

## BAB III METODE PENELITIAN

Pada bab ini, terdapat tahapan yang dilakukan peneliti sebagai alur penelitian sehingga dapat menyelesaikan penelitian yang dilakukan, tahapan pengumpulan data yang telah dikumpulkan berdasarkan teknik pengumpulan data, implementasi metode hingga pengujian sistem. Berikut ini tahapan penelitian yang dilakukan, yaitu:



**Gambar 3. 1** Diagram Alur Penelitian

### 3.1 Identifikasi Masalah

Pertama-tama mengidentifikasi masalah yang akan dipecahkan terlebih dahulu sehingga perlu dibuatkannya sistem identifikasi kerusakan

Vespa. Identifikasi masalah ini bertujuan untuk mengetahui secara detail inti permasalahan yang akan diselesaikan dan juga bagaimana langkah-langkah serta metode yang diambil untuk menyelesaikan masalah tersebut.

### **3.2 Pengumpulan Data**

Pengumpulan data merupakan tahapan awal dalam penelitian. Pada metode penelitian ini, peneliti menggunakan teknik dalam mengumpulkan data yang dibutuhkan sebagai penunjang penelitian, yaitu:

1. Observasi

Teknik pengumpulan data ini dilakukan penulis secara langsung dan tidak langsung. Penulis bertemu secara langsung dengan pakar bengkel vespa. Penulis juga melakukan observasi tidak langsung dengan mencari fakta dan berita/informasi seputar kerusakan pada vespa.

2. Wawancara

Teknik pengumpulan data dengan cara wawancara ini dilakukan untuk mendapatkan informasi yang dibutuhkan dalam penelitian bentuk tanya jawab antara peneliti dengan pakar kerusakan vespa yang bekerja di Mondo vespa

3. Studi Literatur

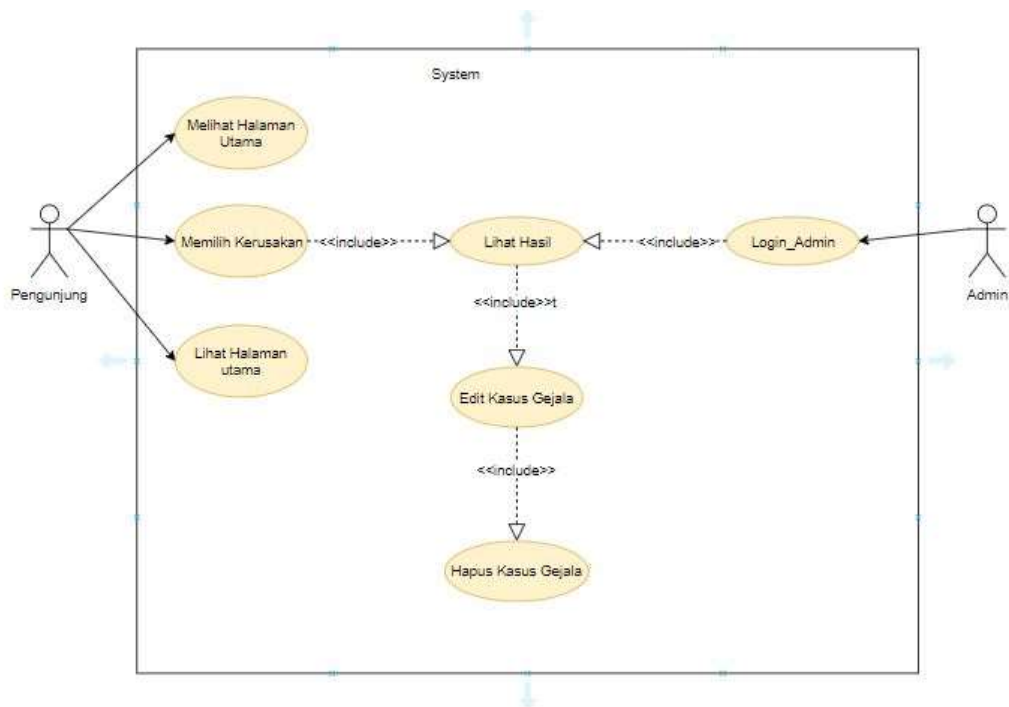
Teknik pengumpulan data secara *study literature* dilakukan dengan membaca referensi penelitian sebelumnya dan teori-teori yang menyangkut penelitian dalam bentuk berupa buku, jurnal, paper, dan sebagainya yang terkait dengan penelitian yang akan dilakukan.

### **3.3 Perancangan Sistem**

Perancangan sistem ini menjelaskan tentang alur sistem menggunakan UML diagram yang terdiri dari *Use Case Diagram*, *Activity Diagram*, dan *ER Diagram* serta merancang *interface* sistem yang digunakan untuk menggambarkan tampilan sistem yang nantinya akan digunakan

### 3.3.1.1 Use Case Diagram

Gambar 3. 2 Tampilan *use case* diagram untuk menggambarkan interaksi antar sistem, pengunjung dan admin dalam sistem pakar penulis. Pengunjung adalah *user* yang akan melakukan konsultasi kerusakan, kemudian sistem mencari data konsultasi yang diajukan pengunjung, sedangkan admin yang mengelola kasus kerusakan.



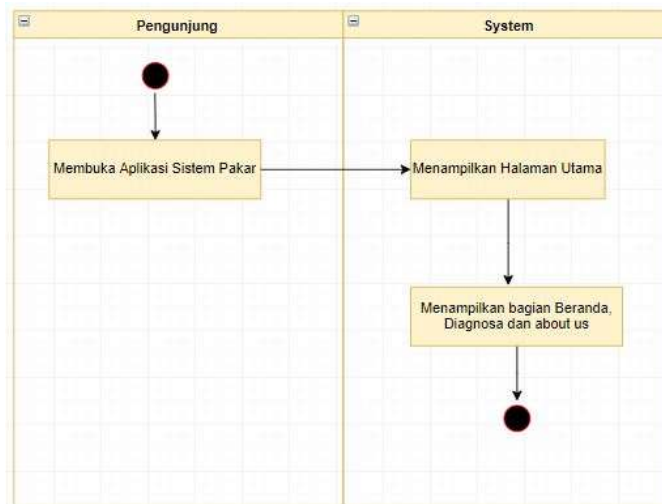
**Gambar 3. 2** Use Case Diagram Sistem Pakar

### 3.3.2 Activity Diagram

*Activity* diagram yang memperlihatkan alur setiap *menu* yang ada sistem pakar kerusakan vespa. Aktivitas yang digambarkan yaitu antara admin dan pengunjung dengan sistem.

#### 3.3.2.1 Melihat Halaman Utama

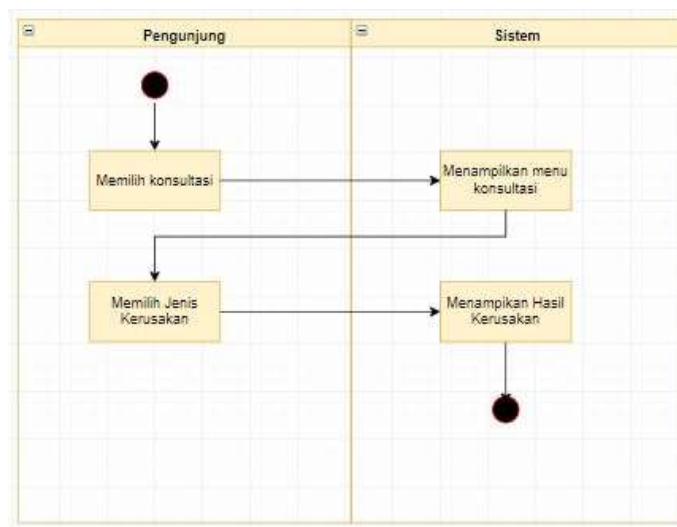
Gambar 3. 3 *activity* halaman utama pada sistem pada saat *login* sebagai user. System akan melihat halaman yang berada pada aplikasi halaman utamayang ada bagian beranda, diagnosa dan *about us*.



**Gambar 3. 3** *Activity* Halaman Utama

### 3.3.2.2 Memilih Kerusakan

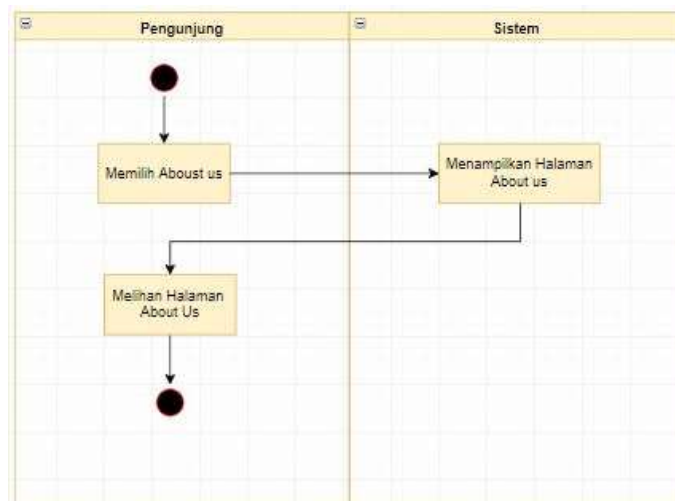
Gambar 3. 4 *activity* diagram konsultasi kerusakan, pengunjung dengan sistem pada saat melakukan memilih kerusakan motor vespa. Pengunjung memilih ciri kerusakan yang dialami motor vespa dan mendapatkan hasil kerusakan.



**Gambar 3. 4** *Activity* Diagram Konsultasi Kerusakan

### 3.3.2.3 About Us

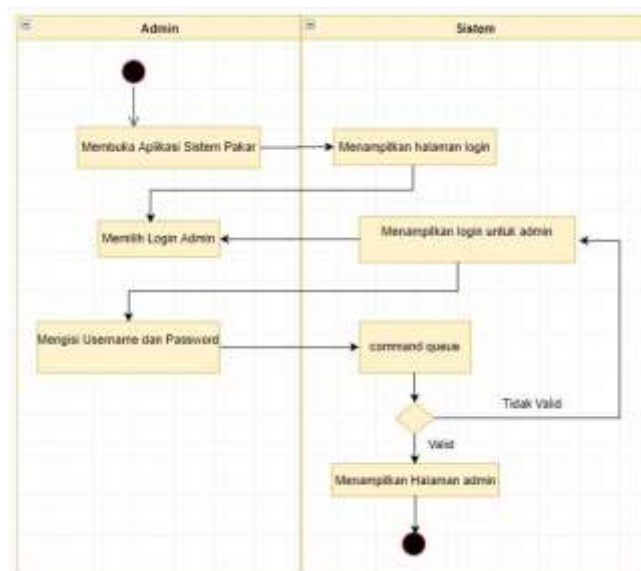
Gambar 3. 5 *activity about us* untuk mengetahui pemilik bengkel dan nama pemilik serta mengetahui mengenai bengkel di Mondo Vespa. Adapun mengetahui harga vespa yang bengkel jual.



**Gambar 3. 5** Activity About Us

### 3.3.2.4 Login admin

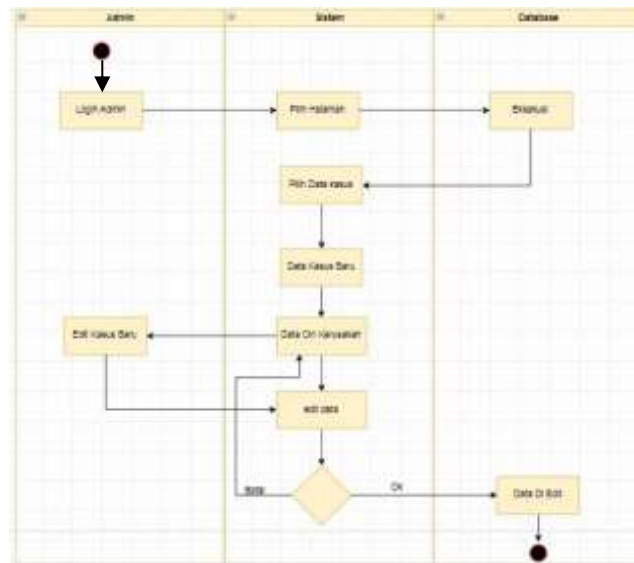
Gambar 3. 6 activity diagram login admin dengan sistem pada saat login sebagai admin. Admin memegang username dan password, kemudian proses validasi login dengan menyesuaikan database. Validasi login yang tidak sesuai, maka kembali ke halaman login admin.



**Gambar 3. 6** Activity Diagram Login Admin

### 3.3.2.5 Edit Kasus

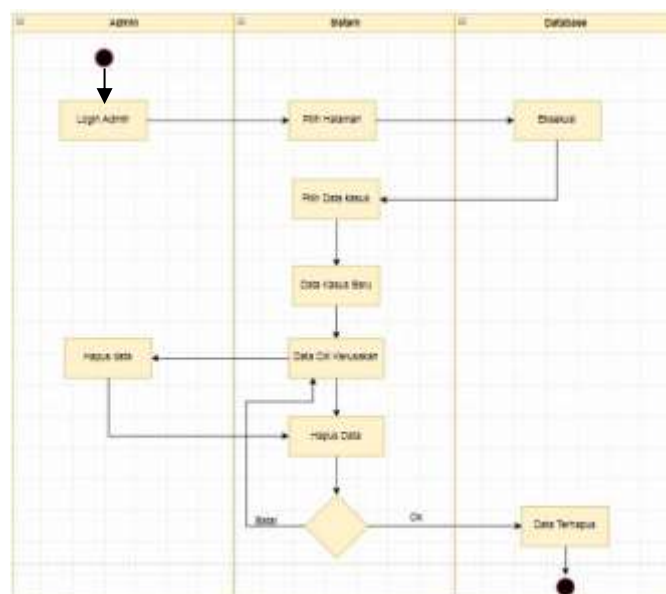
Gambar 3. 7 activity edit kasus, admin untuk edit data kasus baru pada saat ada data yang salah penulisan, untuk dimasukkan ke dalam data yang berada di halaman pengisiin user.



**Gambar 3. 7** Activity Edit Kasus

### 3.3.2.6 Hapus Kasus Kasus

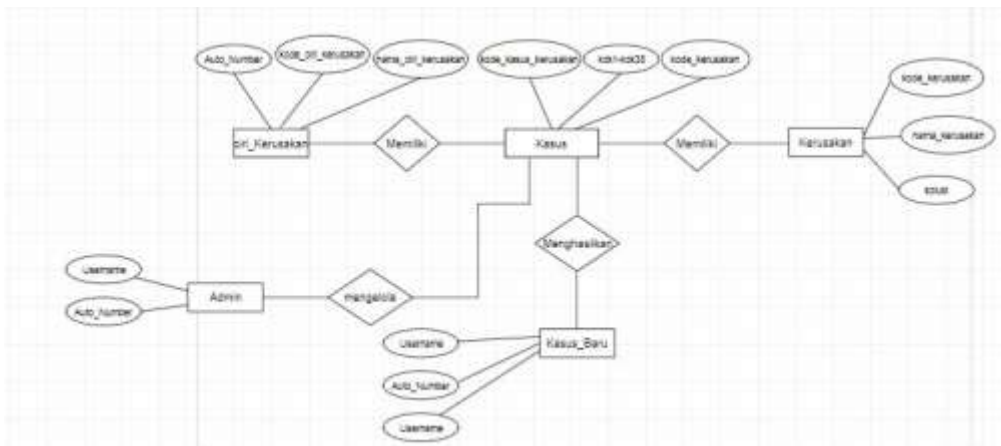
Gambar 3. 8 *activity* hapus kasus, admin untuk hapus data kasus pada saat ada data yang salah penulisan atau data yang sudah tidak dipakai.



**Gambar 3. 8** Activity Hapus Kasus

### 3.3.3 Entity Relation Diagram

Gambar 3. 9 ERD dari relasi entitas dengan *database* sistem pakar yang dirancang. Pada penelitian terdapat 6 entitas yaitu Pengunjung, Kasus, Kerusakan, Admin, dan H\_dagnosis.



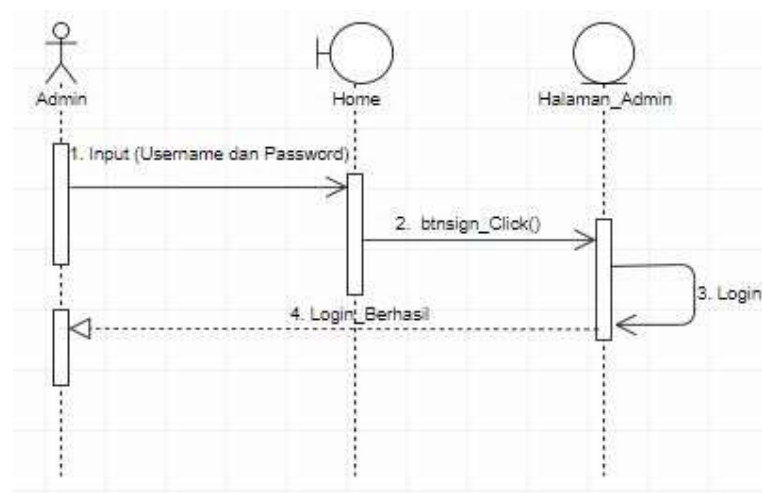
**Gambar 3. 9** ER Diagram

### 3.4 Sequence Diagram

#### 3.4.1 Sequence Diagram Login Admin

Gambar 3. 10 *Sequence Diagram* Halaman Utama Terdapat 2 kelas yang saling berintraksi, yaitu :

1. Menampilkan Admin.
2. Bagian Menu Admin.

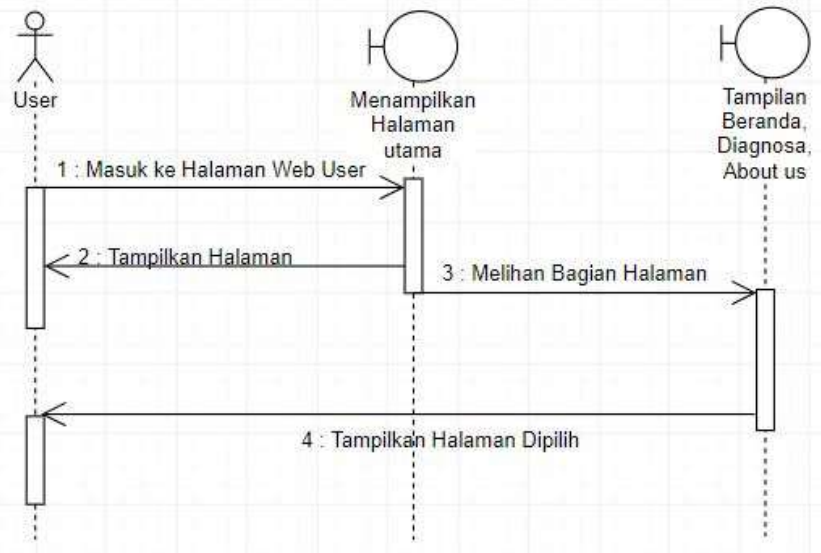


**Gambar 3. 10** Sequence Diagram Login Admin

#### 3.4.2 Sequence Diagram Melihat Halama Utama

Gambar 3. 11 *Sequence Diagram* Halaman Utama Terdapat 2 kelas yang saling berintraksi, yaitu :

1. Menampilkan Halaman Utama
2. Bagian Menu yaitu Bagian Beranda, Dignosa dan About us.

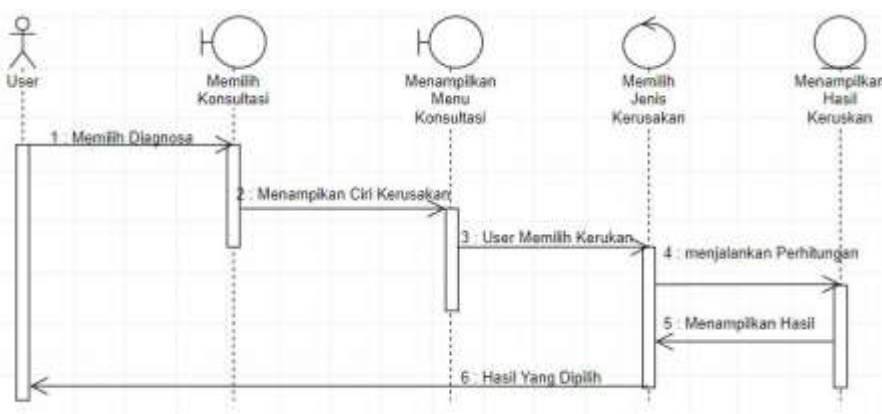


**Gambar 3. 11** *Sequence Diagram* Halaman Utama

### 3. *Sequence Diagram* Memilih Jenis Kerusakan

Gambar 3. 12 *Sequence Diagram* Jenis Kerusakan Terdapat 2 kelas yang saling berintraksi, yaitu :

1. Memilih Konsultasi
2. Menampilkan menu konsultasi
3. Memilih jenis kerusakan
4. Menampilkan Hasil Kerusakan



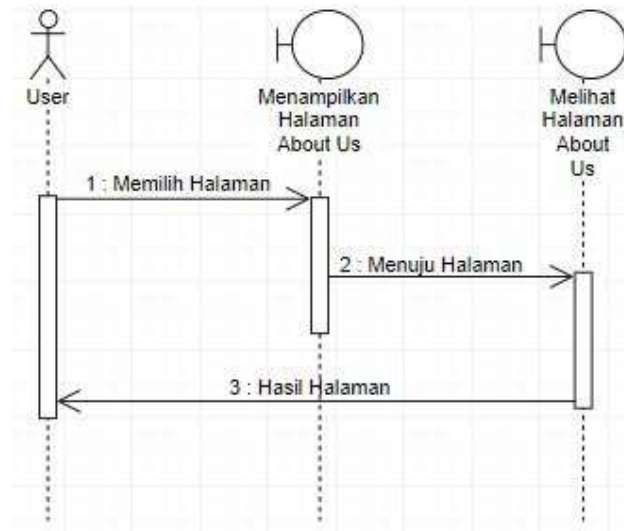
**Gambar 3. 12** *Sequence Diagram* Jenis Kerusakan

### 4. *Sequence Diagram* Halaman About Us

Gambar 3. 13 *Sequence Diagram* About us Terdapat 2 kelas yang saling berintraksi, yaitu :



1. Menampilkan Halaman About us
2. Melihat Halaman About us

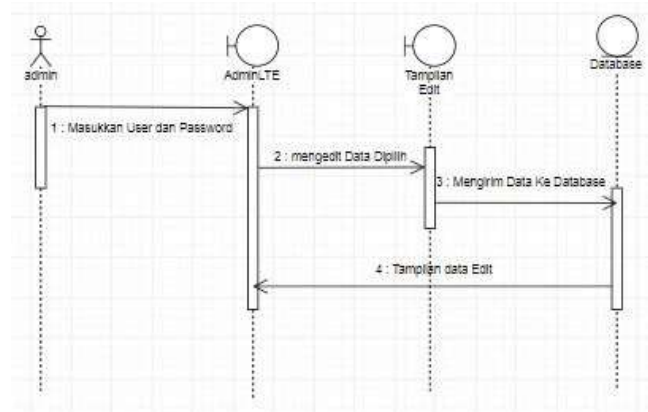


**Gambar 3. 13** *Sequence Diagram About us*

#### 5. *Sequence Diagram Edit Data*

Gambar 3. 14 *Sequence Diagram Edit Data* Terdapat 3 kelas yang saling berintraksi, yaitu :

1. Admin
2. TampilanEdit
3. Database

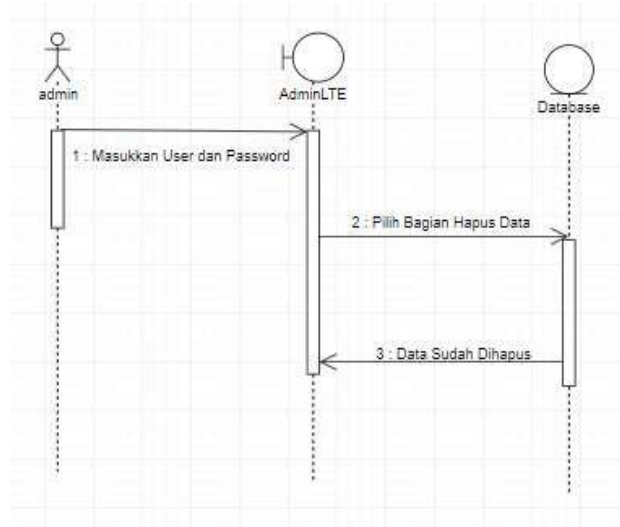


**Gambar 3. 14** *Sequence Diagram Edit Data*

#### 6. *Sequence Diagram Hapus data*

Gambar 3. 15 *Sequence Diagram Edit Data* Terdapat 3 kelas yang saling berintraksi, yaitu :

1. Admin
2. Databsae



**Gambar 3. 15** *Sequence Diagram Edit Data*

### 3.5 Implementasi Program

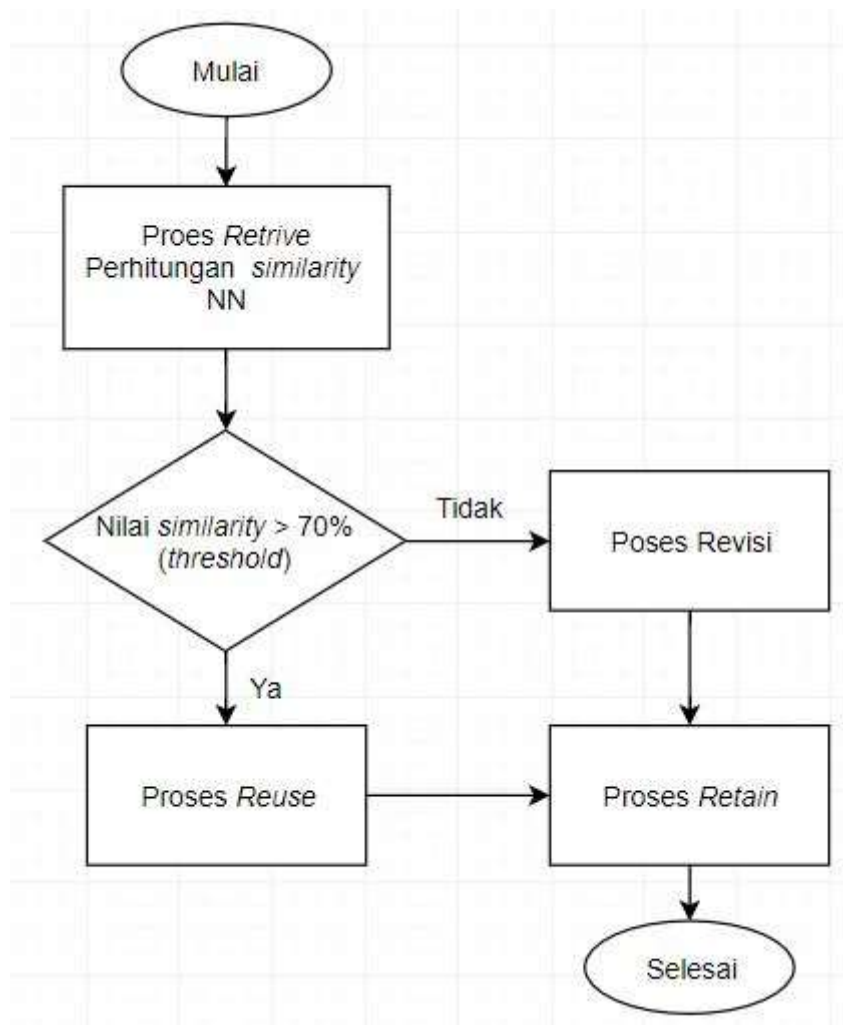
Tahap implementasi program yaitu mengimplementasikan metode kedalam sistm pakar untuk mendukung sistem yang akan dibangun. Metode CBR digunakan untuk menganalisa dan mendiagnosa kerusakan vespa. informasi atau pengetahuan tentang kerusakan vespa dikumpulkan kemudian dilakukan implementasi CBR dengan 4 tahapan yaitu *Retrieve*, *Reuse*, *Revise*, dan *Retain*[17]. Untuk mencari kedekatan dengan kasus yang lama digunakan perhitungan dalam *Nearest Neighbor*. Adapun persamaan dalam *Nearest Neighbor* adalah sebagai berikut.

$$\text{Similarity} = \frac{S_1 \times W_1 + S_2 \times W_2 + \dots + S_n \times W_n}{W_1 + W_2 + \dots + W_n} \dots\dots\dots (2)$$

Keterangan :

S = Similiraty (nilai 1 = sama dan nilai 0 = beda )

W = Bobot Kerusakan



**Gambar 3. 16** Flowchart CBR dan *similarity* NN

Perhitungan *similarity* digunakan untuk mendapatkan nilai *similarity* kesamaan kasus yang paling tinggi, apabila nilai *similarity* dengan *threshold*  $\leq 70\%$  maka akan dilakukan proses revisi. Proses revisi dilakukan untuk meninjau atau merevisi kembali usulan solusi, apabila ditemukan solusi baru.

### 3.6 Pengujian

Pengujian dilakukan dengan cara menghitung akurasi. Pengujian dilakukan untuk mengetahui apakah sistem telah layak untuk diterapkan dalam mendiagnosa kerusakan vespa. Hasil pengujian akurasi  $\geq 70\%$  (*threshold*) maka dapat disimpulkan sistem layak diterapkan dalam mendiagnosa kerusakan vespa. menghitung akurasi yaitu sebagai berikut.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots (3)$$

### **3.7 Penyusunan Laporan**

Laporan dilakukan setelah implementasi program selesai, serta dilakukan penambahan atau revisi laporan saat pengujian selesai dilakukan. Tahapan penyusunan laporan disusun berdasarkan data yang terkumpul dari penelitian yang dilakukan menggunakan metode *case Based Reasoning* dan *Nearest Neighbor* yang diimplementasikan pada sistem pakar.