# BAB III

# ANALISIS DAN PERANCANGAN

## 3.1 Analisis Sistem

Dalam membangun sebuah aplikasi untuk mendeteksi kerusakan pada sepeda motor dan cara penanganannya dilakukan beberapa tahap analisis yaitu :

1. Menentukan masalah yang akan dibangun untuk sebuah aplikasi sistem pakar. Sistem yang akan dibangun merupakan sebuah aplikasi untuk mendeteksi kerusakan pada sepeda motor berbasis web.
2. Mengumpulkan data yang diperlukan untuk membangun sistem, yaitu berupa informasi tentang gejala, kerusakan dan cara penangannya melalui studi literatur dan observasi yang digunakan sebagai *base knowledge*.
3. Mempresentasikan pengetahuan ke dalam tabel gejala yang telah dianalisis, aturan produksi dan penelusuran gejala dan jenis kerusakan.
4. Usulan sistem yang akan dibuat.

### 3.1.1 Analisis Masalah

Secara umum banyak para pengguna sepeda motor tidak mengetahui kerusakan pada motornya sendiri. Dikarenakan sulitnya mendapatan pengetahuan untuk mengatasi masalah yang ada pada sepeda motor, baik dalam buku-buku pengetahuan atau pedoman sepeda motor maupun dari para ahli yang sudah berpengalaman di bidangnya.

Pada saat ini memang sudah banyak yang dapat dipelajari masyarakat pengguna motor, namun masalahnya banyak orang juga yang malas untuk pergi mendatangi perusahaan atau ahli-ahli yang terkait di bidangnya.

Oleh sebab itu dan berdasarkan analisis diatas untuk memudahkan pengetahuan bagi masyarakat pengguna sepeda motor, penulis membuat Tugas Akhir ini dengan alternatif penyajian informasi dan konsultasi tentang kerusakan pada sepeda motor serta cara penanganannya, yang diimplementasikan oleh penulis sendiri secara online melalui web master. Sebuah aplikasi web sebagai sistem pakar yang dapat mendeteksi jenis kerusakan pada sepeda motor dengan menggunakan *option* dan pilihan YA atau TIDAK yaitu menjawab gejala-gejala yang dirasakan dan masalah yang akan dianalisa yaitu tentang jenis kerusakan pada motor beserta gejala, penyebab dan penanganan atau solusinya.

### 3.1.2 Sumber Masalah

Data mengenai jenis kerusakan pada sepeda motor yaitu pengertian jenis kerusakan, gejala, penyebab serta penanganannya yang didapat dari buku, artikel, dan situs internet. Selain itu informasi mengenai jenis kerusakan pada sepeda motor didapat dari bapak Wawan S., S.T. Selaku kepala bagian teknisi di dealer Yamaha Majalengka, yang memberikan bimbingan dan pengarahan sekaligus sebagai pakar dalam mendeteksi kerusakan pada sepeda motor.

### 3.1.3 Identifikasi Masalah

Langkah pertama dalam mengembangkan aplikasi adalah mengidentifikasikan masalah yang akan dikaji. Dalam hal ini adalah dengan mengidentifikasikan permasalahan yang akan dibuat terlebih dahulu, adapun masalah-masalah yang akan diambil dalam aplikasi untuk mendeteksi kerusakan pada sepeda motor serta cara penanganannya atau solusinya.

Jenis kerusakan dan Gejalanya adalah sebagai berikut :

1. Motor sukar hidup

   Gejalanya adalah sebagai berikut :

   - Cek bensin, apakah kosong
   - Cek busi, apakah kotor
   - Cek karburator, apakah kotor
   - Cek selang dari tengki bahan bakar menuju karburator, apakah mampet atau longgar.

2. Mesin tidak dapat hidup sama sekali

   Gejalanya adalah sebagai berikut :

   - Cek bensin, apakah kosong
   - Cek busi, apakah kotor
   - Cek platina atau coil dengan ohm meter, apakah rusak
   - Cek cdi untuk motor yang menggunakan coil, apakah rusak
   - Cek kabel body menuju magnet dengan ohm meter, apakah putus
   - Cek spul magnet, apakah putus.

3. Mesin dapat hidup tapi tidak bertenaga

   Gejalanya adalah sebagai berikut :

   - Cek busi, apakah kotor
   - Cek karburator, apakah kotor
   - Cek saringan udara, apakah kotor
   - Cek bos katup, apakah rusak
   - Cek pegas katup, apakah lemah
   - Cek kompresi, apakah bocor.

4. Timbul suara menggelitik pada bagian blok silinder

   Gejalanya adalah sebagai berikut :

   - Cek bos katup, apakah rusak
   - Cek pegas katup, apakah lemah
   - Cek noken as, apakah aus
   - Cek torak dan ring torak, apakah aus
   - Cek stang torak dan laher stang, apakah longgar.

5. Bahan bakar boros

   Gejalanya adalah sebagai berikut :

   - Cek saringan udara, apakah kotor
   - Cek karburator, apakah kotor
   - Cek silinder head dan torak nya, apakah kotor
   - Cek kompresi, apakah bocor.

6. Keluar asap putih dari knalpot

   Gejalanya adalah sebagai berikut :

   - Cek torak dan cincin torak, apakah aus
   - Cek silinder, apakah aus
   - Cek oli, apakah cc oli sesuai dengan ketentuan yang tertera di blok mesin.

7. Mesin hidupnya tersendat-sendat setelah dipanaskan

   Gejalanya adalah sebagai berikut :

   - Cek busi, apakah kotor
   - Cek platina atau coil dengan ohm meter, apakah rusak
   - Cek karburator, apakah kotor.

8. Lampu besar mati

   Gejalanya adalah sebagai berikut :

   - Apakah kawat dalam lampu putus
   - Cek kabel reflector, apakah putus
   - Cek soket, apakah kabel dalam soket putus.

9. Lampu sein mati atau tidak berkedip

   Gejalanya adalah sebagai berikut :

   - Apakah kawat dalam lampu putus
   - Cek soket, apakah kabel dalam soket putus.

10. Lampu rem mati

    Gejalanya adalah sebagai berikut :

    - Apakah kawat dalam lampu putus
    - Cek soket, apakah kabel dalam soket putus.

11. Transmisi susah dipindahkan

    Gejalanya adalah sebagai berikut :

    - Cek plat kopling, apakah aus
    - Cek otomatis sentripugal, apakah aus.

### 3.1.4 Analisis data Gejala dan Kerusakan

Keberhasilan suatu aplikasi sistem pakar terletak pada pengetahuan dan bagaimana mengolah pengetahuan tersebut agar dapat ditarik suatu kesimpulan. Pengetahuan yang diperoleh dari hasil wawancara dan analisa lewat buku dikonversi kedalam sebuah tabel kerusakan dan gejala guna mempermudah proses pencarian solusi. Tabel kerusakan dan gejala ini digunakan sebagai pola pencocokan informasi yang dimasukan oleh pemakai dan basis pengetahuan.

Pada tabel kerusakan dan gejala terdapat 11 jenis kerusakan yang ditunjukan oleh KS001, KS002, ..,KS011. Dan 24 gejala yang ditunjukkan oleh GJ001, GJ002,…, GJ024. Dari 24 gejala disusun sebagai pernyataan dan 11 jenis kerusakan disusun sebagai kerusakan awal. Gejala ini merupakan basis pengetahuan untuk membuat suatu kesimpulan yang menjadi *goal.* Berikut ini adalah tabel kerusakan dan gejala yaitu pada Tabel 3.1 :

Tabel 3.1 Tabel kerusakan dan gejala

| | KERUSAKAN | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GEJALA | KS001 | KS002 | KS003 | KS004 | KS005 | KS006 | KS007 | KS008 | KS009 | KS010 | KS011 |
| GJ001 | * | * | * | | | | * | | | | |
| GJ002 | * | | * | | * | | * | | | | |
| GJ003 | * | | | | | | | | | | |
| GJ004 | | * | | | | | | | | | |
| GJ005 | | * | | | | | * | | | | |
| GJ006 | | * | | | | | | | | | |
| GJ007 | | * | | | * | | | | | | |
| GJ008 | | | * | | | | | | | | |
| GJ009 | | | * | * | | | | | | | |
| GJ010 | | | * | * | | | | | | | |
| GJ011 | | | * | * | | | | | | | |
| GJ012 | | | | * | | | | | | | |
| GJ013 | | | | * | * | | | | | | |
| GJ014 | | | | | | | | | | | |
| GJ015 | | | | | * | | | | | | |
| GJ016 | | | | | | * | | | | | |
| GJ017 | | | | | | * | | | | | |
| GJ018 | | | | | | * | | | | | |
| GJ019 | | | | | | | | * | * | * | |
| GJ020 | | | | | | | | * | | | |
| GJ021 | | | | | | | | * | * | * | |
| GJ022 | | | | | | | | | | | * |
| GJ023 | | | | | | | | | | | * |
| GJ024 | * | * | | | | | | | | | |

Keterangan :

KS001 : Mesin sukar hidup

KS002 : Mesin tidak dapat hidup sama sekali

KS003 : Mesin dapat hidup tapi tidak bertenaga

KS004 : Timbul suara menggelitik pada bagian blok silinder

KS005 : Bahan bakar boros

KS006 : Keluar asap putih dari knalpot

KS007 : Mesin hidupnya tersendat-sendat setelah dipanaskan

KS008 : Lampu besar mati

KS009 : Lampu sein mati atau tidak berkedip

KS010 : Lampu rem mati

KS011 : Transmisi susah dipindahkan

GJ001 : Cek busi, apakah kotor

GJ002 : Cek karburator, apakah kotor

GJ003 : Cek selang dari tengki bahan bakar menuju karburator, apakah mampet atau longgar

GJ004 : Cek platina atau coil dengan ohm meter, apakah rusak

GJ005 : Cek CDI untuk motor yang menggunakan coil, apakah rusak

GJ006 : Cek kabel body menuju magnet dengan ohm meter, apakah putus

GJ007 : Cek spul magnet, apakah putus

GJ008 : Cek saringan udara, apakah kotor

GJ009 : Cek bos katup, apakah rusak

GJ010 : Cek pegas katup, apakah lemah

GJ011 : Cek kompresi, apakah bocor

GJ012 : Cek noken as, apakah aus

GJ013 : Cek torak dan ring torak, apakah aus

GJ014 : Cek stang torak dan laher stang, apakah longgar

GJ015 : Cek silinder head dan torak nya, apakah kotor

GJ016 : Cek torak dan cincin torak, apakah aus

GJ017 : Cek silinder, apakah aus

GJ018 : Cek oli, apakah cc oli sesuai dengan ketentuan yang tertera di blok mesin.

GJ019 : Apakah kawat dalam lampu putus

GJ020 : Cek kabel reflector, apakah putus

GJ021 : Cek soket, apakah kabel dalam soket putus

GJ022 : Cek plat kopling, apakah aus

GJ023 : Cek otomatis sentripugal, apakah aus

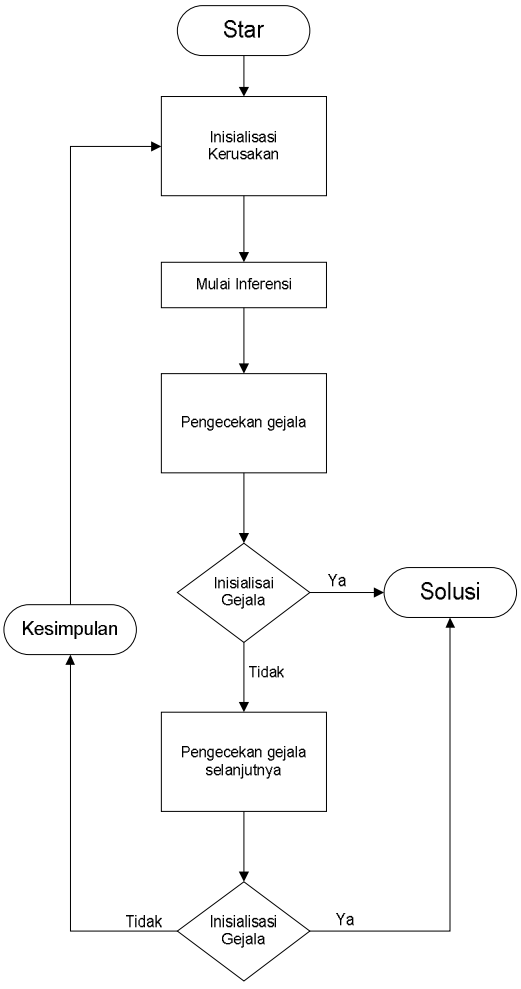
GJ024 : Cek bensin, apakah kosong

### 3.1.5 Metode Inferensi Backward Chaining

Runut balik atau Backward Chaining merupakan strategi pencarian yang arahnya kebalikan dari runut maju. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik jalur yang mengarah ke kesimpulan tersebut. Jika informasi-informasi atau nilai dari atribut-atribut yang mengarah ke kesimpulan tersebut sesuai dengan data yang diberikan maka kesimpulan tersebut merupakan solusi yang dicari, jika tidak sesuai maka kesimpulan tersebut bukan merupakan solusi yang dicari. Runut balik memulai proses pencarian dengan suatu tujuan sehingga strategi ini disebut juga *goal-driven.*

Algoritma runut balik di atas mempunyai 2 keadaan yang menyebabkan proses inferensi berhenti, yaitu :

1. Pada saat tabel Goal kosong, berarti kesimpulan yang merupakan solusi dari permasalahan sudah diperoleh.
2. Pada saat data dari pemakai yang diminta oleh sistem pakar untuk memenuhi prompt kaidah tidak ada, berarti kesimpulan yang merupakan solusi dari permasalahan tidak ditemukan.




**Gambar 3.1 Algoritma Backward Chaining**

### 3.1.6 Kaidah Produksi

Kaidah produksi biasanya dituliskan dalam bentuk jika maka (*IF-THEN*). Kaidah dapat dikatakan sebagai hubungan implikasi dua bagian yaitu premis (jika) dan bagian konklusi (maka). Apabila bagian premis dipenuhi maka bagian konklusi juga akan bernilai benar. Sebuah kaidah terdiri dari klausa-klausa sebuah klausa mirip sebuah kalimat subjek, kata kerja dan objek yang menyatakan suatu fakta. Ada sebuah klausa premis dan klausa konklusi pada sebuah kaidah. Suatu kaidah juga dapat terdiri dari beberapa premis dan lebih dari satu konklusi. Aturan premis dan konklusi dapat berhubungan dengan "OR" atau "AND". Berikut kaidah-kaidah produksi dalam mengidentifikasi kerusakan:

**Rule 1**

IF Mesin sukar hidup

AND Cek bensin, apakah kosong

THEN Isi bensin

Else IF Cek busi, apakah kotor

THEN Bersihkan dengan menggunakan amplas atau ganti busi

Else IF Cek karburator, apakah kotor

THEN Bersihkan semua saluran atau lubang, baik saluran bahan bakar maupun saluran angin dengan menggunakan bensin dan sikat

Else IF Cek selang dari tengki bahan bakar menuju karburator, apakah mampet atau longgar

THEN Bongkar saringan, pastikan tidak mampet dan tidak longgar

**Rule 2**

IF Mesin tidak dapat hidup sama sekali

AND Cek bensin, apakah kosong

THEN Isi bensin

Else IF Cek busi, apakah kotor

THEN Bersihkan dengan menggunakan amplas atau ganti busi

Else IF Cek platina atau coil dengan ohm meter, apakah rusak

THEN Ganti platina atau coil

Else IF Cek CDI untuk motor yang menggunakan coil, apakah rusak

THEN Ganti CDI

Else IF Cek kabel body menuju magnet dengan ohm meter, apakah putus

THEN Ganti kabel body yang putus

Esle IF Cek spul magnet, apakah putus

THEN Spul magnet diganti atau digulung

**Rule 3**

IF Mesin dapat hidup tapi tidak bertenaga

AND Cek busi, apakah kotor

THEN Bersihkan dengan menggunakan amplas atau ganti busi

Else IF Cek karburator, apakah kotor

THEN Bersihkan semua saluran atau lubang, baik saluran bahan bakar maupun saluran angin dengan menggunakan bensin dan sikat

Else IF Cek saringan udara, apakah kotor

THEN Bersihkan filter menggunakan bensin dan keringkan

Else IF Cek bos katup, apakah rusak

THEN Ganti bos katup

Else IF Cek pegas katup, apakah lemah

THEN Ganti pegas katup dan stel kembali pada posisi top

Else IF Cek kompresi, apakah bocor

THEN Ganti paking silinder, kencangkan baut silinder dan baut knalpot

**Rule 4**

IF Timbul suara menggelitik pada bagian blok silinder

AND Cek bos katup, apakah rusak

THEN Ganti bos katup

Else IF Cek pegas katup, apakah lemah

THEN Ganti pegas katup dan stel kembali pada posisi top

Else IF Cek noken as, apakah aus

THEN Ganti noken as

Else IF Cek torak dan ring torak, apakah aus

THEN Ganti torak dan ring torak

Else IF Cek stang torak dan laher stang, apakah longgar

THEN Ganti stang torak dan laher stang

**Rule 5**

IF Bahan bakar boros

AND Cek saringan udara, apakah kotor

THEN Bersihkan filter menggunakan bensin dan keringkan

Else IF Cek karburator, apakah kotor

THEN Bersihkan semua saluran atau lubang, baik saluran bahan bakar maupun saluran angin dengan menggunakan bensin dan sikat

Else IF Cek silinder head dan torak nya, apakah kotor

THEN Bersihkan karbon dari kedua bagian itu

Else IF Cek kompresi, apakah bocor

THEN Ganti paking silinder, kencangkan baut silinder dan baut knalpot

**Rule 6**

IF Keluar asap putih dari knalpot

AND Cek torak dan cincin torak, apakah aus

THEN Ganti torak dan cincin torak

Else IF Cek silinder, apakah aus

THEN Corter atau ganti lubang silinder

Else IF Cek oli, apakah cc oli sesuai dengan ketentuan yang tertera di blok mesin

THEN Sesuaikan cc oli

**Rule 7**

IF Mesin hidupnya tersendat-sendat setelah dipanaskan

AND Cek busi, apakah kotor

THEN Bersihkan dengan menggunakan amplas atau ganti busi

Else IF Cek platina atau coil dengan ohm meter, apakah rusak

THEN Ganti platina atau coil

Else IF Cek karburator, apakah kotor

THEN Bersihkan semua saluran atau lubang, baik saluran bahan bakar maupun saluran angin dengan menggunakan bensin dan sikat

**Rule 8**

IF Lampu besar mati

AND Apakah kawat dalam lampu putus

THEN Ganti lampu besar

Else IF Cek kabel reflector, apakah putus

THEN Ganti kabel reflector

Else IF Cek soket, apakah kabel dalam soket putus

THEN Ganti soket atau kiprok

**Rule 9**

IF Lampu sein mati atau tidak berkedip

AND Apakah kawat dalam lampu putus

THEN Ganti lampu sein

Else IF Cek soket, apakah kabel dalam soket putus

THEN Ganti soket flasher

**Rule 10**

IF Lampu rem mati

AND Apakah kawat dalam lampu putus

THEN Ganti lampu rem

Else IF Cek soket, apakah kabel dalam soket putus

THEN Ganti soket lampu rem atau switc rem

**Rule 11**

IF Transmisi susah dipindahkan

AND Cek plat kopling, apakah aus

THEN Ganti plat kopling

Else IF Cek otomatis sentripugal, apakah aus

THEN Ganti kanvas otomatis sentripugal

### 3.1.7 Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah usulan yang direkomendasikan kepada pengguna agar aplikasi yang akan dibangun menjadi *user friendly* dan perangkat kerasnya yang mendukung secara maksimal terhadap kinerja perangkat lunak.

### 3.1.7.1 Analisis Pengguna

Masyarakat umum adalah *user* para pengguna sepada motor yang dapat mencari informasi dan dapat berkonsultasi melalui proses diagnosa yang terdapat dalam aplikasi tersebut.

### 3.1.7.2 Analisis Perangkat Keras

Perangkat keras minimum server yang direkomendasikan untuk menjalankan aplikasi sistem pakar ini adalah sebagai berikut :

*Processor* : Dengan kecepatan 2.4 GHz

Kapasitas *Harddisk* : 120 GB

*RAM* : 256 MB

*VGA Card* : 128 MB

*Monitor, Mouse, Keyboard*

Perangkat keras minimum client yang direkomendasikan untuk menjalankan aplikasi sistem pakar ini adalah sebagai berikut :

*Processor* : Dengan kecepatan 2.0 GHz

Kapasitas *Harddisk* : 20 GB

*RAM* : 128 MB

*VGA Card* : 68 MB

*Monitor, Mouse, Keyboard*

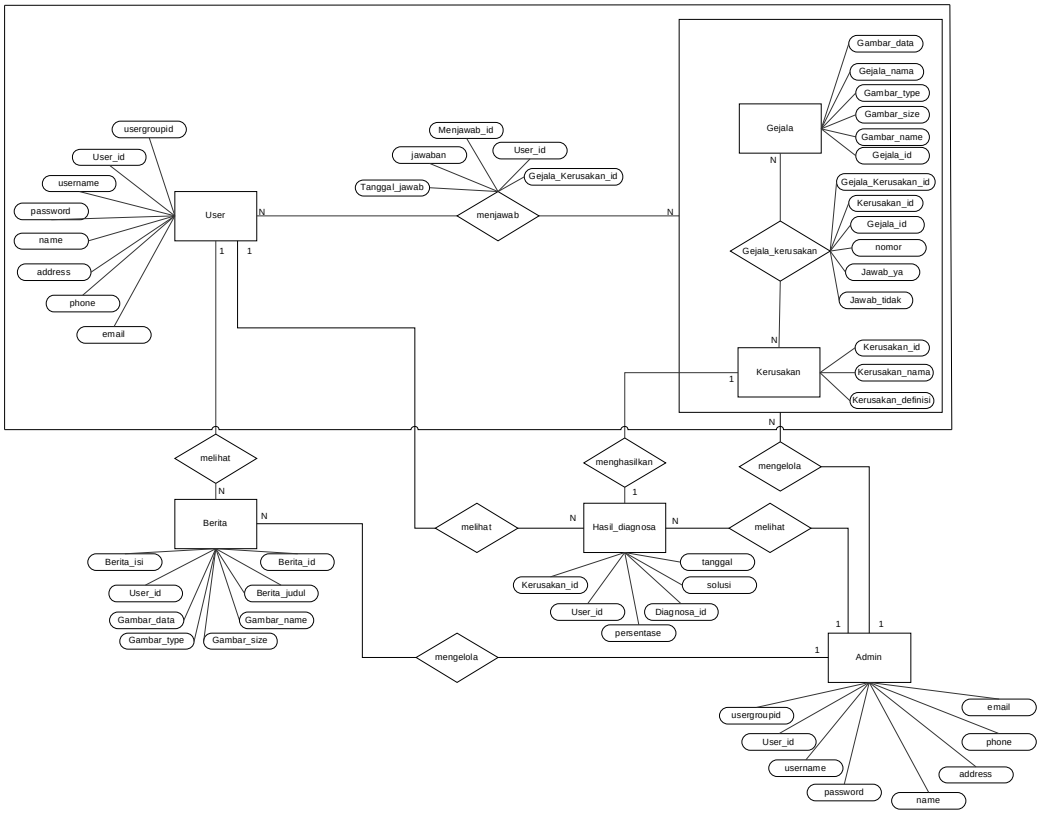
#### 3.1.7.3 Analisis Perangkat Lunak

Pemodelan Analisis Perangkat lunak pada server yang digunakan adalah sistem operasi *Microsoft windows XP Professional*, Bahasa Pemrogramannya menggunakan PHP dengan toolnya *Macromedia dreamweaver* 8, *web browser,* serta menggunakan *databasenya* yaitu MySQL.

Sedangkan untuk analisis perangkat lunak pada client hanya menggunakan *Web browser* seperti *Mozilla firefox, opera,* dan lain lain.

### 3.1.8 Analisis Basis Data

*Entity Relational Diagram* (ERD) merupakan cara untuk mengorganisasikan data, dimana diagram ini akan memperlihatkan hubungan entitas yang terdapat dalam sistem. ERD yang diusulkan untuk sistem yang akan dibangun dapat dilihat pada gambar 3.2 dibawah ini.



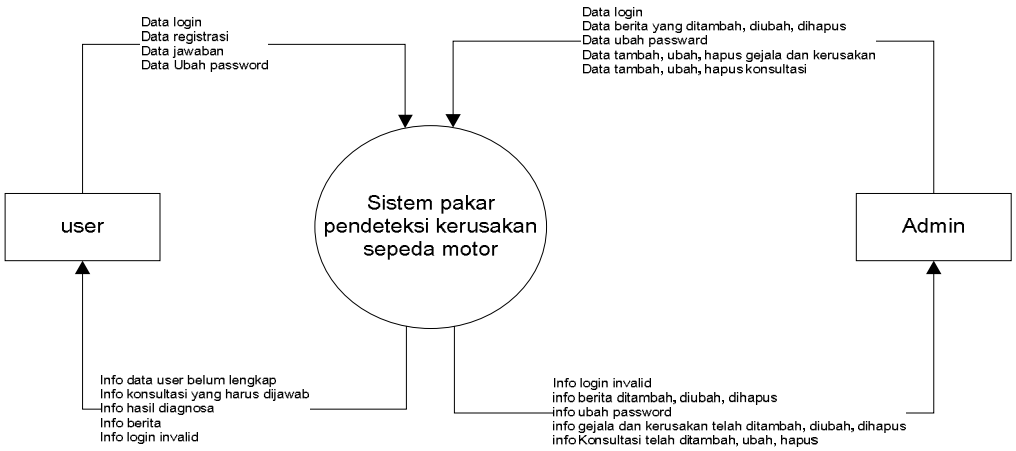

**Gambar 3.2 ERD Sistem Pakar Pendeteksi Kerusakan Sepeda Motor**

### 3.1.9 Analisis Kebutuhan Fungsional

Dalam langkah ini dilakukan penentuan entitas-entitas, data-data yang mengalir serta prosedur-prosedur yang bisa dilakukan oleh masing-masing entitas.

#### 3.1.9.1 Diagram Konteks

Diagram konteks adalah alur data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar. Adapun diagram konteks untuk sistem yang akan dibangun dapat dilihat pada gambar 3.3

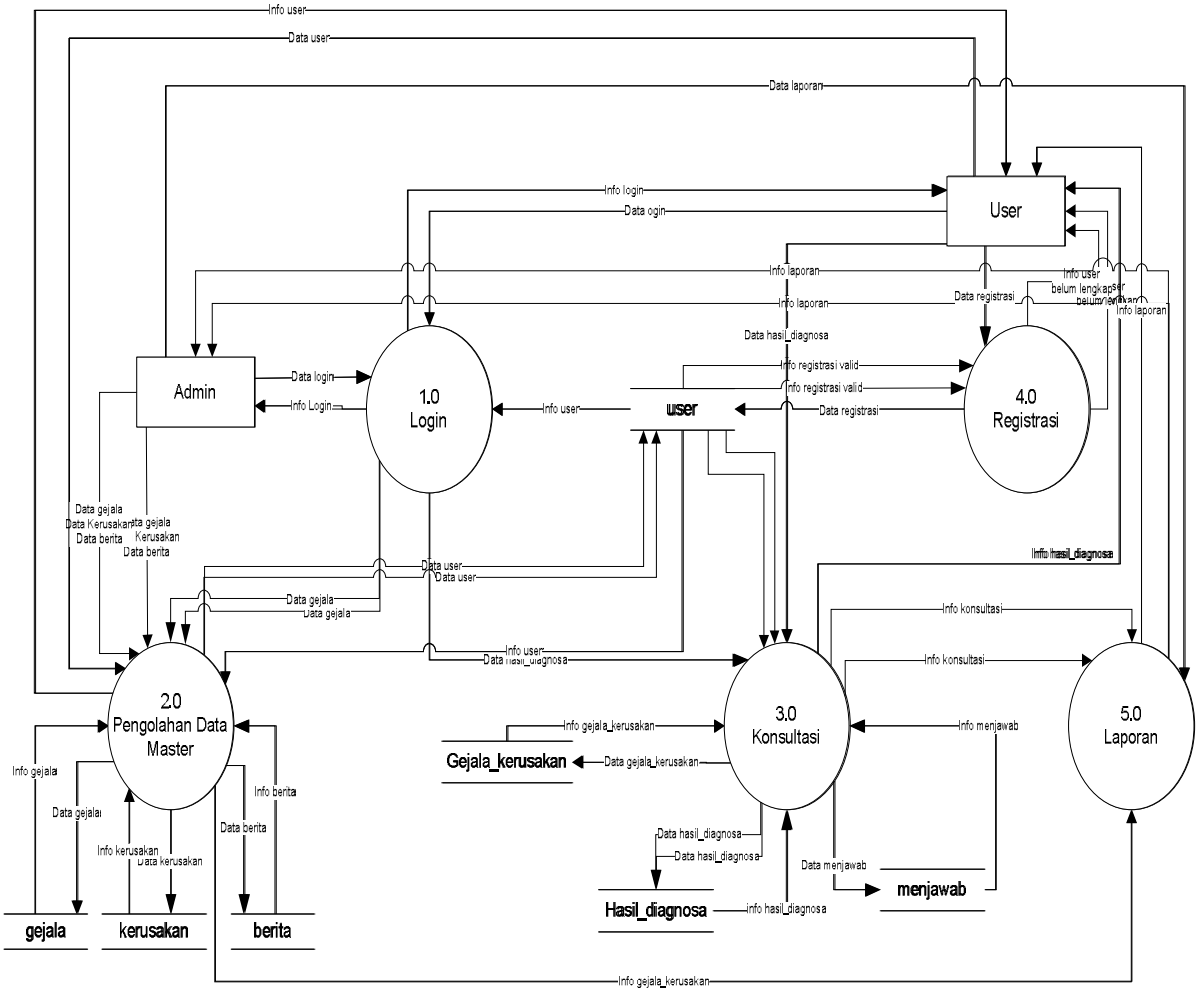



**Gambar 3.3 Diagram Konteks Sistem Pakar Pendeteksi Kerusakan Sepeda Motor**

### 3.1.9.2 Data Flow Diagram (DFD)

DFD adalah sebuah teknik yang menggambarkan aliran data dan transformasi yang digunakan sebagai alur data dari masukan menuju keluaran.

### 3.1.9.2.1 DFD Level 1

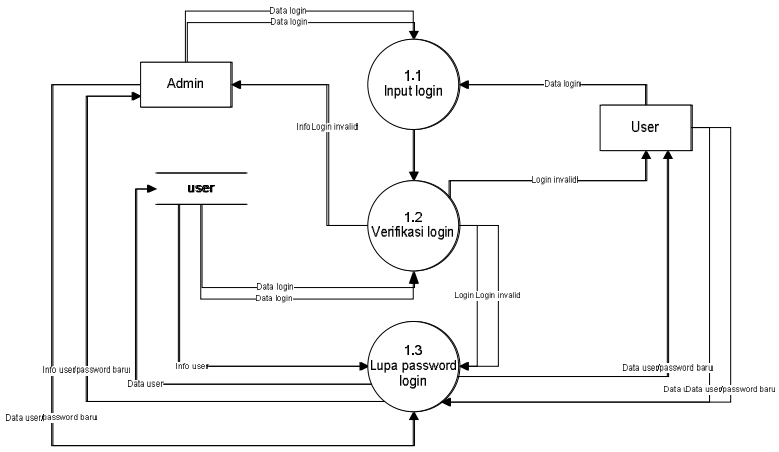



**Gambar 3.4 DFD Level 1 Pendeteksi Kerusakan Sepeda Motor**

### 3.1.9.2.2 DFD Level 2

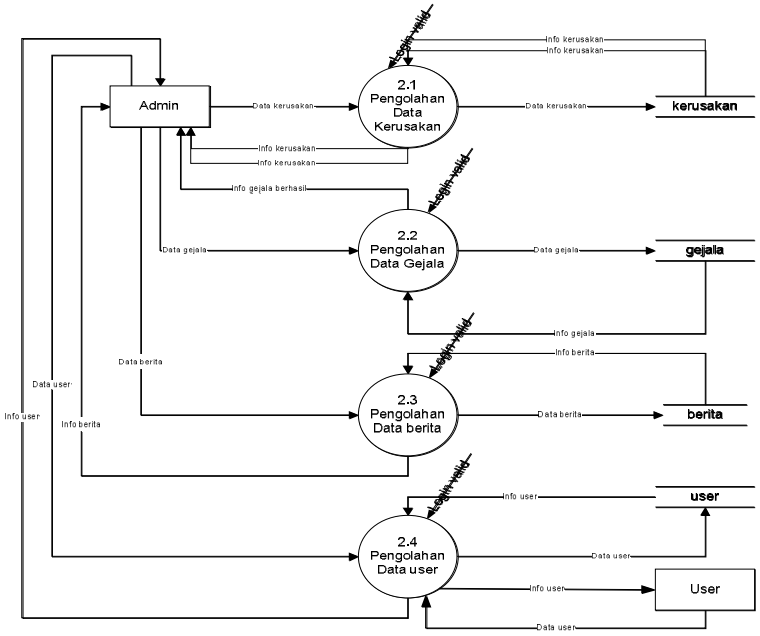
DFD Level 2 mengambarkan tiap-tiap proses pada level 1 yang lebih rinci.

**DFD Level 2 Untuk Proses 1.0**



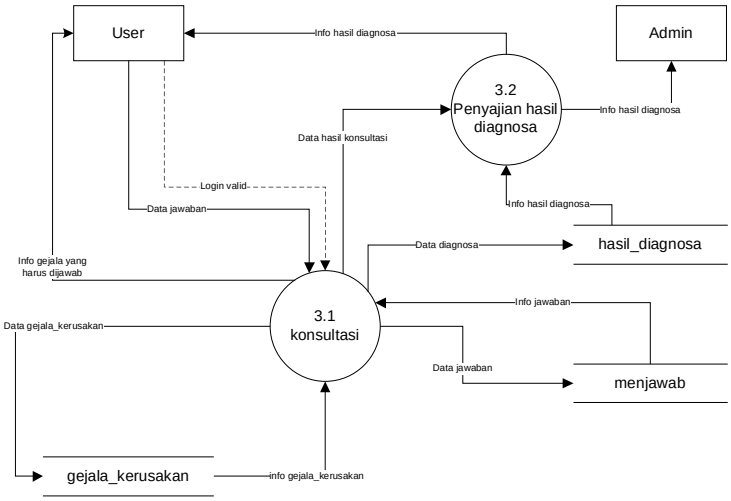

**Gambar 3.5 DFD Level 2 Proses 1.0 proses Login**

**DFD Level 2 Untuk Proses 2.0 pengolahan data master**



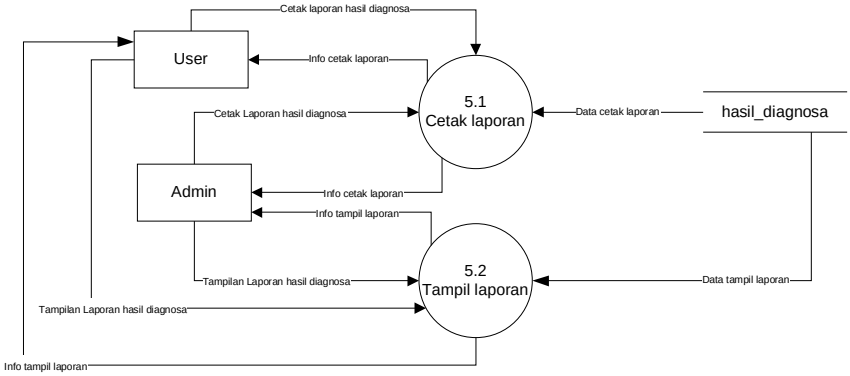

**Gambar 3.6 DFD Level 2 Proses 2.0 Pengolahan data master**

**DFD level 2 untuk proses 3.0 konsultasi**




**Gambar 3.7 DFD Level 2 Proses 3.0 Konsultasi**

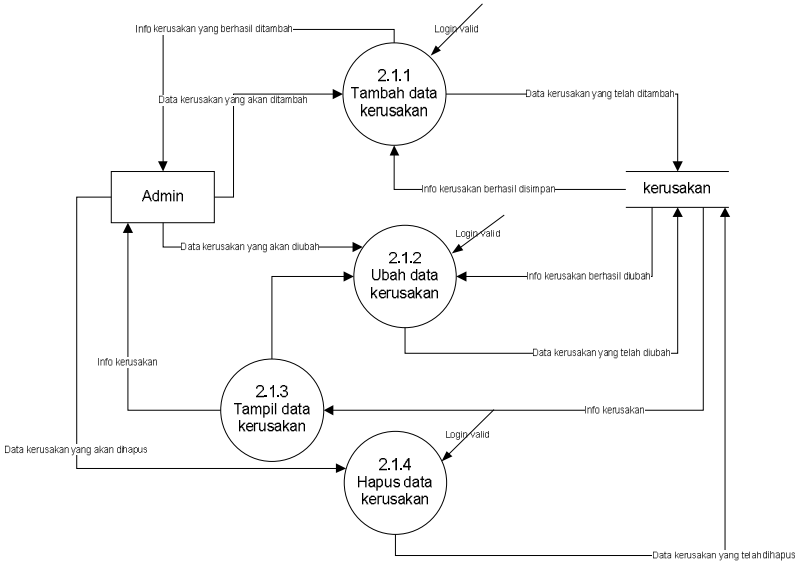
**DFD level 2 untuk proses 5.0**




**Gambar 3.8 DFD Level 2 Proses 5.0 Laporan**

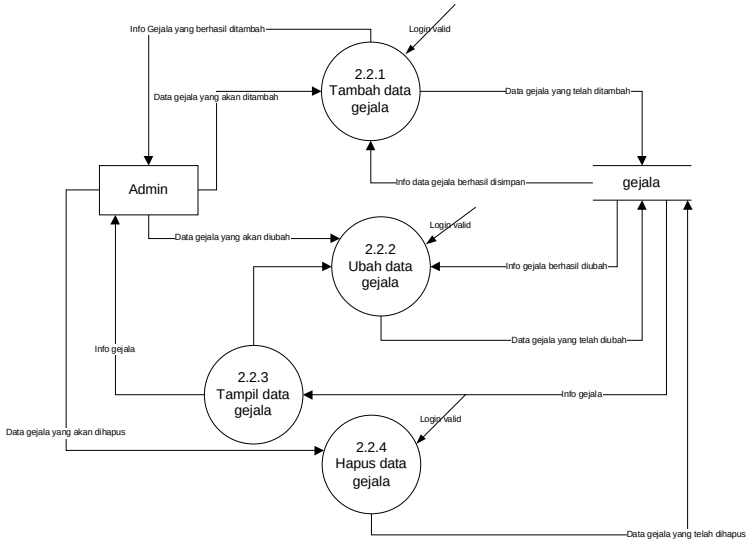
### 3.1.9.2.3 DFD Level 3

### DFD Level 3 untuk proses 2.1



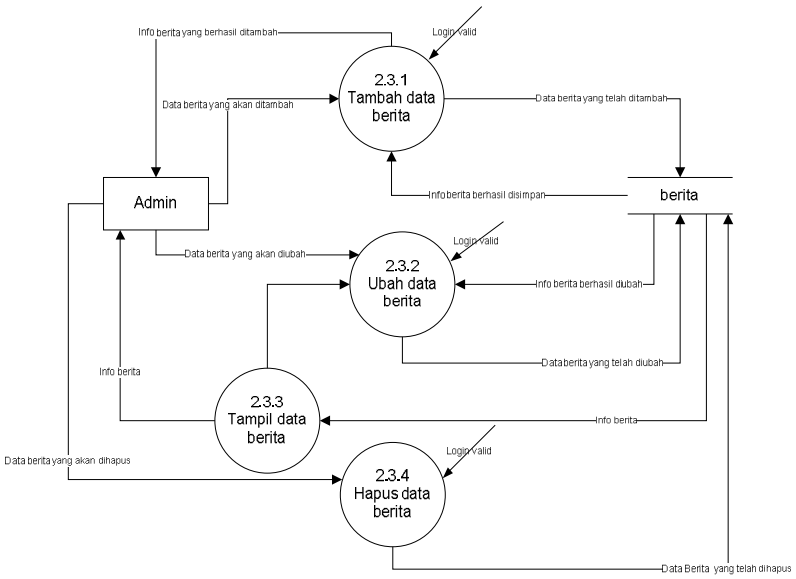

**Gambar 3.9 DFD Level 3 Proses 2.1 Pengolahan data kerusakan**

### DFD level 3 untuk Proses 2.2



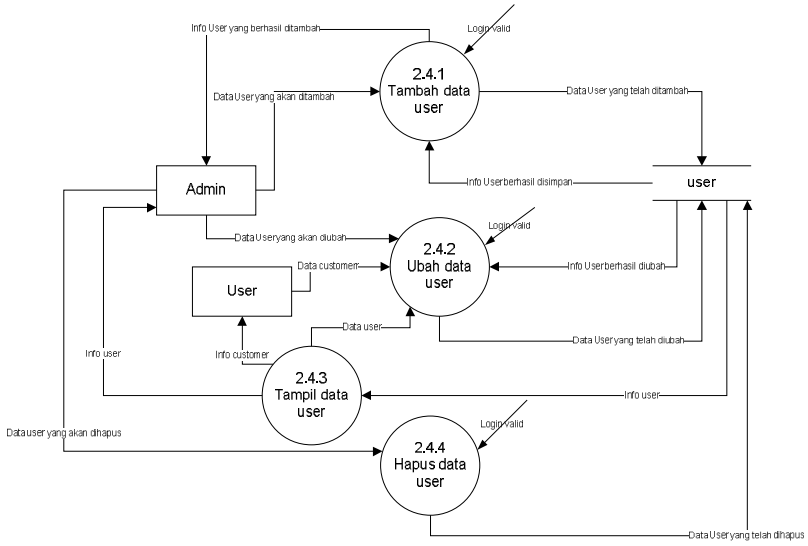

**Gambar 3.10 DFD Level 3 Proses 2.2 Pengolahan data gejala**

**DFD level 3 untuk Proses 2.3**




**Gambar 3.11 DFD level 3 proses 2.3 Pengolahan data berita**

**DFD level 3 untuk Proses 2.4**




**Gambar 3.12 DFD level 3 proses 2.4 Pengolahan data user**

### 3.1.9.3 Spesifikasi Proses

Untuk menjelaskan lebih lanjut tentang proses-proses yang ada di dalam diagram alir data atau DFD maka dibuatlah spesifikasi proses. Adapun penjelasan aplikasi prosesnya pada tabel 3.2 dibawah ini:

**Tabel 3.2 Spesifikasi Proses**

| No | Proses | Keterangan |
|---|---|---|
| 1 | No Proses | 1.0 |
| | Nama Proses | Login |
| | *Source* (sumber) | *Admin, user* |
| | Input | Data Login |
| | Output | Info Login Invalid |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | Begin<br>{admin}<br>If username And password ada<br>Then masuk ke form menu admin<br>Else tampil info login admin invalid<br>End<br>Begin<br>{user}<br>If username And password ada<br>Then masuk ke form menu user<br>Else tampil info login user invalid<br>End |
| 2 | No Proses | 2.0 |
| | Nama Proses | Pengolahan Data master |
| | *Source* (sumber) | *Admin* |
| | Input | Data kerusakan, data gejala, data berita, data user |
| | Output | info kerusakan, gejala, berita, user |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | Begin<br>{Admin melakukan proses tambah data kerusakan, gejala, berita, dan user. Proses Ubah data kerusakan, gejala, berita, dan user. Proses hapus data kerusakan, gejala, berita, user}<br>End |
| | No Proses | 3.0 |
| | Nama Proses | konsultasi |
| | *Source* (sumber) | user |

| | | |
|---|---|---|
| 3 | Input | Data hasil diagnosa, Data menjawab, Data gejala_ kerusakan |
| | Output | Info hasil diagnosa, gejala, menjawab, gejala_ kerusakan |
| | *Destination* (tujuan) | user |
| | Logika Proses | Begin<br>if data kerusakan<br>Then data gejala<br>Else keluar pesan pilih “data kerusakan”<br>End |
| 4 | No Proses | 4.0 |
| | Nama Proses | Registrasi |
| | *Source* (sumber) | user |
| | Input | Data user |
| | Output | Info user |
| | *Destination* (tujuan) | user |
| | Logika Proses | Begin<br>If data user<br>Then data user<br>Else keluar pesan “data belum lengkap”<br>End |
| 5 | No Proses | 5.0 |
| | Nama Proses | Laporan |
| | *Source* (sumber) | *Admin, user* |
| | Input | - |
| | Output | Info cetak dan tampil laporan hasil diagnosa |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | Begin<br>{ *Admin, user* melihat laporan hasil diagnosa}<br>End |
| 6 | No Proses | 2.1 |
| | Nama Proses | Pengolahan data kerusakan |
| | *Source* (sumber) | *Admin* |
| | Input | Data kerusakan |
| | Output | Info kerusakan |
| | *Destination* (tujuan) | *Admin* |
| | Logika Proses | Begin<br>{Admin mengolah data kerusakan}<br>End |
| 7 | No Proses | 2.2 |
| | Nama Proses | Pengolahan data gejala |
| | *Source* (sumber) | *Admin* |
| | Input | Data gejala |
| | Output | Info gejala |
| | *Destination* (tujuan) | *Admin* |

| | | |
|---|---|---|
| | Logika Proses | begin<br>{admin mengolah data gejala}<br>end |
| 8 | No Proses | 2.3 |
| | Nama Proses | Pengolahan data berita |
| | *Source* (sumber) | *Admin* |
| | Input | Data berita |
| | Output | Info berita |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | begin<br>{admin mengolah data berita}<br>end |
| 9 | No Proses | 2.4 |
| | Nama Proses | Pengolahan data user |
| | *Source* (sumber) | *Admin* |
| | Input | Data user |
| | Output | Info Data user |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | Begin<br>{Admin mengolah data user }<br>End |
| 10 | No Proses | 3.1 |
| | Nama Proses | Konsultasi |
| | *Source* (sumber) | *user* |
| | Input | Data hasil diagnosa, data gejala_kerusakan, data menjawab |
| | Output | Info hasil diagnosa, gejala_kerusakan, menjawab |
| | *Destination* (tujuan) | *user* |
| | Logika Proses | - |
| 11 | No Proses | 3.2 |
| | Nama Proses | Penyajian hasil diagnosa |
| | *Source* (sumber) | *user* |
| | Input | Data hasil diagnosa |
| | Output | Info hasil diagnosa |
| | *Destination* (tujuan) | *user* |
| | Logika Proses | - |
| 12 | No Proses | 5.1 |
| | Nama Proses | Cetak laporan |
| | *Source* (sumber) | *Admin, user* |
| | Input | - |
| | Output | Info cetak laporan hasil diagnosa |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | - |
| | No Proses | 5.2 |
| | Nama Proses | Tampil laporan |

| | | |
|---|---|---|
| 13 | *Source* (sumber) | *Admin, user* |
| | Input | - |
| | Output | Info tampil laporan hasil diagnosa |
| | *Destination* (tujuan) | *Admin, user* |
| | Logika Proses | - |

### 3.1.9.4 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Perancangan kamus data pada aplikasi sistem pakar untuk mendeteksi kerusakan pada sepeda motor berbasis web dapat dilihat melalui tabel-tabel berikut ini.

**Tabel 3.3 Kamus Data**

| **Nama** | **Data Login** |
|---|---|
| *Where used / how used* | Proses 1.0 Login (*input*)<br>Proses 2.1 pengolahan data user |
| Deskripsi | Berisi data user yang akan digunakan pada pengolahan data user |
| Struktur Data | *User_id+Name+type* |
| *User_id*<br><br>*Name*<br><br>*Type* | [a..z \| A..Z \| 0..9] {10}<br><br>[a..z \| A..Z \| 0..9] {50}<br><br>[a..z \| A..Z \| 0..9] {10} |
| **Nama** | **Data Kerusakan** |
| *Where used / how used* | Proses 2.1 pengolahan data kerusakan |
| Deskripsi | Berisi data kerusakan yang akan digunakan pada proses konsultasi. |
| Struktur Data | *kerusakan_id + kerusakan_Nama + kerusakan_definisi* |
| *Kerusakan_id*<br><br>*Kerusakan_nama*<br><br>*Kerusakan_definisi* | [a..z \| A..Z \| 0..9] {10}<br><br>[a..z \| A..Z \| 0..9] {50}<br><br>[‘text’] |

| **Nama** | **Data gejala** |
|---|---|
| *Where used / how used* | Proses 2.2 pengolahan data gejala |
| Deskripsi | Berisi data gejala yang digunakan untuk proses konsultasi |
| Struktur Data | Gejala_id + gejala_nama |
| Gejala_id<br><br>Gejala_nama | [a..z \| A..Z \| 0..9] {10)<br><br>[a..z \| A..Z \| 0..9] {50} |
| **Nama** | **Data berita** |
| *Where used / how used* | Proses 2.3 data berita |
| *Deskripsi* | Berisi data berita digunakan pada pengolahan data berita |
| *Struktur Data* | berita_id + berita_judul + berita_isi |
| berita_id<br><br>berita_judul<br><br>berita_isi | [a..z \| A..Z \| 0..9] {10}<br><br>[a..z \| A..Z \| 0..9] {50}<br><br>[‘tekt’] |
| **Nama** | **Data registrasi** |
| *Where used / how used* | Proses 2.4 pengolahan data user |
| Deskripsi | Berisi data registrasi yang dimasukan oleh user |
| Struktur Data | user_id + username + password |
| user_id<br><br>username<br><br>password | [a..z \| A..Z \| 0..9] {5}<br><br>[a..z \| A..Z \| 0..9] {30}<br><br>[a..z \| A..Z \| 0..9] {30} |
| **Nama** | **Data Laporan** |
| *Where used / how used* | Proses 3.0 konsultasi |
| Deskripsi | Berisi data hasil diagnosa yang akan ditampilkan dan di cetak oleh admin atau user |
| Struktur Data | Diagnosa_id + kerusakan_id + user_id + tanggal persentase + solusi |
| diagnosa_id<br><br>kerusakan_id | [a..z \| A..Z \| 0..9] {10}<br><br>[a..z \| A..Z \| 0..9] {10} |

| | |
|---|---|
| user_id | [a..z \| A..Z \| 0..9] {10} |
| tanggal | datetime |
| solusi | [‘tekt’] |

## 3.2 Perancangan Sistem

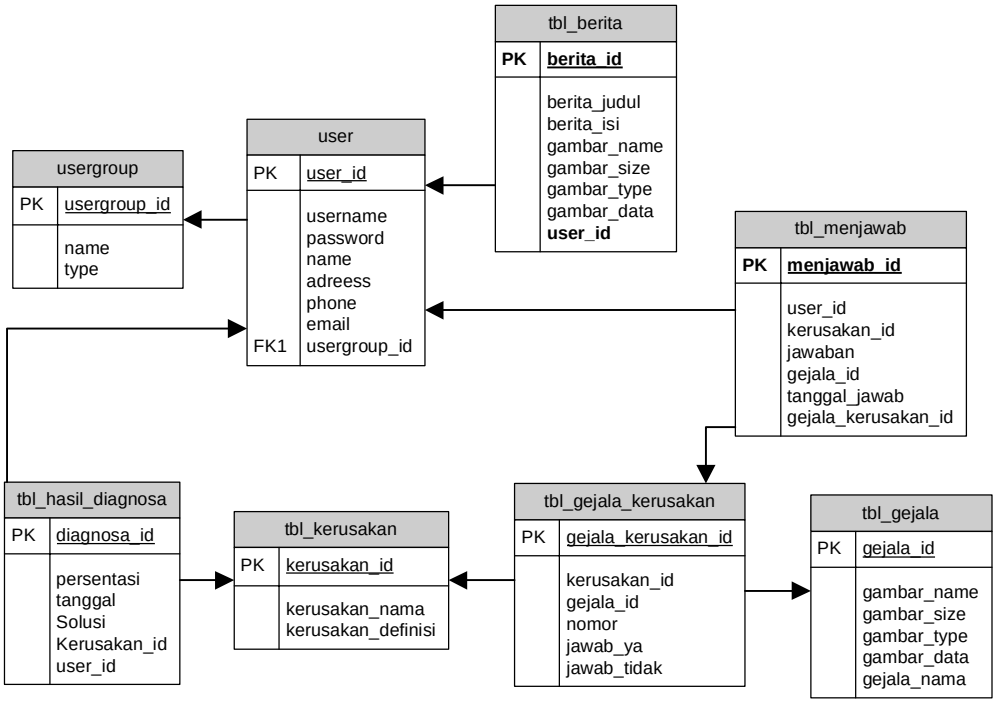
Perancangan sistem pakar untuk mendeteksi kerusakan pada sepeda motor bertujuan untuk menerapkan solusi pemecahan masalah yang telah diajukan pada analisis sistem.

### 3.2.1 Perancangan Data

Perancangan data terdiri dari Tabel Relasi dan struktur tabel yang terdapat pada ERD dari aplikasi sistem pakar pendeteksi kerusakan pada motor.

#### 3.2.1.1 Tabel Relasi

Tabel Relasi menggambarkan suatu hubungan antar tabel yang sudah ada dalam keadaan normal. Perancangan tabel relasi dalam membangun sebuah Sistem Pakar untuk Mendeteksi Jenis Kerusakan Pada Sepeda Motor Berbasis Web adalah sebagai berikut :




**Gambar 3.13 Tabel Relasi**

### 3.2.1.2 Struktur Tabel

Tabel merupakan tempat penyimpanan informasi dari sebuah aliran data dalam sebuah sistem. Berikut merupakan struktur dari beberapa tabel sistem yang akan dibangun.

**1. Tabel user**

Tabel ini berfungsi untuk menyimpan data user yang telah melakukan konsultasi pada aplikasi untuk mendiagnosa jenis kerusakan pada sepeda motor.

**Tabel 3.4 Tabel user**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | User_id | int | 10 | *Primary key* |
| 2 | Username | *Int* | 10 | - |
| 3 | name | *Varchar* | 30 | - |
| 4 | password | *smallint* | 5 | - |
| 5 | address | *text* | - | - |
| 6 | phone | *Varchar* | 30 | - |
| 7 | Email | *Varchar* | 50 | - |
| 8 | Usergroup_id | *int* | 10 | *Foreign Key* |

**2. Tabel tbl_menjawab**

Tabel ini berfungsi untuk menghubungkan antara tabel user dan pertanyaan pada proses diagnosa.

**Tabel 3.5 Tabel tbl_menjawab**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | kerusakan_id | *Int* | 10 | *Foreign key* |
| 2 | User_id | *int* | 10 | *Foreign key* |
| 3 | Gejala_id | *smallint* | 10 | *Foreign key* |
| 4 | Jawaban | *text* | - | - |
| 5 | Tanggal_jawab | *varchar* | 30 | - |
| 6 | Gejala_kerusakan_id | *int* | 10 | - |

**3. Tabel tbl_gejala**

Tabel ini berfungsi untuk menyimpan data gejala pada kerusakan sepeda motor.

**Tabel 3.6 Tabel tbl_gejala**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Gejala_id | *int* | 10 | *Primary Key* |
| 2 | Gejala_nama | *text* | - | - |
| 3 | Gambar_name | *varchar* | 100 | - |
| 4 | Gambar_size | *varchar* | 100 | - |
| 5 | Gambar_type | *varchar* | 100 | - |
| 6 | Gambar_data | *longblob* | - | - |

**4. Tabel tbl_gejala_kerusakan**

Tabel ini berfungsi untuk menghubungkan tabel gejala dan tabel kerusakan berupa aturan atau rule pada proses diagnosa.

**Tabel 3.7 Tabel tbl_gejala_kerusakan**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Gejala_kerusakan_id | *int* | 10 | *Primary key* |
| 1 | kerusakan_id | *int* | 10 | *Foreign key* |
| 2 | Gejala_id | *int* | 4 | *Foreign key* |
| 3 | Jawab_ya | *Varchar* | 50 | - |
| 4 | Jawab_tidak | *Varchar* | 50 | - |
| 5 | nomor | *smallint* | 5 | - |

**5. Tabel tbl_kerusakan**

Tabel ini berfungsi untuk menyimpan data kerusakan sepeda motor.

**Tabel 3.8 Tabel tbl_kerusakan**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Kerusakan_id | *int* | 10 | *Primary key* |
| 2 | Kerusakan _nama | *Varchar* | 30 | - |
| 3 | Kerusakan _definisi | *Text* | - | - |

**6. Tabel tbl_hasil_diagnosa**

Tabel ini berfungsi untuk menghasilkan data hasil diagnosa dari proses diagnosa jenis kerusakan sepada motor.

**Tabel 3.9 tbl_hasil_diagnosa**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Diagnosa_id | *Int* | 10 | *Primary Key* |
| 2 | User_id | *int* | 10 | *Foreign Key* |
| 3 | kerusakan_id | *int* | 10 | *Foreign Key* |
| 4 | Tanggal | *datetime* | - | - |
| 5 | persentase | *Int* | 10 | - |
| 6 | Solusi | *Varchar* | 100 | - |

**7. Tabel tbl_berita**

Tabel ini berfungsi untuk menyimpan data berita yang berkaitan dengan kerusakan atau perawatan sepeda motor.

**Tabel 3.10 Tabel tbl_berita**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Berita_id | *int* | 10 | *Primary Key* |
| 2 | User_id | *Int* | 10 | *Foreign key* |
| 3 | Berita_Judul | *text* | - | - |
| 4 | Berita_Isi | *text* | - | - |
| 5 | Gambar_name | *varchar* | 100 | - |
| 6 | Gambar_size | *varchar* | 100 | - |
| 7 | Gambar_type | *varchar* | 100 | - |
| 8 | Gambar_data | *longblob* | - | - |

**8. Tabel usergroup**

Tabel ini berfungsi untuk menyajikan pilihan login antara administrator dan user.

**Tabel 3.11 Tabel usergroup**

| No | *Field* | *Type* | *Size* | Keterangan |
|---|---|---|---|---|
| 1 | Usergroup_id | *int* | 10 | *Primary Key* |
| 2 | type | *Int* | 10 | *Foreign key* |
| 3 | Name | *varchar* | 50 | - |

**3.2.1.3 Pengkodean**

Kode digunakan untuk mengklasifikasi data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang saling berhubungan. Perancangan pengkodean yang diusulkan dengan tujuan untuk mempermudah dalam proses pengolahan data. Rancangan kode yang diusulkan adalah:

1. Pengkodean kode kerusakan terdiri dari 5 digit, yaitu dengan format sebagai berikut:

   XY 999

   XY : menunjukkan kode jenis kerusakan

   999 : menunjukkan nomor urut

   Contoh : KS001

KS menunjukkan kode jenis kerusakan, 001 menunjukkan nomor urut kerusakan.

2. Pengkodean kode gejala terdiri dari 5 digit, yaitu dengan format sebagai berikut:

   XY 999

   XY : menunjukkan kode gejala

   999 : menunjukkan nomor urut

   Contoh : GJ001

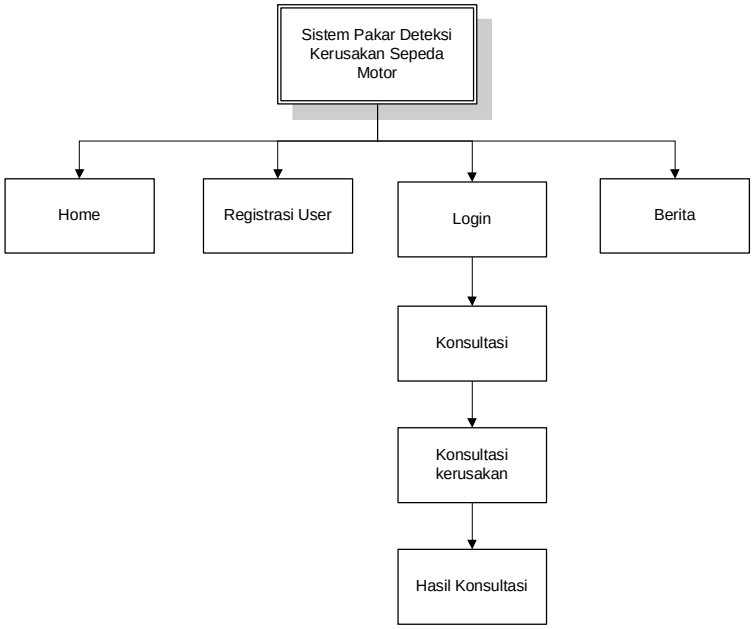
   GJ menunjukkan kode gejala, 001 menunjukkan nomor urut gejala.

### 3.2.2 Perancangan Struktur Menu

Perancangan struktur menu berisikan menu dan submenu yang berfungsi memudahkan user didalam menggunakan sistem. Menu-menu tersebut dibagi berdasarkan hak akses masing-masing user. Berikut ini gambaran mengenai struktur menu Aplikasi Sistem Pakar Untuk Mendeteksi Kerusakan Sepeda Motor Berbasis Web.

**1. Struktur Menu User**

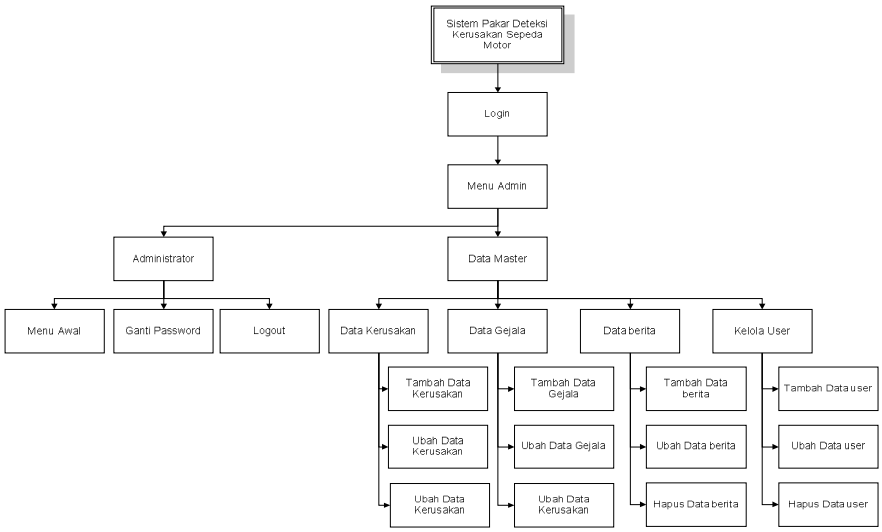
Struktur menu user ditujukan bagi user yang akan melakukan konsultasi. User masuk ke menu konsultasi untuk melakukan daftar user kemudian dapat melakukan proses konsultasi. User dapat mengakses menu home, konsultasi, dan berita.




**Gambar 3.14 Struktur Menu User**

## 2. Struktur Menu Admin

Stuktur menu admin terdiri dari proses login, menu admin, kemudian dapat melakukan proses admin didalamnya ada fungsi ganti password dan keluar. Serta management data berupa data master. Dalam data master dapat melakukan proses tambah, ubah, hapus, dan Logout.

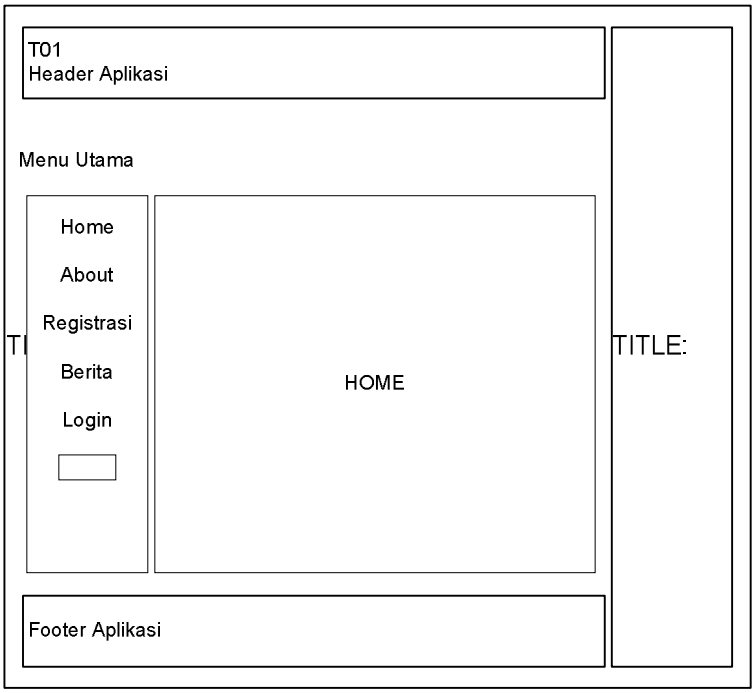



**Gambar 3.15 Struktur Menu Admin**

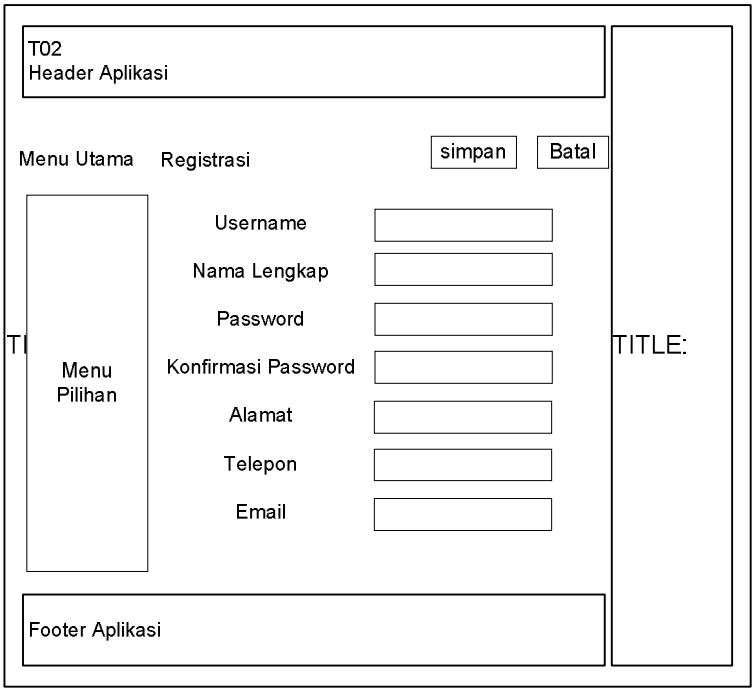
### 3.2.3 Perancangan Antar Muka

Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun. Sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi yang user friendly.

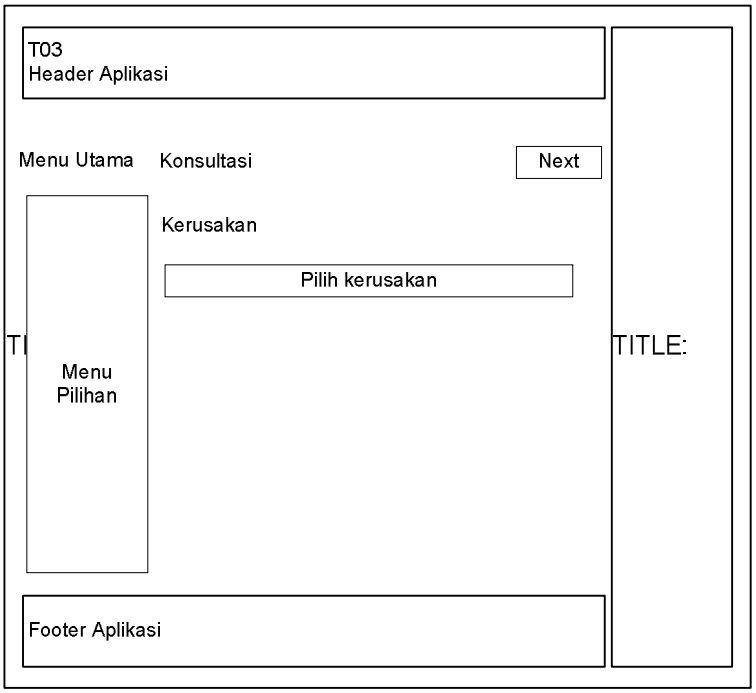
Rancangan aplikasi untuk mendeteksi jenis kerusakan pada sepeda motor yang akan dibuat sebagai berikut :



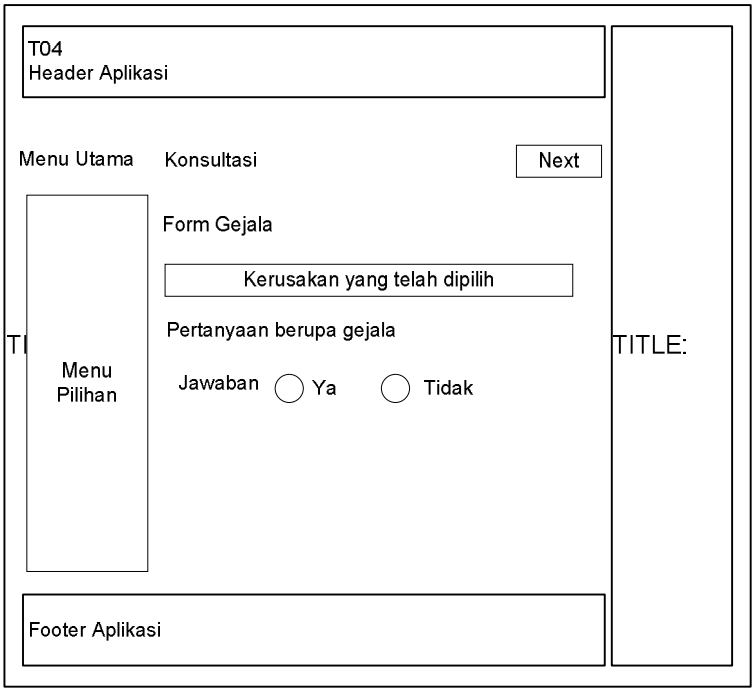

**Gambar 3.16 Tampilan Menu Utama**



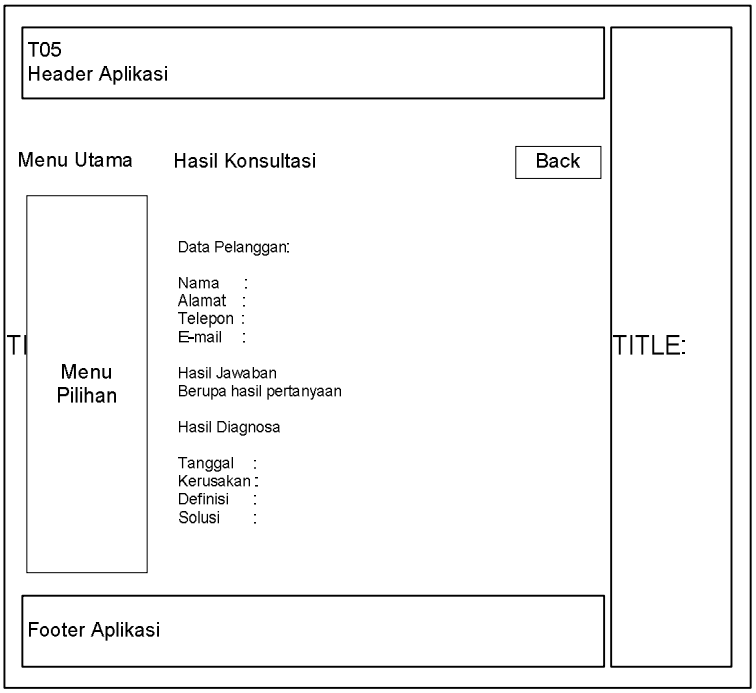

**Gambar 3.17 Tampilan Registrasi User**



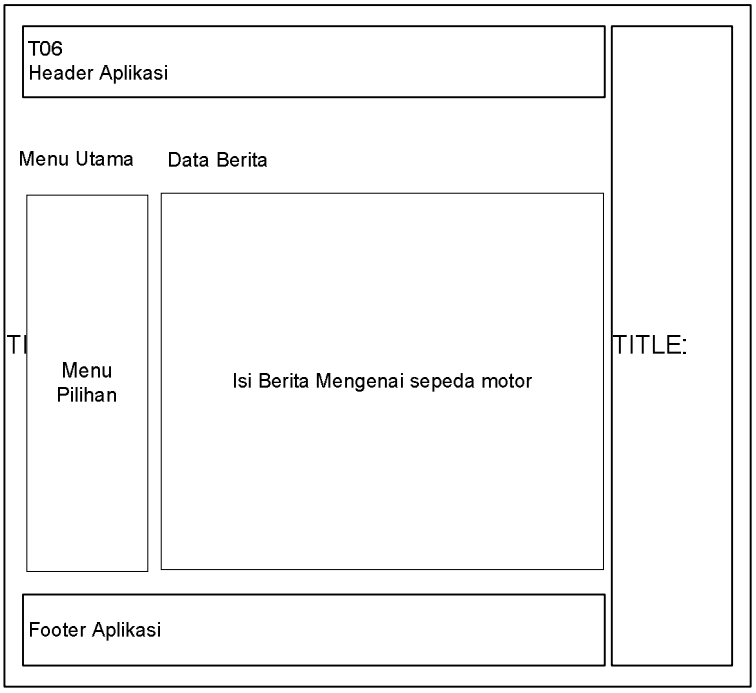

**Gambar 3.18 Tampilan Menu Konsultasi (pilih kerusakan)**



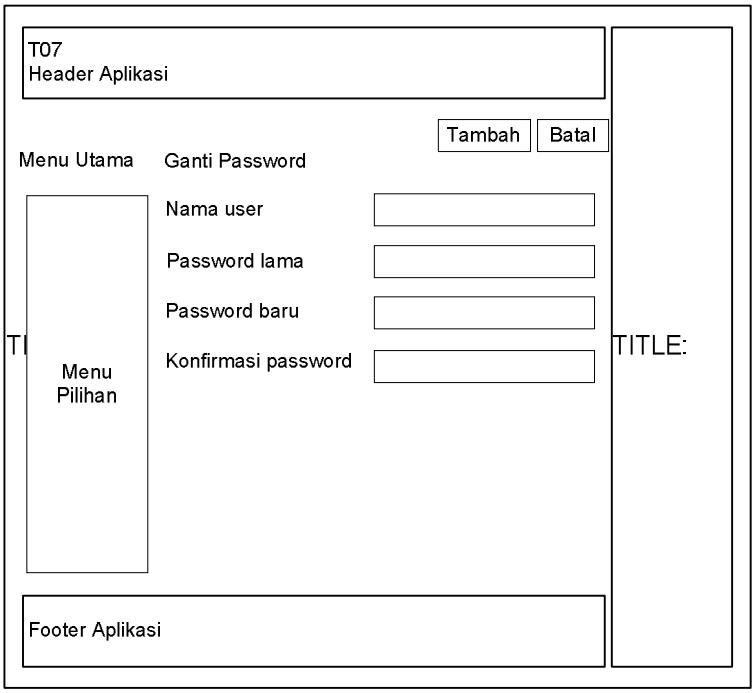

**Gambar 3.19 Tampilan Menu Konsultasi (jawab pertanyaan)**



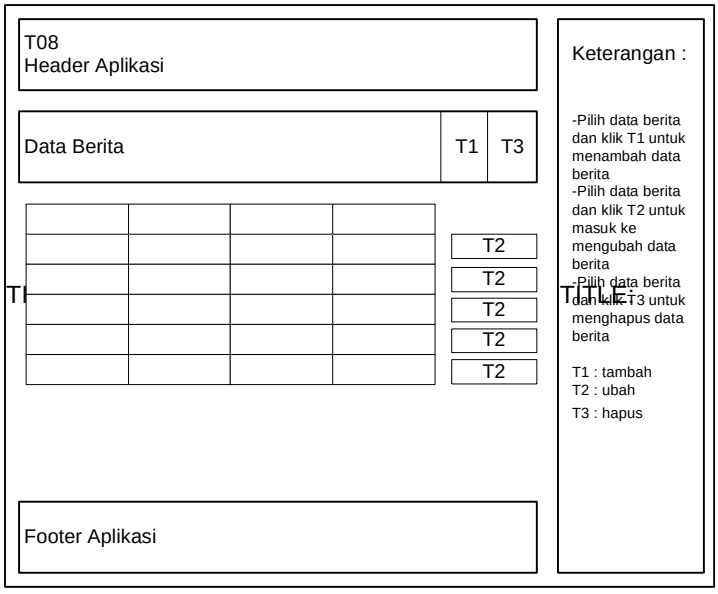

**Gambar 3.20 Tampilan Menu Hasil Konsultasi**



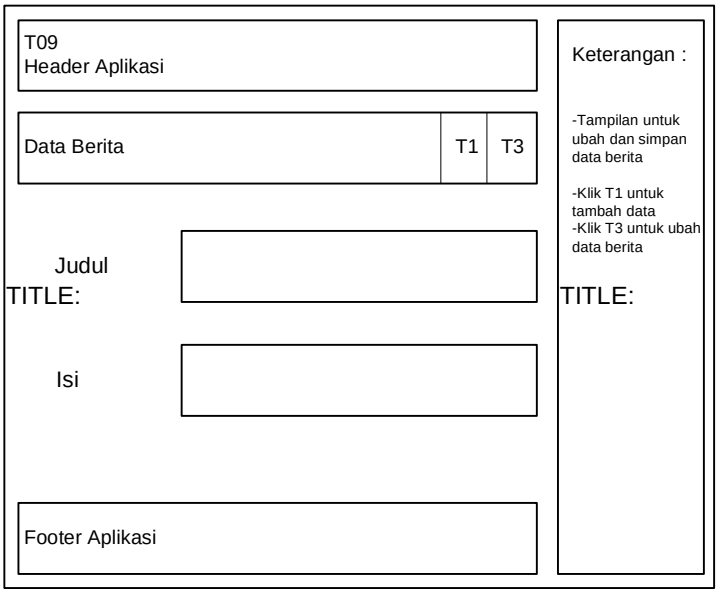

**Gambar 3.21 Tampilan Menu Berita**



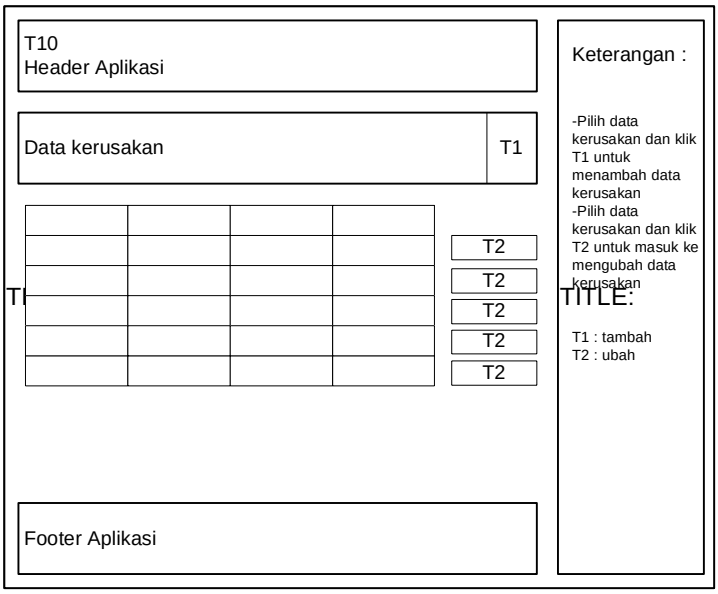

**Gambar 3.22 Tampilan Menu Ganti Password**



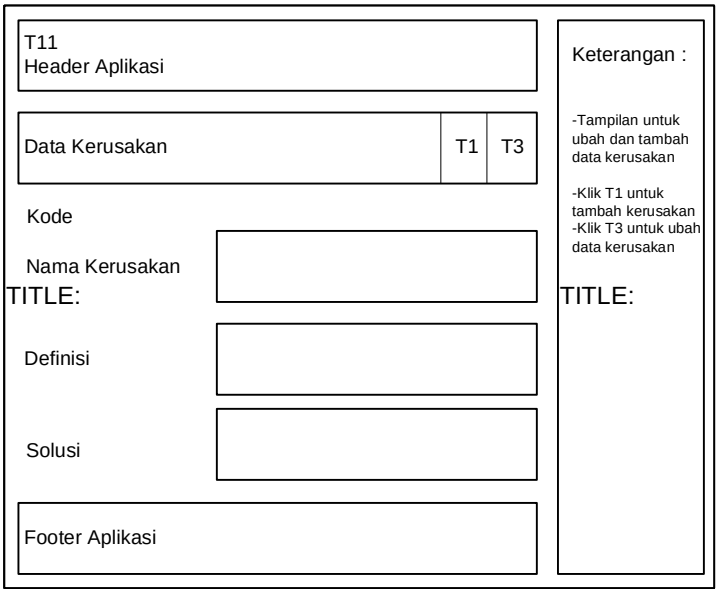

**Gambar 3.23 Tampilan Menu Data Berita**



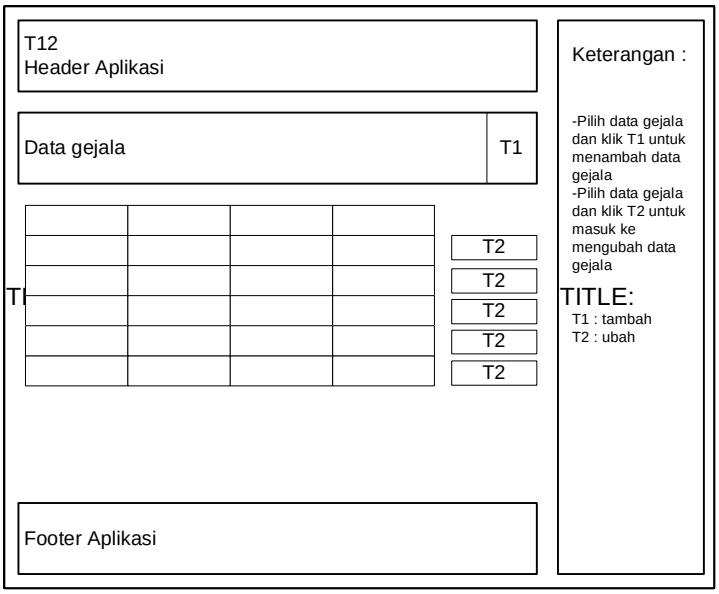

**Gambar 3.24 Tampilan Menu Tambah, Hapus dan Ubah Berita**



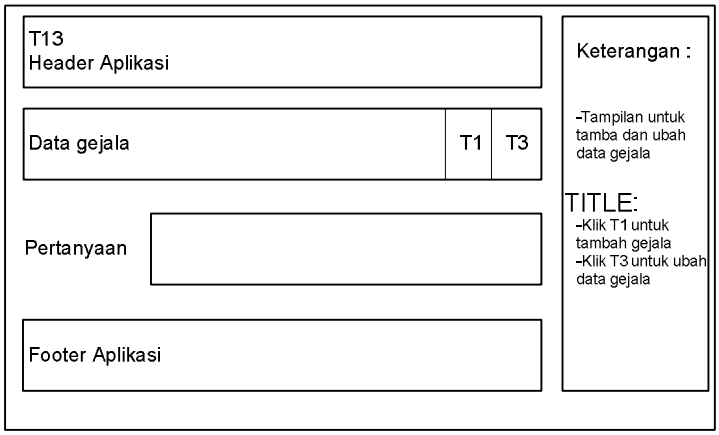

**Gambar 3.25 Tampilan Menu Data Kerusakan**



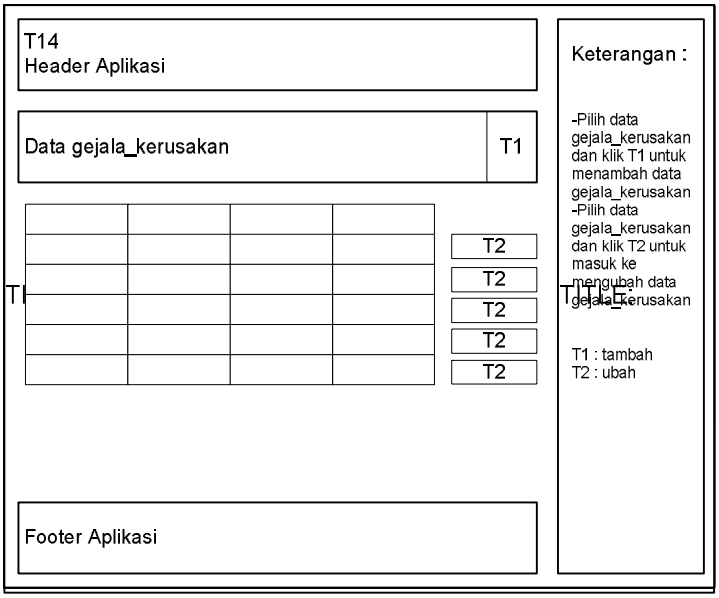

**Gambar 3.26 Tampilan Menu Tambah, Hapus dan Ubah Data Kerusakan**



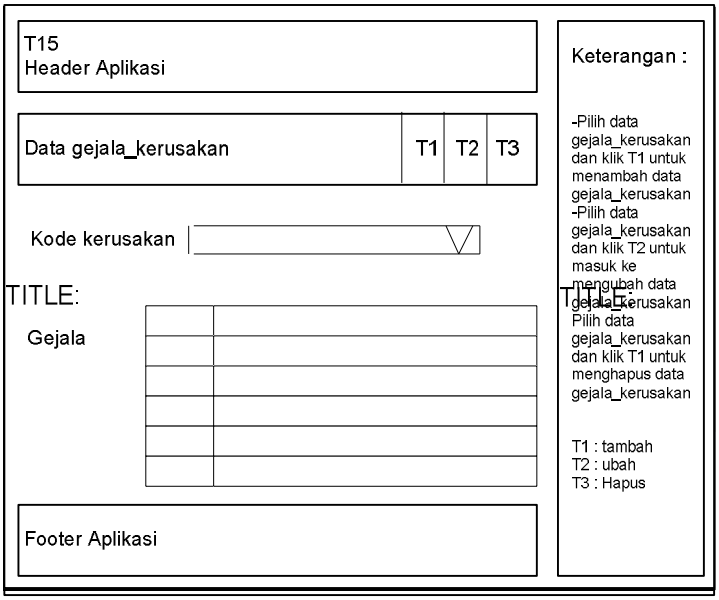

**Gambar 3.27 Tampilan Menu Data gejala**




**Gambar 3.28 Tampilan Menu Tambah, Hapus dan Ubah Data gejala**



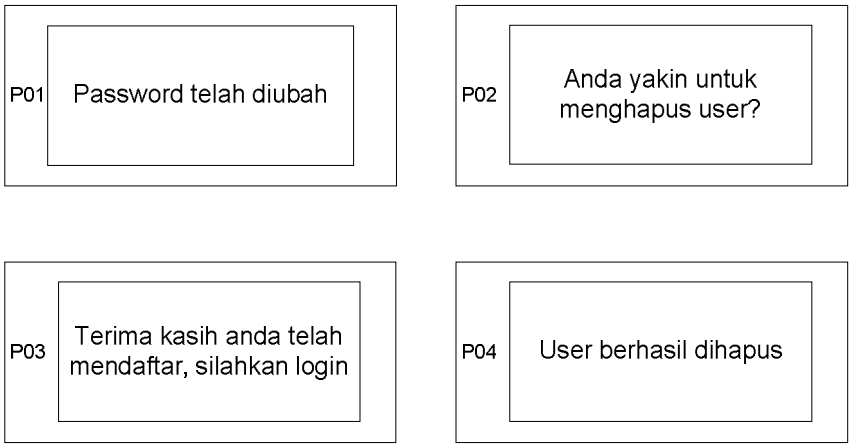

**Gambar 3.29 Tampilan Menu Data gejala_kerusakan**



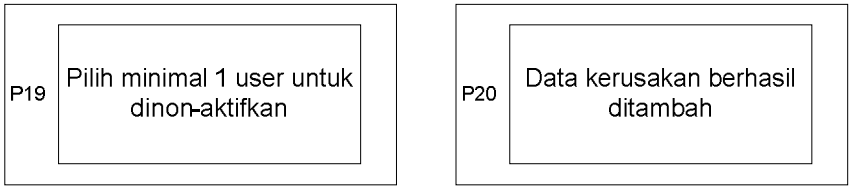

**Gambar 3.30 Tampilan Menu Ubah, Tambah dan Hapus Data gejala_kerusakan**

### 3.2.4 Perancangan Pesan

Perancangan pesan adalah keterangan yang menunjukan ke user apabila sudah melakukan kegiatan dalam sistem yaitu berupa validasi keterangan.




P05
User berhasil diupdate
P06
Non-aktifkan user?
P07
Data berita berhasil diupdate
P08
Data berita berhasil ditambah
P09
Data berita berhasil dihapus
P10
Data kerusakan berhasil diupdate
P11
Data kerusakan berhasil dihapus
P12
Anda yakin untuk menghapus kerusakan?
P13
Data gejala berhasil diupdate
P14
Data gejala berhasil ditambah
P15
Data gejala berhasil dihapus
P16
Data konsultasi dengan kerusakan berhasil diupdate
P17
Data konsultasi dengan kerusakan berhasil ditambah
P18
Data konsultasi dengan kerusakan berhasil dihapus

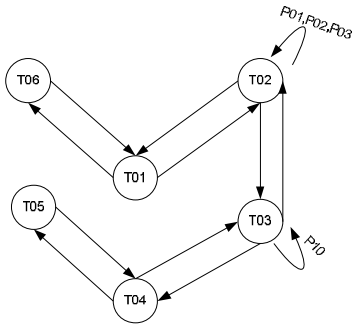



**Gambar 3.31 Perancangan Pesan**

### 3.2.5 Jaringan Semantik

Jaringan semantik merupakan alat efektif untuk merepresentasikan pemetaan data yang bertujuan mencegah terjadinya duplikasi data. Jaringan semantik terdapat sebagai berikut :

Jaringan Semantik User




**Gambar 3. 32 Jaringan Semantik User**

Keterangan :

T01 : Tampilan Menu Utama

T02 : Tampilan Registrasi user

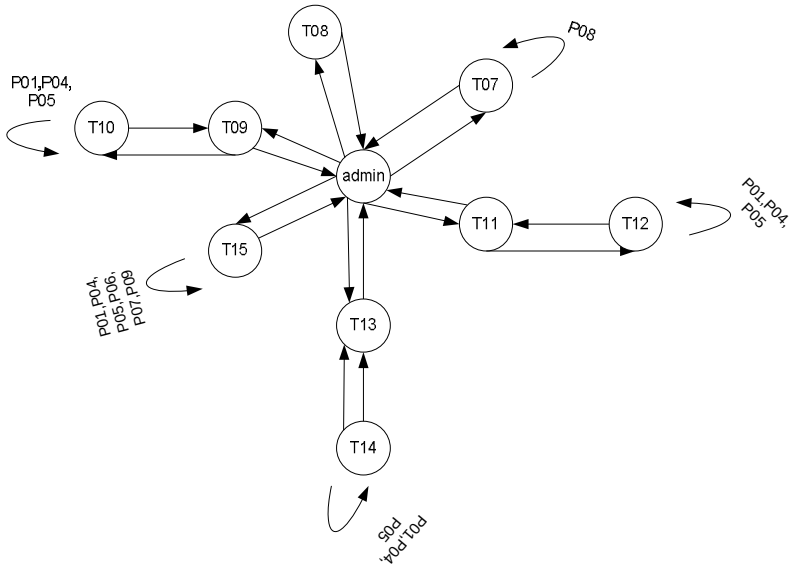
T03 : Tampilan Menu Pertanyaan (pilih kerusakan)

T04 : Tampilan Menu Pertanyaan (jawab pertanyaan)

T05 : Tampilan Menu Hasil Konsultasi

T06 : Tampilan Menu Berita

Jaringan Semantik Admin




**Gambar 3. 33 Jaringan Semantik Admin**

Keterangan :

P07 : Tampilan Ganti Password Admin

P08 : Tampilan Menu Data Berita

P09 : Tampilan Menu Tambah Ubah dan Hapus Berita

P10 : Tampilan Menu Data Kerusakan

T11 : Tampilan Menu Tambah Ubah dan Hapus Kerusakan

T12 : Tampilan Menu Data Gejala

T13 : Tampilan Menu Tambah Ubah dan Hapus Gejala

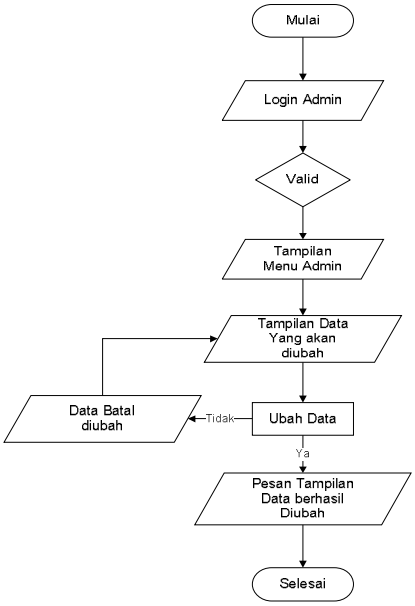
T14 : Tampilan Menu Data User

T15 : Tampilan Menu Tambah Ubah dan Hapus User

### 3.2.6 Perancangan Prosedural

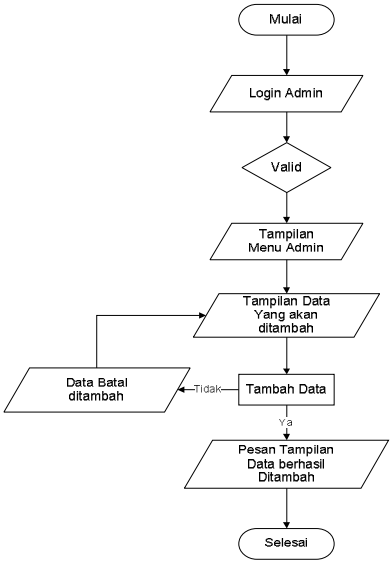
Perancangan prosedural digambarkan dalam bentuk *flow chart*. Merupakan perancangan program bagaimana sebuah aplikasi sistem pakar ini dijalankan.

1. Prosedural admin ubah data



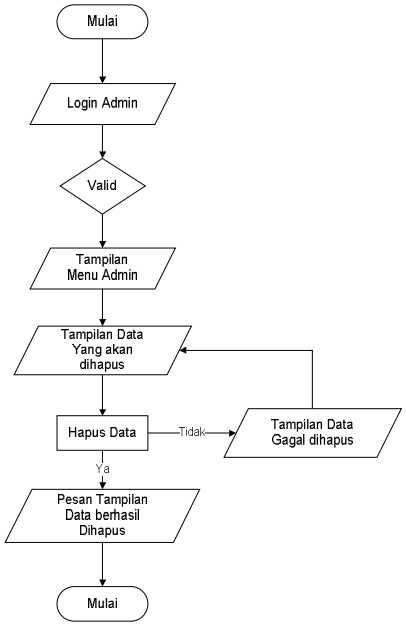

**Gambar 3.34 Prosedur Admin Ubah Data**

2. Prosedural admin tambah data



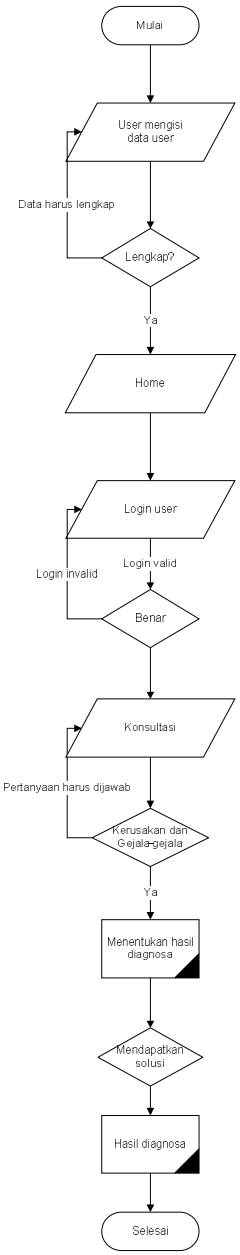

**Gambar 3.35 Prosedur Admin Tambah Data**

3. Prosedural admin hapus data




**Gambar 3.36 Prosedur Admin Hapus Data**

4. Prosedural konsultasi user




**Gambar 3.37 Prosedur Konsultasi User**