

RANCANG BANGUN APLIKASI SISTEM PAKAR UNTUK KERUSAKAN KOMPUTER DENGAN METODE *BACKWARD CHAINING*

Oleh:

Minarni, S. Si., M.T; Rahmad Hidayat, ST

Jurusan Teknik Informatika , Fakultas Teknologi Industri

Institut Teknologi Padang

Jl. Gajah Mada, Kandis Nanggalo, Padang

Abstract

The computer is a machine that can be damaged in carrying out its functions. To provide such information, it takes knowledge of a computer expert. Expert system designed as a computer program, so the expert system program can provide information like a computer expert who always stand beside the computer itself.

Method is by using backward chaining, expert system for computer do trace to retreat wet backward, starting from the hypothesis and then looking for evidence supporting or opposing.

Expert system for computer damage will provide information and solutions to computer users on issues relating to damage the hardware computer quickly.

Keywords : Expert Systems, Computer Damage, Backward Chaining

1. Pendahuluan

Komputer atau sering kali disebut *PC* (*Personal Komputer*) terdiri dari dua bagian penting yaitu perangkat keras (*Hardware*) dan perangkat lunak (*Software*), bagian pertama adalah *Hardware* komputer berupa alat yang dapat di lihat dan di pegang seperti: *monitor, keyboard, mouse, printer, CPU* (*Central Processing Unit*) dan semua perangkat didalamnya. Bagian kedua yang tidak kalah pentingnya adalah *Software* yaitu berupa kumpulan instruksi yang berupa kode-kode dalam bahasa mesin yang dapat dijalankan oleh sistem komputer, *software* ini terdiri dari sistem operasi dan program aplikasi.

Pada penggunaanya terkadang komputer mengalami permasalahan yang membuat kinerja komputer tidak berjalan optimal. Gangguan tersebut sering disebabkan oleh persoalan *hardware* dalam CPU. Persoalan-persoalan *hardware* bagi pengguna komputer pemula akan dianggap sebagai kerusakan komputer secara keseluruhan, dan kemudian langsung menyerahkan kepada teknisi komputer untuk diperbaiki.

Kerusakan komputer biasanya hanya disebabkan oleh salah satu komponen komputer, seperti *memori RAM, hardisk, VGA card, dll.* Bahkan sebenarnya ada kondisi kerusakan ringan yang dianggap fatal oleh pengguna, seperti kurang pasnya pemasangan *memori RAM, hardisk* dan

bagian lain. Untuk mengetahui lebih detail komponen yang mengalami permasalahan dirasa perlu sebuah pengetahuan yang dapat memberikan informasi kepada pengguna komputer. Sehingga penggunapun dapat mencari solusi sendiri untuk menyelesaikan persoalan komputernya.

Salah satu teknik untuk membuat komputer mampu mengolah pengetahuan ini disebut teknik kecerdasan buatan (*Artificial Intelligence Technique*). Dengan pendekatan ini manusia mencoba membuat komputer dapat berpikir seperti cara yang dipakai manusia dalam memecahkan masalah komputer.

Dengan latar belakang masalah tersebut maka penulis tertarik untuk menyajikan judul “ Rancang Bangun Aplikasi Sistem Pakar Untuk Kerusakan Komputer Dengan Metode *Backward Chaining* ”

Berdasarkan latar belakang permasalahan yang dibahas diatas penulis mencoba merumuskan masalah yaitu “ bagaimana merancang sistem pakar untuk kerusakan komputer yang mampu memberikan informasi mengenai kerusakan komputer ”.

Untuk lebih terarahnya ruang lingkup penelitian ini, maka penulis mempersempit pembahasan permasalahan yaitu ” membangun suatu *database* berbagai macam jenis kerusakan komputer dengan menggunakan *Visual Basic* “ dan

menitikberatkan pada kerusakan CPU (*Central Processing Unit*) saja, sehingga pengguna dapat mengetahui kerusakan komputer melalui sistem pakar ini.

Tujuan dari penelitian tugas akhir ini adalah merancang sebuah aplikasi sistem pakar untuk mengetahui kerusakan komputer.

Manfaat dari penelitian tugas akhir ini diantaranya yaitu pengguna komputer dapat mengetahui dengan cepat permasalahan *hardware* komputer serta memberikan solusi kepada pengguna terhadap kerusakan komputer yang dialami.

2. Tinjauan Pustaka

2.1 Studi Literatur

Gunanjar Wiro Sasmito, dalam tesisnya yang berjudul “ Aplikasi Sistem Pakar Untuk Simulasi Diagnosa Hama dan Tanaman Bawang Merah dan Cabai Menggunakan *Forward chaining* dan Pendekatan Berbasis Aturan”. Dalam tesis ini simulasi diagnosa menggunakan metode *Forward chaining* pada mesin inferensi sistem pakar.

Aprilia Sulistyohati, Taufik Hidayat. Dalam skripsinya yang berjudul “ Aplikasi Sistem Pakat Diagnosa Penyakit Ginjal Dengan Metode *Dempster-Shafer* ”. Aplikasi sistem pakar ini menghasilkan keluaran berupa kemungkinan penyakit ginjal yang diderita berdasarkan gejala yang dirasakan oleh *user*. Besarnya nilai kepercayaan tersebut berdasarkan penghitungan dengan metode *Dempster Shafer*.

Dari kedua referensi diatas, Rancang Bangun Aplikasi Sistem Pakar Untuk Kerusakan Komputer yang penulis buat terdapat persamaan yaitunya metode yang digunakan yang menitikberatkan pada komponen penyusun sistem pakar. Sedangkan perbedaannya terletak pada alat bantu yang digunakan dalam merancang program dan basis pengetahuan yang diperoleh.

2.2 Landasan Teori

2.2.1 Sistem Pakar

a. Pakar

Seorang pakar atau ahli adalah seorang *individu* yang memiliki kemampuan pemahaman yang *superior* dari satu

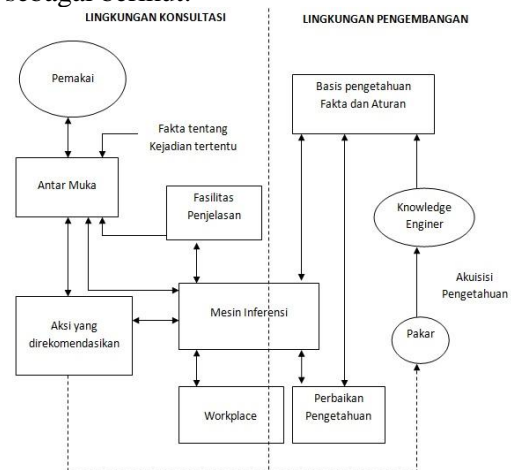
dominan tertentu. Misalnya ahli *farmasi* (dalam bidang farmasi), dokter, penasehat keuangan (dalam bidang ekonomi), dan lain-lain.

b. Sistem Pakar

Sistem pakar adalah suatu program komputer berbasis pengetahuan yang berusaha seorang pakar ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh seorang pakar. Seperti halnya seorang pakar, sistem pakar terfokus pada suatu dominan masalah yang spesifik.

c. Arsitektur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Komponen-komponen sistem pakar dalam dua bagian tersebut ada pada gambar 2.1 sebagai berikut:



Gambar 1 Arsitektur Sistem Pakar

Secara umum sistem pakar terdiri dari komponen penyusun sebagai berikut:

1. Knowledge Base (Basis Pengetahuan)

Basis pengetahuan merupakan hasil *akuisis* dan *representasi* pengetahuan dari seorang pakar. Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah.

2. Inference Engine (Mesin Inferensi)

Mekanisme *inferensi* yang utama pada sistem pakar dapat dibedakan menjadi inferensi dengan mekanisme pelacak mundur (*backward chaining*) dan pelacak maju (*forward chaining*). Penalaran dengan *Backward chaining* dimulai dari sekumpulan hipotesis menuju fakta-fakta yang mendukung hipotesis tersebut. *Forward chaining* merupakan kebalikan dari *Backward chaining*, yaitu

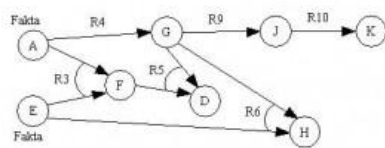
penalaran di mulai sekumpulan data menuju suatu kesimpulan atau goal

3. *User interface* (antar mungka pengguna)
User interface adalah penghubung antar program sistem pakar dengan pengguna

d. *Forward chaining* Dan *Backward chaining*

1. *Forward chaining*

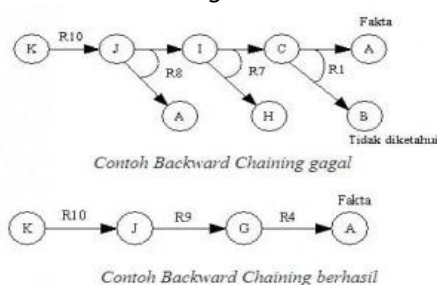
Forward chaining merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika *klausa premis* sesuai dengan situasi (bernilai *TRUE*), maka proses akan menyatakan *konklusi*. *Forward chaining* adalah *data-driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan *tree* yang lebar dan tidak dalam, maka gunakan *forward chaining*.



Gambar 2 Forward chaining

2. *Backward chaining*

Menggunakan pendekatan *goal-driven*, dimulai dari harapan apa yang akan terjadi (*hipotesis*) dan kemudian mencari bukti yang mendukung (atau berlawanan) dengan harapan kita. Sering hal ini memerlukan perumusan dan pengujian hipotesis sementara. Jika suatu aplikasi menghasilkan *tree* yang sempit dan cukup dalam, maka gunakan *backward chaining*.



Gambar 3 Backward chaining

Tabel 1 Karakteristik *Forward chaining* dan *Backward chaining*

<i>Forward chaining</i>	<i>Backward chaining</i>
Perencanaan, monitoring, kontrol	Diagnosis
Disajikan untuk masa depan	Disajikan untuk masa lalu
<i>Antecedent</i> ke <i>konsekuen</i>	<i>Konsekuensi</i> ke <i>antecedent</i>
Data memandu, penalaran dari bawah ke atas	Tujuan memandu, penalaran dari atas ke bawah
Bekerja ke depan untuk mendapatkan solusi apa yang mengikuti fakta	Bekerja ke belakang untuk mendapatkan fakta yang mendukung hipotesis
<i>Breadth first search</i> dimudahkan	<i>Depth first search</i> dimudahkan
<i>Antecedent</i> menentukan pencarian	<i>Konsekuensi</i> menentukan pencarian
Penjelasan tidak difasilitasi	Penjelasan difasilitasi

2.2.2 Konsep Dasar Database

Basis data terdiri dari 2 kata yaitu basis data. Basis data diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi faktor dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pembeli, pelanggan, barang, peristiwa dll) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis data didefinisikan dalam sejumlah sudut pandang, seperti:

- a. Himpunan kelompok data (arsip yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah).
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*Redudansi*) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file* atau tabel arsip yang saling berhubungan yang di simpan dalam media penyimpanan elektronik.

Database adalah kumpulan data / informasi yang teratur berdasarkan kriteria tertentu yang saling berhubungan (Yuswanto:2001). Dalam dunia komputer

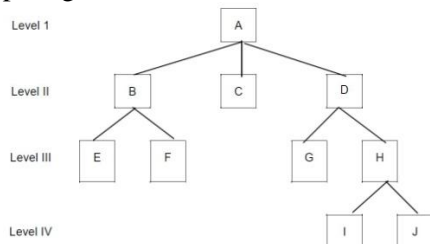
database bisa dikategorikan sangat spesial karena selalu menjadi hal utama dalam perancang sistem komputer.

a. Model Database

Model *database* adalah suatu konsep yang terintegrasi dalam menggambarkan hubungan (*relationships*) antar data dan batasan-batasan (*constraint*) data dalam suatu sistem *database*. Model data yang paling umum, berdasarkan pada bagaimana hubungan antar *record* dalam *database* (*Record Based Data Models*), terdapat tiga jenis, yaitu :

1. Model Database Hirarki (*Hierarchical Database Model*)

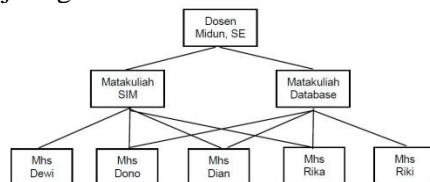
Model *database* hirarki disebut juga model pohon, karena hubungan antar simpul digambarkan seperti struktur pohon (*tree-structured*) yang dibalik dengan pola hubungan orang tua – anak (*parent – child*). Simpul yang paling atas disebut akar (*root*) dan paling bawah disebut daun.



Gambar 4 Contoh model database hirarki

2. Model Database Jaringan (*Network Database Model*)

Model *database* jaringan merupakan pengembangan dari model *database* hirarki, dimana kelemahan yang ada pada model *database* hirarki yaitu ketidakmampuannya dalam mengelola hubungan banyak ke banyak (*Many to Many*) telah dapat diatasi dengan model *database* jaringan ini.



Gambar 5 Contoh konkret model database jaringan

3. Model Database Relasi (*Relational Database Model*)

Model *database* relasi merupakan model *database* yang paling banyak digunakan saat ini, karena

paling sederhana dan mudah digunakan serta yang paling penting adalah kemampuannya dalam mengakomodasi berbagai kebutuhan pengelolaan *database*. Sebuah *database* dalam model ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan kolom (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*), tabel-tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*Key field*) sehingga dapat meminimalkan duplikasi data.

b. Tingkatan Data Dalam Database Relasi

Dalam suatu sistem *database* relasi, data yang tersimpan dalam DBMS mempunyai tingkatan-tingkatan, sebagai berikut :

a) Karakter (*Characters*)

Merupakan bagian terkecil dalam *database*, dapat berupa karakter numerik (angka 0 s.d 9), huruf (A - Z, a - z) ataupun karakter-karakter khusus, seperti *, &. %, # dan lain-lain.

b) Field atau Attribute

Merupakan bagian dari *record* yang menunjukkan suatu item data yang sejenis.

c) Record atau Tuple

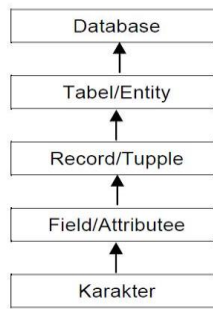
Tuple/*Record* adalah kumpulan data value dari attribute yang berkaitan sehingga dapat menjelaskan sebuah *entity* secara lengkap.

d) Table/Entity

Entity merupakan sesuatu yang dapat diidentifikasi dari suatu sistem *database*, bisa berupa objek, orang, tempat, kejadian atau konsep yang informasinya akan disimpan dalam *database*..

e) Database

Kumpulan dari tabel-tabel yang saling berelasi, disusun secara logis, sehingga menghasilkan informasi yang bernilai guna dalam proses pengambilan keputusan.



Gambar 6 Tingkatan dalam database

Ada beberapa sifat yang melekat pada suatu tabel :

- Tidak boleh ada *record* yang sama (kembar)
- Urutan *record* tidak terlalu penting, karena data dalam *record* dapat diurut sesuai dengan kebutuhan.
- Setiap *field* harus mempunyai nama yang unik (tidak boleh ada yang sama).
- Setiap *field* mesti mempunyai tipe data dan karakteristik tertentu.

Tabel 2 Contoh sebuah tabel

NOBP	NamaMahasiswa	Alamat	
03156001	Helga Elzar Adha	Curup	> baris record ke-1
03156002	Rifky Zarel Putra	Bengkulu	> baris record ke-2
03156003	Fikri Putra Zarel	Padang	> baris record ke-3
03156004	Reinhard Steven	Padang Panjang	> baris record ke-4
03156005	Edi Brocoli	Lubuk Alung	> baris record ke-5
Field NoBP	Kolom Field Nama Mahasiswa	Kolom field Alamat	

Jenis hubungan antar tabel dalam model *database* relasi, juga didefinisikan dengan hubungan :

- Satu ke satu (*One to One*)
- Satu ke Banyak (*One to Many*)
- Banyak ke satu (*Many to One*)
- Banyak ke Banyak (*Many to Many*)

c. Model Data *Entity Relationship*.

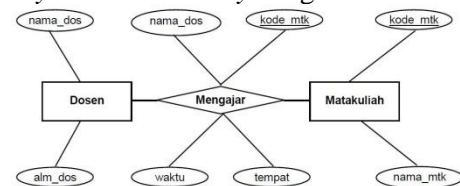
Seperti telah dijelaskan di atas, bahwa hubungan antar data dan batasan-batasannya dalam suatu sistem *database*, dapat diolah secara *hirarki*, jaringan dan relasional. Ketiga tipe model data ini mengacu kepada hubungan antar *record* (*Record Based Data Models*) dalam masing-masing *entity*/tabel. Tapi disisi lain, hubungan dan batasan data ini, dapat juga berupa *Object Based Data Model* (Model Data Berbasis *Object*).

d. Diagram *Entity Relationship* (Diagram *E-R*)

Diagram *E-R* digunakan untuk menggambarkan secara sistematis hubungan

antar *entity-entity* yang ada dalam suatu sistem *database* menggunakan simbol-simbol sehingga lebih mudah dipahami. Simbol-simbol yang boleh digunakan adalah :

- Persegi Panjang, berfungsi untuk menyatakan suatu *entity*.
- Elips*, berfungsi untuk menyatakan attribute, jika diberi garis bawah menandakan bahwa attribute tersebut merupakan attribute/*field* kunci.
- Belah Ketupat, menyatakan jenis relasi.
- Garis, penghubungan antara relasi dengan *entity* dan antara *entity* dengan attribute.



Gambar 6 Contoh sebuah diagram E-R

2.2.3 Gambaran Umum Microsoft Visual Basic 6.0

Visual Basic pada dasarnya adalah sebuah bahasa pemrograman komputer, juga sering disebut sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis windows. *Basic* merupakan pengembangan terakhir dari bahasa *basic*. *Visual Basic 6.0* memiliki kelebihan-kelebihan yang tidak dimiliki oleh versi sebelumnya. Kelebihannya antara lain kompiler (proses *compile*) dapat dilakukan dengan cepat, mendukung kontrol data objek yang baru, mendukung berbagai macam *database*, pembuatan laporan yang lebih mudah dan mendukung pengaksesan terhadap internet.

Komponen *Visual Basic*

Adapun komponen-komponen yang terdapat pada *Visual Basic* antara lain:

1. Control Menu

Control menu adalah menu yang digunakan terutama untuk memanipulasi jendela *Visual Basic*.. Berikutnya akan muncul *control menu*, dimana kita bisa memilih salah satu dari perintah berikut ini:

a. Restore

Mengubah ukuran jendela ke ukuran sebelumnya.

b. Move

Untuk memindahkan letak jendela.

c. Size

Untuk mengubah ukuran jendela.

d. *Minimize*

Untuk meminimalkan ukuran jendela

e. *Maximize*

Untuk memaksimalkan ukuran jendela

f. *Close*

Untuk menutup jendela

2. Menu

Menu merupakan daftar perintah-perintah yang dikelompokkan dalam kriteria tertentu yang berfungsi untuk melaksanakan sebuah perintah. *Visual Basic 6.0* terdapat tiga belas menu utama yaitu *File, Edit, View, Project, Format, Debug, Run, Query, Diagram, Tools, Add-In, Windows* dan *Help*.

3. Toolbar

Toolbar merupakan kumpulan tombol yang dapat melakukan sebuah perintah dengan cepat. Fungsi *toolbar* hampir sama dengan menu hanya *toolbar* berbentuk tombol-tombol yang susunannya tidak secara bertingkat. Contoh *toolbar* tersebut antara lain *Add Form, Menu Editor, Save, Cut, Copy* dan lain sebagainya.

4. Form

Form adalah tempat untuk meletakkan objek-objek yang digunakan untuk melaksanakan perintah yang diberikan. Pada bagian kanan atas form terdapat tombol *Minimize, Maximize/Restore* dan *Close* yang digunakan untuk mengatur tampilan jendela form.

5. Windows Code

Window Code adalah jendela tempat menuliskan kode program. Semua kode perintah ditulis pada jendela ini. Pada bagian *Window Code* terdapat fasilitas kode editing yang cukup lengkap.

6. Toolbox

Toolbox merupakan kumpulan *object* yang digunakan untuk merancang sebuah output program. Karena masing-masing *object* mempunyai sifat yang khas, dan dengan sifat yang khas tersebut kita tinggal memberikan kontrolnya, sehingga menjadi suatu aplikasi program yang kita inginkan, maka *object* ini disebut dengan **Control**. kita bisa menambahkan *control* yang lain ke dalam *toolbox* jika diperlukan.

Penambahan *control* ini dimungkinkan jika ada beberapa fungsi atau *object* yang diperlukan ternyata tidak ada dalam *toolbox* tersebut.

7. Project Explorer

Project Explorer digunakan untuk melihat bagian-bagian proyek pembuatan aplikasi. Bagian-bagian tersebut dapat berupa project, form, data environment dan data report. *Project Explorer* ini berbentuk menu *tree* sehingga mempermudah dalam pengaksesannya. Pada jendela *explorer* terdapat tiga tombol kontrol tampilan antara lain *Window Code* untuk menampilkan kode, *Window Project* untuk menampilkan dalam bentuk *visual* dan *Toggle Folder* untuk pengelompokan jenis objek.

8. Windows Properties

Window Properties menampilkan semua properti dari obyek yang digunakan. Kita dapat mengubah setiap properti dari objek yang ada melalui jendela ini.

9. Windows Form Layout

Windows Form Layout digunakan untuk mengatur letak form pada layar monitor.

2.2.4 Macam-Macam Kerusakan Komputer

- a. Komputer tidak mau hidup
- b. Komputer mau hidup tetapi tidak mau booting
- c. Komputer mau *booting* tetapi selalu "*Safe Mode*"
- d. Komputer sering hang
- e. *Keyboard* tidak dikenali oleh komputer
- f. *Mouse* tidak dikenali oleh komputer (sama dengan kasus *keyboard*)
- g. *Pointer mouse* selalu meloncat-loncat
- h. Komputer sering *crash*
- i. Bila produsen *motherboard*(mb) tidak diketahui
- j. Lupa *password bios*
- k. Jam dan setting tanggal *bios* selalu berubah-ubah
- l. Menambah perangkat *hardware* baru, tapi tidak terdeteksi oleh *bios*

- m. Melacak kerusakan *card* pada mb
- n. Pasang *processor* baru tp tidak terdeteksi
- o. *Crash* setelah memasang *RAM* baru
- p. Menambah *RAM* tapi tidak terdeteksi
- q. Setelah menambah *RAM* proses komputer manjadi semakin lambat
- r. Virtual *RAM* bermasalah
- s. Monitor tidak mau nyala
- t. Monitor menjadi gelap saat loading *windows*
- u. Tampilan tiba-tiba rusak dan komputer manjadi hang
- v. Ukuran tampilan monitor tidak sesuai keinginan
- w. Monitor seperti berkedip saat digunakan
- x. *Sound Card* baru tidak terdeteksi

3. Metodologi Penelitian

Dalam perancangan ini penulis mencoba untuk melakukan *eksperimen* dalam pengembangan sistem dari penelusuran tentang kerusakan komputer dengan berbagai sumber yang pakar dibidangnya menjadi sistem pakar untuk kerusakan komputer dengan metode *backward chaining* dan alat bantu *Visual Basic 6.0* dimana nantinya dapat memberikan informasi bagi pengguna komputer yang mengalami permasalahan komputer.

Rancangan penelitian ini dilakukan sekitar bulan September 2011 sampai bulan Desember 2011 di Institut Teknologi Padang

Dalam penelitian ini *hardware* dan *software* yang digunakan adalah:

1. Perangkat Keras (*Hardware*)
 - a. Komputer PC, Intel Pentium 2 Cores 3 GHz
 - b. *Hardisk* 250 Gb
 - c. *Motherboard* Asus P5KPL-AM SE
 - d. *DVD RW* Asus
 - e. *LCD* Samsung 18 *inchi*
 - f. *Keyboard* dan *Mouse* Logitech
 - g. *Printer* Canon Pixma IP 2770
2. Perangkat Lunak (*Software*)
 - a. Sistem Operasi Windows XP 2
 - b. *Software Visual Basic 6.0* sebagai bentuk *visual* untuk pengguna

- c. *Microsoft Office Access 2007* sebagai desain *database*
- d. Serta *software* pendukung lainnya.

3.1 Rancangan Basis Pengetahuan

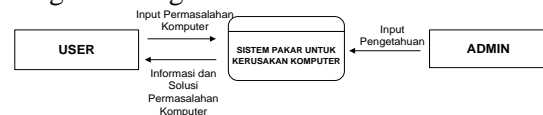
Basis pengetahuan yang bersifat dinamis, sehingga pakar dapat menambah atau mengubah basis pengetahuan tersebut sesuai data yang baru. Sedangkan untuk aturan dapat digambarkan pada tabel 3.

Tabel 3 Rule

Rule	If	Then
1	T1,T2,T3,T4,T5,T15,T19,T20,T21	K1
2	T1,T2,T11,T18,T21	K2
3	T2,T3,T4,T5,T6,T7,T8,T9,T12,T18	K3
4	T3,T4,T5,T11,T14,T18	K4
5	T2,T3,T7,T8,T10,T11,T12,T13,T15,T20	K5
6	T5,T9,T14,T18,T21	K6
7	T16	K7
8	T1,T17	K8
9	T5,T18	K9

3.2 Rancangan Diagram Context

Secara garis besar perancangan sistem ini dapat digambarkan dalam bentuk konteks diagram sebagai berikut:



. Gambar 7 Konteks Diagram Sistem

Adapun penjelasan dari konteks diagram dari sistem ini adalah sebagai berikut:

1. Administrator terlebih dahulu menginputkan pengetahuan kedalam sistem pakar untuk kerusakan komputer berupa tanya jawab sederhana yang mudah di pahami pengguna. Kemudian admin juga menginputkan solusi dari permasalahan komputer itu sendiri.
2. *User* atau pengguna akan menginput permasalahan komputer yang dihadapi, seperti persoalan umum yang di hadapi, atau permasalahan yang lebih spesifik mengenai perangkat komputer.
3. Sistem pakar akan mengolah data yang telah disimpan melalui tanya jawab *interaktif* dan kemudian sistem pakar ini akan memberikan penjelasan dan solusi. Sehingga pengguna mendapatkan informasi dari sitem pakar ini mengenai permasalahan komputer yang dihadapi.

3.3 Rancangan Diagram Context

Dalam membuat suatu program di perlukan adanya *file* yang saling berinteraksi satu sama lainnya. *File-file* program yang dibutuhkan merupakan suatu kesatuan namun dibuat berpisah di tempat yang berbeda dan tidak bisa dijalankan sendiri-sendiri, karna keterkaitannya dengan main program.

Tabel 4 Rancangan Tabel Kerusakan

No	Nama Field	Tipe Data	Panjang	Kunci
1	ID_kerusakan	Teks	8	
2	jenis_kerusakan	Teks	30	

Tabel 5 Rancangan Tabel Pengetahuan

No	Nama Field	Tipe Data	Panjang	Kunci
1	ID	Teks	8	
2	pertanyaan	Teks	255	
3	faktaYa	Teks	20	
4	faktaTIDAK	Teks	20	
5	YA	Teks	8	
6	TIDAK	Teks	8	
7	ID_kerusakan	Teks	8	

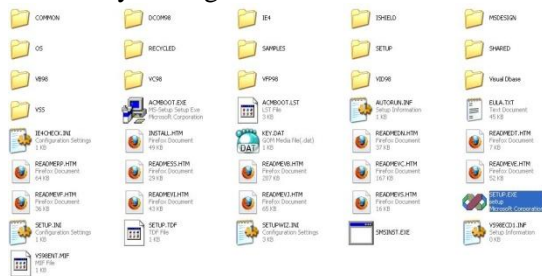
Tabel 6 Rancangan Tabel Solusi

No	Nama Field	Tipe Data	Panjang	Kunci
1	ID_solusi	Teks	8	
2	solusi	Memo	50	

4. Pembahasan

4.1 Instalasi Program Visual Basic 6.0

Rancang bangun aplikasi sistem pakar untuk kerusakan komputer memerlukan beberapa proses. Proses tersebut dituangkan dalam beberapa langkah - langkah pengerjaan. Adapun proses tersebut diantaranya sebagai berikut :



Gambar 8 Instalasi Visual Basic 6.0

4.2 Tampilan User Interface

4.2.1 Tampilan Menu Utama

Halaman menu utama merupakan halaman yang akan langsung tampil sewaktu program dijalankan. Dalam halaman ini akan ada tampil beberapa pilihan sub menu yang bisa dipilih, diantaranya menu sistem untuk *exit*, diagnosis, basis pengetahuan, tentang, dan lain-lain.



Gambar 9 Tampilan Menu Utama

Pada bagian menu, terdapat menu basis pengetahuan untuk edit pengetahuan. Terdapat perbedaan fungsi untuk *admin* dengan pengguna, untuk admin akan aktif menu basis pengetahuan pada halaman menu utama. Basis pengetahuan merupakan halaman khusus untuk *admin*, dimana pada halaman tersebut *admin* menginput pengetahuan. Berikut tampilan menu utama untuk *admin*.



Gambar 10 Login Admin

4.2.2 Input Kerusakan

Sebelum program aplikasi sistem pakar untuk kerusakan komputer ini digunakan, admin terlebih dahulu menginputkan kerusakan, yang nantinya akan menjadi klasifikasi kerusakan komputer dalam program sistem pakar ini. Admin terlebih dahulu *login* kedalam sistem, kemudian memilih menu *edit* pengetahuan.

ID kerusakan	jenis kerusakan
K1	Masalah Umum
K2	Power Supply
K3	Memory RAM
K4	Processor
K5	Hardisk
K6	VGA card
K7	Sound Card
K8	LAN Card

Gambar 11 Input Kerusakan

4.2.3

Solusi Input

Input solusi dilakukan setelah input kerusakan, data inputan berupa ID solusi dan solusi. Dalam penginputan data solusi *variable* kode untuk ID solusi dibuat satu *variable* juga, misalkan S1, S2, S3 dan seterusnya. Seperti yang terlihat pada gambar di bawah ini.

Gambar 12 Input Solusi

4.2.4 Input pengetahuan

Input pengetahuan merupakan kunci dari sistem pakar untuk kerusakan komputer ini. Pada inputan pengetahuan ini, akan direlasikan antara tabel kerusakan dan tabel solusi. Dalam input pengetahuan ini juga akan diinputkan pertanyaan-pertanyaan yang nantinya akan menjadi tanya jawab diagnosis antara pengguna dengan sistem pakar ini.

Gambar 13 Input Pengetahuan

4.2.5 Konsep Relasi Tabel

Dalam pengolahan data, terdapat hubungan relasi antar tabel. Dalam perancangan program, hubungan antara tabel kerusakan dengan tabel pengetahuan terdapat hubungan *one-to-many*. Seperti yang terlihat pada gambar di bawah ini:



Gambar 14 Relasi Tabel

4.2.6 Pilihan Kerusakan

Setelah pengguna melakukan pilihan untuk diagnosis kerusakan maka akan muncul pilihan kerusakan. Di dalam pilihan kerusakan terdapat beberapa bagian pilihan diantaranya kerusakan umum, *power supply*, *hardisk*, *memory RAM*, *prosesor*, *VGA card*, *LAN card*, *sound card*, dan *motherboard*.

Gambar 15 Pilihan Kerusakan

4.2.7 Diagnosis

Masalah umum merupakan salah satu klasifikasi untuk kerusakan komputer yang sering dihadapi pengguna, dan didalamnya terdapat pertanyaan-pertanyaan.

Gambar 16 Diagnosis Masalah Umum

Dari gambar diagnosis, jika pengguna memilih ya yang artinya fakta mendukung pertanyaan yang berkaitan dengan masalah umum, maka akan muncul hasil diagnosis, seperti gambar dibawah ini.

Gambar 17 Hasil Diagnosis Masalah Umum

Setelah hasil diagnosis, pengguna bisa memilih selesai atau alasan. Jika pengguna memilih selesai, artinya pengguna kembali ke halaman pilihan kerusakan. Sedangkan jika pengguna memilih alasan, maka akan muncul halaman penanganan.



Gambar 18 Penanganan Masalah Umum

Untuk lebih jelas dan detail program sistem pakar untuk kerusakan komputer ini menyediakan *form* penanganan. *Form* penanganan ini akan tampil ketika pengguna mengklik alasan pada jendela hasil diagnosis, seperti yang terlihat pada *listing* program berikut ini.

```
frmPenanganan.Label2.Text = ""
rsTemp.MoveFirst
While Not rsTemp.EOF
    frmPenanganan.Label2.Text =
    frmPenanganan.Label2.Text + _
    rsTemp!fakta + vbCrLf
    rsTemp.MoveNext
Wend
frmPenanganan.Label2.Text =
frmPenanganan.Label2.Text + vbCrLf +
vbCrLf + _
"Maka Hasil Diagnosis : " + vbCrLf +
rsSolusi!solusi
frmPenanganan.Show vbModal
```

Penjelasan *listing* program:

Form penanganan ini merupakan kumpulan dari hasil diagnosis sebelumnya berupa fakta-fakta yang disimpan dalam *temporary*, seperti yang diatur dalam *script* :

```
rsTemp.MoveFirst
rsTemp!fakta + vbCrLf
rsTemp.MoveNext
```

Kemudian *temporary* dijadikan sebuah alasan dengan data dan referensi yang akan memberikan penjelasan berupa penanganan kepada pengguna mengenai kerusakan komputer yang dialami.

5. Kesimpulan

Sistem pakar untuk kerusakan komputer ini pelacakan mesin inferensinya adalah pelacakan mundur (*backward chaining*) yang dimulai dari sekumpulan hipotesis gejala kerusakan menuju fakta-fakta yang mendukung hipotesis tersebut. Hasil perancangan akan memberikan informasi kepada pemakai komputer bagaimana mengenali dan menangani kerusakan komputer.

Kepustakaan

Arhamni, Muhammad, "Konsep Dasar Sistem Pakar", Yogyakarta: Andi 2005

Friyadie, "Mudah Belajar Database MySQL Dengan Microsoft Visual Basic 6.0", Yogyakarta: Andi 2010

Kusomo, Ario suryo, "Microsoft Visual Basic 6.0", Jakarta: PT. Elex Media Komputindo 2000

Kusrini, Sistem Pakar Teori dan Aplikasi, Yogyakarta: Andi 2006

Kusumadewi, Sri. Artificial Intelligence (Teknik Dan Aplikasinya) Yogyakarta: Graha Ilmu. 2003

Madcoms, "Pemograman Tingkat Lanjut Dengan Visual Basic 6.0 dan Crystal Report", Yogyakarta: Andi 2010

Sasmito, Ginanjar Wiro, "Aplikasi Sistem Pakar Untuk Simulasi Diagnosa Hama dan Tanaman Bawang Merah dan Cabai Menggunakan Forward Chaining dan Pendekatan Berbasis Aturan", Semarang: 2010

Sulistiyohati, Aprilia dan Hidayat, Taufik. "Aplikasi Sistem Pakat Diagnosa Penyakit Ginjal Dengan Metode Dempster-Shafer", Yogyakarta: 2008

Turban, Efraim. Decision Support and expert systems Management supports (fourth edition). Prentice-Hall International, Inc. 1995