitu Class, step, Tone H:S dan grade. Bagian-bagian ini pada setiap burung berlainan dan dapat berubah setiap saat karena faktor tertentu seperti sangkar habis jatuh, pasangan mati ini juga mempengaruhi nilai dari perkutut tersebut. Melihat sertifikat untuk mengamati sertifikat yang telah dibuat dan disimpan dalam hardisk. Ada dua cara melihat tertentu dengan memasukkan data kode burung tertentu atau melihat secara keseluruhan dan berurutan.



Gambar 4.21 Hasil Print Sertifikat



Gambar 4.21 Menu Sertifikat Burung

4.7. Panduan untuk menjalankan Program

Letak Program :

\FarmBird.

\FarmBird\DataBird.

\FarmBird\DataSertificate.

Cara Penggunaan Program :

- Langkah 1 : Tekan Icon (Gambar). Untuk masuk ke Awal Program.
- Langkah 2 : Tekan Tombol 'Entry the program' Untuk masuk ke Menu Program.

Menu Program Terdiri dari 5 Bagian yaitu :

1. Identitas Burung.
2. Mencari Silsilah Orang Tua Burung.
3. Inventori.
4. Laporan Burung.

5. Membuat Sertifikat Burung.
6. Close.

1. Identitas Burung

Bagian ini pengguna dapat melakukan :

1. Penambahan Data Burung.
2. Pembedulan Data Burung.
3. Penghapusan Data Burung.
4. Melihat Data Burung.
5. Kembali ke Menu Utama.

Cara Menambah Data Burung :

- Langkah 1 : Tekan Tombol 'Tambah'.
- Langkah 2 : Masukkan Data Burung (Kode Burung, Tgl Lahir, Jenis Kelamin, Kode Bapak, dan Kandang).
- Langkah 3 : Jika Data Burung ingin disimpan maka tombol 'Simpan Tambah'. Jika batal tekan tombol 'Akhir Proses'.
- Langkah 4 : Selesai.

Cara Pembedulan Data Burung

- Langkah 1 : Tekan Tombol 'Edit'.
- Langkah 2 : Masukkan Kode Burung yang akan di edit ke kolom Kode Burung.
- Langkah 3 : Tekan Tombol 'Cari Edit'. Jika Kode Burung ada akan ditampilkan seluruh data burung tersebut. Jika Kode Burung tidak ada maka data tampilan akan kosong. Untuk mengulangi Pembedulan

Lakukan kembali langkah 1. Untuk Selesai tekan tombol 'Akhir Proses'.

- Langkah 4 : Jika Kode Burung ada maka dapat untuk mengganti data-data Tersebut. Setelah itu tekan tombol 'Simpan Edit' untuk menyimpannya.
- Langkah 5 : Jika ingin merubah lagi ke Langkah1. Jika selesai tekan Tombol 'Akhir Proses'.
- Langkah 6 : Selesai.

Cara Menghapus Data Burung :

- Langkah 1 : Tekan Tombol 'Hapus'.
- Langkah 2 : Masukkan Kode Burung yang akan dihapus ke kolom kode Burung.
- Langkah 3 : Tekan Tombol 'Cari Hapus'. Jika Kode Burung ada akan ditampilkan seluruh data dari burung tersebut. Jika Kode Burung tidak ada maka data tampilan akan kosong. Untuk mengulangi Penghapusan lakukan kembali langkah 1. Untuk Selesai tekan tombol 'Akhir Proses'.
- Langkah 4 : Jika Kode Burung ada dan yakin akan dihapus tekan tombol ' Jadi 'Hapus'. Jika tidak yakin tekan tombol 'Akhir Proses' untuk selesai.
- Langkah 5 : Jika ingin menghapus lagi ke Langkah1. Jika selesai tekan tombol 'Akhir Proses'.
- Langkah 6 : Selesai.



UNIVERSITAS
Dinamika

Cara Lihat Data Burung :

- Langkah 1 : Tekan tombol 'Lihat'.
- Langkah 2 : Tekan tombol 'Next' untuk Data Burung berikut. Tekan tombol 'Prev' untuk Data Burung sebelumnya. Tekan tombol 'End' untuk selesai.
- Langkah 3 : Selesai.

Cara Kembali ke Menu Utama :

- Langkah 1 : Tekan tombol 'Menu Utama'.
- Langkah 2 : Selesai.

2. Mencari Silsilah Orang Tua Burung

Bagian ini pengguna dapat melakukan :

1. Melihat Silsilah Orang Tua Burung.
2. Kembali ke Menu utama.

Cara Melihat Silsilah Orang Tua Burung :

- Langkah 1 : Masukkan Kode Burung.
- Langkah 2 : Tekan tombol 'Cari'. Jika ada maka akan ditampilkan data-data dari Orang tua burung tersebut. Jika tidak ada data akan kosong. Karena data belum dimasukkan ke komputer. Lihat Cara Menambah Data Burung. Untuk mengulangi Pencarian lakukan kembali langkah 1. Untuk Selesai tekan tombol 'Akhir Proses'.
- Langkah 3 : Selesai.

Cara Kembali ke Menu Utama :

- Langkah 1 : Tekan tombol 'Menu Utama'.
- Langkah 2 : Selesai.

3. Inventori

Bagian ini pengguna dapat melakukan :

1. Melihat Inventori anak perkutut dari sepasang burung perkutut.
2. Kembali ke Menu utama.

Cara melihat inventori :

- Langkah 1 : Masukkan Kode Burung ayah dan ibu.
- Langkah 2 : Tekan tombol 'Hitung Anak'. Jika ada maka akan ditampilkan data-data anak dari Orang tua burung tersebut. Jika tidak ada data akan kosong. Karena data belum dimasukkan ke komputer.

Lihat Cara Menambah data Burung. Untuk mengulangi Pencarian

lakukan kembali langkah 1. Untuk Selesai tekan tombol 'Close'.

- Langkah 3 : Selesai.

Cara Kembali ke Menu Utama :

- Langkah 1 : Tekan tombol 'Colse'.
- Langkah 2 : Selesai.

4. Laporan Burung

Bagian ini pengguna dapat melakukan:

1. Melihat semua data burung yang terjual.
2. Melihat semua data burung yang belum terjual.

Cara melihat Data burung terjual :

- Langkah 1 : Tekan tombol terjual.
- Langkah 2 : Menampilkan data burung terjual.
- Langkah 3 : Selesai.

Cara melihat Data burung belum terjual :

- Langkah 1 : Tekan tombol sisa stock.
- Langkah 2 : Menampilkan data burung belum terjual.
- Langkah 3 : Selesai.

Cara Kembali ke Menu Utama :

- Langkah 1 : Tekan tombol 'Close'.
- Langkah 2 : Selesai.

5. Membuat Sertifikat Burung.

Bagian ini pengguna dapat melakukan

- 1 Pembuatan Sertifikat + Cetak.
- 2 Melihat Sertifikat Tertentu + Cetak.
- 3 Melihat Semua Sertifikat + Cetak.
- 4 Kembali ke Menu Utama.

Cara Pembuatan Sertifikat + Cetak

- Langkah 1 : Tekan tombol 'Sertifikat'.
- Langkah 2 : Masukkan Kode Burung, Tgl, Class, Step, Tone H:S, Grade.
- Langkah 3 : Tekan tombol 'Buat'. Jika Kode Burung ada maka sertifikat akan tampak. Jika tidak ada data sertifikat akan kosong. Karena data belum dimasukkan ke komputer. Lihat Cara Menambah Data Burung.

- Langkah 4 : Jika ada dan ingin disimpan tekan tombol 'Simpan' atau ingin dicetak tekan tombol 'Cetak'.
- Langkah 5 : Jika ingin buat sertifikat lagi ke Langkah 1. Jika Selesai tekan tombol 'Akhir Proses'.
- Langkah 6 : Selesai.

Cara Melihat Sertifikat Tertentu + Cetak :

- Langkah 1 : Tekan tombol 'Lihat Tertentu'.
- Langkah 2 : Masukkan Kode Burung.
- Langkah 3 : Tekan tombol 'Cari'. Jika ada maka akan ditampilkan sertifikat tersebut. Jika tidak ada Sertifikat akan kosong.

Karena sertifikat belum dimasukkan ke komputer. Lihat. Cara Pembuatan Sertifikat. Untuk mengulangi pencarian lakukan kembali langkah 1. Untuk Selesai tekan tombol 'Akhir Proses'.

- Langkah 4 : Jika Sertifikat Ada. Tekan tombol 'Cetak' untuk Mencetak.
- Langkah 5 : Selesai.

Cara Melihat Semua Sertifikat + Cetak :

- Langkah 1 : Tekan tombol 'Lihat Semua'.
- Langkah 2 : Tekan tombol 'First'.
- Langkah 3 : Tekan tombol 'Next' untuk sertifikat berikut. Tombol 'Prev' untuk sertifikat sebelum, Tombol 'Cetak' untuk Cetak Sertifikat, Tombol 'Akhir Proses' untuk selesai.
- Langkah 4 : Selesai.

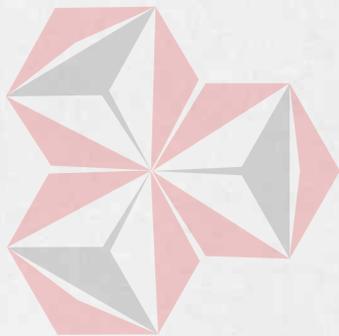
Cara Kembali ke Menu Utama :

- Langkah 1 : Tekan tombol 'Menu Utama'.
- Langkah 2 : Selesai.

6. Close.

Untuk keluar dari program.

- Langkah 1 : Tekan tombol 'Close'.
- Langkah 2 : Selesai.



UNIVERSITAS
Dinamika

BAB V

PENUTUP

5.1. Kesimpulan

Dari keseluruhan penelitian, pembahasan, pembuatan tugas akhir ini, dapat diambil beberapa kesimpulan sebagai berikut :

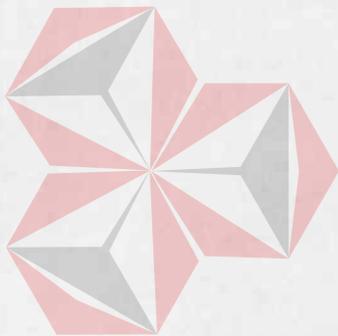
1. Dengan dibuatnya program sistem informasi pada peternakan burung perkutut ini sangat dapat membantu terhadap perkembangan peternakan tersebut. Terutama untuk mencari garis keturunan dari seekor burung perkutut dan hasil peternakan.
2. Komputerisasi Sistem pembuatan sertifikat dengan menggunakan komputer menaikkan status dari peternakan perkutut tersebut. Karena penggunaan sertifikat dengan sistem komputer akan kelihatan lebih menarik dan masih jarang digunakan di Indonesia kecuali peternakan di luar negeri, disamping itu untuk memudahkan pelacakan.
3. Inventory anak burung dari masing-masing pasangan sangat mudah dihitung. sehingga dapat memberikan informasi yang sangat berharga.

5.2. Saran

Pada kesempatan kali ini penulis ingin menyampaikan beberapa saran yang bermanfaat dan membangun, antara lain sebagai berikut :

1. Dalam pengembangan berikutnya penulis mengharapkan program dapat digabungkan dengan proses penjualan.

2. Sistem Informasi ini ditambahkan dengan jumlah biaya perawatan yang dikeluarkan untuk membantu menentukan harga jual.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

Charles Calvert, 1995, Delphi Programming Unleashed, SAM Publishing, 201 West 103rd Street Indianapolis. USA

Todd Bill and Vince Kellen, 1995, Delphi A Developer Guide, M&T Books, 115 West 18th Street, New York 10011.

Blake Watson, 1995, Delphi by Example, Que Corporation, 201 West. 103rd Street Indianapolis. USA

Ken Henderson, 1996, Database developer's guide with delphy, SAMS Publishing

David R Faulkner, 1996, Panduan Penggunaan Delphi, PT. Andy Yogyakarta

Insap Santosa, MSc, 1995, Struktur Data menggunakan Turbo Pascal 6.0, Andy Offset Yogyakarta

Raymond Mcleod Jr, 2001, Management Information System, Prentice-Hall

B. Sarwono, 1999 Sukses Dalam Konkurs Perkutut, Penebar Surabaya