

BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Sistem yang Sedang Berjalan

Proses analisa sistem merupakan langkah kedua pada pengembangan sistem. Analisa sistem dilakukan untuk memahami informasi-informasi yang didapat dan dikeluarkan oleh sistem itu sendiri. Apabila ingin mencari solusi tentang masalah yang timbul didalam sebuah perangkat keras *processor* harus mencari seseorang yang ahli didalam bidang perbaikan *processor* tersebut. Hal ini secara otomatis membutuhkan waktu untuk menunggu seseorang tersebut untuk menyelesaikan masalah yang timbul didalam kerusakan *processor* tersebut.

Dalam mencari kerusakan didalam *processor*, analisa sistem merupakan hal yang harus dilakukan sebelum proses perancangan sistem. Pada proses analisa sistem terdapat 3 (tiga) langkah analisa yang harus dilakukan yaitu analisa *input*, analisa proses dan analisa *output*. Adapun analisa sistem yang sedang berjalan sebagai berikut :

III.1.1. *Input*

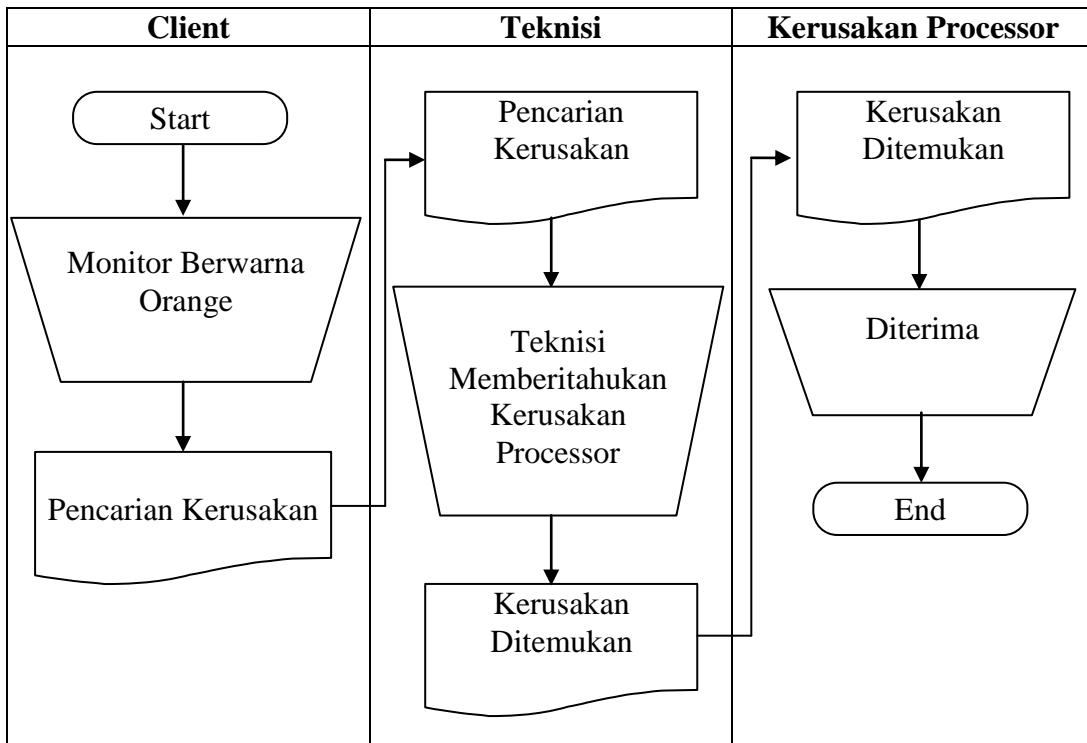
Sistem yang berjalan berkaitan dengan sistem pakar deteksi kerusakan *processor* yang ada. Yang kemudian kerusakan *processor* tersebut dikumpulkan dalam satu wadah yang disebut dengan dokument, kemudian hasil dari pengumpulan kerusakan tersebut diberitahukan kepada seseorang ahli didalam memperbaiki *processor* tersebut.

FORM KERUSAKAN PROCESSOR			
Tanggal		Kepada	
No	Nama	Kerusakan	Solusi

Gambar III.1. Form Kerusakan Processor

III.1.2. Analisa Proses

Setelah data kerusakan *processor*, maka akan dilakukan analisa proses pengolahan data tersebut. Analisa proses adalah suatu bagian dimana suatu *input* data akan dikelola agar menjadi *output* yang diinginkan. Penginputan data dilakukan oleh seorang yang berkepentingan, kemudian data tersebut diserahkan kebagian tertentu yang berhak menerimanya. Untuk lebih jelasnya dapat dilihat gambar III.2.



Gambar III.2. FOD Kerusakan Processor

III.1.3 Output

Data *output* adalah data hasil pengolahan data mendeteksi kerusakan *processor* telah dilakukan dalam bentuk yang akan ditujukan kepada pihak yang menerima kerusakan tersebut.

Tabel III.1. Output Daftar Kerusakan Komputer

Dari : Rizky		Kepada : Manager IT					
Tanggal : 15-06-2014							
Daftar Kerusakan Komputer							
No	Tanggal	Kerusakan	Solusi	Perbaikan			
1.	10-01-2014	<ul style="list-style-type: none"> - Komputer mati total - Processor terdeteksi tidak - Komputer Total mati 	<ul style="list-style-type: none"> - Power Supply Rusak - Jarum Pin Patah - Motherboard Rusak - Post tidak berjalan 	<ul style="list-style-type: none"> - Ganti Baru - Ganti Processor - Ganti Motherboard 			

		<ul style="list-style-type: none"> - Laptop tidak bisa tampilan windows - Processor tidak bisa baca data - Hard disk terdeteksi tetapi root direktori tidak tampil - Processor proses baca lambat 	<ul style="list-style-type: none"> dengan baik - MBR hard disk crash - Kerusakan pada Soket - Banyak yang looading 	<ul style="list-style-type: none"> - Ganti memori Fisik - Ganti Soket - Atur ulang partisi - Format Ulang
2.	11-01-2014	<ul style="list-style-type: none"> - Tampilan display hitam. - Lampu LCD warna orange. - Tidak bisa instal OS - Keyboard dan mouse tidak terbaca - Terdengar suara beep 3 x di laptop 	<ul style="list-style-type: none"> - VGA rusak - Komputer dalam kondisi crash - Sector hard disk ruask - Driver laptop keypad tidak ditemui - RAM rusak 	<ul style="list-style-type: none"> - Ganti VGA - Servis sistem hardware - Ganti hard disk - Uninstal driver ganti sesuai dengan driver laptop - Ganti RAM

III.2. Evaluasi Sistem Yang Berjalan

Sistem pakar mendeteksi kerusakan *processor* yang sedang berjalan saat ini belum memadai karena masih secara manual menggunakan aplikasi excel. Jika ada masalah dengan kerusakan komputer masih sulit mendeteksi kerusakan tersebut. Kelemahan dari sistem manual ini, apabila data itu terkena virus maka data tersebut tidak bias dibuka.

Maka solusi yang penulis buat untuk mengatasi masalah tersebut adalah membuat suatu sistem pakar untuk mendeteksi kerusakan *processor*, yang hanya tinggal menjalankan sekali klik sudah dapat mengatasi permasalahan tersebut.

III.3. Basis Rules

Basis rules digunakan memperjelaskan data gejala dan kerusakan pada *processor*, untuk lebih jelasnya dapat dilihat pada tabel dibawah ini :

1. Tabel Gejala

Tabel basis rules gejala ini terdiri dari dua *field*, dapat dilihat pada Tabel III.2.

Tabel III.2. Gejala Kerusakan Processor

Kode Gejala	Nama Gejala	Nilai
GP1	Processor tidak terdeteksi	0.1
GP2	Lambat proses baca data	0.6
GP3	Komputer menyala tetapi tidak terjadi proses apapun	0.1
GP4	Komputer hank	0.2
GP5	Komputer sering restart sendiri	0.1
GP6	Komputer hidup tapi dilayar monitor tidak muncup apa-apa	0.3
GP7	Terdengar suara BEEP sebanyak 4 kali	0.3

2. Tabel Kerusakan

Tabel basis rules kersakan ini terdiri dari empat *field*, dapat dilihat pada Tabel III.3.

Tabel III.3. Kerusakan Processor

Kode kerusakan	Nama kerusakan	Definisi	Solusi
P1	Kaki pin processor patah	Komputer hidup tetapi tidak muncup reaksi pada computer	Ganti processor baru
P2	Sering terjadi restart sendiri	Kipas FUN pada processor mati atau tidak hidup	Ganti kipas FUN dengan yang baru
P3	Tampilan dilayar monitor gelap	Kerusakan pada processor	Ganti processor baru
P4	Terdengar suara BEEP 3 Kali	Processor terbakar	Ganti processor dengan yang baru
P5	Komputer sering Hank	Kipas FUN mati	Ganti kipas FUN

Adapun contoh kaidah sistem pakar kerusakan processor pada komputer pada lambang adalah sebagai berikut :

3. Data rule

Data rule yang terdapat pada kerusakan processor adalah sebagai berikut :

a. Rule 1

JIKA processor tidak terdeteksi **DAN** Apakah lambat proses baca data **DAN** Apakah komputer menyala tetapi tidak terjadi proses.

MAKA

Kaki PIN processor patah

b. Rule 2

JIKA Apakah lambat proses baca data **DAN** Apakah komputer hank.

Maka

Sering terjadi restart sendiri

c. Rule 3

JIKA processor tidak terdeteksi **DAN** Apakah lambat proses baca data **DAN** Apakah komputer menyala tetapi tidak terjadi proses apapun **DAN** komputer hank **DAN** Komputer sering restart sendiri **DAN** Komputer hidup tapi layar monitor tidak muncup apa-apa **DAN** terdengar beep sebanyak 4 kali.

Sering terjadi restart sendiri

Tampilan dilayar monitor gelap

Terdengar suara Beep 3 kali

Rule 4

JIKA processor tidak terdeteksi **DAN** Apakah komputer menyala tetapi tidak terjadi proses apapun **DAN** komputer hank **DAN** Komputer sering restart sendiri

Maka

Kaki pin processor patah
 Sering terjadi restart sendiri
 Tampilan dilayar monitor gelap
 Terdengar suara beep 3 kali

d. Rule 5

JIKA processor tidak terdeteksi **DAN** Apakah komputer menyala tetapi Tidak Terjadi Proses Apapun **DAN** komputer hank

Maka

Sering terjadi restart sendiri

Pertanyaan-pertanyaan ini akan diproses dengan bentuk sebagai berikut :

Tabel III.4. Daftar Keputusan Kerusakan Processor

No	Id	Gejala Id
1.	Rule 1	If GP01 And GP02 And GP03 Then P1
2.	Rule 2	If GP03 And GP04 Then P2
3.	Rule 3	If GP02 And GP03 And GP04 And GP05 And GP06 Then P2,P3, P4,
4.	Rule 4	If GP01 And GP02 GP03 And GP04 And GP05 GP06 And

		GP07 Then P1, P2,P3,P4
5.	Rule 5	If GP01 And GP03 And GP04 Then P2

III.3. Metode Fuzzy Logic

Adapun metode yang dipakai dalam pemecahan masalah dari sistem pakar kerusakan *processor* adalah metode *fuzzy* tsukamoto. Inferensi adalah proses penggabungan banyak aturan berdasarkan data yang tersedia. Komponen yang melakukan inferensi dalam sistem pakar disebut mesin inferensi. Dua pendekatan untuk menarik kesimpulan pada IF-THEN rule.

Contoh kasus kerusakan processor dalam mencari nilai suatu gejala menggunakan metode fuzzy logic tsukamoto adalah sebagai berikut :

$$R(B(G_1), P_4(G_1)) = \text{Max} (0, 1 - \frac{1|0.6 - 0.4|}{0.4})$$

Keterangan dari rumus tersebut adalah :

$R \rightarrow$ Kode

$B(G_1) \rightarrow$ Nilai Gejala G1 yang diinputkan client B

$P_4(G_1) \rightarrow$ Nilai gejala G1 di dalam tabel untuk kerusakan P_4

$$\text{Max} (0,1 - \frac{1 | 0.6 - 0.4 |}{0.4})$$

0.6 adalah angka yang inputkan user untuk gejala lambat proses baca data tersebut.

0.4 adalah angka didalam tabel untuk kerusakan kaki pin processor patah tersebut.

Cara menghitung :

$$1 | 0.6 - 0.4 |$$

Hitung dulu bagian ini, $0.6 - 0.4 = 0.2$

Tanda kurung garis tegak gunanya untuk mengambil nilai mutlak dan menghilangkan tanda minus.

Jadi seandainya hasil didalam kurung bernilai -0.2, maka yang diambil hanya angka 0.2 (angka mutlak), Tanda minusnya hilang.

Maka rumusnya menjadi :

$$\begin{aligned} \text{Max} (0.1 - 0.2 / 0.4) &= \text{Max} (0, 1 - 0.5) \\ &= \text{Max} (0, 0.5) \end{aligned}$$

Yang diambil yang maximum antara 0 dan 0.5, yaitu 0.5 maka hasilnya 0.5.

III.4. Desain Sistem

Setelah tahapan analisis sistem, maka selanjutnya dibuat suatu rancangan sistem. Perancangan sistem adalah tahapan yang berguna untuk memperbaiki efisiensi kerja dari sistem yang telah ada.

III.4.1. Desain Sistem Secara Global

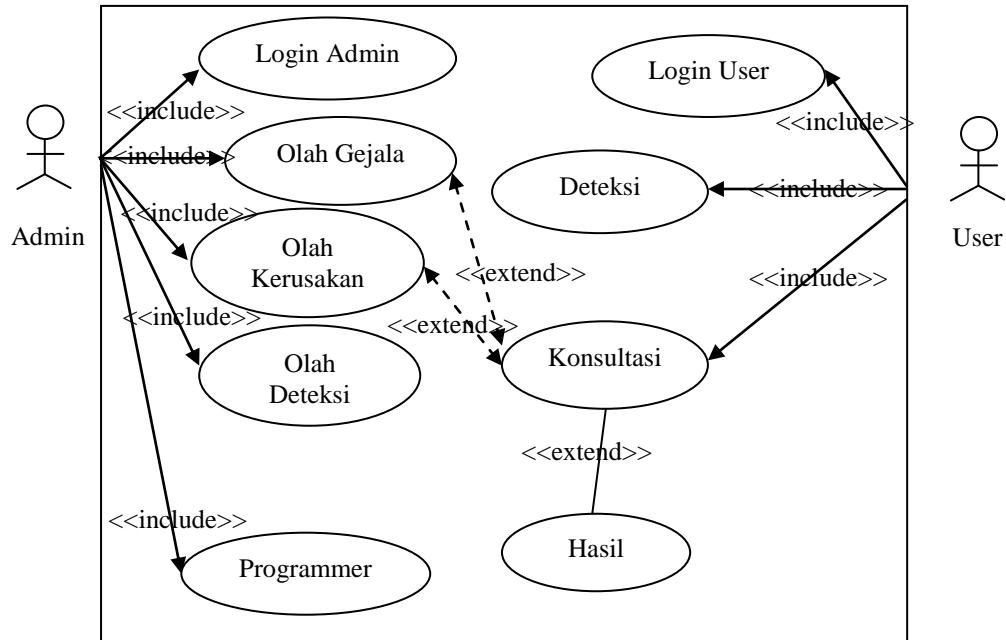
Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

1. Perancangan *Use Case Diagram*
2. Perancangan *Sequence Diagram*
3. Perancangan *Activity Diagram*
4. Perancangan *Class Diagram*

III.4.1.1 *Use Case Diagram*

Use case menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem dan menggambarkan fungsionalitas yang

diharapkan dari sebuah sistem pakar mendekripsi kerusakan *processor*. Diagram *Use Case* tersebut dapat dilihat pada gambar III.3.



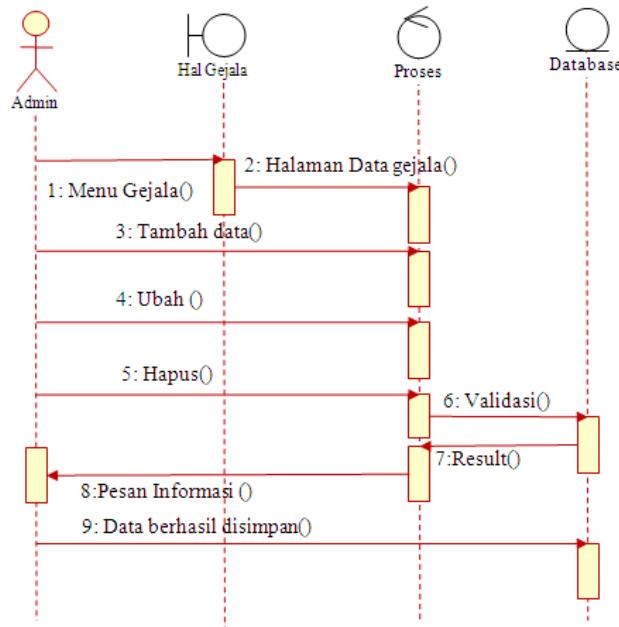
Gambar III.3. Use Case Diagram Sistem Pakar

III.4.1.2. Sequence Diagram

Sequence diagram menunjukkan bagaimana detil operasi dilakukan. *Sequence* diagram menjelaskan interaksi obyek yang disusun dalam suatu urutan waktu. Urutan waktu yang dimaksud adalah urutan kejadian yang dilakukan oleh seorang *actor* dalam menjalankan sistem.

1. Sequence Daftar Gejala

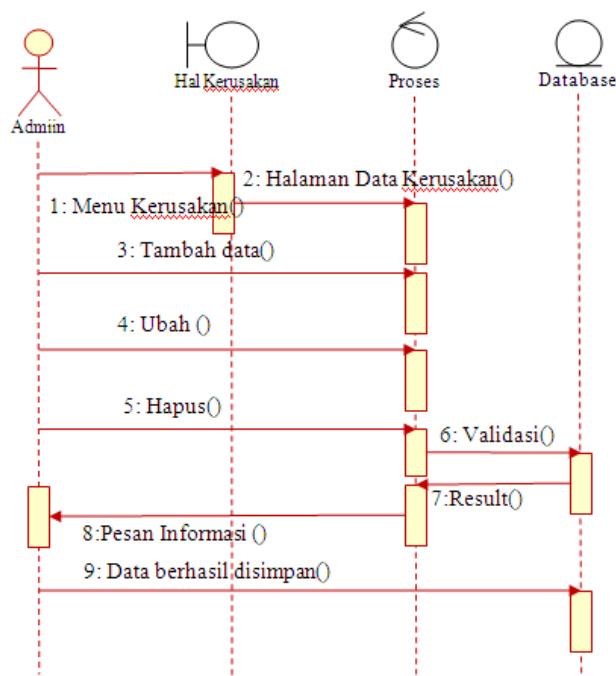
Sequence daftar gejala menunjukkan bagaimana operasi dilakukan antara user dengan objek daftar gejala, *sequence* ini dapat dilihat pada gambar III.4.



Gambar III.4. Sequence Diagram Daftar Gejala

2. Sequence Daftar Kerusakan

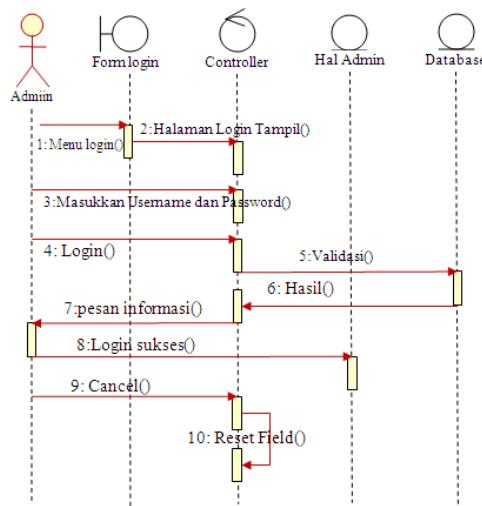
Sequence daftar kerusakan menunjukkan bagaimana operasi dilakukan antara *user* dengan objek kerusakan, *sequence* ini dapat dilihat pada gambar III.5.



Gambar III.5. Sequence Diagram Daftar Solusi

3. Sequence Login Admin

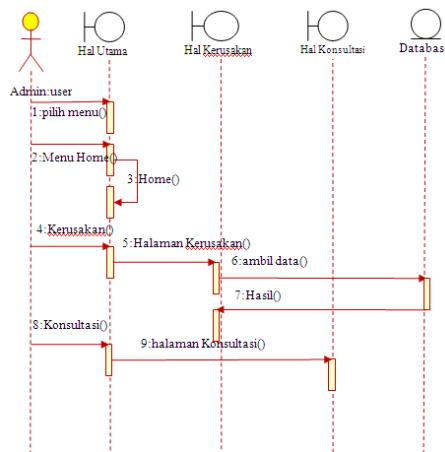
Sequence login admin dilakukan untuk mengaktifkan form admin, sequence ini dapat dilihat pada gambar III.6.



Gambar III.6. Sequence Diagram Login Admin

4. Sequence Halaman Utama

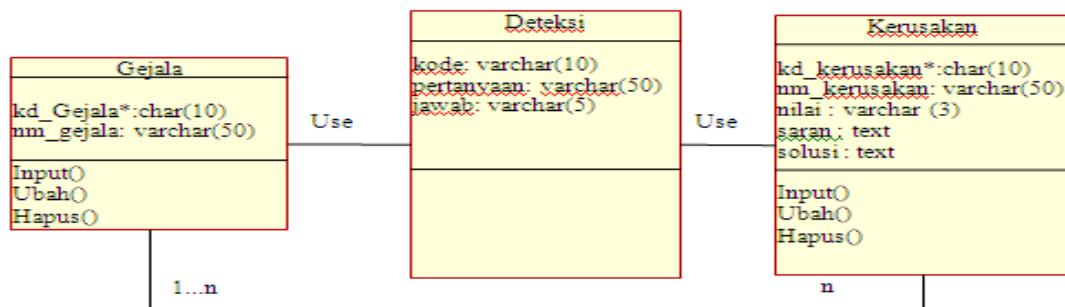
Sequence halaman utama menunjukkan untuk kembali ke awal menu utama, sequence ini dapat dilihat pada gambar III.7.



Gambar III.7. Sequence Diagram Halaman Utama

III.4.1.2. Class Diagram

Class diagram sangat membantu penulis dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model disain dari suatu sistem. Adapun class diagram yang diusulkan dapat dilihat pada gambar berikut ini :



Gambar III.8. Diagram Class Sistem Pakar

III.5. Desain Sistem Secara Detail

Perancangan bentuk fisik atau bagan arsitektur sistem yang diusulkan. Dalam merancang suatu sistem perlu diketahui hal yang akan menunjang sistem, agar dapat mempermudah pengolahan data nantinya. Pengolahan data ini diharapakan dapat mempermudah dalam hal penyajian, pelayanan dan pembuatan berbagai laporan data yang dibutuhkan. Berdasarkan hal tersebut diatas, penulis akan menguraikan lebih detail rancangan sistem yang diusulkan.

III.5.1 Desain Output

Desain sistem pakar ini berisikan pemilihan menu dan hasil pencarian yang telah dilakukan. Adapun bentuk rancangan output dari sistem pakar deteksi kerusakan processor ini adalah sebagai berikut :

1. Menu Utama

Adapun bentuk form menu utama dapat dilihat pada gambar III.9.

Sistem Pakar Kerusakan Processor					
Gejala	Kerusakan	Diteksi	Hasil	Programmer	Keluar

Gambar III.9. Menu Utama

2. Input Data Gejala

Menu dari daftar gejala untuk melihat informasi gejala kerusakan processor, dapat dilihat pada gambar III.10.

Gejala Processor Rusak	
Solusi Pakar	
Kode	<input type="text"/>
Keterangan	<input type="text"/>
Nilai	<input type="text"/>
Rujukan Kerusakan	<input type="text"/>
<input style="width: 150px; height: 50px; margin-bottom: 10px;" type="text"/> <div style="display: flex; justify-content: space-around;"> <input type="button" value="New"/> <input type="button" value="Save"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Cancel"/> <input type="button" value="Exit"/> </div>	

Gambar III.10. Desain Gejala

3. Input Data Kerusakan

Menu input kerusakan terdiri dari beberapa *field* yang harus diinputkan, dapat dilihat pada gambar III.11.

Input Data Kerusakan		
Kode	<input type="text"/>	New
Keterangan	<input type="text"/>	Save
Solusi	<input type="text"/>	Edit
		Delete
		Cancel
		Exit

Gambar III.11. Input Data Kerusakan

5. Desain Deteksi

Desain deteksi ini digunakan untuk mendeteksi kerusakan processor, mulai dari penginputan *user*, menjawab pertanyaan dari pakar sampai dengan menampilkan solusi kerusakan *processor*. terdiri dari beberapa *field* yang harus diinputkan, dapat dilihat pada gambar III.12.

Deteksi Kerusakan	
Pertanyaan	
<input type="text"/>	
Tidak	Ya

Gambar III.12. Desain Input User

III.6. Kamus Data

Adapun kamus data dari aplikasi ini adalah sebagai berikut :

- 1) Analisa hasil : #ID, nama, kelamin, alamat, pekerjaan, KD_kerusakan
- 2) Kerusakan : #ID_Gejala, Kerusakan, definisi, solusi
- 3) Rekaman : # nama, kerusakan, tanggal
- 4) Gejala : #ID_Gejala, nm_gejala

Keterangan :

: Primary Key

III.7. Normalisasi

Normalisasi dari aplikasi yang akan dibuat:

1. Kerusakan : #ID_Gejala, Kerusakan, definisi, solusi
 2. Gejala : #ID_Gejala, nm_gejala
1. Normalisasi pertama 1NF

Tabel III.6. Normalisasi Pertama 1NF

Id_Gejala	Kerusakan	Definisi	Solusi
Xxxx	Xxxx	Xxx	Xxx

2. Normalisasi kedua 2NF

- a. Tabel Kerusakan

Tabel III.7. Normalisasi Kedua 2NF

Id_gejala	Kerusakan	Definisi	Solusi
Xxx	Xxx	Xxx	Xxx

b. Tabel Gejala

Tabel III.8. Normalisasi Kedua 2NF

Id_gejala	Nama_gejala
XXX	Xxx

3. Normalisasi ketiga 3NF

a. Tabel Gejala

Tabel III.9. Normalisasi Ketiga 3NF

*Id	Kerusakan	definisi	solusi
Xxx	Xxx	Xxx	Xxx

b. Tabel Kerusakan

Tabel III.10. Normalisasi Ketiga 3NF

*Id	Kerusakan
Xxx	Xxx

III.8. Struktur Tabel Database

Adapun rancangan tabel database yang penulis gunakan dalam Sistem Pakar kerusakan processor pada komputer adalah sebagai berikut:

1. Tabel User

Tabel admin ini digunakan untuk menyimpan *record* data akun dengan properti atau atribut id, *user*, *password* dan namanya.

Nama Database : processor

Nama Tabel : user

Primary Key : username

Tabel III.11. Tabel Data User

Nama Field	Tipe Data	Panjang	Keterangan
Username	varchar	20	Nama User
Password	varchar	10	Password User
Nama_lengkap	varchar	35	Nama Lengkap

2. Tabel Gejala

Tabel gejala ini digunakan untuk menyimpan *record* data tentang gejala

Nama Database : processor

Nama Tabel : gejala

Primary Key : id

Tabel III.12. Tabel Data Gejala

Nama Field	Tipe Data	Panjang	Keterangan
Id	int	5	Id Kecamatan
Kode	varchar	16	Kode Gejala
Gejala	Text	-	Gejala
Status	Tinyint	2	Status

3. Tabel Kerusakan

Tabel kerusakan ini digunakan untuk menyimpan *record* data tentang kerusakan

Nama Database : processor

Nama Tabel : kerusakan

Primary Key : id

Tabel III.13. Tabel Data Kerusakan

Nama Field	Tipe Data	Panjang	Keterangan
Id	int	11	Id kerusakan
Nama	varchar	50	Nama Kerusakan
Solusi	Text	-	Solusi

4. Tabel Rekam

Tabel rekam ini digunakan untuk menyimpan *record* data tentang kerusakan

Nama Database : processor

Nama Tabel : rekam

Tabel III.14. Tabel Data Rekam

Nama Field	Tipe Data	Panjang	Keterangan
Nama	Varchar	50	Nama kerusakan
Tanggal	Date		Tanggal
Kerusakan	Text	-	Kerusakan
Nilai	Double	-	Nilai
Keterangan	Text	-	Keterangan

III.9. Menentukan Variabel

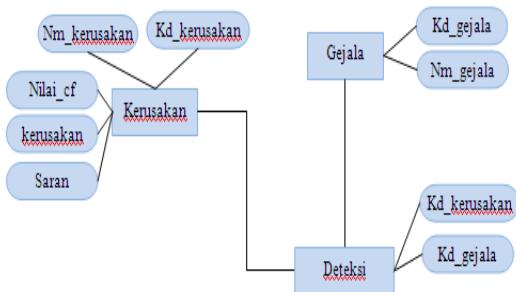
Input dari sistem adalah gejala dan intensitasnya. Intensitasnya merupakan variable fuzzy yang merepresentasikan gejala, sehingga harus ditentukan variable linguistic yang akan menentukan himpunan fuzzy dari nilai intensitas.

Sebagai output dari sistem adalah peluang dari kerusakan. Maka harus ditentukan pula variable linguistic yang akan digunakan untuk merepresentasikan himpunan fuzzy untuk intensitas gejala serta domainnya ditentukan

Himpunan intensitas fuzzy rendah memiliki domain 0.1 dan 0.9, Sedang memiliki domain 0.1 dan 1.0. Domain dari himpunan fuzzy akan mempengaruhi output dari sistem, maka dalam sistem dibuat sebuah prosedur untuk melakukan perubahan pada domain tersebut.

III.10. ERD (*Entity Relationship Diagram*)

Adapun ERD (*Entity Relationship Diagram*) dari aplikasi ini ditunjukkan pada gambar III.13.



Gambar III.13. Diagram ERD

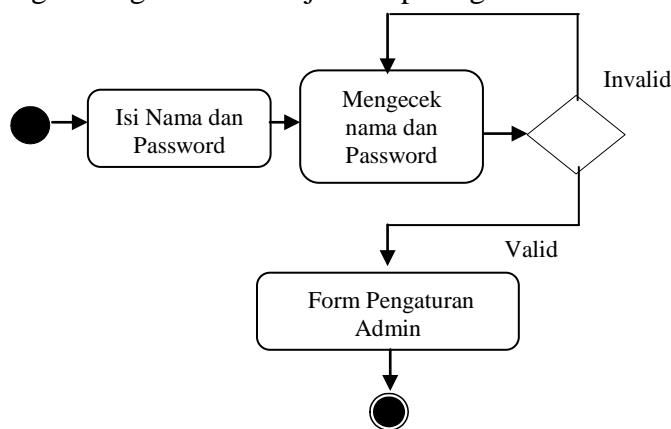
III.11. Logika Program

III.12. *Activity Diagram*

1. *Activity Diagram Login Admin*

Activity diagram login merupakan *activity diagram* untuk proses *login user*.

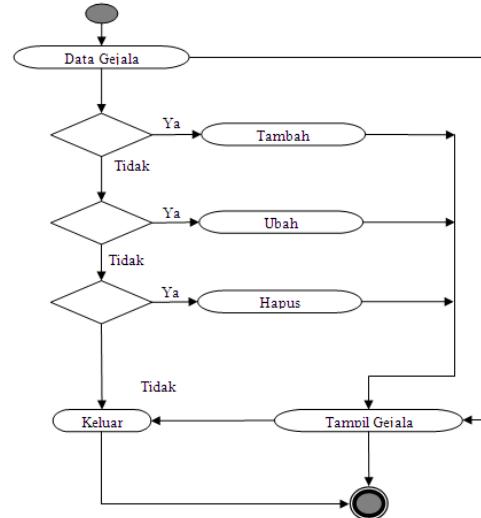
Activity diagram login user ditunjukkan pada gambar III.14.



Gambar III.14 Activity Diagram login Admin

2. *Activity Diagram Gejala*

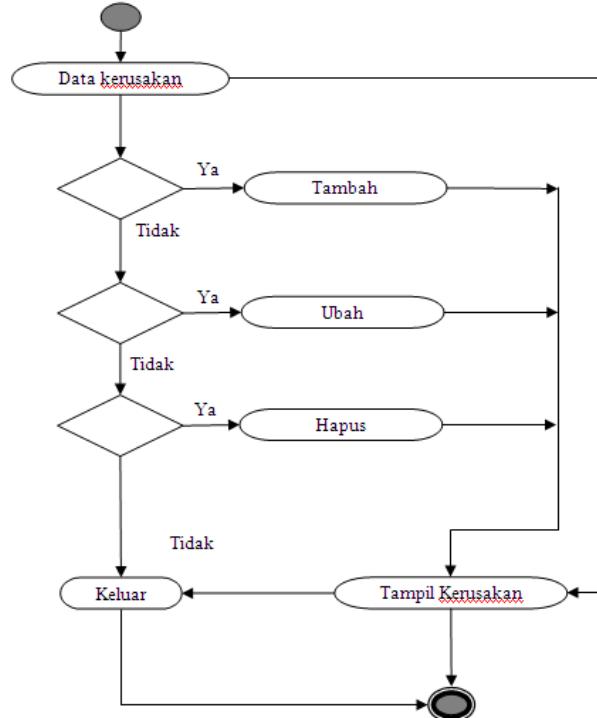
Activity diagram gejala merupakan *activity diagram* untuk proses pemberian gejala mengenai kerusakan *processor*. *Activity diagram* tersebut ditunjukkan pada gambar III.15.



Gambar III.15. Activity Gejala

3. Activity Diagram Kerusakan

Activity diagram kerusakan merupakan *activity* diagram untuk memproses kerusakan *processor*. *Activity* diagram tersebut ditunjukkan pada gambar III.16.



Gambar III.16. Activity Kerusakan