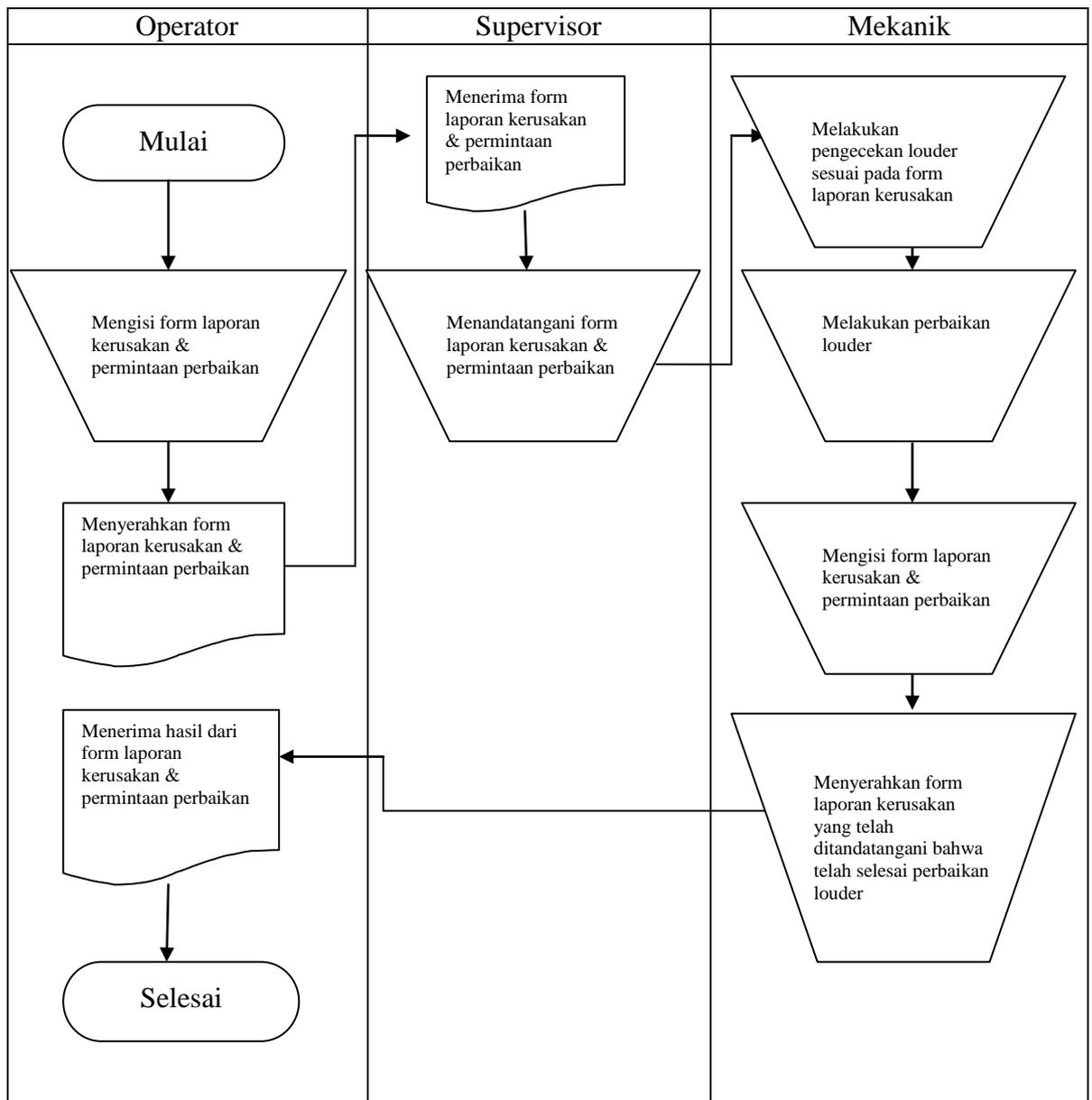




Dari gambar III.1 diatas dapat dilihat proses *input* data dalam laporan kerusakan dan permintaan perbaikan, bahwa operator mengetahui adanya kerusakan pada *Louder* kemudian mengisi form laporan kerusakan dan permintaan perbaikan lalu menginformasikan ke Kepala bagian/Supervisor, kemudian form laporan kerusakan dan permintaan perbaikan yang sudah ditanda tangani oleh operator dan Kepala bagian/Supervisor kemudian dilaporkan ke mekanik untuk melakukan perbaikan pada *Louder*.

### **III.1.2. Proses**

Adapun proses pengolahan data kerusakan *Louder* pada PT. Growth Sumatra yang sedang berjalan dapat digambarkan dalam bentuk FOD ( *Flow Of Document* ) berikut ini :



**Gambar III.2. FOD ( Flow Of Document ) Sistem Pakar Mendiagnosa Kerusakan Loudrer**  
*(Sumber : PT. Growth Sumatra Medan)*

Dari gambar III.2 di atas dapat dilihat proses pengolahan data kerusakan Loudrer pada PT. Growth Sumatra yang sedang berjalan, bahwa aliran dokumen ini sudah cukup baik, sebab terdapat proses pengolahan data yang berguna untuk



Dari gambar III.3 di atas dapat dilihat proses *output* data dalam laporan kerusakan dan permintaan perbaikan, bahwa mekanik mengerjakan kerusakan *Louder* berdasarkan laporan yang ada pada form laporan kerusakan dan permintaan perbaikan yang diberikan supervisor, kemudian mekanik mengisi form laporan kerusakan dan permintaan perbaikan sesuai dengan tindakan yang diambil dari kerusakan *Louder* tersebut dan setelah selesai, form laporan kerusakan dan permintaan perbaikan selanjutnya di informasikan ke operator bahwa perbaikan *Louder* telah selesai dikerjakan.

### **III.2. Evaluasi Sistem yang Berjalan**

Dalam hal ini sistem yang digunakan belum efektif dikarenakan mendiagnosa kerusakan *Louder* yang ada masih tergolong manual. Pengolahan data mendiagnosa kerusakan *Louder* pada PT. Growth Sumatra yang masih sederhana ini membuat pelaporan terkadang bermasalah dalam ketepatan data kerusakan *Louder*, tidak jarang juga bermasalah dari segi pendataan tanggal pelaporan. Dengan masalah tersebut penulis membuat sistem dengan bahasa pemrograman *Java* dengan editor *Netbeans* dan database *MySQL*.

### **III.3. Desain Sistem**

Untuk membantu membangun rancang bangun sistem pakar mendiagnosa kerusakan *Louder* pada PT. Growth Sumatra, penulis mengusulkan pembuatan sebuah sistem dengan menggunakan aplikasi program yang lebih akurat dan lebih

mudah dalam pengolahannya. Dengan menggunakan bahasa pemrograman *Java* dengan editor *Netbeans* dan database *MySQL*. Desain perancangan menggunakan UML (*Unified Modeling Language*).

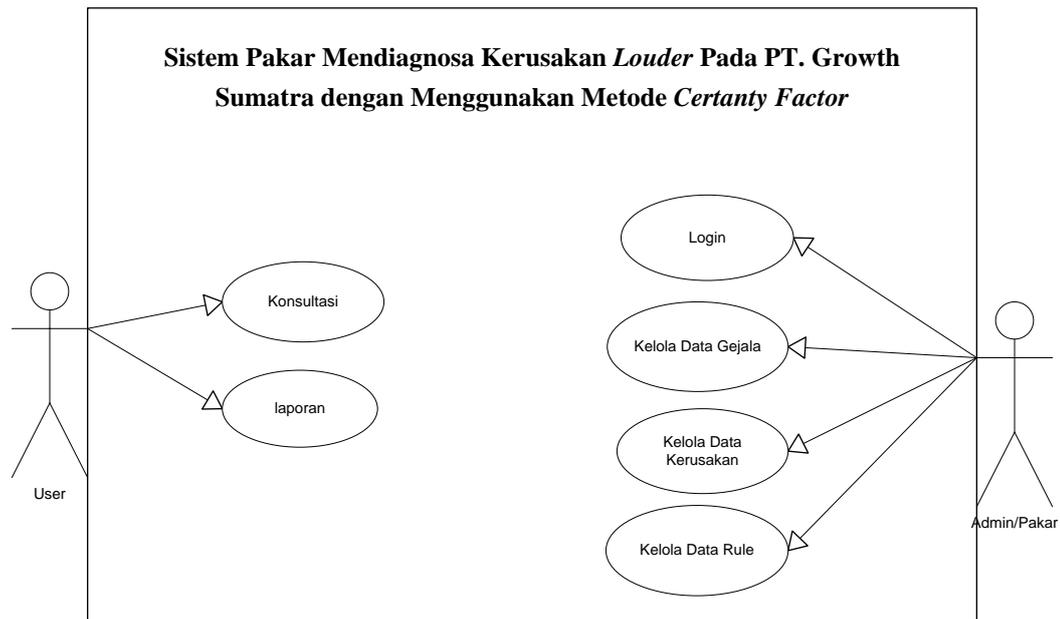
### **III.3.1 Desain Sistem Secara Global**

Pada perancangan sistem ini terdiri dari tahap perancangan yaitu :

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

#### **III.3.1.1. Use Case Diagram**

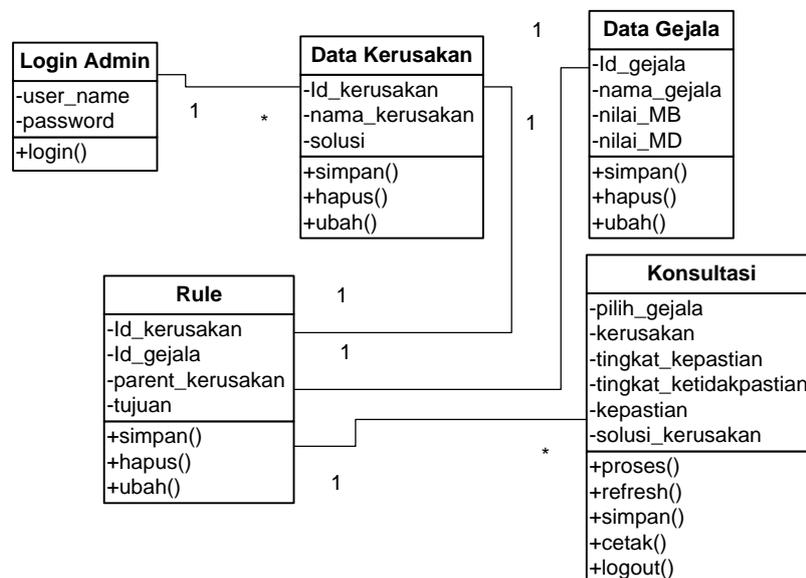
Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk *actor*. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*, dapat dilihat pada gambar III.4. Sebagai berikut :



**Gambar III.4. Use Case Diagram Sistem Pakar Mendiagnosa Kerusakan *Louder* Pada PT. Growth Sumatra dengan Menggunakan Metode *Certainty Factor***

### III.3.1.2. Class Diagram

*Class Diagram* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi), berikut gambar *Class Diagram* :



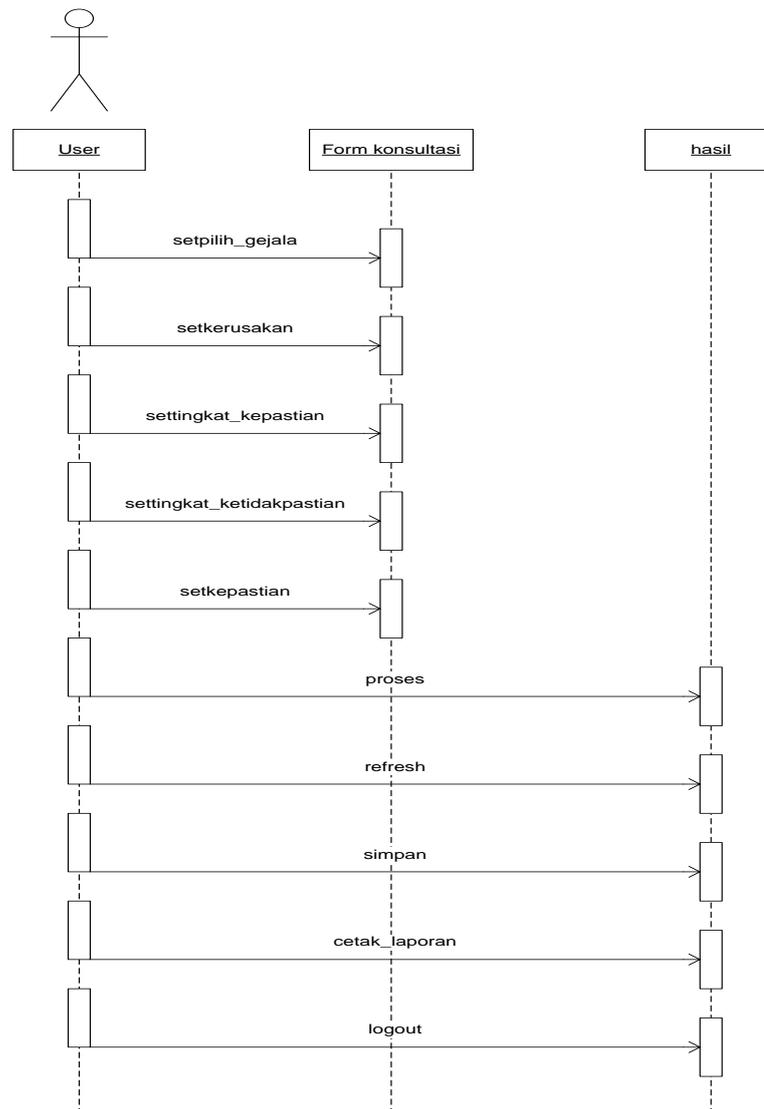
**Gambar III.5. Class Diagram Sistem Pakar Mendiagnosa Kerusakan Louder Pada PT. Growth Sumatra dengan Menggunakan Metode Certainty Factor**

### III.3.1.3. Sequence Diagram

*Sequence diagram* menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*, berikut beberapa gambar *sequence diagram* :

#### 1. Sequence Diagram Konsultasi

*Sequence Diagram* Konsultasi menggambarkan aktivitas *user* untuk melakukan konsultasi kerusakan dan mengetahui bagaimana solusinya. Adapun *sequence diagram* dapat dilihat pada gambar III.6 sebagai berikut :

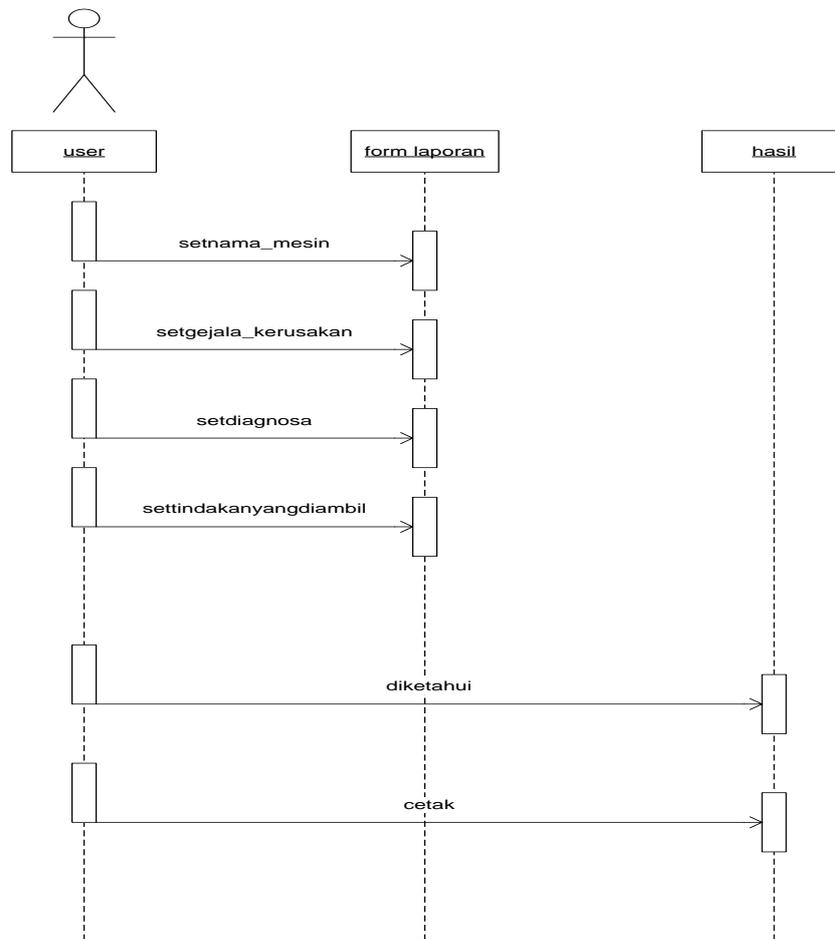


**Gambar III.6. Sequence Diagram Konsultasi**

## 2. Sequence Diagram Laporan

*Sequence Diagram* Laporan menggambarkan aktivitas *user* untuk melakukan laporan kerusakan dan mencetak hasil laporan kerusakan tersebut.

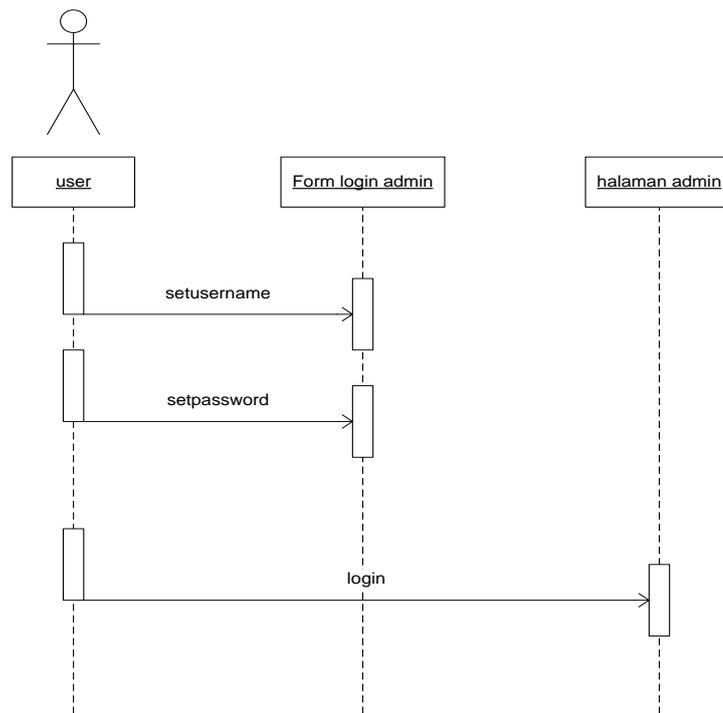
Adapun *sequence diagram* dapat dilihat pada gambar III.7 sebagai berikut :



**Gambar III.7. Sequence Diagram Laporan**

### 3. Sequence Diagram Login Admin/Pakar

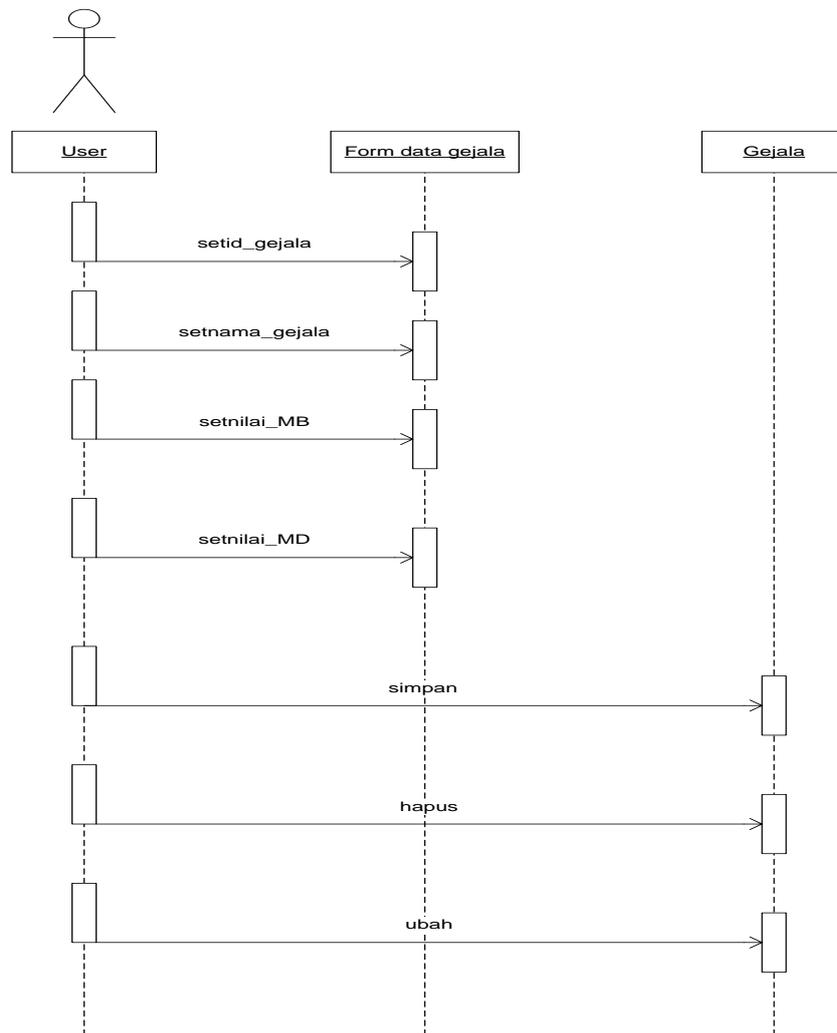
*Sequence Diagram Login Admin* menggambarkan aktivitas administrasi melakukan *login* sebelum masuk ke halaman admin untuk melakukan aktivitas yang lain administrasi harus menginputkan *username* dan *password* yang valid dan kemudian akan diproses dan akan diketahui *username* dan *password* tersebut valid atau tidak. Adapun *sequence diagram* dapat dilihat pada gambar III.8 sebagai berikut :



**Gambar III.8. Sequence Diagram Login Admin/Pakar**

#### 4. Sequence Diagram Data Gejala

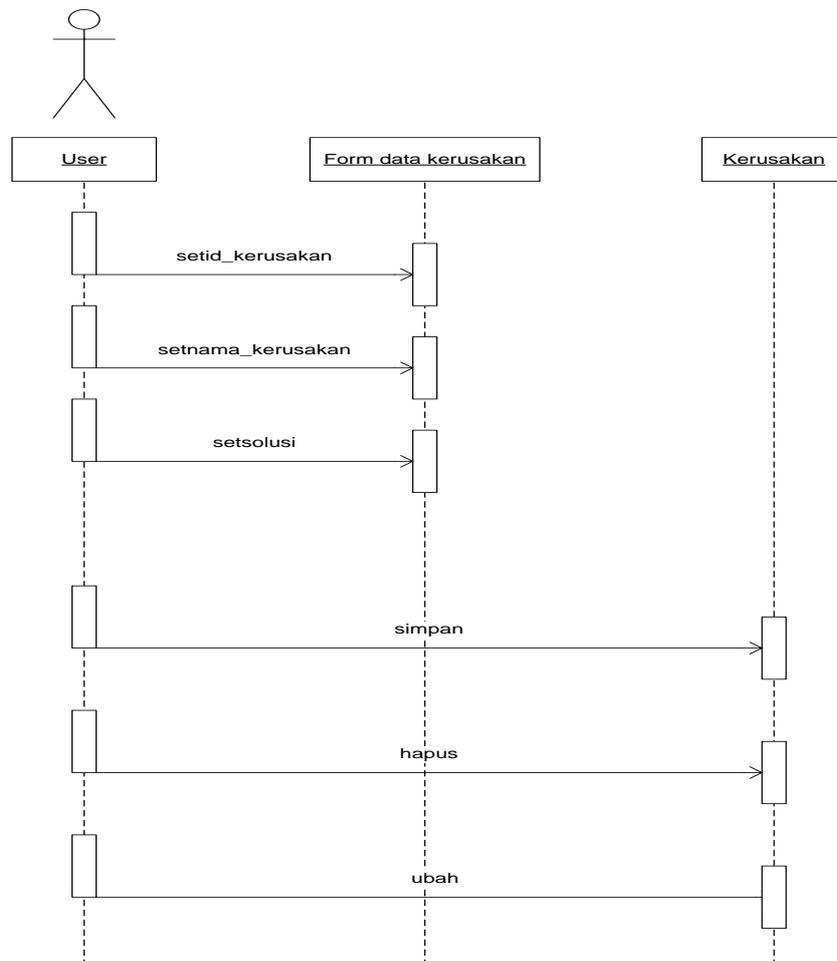
*Sequence Diagram* Data Gejala menggambarkan aktivitas administrasi untuk melakukan entri data gejala ke *database* dimana administrasi dapat melakukan menyimpan dan menghapus data gejala tersebut. Adapun *sequence diagram* dapat dilihat pada gambar III.9 sebagai berikut :



**Gambar III.9. Sequence Diagram Data Gejala**

### 5. Sequence Diagram Data Kerusakan

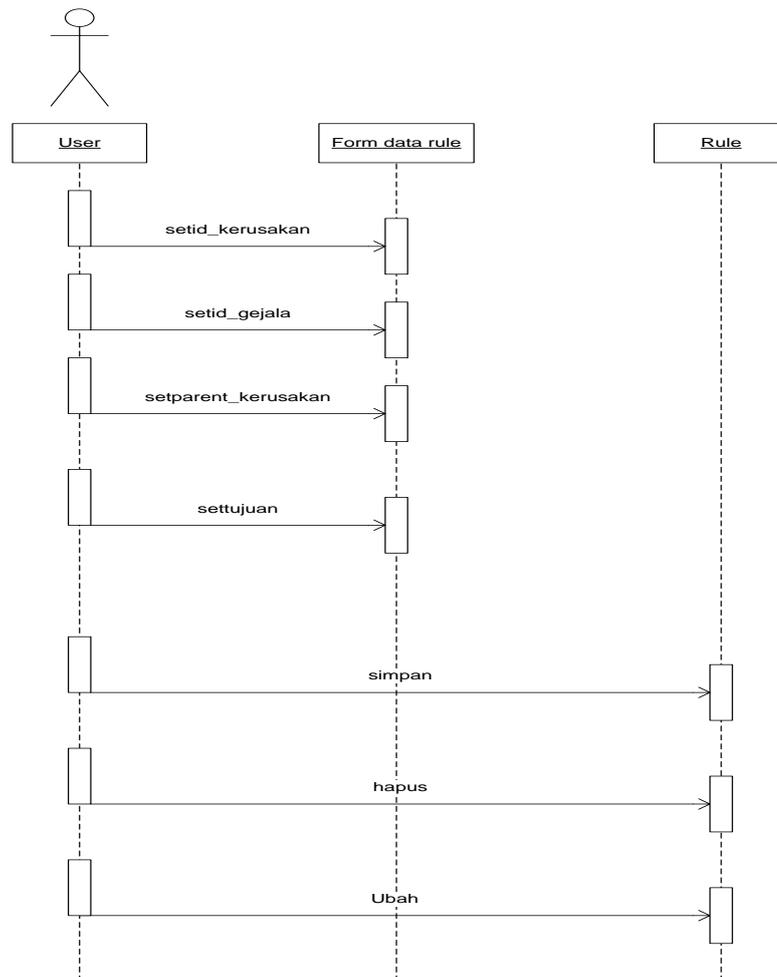
*Sequence Diagram Data Kerusakan* menggambarkan aktivitas administrasi untuk melakukan entri data kerusakan ke *database* dimana administrasi dapat melakukan menyimpan dan menghapus data kerusakan tersebut. Adapun *sequence diagram* dapat dilihat pada gambar III.10 sebagai berikut :



**Gambar III.10. Sequence Diagram Data Kerusakan**

#### 6. Sequence Diagram Data Rule

*Sequence Diagram Data Rule* menggambarkan aktivitas administrasi untuk melakukan kumpulan *rule - rule* kerusakan *Louder* dimana administrasi dapat melakukan menyimpan dan menghapus data kumpulan *rule - rule* kerusakan *Louder* tersebut. Adapun *sequence diagram* dapat dilihat pada gambar III.11 sebagai berikut :



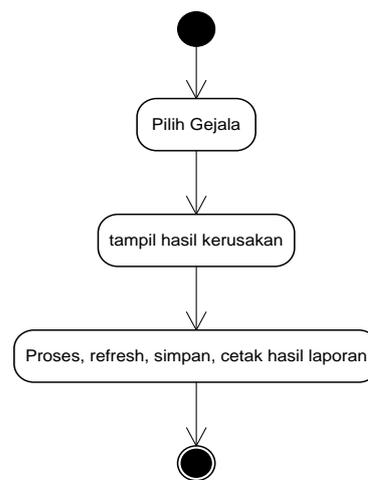
**Gambar III.11. Sequence Diagram Data Rule**

#### III.3.1.4. Activity Diagram

Menggambarkan aktifitas - aktifitas, objek, *state*, transisi *state* dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas, berikut beberapa gambar *Activity Diagram* :

### 1. *Activity Diagram* Konsultasi

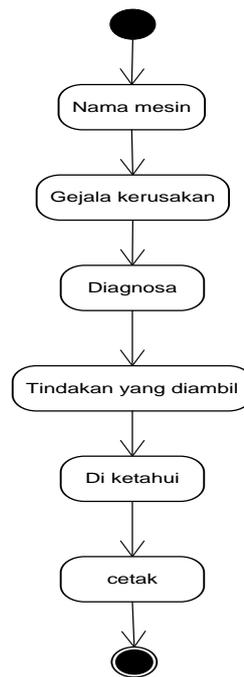
*Activity Diagram* Konsultasi menggambarkan alir aktivitas *user* untuk melakukan konsultasi kerusakan dengan jawab pertanyaan dan mengetahui bagaimana solusinya yang dapat disimpan dan dicetak hasil laporan. Adapun *activity diagram* dapat dilihat pada gambar III.12 sebagai berikut :



**Gambar III.12. *Activity Diagram* Konsultasi**

### 2. *Activity Diagram* Laporan

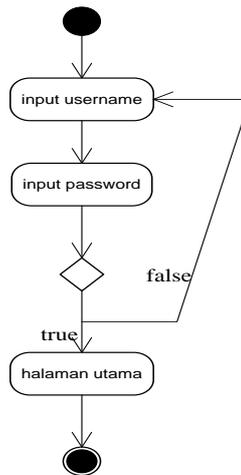
*Activity Diagram* Laporan menggambarkan alir aktivitas *user* untuk melakukan laporan kerusakan dan mencetak hasil laporan kerusakan tersebut. Adapun *activity diagram* dapat dilihat pada gambar III.13 sebagai berikut :



**Gambar III.13. Activity Diagram Laporan**

### 3. Activity Diagram Login Admin/Pakar

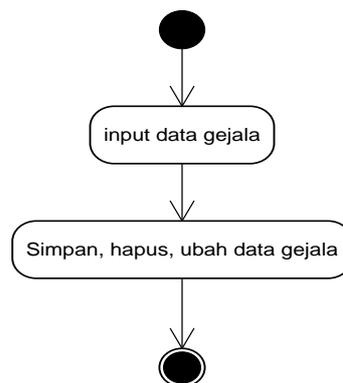
*Sequence Diagram Login Admin* menggambarkan alir aktivitas administrasi melakukan *login* sebelum masuk ke halaman admin untuk melakukan aktivitas yang lain administrasi harus menginputkan *username* dan *password* yang valid dan kemudian akan diproses dan akan diketahui *username* dan *password* tersebut valid atau tidak. Adapun *activity diagram* dapat dilihat pada gambar III.14 sebagai berikut :



**Gambar III.14. Activity Diagram Login Admin/Pakar**

#### 4. Activity Diagram Data Gejala

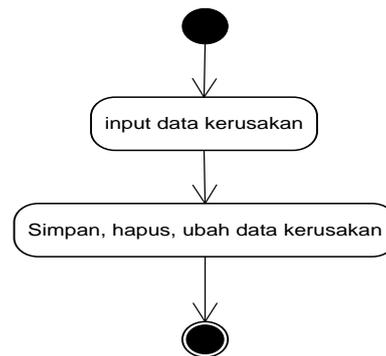
*Activity Diagram* Data Gejala menggambarkan alir aktivitas administrasi untuk melakukan input data gejala dimana administrasi dapat melakukan menyimpan dan menghapus data gejala tersebut. Adapun *activity diagram* dapat dilihat pada gambar III.15 sebagai berikut :



**Gambar III.15. Activity Diagram Data Gejala**

### 5. Activity Diagram Data Kerusakan

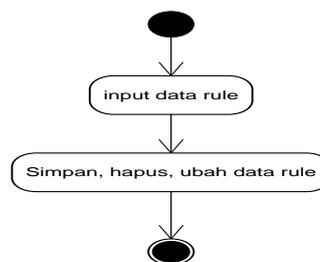
*Activity Diagram* Data Kerusakan menggambarkan alir aktivitas administrasi untuk melakukan input data kerusakan dimana administrasi dapat melakukan menyimpan dan menghapus data kerusakan tersebut. Adapun *activity diagram* dapat dilihat pada gambar III.16 sebagai berikut :



**Gambar III.16. Activity Diagram Data Kerusakan**

### 6. Activity Diagram Data Rule

*Activity Diagram* Data Rule menggambarkan alir aktivitas administrasi untuk melakukan input data *rule - rule* kerusakan *Louder* dimana administrasi dapat melakukan menyimpan dan menghapus data *rule - rule* kerusakan *Louder* tersebut. Adapun *activity diagram* dapat dilihat pada gambar III.17 sebagai berikut :



**Gambar III.17. Activity Diagram Data Rule**

### III.3.1.5. Rule - rule Certainty Factor

Faktor Kepastian (*Certainty Factor*) adalah salah satu metode yang menunjukkan ukuran kepastian terhadap suatu fakta atau aturan. Variabel yang akan dibahas dalam suatu metode *Certainty Factor* yaitu dalam bentuk kumpulan *rule - rule* kerusakan *Louder*, dapat dilihat pada tabel *rule* kerusakan *Louder* sebagai berikut :

**Tabel III.1. Kumpulan Rule - rule Gejala Louder**

Aksi	Kode Gejala	Gejala	Nilai MB	Nilai MD	Kerusakan
√	G01	Ada kebocoran pada pompa	0.1	0.01	K01
√	G02	Rumah blok pompa sudah termakan gear	0.2	0.02	K01
√	G03	Minyak terlalu encer	0.3	0.03	K01
√	G04	Keluar asap tebal	0.4	0.04	K02
√	G05	Tenaga berkurang	0.5	0.05	K02
√	G06	Siklus tidak lancar	0.6	0.06	K02
√	G07	Air cleaner cepat kotor	0.7	0.07	K02
√	G08	Rem blong	0.8	0.08	K03
√	G09	Kebocoran dari seal master rem	0.9	0.09	K03
√	G10	Minyak rem sering berkurang	0.1	0.011	K03
√	G11	Setiur sulit digerakkan	0.11	0.01	K04
√	G12	Pomp hidrolis ada kebocoran	0.9	0.02	K04
√	G13	Break/disk aus	0.8	0.03	K04
√	G14	Tenaga untuk berjalan berkurang	0.7	0.04	K05
√	G15	Pomp oli transmisi berfungsi	0.6	0.05	K05
√	G16	Disk/plat aus	0.5	0.06	K05
√	G17	Seal yang bocor	0.4	0.07	K05
√	G18	Ring seal bocor	0.3	0.08	K05
√	G19	As roda silang 4 ada yang patah	0.2	0.09	K06
√	G20	Gear aus	0.1	0.02	K06
√	G21	Terdengar suara kasar waktu berjalan	0.5	0.01	K06
√	G22	Bearing pecah	0.8	0.02	K06
√	G23	Pomp hidrolis preser tidak mencukupi	0.4	0.01	K07
√	G24	Jarum relay valve aus	0.7	0.02	K07
√	G25	Jarum relay valve bocor	0.2	0.03	K07

√	G26	Kontrol valve bocor/sumbat	0.6	0.04	K07
√	G27	Filter oli hidrolis kotor	0.3	0.05	K07
√	G28	Tidak mau stater	0.8	0.06	K08
√	G29	Relay bagus	0.5	0.07	K08
√	G30	Kunci kontak bagus	0.9	0.08	K08
√	G31	String tidak putus	0.4	0.01	K08
√	G32	Mesin cepat panas	0.7	0.02	K09
√	G33	Sirkulasi radiator bagus	0.5	0.03	K09
√	G34	Pengaturan sirkulasi air bagus	0.8	0.02	K09
√	G35	Air ada berkurang	0.5	0.01	K09

(Sumber : PT. Growth Sumatra Medan)

Adapun tabel gejala pada *Louder* yaitu dapat dilihat sebagai berikut :

**Tabel III.2. Kerusakan pada Louder**

KODE	KERUSAKAN	SOLUSI
K01	Pomp Hidrolik	Ganti pomp hidrolik dengan yang baru
K02	Turbo Charge	Turbo charge dapat diganti dengan yang baru
K03	Master Rem	Ganti master rem dengan yang baru
K04	Power Stearing	Dapat diganti dengan pomp hidrolik yang baru
K05	Transmision	Ganti pom oli, ganti plat yang aus dengan yang baru, ganti seal dan ring seal yang baru
K06	Gerdang	Ganti gerdang dengan yang baru
K07	Bucket	Perbaiki pomp hidrolik preser
K08	Dinamo Start	Ganti dinamo start dengan yang baru
K09	Air Radiator	Ganti air radiator dengan yang baru

(Sumber : PT. Growth Sumatra Medan)

### III.3.2. Desain Sistem Secara Detail

Desain sistem secara detail dari sistem pakar mendiagnosa kerusakan *Louder* (Alat Berat) pada PT. Growth Sumatra dengan menggunakan metode *certanty factor* ini adalah sebagai berikut :

#### III.3.2.1. Desain Output

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

##### 1. Desain Output Laporan

Rancangan *output* laporan berfungsi menampilkan data laporan anggota. Adapun rancangan *output* laporan dapat dilihat pada Gambar III.18 sebagai berikut :

LAPORAN KERUSAKAN & PERMINTAAN PERBAIKAN		No.Dokumen/Revisi GSI-SMS-MNT-FM-07/0		
Kepada : xxxxx				
Dari Bagian : xxxxx				
Bersama ini kami memohon perbaikan mesin/alat sebagai berikut :				
No	Nama Mesin	Gejala Kerusakan	Diagnosa	Tindakan Yang Diambil
x	xxxx	xxxx	xxxx	xxxx
x	xxxx	xxxx	xxxx	xxxx
x	xxxx	xxxx	xxxx	xxxx
Tanggal : Dilaporkan Oleh User,		Tanggal : Dilaksanakan Oleh :		Tanggal : Verifikasi Penyelesaian :
Tanggal : Diketahui Oleh :				
( xxxxx )	( xxxxx )	( xxxxx )	( xxxxx )	( xxxxx )

**Gambar III.18. Desain Output Laporan**

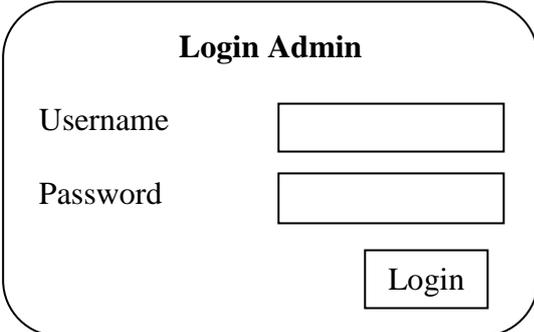
### III.3.2.2. Desain Input

Perancangan *input* merupakan masukan yang penulis desain guna lebih memudahkan dalam *entry* data. *Entry* data yang dirancang akan lebih mudah dan cepat untuk meminimalisir kesalahan penulisan dan memudahkan perubahan.

Perancangan *input* tampilan yang dirancang adalah sebagai berikut :

#### 1. Desain Input Login Admin/Pakar

Perancangan *input login admin/pakar* berfungsi untuk verifikasi pengguna yang berhak menggunakan sistem. Adapun rancangan *login admin/pakar* dapat dilihat pada Gambar III.19 sebagai berikut :



**Login Admin**

Username

Password

**Gambar III.19. Desain Input Login Admin/Pakar**

#### 2. Desain Input Data Gejala

Perancangan *input* data gejala merupakan form untuk penyimpanan data - data gejala, nilai kepastian dan nilai ketidakpastian. Adapun bentuk *input* data gejala dapat dilihat pada Gambar III.20 Sebagai berikut :

Konsultasi	Data_Gejala	Data_Kerusakan	Buat_Rule	History Konsultasi	Logout
------------	-------------	----------------	-----------	--------------------	--------

**Data Gejala**

Id_Gejala	Nama_Gejala	MB	MD
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx

Id Gejala

Nama Gejala

Nilai MB

Nilai MD

**Gambar III.20. Desain Input Data Gejala**

### 3. Desain *Input* Data Kerusakan

Perancangan *input* data kerusakan merupakan form untuk penyimpanan data - data kerusakan dan solusi kerusakan. Adapun bentuk *input* data kerusakan dapat dilihat pada Gambar III.21 Sebagai berikut :

Konsultasi
Data\_Gejala
Data\_Kerusakan
Buat\_Rule
History Konsultasi
Logout

**Data Kerusakan**

Id_Kerusakan	Nama_Kerusakan	Solusi
xxx	xxx	xxx
xxx	xxx	xxx
xxx	xxx	xxx

Id\_Kerusakan

Nama\_Kerusakan

Solusi

**Gambar III.21. Desain Input Data Kerusakan**

#### 4. Desain *Input Data Rule*

Perancangan *input data rule* merupakan form untuk penyimpanan kumpulan data - data kerusakan *Louder*. Adapun bentuk *input data rule* dapat dilihat pada Gambar III.22 sebagai berikut :



Konsultasi
Data\_Gejala
Data\_Kerusakan
Buat\_Rule
History Konsultasi
Logout

**Pembuatan Rule**

Id_Rule	Id_Kerusakan	Parent_Kerusakan	Tujuan
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX

Id\_Kerusakan

Id\_Gejala

**Gambar III.22. Desain Input Data Rule**

### III.3.2.3. Desain Database

#### 1. Kamus data (*Data Dictionaries*)

Kamus data merupakan suatu daftar terorganisasi tentang komposisi elemen data, aliran data dan data store yang digunakan. Pengisian data dictionary dilakukan setiap saat selama proses pengembangan berlangsung, ketika diketahui adanya data atau saat diperlukan penambahan data item ke dalam sistem. Berikut kamus data dari sistem pakar mendiagnosa kerusakan *Louder* sebagai berikut :

1. Login = ({username + password})
2. Data\_gejala = ({id\_gejala + nama\_gejala + nilai\_MB + nilai\_MD})
3. Data\_kerusakan = ({id\_kerusakan + nama\_kerusakan + solusi})

4. Rule = ({id\_kerusakan + id\_gejala + parent\_kerusakan + tujuan})
5. Konsultasi = ({pilih\_gejala + proses + simpan + cetak})

## 2. Normalisasi

Normalisasi adalah proses pengelompokan data ke dalam bentuk tabel atau relasi atau file untuk menyatakan entitas dan hubungan mereka sehingga terwujud satu bentuk database yang mudah untuk dimodifikasi. Berikut ini adalah bentuk normalisasi yang dilakukan dalam sistem mendiagnosa kerusakan *Louder* sebagai berikut :

### First Normal Form (1NF)

Bentuk normal yang pertama atau 1NF mensyaratkan beberapa kondisi dalam sebuah database, berikut adalah fungsi dari bentuk normal pertama ini.

- Menghilangkan duplikasi kolom dari tabel yang sama.
- Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (primary key).

Contoh Normalisasi Database 1NF :

Id_gejala	Nama_gejala	MB	MD
G01	Kebocoran pada pompa	0.1	0.01
G02	Rumah blok pompa sudah termakan	0.2	0.02
G04	Keluar asap tebal	0.4	0.04

Id_kerusakan	Nama_kerusakan	Solusi
K01	Pomp Hidrolik	Ganti pomp hidrolik dengan yang baru
K02	Turbo Carge	Turbo carge ganti dengan yang baru

Id_Rule	Id_Kerusakan	Parent_Kerusakan	Tujuan
R01	K01		K01
R02		K01	G01
R03		K01	G02
R04		K01	G03
R05	K02		K02

Pada intinya bentuk normalisasi 1NF ini mengelompokkan beberapa tipe data atau kelompok data yang sejenis agar dapat dipisahkan sehingga data dapat di atasi. Contoh adalah ketika kita ingin menghapus, mengupdate, atau menambahkan data di tabel gejala, maka kita tidak bersinggungan dengan data tabel kerusakan atau data tabel rule. Sehingga data dapat mulai di jaga.

### **Second Normal Form (2NF)**

Syarat untuk menerapkan normalisasi bentuk kedua ini adalah data telah dibentuk dalam 1NF, berikut adalah beberapa fungsi normalisasi 2NF.

- Menghapus beberapa subset data yang ada pada tabel dan menempatkan mereka pada tabel terpisah.
- Menciptakan hubungan antara tabel baru dan tabel lama dengan menciptakan foreign key.

Contoh Normalisasi Database 2NF :

Nama_Gejala	Id_Gejala	Id_Kerusakan	Id_Rule	Nama_Kerusakan
Kebocoran pada pompa	G01	K01	R01	Pomp Hidrolik
Rumah blok pompa sudah termakan	G02	K01	R02	Pomp Hidrolik

Bentuk 2NF dari tabel diatas

Id_Gejala	Id_Kerusakan	Id_rule
G01	K01	R01
G02	K01	R02

pada intinya bentuk kedua ini adalah tidak boleh ada field yang berhubungan dengan field lainnya secara fungsional. Contoh nama gejala tergantung dengan id\_gejala sehingga dalam bentuk 2NF nama gejala dapat di hilangkan karena telah memiliki tabel master tersendiri.

### 3. Desain Tabel/File

Perancangan struktur database adalah untuk menentukan *file database* yang digunakan seperti *field*, tipe data, ukuran data. Sistem ini didesain dengan menggunakan database *MYSQL*. Berikut adalah desain database dan tabel dari sistem yang dirancang.

#### a. Tabel Admin

Nama Database : sp\_loader

Nama Tabel : admin

Primary Key : username

**Tabel III.3. Tabel Admin**

Nama Field	Tipe Data	Ukuran	Keterangan
*username	varchar	20	Primary key
password	varchar	20	password

## b. Tabel Gejala

Nama Database : sp\_loader

Nama Tabel : gejala

Primary Key : id\_gejala

**Tabel III.5. Tabel Gejala**

Nama Field	Tipe Data	Ukuran	Keterangan
*Id_gejala	varchar	10	Primary key
Nama_gejala	varchar	50	Nama_gejala
MB	double	-	MB
MD	double	-	MD

## c. Tabel Kerusakan

Nama Database : sp\_loader

Nama Tabel : kerusakan

Primary Key : id\_kerusakan

**Tabel III.6. Tabel Kerusakan**

Nama Field	Tipe Data	Ukuran	Keterangan
*Id_keruskaan	varchar	10	Primary key
Nama_kerusakan	varvhar	30	Nama_kerusakan
solusi	text	-	solusi

## d. Tabel Rule

Nama Database : sp\_loader

Nama Tabel : rule

Primary Key : kode\_rule

**Tabel III.7. Tabel Rule**

Nama Field	Tipe Data	Ukuran	Keterangan
*Id_rule	int	20	Primary key
Id_kerusakan	varchar	20	Id_kerusakan
Parent_kerusakan	varchar	20	Parent_kerusakan
Tujuan	varchar	20	Tujuan

