BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

4.1.1 Tujuan Perancangan Sistem

Perancangan sistem dibuat untuk memberikan gambaran umum kepada pengguna tentang sistem informasi yang akan didesain secara rinci. Tujuan perancangan sistem pakar ini akan diuraikan sebagai berikut :

1. Untuk mengurangi kekurangan dan kelemahan yang terdapat pada sistem sebelumnya dan meningkatkan ketepatan, kecepatan, dan keakuratan informasi yang dibutuhkan.
2. Diharapkan dapat memberikan gambaran yang jelas mengenai sistem yang akan dibuat.

4.1.2 Gambaran Umum Sistem yang Diusulkan

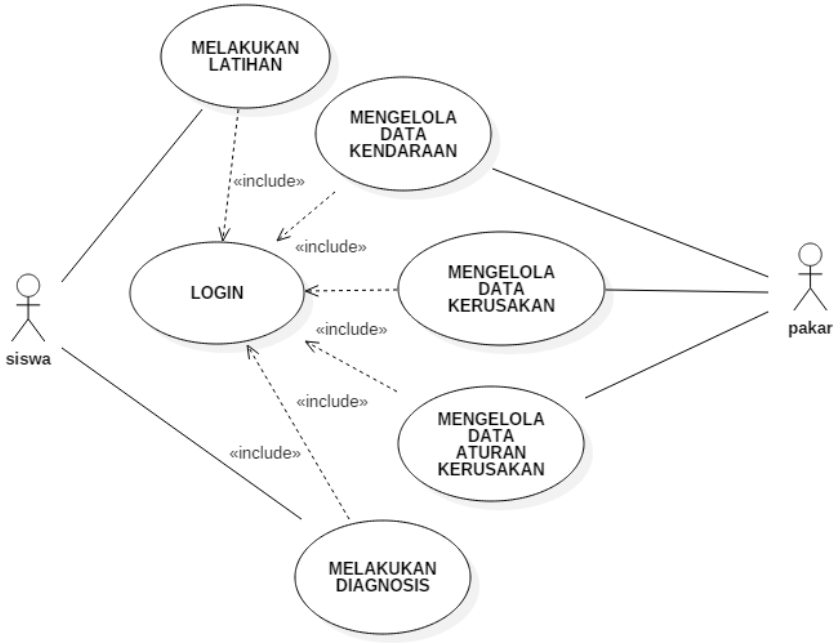
Sistem dibuat untuk membantu pengajar dan siswa dalam proses perbaikan kerusakan, untuk pengajar berguna sebagai asisten untuk mendampingi siswa ketika melakukan perbaikan dengan mengelola kebutuhan data dan pengetahuan untuk sistem, untuk siswa mempermudah mereka dalam melakukan perbaikan khususnya dalam perkiraaan kerusakan berdasarkan gejala – gejala yang terjadi.

4.1.3 Perancangan Prosedur yang Diusulkan

Perancangan proses ini akan memberikan gambaran mengenai dokumen-dokumen, proses-proses dan aliran data apa saja yang akan terlibat dalam sistem yang akan diusulkan.

4.1.3.1 *Use Case Diagram*

Use Case Diagram menunjukan bagaimana *behavior* dan interaksi aktor yang diharapkan dari sistem yang diusulkan.



Gambar 4.1 Use Case Diagram yang diusulkan

a. Definisi Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibat dalam sistem pakar yang akan diusulkan.

Tabel 4.1 Definisi Aktor

No	Aktor	Deskripsi
1	Pakar	Mengelola Data Kendaraan
		Mengelola Data kerusakan
		Mengelola Data Aturan Kerusakan
2	Siswa	Melakukan Diagnosis kerusakan
		Melakukan Latihan

b. Definisi Use Case

Use case merupakan urutan kejadian yang menggambarkan interaksi *user* dengan sistem.

Tabel 4.2 Definisi Use Case

No	Use Case	Deskripsi
1	*Login*	Merupakan aktivitas user yang bisa melakukan akses kedalam sistem
2	Mengelola data Kendaraan	Merupakan serangkaian aktivitas yang mengelola data kendaraan seperti menambah, menghapus, dan mengubah data kendaraan.
3	Mengelola data kerusakan	Merupakan serangkaian aktivitas yang mengelola data kerusakan seperti menambah, menghapus, dan mengubah data kerusakan.
4	Mengeloa data aturan kerusakan	Merupakan serangkaian aktivitas yang mengelola data aturan kerusakan seperti mengubah nilai dari aturan suatu gejala kerusakan terhadap kerusakan.
5	Melakukan Diagnosis	Merupakan aktivitas user dalam mendiagnosis suatu kerusakan kendaraan dengan memilih pilihan gejala yang tersedia sesuai dengan gejala yang diamati untuk mendapatkan hasil diagnosis kerusakan.
6	Melakukan Latihan	Merupakan aktivitas user berlatih mendiagnosis suatu kerusakan kendaraan dengan memilih gejala yang tersedia sesuai dengan gejala yang diamati dan memilih kerusakan yang di duga berdasarkan gejala yang diamati untuk di bandingkan dengan hasil diagnosis kerusakan berdasarkan sistem dari gejala yang dipilih.

4.1.3.2 Skenario Use Case

1. Nama use case : Login
   Deskripsi : Mengelola validasi pengguna
   Aktor : Pakar, Siswa
   Kondisi awal : *User* memasukan *username* dan *password*
   Kondisi akhir : Menampilkan halaman sesuai dengan status *user*

Tabel 4.3 Skenario Use Case Login yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Mengisi *username* dan *password* lalu klik tombol *sign in*		
		2.	Sistem memeriksa *username* dan password ke basis data
		3.	Sistem menampilkan halaman sesuai dengan status *user*
Exception			
Jika *username* dan *password* yang diinputkan salah (tidak ditemukan di dalam basis data), maka sistem akan memunculkan pesan teks *username* atau *password* salah di tombol *sign in.*			

2. Nama use case : Mengelola data kendaraan
   Deskripsi : Mengelola data – data kendaraan
   Aktor : Pakar
   Kondisi awal : Pakar memilih menu data kendaraan
   Kondisi akhir : Setiap perubahan data kendaraan tersimpan dalam basis data

Tabel 4.4 Skenario Use Case Mengelola Data Kendaraan yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Memilih menu data kendaraan		

		2.	Sistem menampilkan halaman kelola data kendaraan
3.	Melakukan pengelolaan data kendaraan		
		4.	Sistem menampilkan pengelolaan data sesuai yang di inginkan
5	Melakukan pengelolaan data kendaraan dan menyimpan hasil pengelolaan		
		6	Setiap perubahan data kendaraan disimpan dalam basis data

3. Nama use case : Mengelola data kerusakan
   Deskripsi : Mengelola data – data kerusakan
   Aktor : Pakar
   Kondisi awal : Pengajar memilih menu data kerusakan
   Kondisi akhir : Setiap perubahan data kerusakan kendaraan tersimpan dalam basis data

Tabel 4.5 Skenario Use Case Mengelola Data Kerusakan yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Memilih menu data kerusakan		
		2.	Sistem menampilkan halaman kelola data kerusakan
3.	Memilih data kendaraan lalu klik tombol *open data*		
		4.	Sistem menampilkan data kerusakan kendaraan sesuai dengan kendaraan yang dipilih
5.	Mengelola data kerusakan kendaraan		

		6.	Sistem menampilkan *popup* pengelolan data sesuai dengan yang di inginkan
7	Mengelola data kerusakan kendaraan dan menyimpan hasil pengelolaan		
		8	Setiap perubahan data kerusakan kendaraan tersimpan dalam basis data

4. Nama use case : Mengelola Data Aturan Diagnosis Kerusakan
   Deskripsi : Mengelola data – data peluang kerusakan
   Aktor : Pakar
   Kondisi awal : Pakar memilih menu *basic rule*
   Kondisi akhir : Setiap perubahan data aturan diagnosis kerusakan tersimpan di basis data

Tabel 4.6 Skenario Use Case Mengelola Data Kerusakan yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Memilih menu *basic rule*		
		2.	Sistem menampilkan halaman kelola data aturan diagnosis kerusakan
3.	Memilih data kendaraan lalu klik tombol *open data*		
		4.	Sistem menampilkan pilihan data komponen dan pengelolaan aturan diagnosis kerusakan berdasarkan komponen secara *default* sesuai dengan data kendaraan yang dipilih
5.	Memilih data komponen lalu klik tombol *open rules*		

		6.	Sistem menampilkan pengelolaan data aturan diagnosis kerusakan berdasarkan data komponen yang dipilih
7.	Mengelola data aturan diagnosis kerusakan dan menyimpan hasil pengelolaan		
		8.	Setiap perubahan data aturan diagnosis kerusakan tersimpan di basis data

5. Nama use case : Melakukan Diagnosis
   Deskripsi : Memilih kendaraan dan mengisi data gejala kerusakan untuk analisis kerusakan
   Aktor : Siswa
   Kondisi awal : Menceklis pilihan gejala sesuai dengan gejala yang terjadi
   Kondisi akhir : Menampilkan hasil perhitungan diagnosis perkiraan kerusakan sesuai dengan data gejala yang dipilih oleh siswa

Tabel 4.7 Skenario Use Case Melakukan Diagnosis yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Memilih kendaraan lalu menekan tombol *open data*		
		2.	Sistem mencari data gejala kerusakan di basis data berdasarkan kendaraan yang di pilih

		3.	Sistem menampilkan data gejala kerusakan sesuai dengan data kendaraan yang dipilih
4.	Memilih pilihan gejala kerusakan		
		5.	Sistem mencari data kerusakan dan aturan kerusakan berdasarkan gejala yang dipilih ke basis data lalu menghitung data aturan kerusakan dan menampilkan hasil perhitungan
Exception			
Jika data gejala kerusakan berdasarkan kendaran yang dipilih tidak ditemukan di basis data, maka sistem akan memunculkan pesan data gejala kerusakan tidak ada			

6. Nama use case : Melakukan Latihan
   Deskripsi : Memilih kendaraan lalu memilih data gejala dan data kerusakan untuk mengetahui kerusakan yang dipilih benar atau tidak
   Aktor : Siswa
   Kondisi awal : Menceklis pilihan gejala sesuai dengan gejala yang terjadi dan data kerusakan yang tersedia
   Kondisi akhir : Menampilkan hasil perhitungan diagnosis perkiraan kerusakan sesuai dengan data gejala yang dipilih oleh siswa dan menampilkan benar atau tidak kerusakan yang dipilih oleh siswa.

Tabel 4.8 Skenario Use Case Melakukan Latihan yang diusulkan

No	Aksi Aktor	No	Aksi Sistem
1.	Memilih kendaraan lalu menekan tombol *open data*		

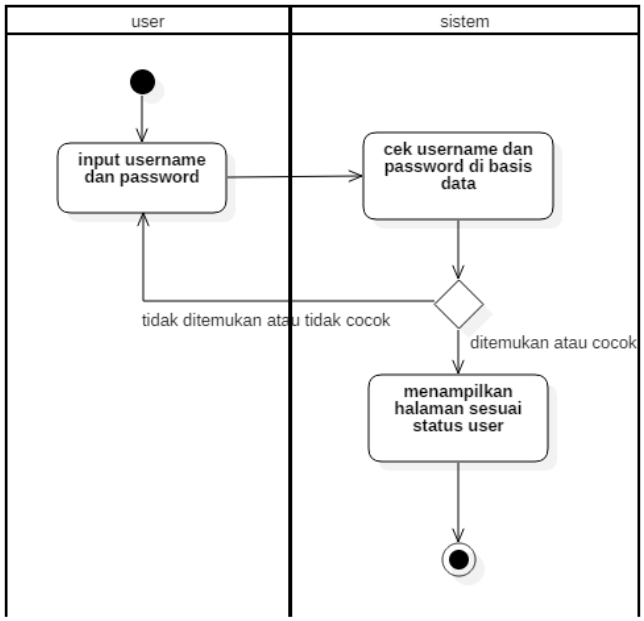
		2.	Sistem mencari data gejala dan data kerusakan di basis data berdasarkan kendaraan yang di pilih
		3.	Sistem menampilkan data gejala dan data kerusakan sesuai dengan data kendaraan yang dipilih
4.	Memilih pilihan gejala dan memilih pilihan kerusakan		
		5.	Sistem mencari data kerusakan dan aturan kerusakan berdasarkan gejala yang dipilih ke basis data lalu menghitung data aturan kerusakan, membandingkan kerusakan yang didapat dengan yang dipilih oleh siswa dan menampilkan hasil perhitungan serta hasil perbandingan kerusakan yang dilakukan.

4.1.3.3 *Activity Diagram*

Activity diagram digunakan untuk menggambarkan kegiatan – kegiatan yang ada dalam suatu sistem.

1. Deskripsi

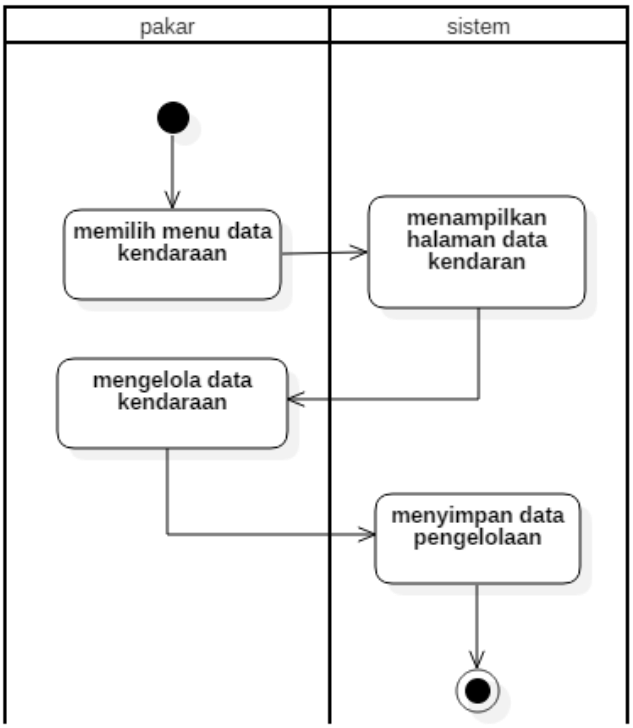
   User mengakses aplikasi lalu memasukan *username* dan *password*, sistem mengecek *username* dan *password* ke basis data, jika tidak sesuai memunculkan pesan *username* atau *password* salah dan *user* memasukan kembali *username* dan *password*, jika sesuai maka *user* akan diarahkan menuju halaman sesuai status *user* di basis data.



Gambar 4.2 Activity Diagram Login yang diusulkan

2. Deskripsi

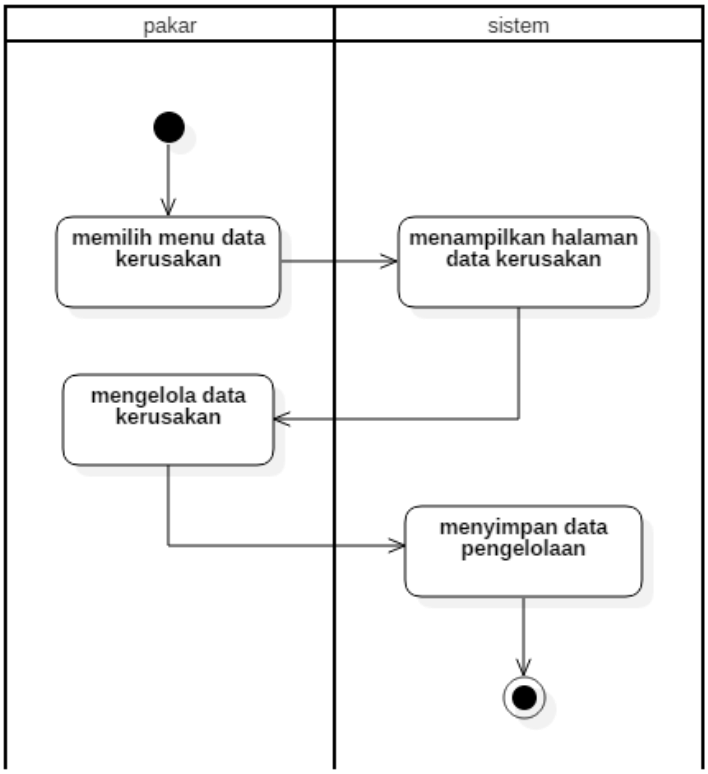
   Pakar mengakses menu data kendaraan dan mengelola data kendaraan lalu menyimpan setiap perubahan data kendaraan.



Gambar 4.3 Activity Diagram Pengelolaan Data Kendaraan yang diusulkan

3. Deskripsi

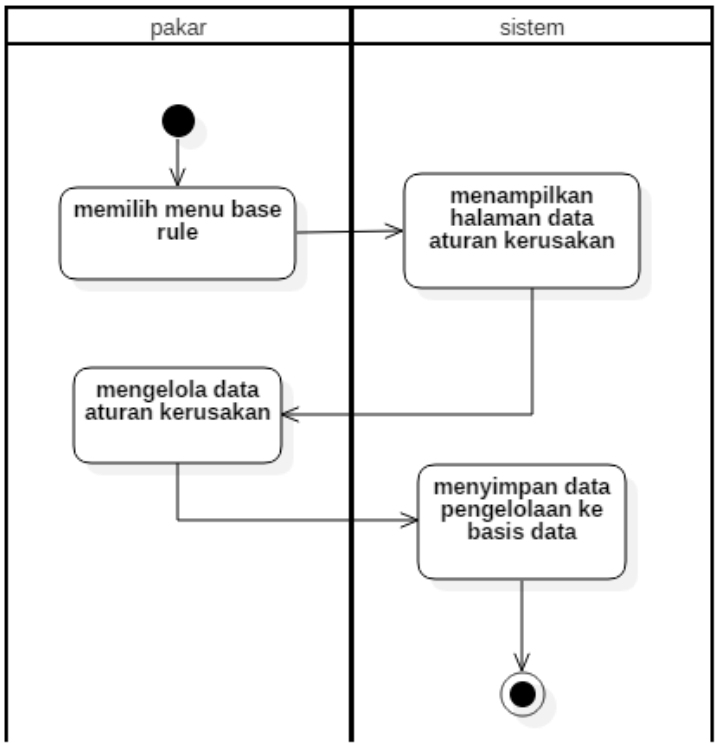
   Pakar mengakses menu data kerusakan dan mengelola data kendaraan lalu menyimpan setiap perubahan data kerusakan.



Gambar 4.4 Activity Diagram Pengelolaan Data Kerusakan yang diusulkan

4. Deskripsi

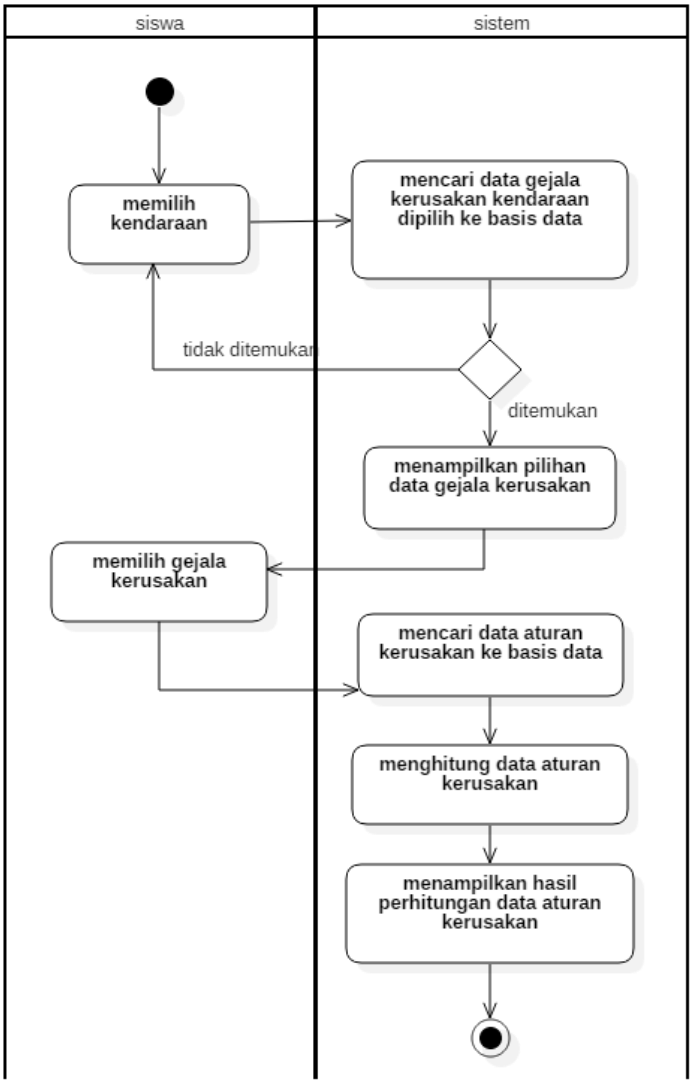
   Pakar mengakses menu *base rule* dan mengelola data aturan kerusakan lalu menyimpan setiap perubahan data aturan kerusakan.



Gambar 4.5 Activity Diagram Pengelolaan Data Aturan Kerusakan yang diusulkan

5. Deskripsi

   Ketika siswa berhasil *login*, siswa memilih data kendaraan lalu sistem mencari dan menampilkan data gejala kerusakan berdasarkan data kendaraan dipilih di basis data, siswa memilih data gejala, sistem mengambil data aturan kerusakan berdasarkan gejala yang dipilih dan melakukan perhitungan diagnosis lalu menampilkan hasil dari perhitungan diagnosis.

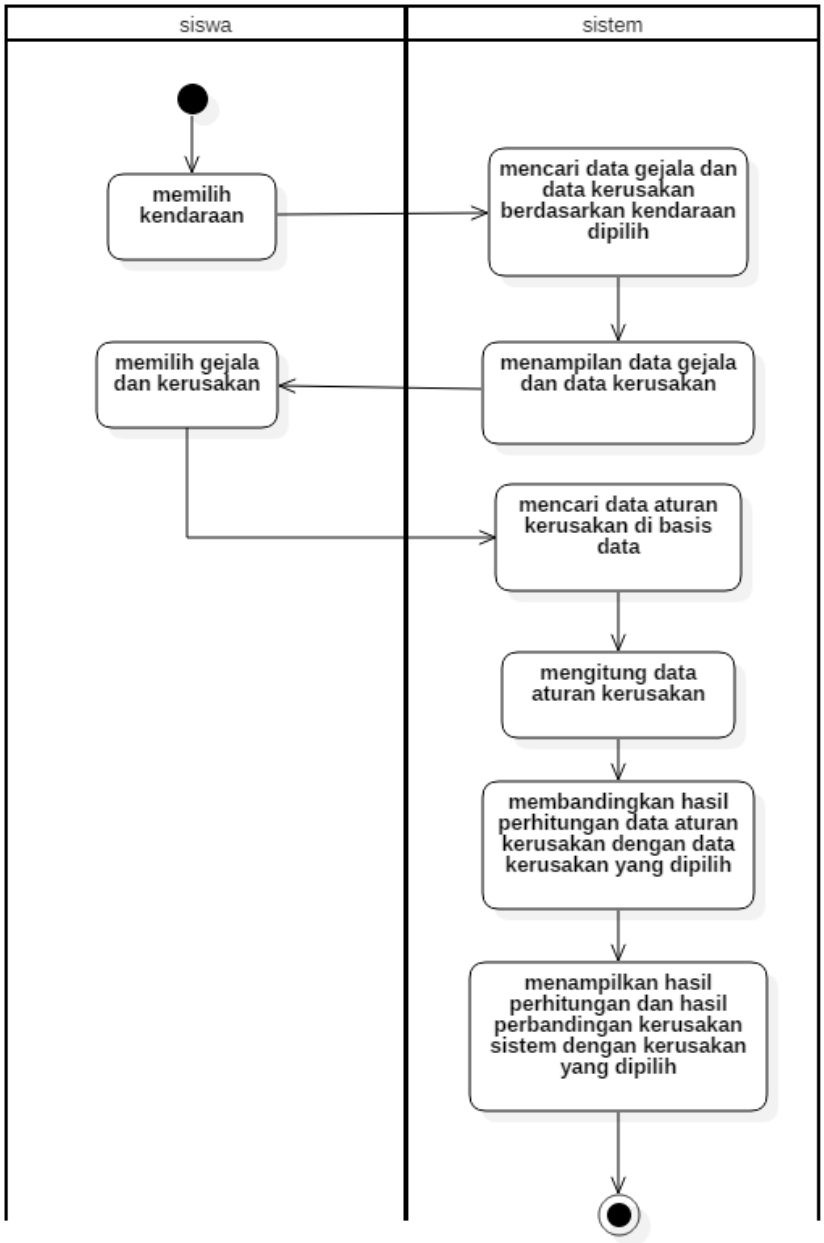


Gambar 4.6 Activity Diagram Diagnosis Kerusakan yang diusulkan

6. Deskripsi

Ketika siswa berhasil *login*, siswa memilih data kendaraan lalu sistem mencari dan menampilkan data gejala dan data kerusakan berdasarkan data kendaraan dipilih di basis data, siswa memilih data gejala dan data kerusakan, sistem mengambil data aturan kerusakan berdasarkan gejala yang dipilih dan melakukan perhitungan diagnosis lalu membandingkan hasil diagnosis (kerusakan) dengan kerusakan yang dipilih oleh siswa,

menampilkan hasil dari perhitungan diagnosis dan perbandingan hasil diagnosis (kerusakan).

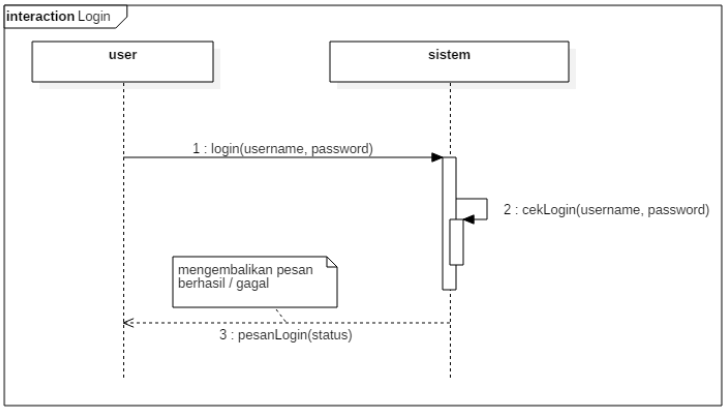


Gambar 4.7 Activity Diagram Latihan Diagnosis yang diusulkan

4.1.3.4 Sequence Diagram

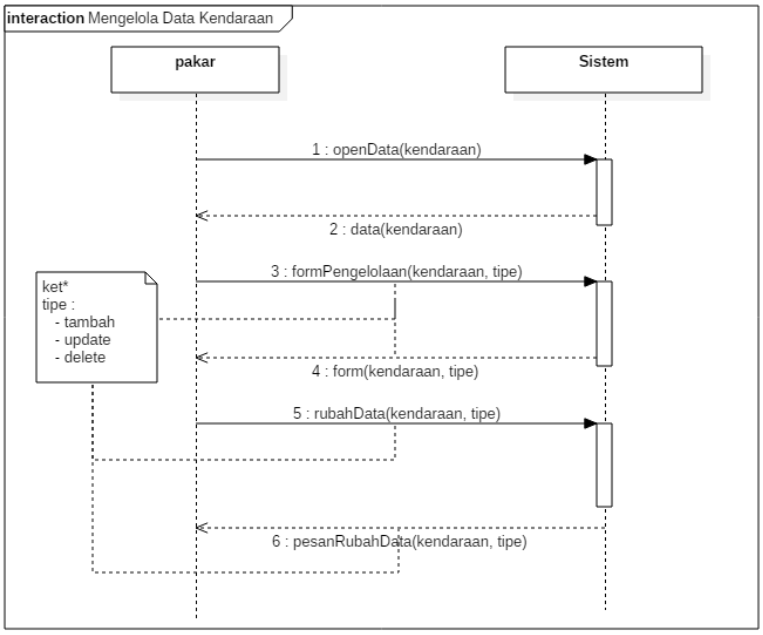
Sequence diagram mengambarkan interaksi aktor pada sistem tersebut berdasarkan garis waktu.

1. *Sequence Diagram Login*



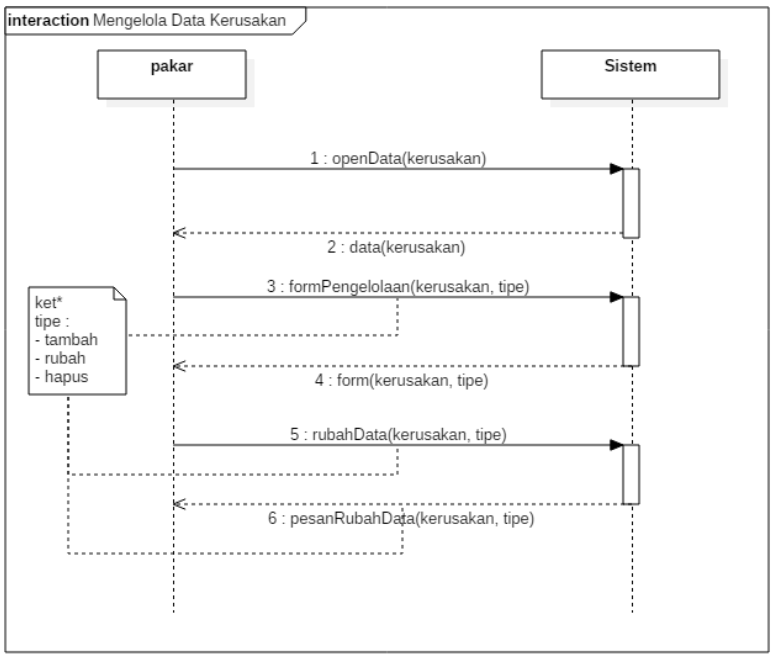
Gambar 4.8 Sequence Diagram Login yang diusulkan

2. *Sequence Diagram* Mengelola Data Kendaraan



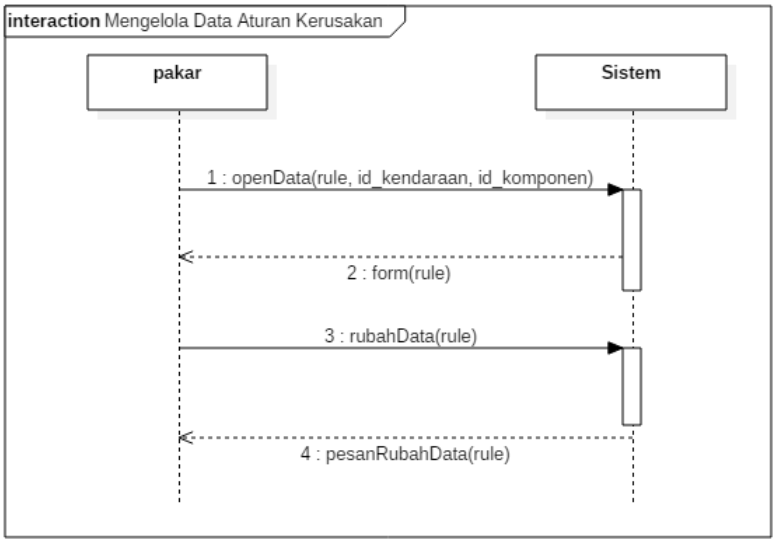
Gambar 4.9 Sequence Diagram Mengelola Data Kendaraan yang diusulkan

3. *Sequence Diagram* Mengelola Data Kerusakan



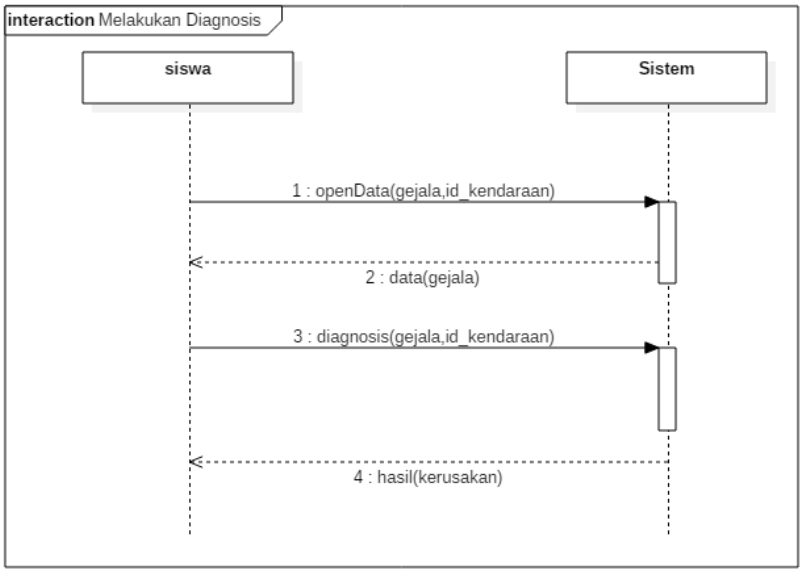
Gambar 4.10 Sequence Diagram Mengelola Data Kerusakan yang diusulkan

4. *Sequence Diagram* Mengelola Data Aturan Kerusakan



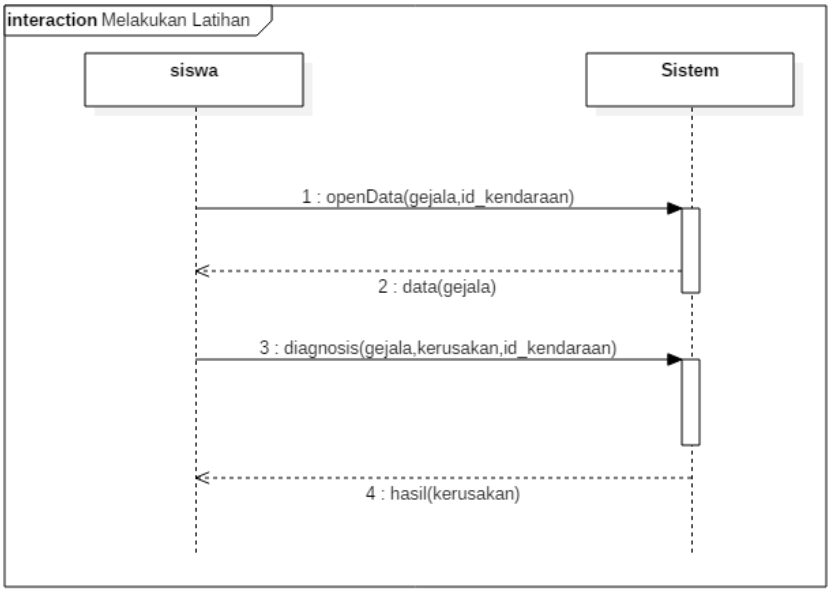
Gambar 4.11 Sequence Diagram Mengelola Data Aturan Kerusakan yang diusulkan

5. *Sequence Diagram* Melakukan Diagnosis Kerusakan



Gambar 4.12 Sequence Diagram Melakukan Diagnosis Kerusakan yang diusulkan

6. *Sequence Diagram* Melakukan Diagnosis Kerusakan

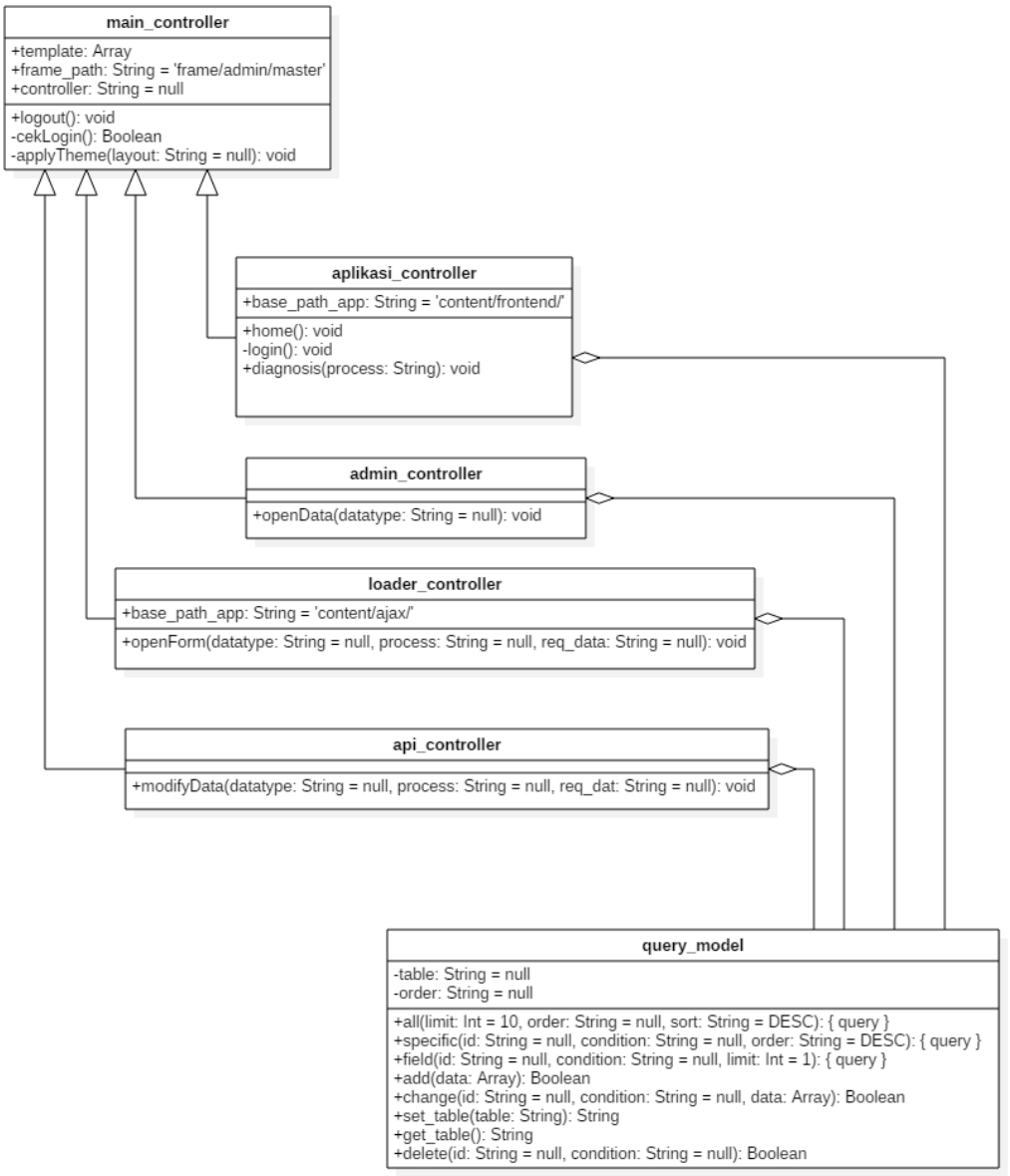


Gambar 4.13 Sequence Diagram Melakukan Latihan Diagnosis Kerusakan yang diusulkan

4.1.4 Perancangan Data

4.1.4.1 *Class Diagram*

Class diagram merupakan representasi dari kegiatan yang ditangani oleh sistem. Dengan class diagram memberikan gambaran tentang sistem dan relasi – relasi yang akan dibuat di dalamnya.

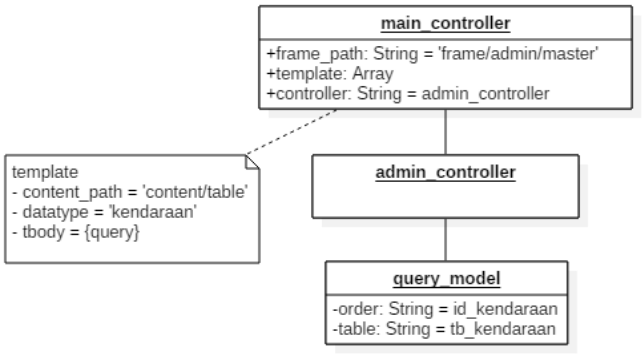


Gambar 4.14 Class Diagram yang diusulkan

4.1.4.2 *Object Diagram*

Object diagram merupakan diagram yang memberikan struktur model sebuah sistem dalam kurun waktu tertentu. *Object diagram* juga lebih konkrit dan dapat menggambarkan sebuah struktur dengan jelas. *Object diagram* dapat digambarkan dengan mengkondisikan sebuah sistem sedang berjalan.

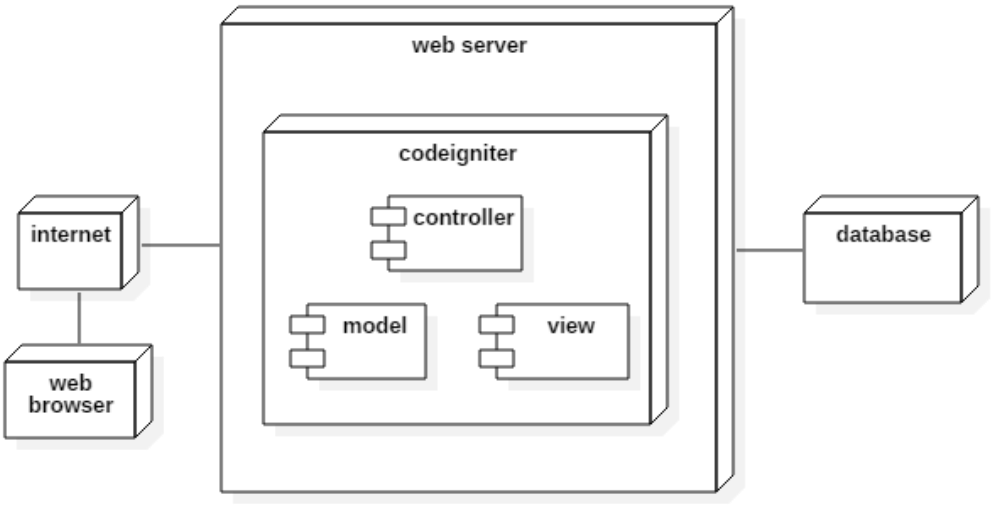
Di bawah ini merupakan *object diagram* ketika user memilih menu data kendaraan.



Gambar 4.15 Object Diagram yang diusulkan

4.1.4.3 *Deployment Diagram*

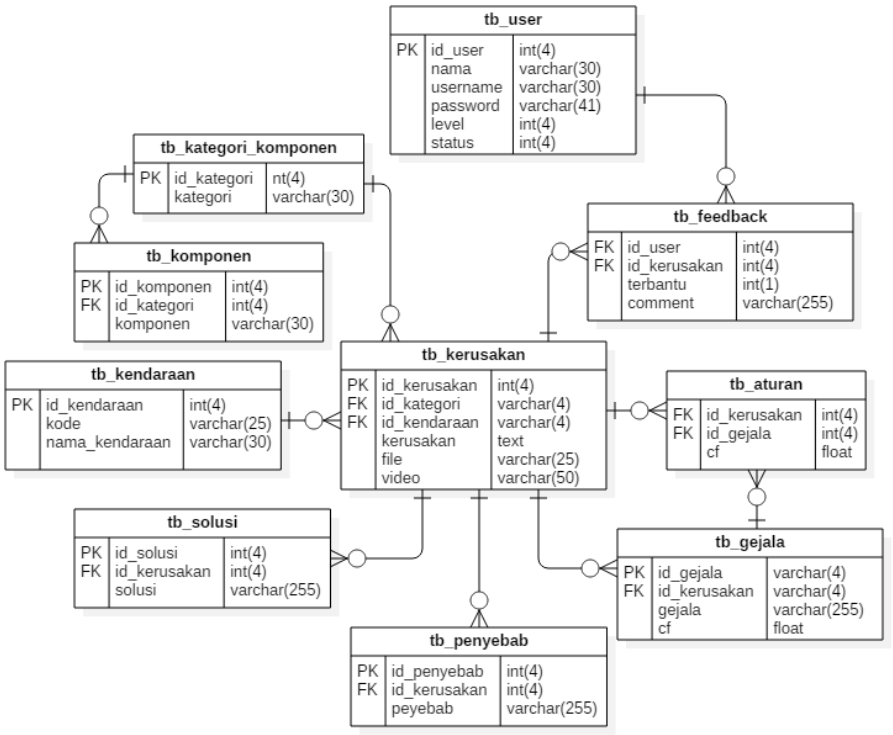
Deployment diagram memperlihatkan setiap hubungan komponen-komponen *hardware* dalam jaringan komputer untuk menyebarkan sistem yang dibangun. Berikut *deployment diagram* untuk sistem yang diusulkan.



Gambar 4.16 Deployment Diagram yang diusulkan

4.1.4.4 *ER Diagram*

ER (Entity Relationship) diagram merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu sistem. Berikut ER diagram dari sistem yang diusulkan.



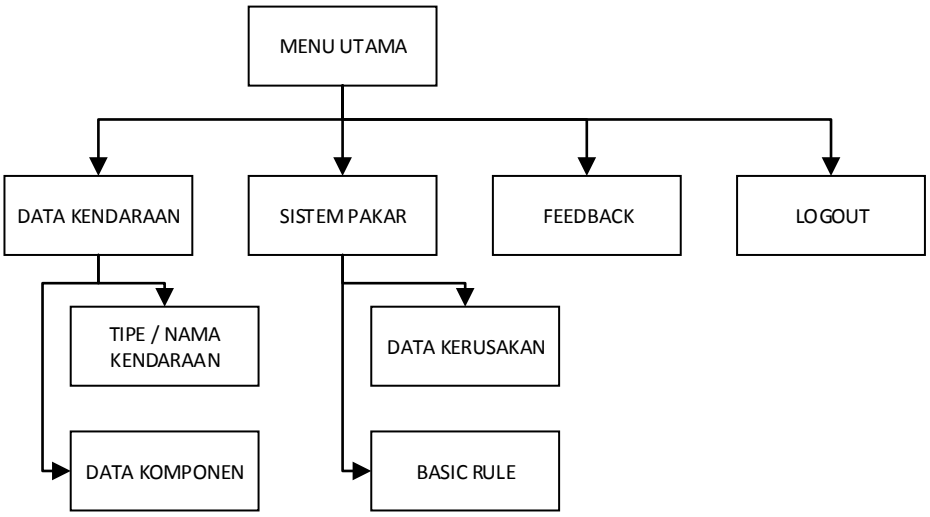
Gambar 4.17 ER Diagram sistem yang diusulkan

4.2 Perancangan Antar Muka

4.2.1 Struktur Menu

Struktur menu merupakan bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer sehingga pada saat menjalankan program komputer, user tidak mengalami kesulitan dalam memilih menu - menu yang diinginkan. Pada perancangan ini dibuat menu untuk pengajar yang dapat mengintegrasikan seluruh data dalam suatu sistem dan disertai dengan instruksi yang ada pada pilihan menu tersebut.

Adapun menu tersebut dapat dilihat pada gambar sebagai berikut:

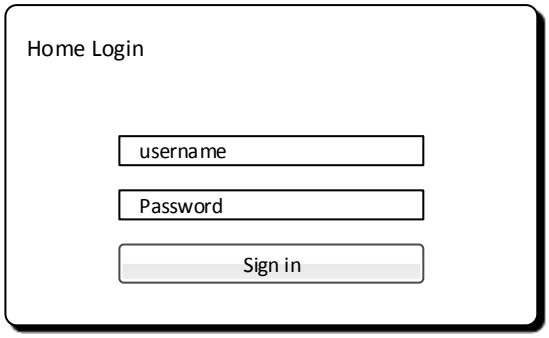


Gambar 4.18 Struktur menu sistem yang diusulkan

4.2.2 Perancangan *Input*

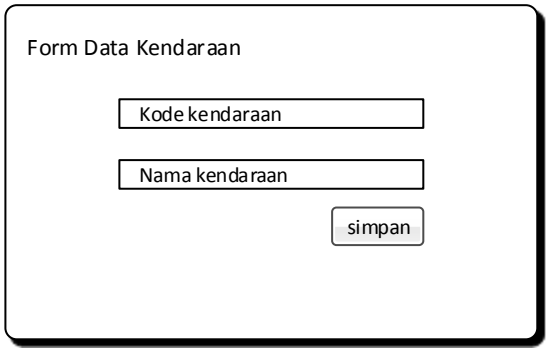
Masukan (*input*) merupakan awal dimulainya suatu proses di dalam sistem, dimana data dimasukan untuk diolah menjadi informasi yang lebih berguna. Perancangan *input* yaitu pembuatan antarmuka (*interface*) untuk menerima masukan dari pengguna sistem, antarmuka untuk *input* ini harus memberikan kejelasan kepada *user* baik bentuk maupun masukan – masukan yang harus diisi.

1. *Form Login*



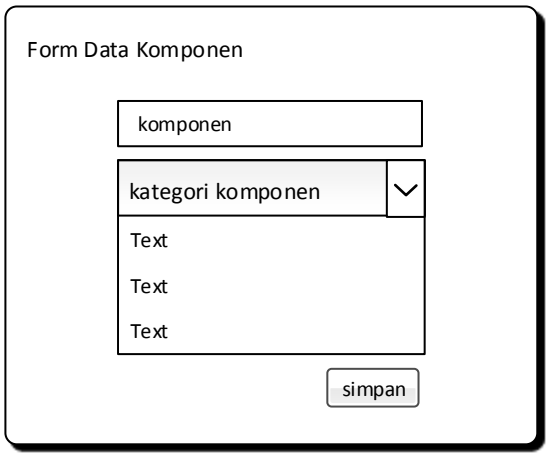
Gambar 4.19 Form Login

2. *Form* Data Kendaraan



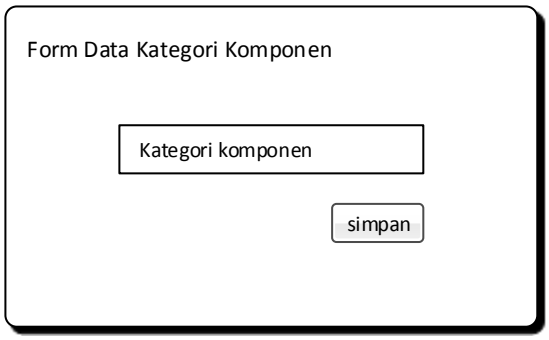
Gambar 4.20 Form Data Kendaraan

3. Form Data Komponen



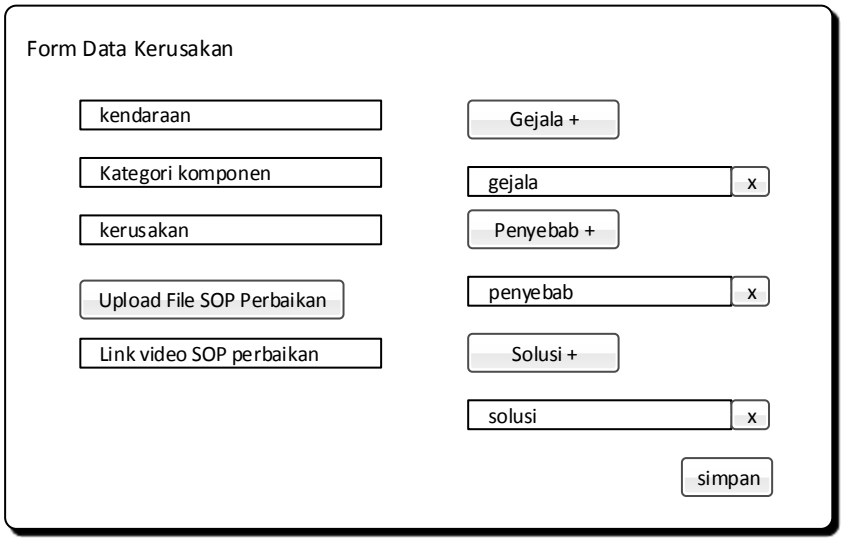
Gambar 4.21 Form Data Komponen

4. *Form* Data Kategori Komponen



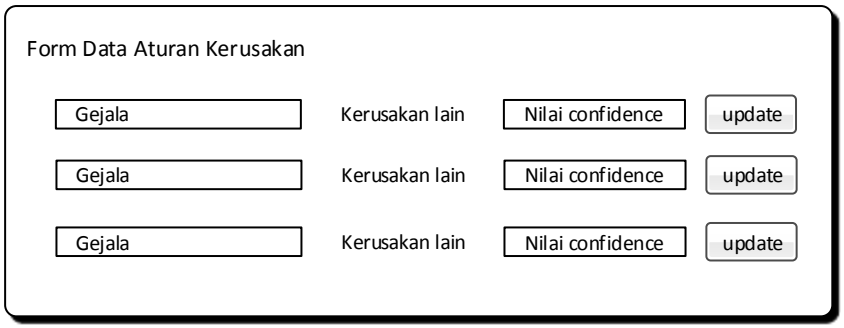
Gambar 4.22 Form Data Kategori Komponen

5. *Form* Data Kerusakan



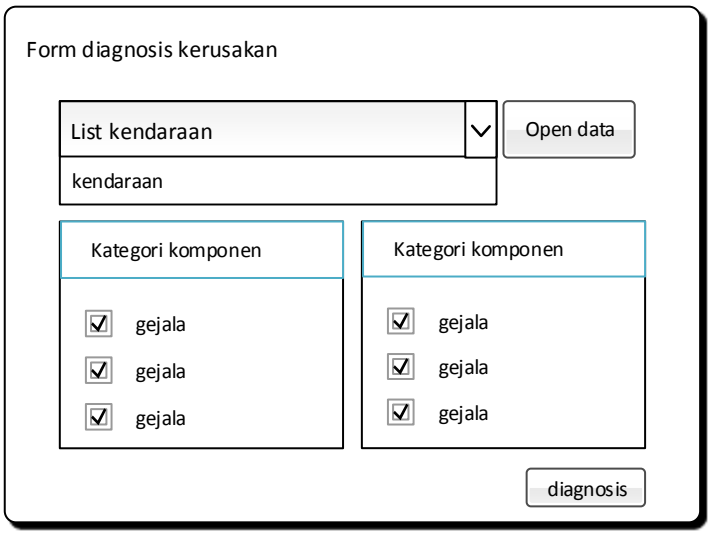
Gambar 4.23 Form Data Kerusakan

6. *Form* Data Aturan Kerusakan



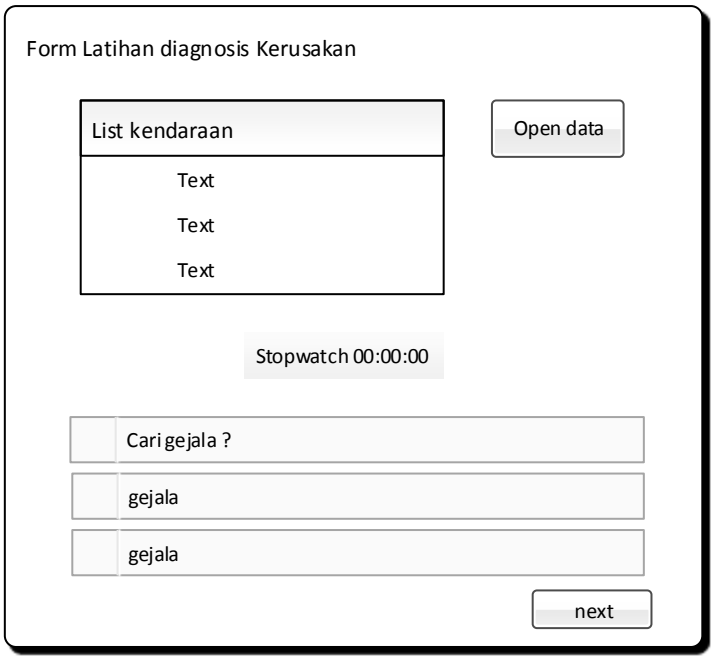
Gambar 4.24 Form Data Aturan Kerusakan

7. *Form* Diagnosis Kerusakan

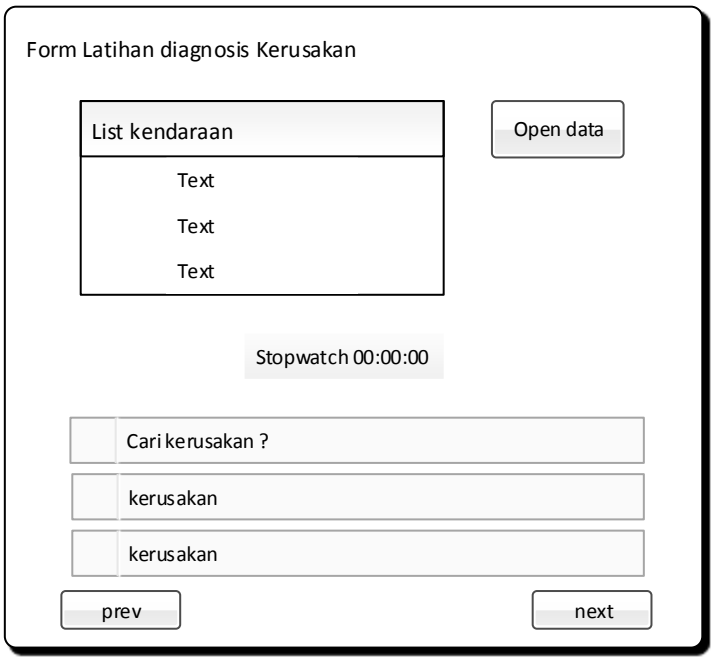


Gambar 4.25 Form Diagnosis Kerusakan

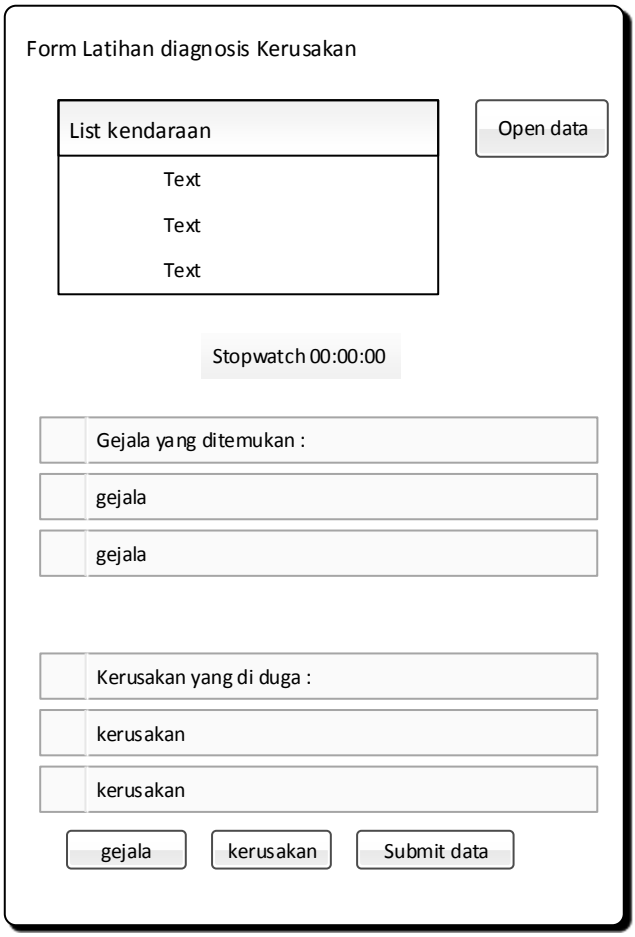
8. *Form* Latihan Diagnosis Kerusakan



Gambar 4.26 Form Pilih Data Gejala Latihan Diagnosis



Gambar 4.27 Form Pilih Data Kerusakan Latihan Diagnosis

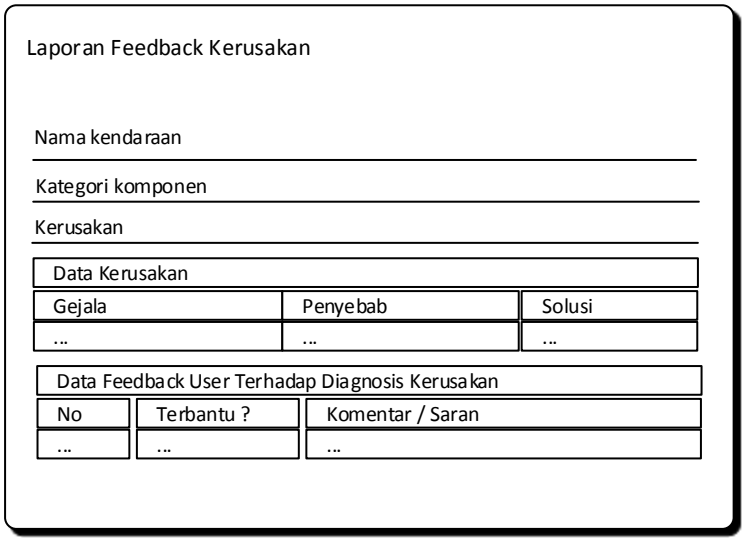


Gambar 4.28 Form Preview Latihan Diagnosis

4.2.3 Perancangan *Output*

Perancangan *output* merupakan informasi yang dihasilkan oleh sistem berupa laporan dari hasil proses masukan yang diterima oleh sistem. Berikut adalah rancangan keluaran dalam gambar berikut.

1. *Feedback* diagnosis kerusakan



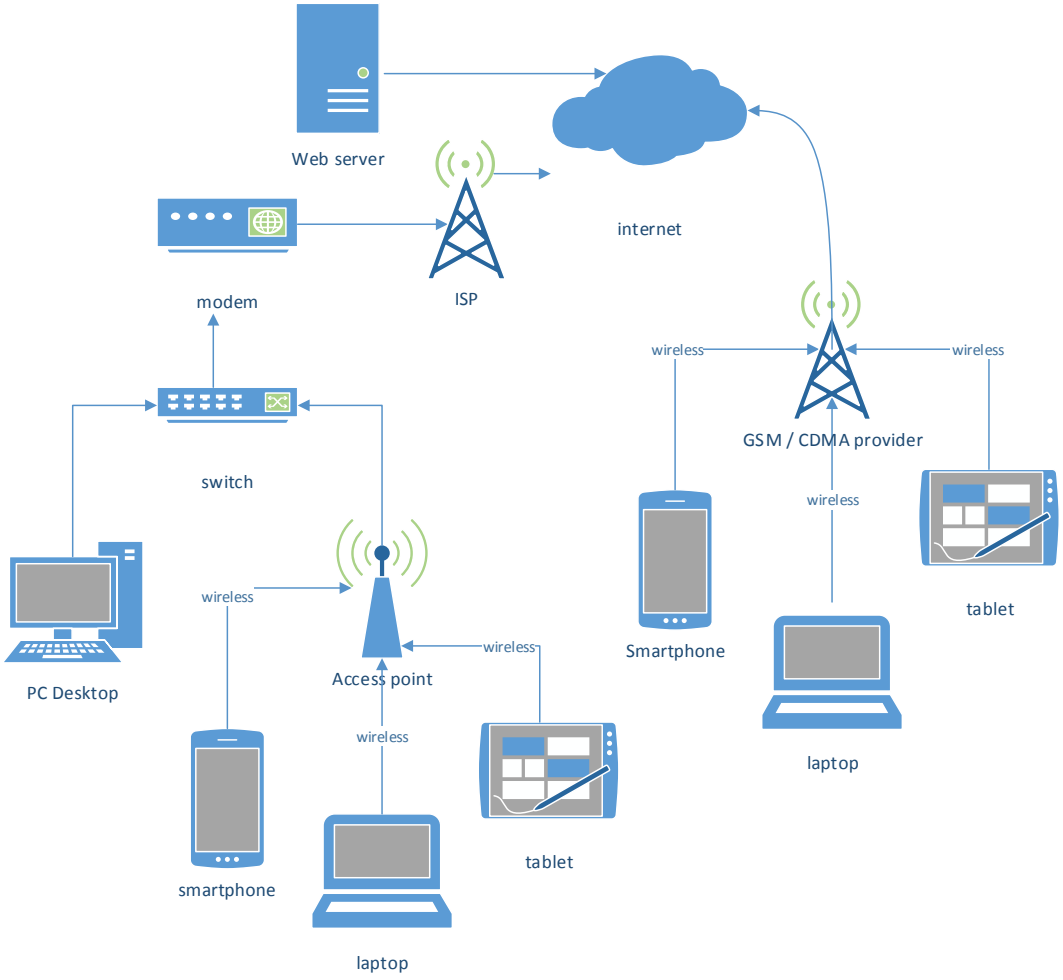
Gambar 4.29 Output Feedback Diagnosis Kerusakan

4.3 Perancangan Arsitektur Jaringan

Topologi jaringan adalah bentuk hubungan dari suatu jaringan. Karena sistem yang dirancang diakses secara *online*, kemungkinan pengaksesan sistem dapat dilakukan melalui perangkat yang berbeda – beda selama perangkat tersebut terkoneksi dengan jaringan internet. Maka dari itu topologi jaringan yang dipakai adalah topologi *hybrid*.

Topologi *hybrid* merupakan topologi jaringan yang terdiri dari beberapa topologi jaringan. Bisa disebut topologi *hybrid* ketika beberapa jaringan komputer saling terkoneksi dengan berbeda bentuk topologi jaringannya. Gambar di bawah

ini akan menunjukan bagaimana topologi *hyrbid* dipakai untuk sistem yang diusulkan.



Gambar 4.30 Perancangan Arsitektur Jaringan yang diusulkan

4.4 Pengujian

Pengujian sistem merupakan tahapan untuk melakukan serangkaian tes dengan mencoba sistem yang telah dibangun yang bertujuan untuk menemukan kesalahan kesalahan atau kekurangan – kekurangan pada perangkat lunak yang diuji.

Dalam melakukan pengujian terhadap sistem pakar diagnosis kerusakan dilakukan dengan dua macam pengujian, yaitu pengujian *black box* dan *white box*.

Adapun pengujian *black box* berfokus pada fungsional dari sebuah sistem yang dibangun dan menemukan kesalahan program, sedangkan pengujian *white box* berfokus kepada struktur internal (*source code*) program untuk mengetahui apakah terjadi *error*.

4.4.1 Rencana Pengujian

Rencana dari pengujian perangkat lunak ini meliputi beberapa pengujian, diantaranya sebagai berikut.

Tabel 4.9 Rencana Pengujian

No	Kelas Uji	Butir Uji	Jenis Pengujian
1.	API_controller	Mengisi kode kendaraan dan nama kendaraan	Black box
		Tidak mengisi kode kendaraan atau nama kendaraan	Black box
		Mengisi nama komponen kendaraan	Black box
		Tidak mengisi nama komponen kendaraan	Black box
		Mengisi data kerusakan	Black box
		Tidak mengisi data kerusakan	Black box
		Mengubah data aturan kerusakan	Black box
		Memasukan nilai salah pada data aturan	Black box
2.	Admin_controller	Export pdf data kerusakan	Black box

3.	Aplikasi_controller	Melakukan diagnosis kerusakan	Black box
		Method diagnosis kerusakan menggunakan teori probabilitas bayesian	White box

4.4.2 Kasus dan Hasil Pengujian

4.4.2.1 Pengujian White Box

Pengujian pada method diagnosis dengan teori probabilitas bayesian menggunakan pengujian *white box*, hal tersebut dilakukan untuk menguji dan mengukur kualitas dari algoritma berdasarkan koding yang telah dibuat. Berikut tahapan – tahapan pada pengujian method diagnosis dengan teori probabilitas bayesian :

1. Mengubah bentuk kode menjadi *flowchart* kemudian menjadi *flowgraph*

   a. Kode

```
    diagnosis(dataKomponen, kendaraan) {

    // cek apakah user memilih gejala ?
1   if (dataKomponen != NULL) then
2    i = 0;
2    for i+1 to dataKomponen do

       // cek jika ada atau tidak data gejala di input oleh user
       // berdasarkan kategori komponen
3      if (dataKomponen[i]['gejala'] != null) then

         // mengambil data pembagi ( sum((cf*0.33)) ) dari tiap kerusakan
         // di tb aturan, id kerusakan, dan kategori komponen berdasarkan id
         // kategori komponen dan id kendaraan di tb kerusakan
         // id kategori komponen = dataKomponen[i]['kategori']
         // id kendaraan         = kendaraan
4        dataKerusakan = {query}

         // mengambil nilai condition factor (cf) setiap gejala
         // untuk dijadikan pembilang berdasarkan data kerusakan
```

```
    // disini teori probabilitas bayesian mulai di implementasikan
5   j = 0;
5   for j+1 to dataKerusakan do
     // set nilai awal pembilang
6    pembilang = 0.0;

     // ambil nilai pembagi dari tb_gejala berdasarkan
     // dataKerusakan[j][‘id_kerusakan’]
6    p_pembagi = {query}

     // tambahkan nilai pembagi p_gejala kedalam pembagi
     // dataKerusakan[j][‘pembagi’]
6    dataKerusakan[j][ pembagi] = dataKerusakan[i][‘pembagi’] +
p_pembagi ;

7    k = 0;
7    for k+1 to dataKomponen[i]['gejala'] do

      // ambil nilai condition factor (kolom cf*0.33) di tabel gejala
      // berdasarkan id gejala dan id kerusakan
      // id gejala       = datakomponen[i]['gejala'][k]
      // id kerusakan = dataKerusakan[j]['id_kerusakan']
8     cf = {query}

      // jika nilai condition factor (kolom cf*0.33) tidak ditemukan
      // di tabel gejala maka akan mencari di tabel aturan
      // berdasarkan id gejala dan id kerusakan
8     if(cf == NULL) then
8      cf = {query}
      endif

      // tambahkan nilai pembilang dengan nilai cf yang di dapat
9     pembilang = pembilang + cf;
     endfor

     // lakukan perhitungan diagnosis berdasarkan
     // rumus teori probabilitas bayesian
10   diagnosis = pembilang / (dataKerusakan[j]['pembagi']) * 100;

     // masukan nilai hasil perhitungan untuk di urutkan
10   dataHasil[i]['diagnosis'][j]['probabilitas'] = diagnosis;

     // lakukan pengurutan dari nilai terbesar ke terkecil
     // di tiap kerusakan pada tiap kategori komponen
11   l = 0;
11   for l+1 to dataHasil['diagnosis'] do
```

```
        // cek apakah ada data hasil sebelumnya ?
12      if(dataHasil['diagnosis'][l-1] != NULL) then
          // simpan nilai di temp var
13        old = dataHasil[i]['diagnosis'][l-1];
13        new = dataHasil[i]['diagnosis'][j];

14        if(new['probabilitas'] > old['probabilitas']) then

           // tukar nilai old dan new
15         dataHasil[i]['diagnosis'][l-1] = new;
15         dataHasil[i]['diagnosis'][j]     = old;

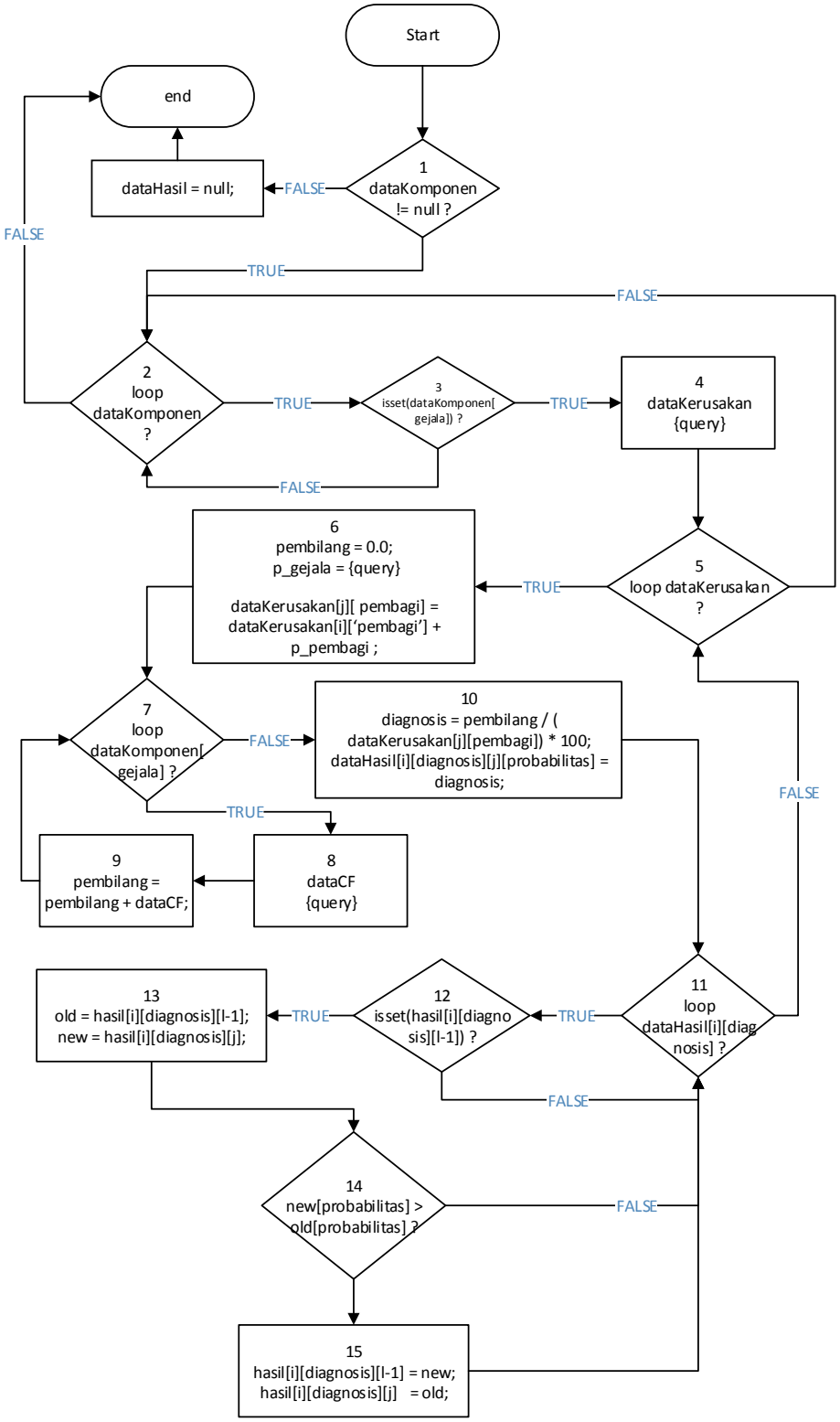
          endif
         endif
        endfor
       endfor
      endif
     endfor

    else :
16   dataHasil = NULL;
    endif

17  return dataHasil;
    }
```

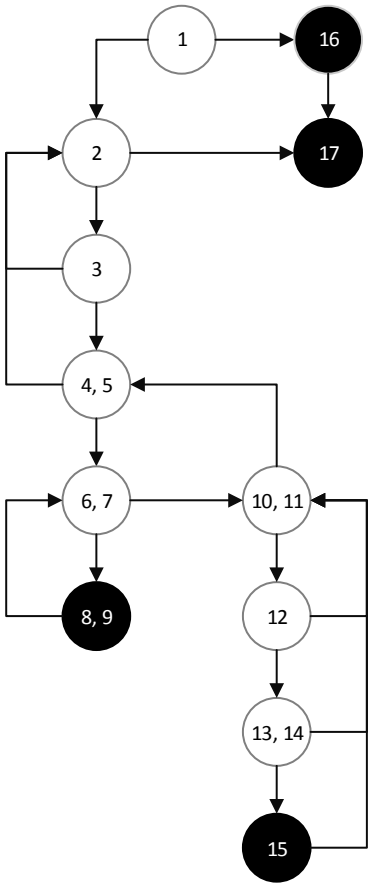
Gambar 4.31 Kode method dengan Teori Probabilitas Bayesian

b. Flowchart



Gambar 4.32 Flowchart Method dengan Teori Probabilitas Bayesian

c. Flowgraph



Gambar 4.33 Flowgraph Method dengan Teori Probabilitas Bayesian yang dibuat

2. Cyclometic complexity

Berdasarkan *flowgraph* terdapat data yang didapat yaitu :

a. Independent Path

1. Path 1 = 1 – 16 - 17
2. Path 2 = 1 – 2 – 17
3. Path 3 = 1 – 2 – 3 – 2 – 17
4. Path 4 = 1 – 2 – 3 – 4 – 5 – 2 – 17
5. Path 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 10 – 11 – 4 – 5 – 2 – 17

6. Path 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 4 – 5 – 2 - 17
7. Path 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 6 – 7 – 10 – 11 – 12 – 10 – 11 – 4 – 5 – 2 – 17
8. Path 8 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 6 – 7 – 10 – 11 – 12 – 13 – 14 – 10 – 11 – 4 – 5 – 2 – 17
9. Path 9 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 6 – 7 – 10 – 11 – 12 – 13 – 14 – 15 – 10 – 11 – 4 – 5 – 2 – 17

b. Jumlah *edge* = 19 *edge*

c. Jumlah *node* = 12 *node*

d. Jumlah predikat *node* = 8 predikat *node*

e. Jumlah *region* = 9 *region*

Berdasarkan data yang didapat, maka dapat dihitung nilai *cyclometic complexity* dari algoritma yang dibuat. Berikut perhitungannya:

a. V(G) = E – N + 2

   V(G) = 19 *edge* – 12 *node* + 2

   V(G) = 9

b. V(G) = P + 1

   V(G) = 8 predikat *node* + 1

   V(G) = 9

3. Graph Matrix

Tabel 4.10 Graph Matrix Method dengan Teori Probabilitas Bayesian yang dibuat

N	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	
1		1														1	1
2			1													1	1
3		1		1													1
4					1												
5		1				1											1
6							1										
7								1		1							1
8						1											
9						1											
10											1						
11					1							1					1
12											1		1				1
13														1			
14															1	1	1
15											1						
16																	
																Jumlah +1	9

4. Pengujian *Path Method* dengan Teori Probabilitas Bayesian yang dibuat

Tabel 4.11 Pengujian Path Method dengan Teori Probabilitas Bayesian yang dibuat

No	Kasus Uji	Hasil yang diharapkan	Hasil uji kasus	Keterangan
1	dataKomponen = Null	Diagnosis tidak dilakukan	dataKomponen = Null diagnosis tidak dilakukan	Alur terlewati
2	memberikan dataKomponen = 6 for i+1 to 5 do endofor	keluar dari perulangan	dataKomponen = 6 keluar dari perualangan	Alur terlewati
3	Jika siswa tidak memilih gejala dalam suatu kategori komponen	melewati proses dan melakukan perulangan ke index dataKomponen selanjutnya	dataKomponen[i][‘gejala’] = null melewati proses dan melakukan perulangan ke index dataKomponen selanjutnya	Alur terlewati

	dataKomponen [i][‘gejala’] = null			

4	memberi dataKerusakan = 5
	
	for j+1 to 4 do
	
	endfor
		melewati perulangan dan melanjutkan ke index dataKomponen selanjutnya	dataKerusakan = 5 melewati perulangan dan melanjutkan ke index dataKomponen selanjutnya	Alur terlewati

5	Proses perulangan dataKerusakan	mendapatkan nilai probabilitas kerusakan dan melanjutkan perulangan ke index dataKerusakan berikutnya	Mendapatkan nilai probabilitas kerusakan dan melanjutkan perulangan ke index dataKerusakan berikutnya	Alur terlewati

6	memberi dataKomponen [i][‘gejala’] = 6
	
	for k+1 to 5 do
	
	endfor
		melewati perulangan dan menghitung pembilang dan penyebut sesuai dengan rumus persamaan bayes	melewati perulangan dan mendapatkan hasil diagnosis	Alur terlewati

7	Proses perulangan dataKomponen [i][‘gejala’]	Mendapatkan nilai cf dan menjumlahkan pembilang dengan nilai cf yang didapat	nilai pembilang berubah	Alur terlewati

8	memberi dataHasil[i][‘diagnosis’] = 6
	
	for l+1 to 5 do
	
	endfor
		Melewati perulangan dan melanjutkan perulangan dataKerusakan ke index berikutnya	Melewati perulangan dan melanjutkan perulangan dataKerusakan ke index berikutnya	Alur Terlewati

9	Proses perulangan	Melakukan pengurutan	Mendapatkan hasil diagnosis yang terurut	Alur terlewati

	datahasil[i][‘diagnosis’]	hasil perhitungan probabilitas kerusakan dari terbesar ke terkecil berdasarkan kategori komponen	dari terbesar hingga terkecil berdasarkan kategori komponen	

5. Penelusuran Data ketika *user* memilih gejala kerusakan

a. Kendaraan dipilih : motor konvensional 4T tipe bebek [id_kendaraan = 1]

b. Gejala yang dipilih :

1. Pengereman kurang pakem [id_gejala = 23]
2. Terjadi bunyi pada saat di rem [id_gejala = 25]
3. Mesin bersuara mengelitik [id_gejala = 20]
4. Suara kasar pada kepala silinder [id_gejala = 1]

Tabel 4.12 Penelusuran Data

kendaraan	dataKomponen				dataKerusakan					pembilang	cf	diagnosis	dataHasil		
	i	Gejala			j	pembagi	id_kerusakan	kerusakan	kategori komponen				i	diagnosis	
		k	id_gejala	kategori										j	probabilitas
1	0	0	20	1	0	1.6500000196695	8	kompresi mesin lemah	mesin	0.264	0.264	22	0	0	22
		1	1	1						0.363	0.099				
		0	20	1	1	2.2109999960661	1	kerusakan pada Piston	mesin	0.099	0.099	17.91		1	17.91
		1	1	1						0.396	0.297				
		0	20	1	2	1.5840000236034	3	kerusakan pada klep	mesin	0.099	0.099	12.5		2	12.5
		1	1	1						0.198	0.099				
	1	-	-		-	-	-	-		-	-	-	-	-	
	2	-	-		-	-	-	-		-	-	-	-	-	
	3	0	23	5	0	0.66	9	rem tidak pakem	sistem rem tromol	0.297	0.264	55	3	0	55
		1	25	5						0.363	0.099				
		0	23	5	1	0.6929999819828	10	tromol macet	sistem rem tromol	0.0	0.0	42.86	3	1	42.86
		1	25	5						0.297	0.297				

4.4.2.2 Pengujian *Black Box*

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut, diantaranya sebagai berikut.

Tabel 4.13 Pengujian *Black Box*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data berupa nama kendaraan dan kode kendaraan	Menampilkan informasi data tersimpan	Nama kendaraan dan kode terdapat dalam list data kendaraan	OK
Data berupa nama komponen kendaraan	Menampilkan informasi data tersimpan	Nama komponen kendaraan terdapat pada list data komponen kendaraan	OK
Data kerusakan baru	Menampilkan informasi data tersimpan	Data kerusakan terdapat pada list data kerusakan	OK
Mengubah data aturan diagnosis kerusakan	Menampilkan data berhasil dirubah	Menampilkan data berhasil dirubah	OK
Kasus dan Hasil Uji (Data Salah)			
Nama kendaraan atau kode kendaraan tidak di isi	Menampilkan informasi harus di isi	Menampilkan informasi kode kendaraan atau nama kendaraan harus di isi	OK

Nama komponen tidak di isi	Menampilkan informasi harus di isi	Menampilkan informasi nama komponen harus di isi	OK
Nama kerusakan tidak di isi	Menampilkan informasi harus di isi	Menampilkan informasi nama kerusakan harus di isi	OK
Memasukan nilai aturan diatas 1 atau di bawah 0 atau huruf atau simbol	Menampilkan informasi data gagal di simpan	Menampilkan data gagal di simpan	OK
Memilih data kendaraan dan memilih gejala	Menampilkan data gejala dan menampilkan hasil diagnosis kerusakan berdasarkan gejala yang dipilih	Menampilkan data gejala dan menampilkan hasil diagnosis kerusakan berdasarkan gejala yang dipilih	OK

4.4.3 Kesimpulan Hasil Pengujian

Kesimpulan hasil pengujian merupakan penjelasan mengenai suatu pemikiran yang diambil berdasarkan data pengujian yang dilakukan. Berdasarkan pengujian yang dilakukan secara keseluruhan fungsional dari sistem berjalan sesuai dengan yang diharapkan tidak ada kesalahan baik proses dan secara fungsional.

4.5 Implementasi

Implementasi atau tahapan penerapan adalah suatu bentuk kegiatan yang merupakan rangkaian lanjutan dari kegiatan perancangan perangkat lunak. Implementasi dimaksudkan sebagai usaha untuk mewujudkan hasil dari perancangan perangkat lunak. Hasil ini berarti hasil dari proses aplikasi yang telah dibuat dan sudah dapat digunakan serta berfungsi dengan baik.

4.5.1 Implementasi Perangkat Lunak

Untuk membangun sistem yang diusulkan, penulis menggunakan bahasa pemrograman basis *web* yaitu *PHP, CSS,* Javascript dengan *framework* Codeigniter versi 2.2.0 untuk *PHP,* INK versi 3.10 untuk *CSS,* dan Jquery versi 1.10.1 untuk Javascript.

Sublime text versi 3 penulis gunakan untuk pengkodean program, software paketan dari apache yaitu XAMPP ver 1.8.3 untuk menjalankan aplikasi server lokal berupa apache server dan aplikasi database seperti MySQL versi 5 agar penulis dapat melakukan testing terhadap program yang diusulkan selama tahap pembangunan dan *browser chrome* untuk pengaksesan secara offline dari program yang dibuat.

4.5.2 Implementasi Perangkat Keras

Adapun spesifikasi minimum perangkat keras yang digunakan untuk menjalankan aplikasi yang digunakan untuk menjalankan program yang dibangun dapat digunakan dalam keadaan *offline*. Berikut spesifikasinya :

1. *Processor* dengan *dual core* 1 Ghz *clockspeed*
2. 2 Gb RAM

3. 1 Gb *hardisk space*.

4.5.3 Implementasi Basis Data

Pembuatan basis data, dilakukan dengan menggunakan aplikasi MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut :

Tabel 4.14 Implementasi Basis Data

No	Nama Tabel	SQL
1.	tb_kendaraan	CREATE TABLE IF NOT EXISTS `tb_kendaraan` (`id_kendaraan` int(4) NOT NULL, `kode` varchar(25) NOT NULL, `nama_kendaraan` varchar(100) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
2.	tb_user	CREATE TABLE IF NOT EXISTS `tb_user` (`id_user` int(4) NOT NULL, `nama` varchar(30) NOT NULL, `username` varchar(25) NOT NULL, `password` varchar(41) NOT NULL, `email` varchar(50) NOT NULL, `level` int(4) NOT NULL, `status` int(4) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
3.	tb_kerusakan	CREATE TABLE IF NOT EXISTS `tb_kerusakan` (`id_kerusakan` int(4) NOT NULL, `id_kategori` int(4) NOT NULL, `id_kendaraan` int(4) NOT NULL,

		`kerusakan` text NOT NULL, `file` varchar(50) DEFAULT NULL, `video` varchar(50) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
4.	tb_gejala	CREATE TABLE IF NOT EXISTS `tb_gejala` (`id_gejala` int(11) NOT NULL, `id_kerusakan` int(4) NOT NULL, `gejala` varchar(250) NOT NULL, `cf` float NOT NULL DEFAULT '0.9') ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
5.	tb_solusi	CREATE TABLE IF NOT EXISTS `tb_solusi` (`id_solusi` int(4) NOT NULL, `id_kerusakan` int(4) NOT NULL, `solusi` varchar(200) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
6.	tb_penyebab	CREATE TABLE IF NOT EXISTS `tb_penyebab` (`id_penyebab` int(4) NOT NULL, `id_kerusakan` int(4) NOT NULL, `penyebab` varchar(250) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

7.	tb_kategori_komponen	CREATE TABLE IF NOT EXISTS `tb_kategori_komponen` (`id_kategori` int(4) NOT NULL, `kategori_komponen` varchar(100) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
8.	tb_komponen	CREATE TABLE IF NOT EXISTS `tb_komponen` (`id_komponen` int(4) NOT NULL, `id_kategori` int(4) NOT NULL, `nama_komponen` varchar(100) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
9.	tb_aturan	CREATE TABLE IF NOT EXISTS `tb_aturan` (`id_gejala` int(4) NOT NULL, `id_kerusakan` int(4) NOT NULL, `cf` float NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;
10.	tb_feedback	CREATE TABLE IF NOT EXISTS `tb_feedback` (`id_user` int(4) NOT NULL, `id_kerusakan` int(4) NOT NULL, `terbantu` int(1) NOT NULL, `komen` varchar(255) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.5.4 Impelementasi Antar Muka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan dari program yang dibangun dan pengkodeannya berbentuk *file program*.

Berikut ini adalah file – file yang digunakan dalam membuka tampilan antar muka pengguna.

Tabel 4.15 Implementasi Antar Muka

No	Nama Antarmuka	Deskripsi	Nama File Executable
1.	Data Kendaraan	Menampilkan informasi data kendaraan dan pengelolaan kendaraan	t_kendaraan.php
			f_add_kendaraan.php
2.	Data Komponen	Menampilkan informasi data komponen kendaraan dan pengelolaan data komponen kendaraan	t_komponen.php
			f_add_kategori.php
			f_add_komponen.php
3.	Data Kerusakan	Menampilkan informasi data kerusakan dan pengelolaan kerusakan	t_kerusakan.php
			f_add_kerusakan.php
4.	Data Aturan Diagnosis Kerusakan	Menampilkan informasi data aturan diagnosis kerusakan dan pengelolaan data aturan diagnosis kerusakan	t_rule.php
			f_add_rule.php

5.	Data Feedback User	Menampilkan informasi feedback user terhadap kerusakan dan export data kerusakan ke file pdf	t_feedback.php
			f_trigger_feedback.php
6.	Diagnosis Kerusakan	Menampilkan data gejala untuk proses diagnosis dan hasil dari diagnosis kerusakan tersebut	diagnosis.php
			result.php

4.5.5 Implementasi Instalasi Program

Implementasi instalasi program merupakan aspek awal untuk menjelaskan bagaimana perancangan aplikasi Sistem Pakar Diagnosis Kerusakan Kendaraan dapat dijalankan di komputer. Berikut langkah-langkah yang dilakukan dalam melakukan proses instalasi program agar berjalan pada kondisi *offline*.

1. Pastikan sudah memiliki software XAMPP untuk servernya.
2. Lakukan konfigurasi Software XAMPP (xampp-control.exe), dengan mengaktifkan Apache dan MySQL.
3. Konfigurasi database yang ada pada PhpMyAdmin, dengan menuju link di *browser*, http://localhost/phpmyadmin/.
4. Konfigurasi penyimpanan data *web* Sistem Pakar Diagnosis Kerusakan Kendaraan secara *offline* pada folder htdocs di folder instalan XAMPP.
5. Konfigurasi koneksi database, pada folder application – config – database.php . Isi *hostname, username, password* dan nama databasenya.

Setelah proses instalasi selesai semua maka untuk mengakses menu utama sistem Pakar Diagnosis Kerusakan Kendaraan menggunakan alamat: http://localhost/sipadik2.

Atau dapat diakses secara *online* dengan mengakses alamat website : http://sipadik2.esy.es.

4.5.6 Penggunaaan Program

1. Halaman *Login User*

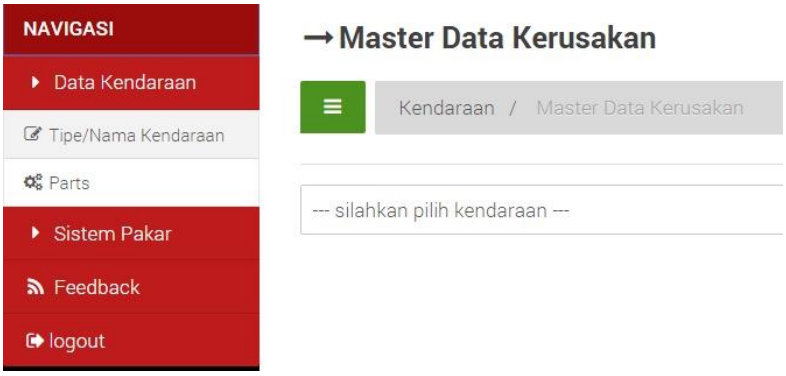
   Untuk penggunaan program pertama – tama mengisi *username* dan *password* sebagai pakar untuk melakukan *login*.



Gambar 4.34 Login Form User

2. Halaman Data Kendaraan

   Setelah *user* melakukan *login* sebagai pakar, pilih menu data kendaraan untuk memasukan data kendaraan terlebih dahulu.



Gambar 4.35 Menu User Pakar

Setelah masuk ke halaman data kendaraan pilih tombol *new* untuk memasukkan data kendaraan.



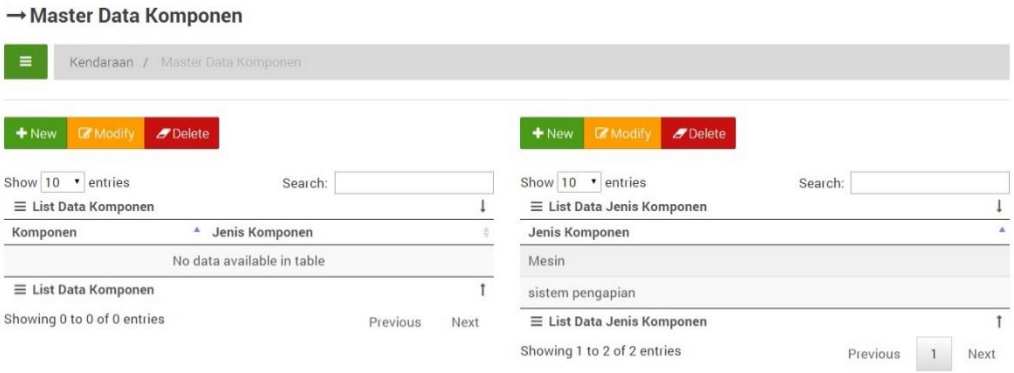
Gambar 4.36 Halaman Data Kendaraan



Gambar 4.37 Form Data Kendaraan

3. Halaman Data Komponen

Setelah data kendaraan dimasukan, masuk ke halaman data komponen kendaraan lalu klik tombol *new* dan masukan data kategori komponen kendaraan.



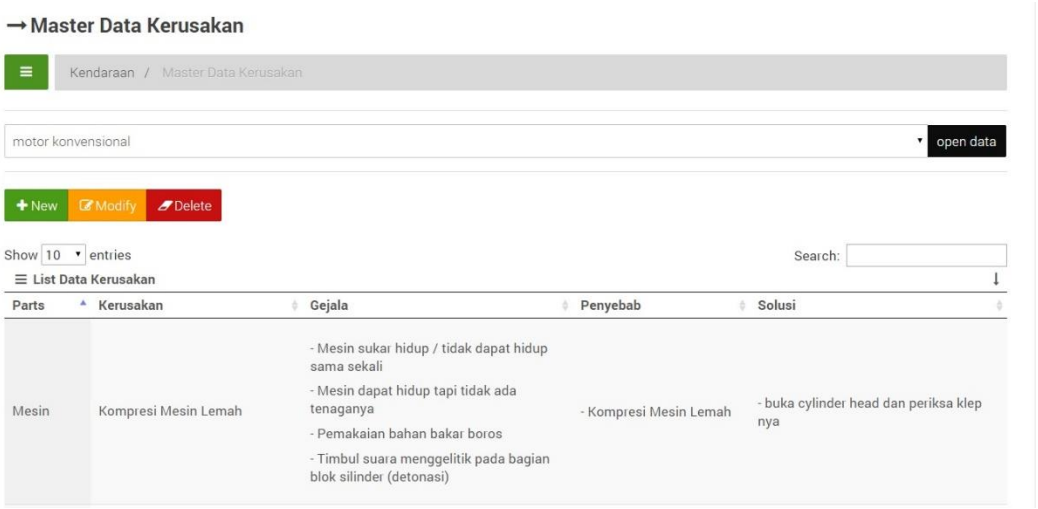
Gambar 4.38 Halaman Data komponen



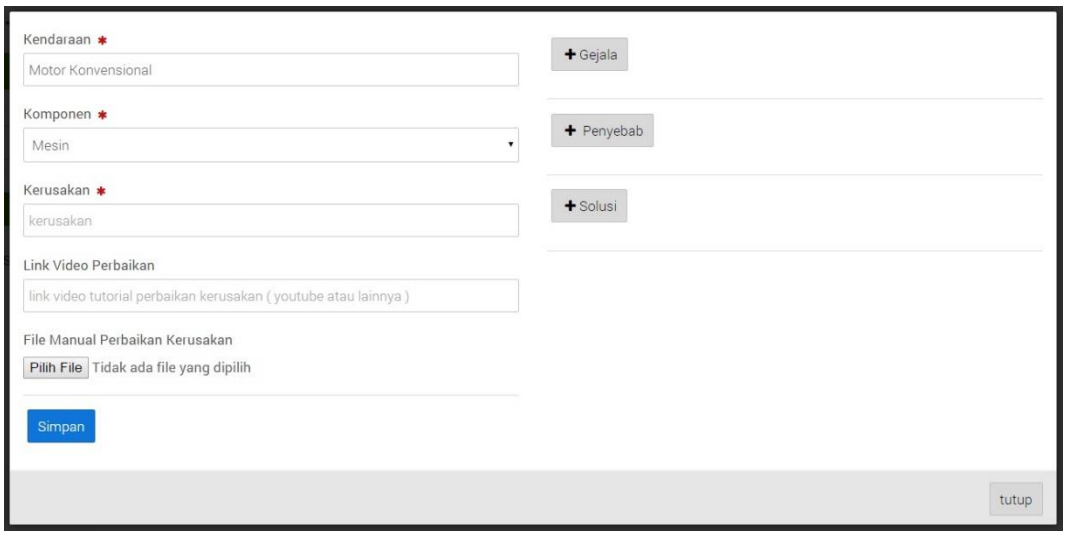
Gambar 4.39 Form Kategori Komponen Kendaraan

4. Halaman Data Kerusakan

Setelah data kendaraan dan kategori dimasukan, masuk ke halaman data kerusakan dengan memilih menu data kerusakan lalu pilih kendaraan yang ada dan klik tombol *new* untuk membuat data kerusakan baru.



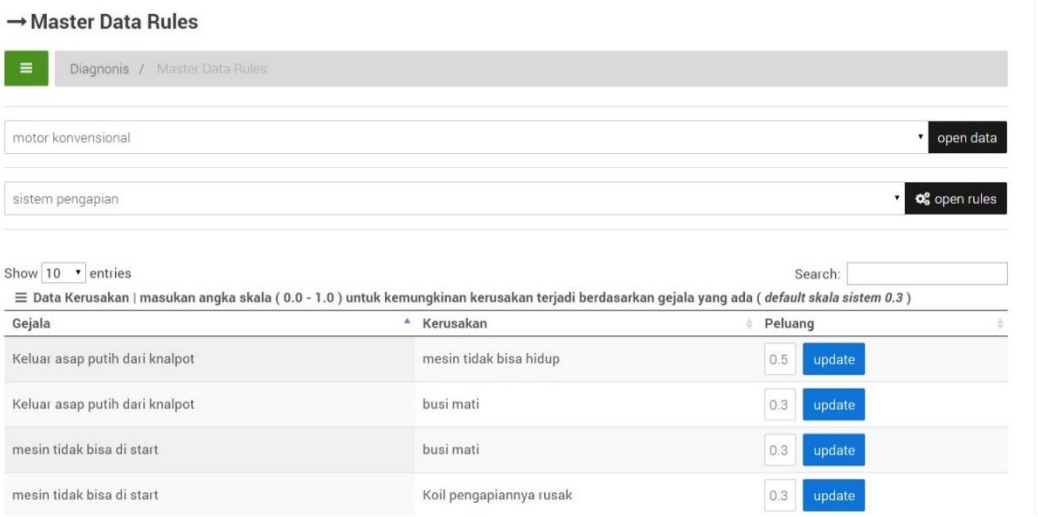
Gambar 4.40 Halaman Data Kerusakan



Gambar 4.41 Form Data Kerusakan Baru

5. Halaman Data Aturan Kerusakan

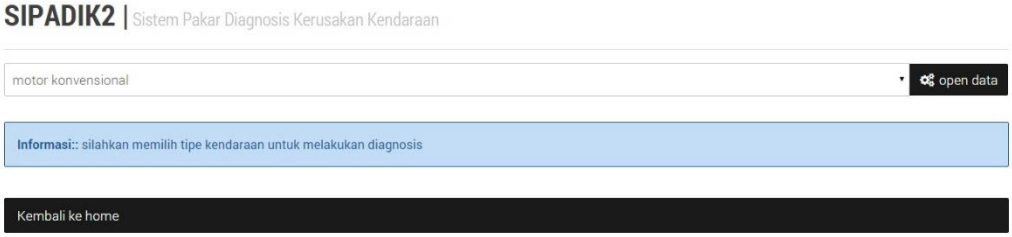
   Setelah data kerusakan dimasukan, masuk ke halaman data aturan kerusakan dengan memilih menu *basic rule*. Pilih kendaraan yang ada lalu pilih kerusakan dan masukan nilai data aturan tiap gejala kerusakan yang dipilih terhadap kerusakan lainnya dengan nilai dari skala 0 sampai 1.



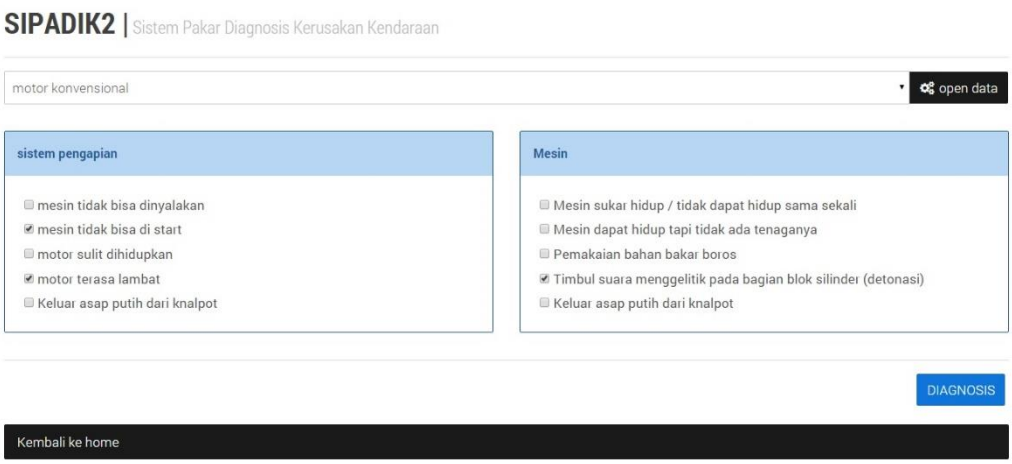
Gambar 4.42 Input Data Aturan Kerusakan

6. Halaman Diagnosis Kerusakan Kendaraan

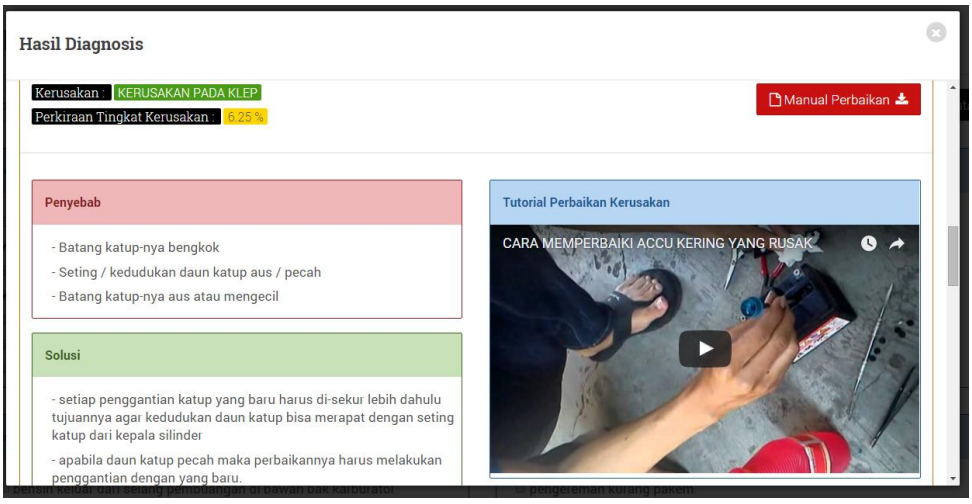
   Apabila data kerusakan dan aturan kerusakan sudah tersimpan, maka diagnosis kerusakan dapat dilakukan. *Login* sebagai siswa untuk dapat masuk ke halaman diagnosis kerusakan lalu pilih kendaraan dan gejala – gejala kerusakan yang ada, tekan tombol diagnosis untuk mendapatkan hasil diagnosis kerusakan berdasarkan gejala yang dipilih. *User* pun dapat memberikan *feedback* terhadap hasil diagnosis kerusakan yang ada dengan mengisi form *feedback* di tiap kerusakan yang muncul.



Gambar 4.43 Halaman Diagnosis Kerusakan Kendaraan



Gambar 4.44 Data Diagnosis Gejala Kerusakan Kendaraan



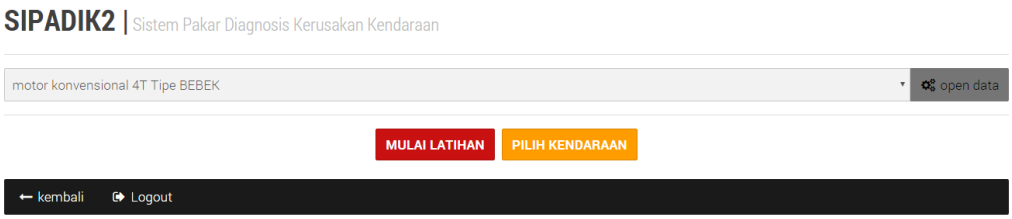
Gambar 4.45 Hasil Diagnosis Kerusakan



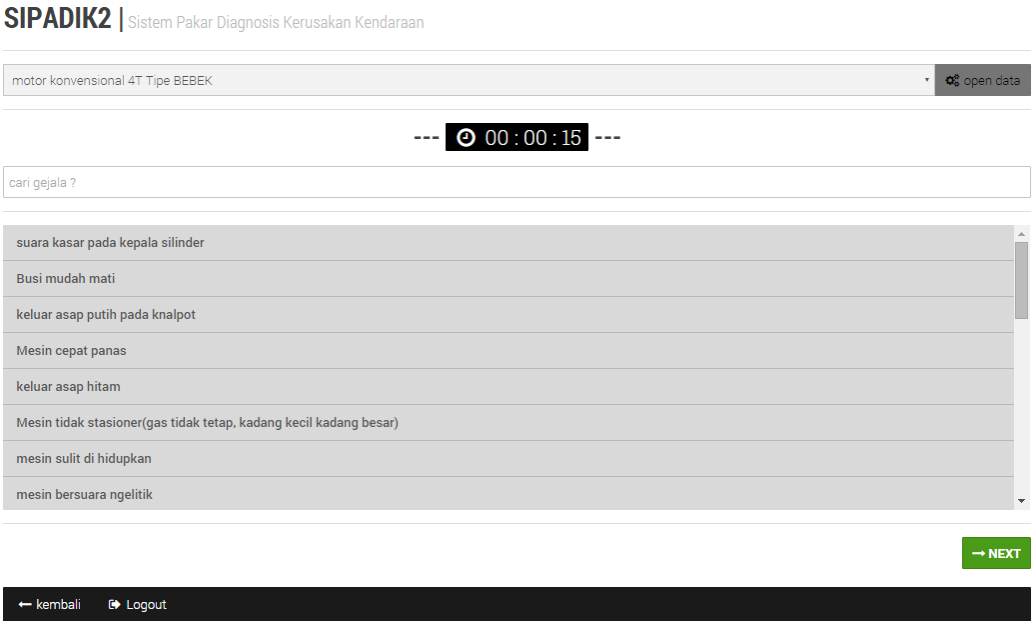
Gambar 4.46 Form Feedback User

7. Halaman Latihan Diagnosis

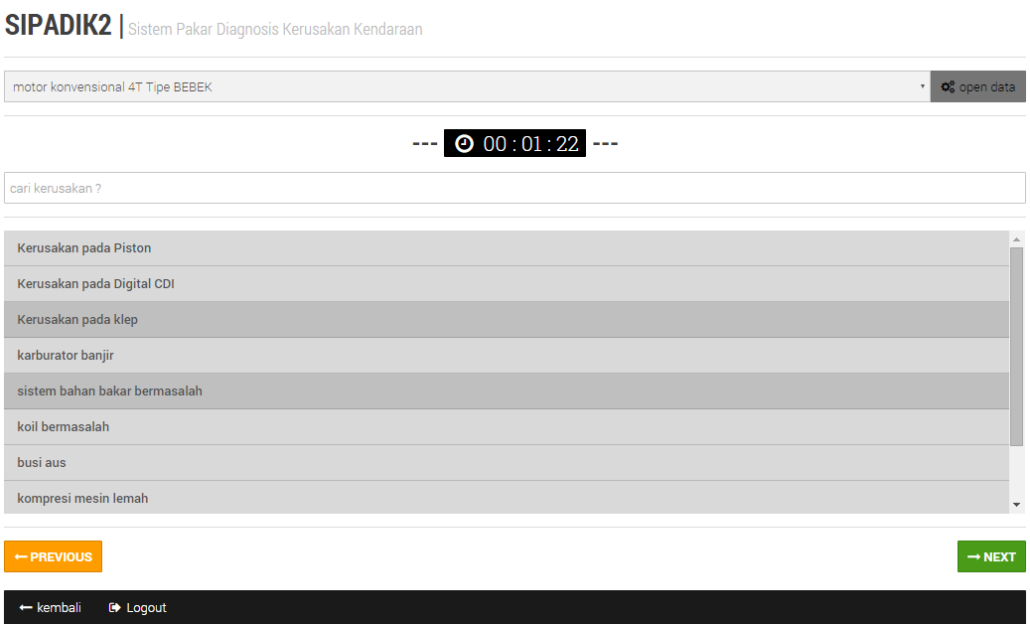
Login sebagai siswa untuk dapat masuk ke halaman latihan diagnosis kerusakan lalu pilih kendaraan. Tekan tombol mulai untuk memulai latihan lalu pilih gejala – gejala dan kerusakan – kerusakan yang ada, tekan tombol submit data untuk mendapatkan hasil diagnosis kerusakan dan benar atau tidaknya kerusakan yang dipilih.



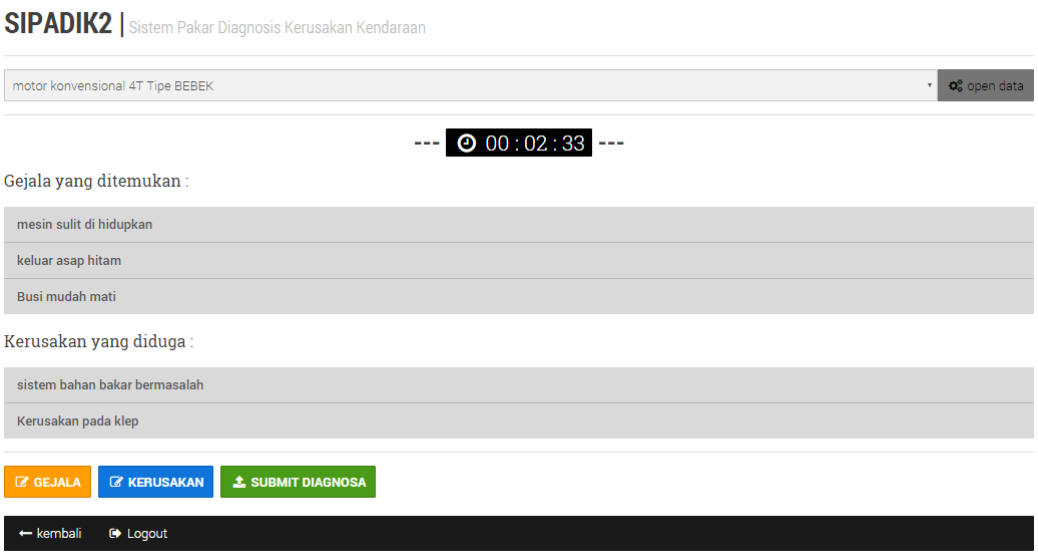
Gambar 4.47 Form mulai Latihan Diagnosis



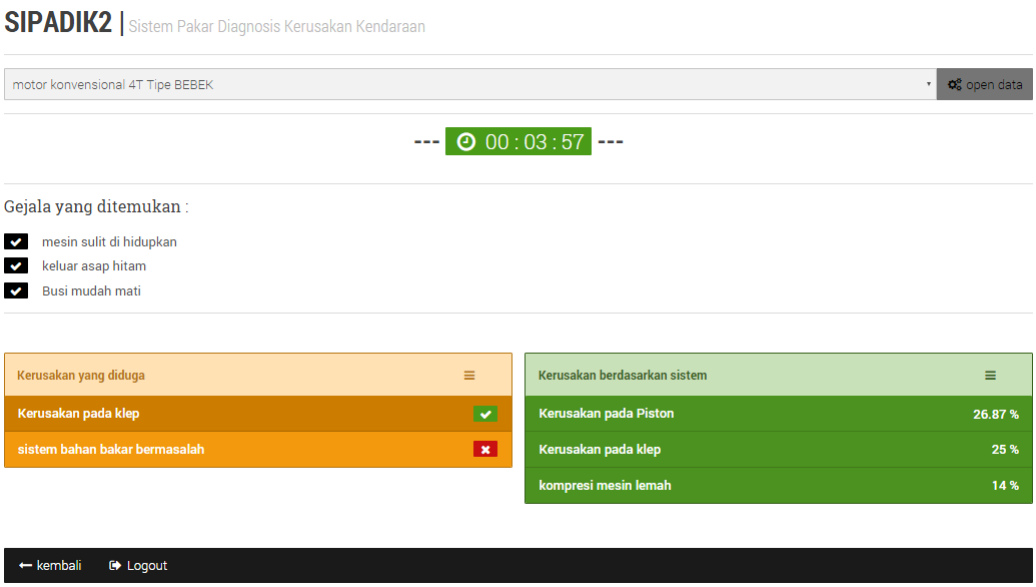
Gambar 4.48 Form pilih Gejala Latihan Diagnosis



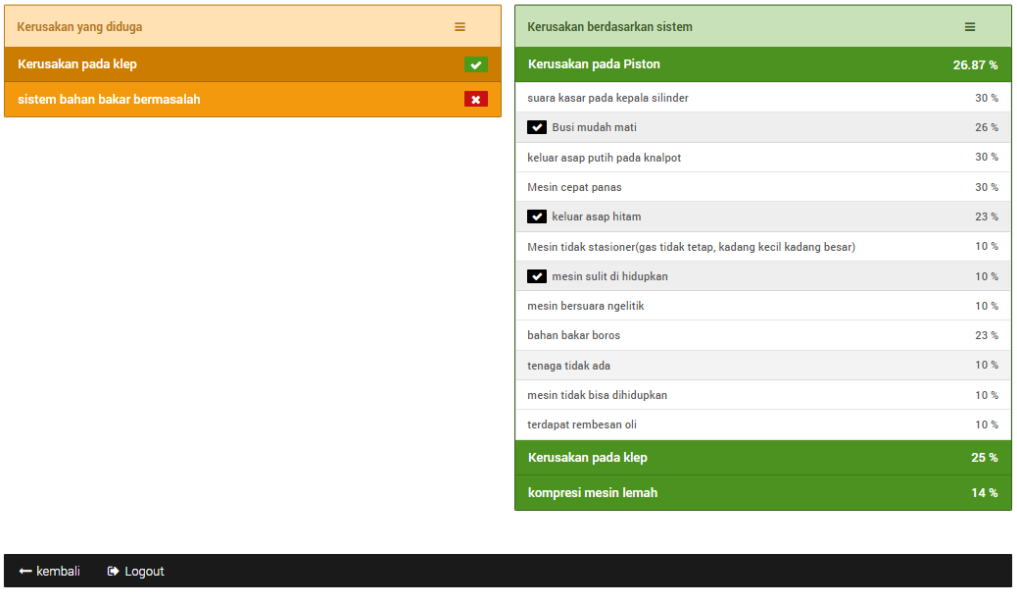
Gambar 4.49 Form Pilih Kerusakan Latihan Diagnosis



Gambar 4.50 Form Preview Latihan Diagnosis



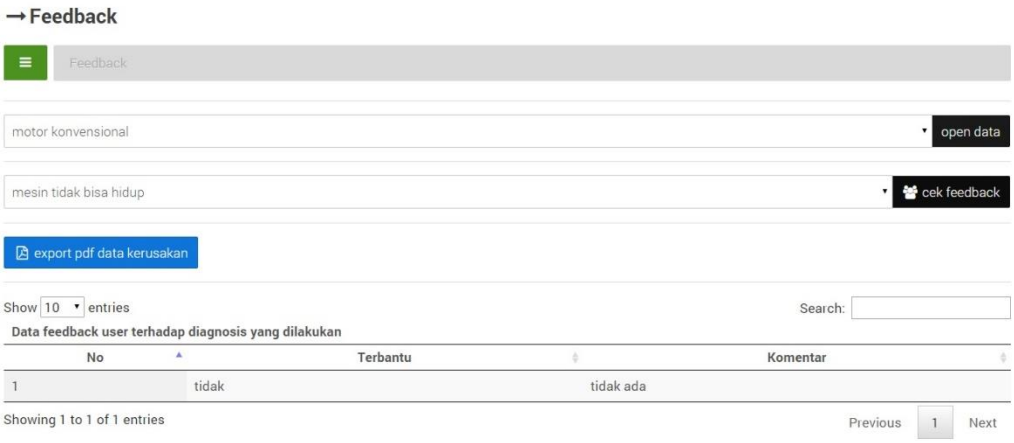
Gambar 4.51 Tampilan Hasil Latihan Diagnosis



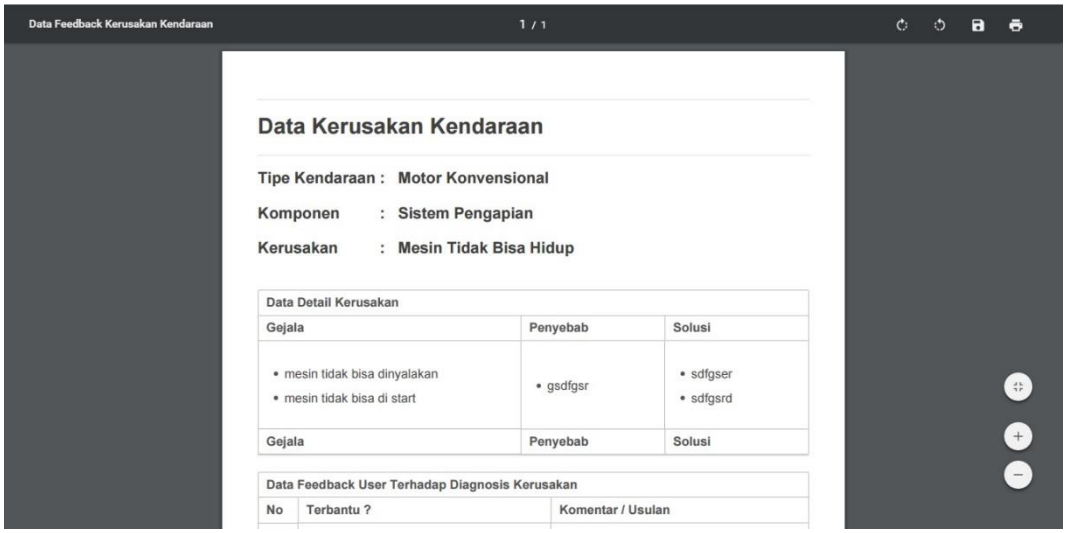
Gambar 4.52 Detail Gejala dari Kerusakan Hasil Latihan Diagnosis

8. Halaman *Feedback User*

   User pengajar / pakar dapat mendowload *feedback user* dan detail data kerusakan dengan memilih menu *feedback*. Pilih kendaraan lalu komponen dan kerusakan untuk melihat data *feedback user*. Tekan tombol *export pdf* data kerusakan untuk mengexport data kerusakan ke *file* pdf.



Gambar 4.53 Halaman Feedback User



Gambar 4.54 File Export pdf Data Kerusakan Kendaraan