BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Salah satu masalah yang dihadapi dalam penanganan kerusakan Yamaha P 900 pada Polda Metro Jaya adalah keterbatasan sumber daya manusia khususnya untuk teknisi-teknisi yang baru mengenal Yamaha P 900, sementara kebutuhan akan tenaga teknisi semakin meningkat, sedangkan aplikasi yang berbasis pengetahuan tentang penanganan kerusakan yang dipadupadankan dengan kolom konsultasi sangat terbatas.

3.1.1 Tahapan Analisis Pembangunan Sistem Pakar

Dalam pembangunan sistem pakar untuk mendeteksi kerusakan pada motor Yamaha P 900 akan dilakukan beberapa tahapan analisis, yaitu:

1. Menganalisis data yang diperlukan dalam membangun sistem berupa jenis gejala dan kerusakan melalui studi literatur dan wawancara pada teknisi yang nantinya akan digunakan sebagai *knowledge base*.
2. Merepresentasikan pengetahuan yang telah didapat kedalam tabel gejala yang telah dianalisis (domain kasus).
3. Menganalisis *Graf Or* pada kasus.
4. Menentukan *tree* dan pada sistem yang akan dibangun (Lampiran A).
5. Menentukan evaluasi nilai heuristik, metode inferensi yang digunakan, kaidah produksi, dan penelusuran gejala dan kerusakan.

3.1.2 Analisis Data

Setiap jenis kerusakan pasti memiliki gejala-gejala yang dapat ditentukan. Antara kerusakan yang satu dengan yang lain pasti terdapat perbedaan. Demikian pula pada kerusakan Yamaha P 900, walaupun ada yang mirip tetapi salah satu diantaranya memiliki gejala khusus yang tidak dimiliki oleh kerusakan lainnya. Data mengenai jenis kerusakan dan gejala-gejalanya dari proses studi literatur dan wawancara adalah sebagai berikut:

A Kepala Silinder

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin berbunyi tidak normal, Tenaga mesin berkurang, Mesin tidak mau menyala, Generator tidak menyala.

B Piston

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin berbunyi tidak normal, Mesin kotor dan berasap tebal, Mesin terlalu panas, Tenaga mesin berkurang, Mesin tidak mau menyala, Pada kecepatan rendah motor bergerak kencang, Pada kecepatan tinggi motor bergerak lambat, Kelebihan atau kekurangan bensin, Minyak rem bocor, Busi cepat kotor.

C Katup

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin mudah mati, Mesin berbunyi tidak normal, Mesin kotor dan berasap tebal, Tenaga mesin berkurang,

Putaran stasioner tidak sempurna, Putaran mesin tidak sempurna pada saat putaran tinggi, Busi cepat kotor.

D Karburator

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin mudah mati, Tenaga mesin berkurang, Mesin tidak mau menyala, Pada kecepatan rendah motor bergerak kencang, Pada kecepatan tinggi motor bergerak lambat, Gas macet.

E Crankshaft

Gejala: Mesin berbunyi tidak normal, Busi menjadi cepat tertutup arang.

F Camshaft

Gejala: Putaran mesin tidak sempurna pada saat putaran tinggi, Generator tidak menyala.

G Crankcase

Gejala: Putaran stasioner tidak sempurna, Putaran mesin tidak sempurna pada saat putaran tinggi.

H Clutch

Gejala: Mesin berbunyi tidak normal, Mesin hidup tapi kendaraan tidak dapat berjalan, Kopling selip, Kopling lengket, Kopling selip pada pemasukan kopling pertama, Kopling selip pada pemindahan gigi transmisi, Pemindahan gigi

transmisi keras pada kopling pemindah gigi, saat bekerja pemindahan gigi transmisi terlalu cepat, saat bekerja pemindahan gigi transmisi terlalu lambat.

I Gearbox

Gejala: Gigi transmisi sulit masuk/sulit bekerja, Gigi transmisi sulit kembali ke posisi semula, Gigi transmis menghentak/lompat, Pemindahan gigi transmisi keras pada kopling pemindah gigi.

J Choke

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Putaran stasioner tidak sempurna, Mesin tidak mau menyala, Daya kerja accu menurun.

K Kick Starter

Gejala: Gigi transmisi sulit masuk/sulit bekerja, Gigi transmisi menghentak/lompat.

L Motor Starter

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Tombol starter tidak berfungsi.

M Kabel

Gejala: Sirine tidak menyala, Lampu rotator tidak menyala, Lampu hazard tidak menyala, Lampu hazard tidak berkedip normal, Lampu utama depan dan belakang tidak menyala, Public address tidak menyala, Speedometer tidak menyala, Jam

tidak menyala, Tidak ada bunga api saat penyalaan mesin, Generator tidak menyala, Generator tidak mengisi dan pengisianpun dibawah spesifikasi, Pengisian di generator tidak stabil.

N CDI

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin mudah mati, Putaran mesin tidak sempurna pada saat putaran tinggi, Tidak ada bunga api saat penyalaan mesin, Generator tidak mengisi dan pengisianpun dibawah spesifikasi.

O Busi

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin mudah mati, Tenaga mesin berkurang, Putaran stasioner tidak sempurna, Gas macet, Tidak ada bunga api saat penyalaan mesin, Busi menjadi cepat tertutup arang, Busi terlalu panas/hangus.

P Accu

Gejala: Sirine tidak berbunyi dengan stabil, Lampu rotator tidak berputar, Lampu hazard tidak berkedip normal, Lampu utama depan dan belakang redup, Public address tidak menyala, Speedometer tidak menyala, Jam tidak menyala, Generator tidak mengisi dan pengisianpun dibawah spesifikasi, Pengisian di generator berlebihan, Tombol starter tidak berfungsi, Terdapat sulfasi pada pelat sel yang berhubungan, Kapasitas accu cepat menurun, Pemasangan accu terbalik, Terdapat sulfasi pada accu, Daya kerja accu menurun.

Q Sekering

Gejala: Sirine tidak menyala, Sirine tidak berbunyi dengan stabil, Lampu rotator tidak menyala, Lampu rotator tidak berputar, Lampu hazard tidak menyala, Lampu hazard tidak berkedip normal, Lampu utama depan dan belakang tidak menyala, Lampu utama depan dan belakang redup, Public address tidak menyala, Jam tidak menyala, Tombol starter tidak berfungsi.

R Regulator

Gejala: Generator tidak menyala, Generator tidak mengisi dan pengisianpun dibawah spesifikasi, Pengisian di generator berlebihan, Pengisian di generator tidak stabil.

S Pompa Oli

Gejala: Mesin terlalu panas, Tenaga mesin berkurang, Pemindahan gigi transmisi keras pada kopling pemindah gigi.

T Pipa Oli

Gejala: Mesin kotor dan berasap tebal.

U Membran dan Magnet Sirine

Gejala: Sirine tidak menyala, Sirine tidak berbunyi dengan stabil.

V Speaker

Gejala: Sirine tidak berbunyi dengan stabil, Public addres tidak menyala.

W Bohlam Lampu

Gejala: Lampu rotator tidak menyala, Lampu hazard tidak menyala, Lampu hazard tidak berkedip normal, Lampu utama depan dan belakang tidak menyala, Lampu utama depan dan belakang redup.

X Mic

Gejala: Public address tidak menyala.

Y Speedometer

Gejala: Speedometer tidak menyala.

Z Dinamo

Gejala: Lampu rotator tidak berputar.

AA Suspensi Depan

Gejala: Kemudi goyang, Roda depan goyang, Suspensi depan terlalu lemah, Suspensi depan terlalu keras, Suspensi depan berbunyi.

AB Suspensi Belakang

Gejala: Roda belakang goyang, Suspensi belakang terlalu lemah, Suspensi belakang terlalu keras, Suspensi belakang berbunyi.

AC Kemudi

Gejala: Sukar mengemudikan/kemudi berat, Kemudi goyang.

AD Rem

Gejala: Daya rem berkurang, Rem berbunyi, Gerak tuas rem sulit, Minyak rem bocor.

AE Velg

Gejala: Kemudi goyang, Roda depan goyang, Roda belakang goyang, Rem berbunyi.

AF Gardan

Gejala: Sukar mengemudikan/kemudi berat, Roda belakang terasa berat, Gardan berbunyi tidak normal.

AG Tanki Bensin

Gejala: Kelebihan/kekurangan bensin.

AH Selang Bensin

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin mudah mati, Putaran stasioner tidak sempurna, Mesin tidak mau menyala, Gas macet.

AI Selang Ventilasi Udara

Gejala: Mesin tidak mau hidup/sulit dihidupkan.

AJ Selang Vaccum

Gejala: Mesin mudah mati, Mesin terlalu panas, Mesin tidak mau menyala.

AK Filter Bensin

Gejala: Mesin hidup tetapi kendaraan tidak dapat berjalan, Putaran stasioner tidak sempurna, Gas macet, Busi menjadi cepat tertutup arang.

AL Filter Udara

Gejala: Mesin terlalu panas, Tenaga mesin berkurang, Mesin hidup tetapi kendaraan tidak dapat berjalan, Putaran stasioner tidak sempurna.

AM Ban

Gejala: Sukar mengemudikan/kemudi berat, Roda depan goyang, Roda belakang goyang, Roda belakang terasa berat

AN Rantai

Gejala: Mesin terlalu panas. Tenaga mesin berkurang, Mesin hidup tetapi kendaraan tidak dapat berjalan.

AO Knalpot

Gejala: Gas macet, Busi menjadi cepat tertutup arang, Busi terlalu panas/hangus, Bunyi knalpot tidak normal, Bunyi knalpot meletup/meledak.

AP Oli

Gejala: Mesin terlalu panas, Gas Macet, Rem berbunyi.

Kemudian gejala dan kerusakan analisis data diatas dikonversikan ke dalam relasi domain antara gejala dan kerusakan Yamaha P 900 (Lihat Tabel 3.1 Domain Gejala dan Kerusakan Yamaha P 900).

Tabel 3.1 Domain Gejala dan Kerusakan Yamaha P 900
(Pada Kertas A3)

3.1.3 Analisis Algoritma Graf Or Pada Kasus

Pada dasarnya Graf merepresentasikan graph yang bernilai *true* and *false* kedalam grafik biner yang memiliki 0, 1 atau 2 node. Pada kasus ini, *root* tidak akan mungkin memiliki 0 node (hanya bisa memiliki 1 atau 2), kecuali dia bertindak sebagai *leaf*. Implementasi Graf Or dibutuhkan 2 antrian node, yaitu:

1. OPEN, yang berisi node-node yang sudah dibangkitkan, sudah memiliki fungsi heuristik namun belum diuji. Umumnya berupa antrian berprioritas yang berisi elemen-elemen dengan nilai heuristik tertinggi.
2. CLOSED, berisi node-node yang sudah diuji.

Dalam 2 antrian tersebut harus memiliki fungsi heuristik seberapa baik dibangkitkannya setiap node (Lihat Tabel 3.4 Pencarian Heuristik Node Kerusakan T). Dalam hal ini fungsi f'(n) merupakan kombinasi dari 2 komponen yaitu g(n) dan h(n). Adapun fungsi heuristik tersebut meliputi nilai probabilitas g(n) dan nilai lintasan terpendek h(n).

Algoritma Graf Or:

1. Tempatkan node awal (*root*) pada antrian OPEN
2. Kerjakan langkah-langkah berikut hingga tujuan ditemukan atau antrian OPEN sudah kosong:
    a. Ambil node terbaik dari OPEN;
    b. Bangkitkan semua *successor*nya;
    c. Untuk tiap-tiap *successor* kerjakan:
        i. Jika node tersebut belum pernah dibangkitkan sebelumnya, evaluasi node tersebut dan masukan ke OPEN;

ii. Jika node tersebut sudah pernah dibangkitkan sebelumnya, ubah *parent* jika lintasan baru lebih menjanjikan. Hapus node tersebut dari antrian OPEN.

Berikut adalah penjelasan langsung ke dalam contoh kasus pencarian kerusakan motor Yamaha P 900 pada Kerusakan T (Lihat Gambar 3.1 Tree Penelusuran Kerusakan T (Pipa Oli)):

1. Tempatkan node 1 pada antrian OPEN.
2. Ambil node terbaik dari OPEN, dalam hal ini node yang terbaik adalah node 2 dan node 3 (nilai evaluasi heuristik terbaik, lihat Tabel 3.3 Pencarian Heuristik Node Kerusakan T (*sample*)).
3. Node 2 dan node 3 dibangkitkan sebagai *successor* dari node 1.
4. Bangkitkan node 2 dan node 3, evaluasi node 2 dan node 3, kemudian masukan ke dalam antrian OPEN.
5. Node 3 memiliki nilai evaluasi yang lebih baik dan *parent* lintasan diubah menuju node 4 dan node 5.
6. Tempatkan node 3 pada antrian OPEN.
7. Node 2 di CLOSED.
8. Ambil node terbaik dari OPEN, dalam hal ini node yang terbaik adalah node 4 dan node 5 (nilai evaluasi heuristik terbaik).
9. Node 4 dan node 5 dibangkitkan sebagai *successor* dari node 3.
10. Bangkitkan node 4 dan node 5, evaluasi node 4 dan node 5, kemudian masukan ke dalam antrian OPEN.

11. Node 4 memiliki nilai evaluasi yang lebih baik dan tujuan ditemukan (*goal*), yaitu node T.
12. Pencarian berakhir dengan node T sebagai *goal state*.

Untuk memperjelas hasil penerapan algoritma, akan diperjelas dengan menggunakan metode pencarian heuristik antara gejala dan kerusakan dibawah ini. Cara untuk membangun Graf Or yang merupakan bagian dari sistem pencarian heuristik *best first search* dalam penelusuran gejala hingga menuju *goal*, penjelasannya adalah sebagai berikut:

1. Penentuan Initial State

   Dalam penentuan *initial state* yang berarti awal mula sistem ini akan berjalan dengan menempatkan node awal sebagai *root*, dimana *root* awal adalah nilai probabilitas gejala terbaik (frekuensi gejala yang paling banyak keluar pada domain kerusakan).

2. Current State (Graf Or dalam Pencarian Heuristik)

   Menggerjakan pencarian selanjutnya setelah *initial state* terbentuk, yaitu mengambil node terbaik dari OPEN. Maksudnya adalah mencari node terbaik dibawah *root* (OPEN). Dengan probabilitas yang dimiliki kemungkinan berikutnya akan dibentuk 1 atau 2 node yang dianggap terbaik. Dan node tersebut akan bernilai *true* atau *false* (*Boolean Function*). Setelah Node tersebut (*successor*) terbentuk, tiap-tiap *successor* akan dievaluasi sesuai dengan nilai heuristik yang terbaik (lintasan terbaik) dan kemudian akan dideklarasikan atau dimasukan kedalam OPEN agar bisa ditelusuri kembali. Dalam hal ini *successor* tersebut bernilai *true.* Sedangkan node berikutnya

yang sudah dibangkitkan sebelumnya, *parent*-nya akan diubah kedalam lintasan lain yang lebih menjanjikan (lintasan dengan nilai evaluasi heuristik terbaik). Hapus node tersebut dari antrian OPEN. Node dengan kondisi *false* akan dimasukan pada kategori *end node*.

3. Evaluasi Goal State

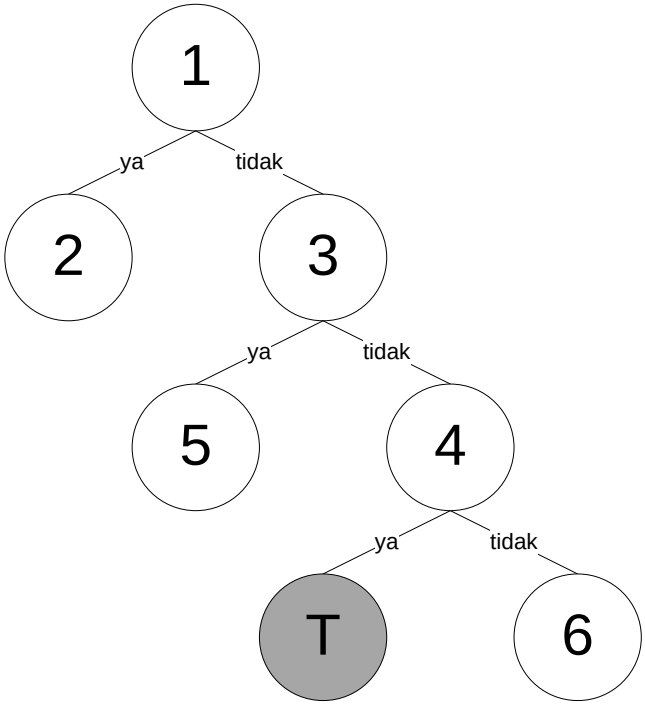
   Hasil dari evaluasi nilai heuristik pada tiap-tiap node gejala akan menentukan node penyebab kerusakan (*goal state*). Saat menemukan *goal state* pencarian berakhir.

3.1.4 Analisis Evaluasi Nilai Heuristik Pada Kasus

Ketika melakukan pencarian kerusakan dengan menggunakan Graf Or, maka hasil yang akan dicari dari pencarian gejala yang dialami akan ditampilkan sesuai nilai evaluasi heuristik terbaik. Nilai evaluasi heuristik terbaik adalah nilai yang memiliki fungsi heuristik yang akan mengestimasi seberapa baik tiap-tiap node (hanya 2 node dengan kondisi true atau false) dibangkitkan. Biasanya fungsi evaluasi dideklarasikan sebagai f(n) dan f'(n). fungsi tersebut merupakan evaluasi hasil estimasi lintasan dari *initial state* sampai kepada *goal state*. Nilai evaluasi terbaik adalah lintasan yang terpendek dengan node yang memiliki nilai probabilitas terbaik. Jadi, jarak terpendek pencarian dengan nilai probabilitas tiap node dari bagian sistem pencarian *best first search* yang maksimal adalah penerapan Graf Or di aplikasi sistem pakar kerusakan Yamaha P 900. Pada sistem penerapan gejala kerusakan diinisialisasikan dengan kode 1-67 (terdapat 67 gejala kerusakan) dan penyebab kerusakan diinisialisasikan menggunakan kode A-AP

(terdapat 42 penyebab kerusakan) (lihat Tabel 3.1 Domain Gejala dan Kerusakan Yamaha P 900).

Gambar Analisis Tree Graf Or mengacu pada Tabel 3.1 Domain Gejala dan Kerusakan Yamaha P 900, dan Gambar Tree Graf Or 67 Gejala dengan 42 Kerusakan dapat dilihat pada Lampiran A.



Gambar 3.1 Tree Penelusuran Kerusakan T (Pipa Oli)

Dalam pencariannya, tiap node terdapat nilai *heuristik* sehingga dapat dibandingkan nilai *heuristiknya* pada tiap node. Dengan asumsi jarak terpendek dan nilai probabilitas yang terbaik menjadi nilai evaluasi *heuristik* terbaik.

Tabel 3.2 Nilai Probabilitas Gejala

Kode Gejala	Frekuensi Gejala	Probabilitas (Frekuensi Gejala / 42)	Keterangan
1	10	0.2381	root
2	6	0.1429	-
3	5	0.1190	-
4	3	0.0714	-
5	6	0.1429	-
6	8	0.1905	-
7	4	0.0952	-
8	7	0.1667	-
9	4	0.0952	-
10	1	0.0238	-
11	1	0.0238	-
12	2	0.0476	-
13	1	0.0238	-
14	2	0.0476	-
15	1	0.0238	-
16	1	0.0238	-
17	3	0.0714	-
18	1	0.0238	-
19	1	0.0238	-
20	6	0.1429	-
21	2	0.0476	-
22	2	0.0476	-
23	2	0.0476	-
24	6	0.1429	-
25	3	0.0714	-
26	3	0.0714	-
27	3	0.0714	-
28	1	0.0238	-
29	1	0.0238	-
30	1	0.0238	-
31	3	0.0714	-
32	1	0.0238	-
33	1	0.0238	-
34	1	0.0238	-
35	2	0.0476	-
36	1	0.0238	-
37	1	0.0238	-
38	3	0.0714	-
39	1	0.0238	-
40	2	0.0476	-
41	3	0.0714	-

Kode Gejala	Frekuensi Gejala	Probabilitas (Frekuensi Gejala / 42)	Keterangan
42	4	0.0952	-
43	3	0.0714	-
44	3	0.0714	-
45	3	0.0714	-
46	4	0.0952	-
47	3	0.0714	-
48	3	0.0714	-
49	5	0.1190	-
50	3	0.0714	-
51	3	0.0714	-
52	3	0.0714	-
53	4	0.0952	-
54	2	0.0476	-
55	2	0.0476	-
56	4	0.0952	-
57	4	0.0952	-
58	2	0.0476	-
59	2	0.0476	-
60	3	0.0714	-
61	1	0.0238	-
62	1	0.0238	-
63	1	0.0238	-
64	1	0.0238	-
65	2	0.0476	-
66	1	0.0238	-
67	1	0.0238	-

Keterangan :

a. Kode Gejala

   Kode Gejala terdapat pada Tabel 3.1 Domain Gejala dan Kerusakan (1-67).

b. Frekuensi Gejala Pada Domain Kerusakan

   Frekuensi gejala adalah banyaknya jumlah gejala yang ada pada kerusakan.

c. Probabilitas

Probabilitas = Frekuensi Gejala / Jumlah Kerusakan

Terdapat 67 Gejala dan 42 Jumlah Kerusakan

Sample : Probabilitas Gejala 1 = 10 / 42 = **0.2381**

Nilai heuristik yang dimungkinkan terbaik adalah dengan asumsi nilai dengan jarak lintasan terbaik yang lebih kecil dengan node yang memiliki nilai probabilitas terbaik.

Batasan pencarian nilai probabilitas pencariannya berdasarkan perhitungan frekuensi gejala yang sering muncul pada 42 kerusakan yang terjadi. Lihat Tabel 3.2. Probabilitas yang terbaik adalah nilai probabilitas terbesar dimana frekuensi gejala yang muncul lebih banyak dibanding yang gejala yang lain pada banyaknya kerusakan.

Contoh kasus (*sample*) nilai probabilitas Gejala 1

P(1) = 10 / 42 = 0,238

Keterangan:

P = Probabilitas

1 = Gejala Pertama

10 = Frekuensi Gejala Pertama

42 = Total Jumlah Kerusakan

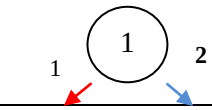
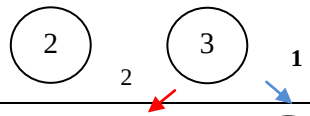
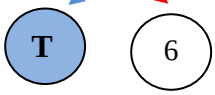
0,238 = Nilai Probabilitas

Selanjutnya untuk perhitungan nilai evaluasi heuristik adalah mencakup *gabungan aspek lintasan yang terpendek dan nilai probabilitas yang terbesar*. Dengan kata lain:

Nilai Evaluasi Heuristik = Lintasan Terpendek ∪ Nilai Probabilitas Terbesar

(Gabungan lintasan terpendek dan nilai probabilitas terbesar / lintasan terpendek *union* nilai probabilitas terbesar).

Tabel 3.3 Pencarian Heuristik Node Kerusakan T (*sample*)

Posisi	Kondisi	Lintasan	Probabilitas Node	Nilai Evaluasi Heuristik	Hasil
Root = 1		-	1 = 0,2381	-	1 ≠ **T**
Child = 3		2	2 = 0,1429 3 = 0,1190	2 ∪ 0,1190	3 ≠ **T**
Child = 4		1	4 = 0,0714 5 = 0,1429	1 ∪ 0,0714	4 ≠ **T**
Goal = T		-	6 = 0,1905	-	T = **T**

Kesimpulan:

Lintasan yang didapat untuk mencapai Node Kerusakan **T** adalah **1-3-4**

Terdapat 3 kemungkinan yang terjadi pada contoh kasus:

1. Algoritma akan berhenti saat mencapai pencarian optimum, yaitu saat ditemukannya *goal state*.

2. *Current state* pada lintasan yang tidak memiliki nilai heuristik terbaik akan dihapus pada antrian OPEN (tidak akan dilacak dan ditelusuri). Dan tidak diizinkan untuk melihat satupun langkah sebelumnya.
3. Saat menuju *goal state*, pencarian berakhir.

3.1.5 Struktur Data Graf Or

Struktur Data Graf Or

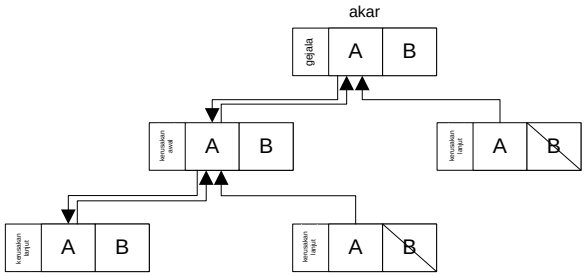
```
Antrian = ^Point;
  Point = record
    Data : Tree;
    Next : Antrian;
    End;
```

Penjelasan pada struktur data diatas adalah sebagai berikut:

Terdapat antrian yaitu antrian1 dan antrian2. Antrian1 menjelaskan penyimpanan simpul sementara yang sudah diperiksa (diperiksa apakah *goal state* atau tidak) dan antrian 2 menjelaskan penyimpanan simpul sementara yang akan diperiksa (Lihat Gambar 3.2 Struktur Data Graf Or). Datanya berupa *tree* dengan 2 cabang (*child*) dengan kondisi masing-masing yaitu *true* dan *false*. Setiap antrian berikutnya akan ditetapkan menjadi *tree* dengan 2 anak cabang sampai menuju *goal state*.

Diasumsikan terdapat 2 anak cabang yang termasuk kategori *low* dan *high*, hal ini berkaitan dengan nilai evaluasi yang terbaik. Untuk selanjutnya tiap-tiap anak cabang akan memiliki sebuah fungsi nilai *boolean*, dimana kondisinya hanya 1 atau 0, yaitu *true* atau *false* (pada implementasi kasus yaitu ‘ya’ atau

'tidak'). Untuk melintasi suatu lintasan akan ada beberapa operasi pada simpul, operasi tersebut merupakan biaya estimasi nilai evaluasi heuristik. Pada *high vertek* dan *low vertek* akan ditandai dari simpul, mana yang *true* dan mana yang *false* (high vertek maksudnya adalah menelusuri node dengan nilai evaluasi heuristik terbaik demikian sebaliknya pada low vertek). Kemudian sistematis kunjungan tiap node dengan rekursif akan menuju subgraf berakar 2 anak (*leaf*) dan akan dibandingkan sesuai dengan nilai evaluasi masing-masing.



Keterangan: A = Parent

B = Sibling

Gambar 3.2 Struktur Data Graf Or

3.1.6 Penerapan Algoritma Graf Or Dalam Kaidah Produksi

Pada tahap awal akan tampil sebuah pertanyaan yang bersangkutan dengan sebuah gejala. Terdapat 67 Gejala dengan 42 Kerusakan. Adapun relasi antar objeknya dapat dilihat pada Domain Gejala dan Kerusakan Yamaha P 900 (Tabel 3.1) dan ketentuan-ketentuan kondisi sebab-akibat dapat dilihat pada Halaman 55 subbab Analisis Data. Data yang ditelusuri adalah menggunakan inferensi *Forward Chaining*, pendekatan penalaran yang dimotori data (*data driven*). Dalam pendekatan ini pelacakan dimulai dari fakta terlebih dahulu

kemudian menuju konklusi. Proses pelacakan menggunakan *premis* untuk aturan sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses ini berlanjut sampai semua kemungkinan ditemukan. Contoh Kasus (*sample*):

Kerusakan A: Kepala Silinder

Gejala: Mesin tidak mau hidup/sulit dihidupkan, Mesin berbunyi tidak normal, Tenaga mesin berkurang, Mesin tidak mau menyala, Generator tidak menyala.

Pengkodean:

1, 3, 6, 20, 56 → A

Penjelasan:

a. Pada level pertama sudah jelas, bahwa node 1 bertindak sebagai root. Node 1 memiliki fungsi Boolean, dimana memiliki kondisi *true* atau *false* yang merupakan inisialisasi dari *high vertek* dan *low vertek*. Graf Or akan mengunjungi node kerusakan selanjutnya sesuai inputan kondisi. Dan berlanjut kepada cabang yang memiliki 2 anak yang menjadi *current state*.
b. Pada level kedua, Graf Or akan mengunjungi node kerusakan selanjutnya (node 3) sesuai inputan kondisi. Dan berlanjut kepada cabang kembali yang memiliki 2 anak yang menjadi *current state*.
c. Pada level ketiga, Graf Or akan mengunjungi node kerusakan selanjutnya (node 6) sesuai inputan kondisi. Dan berlanjut kepada cabang kembali yang memiliki 2 anak yang menjadi *current state*.

d. Pada level keempat, Graf Or akan mengunjungi node kerusakan selanjutnya (node 20) sesuai inputan kondisi. Dan berlanjut kepada cabang kembali yang memiliki 2 anak yang menjadi *current state*.
e. Pada level kelima, Graf Or akan mengunjungi node kerusakan selanjutnya (node 56) sesuai inputan kondisi. Dan berlanjut kepada cabang kembali yang memiliki 2 anak yang menjadi *current state*.
f. Pada level keenam, Graf Or akan mengunjungi node solusi (*goal state*) setelah melalui penelusuran dari node 1, 3, 6, 20, dan 56 (setelah node gejala ditandai/*mark*).

Algoritma yang didapat dari kasus ini menerapkan aturan kaidah produksi yaitu sebagai berikut:

1. Gejala menjadi *initial state*, karena dalam kasus ini yang menjadi kunci yaitu kerusakan Yamaha P 900.
2. Gejala berikutnya yang tampil setelah pencarian sebelum-sebelumnya menjadi *current state*.
3. Kerusakan dan Solusi menjadi *goal state*.

Sebagai contoh dapat dilihat pada pernyataan If-Then diagnosis penyebab kerusakan dari gejala kerusakan motor Yamaha P 900.

Algoritma A - AP:

if Mesin tidak mau hidup/sulit dihidupkan, if Mesin berbunyi tidak normal, if Tenaga mesin berkurang, if Mesin tidak mau menyala, if Generator tidak menyala. then

Kerusakan Kepala Silinder

return

end

…

if Mesin terlalu panas, if Gas Macet, if Rem berbunyi. then

Kerusakan Oli

return

end

3.1.7 Analisis Kasus Pada Prolog

Prolog terdiri dari kumpulan-kumpulan data-data objek yang merupakan suatu fakta. Karena fakta menunjukan suatu keadaan atau situasi nyata maka fakta selalu benar (‘true’).

Aturan adalah suatu pernyataan yang menunjukan bagaimana fakta-fakta berinteraksi satu dengan yang lain dan membentuk relasi menuju suatu kesimpulan. Apabila sebuah relasi yang sama dapat dituliskan lebih dari satu aturan, maka aturan-aturan tersebut bersifat “atau” (sebagai pilihan).

Setiap aturan terdiri dari kesimpulan (kepala) dan tubuh. Tubuh dapat terdiri dari 1 atau lebih pernyataan atau aturan yang lain yang disebut *subgoal*. Antara *subgoal* yang satu dengan yang lain dihubungkan dengan logika “and”, yang bersifat: *kesimpulan (kepala) menjadi benar jika dan hanya jika semua bagian tubuh tersebut benar*. Dengan kata lain: *kepala (head) benar jika tubuh (body) benar*. Sifat ini disebut *then/if conditional*.

Apabila semua *subgoal* didalam aturan tersebut benar, maka eksekusi akan dicapai. Kondisi tersebut disebut berhasil (*succeed*). Dan apabila salah satu bagian tubuh salah, maka kesimpulan (kepala) tersebut menjadi salah dan eksekusi tidak akan dicapai. Kondis tersebut disebut gagal (*failed*).

Berikut adalah contoh kasus (*sample*) pada Kerusakan A, yaitu sebagai berikut:

Domain Kerusakan A

Kerusakan: Kepala Silinder

Gejala: Mesin tidak mau hidup/sulit dihidupkan,

Mesin berbunyi tidak normal,

Tenaga mesin berkurang,

Mesin tidak mau menyala,

Generator tidak menyala.

Algoritma Kerusakan A

A. IF Mesin tidak mau hidup/sulit dihidupkan

AND Mesin berbunyi tidak normal

AND Tenaga Mesin berkurang

AND Mesin tidak mau menyala

AND Generator tidak menyala

THEN Kerusakannya adalah Kepala Silinder

Implementasi Algoritma Kerusakan A pada Prolog

```
Diagnosis(“kepala silinder"):-
    gejala("mesin_sulit_hidup"),
    gejala("bunyi_mesin_abnormal"),
    gejala("tenaga_mesin_berkurang"),
    gejala("mesin_tidak_menyala"),
    gejala("generator_tidak_menyala"),
    penyebab("kepala silinder").
```

3.1.8 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikannya.

3.1.8.1 Analisis Kebutuhan Non Fungsional

Penjabaran kebutuhan-kebutuhan fungsional yang diperlukan dalam pembangunan sistem pakar untuk mendeteksi kerusakan pada motor Yamaha P 900.

3.1.8.1.1 Analisis User

Sistem deteksi kerusakan Yamaha P 900 yang akan dibangun, ditujukan kepada teknisi dan polisi yang menggunakan Yamaha P 900 dengan spsesifikasi yaitu: dapat menggunakan komputer dan terbiasa dengan tampilan under DOS.

3.1.8.1.2 Analisis Perangkat Keras

Spesifikasi perangkat keras yang terdapat pada Polda Metro Jaya adalah sebagai berikut:

1. Intel Dual Core ~2,6 GHz
2. Motherboard ASUS P4P800
3. Harddisk 160 Gb
4. Memory 1 Gb
5. VGA Onboard 256 Mb
6. Monitor LCD 17 Inch
7. Optical Mouse
8. Keyboard

Adapun kebutuhan minimal perangkat keras yang diperlukan dalam pengimplementasian sistem, yaitu:

1. Intel Pentium 4 ~1,86 GHz
2. Harddisk Space 10 Mb
3. Memory 128 Mb
4. VGA 32 Mb
5. Monitor 800 x 600 pixel
6. Optical Mouse
7. Keyboard

Berdasarkan hasil evaluasi dari keadaan dilapangan dengan kebutuhan minimal perangkat keras yang dibutuhkan dalam pengimplementasian, aplikasi ini dapat berjalan sebagaimana mestinya, sesuai dengan yang diharapkan.

3.1.8.1.3 Analisis Perangkat Lunak

Spesifikasi perangkat lunak yang terdapat pada Polda Metro Jaya adalah sebagai berikut:

1. Sistem Operasi Windows 7
2. Delphi 7
3. MySQL

Kebutuhan minimal perangkat lunak yang dibutuhkan adalah :

1. Sistem Operasi Windows XP1
2. Turbo Prolog 2.0
3. Delphi 7
4. WinMysqlAdmin 1.4

Berdasarkan hasil evaluasi dari keadaan dilapangan dengan kebutuhan minimal perangkat lunak yang dibutuhkan dalam pengimplementasian, aplikasi ini dapat berjalan sebagaimana mestinya, sesuai dengan yang diharapkan.

Kebutuhan minimal perangkat keras dan perangkat lunak sudah tercakup pada Polda Metro Jaya, sehingga dapat diasumsikan sistem akan dapat berjalan sebagaimana mestinya, tentunya dengan sedikit pengarahan dari perekayasa sistem.

Mengulas dari bab sebelumnya bahwa peran Delphi dalam kaitannya dengan prolog adalah sebagai media interface dalam melakukan update gejala, kerusakan, dan solusi (perbaikan basis pengetahuan, aturan dan fakta) sebagaimana prolog dalam tampilan Under DOS dapat ditangani dengan menggunakan Delphi yang telah terintegrasi dengan mysql sebagai databasenya.

Sistem update disini, merupakan update query pada file prolog (.PRO). Update yang tersimpan pada mysql akan diinsert pada baris file .PRO ketika melakukan update gejala, kerusakan dan solusi pada Delphi. Kemudian file .PRO akan dikonversikan ke turbo prolog 2.0 untuk selanjutnya di*compile* dan di*run* sesuai dengan maksud aplikasi sistem pakar yaitu akan menampilkan data-data berupa pertanyaan gejala yang mengarah kepada kerusakan dan solusi sebuah masalah.

3.1.8.1.4 Analisis Sistem Usulan

Untuk mengatasi masalah keterbatasan tenaga spesialis maka akan dibangun suatu aplikasi sistem pakar yang dapat membantu teknisi dalam mendiagnosis suatu jenis kerusakan yang dialami oleh Yamaha P 900. Sistem diagnosis yang diusulkan mencakup penyajian pertanyaan gejala mengenai kerusakan Yamaha P 900 serta solusi yang disarankan. Sistem pakar deteksi kerusakan motor Yamaha P 900 memiliki ruang lingkup sebagai berikut:

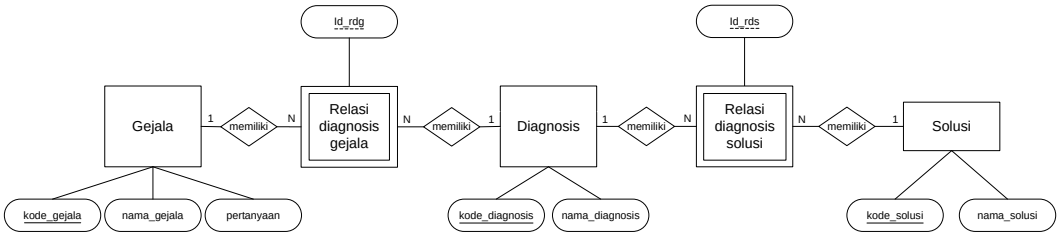
1. Informasi yang didapat dari seorang pakar, yaitu seorang teknisi yang memahami mesin dan komponen yang terdapat pada Yamaha P 900.
2. Informasi yang diberikan kepada pemakai yaitu berupa rentetan pertanyaan gejala-gejala yang telah diatur oleh sistem dan nantinya akan diperoleh kesimpulan tentang kerusakan yang dialami serta solusi penanganannya.

3.1.8.2 Kebutuhan Fungsional

Kebutuhan fungsional merupakan tahap perancangan pada aliran informasi yang meliputi perancangan diagram konteks dan aliran informasi yang direpresentasikan dengan menggunakan Data Flow Diagram.

3.1.8.2.1 Entity Relationship Diagram

Dalam analisis data terdapat 5 entitas yaitu 3 *strong entity* dan 2 *weak entity* (entitas assosiatif). 3 entitas kuat diantaranya entitas Gejala, Diagnosis, dan Solusi. Entitas Gejala memiliki 3 atribut, yaitu kode_gejala, nama_gejala, dan pertanyaan. Entitas Diagnosis memiliki 2 atribut, yaitu kode_diagnosis dan nama_diagnosis. Sedangkan Entitas Solusi memiliki 2 atribut, yaitu kode_solusi dan nama_solusi. 2 entitas lemah yang ada karena relasi antara ketiga entitas kuat yang ada keterhubungannya adalah N ke N (dikondisikan banyak gejala memiliki banyak diagnosis, demikian seterusnya), sehingga perlu dijabarkan melalui penyediaan entitas lemah, yaitu Entitas Relasi Diagnosis-Gejala dengan id_rdg sebagai Primary Key dan Entitas Relasi Diagnosis-Solusi dengan id_rds sebagai Primary Key.

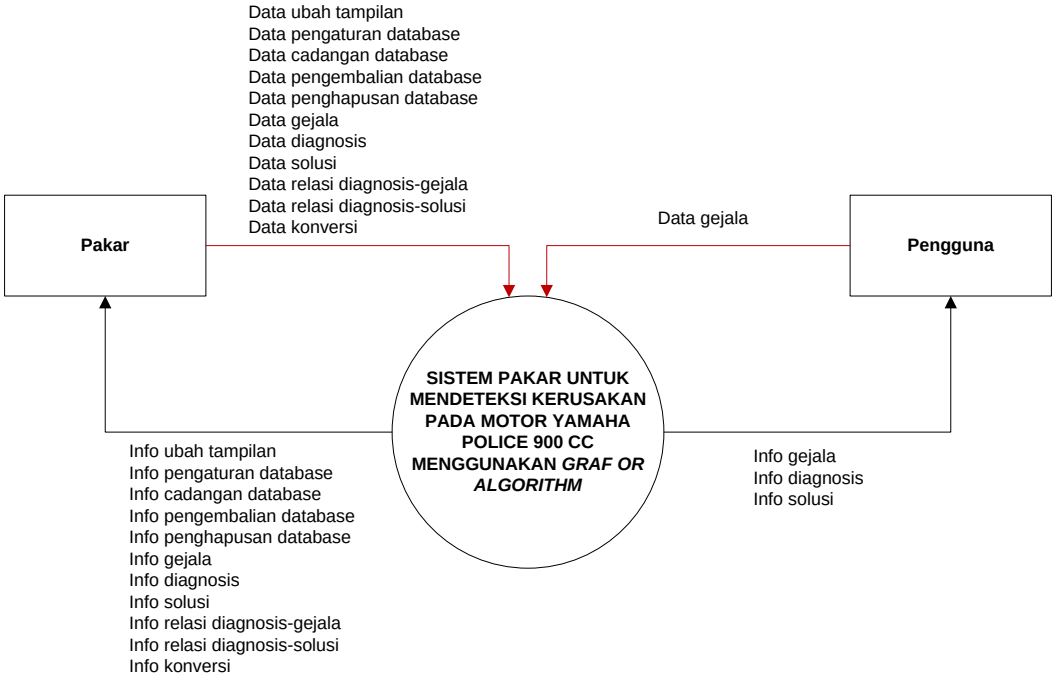


Gambar 3.3 ERD Aplikasi Sistem Pakar

Satu gejala menempatkan pada relasi diagnosis_gejala bisa satu atau lebih dari satu dan satu diagnosis memiliki satu atau lebih dari satu gejala yang direlasikan pada relasi diagnosis-gejala. Kemudian satu solusi menempatkan pada relasi diagnosis_solusi bisa satu atau lebih dari satu dan diagnosis memiliki juga satu atau lebih dari satu solusi yang direlasikan pada relasi diagnosis-solusi. Entitas relasi diagnosis-gejala dan entitas relasi diagnosis-solusi merupakan entitas assosiatif. Dalam hal ini entitas assosiatif kardinalitas dari anak ke induk selalu satu dan hanya satu. Hal tersebut yang membuat pendeklarasian dari entitas kuat ke entitas kuat selalu satu dan hanya satu.

3.1.8.2.2 Diagram Konteks

Diagram konteks dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 adalah sebagai berikut:



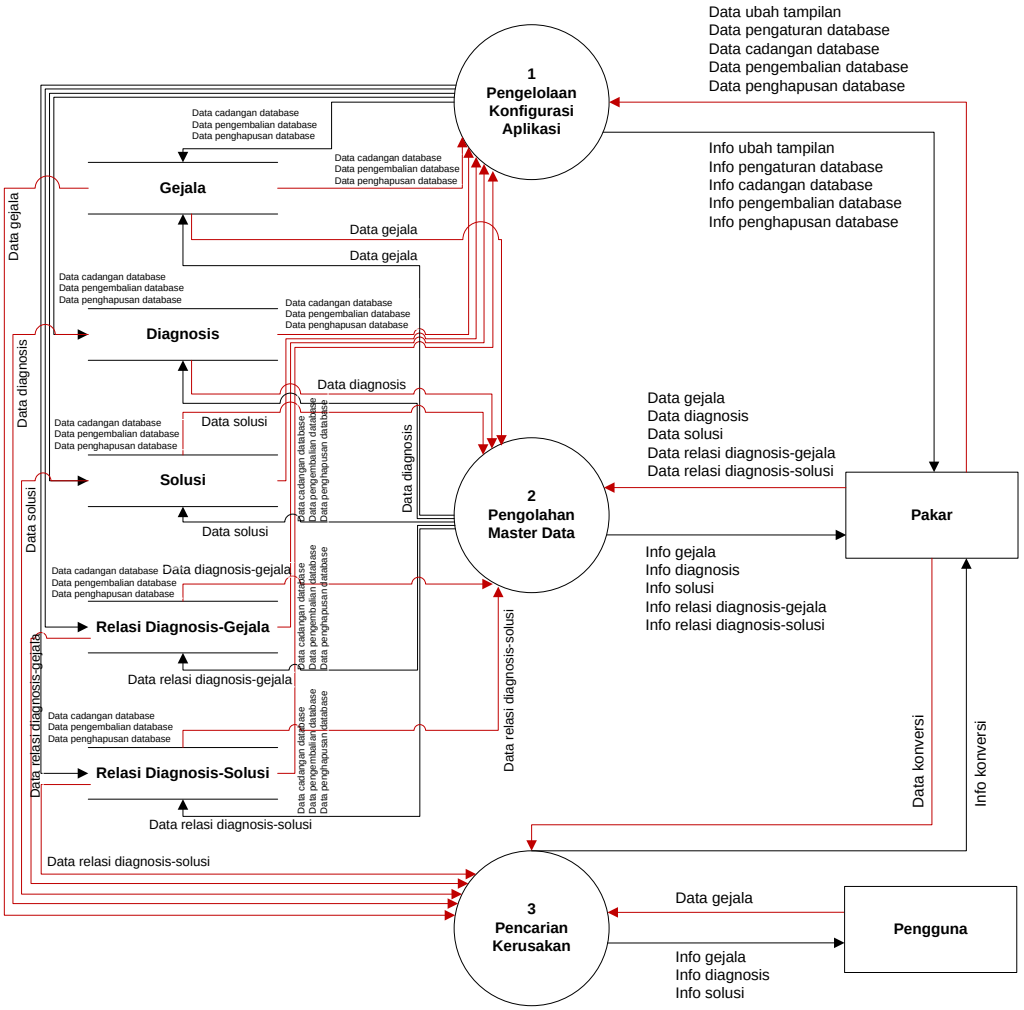
Gambar 3.4 Diagram Konteks Sistem Pakar Untuk Mendeteksi Kerusakan Yamaha Police 900 cc Menggunakan Graf Or Algorithm

3.1.8.2.3 Data Flow Diagram

Data flow diagram merupakan pemodelan sistem yang menggambarkan aliran data dalam sistem.

3.1.8.2.3.1 DFD Level 1

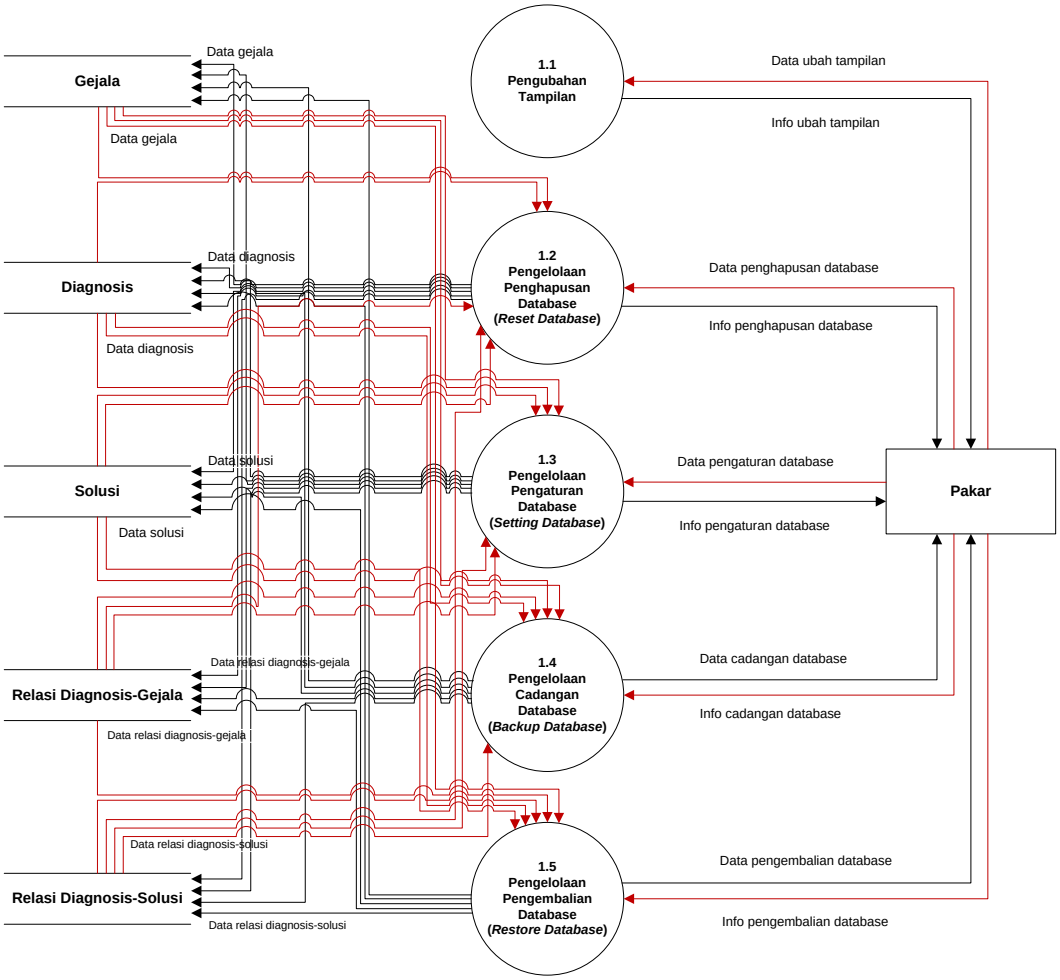
Terdapat 3 proses pada DFD Level 1 Proses. Berikut adalah DFD Level 1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.5 DFD Level 1

3.1.8.2.3.2 DFD Level 2 Proses 1

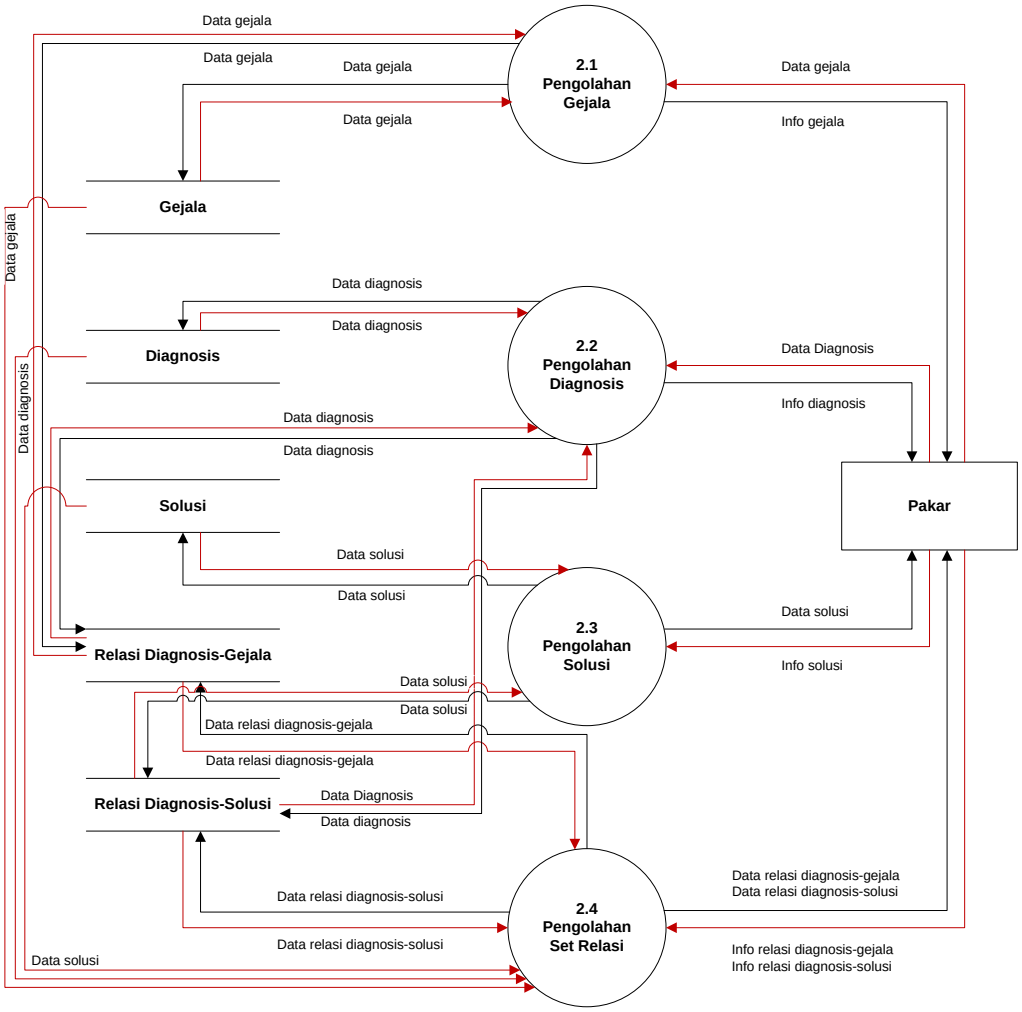
Terdapat 5 proses pada DFD Level 2 Proses 1. Berikut adalah DFD Level 2 Proses 1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.6 DFD Level 2 Proses 1

3.1.8.2.3.3 DFD Level 2 Proses 2

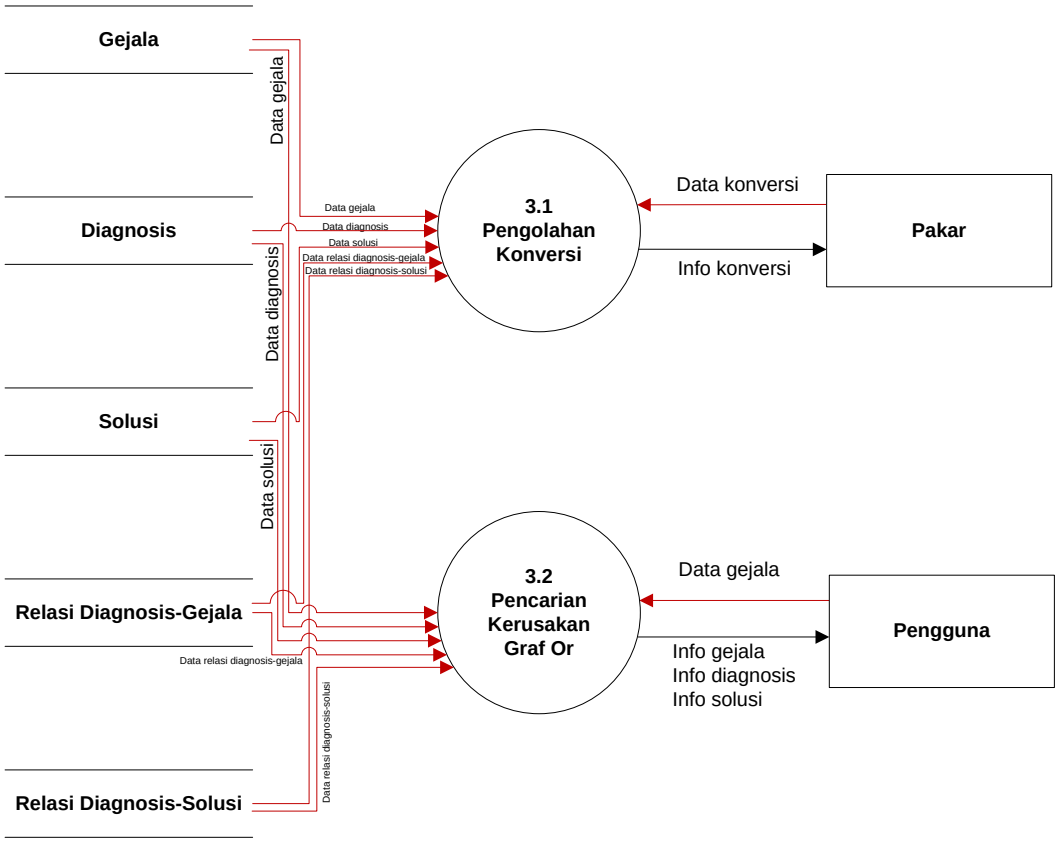
Terdapat 4 proses pada DFD Level 2 proses 2. Berikut adalah DFD Level 2 Proses 2 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.7 DFD Level 2 Proses 2

3.1.8.2.3.4 DFD Level 2 Proses 3

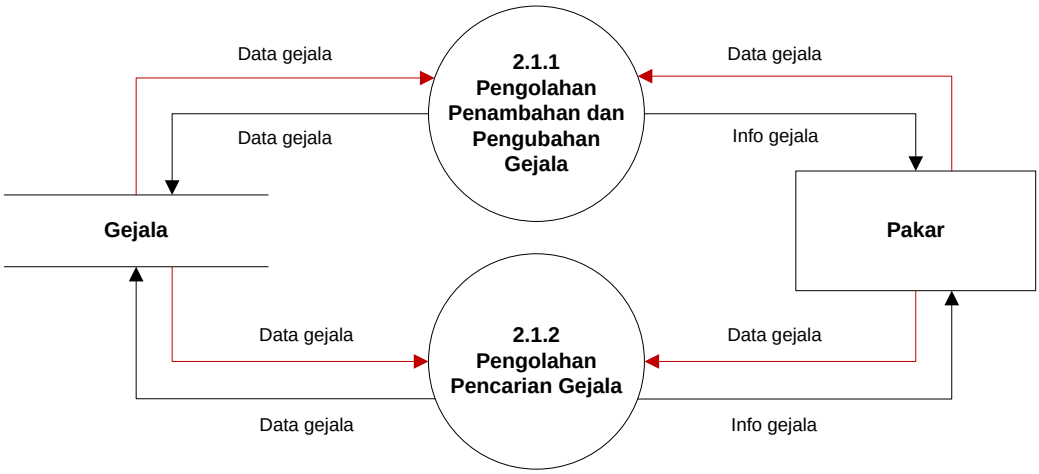
Terdapat 2 proses pada DFD Level 2 proses 3. Berikut adalah DFD Level 2 Proses 3 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.8 DFD Level 2 Proses 3

3.1.8.2.3.5 DFD Level 3 Proses 2.1

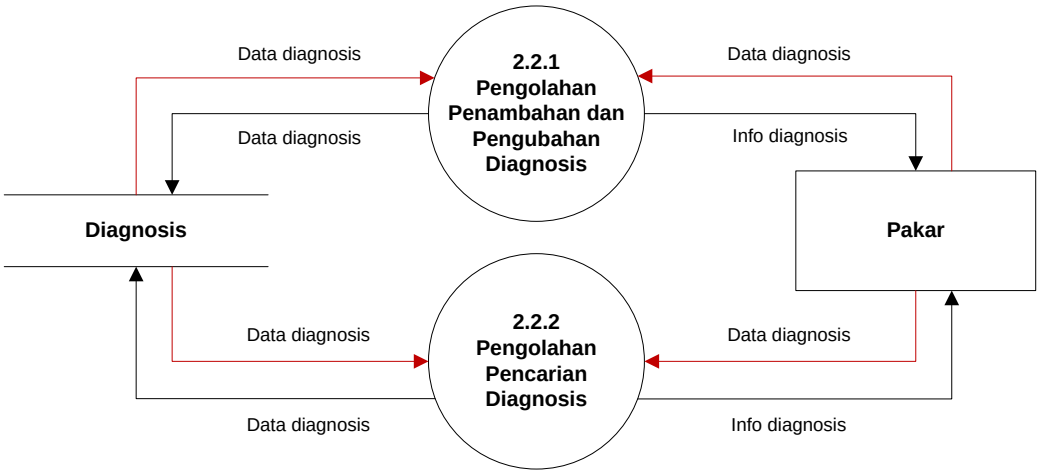
Terdapat 2 proses pada DFD Level 3 proses 2.1. Berikut adalah DFD Level 3 Proses 2.1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.9 DFD Level 3 Proses 2.1

3.1.8.2.3.6 DFD Level 3 Proses 2.2

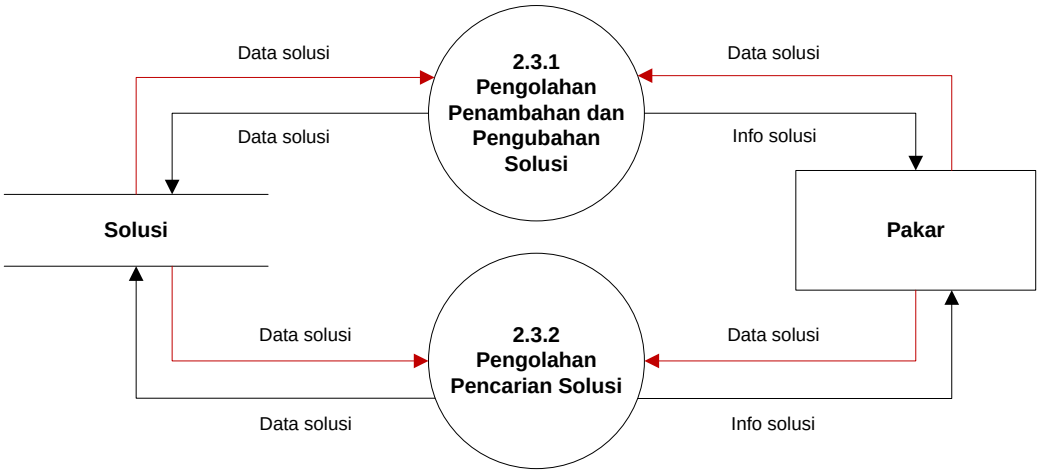
Terdapat 2 proses pada DFD Level 3 proses 2.2. Berikut adalah DFD Level 3 Proses 2.2 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.10 DFD Level 3 Proses 2.2

3.1.8.2.3.7 DFD Level 3 Proses 2.3

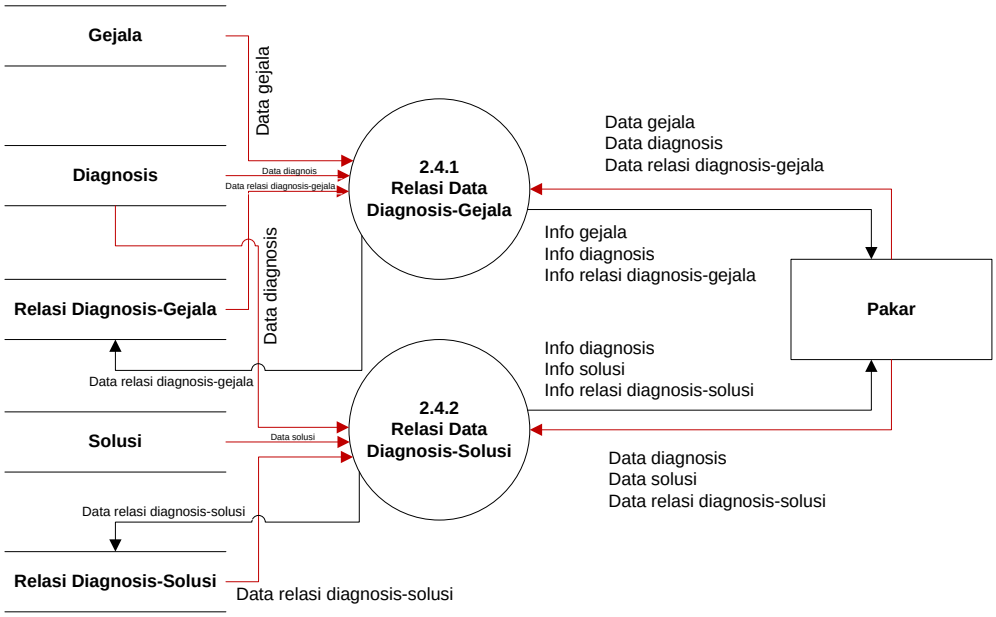
Terdapat 2 proses pada DFD Level 3 proses 2.3. Berikut adalah DFD Level 3 Proses 2.3 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.11 DFD Level 3 Proses 2.3

3.1.8.2.3.8 DFD Level 3 Proses 2.4

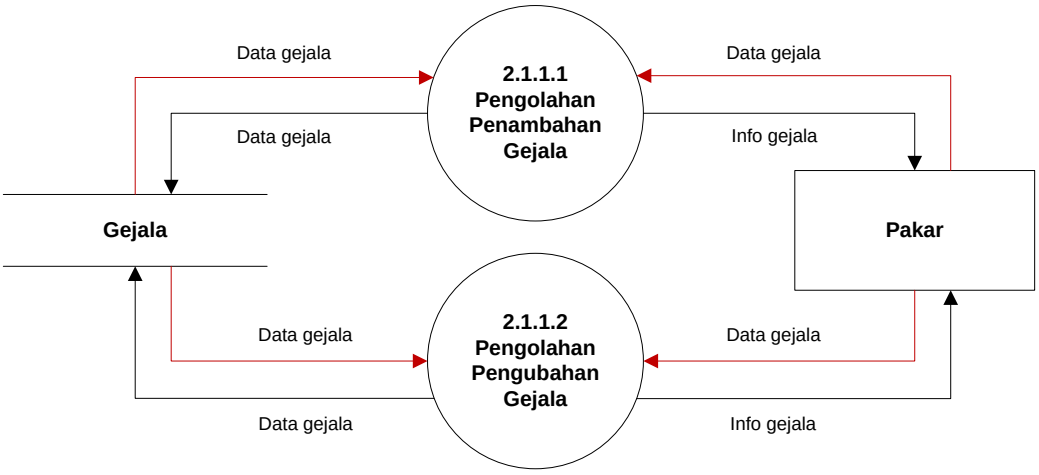
Terdapat 2 proses pada DFD Level 3 proses 2.4. Berikut adalah DFD Level 3 Proses 2.4 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.12 DFD Level 3 Proses 2.4

3.1.8.2.3.9 DFD Level 4 Proses 2.1.1

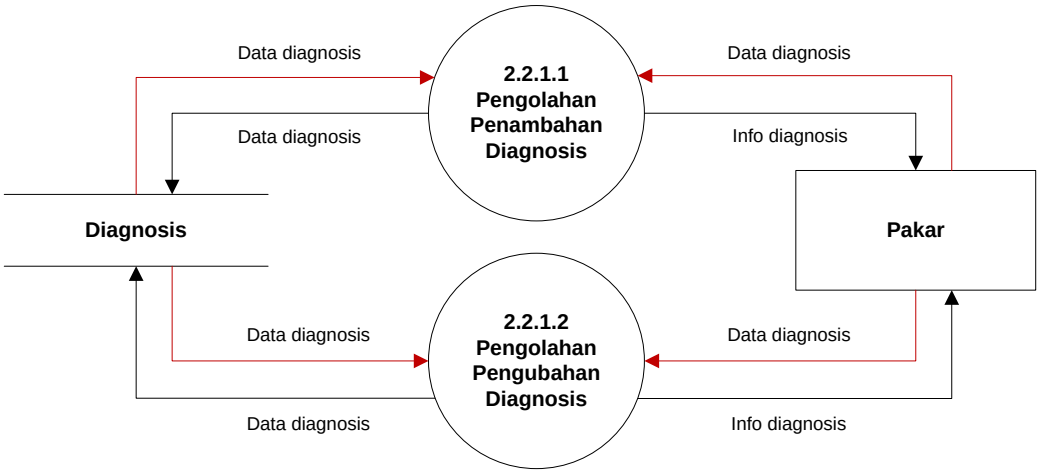
Terdapat 2 proses pada DFD Level 4 proses 2.1.1. Berikut adalah DFD Level 4 Proses 2.1.1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.13 DFD Level 4 Proses 2.1.1

3.1.8.2.3.10 DFD Level 4 Proses 2.2.1

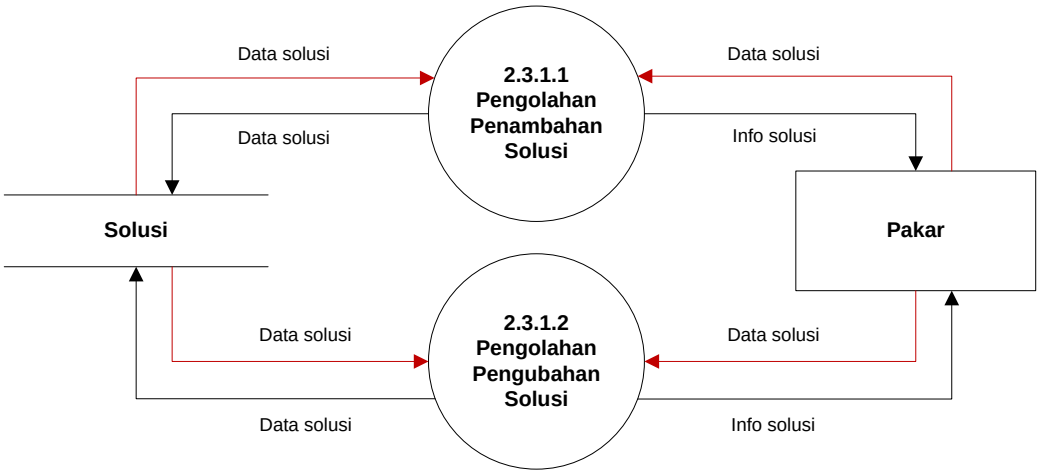
Terdapat 2 proses pada DFD Level 4 proses 2.2.1. Berikut adalah DFD Level 4 Proses 2.2.1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.14 DFD Level 4 Proses 2.2.1

3.1.8.2.3.11 DFD Level 4 Proses 2.3.1

Terdapat 2 proses pada DFD Level 4 proses 2.3.1. Berikut adalah DFD Level 4 Proses 2.3.1 dari aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 yang akan dibangun:



Gambar 3.15 DFD Level 4 Proses 2.3.1

3.1.8.2.4 Spesifikasi Proses

Spesifikasi proses digunakan untuk menjelaskan proses-proses yang terjadi pada *Data Flow Diagram* (DFD). Spesifikasi proses dari gambaran DFD diatas akan dijelaskan pada tabel dibawah ini :

Tabel 3.4 Spesifikasi Proses

No	Proses	Keterangan
1	No.Proses	1
	Nama	Pengelolaan Konfigurasi Aplikasi
	Source	Pakar
	Input	Data pengaturan database, data cadangan database, data pengembalian database, data penghapusan database, data ubah tampilan.
	Output	Info pengaturan database, info cadangan database, info pengembalian database, info penghapusan database, info ubah tampilan.
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan konfigurasi aplikasi maka sistem akan menampilkan: a. Proses pengubahan tampilan b. Proses pengelolaan penghapusan database (*reset database*) c. Proses pengelolaan pengaturan database

No	Proses	Keterangan
		(*setting database*) d. Proses pengelolaan cadangan database (*backup database*) e. Proses pengelolaan pengembalian database (*restore database*)
2	No.Proses	2
	Nama	Pengolahan Master Data
	Source	Pakar
	Input	Data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, data relasi diagnosis-solusi.
	Output	Info gejala, info diagnosis, info solusi, info relasi diagnosis-gejala, info relasi diagnosis-solusi.
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan master data maka sistem akan menampilkan: a. Proses pengolahan gejala b. Proses pengolahan diagnosis c. Proses pengolahan solusi d. Proses set relasi
3	No.Proses	3
	Nama	Pencarian Kerusakan
	Source	Pakar, Pengguna.
	Input	Data konversi, data gejala.
	Output	Info konversi, info gejala, info diagnosis, info solusi, data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, data relasi diagnosis-solusi.
	Destination	Pakar, Pengguna.
	Logika Proses	1. Jika pakar memilih proses pencarian kerusakan, maka sistem akan menampilkan proses pengolahan aplikasi, dimana data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, dan data relasi diagnosis-solusi akan disimpan pada proses pengolahan konversi. Proses pengolahan konversi mengkonversikan data yang ada kedalam format file yang dapat di compile oleh prolog. 2. Jika pengguna memilih proses pencarian kerusakan, maka sistem akan menampilkan proses pencarian kerusakan graf or, dimana pengguna hanya memasukan data gejala berupa kondisi ya atau tidak (tombol y/t pada keyboard) hingga menampilkan sebuah kerusakan dan solusi.
4	No.Proses	1.1
	Nama	Pengubahan Tampilan
	Source	Pakar

No	Proses	Keterangan
	Input	Data ubah tampilan
	Output	Info ubah tampilan
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses ubah tampilan, maka sistem akan menampilkan *field* background, latar form, dan font 2. Pakar memasukan data ubah tampilan yang berisi data background, data latar form, dan data font 3. Jika pakar memilih simpan data ubah tampilan, maka proses ubah tampilan berhasil dan sistem akan menampilkan info ubah tampilan berhasil 4. Jika pakar memilih batal simpan data ubah tampilan, maka proses ubah tampilan gagal
5	No.Proses	1.2
	Nama	Pengelolaan Penghapusan Database (*Reset Database*)
	Source	Pakar
	Input	Data penghapusan database
	Output	Info penghapusan database
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan penghapusan database (*reset database*), maka sistem akan menampilkan data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, data relasi untuk dihapus (*reset*) 2. Pakar dapat memilih data mana saja yang akan dihapus (*reset*) 3. Jika pakar memilih tombol reset data, maka data yang dipilih akan dihapus/direset dan sistem akan menampilkan info penghapusan data berhasil 4. Jika pakar memilih tombol batal reset data, maka data yang dipilih tidak akan dihapus/direset dan sistem akan menampilkan info penghapusan data gagal
6	No.Proses	1.3
	Nama	Pengelolaan Pengaturan Database (*Setting Database*)
	Source	Pakar
	Input	Data pengaturan database
	Output	Info pengaturan database
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan pengaturan database (*setting database*) maka sistem akan menampilkan *field* pengaturan database, yang terdiri dari ip server, nama server, kata sandi, dan nama database

No	Proses	Keterangan
		2. Pakar memasukan data pengaturan database 3. Jika pakar memilih simpan data pengaturan database maka proses pengelolaan pengaturan database (*setting database*) berhasil dan sistem akan menampilkan info pengaturan database berhasil 4. Jika pengguna memilih batal simpan data pengaturan database maka proses pengaturan data gagal dan sistem akan menampilkan info pengaturan database gagal
7	No.Proses	1.4
	Nama	Pengelolaan Cadangan Database (*Backup Database*)
	Source	Pakar
	Input	Data cadangan database
	Output	Info cadangan database
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan cadangan database (*backup database*) maka sistem akan menampilkan *field* data cadangan database yang terdiri dari direktori database dan nama database 2. Pakar memasukan data cadangan database 3. Jika pakar memilih simpan data cadangan database maka proses pengelolaan cadangan database (*backup database*) berhasil dan sistem akan menampilkan info cadangan database berhasil 4. Jika pakar memilih batal simpan data cadangan database maka proses pengelolaan cadangan database (*backup database*) gagal dan sistem akan menampilkan info cadangan database gagal
8	No.Proses	1.5
	Nama	Pengelolaan Pengembalian Data (*Restore Database*)
	Source	Pakar
	Input	Data pengembalian database
	Output	Info pengembalian database
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan pengembalian database (*restore database*) maka sistem akan menampilkan *field* data pengembalian database yang terdiri dari direktori database dan nama database 2. Pakar memasukan data pengembalian database 3. Jika pakar memilih simpan data pengembalian database maka proses pengelolaan pengembalian database (*restore database*) berhasil dan sistem akan menampilkan info pengembalian database berhasil 4. Jika pakar memilih simpan data pengembalian database maka proses pengelolaan pengembalian

No	Proses	Keterangan
		database (*restore database*) berhasil dan sistem akan menampilkan info pengembalian database berhasil
9	No.Proses	2.1
	Nama	Pengolahan Gejala
	Source	Pakar
	Input	Data gejala
	Output	Info gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan gejala, maka sistem akan menampilkan *field* gejala dan pertanyaan 2. Pakar memasukan data gejala 3. Jika pakar memilih simpan data gejala, maka proses pengolahan gejala berhasil dan sistem akan menampilkan info penyimpanan data gejala berhasil 4. Jika pengguna memilih batal simpan data gejala, maka proses pengolahan gejala gagal dan sistem akan menampilkan info penyimpanan data gejala gagal
10	No.Proses	2.2
	Nama	Pengelolaan Diagnosis
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan diagnosis, maka sistem akan menampilkan *field* diagnosis 2. Pakar memasukan data diagnosis 3. Jika pakar memilih simpan data diagnosis, maka proses pengelolaan diagnosis berhasil dan sistem akan menampilkan info penyimpanan data diagnosis berhasil 4. Jika pakar memilih batal simpan data diagnosis, maka proses pengelolaan diagnosis gagal dan sistem akan menampilkan info penyimpanan data diagnosis gagal
11	No.Proses	2.3
	Nama	Pengelolaan Solusi
	Source	Pakar
	Input	Data solusi
	Output	Info solusi
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengelolaan solusi, maka sistem akan menampilkan *field* solusi 2. Pakar memasukan data solusi 3. Jika pakar memilih simpan data solusi, maka proses

No	Proses	Keterangan
		pengelolaan solusi berhasil dan sistem akan menampilkan info penyimpanan data solusi berhasil 4. Jika pakar memilih batal simpan data solusi, maka proses pengelolaan solusi gagal dan sistem akan menampilkan info penyimpanan data solusi gagal
12	No.Proses	2.4
	Nama	Set Relasi
	Source	Pakar
	Input	Data relasi diagnosis-gejala, data relasi diagnosis-solusi
	Output	Info relasi diagnosis-gejala, info relasi diagnosis-solusi
	Destination	Pakar
	Logika Proses	1. Jika pengguna memilih proses set relasi maka sistem akan menampilkan: a. Proses relasi data diagnosis-gejala b. Proses relasi data diagnosis-solusi 2. Saat pakar memilih proses relasi data diagnosis-gejala, maka sistem akan menampilkan tabel diagnosis untuk selanjutnya akan direlasikan ke dalam gejala 3. Saat pakar memilih proses relasi data diagnosis-solusi, maka sistem akan menampilkan tabel diagnosis untuk selanjutnya akan direlasikan ke dalam solusi
13	No.Proses	3.1
	Nama	Pengolahan Konversi
	Source	Pakar
	Input	Data konversi, data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, data relasi diagnosis-solusi.
	Output	Info konversi
	Destination	Pakar
	Logika Proses	1. Saat pakar memilih proses pengolahan konversi, maka data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, dan data relasi diagnosis-solusi akan ditampilkan pada *field* konversi 2. Jika pakar memilih tombol simpan konversi, maka data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, dan data relasi diagnosis-solusi. akan disimpan dalam format *.PRO
14	No.Proses	3.2
	Nama	Pencarian Kerusakan Graf Or
	Source	Pengguna
	Input	Data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, data relasi diagnosis-solusi.
	Output	Info gejala, info diagnosis, info solusi

No	Proses	Keterangan
	Destination	Pengguna
	Logika Proses	1. Jika pengguna memilih proses pencarian kerusakan graf or, maka pengguna dihadapkan pada sebuah pertanyaan dengan kondisi *boolean* (ya/tidak), untuk selanjutnya tiap kondisi akan dilanjutkan dengan info pertanyaan dan akan berakhir pada sebuah info diagnosis dan info solusi 2. Ketika berakhir pada sebuah info diagnosis dan info solusi, maka pengguna memilih ingin melanjutkan kembali pencarian atau tidak 3. Jika pengguna memilih 'ia', sistem pencarian diulang kembali dari awal 4. Jika pengguna memilih 'tidak', sistem pencarian tidak akan diulang
15	No.Proses	2.1.1
	Nama	Pengolahan Penambahan dan Pengubahan Gejala
	Source	Pakar
	Input	Data gejala
	Output	Info gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses penambahan dan pengubahan gejala, maka sistem akan menampilkan: a. Proses pengolahan penambahan gejala b. Proses pengolahan pengubahan gejala
16	No.Proses	2.1.2
	Nama	Pengolahan Pencarian Gejala
	Source	Pakar
	Input	Data gejala
	Output	Info gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan pencarian gejala, maka sistem akan menampilkan *field* pencarian gejala dan tabel gejala yang akan dicari 2. Pakar memasukan data gejala yang dicari 3. Jika data yang dicari ditemukan, maka sistem akan menampilkan info gejala ditemukan 4. Jika data yang dicari tidak ditemukan, maka sistem akan menampilkan info gejala tidak ditemukan
17	No.Proses	2.2.1
	Nama	Pengolahan Penambahan dan Pengubahan Diagnosis
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis
	Destination	Pakar

No	Proses	Keterangan
	Logika Proses	1. Jika pakar memilih proses penambahan dan pengubahan diagnosis, maka sistem akan menampilkan: a. Proses pengolahan penambahan diagnosis b. Proses pengolahan pengubahan diagnosis
18	No.Proses	2.2.2
	Nama	Pengolahan Pencarian Diagnosis
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pencarian diagnosis, maka sistem akan menampilkan *field* pencarian diagnosis dan tabel diagnosis yang akan dicari. 2. Pakar memasukan data diagnosis yang dicari 3. Jika data yang dicari ditemukan, maka sistem akan menampilkan info diagnosis ditemukan 4. Jika data yang dicari tidak ditemukan, maka sistem akan menampilkan info diagnosis tidak ditemukan
19	No.Proses	2.3.1
	Nama	Pengolahan Penambahan dan Pengubahan Solusi
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses penambahan dan pengubahan solusi, maka sistem akan menampilkan: a. Proses pengolahan penambahan solusi b. Proses pengolahan pengubahan solusi
20	No.Proses	2.3.2
	Nama	Pengolahan Pencarian Solusi
	Source	Pakar
	Input	Data solusi
	Output	Info solusi
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan pencarian solusi, maka sistem akan menampilkan *field* pencarian solusi dan tabel solusi yang akan dicari. 2. Pakar memasukan data solusi yang dicari 3. Jika data yang dicari ditemukan, maka sistem akan menampilkan info solusi ditemukan 4. Jika data yang dicari tidak ditemukan, maka sistem akan menampilkan info solusi tidak ditemukan
21	No.Proses	2.4.1

No	Proses	Keterangan
	Nama	Relasi Data Diagnosis-Gejala
	Source	Pakar
	Input	Data gejala, data diagnosis, data relasi diagnosis-gejala
	Output	Info gejala, info diagnosis, info relasi diagnosis-gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses relasi data diagnosis-gejala, maka sistem akan menampilkan tabel diagnosis 2. Jika pakar sudah memilih diagnosis maka akan ditampilkan *form* tambah set relasi diagnosis-gejala 3. Jika pakar memilih simpan data relasi diagnosis-gejala, maka sistem akan menampilkan info penyimpanan data relasi diagnosis-gejala berhasil 4. Jika pakar memilih batal simpan data relasi diagnosis-gejala, maka sistem tidak akan menyimpan data relasi diagnosis-gejala dan sistem akan menampilkan info penyimpanan data relasi diagnosis-gejala gagal
22	No.Proses	2.4.2
	Nama	Relasi Data Diagnosis-Solusi
	Source	Pakar
	Input	Data solusi, data diagnosis, data relasi diagnosis-solusi
	Output	Info solusi, info diagnosis, info relasi diagnosis-solusi
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses relasi data diagnosis-solusi, maka sistem akan menampilkan tabel diagnosis 2. Jika pakar sudah memilih diagnosis maka akan ditampilkan *form* tambah set relasi diagnosis-solusi 3. Jika pakar memilih simpan data relasi diagnosis-solusi, maka sistem akan menampilkan info penyimpanan data relasi diagnosis-solusi berhasil 4. Jika pakar memilih batal simpan data relasi diagnosis-solusi, maka sistem tidak akan menyimpan data relasi diagnosis-solusi dan sistem akan menampilkan info penyimpanan data relasi diagnosis-solusi gagal
23	No.Proses	3.2.1
	Nama	Initial State
	Source	Pengguna
	Input	Data gejala
	Output	Info gejala, info diagnosis, info solusi, info relasi diagnosis-gejala, info relasi diagnosis-solusi
	Destination	Pengguna

No	Proses	Keterangan
	Logika Proses	1. Saat pengguna memasukan data gejala, sistem akan menampilkan info gejala berupa pertanyaan 2. Kemudian akan berlanjut ke pertanyaan berikutnya sesuai dengan pencarian graf or (melanjutkan ke current state)
24	No.Proses	2.1.1.1
	Nama	Pengolahan Penambahan Gejala
	Source	Pakar
	Input	Data gejala
	Output	Info gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan penambahan gejala, maka sistem akan menampilkan *field* gejala dan pertanyaan 2. Pengguna memasukan data gejala 3. Jika pengguna memilih simpan data gejala, maka sistem akan menampilkan info penyimpanan data gejala berhasil 4. Jika pengguna memilih batal simpan data gejala, maka sistem tidak akan menyimpan data gejala dan sistem akan menampilkan info penyimpanan data gejala gagal
25	No.Proses	2.1.1.2
	Nama	Pengolahan Pengubahan Gejala
	Source	Pakar
	Input	Data gejala
	Output	Info gejala
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengubahan data gejala, maka sistem akan menampilkan *field* gejala dan pertanyaan 2. Pakar mengubah data gejala 3. Jika pakar memilih simpan data gejala, maka sistem akan menampilkan info penyimpanan data gejala berhasil 4. Jika pakar memilih batal simpan data gejala, maka sistem tidak akan menyimpan data gejala dan sistem akan menampilkan info penyimpanan data gejala gagal
26	No.Proses	2.2.1.1
	Nama	Pengolahan Penambahan Diagnosis
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis

No	Proses	Keterangan
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan penambahan diagnosis, maka sistem akan menampilkan *field* diagnosis 2. Pengguna memasukan data diagnosis 3. Jika pengguna memilih simpan data diagnosis, maka sistem akan menampilkan info penyimpanan data diagnosis berhasil 4. Jika pengguna memilih batal simpan data diagnosis, maka sistem tidak akan menyimpan data diagnosis dan sistem akan menampilkan info penyimpanan data diagnosis gagal
27	No.Proses	2.2.1.2
	Nama	Pengolahan Pengubahan Diagnosis
	Source	Pakar
	Input	Data diagnosis
	Output	Info diagnosis
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan pengubahan diagnosis, maka sistem akan menampilkan *field* diagnosis 2. Pakar mengubah data diagnosis 3. Jika pakar memilih simpan data diagnosis, maka sistem akan menampilkan info penyimpanan data diagnosis berhasil 4. Jika pakar memilih batal simpan data diagnosis, maka sistem tidak akan menyimpan data diagnosis dan sistem akan menampilkan info penyimpanan data diagnosis gagal
28	No.Proses	2.3.1.1
	Nama	Pengolahan Penambahan Solusi
	Source	Pakar
	Input	Data solusi
	Output	Info solusi
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses tambah solusi, maka sistem akan menampilkan *field* solusi 2. Pakar memasukan data solusi 3. Jika pakar memilih simpan data solusi, maka sistem akan menampilkan info penyimpanan data solusi berhasil 4. Jika pakar memilih batal simpan data solusi, maka sistem tidak akan menyimpan data solusi dan sistem akan menampilkan info penyimpanan data solusi

No	Proses	Keterangan
		gagal
29	No.Proses	2.3.1.2
	Nama	Pengolahan Pengubahan Solusi
	Source	Pakar
	Input	Data solusi
	Output	Info solusi
	Destination	Pakar
	Logika Proses	1. Jika pakar memilih proses pengolahan pengubahan solusi, maka sistem akan menampilkan *field* solusi 2. Pakar mengubah data solusi 3. Jika pakar memilih simpan data solusi, maka sistem akan menampilkan info penyimpanan data solusi berhasil 4. Jika pakar memilih batal simpan data solusi, maka sistem tidak akan menyimpan data solusi dan sistem akan menampilkan info penyimpanan data solusi gagal

3.1.8.2.5 Kamus Data

Kamus data (*data dictionary*) merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem. Kamus data dibuat berdasarkan arus data yang ada pada DFD. Kamus data untuk DFD pada aplikasi sistem pakar untuk mendeteksi kerusakan Yamaha P 900 akan dijelaskan pada tabel berikut:

Tabel 3.5 Kamus Data

Nama aliran data	Data Gejala
Digunakan pada	Pakar-Proses 1 Pakar-Proses 2 Pakar-Proses 3 Pakar-Proses 1.2 Pakar-Proses 1.3 Pakar-Proses 1.4 Pakar-Proses 1.5 Pakar-Proses 2.1 Pakar-Proses 3.1 Pengguna-Proses 3.2 Pakar-Proses 2.1.1

	Pakar-Proses 2.1.2 Pakar-Proses 2.4.1 Pakar-Proses 2.1.1.1 Pakar-Proses 2.1.1.2
Deskripsi	Berisi data gejala
Struktur data	kode_gejala+nama_gejala+pertanyaan
kode_gejala nama_gejala pertanyaan	[A-Z|0-9] [A-Z|0-9] [A-Z|0-9]
Nama aliran data	Data Diagnosis
Digunakan pada	Pakar-Proses 1 Pakar-Proses 2 Pakar-Proses 3 Pakar-Proses 1.2 Pakar-Proses 1.3 Pakar-Proses 1.4 Pakar-Proses 1.5 Pakar-Proses 2.2 Pakar-Proses 3.1 Pengguna-Proses 3.2 Pakar-Proses 2.2.1 Pakar-Proses 2.2.2 Pakar-Proses 2.4.1 Pakar-Proses 2.4.2 Pakar-Proses 2.2.1.1 Pakar-Proses 2.2.1.2
Deskripsi	Berisi data diagnosis
Struktur data	kode_diagnosis+nama_diagnosis
kode_diagnosis nama_diagnosis	[A-Z|0-9] [A-Z|0-9]
Nama aliran data	Data Solusi
Digunakan pada	Pakar-Proses 1 Pakar-Proses 2 Pakar-Proses 3 Pakar-Proses 1.2 Pakar-Proses 1.3 Pakar-Proses 1.4 Pakar-Proses 1.5 Pakar-Proses 2.3 Pakar-Proses 3.1 Pengguna-Proses 3.2 Pakar-Proses 2.3.1 Pakar-Proses 2.3.2 Pakar-Proses 2.4.2 Pakar-Proses 2.3.1.1 Pakar-Proses 2.3.1.2

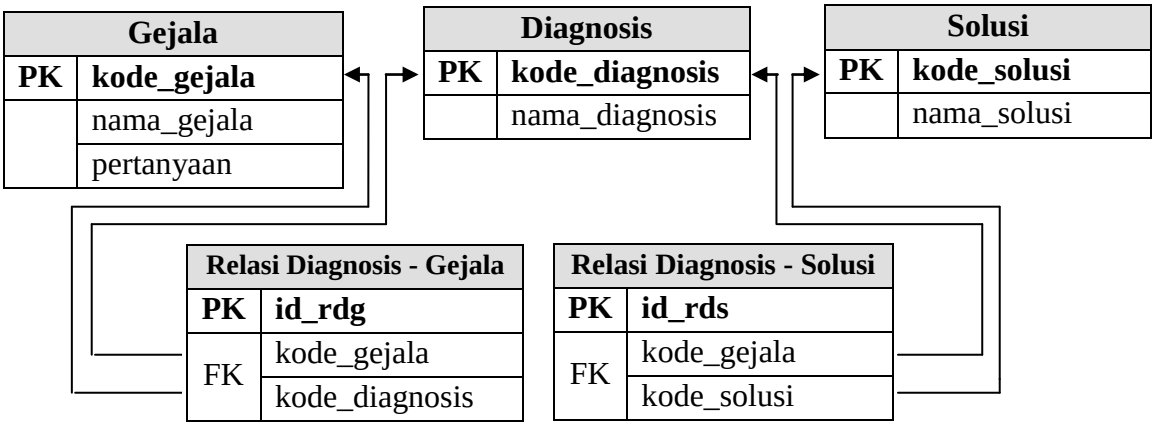
Deskripsi	Berisi data solusi
Struktur data	kode_solusi+nama_solusi
kode_solusi nama_solusi	[A-Z|0-9] [A-Z|0-9]
Nama aliran data	Data Relasi Diagnosis-Gejala
Digunakan pada	Pakar-Proses 1 Pakar-Proses 2 Pakar-Proses 3 Pakar-Proses 1.2 Pakar-Proses 1.3 Pakar-Proses 1.4 Pakar-Proses 1.5 Pakar-Proses 2.4 Pakar-Proses 3.1 Pengguna-Proses 3.2 Pakar-Proses 2.4.1
Deskripsi	Berisi data relasi diagnosis-gejala
Struktur data	id_rdg
id_rdg	[0-9]
Nama aliran data	Data Relasi Diagnosis-Solusi
Digunakan pada	Pakar-Proses 1 Pakar-Proses 2 Pakar-Proses 3 Pakar-Proses 1.2 Pakar-Proses 1.3 Pakar-Proses 1.4 Pakar-Proses 1.5 Pakar-Proses 2.4 Pakar-Proses 3.1 Pengguna-Proses 3.2 Pakar-Proses 2.4.2
Deskripsi	Berisi data relasi diagnosis-solusi
Struktur data	id_rds
id_rds	[0-9]

3.1.9 Analisis Basis Data

Analisis Basis Data adalah menciptakan atau merancang kumpulan data yang terhubung dan tersimpan secara bersama-sama.

3.1.9.1 Skema Relasi

Berikut adalah skema relasinya. Pada ke lima tabel terdapat masing-masing *primary key*, yaitu kode_gejala (tabel gejala), kode_diagnosis (tabel diagnosis), kode_solusi (tabel solusi), id_rdg (tabel relasi diagnosis - gejala), id_rds (tabel relasi diagnosis - solusi).



Gambar 3.16 Perancangan Skema Relasi Aplikasi Sistem Pakar

3.1.9.2 Struktur Tabel

Struktur tabel yang terdapat pada aplikasi sistem pakar yang dibangun ada 5 tabel yaitu tabel gejala, tabel diagnosis, tabel solusi, tabel relasi diagnosis-gejala, dan tabel relasi diagnosis–solusi.

Tabel 3.6 File Gejala

Nama Field	Type	Size	Ket.
kode_gejala	Varchar	6	Primary Key
nama_gejala	Varchar	255	-
pertanyaan	Varchar	255	-

Tabel 3.7 File Diagnosis

Nama Field	Type	Size	Ket.
kode_diagnosis	Varchar	6	Primary Key
nama_diagnosis	Varchar	255	-

Tabel 3.8 File Solusi

Nama Field	Type	Size	Ket.
kode_solusi	Varchar	6	Primary Key
nama_solusi	Varchar	255	-

Tabel 3.9 Relasi Diagnosis - Gejala

Nama Field	Type	Size	Ket.
id_rdg	Int	3	Primary Key
kode_diagnosis	Varchar	6	Foreign Key
kode_gejala	Varchar	6	Foreign Key

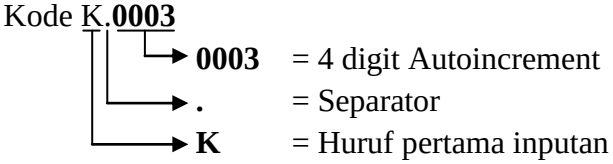
Tabel 3.10 Relasi Diagnosis - Solusi

Nama Field	Type	Size	Ket.
id_rds	Int	3	Primary Key
kode_diagnosis	Varchar	6	Foreign Key
kode_solusi	Varchar	6	Foreign Key

3.1.9.3 Perancangan Pengkodean

Perancangan pengkodean yang terdapat pada Aplikasi Pendeteksi Kerusakan Motor Yamaha P 900 adalah sebagai berikut :

Pengkodean untuk Gejala, Diagnosis, dan Solusi

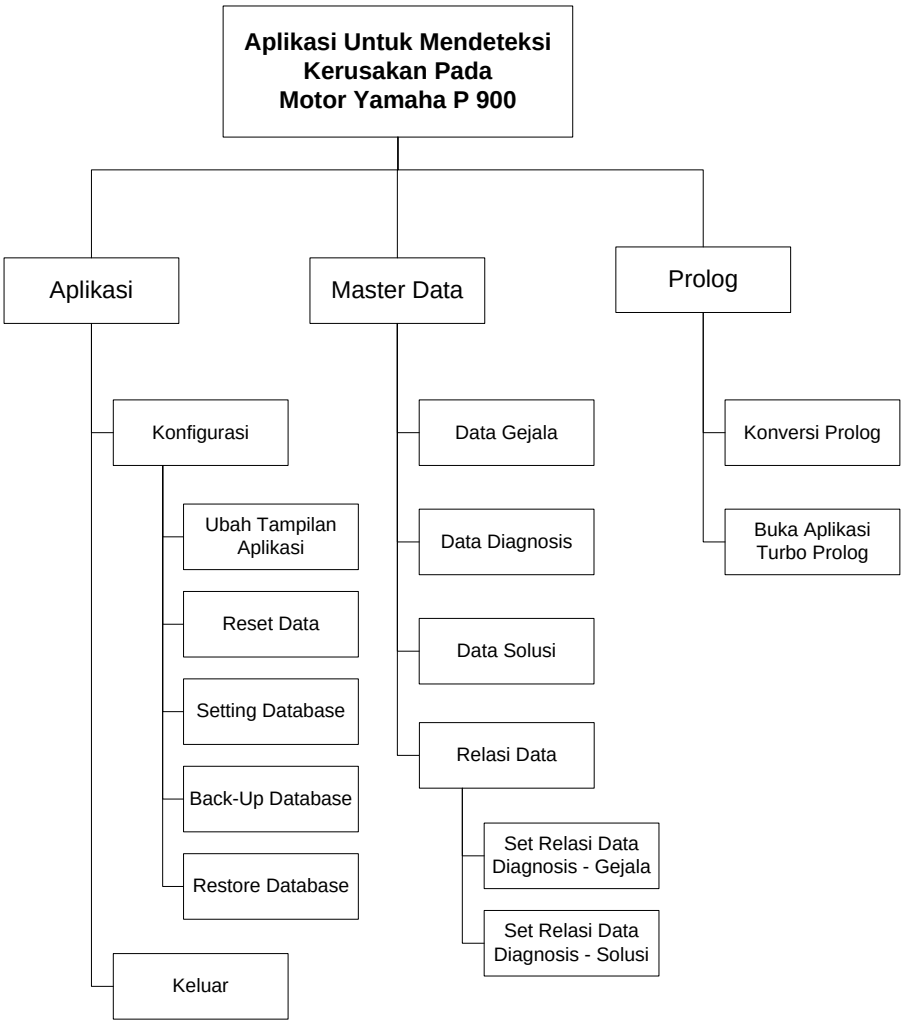


3.2 Perancangan Sistem

Perancangan struktur menu berisikan menu dan submenu yang berfungsi memudahkan user didalam menggunakan sistem dari aplikasi yang akan dibangun.

3.2.1 Perancangan Struktur Menu

Aplikasi yang ditelah dirancang memiliki 3 menu. Berikut adalah gambar bagian-bagian dari menu-menu yang disajikan dalam aplikasi.



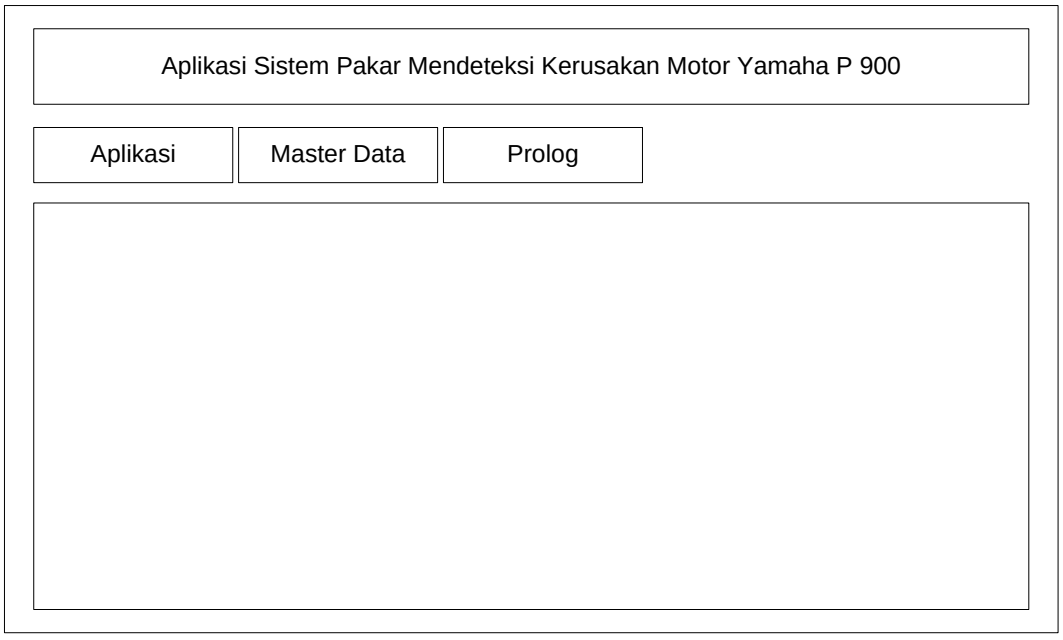
Gambar 3.17 Struktur Menu Aplikasi Sistem Pakar

3.2.2 Perancangan Antar Muka

Perancangan antar muka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam pengimplementasian aplikasi.

3.2.2.1 Perancangan Antar Muka Awal

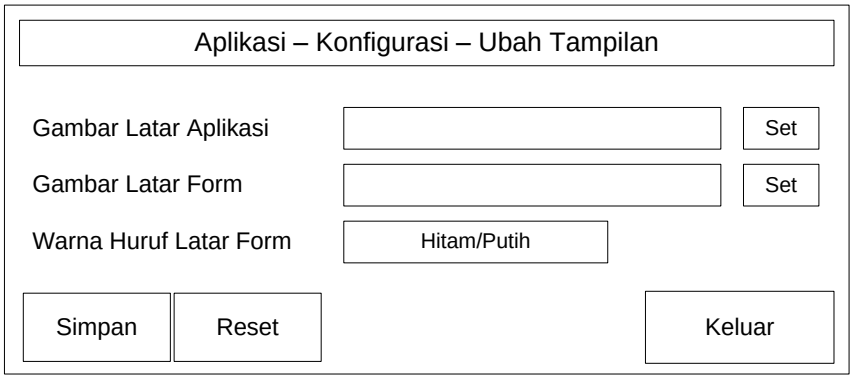
Tampilan awal aplikasi, dimana hanya terdapat 3 menu, yaitu; Menu Aplikasi (Submenu: Konfigurasi Aplikasi dan Keluar Aplikasi), Menu Master Data (Submenu: Data Gejala, Data Diagnosis Kerusakan, Data Solusi, dan Relasi Data) dan Menu Prolog (Submenu: Konversi Prolog dan Buka Aplikasi Turbo Prolog)



Gambar 3.18 Perancangan Antar Muka Awal

3.2.2.2 Perancangan Antar Muka Tampilan Aplikasi

Perancangan tampilan form untuk mengubah latar aplikasi, latar form, dan warna huruf pada aplikasi.



Gambar 3.19 Perancangan Antar Muka Ubah Tampilan

3.2.2.3 Perancangan Antar Muka Reset Data

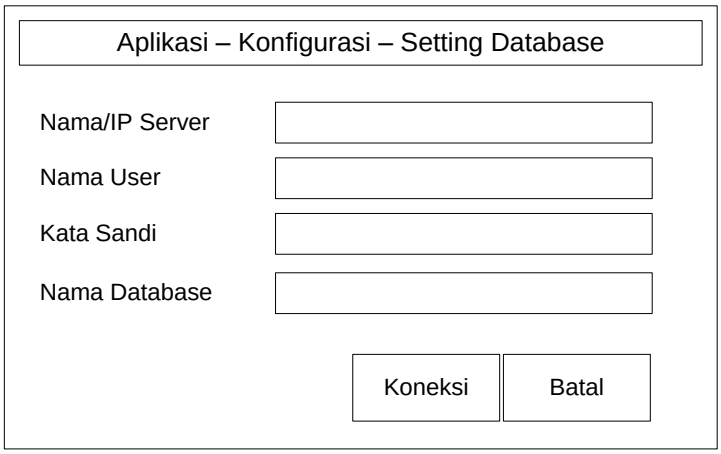
Perancangan tampilan form untuk menghapus data gejala, data diagnosis, data solusi, data relasi diagnosis-gejala, dan data relasi diagnosis-solusi yang terdapat pada master data.



Gambar 3.20 Perancangan Antar Muka Reset Data

3.2.2.4 Perancangan Antar Muka Setting Database

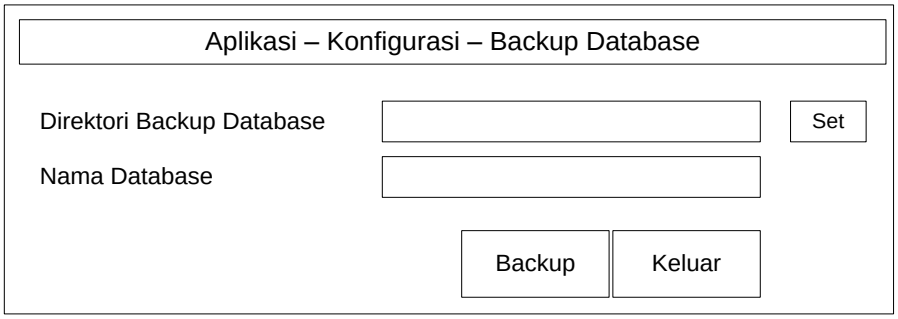
Perancangan tampilan form untuk mengkoneksikan aplikasi ke database sehingga data pada Master Data dapat diakses.



Gambar 3.21 Perancangan Antar Muka Setting Database

3.2.2.5 Perancangan Antar Muka Backup Database

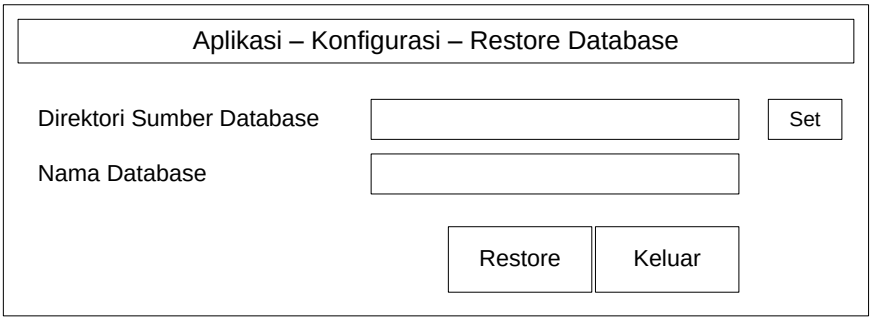
Perancangan tampilan form untuk membackup database yang telah dibuat sebelumnya agar sewaktu-waktu dapat diakses kembali.



Gambar 3.22 Perancangan Antar Muka Backup Database

3.2.2.6 Perancangan Antar Muka Restore Database

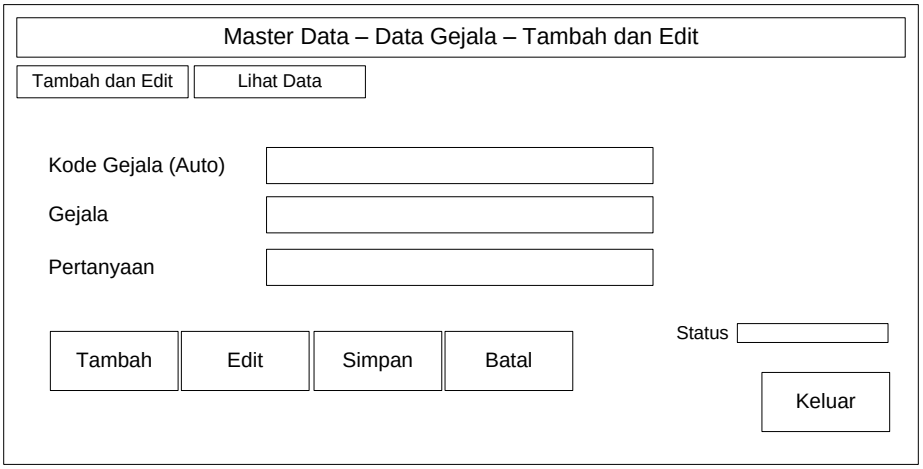
Perancangan tampilan form untuk mengembalikan database yang sebelumnya telah dibackup.



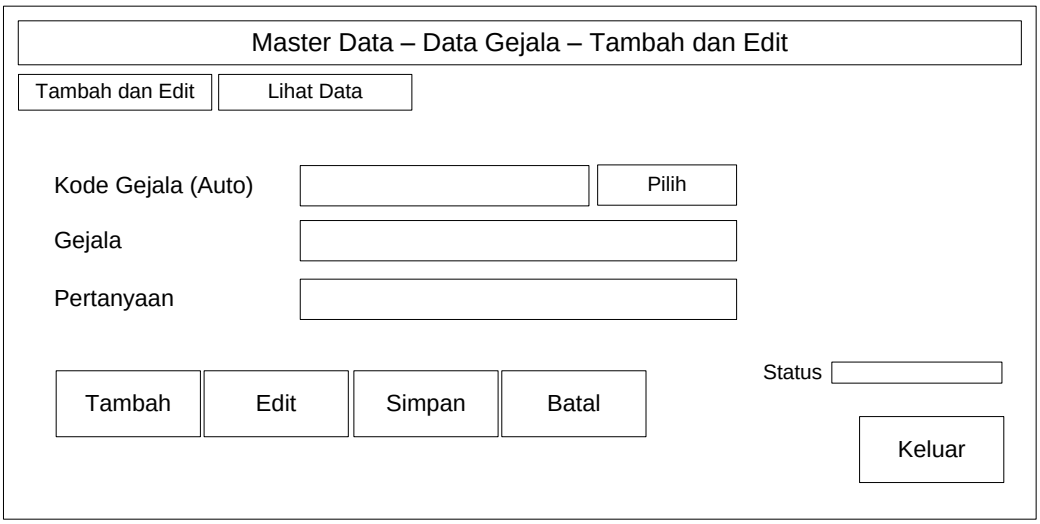
Gambar 3.23 Perancangan Antar Muka Restore Database

3.2.2.7 Perancangan Perbaikan Basis Pengetahuan, Aturan dan Fakta

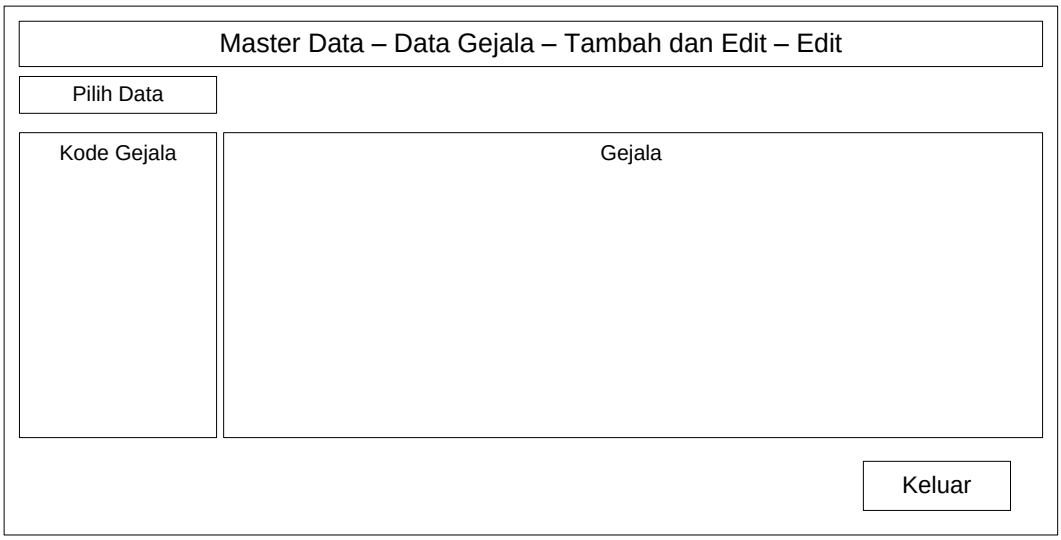
Berikut adalah tampilan Perbaikan Basis Pengetahuan, Aturan dan Fakta yang merupakan *update sistem* (terdapat dalam menu Master Data). Berikut adalah perbaikan Basis Pengetahuan, Aturan dan Fakta berdasarkan *Gejala*, *Diagnosis*, dan *solusi* serta *Relasi Data*.



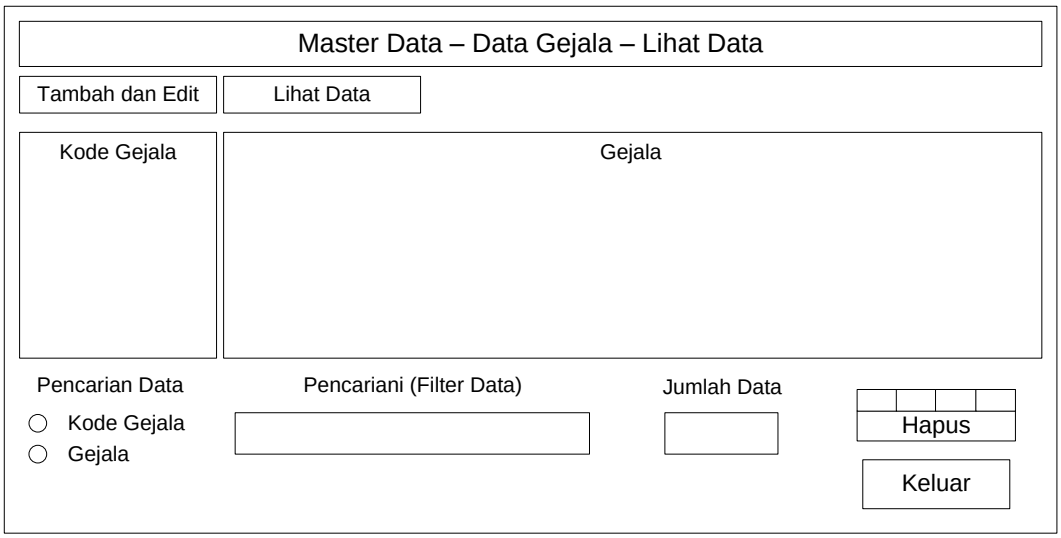
Gambar 3.24 Perancangan Tampilan Perbaikan Basis Pengetahuan, Aturan dan Fakta Aplikasi Berdasarkan Data Gejala



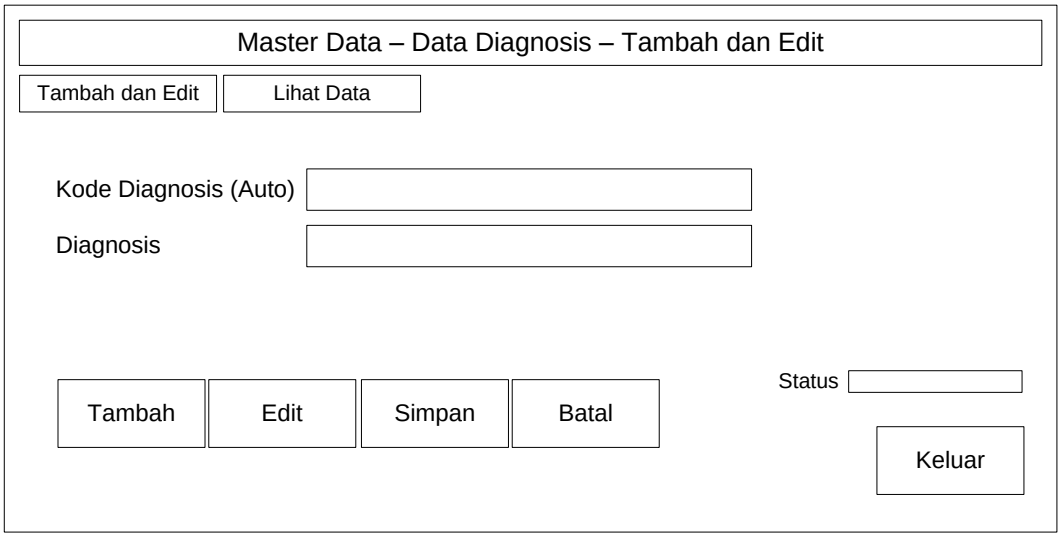
Gambar 3.25 Perancangan Tampilan Edit Data pada Data Gejala



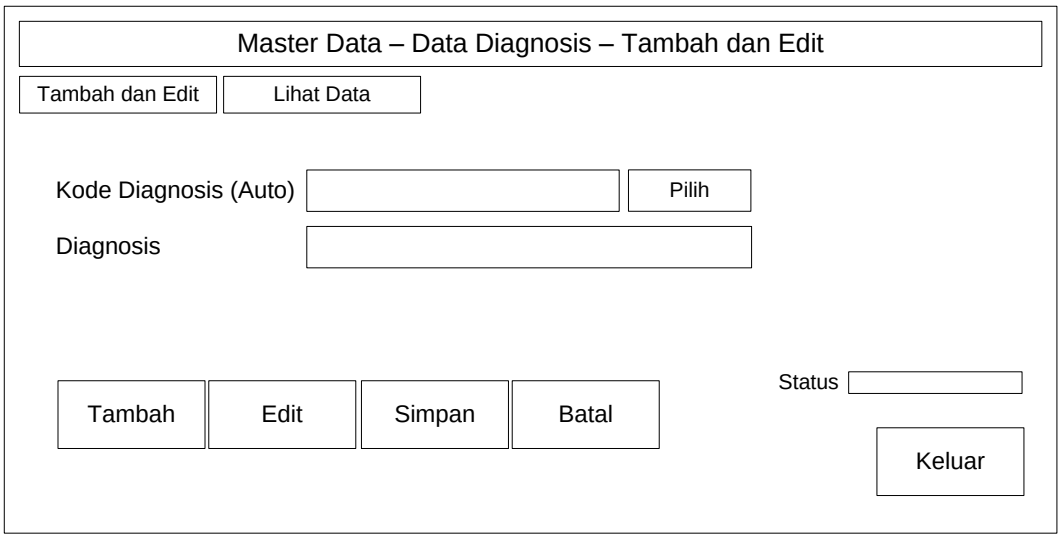
Gambar 3.26 Perancangan Tampilan Edit Data pada Tombol Pilih Data Gejala



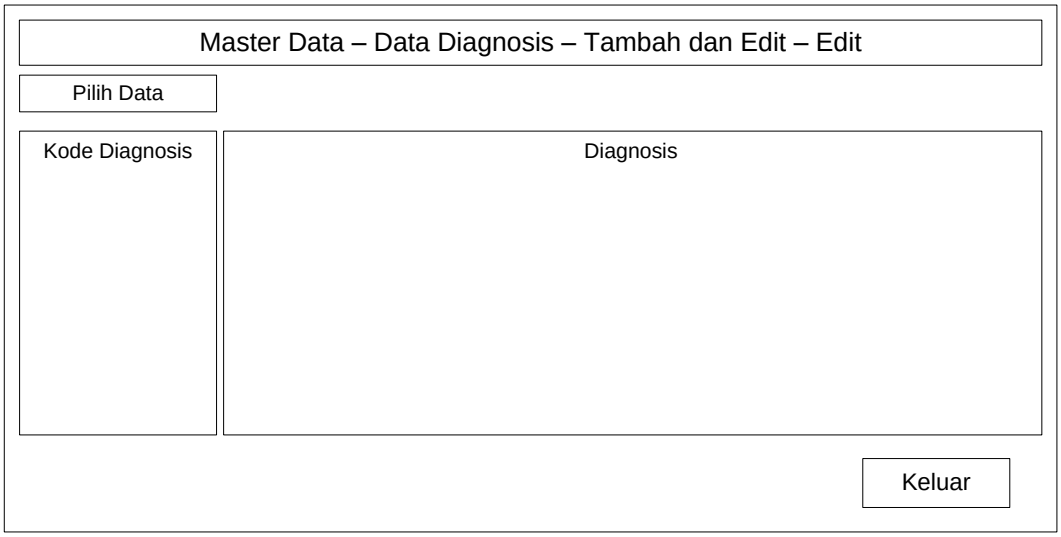
Gambar 3.27 Perancangan Tampilan Lihat Data pada Data Gejala



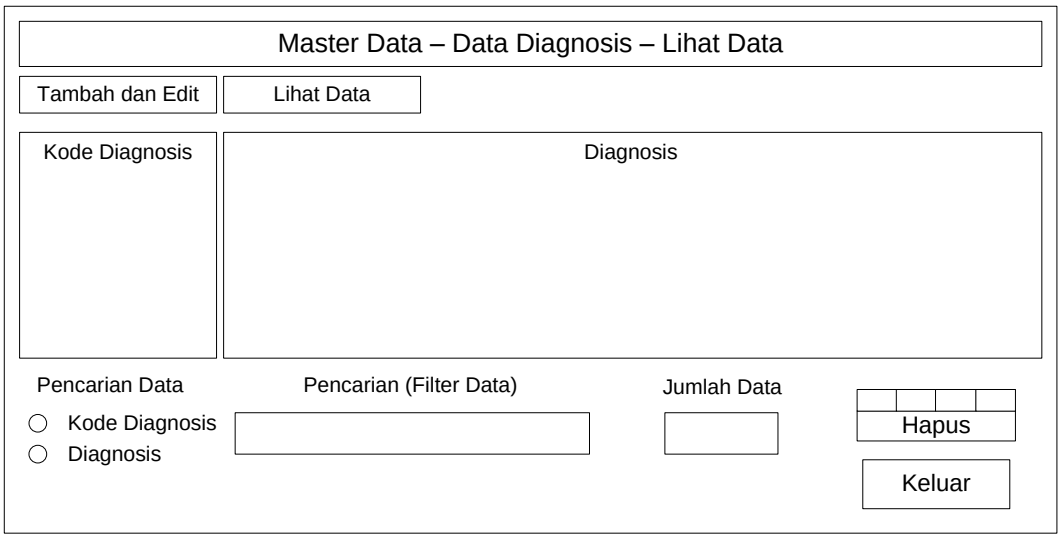
Gambar 3.28 Perancangan Tampilan Perbaikan Basis Pengetahuan, Aturan dan Fakta Aplikasi Berdasarkan Data Diagnosis



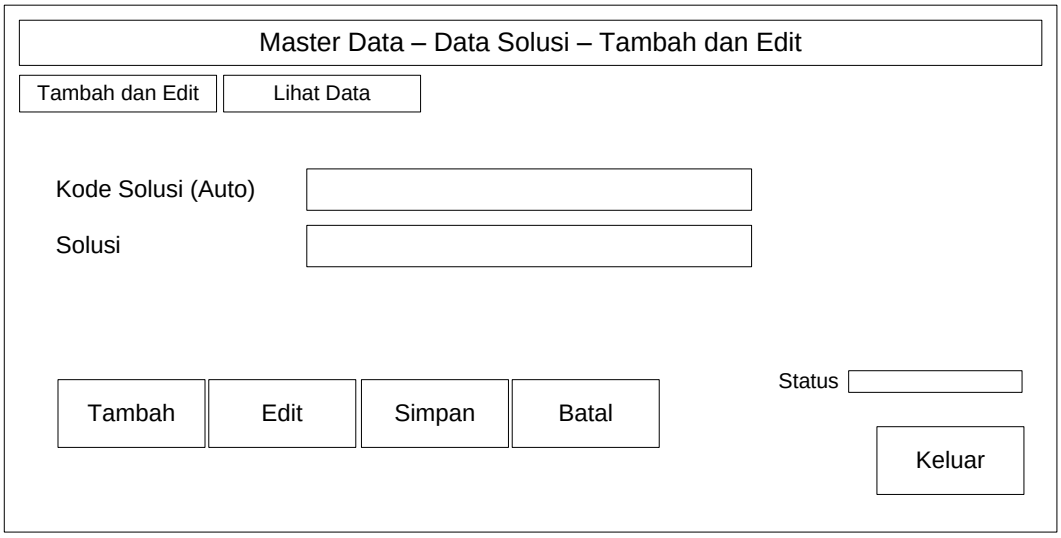
Gambar 3.29 Perancangan Tampilan Edit Data pada Data Diagnosis



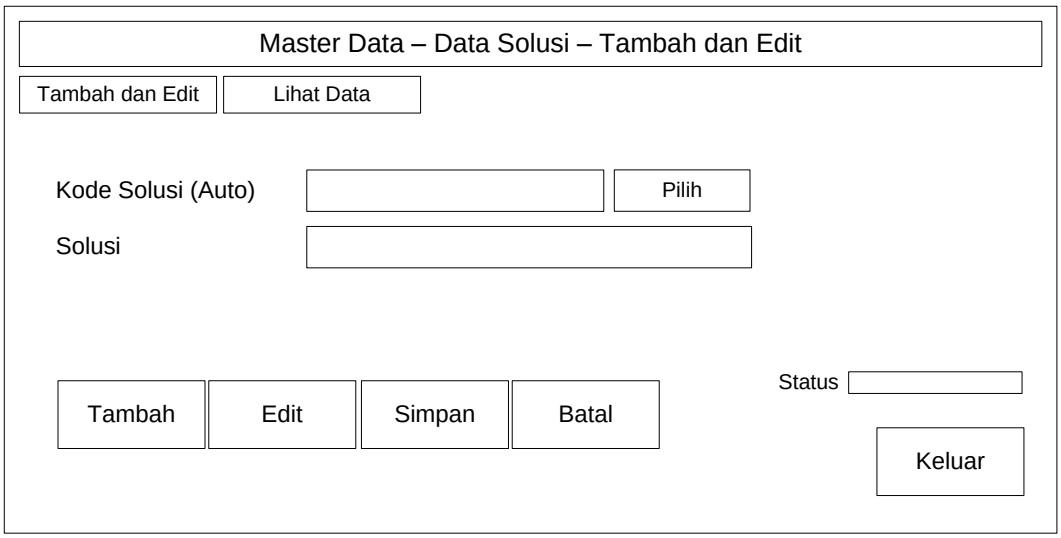
Gambar 3.30 Perancangan Tampilan Edit Data pada Tombol Pilih Data Diagnosis



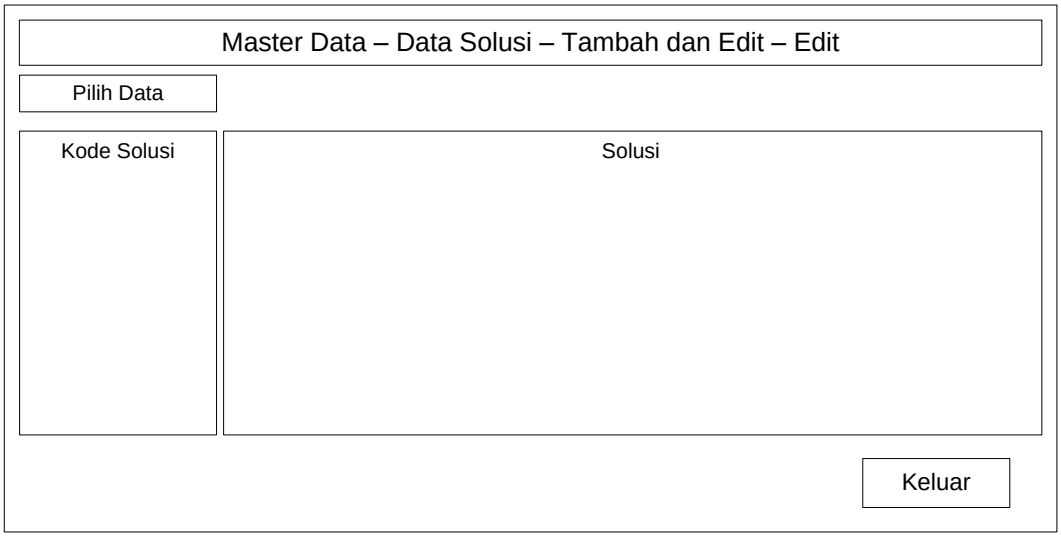
Gambar 3.31 Perancangan Tampilan Lihat Data pada Data Diagnosis



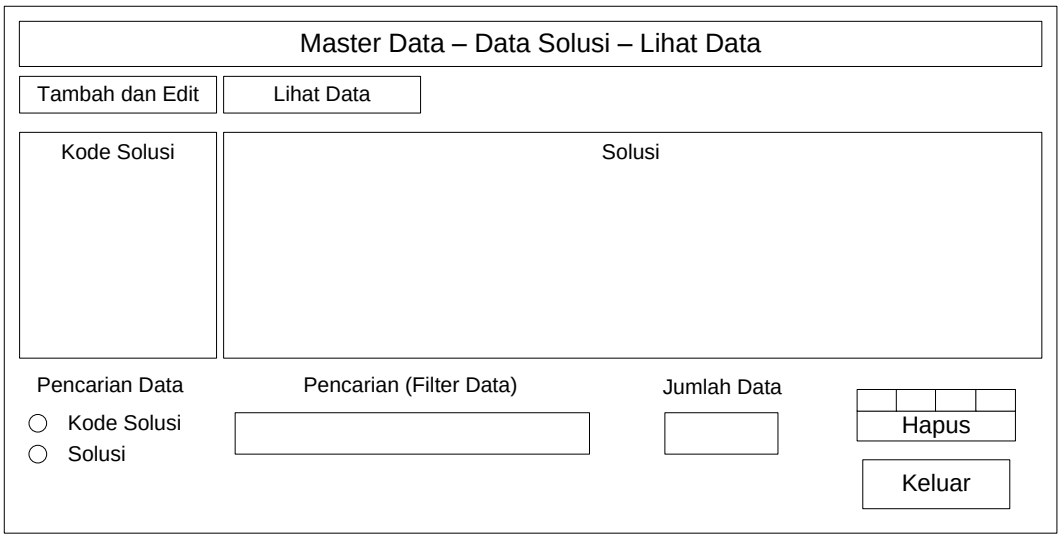
Gambar 3.32 Perancangan Tampilan Perbaikan Basis Pengetahuan, Aturan dan Fakta Aplikasi Berdasarkan Data Solusi



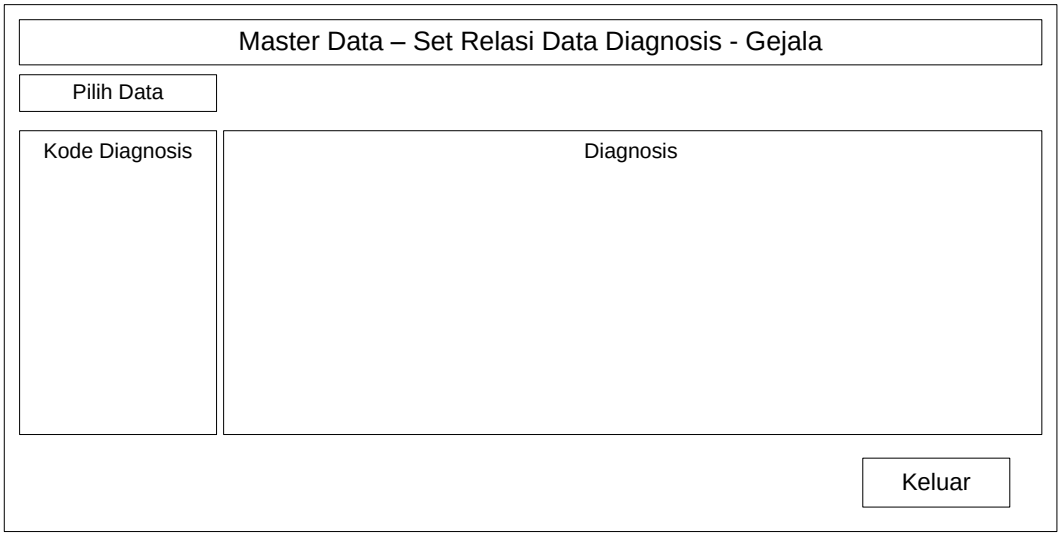
Gambar 3.33 Perancangan Tampilan Edit Data pada Data Solusi



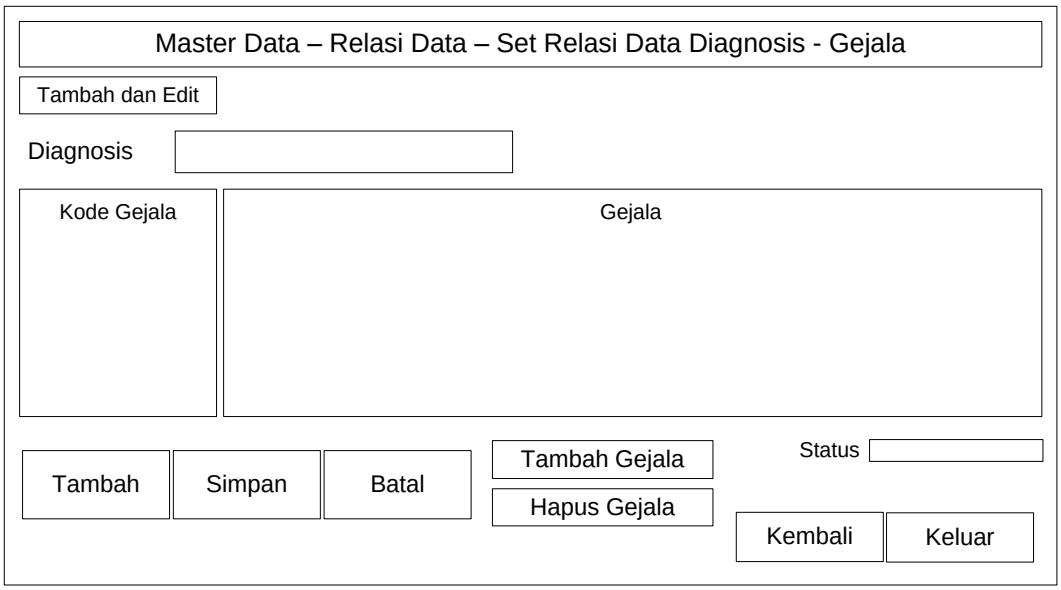
Gambar 3.34 Perancangan Tampilan Edit Data pada Tombol Pilih Data Solusi



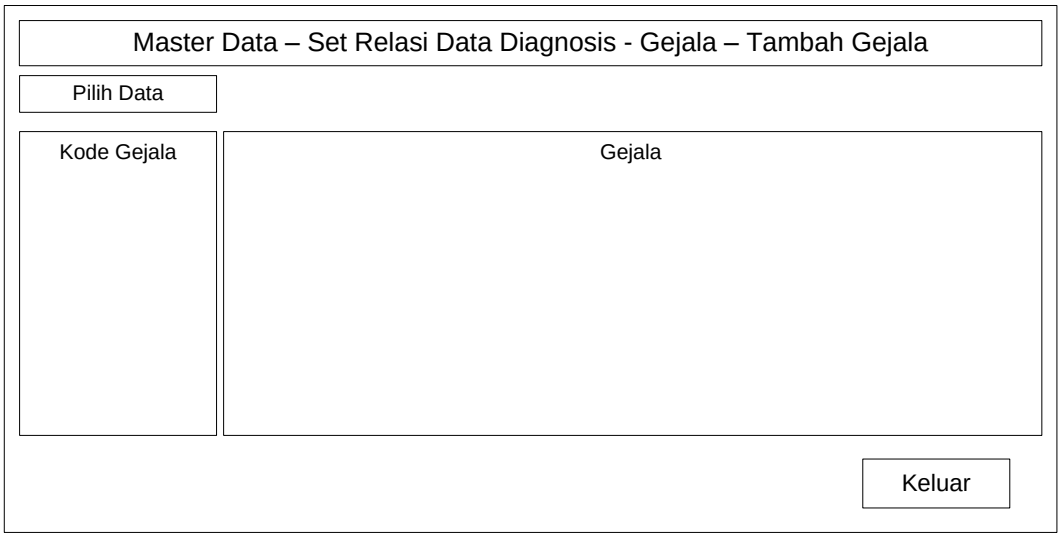
Gambar 3.35 Perancangan Tampilan Lihat Data pada Data Solusi



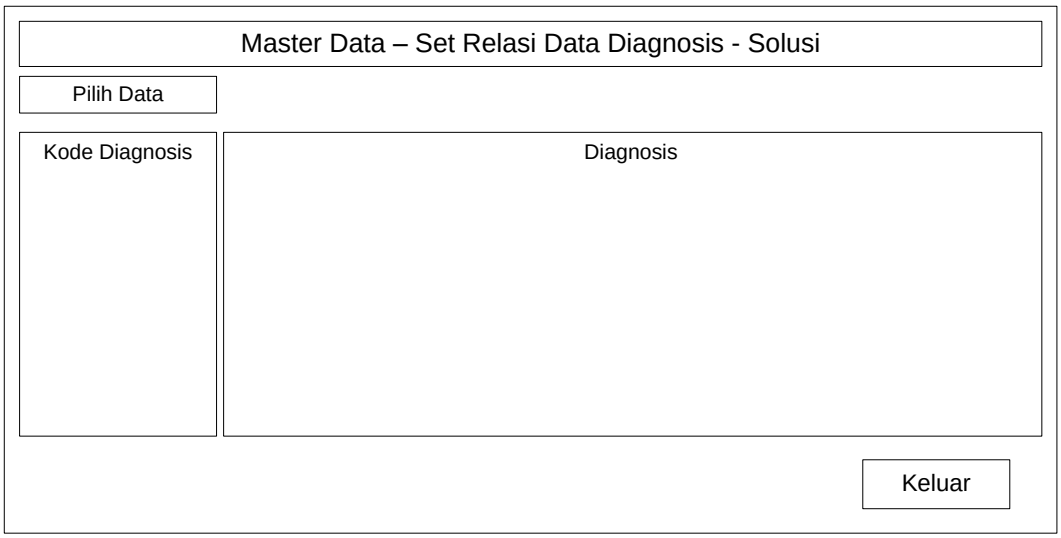
Gambar 3.36 Perancangan Pilih Data pada Set Relasi Data Diagnosis - Gejala



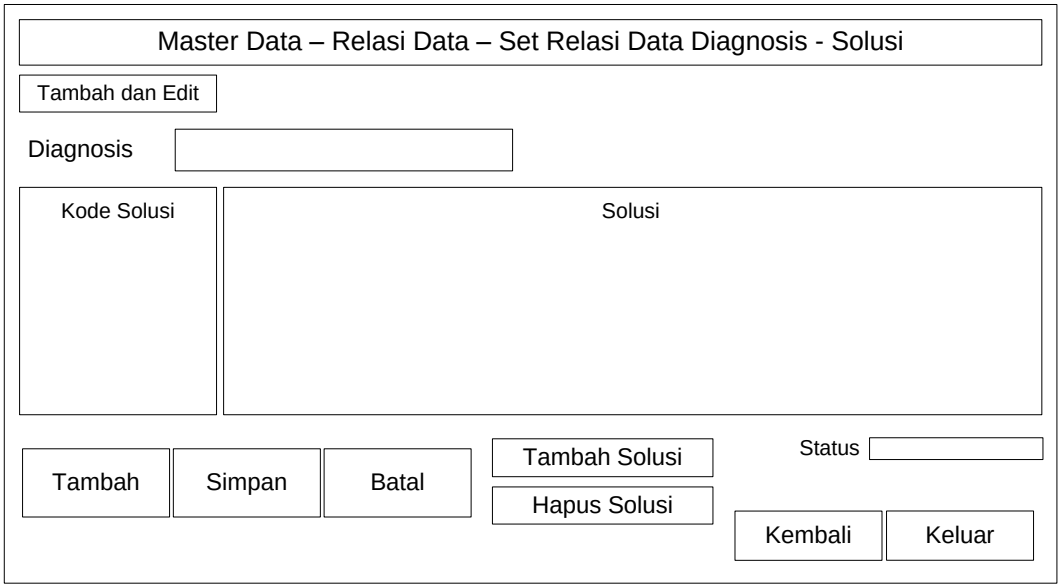
Gambar 3.37 Perancangan Tampilan Set Relasi Data Diagnosis – Gejala



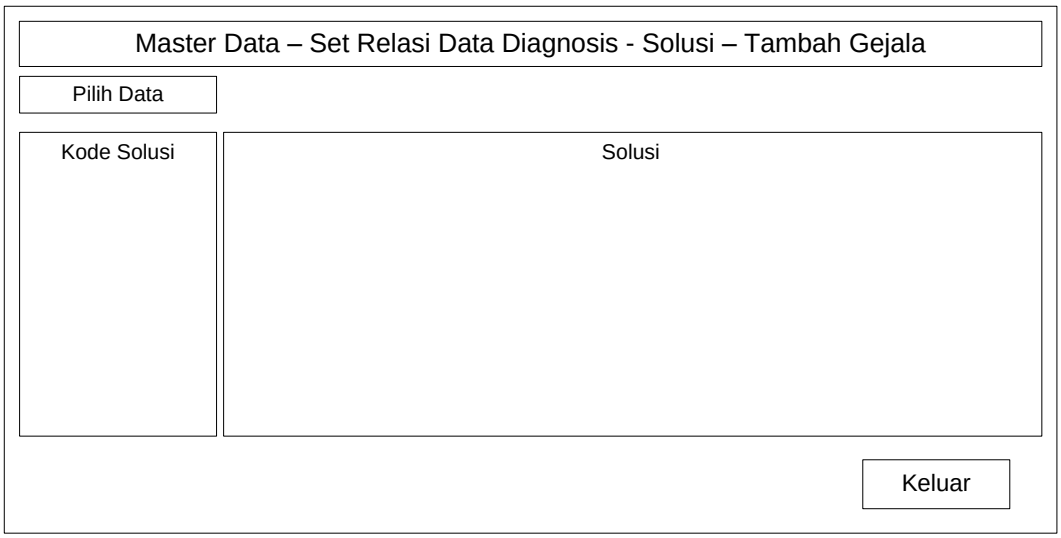
Gambar 3.38 Perancangan Tampilan Tambah Gejala



Gambar 3.39 Perancangan Pilih Data pada Set Relasi Data Diagnosis - Solusi



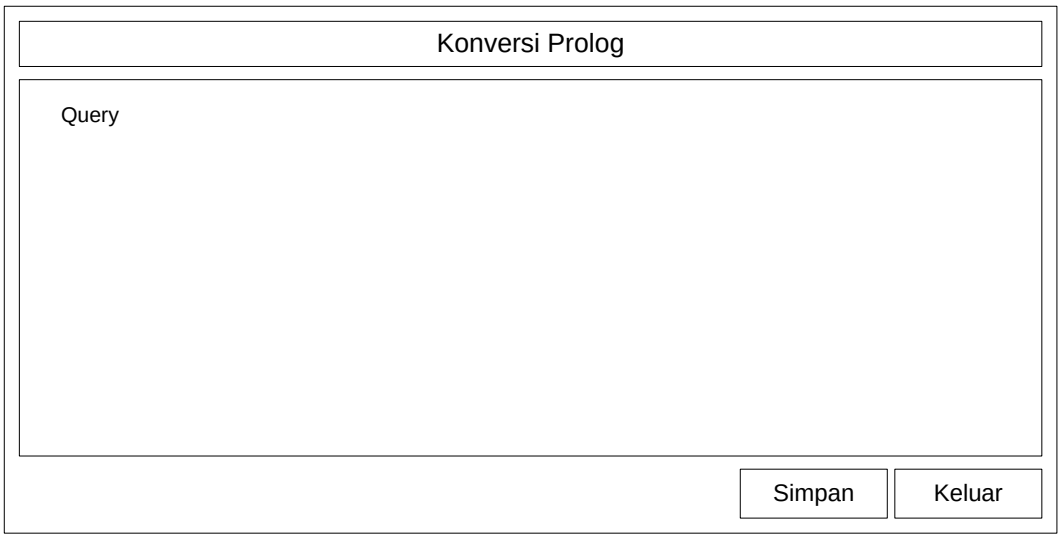
Gambar 3.40 Perancangan Tampilan Set Relasi Data Diagnosis – Solusi



Gambar 3.41 Perancangan Tampilan Tambah Solusi

3.2.2.8 Perancangan Konversi Prolog

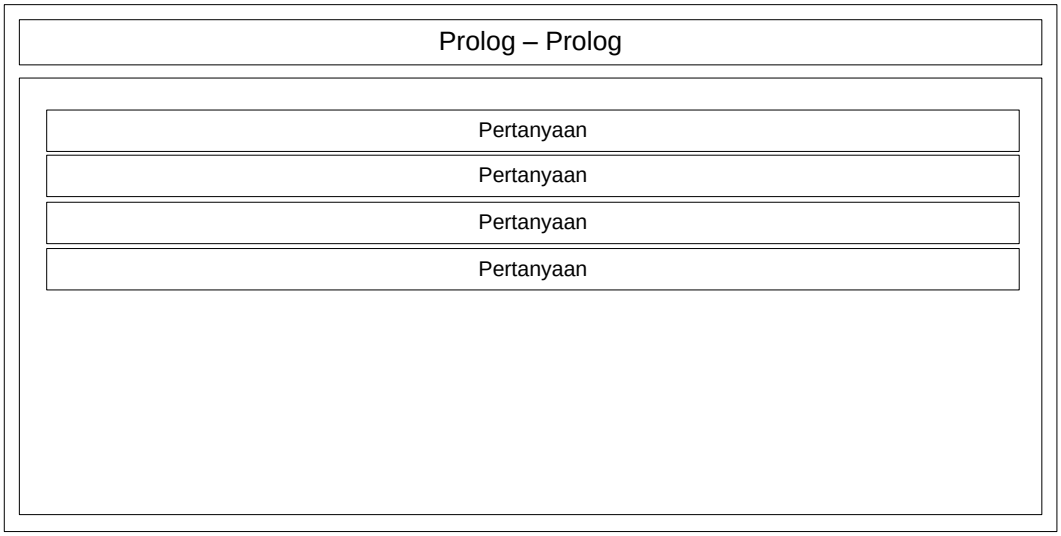
Maksud dari perancangan konversi prolog adalah data gejala, kerusakan dan solusi yang sudah diupdate pada delphi akan di*insert* pada baris-baris di file .PRO yang akan dikonversikan untuk di*compile* di Turbo Prolog 2.0. Nantinya file .PRO yang telah diaktifkan pada Turbo Prolog akan menampilkan Aplikasi Sistem Pakar dengan tampilan under DOS.



Gambar 3.42 Perancangan Tampilan Konversi Prolog

3.2.2.9 Perancangan Input pada Prolog

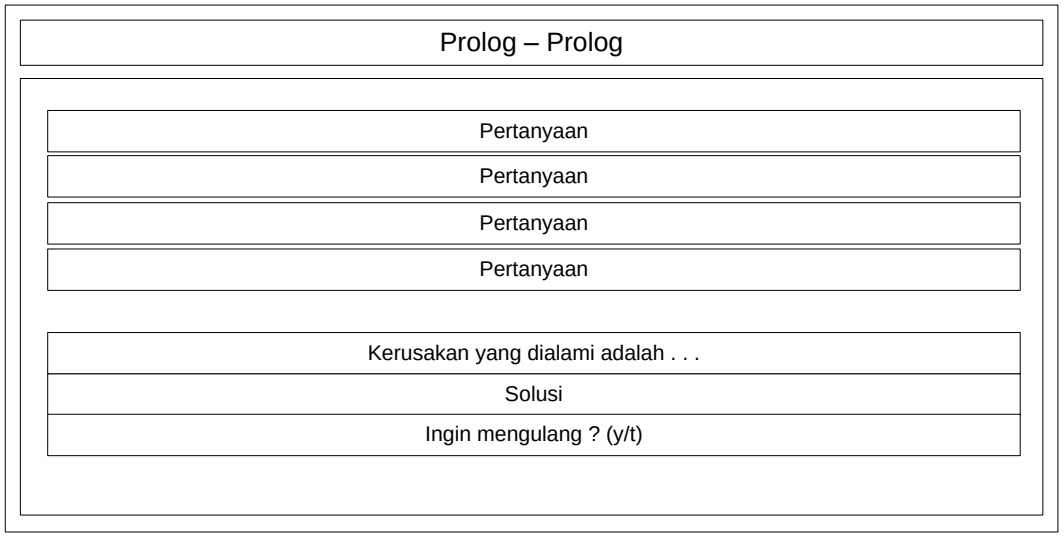
Pada tampilan input, akan ditampilkan beberapa pertanyaan dimana *user* hanya memilih jawaban ‘y’ untuk ya dan ‘t’ untuk tidak. Berikut adalah tampilan rancangan input dari aplikasi yang dibangun:



Gambar 3.43 Perancangan Tampilan Input Pertanyaan

3.2.2.10 Perancangan Output pada Prolog

Hasil output berupa tampilan kerusakan yang dialami dan solusi untuk penanganan kerusakan tersebut. Setelah itu akan ada perintah selanjutnya untuk mengulang atau tidak. Berikut adalah tampilan rancangan output dari aplikasi yang dibangun:



Gambar 3.44 Perancangan Tampilan Output Solusi

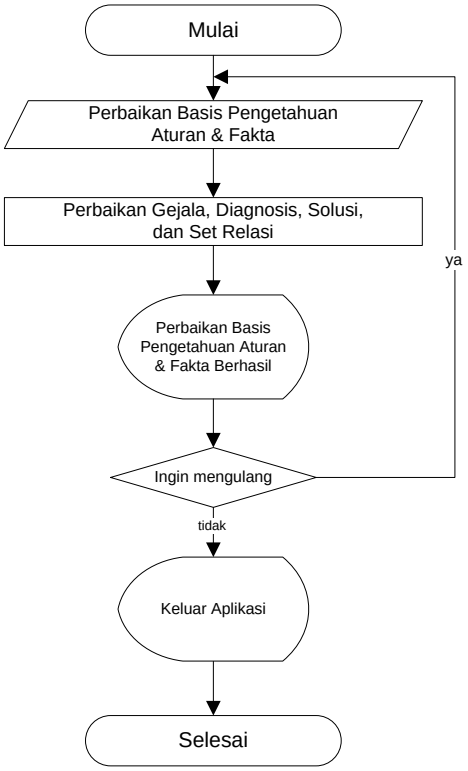
3.2.3 Perancangan Prosedural

Perancangan ini ditujukan untuk memperlihatkan urutan dan hubungan proses dalam aplikasi. Perancangan ini menggunakan metode *conceptual flowchart* yang menggambarkan alur pemecahan kasus secara global.

Sistem pada level pengguna akan dimulai dengan masukan gejala yang disajikan pada sistem berupa pertanyaan dari user melalui keyboard. Kemudian langkah selanjutnya masukan akan di proses ke dalam proses diagnosis yang berelasi dengan logika sistem basis pengetahuan, fakta dan aturan yang

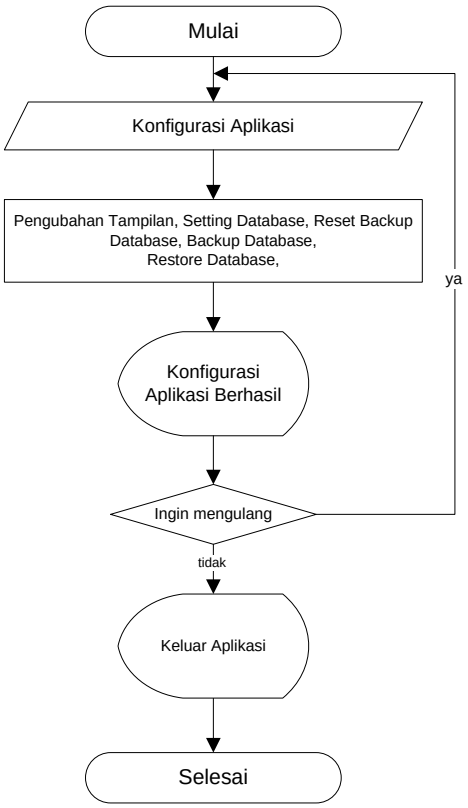
sebelumnya sudah dideklarasikan. Outputnya akan berupa tampilan DOS mengenai kerusakan dan solusi melalui *respon question* pada Masukan Gejala.

Pada Perbaikan Basis Pengetahuan, Aturan dan Fakta akan dimasukkan gejala, diagnosis dan solusi dengan maksud mengupdate sistem yang berjalan, yang nantinya akan terjadi pembaharuan logika sistem (penelusuran baru).



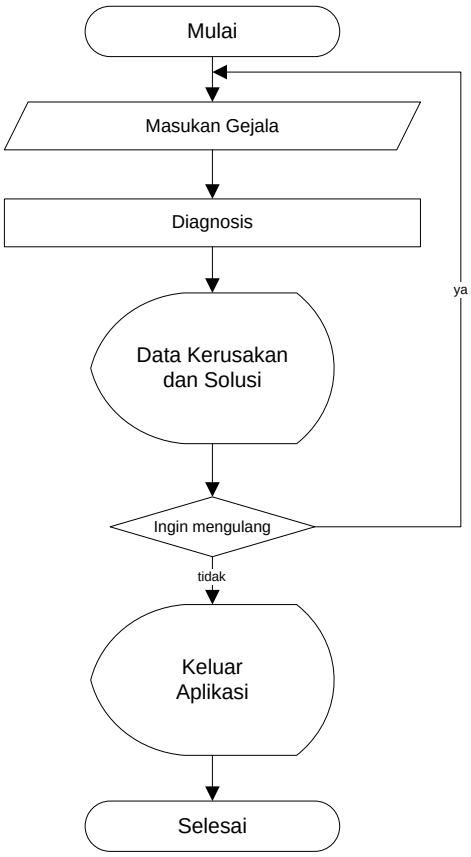
Gambar 3.45 Flowchart Sistem Aplikasi Pendeteksi Kerusakan Yamaha P 900 Perbaikan Basis Pengetahuan Aturan dan Fakta Pada Level Pakar

Keterangan: Perbaikan Basis Pengetahuan Aturan & Fakta merupakan masukan data gejala, data solusi, data diagnosis, dan set relasi data pada sistem.



Gambar 3.46 Flowchart Sistem Aplikasi Pendeteksi Kerusakan Yamaha P 900 Konfigurasi Aplikasi Pada Level Pakar

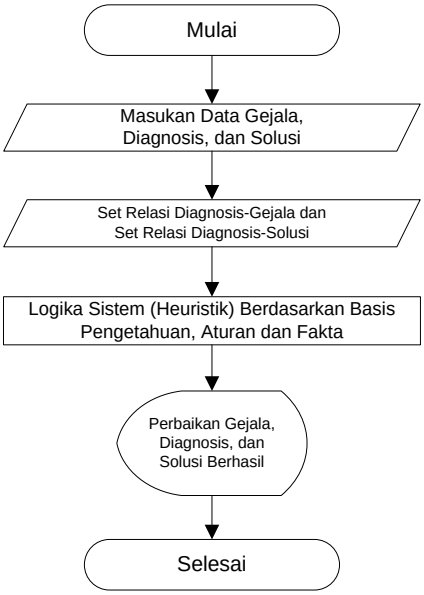
Keterangan: Konfigurasi Aplikasi merupakan pengelolaan dari proses pengubahan tampilan, setting database, reset database, backup database, dan restore database.



Gambar 3.47 Flowchart Sistem Aplikasi Pendeteksi Kerusakan Yamaha P 900 Pada Level Pengguna

Keterangan: Masukan gejala berupa pertanyaan-pertanyaan yang disajikan pada sistem. Hasil masukan akan diproses pada sistem melibatkan pencarian dan penelusuran diagnosis. Kemudian hasilnya akan ditampilkan berupa sumber kerusakan dan solusi penanganan kerusakannya.

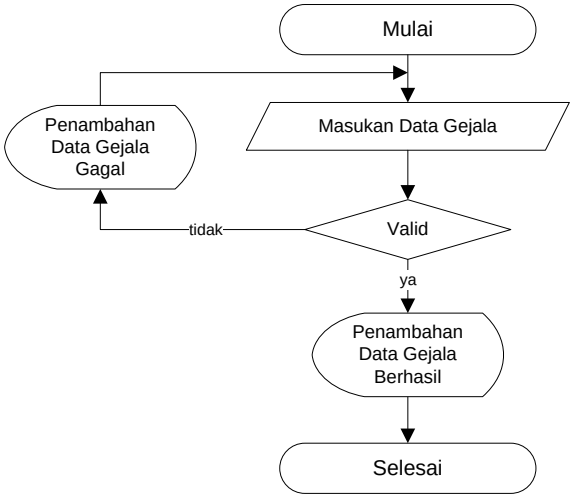
1. Prosedural Perbaikan Gejala, Diagnosis, dan Solusi.



Gambar 3.48 Flowchart Pada Perbaikan Gejala, Diagnosis, dan Solusi Aplikasi Pendeteksi Kerusakan Yamaha P 900 Pada Level Pakar

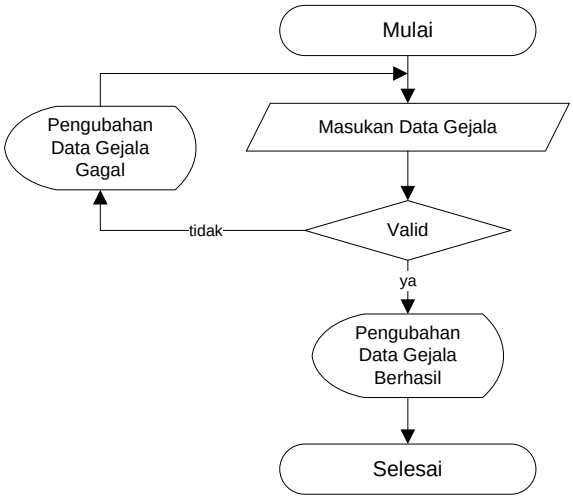
Keterangan: Perbaikan gejala, diagnosis, dan solusi adalah perpanjangan tangan dari prosedural perbaikan basis pengetahuan aturan dan fakta. Masukannya berupa data gejala, data diagnosis, dan data solusi, serta set relasinya.

2. Prosedural Tambah Data Gejala



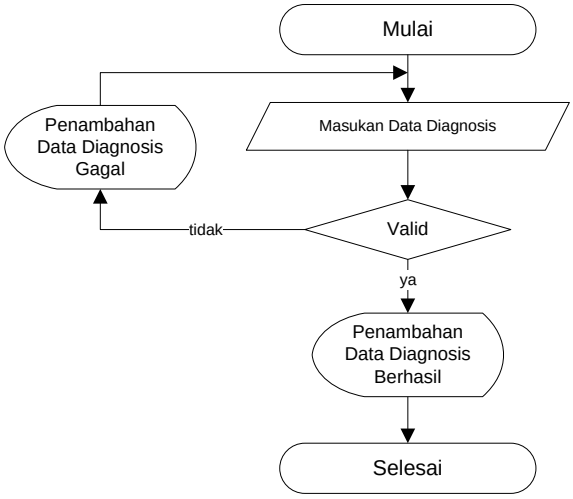
Gambar 3.49 Flowchart Tambah Data Gejala

3. Prosedural Ubah Data Gejala



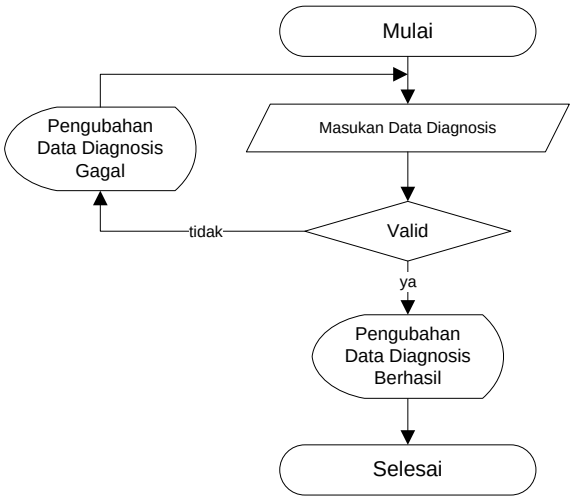
Gambar 3.50 Flowchart Ubah Data Gejala

4. Prosedural Tambah Data Diagnosis



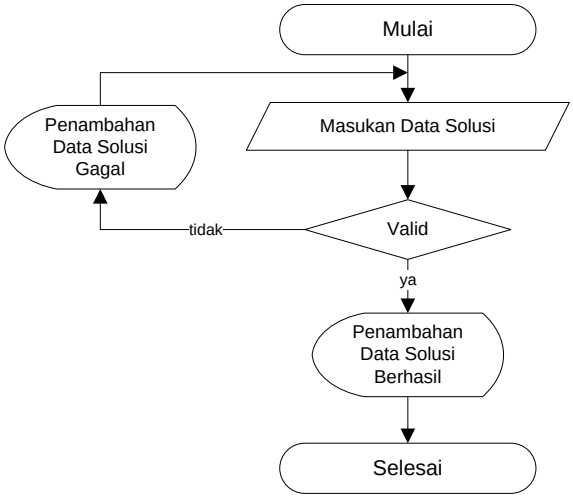
Gambar 3.51 Flowchart Tambah Data Diagnosis

5. Prosedural Ubah Data Diagnosis



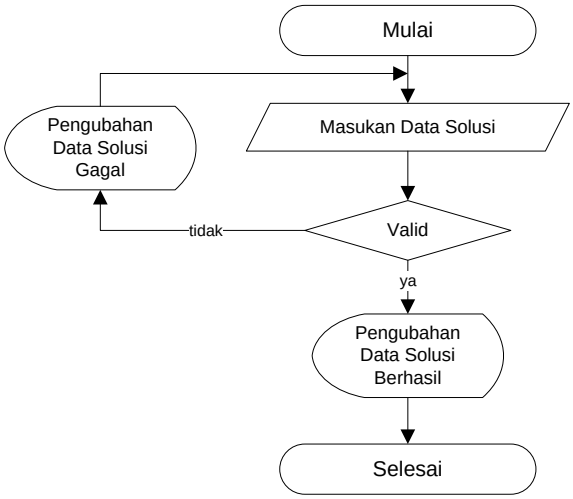
Gambar 3.52 Flowchart Ubah Data Diagnosis

6. Prosedural Tambah Data Solusi



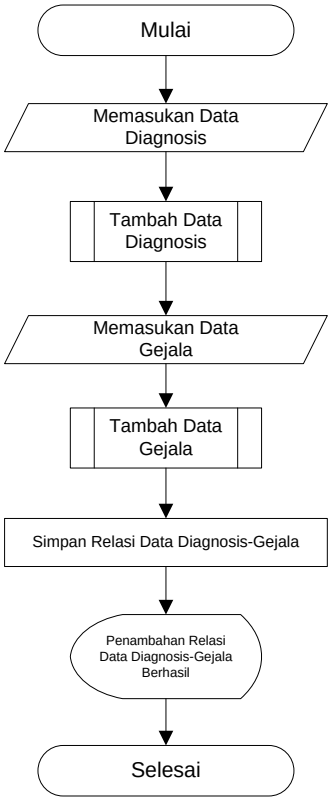
Gambar 3.53 Flowchart Tambah Data Solusi

7. Prosedural Ubah Data Solusi



Gambar 3.54 Flowchart Ubah Data Solusi

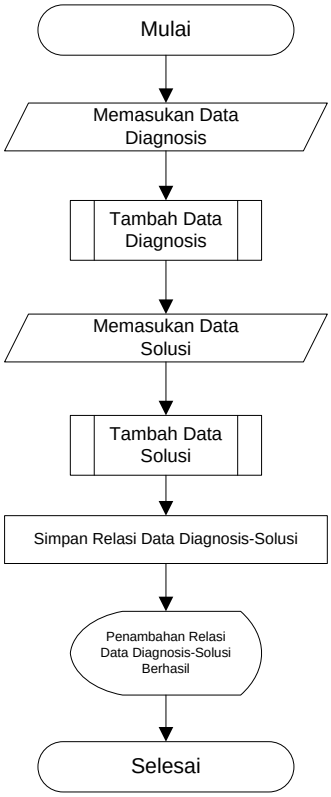
8. Prosedural Tambah Relasi Data Diagnosis – Gejala



Gambar 3.55 Flowchart Tambah Relasi Data Diagnosis - Gejala

Keterangan: Data diagnosis dipilih sebagai acuan untuk merelasikan data gejala. Data yang sudah terdapat pada prosedural tambah data diagnosis dan prosedural tambah data gejala dihubungkan untuk membentuk relasi antar data tersebut.

9. Prosedural Tambah Relasi Data Diagnosis – Solusi



Gambar 3.56 Flowchart Tambah Relasi Data Diagnosis - Solusi

Keterangan: Data diagnosis dipilih sebagai acuan untuk merelasikan data solusi. Data yang sudah terdapat pada prosedural tambah data diagnosis dan prosedural tambah data solusi dihubungkan untuk membentuk relasi antar data tersebut.

Tabel 3.1 Domain Gejala dan Kerusakan Yamaha P 900

KODE (NODE)	GEJALA	KERUSAKAN																																										
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	Mesin tidak mau hidup/sulit dihidupkan	*	*	*	*						*		*		*	*																			*	*								
2	Mesin mudah mati			*	*										*	*																			*		*						
3	Mesin berbunyi tidak normal	*	*	*		*			*																												*							
4	Mesin kotor dan berasap tebal		*	*																	*																						
5	Mesin terlalu panas		*																	*																	*		*		*		*
6	Tenaga mesin berkurang	*	*	*	*											*				*																			*		*			
7	Mesin hidup tetapi kendaraan tidak dapat berjalan								*																													*	*		*		
8	Putaran stasioner tidak sempurna			*				*			*					*																				*			*	*				
9	Putaran mesin tidak sempurna pada saat putaran tinggi			*			*	*							*																				*			*	*				
10	Kopling selip								*																																			
11	Kopling lengket								*																																			
12	Gigi transmisi sulit masuk/sulit bekerja									*		*																																
13	Gigi transmisi sulit kembali ke posisi semula									*																																		
14	Gigi transmisi menghentak/lompat									*		*																																
15	Kopling selip pada pemasukan kopling pertama								*																																			
16	Kopling selip pada pemindahan gigi transmisi								*																																			
17	Pemindahan gigi transmisi keras pada kopling pemindah gigi								*	*											*																						
18	Saat bekerja pemindahan gigi transmisi terlalu cepat								*																																			
19	Saat bekerja pemindahan gigi transmisi terlalu lambat								*																																			
20	Mesin tidak mau menyala	*	*		*						*																										*		*					
21	Pada kecepatan rendah motor bergerak kencang		*		*																																						
22	Pada kecepatan tinggi motor bergerak lambat		*		*																																						
23	Kelebihan/kekurangan bensin		*																																*								
24	Gas macet				*											*																				*			*			*	*
25	Sukar mengemudikan/kemudi berat																													*			*										
26	Kemudi goyang																											*		*		*											
27	Roda depan goyang																											*				*								*			
28	Suspensi depan terlalu lemah																											*															
29	Suspensi depan terlalu keras																											*															
30	Suspensi depan berbunyi																											*															
31	Roda belakang goyang																												*			*								*			
32	Suspensi belakang terlalu lemah																												*														
33	Suspensi belakang terlalu keras																												*														
34	Suspensi belakang berbunyi																												*														
35	Roda belakang terasa berat																																*							*			
36	Gardan berbunyi tidak normal																																*										
37	Daya rem berkurang																														*												
38	Rem berbunyi																														*	*										*	
39	Gerak tuas rem sulit																														*												
40	Minyak rem bocor		*																												*												
41	Sirine tidak menyala													*				*				*																						
42	Sirine tidak berbunyi dengan stabil															*		*				*	*																						
43	Lampu rotator tidak menyala													*				*						*																				
44	Lampu rotator tidak berputar															*		*									*																
45	Lampu hazard tidak menyala													*				*						*																				
46	Lampu hazard tidak berkedip normal													*		*		*						*																				
47	Lampu utama depan dan belakang tidak menyala													*				*						*																				
48	Lampu utama depan dan belakang redup															*		*						*																				
49	Public address tidak menyala													*		*		*					*		*																				
50	Speedometer tidak menyala													*		*										*																		
51	Jam tidak menyala													*		*		*																										
52	Tidak ada bunga api saat penyalaan mesin													*	*	*																												
53	Busi menjadi cepat tertutup arang					*										*																						*				*	
54	Busi cepat kotor		*	*																																							
55	Busi terlalu panas/hangus															*																										*	
56	Generator tidak menyala	*					*							*						*																						
57	Generator tidak mengisi dan pengisianpun dibawah spesifikasi													*	*		*			*																						
58	Pengisian digenerator berlebihan																*			*																						
59	Pengisian digenerator tidak stabil													*						*																						
60	Tombol starter tidak berfungsi												*				*	*																									
61	Terdapat sulfasi pada pelat sel yang berhubungan																*																										
62	Kapasitas accu cepat menurun																*																										
63	Pemasangan accu terbalik																*																										
64	Terdapat sulfasi pada accu																*																										
65	Daya kerja accu menurun										*						*																										
66	Bunyi knalpot tidak normal																																									*	
67	Bunyi knalpot meletup/meledak																																								*	