

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis**

##### **3.1.1 Identifikasi Masalah**

Awal dari penelitian ini dimulai dengan mengidentifikasi masalah yang terdapat di CV Anugrah Giant Abadi. Proses pengumpulan informasi dilakukan wawancara dengan Direksi CV Anugrah Giant Abadi, setelah dilakukan wawancara terdapat beberapa kendala yang dialami yaitu:

- a. Karyawan pada CV Anugrah Giant Abadi mengalami pekerjaan yang *overload* karena waktu terbuang untuk menanggapi telepon via Whatsapp pribadi maupun telepon kantor dan email yang banyak dalam satu hari mengenai pertanyaan, keluhan dan layanan.
- b. Banyak pertanyaan yang sama saat diajukan kepada CV Anugrah Giant Abadi sehingga membuat tidak efisien dalam menjawab pertanyaan satu persatu. Dan banyak PIC Asuransi maupun IT dari setiap BPR dan KSP yang tidak sabar dengan solusi permasalahan yang diberikan, sehingga mereka mengulangi untuk menghubungi kembali ke CV Anugrah Giat Abadi.
- c. Dengan banyaknya pertanyaan yang diajukan, maka belum adanya klasifikasi sistem yang bertujuan untuk mengelola pertanyaan / permasalahan yang diajukan kepada divisi atau bagian yang dapat menangani *ticket* tersebut.
- d. Sehubungan dengan banyaknya pertanyaan atau keluhan, PIC CV Anugrah Giant Abadi maupun Direksi dari CV Anugrah Giant Abadi tidak dapat mengetahui dalam setiap bulan pertanyaan atau keluhan apa yang sering muncul.

### 3.1.2 Pemecahan Masalah

Berdasarkan uraian identifikasi masalah di atas, dapat dilakukan proses pemecahan masalah dengan membuat perancangan sistem helpdesk berbasis website pada CV Anugrah Giant Abadi, yaitu:

- a. Pertanyaan umum, keluhan, layanan dapat terkontrol dan terstruktur dengan baik.
- b. Pada saat membuat ticket helpdesk setelah mengisi biodata dan mengisi diagnosa permasalahan, dan ticket tersebut akan di setujui oleh admin. Maka dengan klasifikasi Naïve Bayes yang akan menentukan teknisi yang di dapat ditugaskan untuk menangani ticket tersebut.
- c. Pertanyaan yang umum ditanyakan, akan ditampilkan pada halaman “Pertanyaan Umum”. Pada halaman tersebut pertanyaan yang banyak dan sering ditanyakan, akan ditambahkan pada halaman tersebut secara manual oleh staff helpdesk CV Anugrah Giant Abadi.
- d. Dengan banyaknya pertanyaan atau keluhan dari user, CV Anugrah Giant Abadi dapat mengetahui pertanyaan atau keluhan apa yang sering muncul sehingga dapat ditanggulangi setiap bulan.

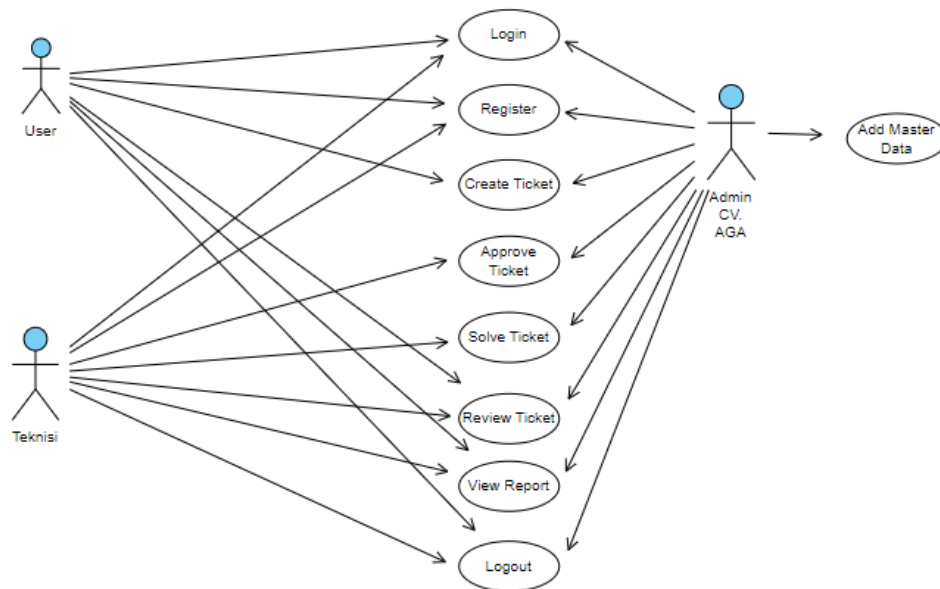
## 3.2 Perancangan

### 3.2.1 Perancangan Sistem

Pada perancangan system penelitian ini, yang berjalan saat ini di CV Anugrah Giant Abadi yaitu PIC IT dari setiap BPR / KSP melaporkan keluhan yang dialami melalui WA pribadi, email, telepon kantor. Oleh karena itu, penulis menggunakan metode *Naive Bayes* dan metode pengembangan *Scrum* untuk dapat

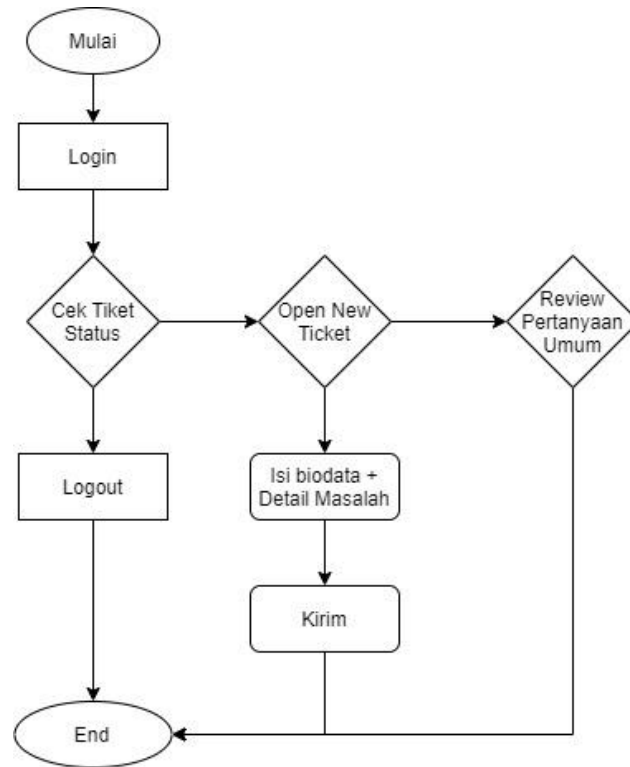
mencapai tujuan dan manfaat pada penelitian ini.. Pada metode tersebut terdapat beberapa tahap yang dilakukan untuk melakukan klasifikasi pada Sistem Informasi Helpdesk CV Anugrah Giant Abadi. Selanjutnya penulis sudah melakukan wawancara dengan teknisi dan staff pada CV Anugrah Giant Abadi mengenai permasalahan hardware dan software yang sering ditangani dan diselesaikan.

Berikut *Use Case Diagram* pada Pengembangan Sistem Informasi Helpdesk pada CV Anugrah Giant Abadi



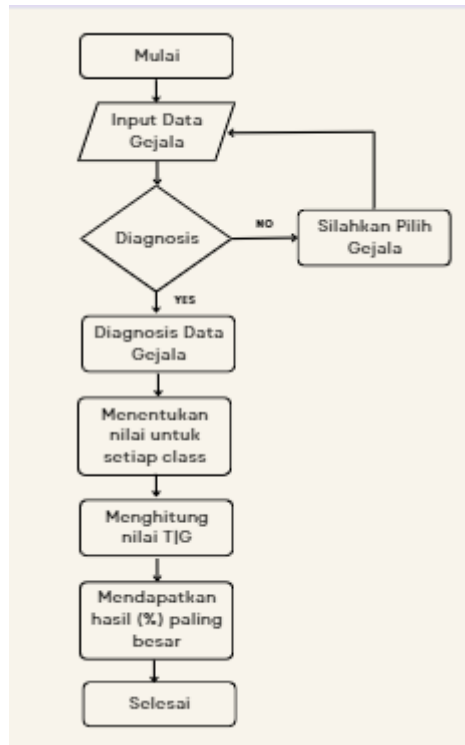
**Gambar 3. 1** Use Case Diagram

Berdasarkan gambar di atas dapat dijelaskan bahwa dalam Helpdesk Sistem penulis akan melibatkan 3 Role Akses yaitu Helpdesk (Admin) , Teknisi, PIC Asuransi & IT KSP & BPR (User). Admin dapat mengakses semua fitur, User dapat mengakses login, register, membuat tiket, melihat tiket, melihat hasil dan logout. Sedangkan Teknisi hanya bisa mengakses login, register, menyetujui tiket, menyelesaikan tiket, melihat tiket, meliha hasil dan logout.



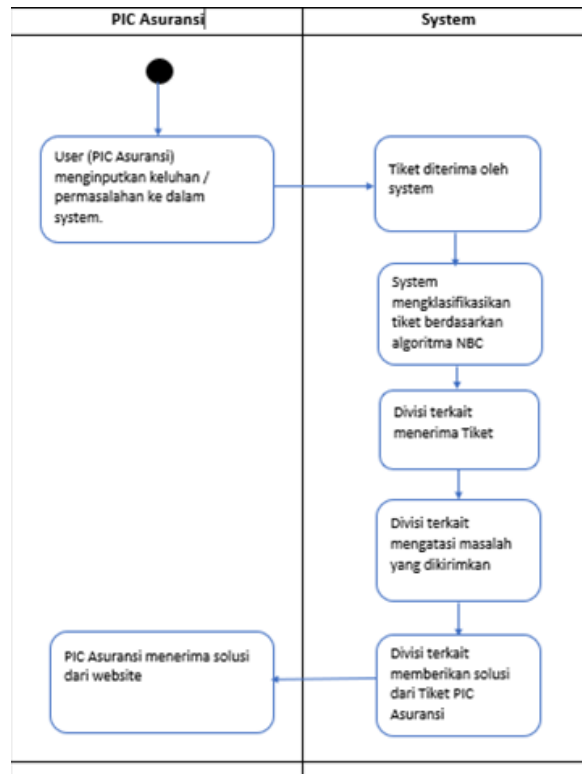
**Gambar 3. 2** *Flowchart* Sistem Helpdesk

Pada *Flowchart* Sistem Helpdesk, dijelaskan bahwa ada 3 proses. Yang pertama proses Cek Tiket Status, lalu proses selesai. Yang kedua proses Open New Ticket yang dimaksudkan bahwa untuk membuat tiket baru pada saat mengajukan pertanyaan dan keluhan lalu menuju proses kirim tiket dan proses selesai. Dan yang ketiga yaitu proses Review Pertanyaan Umum. Proses yang terakhir dibuat untuk melihat pertanyaan yang sering muncul dan dapat di *review* kembali oleh *user*.



**Gambar 3. 3** Flowchart Diagnosis

Pada *flowchart* diagnosis, dijelaskan bahwa saat proses dimulai oleh *user* maka akan masuk ke menu input data gejala. Setelah memasukan data gejala yang dialami pengguna mengenai permasalahan hardware atau software, sistem akan melakukan proses diagnosa dan melakukan klasifikasi *Naive Bayes* dengan menentukan nilai untuk setiap *class*, menghitung nilai Teknisi terhadap setiap Gejala yang dikeluhkan, dan proses terakhir perhitungan yaitu menentukan prosentase yang terbesar. Yang merupakan prosentase terbesar, yang menjadi hasil klasifikasi untuk penentuan teknisi yang ditugaskan.

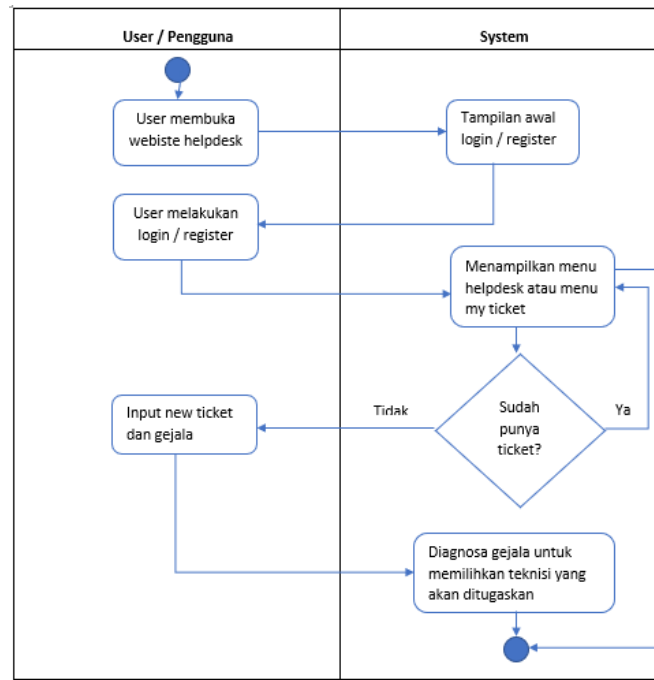


**Gambar 3. 4** Workflow

Pada Gambar 3.6 *Workflow*, terdapat 2 bagan yaitu PIC IT / Asuransi dan System. Paada penelitian ini, PIC Asuransi menginputkan keluhan / permasalahan ke dalam system, lalu tiket diterima oleh system, system mengklasifikasikan tiket berdasarkan algoritma NBC, Divisi terkait menerima tiket, Divisi terkait mengatasi masalah, Divisi terkait memberikan solusi lalu proses berpindah ke bagan PIC IT / Asuransi dan sudah menerima solusi.

*Activity* diagram yaitu menggambarkan aktivitas dalam perancangan yang dibangun dari awal, keputusan yang mungkin terjadi, dan bagaimana akhirnya. *Activity* diagram hanya memberikan gambaran proses global sesuai urutan. Pada *website helpdesk* terdapat tiga *activity* diagram, yaitu:

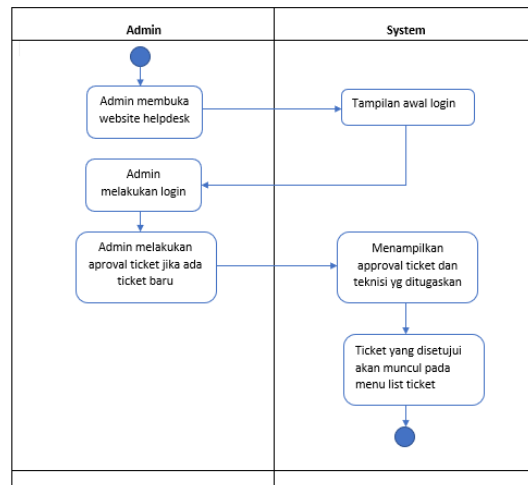
a. *User / Pengguna*



**Gambar 3. 5** Activity Diagram User

Gambar 3.17 menunjukkan aliran aktivitas yang dapat dilakukan oleh user (pengguna) adalah melakukan login dan register, membuat ticket helpdesk dengan menambahkan gejala yang dialami dan melihat progress pengerjaan ticket yang dikerjakan oleh teknisi.

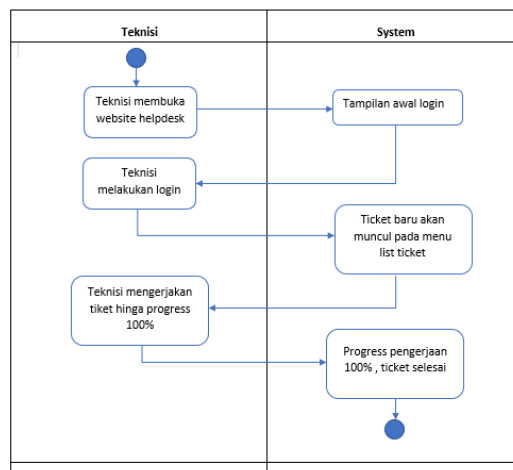
## b. Admin



**Gambar 3. 6** Activity Diagram Admin

Gambar 3.6 menunjukkan aliran aktivitas yang dapat dilakukan admin adalah melakukan login menyetujui ticket yang telah dibuat user, menyimpan data aproval dengan teknisi yang sudah diklasifikasikan Naive Bayes, melihat status ticket terselesaikan atau belum.

## c. Teknisi



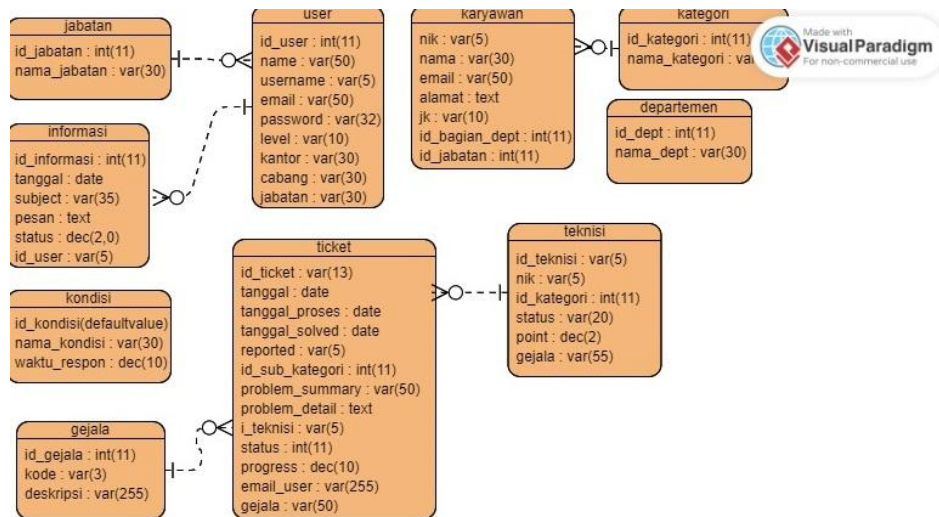
**Gambar 3. 7** Activity Diagram Teknisi



Gambar 3.7 menunjukkan aliran aktivitas yang dapat dilakukan oleh teknisi adalah melakukan login, mengerjakan ticket yang diterima hingga progress 100%.

### 3.2.2 Perancangan Data

Pada bab ini dilakukan perancangan data menggunakan conceptual data model untuk menggambarkan keseluruhan struktur logika dari basis data dan physical data model sebagai implementasi dari conceptual data model. Berikut ini adalah gambar conceptual data model dan physical data model yang digunakan.



**Gambar 3. 8** Conceptual Data Model

Pada gambar 3.8, terdapat beberapa entitas dan relasinya dengan entitas lain. Berikut adalah penjelasan setiap entitas dan relasinya.

#### 1. Jabatan.

Entitas jabatan memiliki atribut id\_jabatan dan nama\_jabatan. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas user.

## 2. Informasi

Entitas jabatan memiliki id\_informasi, tanggal, subject, pesan, status, id\_user. Primary identifier pada entitas ini adalah id.

## 3. Kondisi

Entitas jabatan memiliki id\_kondisi, nama\_kondisi, waktu\_respon. Primary identifier pada entitas ini adalah id.

## 4. Gejala

Entitas jabatan memiliki id\_gejala, kode, deskripsi. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas ticket.

## 5. User

Entitas jabatan memiliki id\_user, name, username, email, password, level, kantor, cabang, jabatan. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas informasi.

## 6. Ticket

Entitas jabatan memiliki id\_ticket, tanggal, tanggal\_proses, tanggal\_solved, reported, id\_sub\_kategori, problem\_summary, problem\_detail, id\_teknisi, status, progress, email\_user, gejala. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas informasi.

#### 7. Karyawan

Entitas jabatan memiliki nik, nama, email, alamat, jk, id\_bagian\_dept, id\_jabatan. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas informasi.

#### 8. Kategori

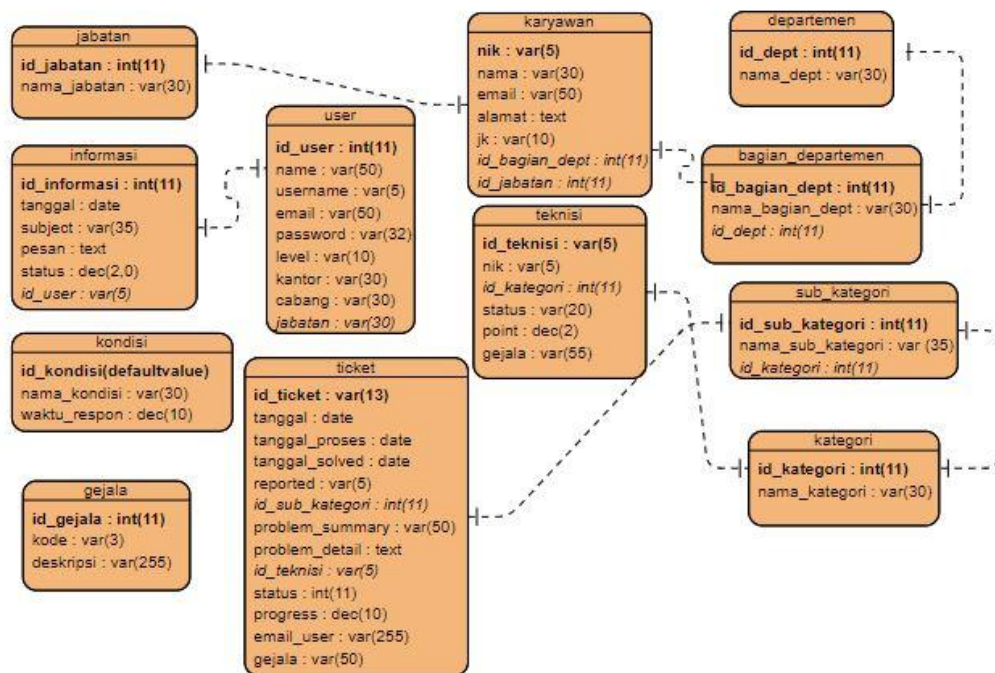
Entitas jabatan memiliki atribut id\_kategori dan nama\_kategori. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas karyawan.

#### 9. Departemen

Entitas jabatan memiliki atribut id\_dept dan nama\_dept. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas karyawan.

#### 10. Teknisi

Entitas jabatan memiliki atribut id\_teknisi, nik, id\_kategori, status, point, gejala. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas ticket.



**Gambar 3. 9** Physical Data Model

Pada gambar 3.9, terdapat beberapa table dan relasinya dengan table lain.

Berikut adalah penjelasan setiap table dan relasinya.

### 1. Jabatan.

Entitas jabatan memiliki atribut `id_jabatan` dan `nama_jabatan`. Primary identifier pada entitas ini adalah `id`. Relasi pada entitas ini adalah one to many dengan entitas `user` dan `karyawan`.

### 2. Informasi

Entitas jabatan memiliki `id_informasi`, `tanggal`, `subject`, `pesan`, `status`, `id_user`. Primary identifier pada entitas ini adalah `id`. Relasi pada entitas ini adalah one to many dengan entitas `user`.

### 3. Kondisi

Entitas jabatan memiliki id\_kondisi, nama\_kondisi, waktu\_respon. Primary identifier pada entitas ini adalah id.

#### 4. Gejala

Entitas jabatan memiliki id\_gejala, kode, deskripsi. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas ticket.

#### 5. User

Entitas jabatan memiliki id\_user, name, username, email, password, level, kantor, cabang, jabatan. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas informasi.

#### 6. Ticket

Entitas jabatan memiliki id\_ticket, tanggal, tanggal\_proses, tanggal\_solved, reported, id\_sub\_kategori, problem\_summary, problem\_detail, id\_teknisi, status, progress, email\_user, gejala. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas sub\_kategori dan teknisi.

#### 7. Karyawan

Entitas jabatan memiliki nik, nama, email, alamat, jk, id\_bagian\_dept, id\_jabatan. Primary identifier pada entitas ini adalah id. Relasi pada entitas ini adalah one to many dengan entitas bagian\_dept dan jabatan.

#### 8. Kategori

Entitas jabatan memiliki atribut id\_kategori dan nama\_kategori. Primary identifier pada entitas ini adalah id.

#### 9. Departemen

Entitas jabatan memiliki atribut `id_dept` dan `nama_dept`. Primary identifier pada entitas ini adalah `id`.

#### 10. Teknisi

Entitas jabatan memiliki atribut `id_teknisi`, `nik`, `id_kategori`, `status`, `point`, `gejala`. Primary identifier pada entitas ini adalah `id`. Relasi pada entitas ini adalah one to many dengan entitas kategori.

#### 11. Sub Kategori

Entitas jabatan memiliki atribut `id_sub_kategori`, `nama_sub_kategori`, `id_kategori`. Primary identifier pada entitas ini adalah `id`. Relasi pada entitas ini adalah one to many dengan entitas kategori.

#### 12. Bagian Departemen

Entitas jabatan memiliki atribut `id_bagian_dept`, `nama_bagian_dept`, `id_dept`. Primary identifier pada entitas ini adalah `id`. Relasi pada entitas ini adalah one to many dengan entitas departemen.

Untuk metode *Naïve Bayes* yang dilakukan penulis pada penelitian ini, memiliki 4 data teknisi (T) dan 13 gejala (G). Gejala yang menjadi dasar penyebab penyakit disusun menggunakan basis aturan. Tahapan – tahapan tersebut meliputi:

Data hasil yang merupakan data yang menjadi jawaban setelah menghitung probabilitas setiap *class*.

**Tabel 3. 1** Data Hasil

Kode	Teknisi
T1	Teknisi 1
T2	Teknisi 2
T3	Teknisi 3
T4	Teknisi 4
T5	Teknisi 5

Data Nilai atribut merupakan pilihan nilai yang mungkin untuk masing masing data hasil berdasarkan basis aturan, meliputi:

**Tabel 3. 2** Data Gejala

Kode	Gejala
G1	Sebagian tombol keyboard tidak berfungsi
G2	Keyboard mengetik sendiri
G3	PC menyala
G4	Komponen hardware lain berjalan normal
G5	Gambar pada monitor tidak muncul
G6	Sinyal gambar tidak muncul
G7	Komputer restart terus menerus
G8	Banyak file yang corrupt
G9	Kinerja PC melambat
G10	PC tidak dapat connect jaringan
G11	LED Router menyala
G12	Telah melakukan restart PC
G13	Aplikasi PGBS tidak dapat muncul
G14	CPU menyala, proses post muncul "Operating System not found"
G15	PC tidak masuk ke bios
G16	CD tidak terbaca pada windows
G17	CD hanya dapat dibaca tidak dapat diedit
G18	Tidak muncul lampu indikator yang menyala
G19	Kipas power supply tidak berputar
G20	Sudah reinstall aplikasi PGBS

Basis aturan merupakan aturan untuk setiap diinputnya data nilai agar mendapatkan data hasil.

**Tabel 3. 3** Basis Aturan

Kode Teknisi	Gejala Permasalahan
<b>T1</b>	(G1) (G2) (G3) (G4) (G5) (G6)
<b>T2</b>	(G7) (G8) (G9) (G10) (G11) (G12)
<b>T3</b>	(G3) (G4) (G12) (G13) (G20)
<b>T4</b>	(G14) (G15) (G16) (G17) (G18) (G19) (G20)
<b>T5</b>	Keluhan yang diinputkan pada textbox

Data Training merupakan data aktual sebagai acuan untuk melakukan perhitungan Naive Bayes. Berikut adalah data training yang diketahui:

**Tabel 3. 4** Data Training  
**Tabel Keputusan antara Kerusakan & Gejala**

	T1	T2	T3	T4	T5
G1	1	0	0	0	0
G2	1	0	0	0	0
G3	1	0	1	0	0
G4	1	0	1	0	0
G5	1	0	0	0	0
G6	1	0	0	0	0
G7	0	1	0	0	0
G8	0	1	0	0	0
G9	0	1	0	0	0
G10	0	1	0	0	0
G11	0	1	0	0	0
G12	0	1	1	0	0
G13	0	0	1	0	0
G14	0	0	0	1	1
G15	0	0	0	1	1
G16	0	1	0	1	1
G17	0	1	0	1	1
G18	0	0	0	1	1



G19	0	0	0	1	1
G20	0	0	1	1	1

**a. Pemetaan data uji:**

Sebelum melakukan proses algoritma naive bayes, tahap awal yang dikerjakan yaitu menentukan nilai probabilitas teknisi yang dapat menangani gejala kerusakan berdasarkan basis aturan. Nilai probabilitas berdasarkan dari keluhan pengguna. Kemudian system akan melakukan perhitungan. Berikut adalah data gejala keluhan yang dipilih pengguna:

**Tabel 3. 5** Data Uji

Kode Gejala	Gejala
G5	Gambar pada monitor tidak muncul
G6	Sinyal gambar tidak muncul
G3	PC menyala
G4	Komponen hardware lain berjalan normal
G1	Sebagian tombol keyboard tidak berfungsi
G2	Keyboard mengetik sendiri

Dari table di atas, gejala yang muncul terdapat pada T1 , T2 , T3 (perhatikan lagi table basis aturan). Kemudian perhitungan probabilitas dilakukan pada setiap Teknisi yang merujuk persamaan (1).

**b. Mencari Probabilitas**

Langkah pertama yaitu mencari probabilitas gejala yang muncul pada jumlah semua gejala. Berikut probabilitas kemunculan nilai atribut label:

$$T1 = 1/6 = 0,17$$

Prediksi minimal dinyatakan dengan angka 1 untuk teknisi dan 6 dari jumlah gejala. Lalu perhitungan dilakukan pada semua gejala (G) yang ada pada T1 berdasarkan ketentuan berikut:

$$G_5 = \frac{\text{Jumlah gejala } G_5 \text{ yang muncul}}{\text{Jumlah penyakit yang muncul akibat gejala}}$$

$$= 1/3 = 0,33$$

1 adalah gejala yang muncul pada T1. Menghitung gejala yang lain menggunakan cara yang sama, seperti berikut:

$$G_6 = 1/3 = 0,33$$

$$G_3 = 1/3 = 0,33$$

$$G_4 = 1/3 = 0,33$$

$$G_1 = 1/3 = 0,33$$

$$G_2 = 1/3 = 0,33$$

### **Probabilitas T2**

$$T_2 = 1/6$$

$$= 0,17$$

Probabilitas G terhadap T2

$$G_5 = 1/3 = 0,33$$

$$G_6 = 1/3 = 0,33$$

$$G_3 = 1/3 = 0,33$$

$$G_4 = 1/3 = 0,33$$

$$G_1 = 1/3 = 0,0$$

$$G_2 = 1/3 = 0,0$$

### **Probabilitas T3**

$$T_3 = 1/6$$

$$= 0,17$$

Probabilitas G terhadap T3

$$G_5 = 1/3 = 0,0$$

$$G_6 = 1/3 = 0,0$$

$$G_3 = 1/3 = 0,33$$

$$G_4 = 1/3 = 0,33$$

$$G_1 = 1/3 = 0,33$$

$$G_2 = 1/3 = 0,33$$

Untuk T1:

$$\begin{aligned} T(T1|G5) &= [T(T1|G5) \times T(T1)] / [T(T1|G5) \times T(T1) + T(T2|G5) \times T(T2) + \\ &T(T3|G5) \times T(T3)] \\ &= (0,33 \times 0,17) / [(0,33 \times 0,17) + (0,33 \times 0,17) + (0,00 \times 0,17)] \\ &= 0,0561 / 0,1122 \\ &= 0,5 \end{aligned}$$

Menghitung nilai setiap T dengan cara yang sama

$$T(T1|G6) = 0,5$$

$$T(T1|G3) = 0,3$$

$$T(T1|G4) = 0,3$$

$$T(T1|G1) = 0,5$$

$$T(T1|G2) = 0,5$$

Maka:

$$\begin{aligned} \text{Total T1} &= 0,5 + 0,5 + 0,3 + 0,3 + 0,5 + 0,5 \\ &= 2,6 \end{aligned}$$

Perhitungan serupa juga dapat digunakan untuk menghitung T2 dan T3 dengan hasil masing masing 1,6 dan 1,6.

$$\begin{aligned} \text{Seluruh total dijumlahkan, maka Total Bayes T} &= T1 + T2 + T3 = 2,6 + 1,6 + 1,6 \\ &= 5,8 \end{aligned}$$

Berikutnya menghitung persentase dari nilai prediksi setiap T, yakni:

$$T1 = (2,6 / 5,8) \times 100 = 44,8\%$$

$$T2 = (1,6 / 5,8) \times 100 = 27,6\%$$

$$T3 = (1,6 / 5,8) \times 100 = 27,6\%$$

Hasil dari perhitungan persentase nilai tertinggi adalah T1 sebesar 44,8%. Dari hasil ini dapat disimpulkan gejala kerusakan yang dialami pengguna (G5, G6, G3, G4, G1, G2) di prediksi yang mengerjakan ticket helpdesk tersebut adalah Teknisi 1.

Pada **Gambar 3**. Merupakan metodologi Pengembangan Sistem menurut metode *Scrum* pada Pengembangan Sistem Informasi Helpdesk pada CV Anugrah Giant Abadi



**Gambar 3. 10** Metodologi Pengembangan Sistem

### c. Scrum Agile Development

Pada metode *Scrum Agile Development* terdapat beberapa tahap yang dilakukan untuk pengembangan Sistem Informasi Helpdesk CV Anugrah Giant Abadi, tahapan – tahapan tersebut meliputi:

#### 1. User Stories

*User Stories* digunakan untuk membuat *product backlog*. Dalam *user stories* berisi fitur atau permintaan dari *User* yang akan menggunakan sistem atau aplikasi yang akan dibuat tujuan dari fitur – fitur tersebut.

- a) Sebagai staff CV Anugrah Giant Abadi Divisi IT, penulis ingin kinerja saya lebih cepat dan efektif. Ketika ada pertanyaan atau keluhan yang diajukan dari PIC setiap BPR / KSP, hanya membuka satu website saja.
- b) Sebagai staff CV Anugrah Giant Abadi Divisi IT, penulis ingin melihat progress Teknisi maupun Admin dalam membantu menyelesaikan permasalahan setiap BPR / KSP.
- c) Sebagai staff CV Anugrah Giant Abadi Divisi IT, penulis ingin semua pertanyaan ataupun keluhan yang sudah diajukan dapat diatasi semua. Karena data yang sudah diajukan akan ter-*record* pada *database*.
- d) Sebagai staff CV Anugrah Giant Abadi Divisi IT, penulis ingin menambahkan pertanyaan dan keluhan yang sering diajukan pada menu khusus yang diisi oleh admin agar pengguna dapat melihat menu tersebut tanpa harus mengajukan ticket baru.

- e) Direksi CV Anugrah Giant Abadi hanya dapat memantau *dashboard* dan tidak bisa mengedit.
- f) Sebagai staff CV Anugrah Giant Abadi, dapat melihat daftar tiket, tiket yang disetujui dan daftar tiket user.
- g) Sebagai staff CV Anugrah Giant Abadi, penulis ingin dapat menambahkan, mengedit dan menghapus data pada master User.
- h) Sebagai staff CV Anugrah Giant Abadi, penulis ingin dapat menambahkan, mengedit dan menghapus data pada master Departemen.
- i) Sebagai staff CV Anugrah Giant Abadi, penulis ingin dapat menambahkan, mengedit dan menghapus data pada master Kategori.
- j) Sebagai staff CV Anugrah Giant Abadi, penulis ingin dapat menambahkan, mengedit dan menghapus data pada master User.
- k) Sebagai staff CV Anugrah Giant Abadi, a ingin teknisi yang ditugaskan berdasarkan klasifikasi *Naive Bayes*.

## 2. Menentukan Product Backlog

Pada tahap ini, penulis akan membuat daftar dari setiap hal yang dibutuhkan di dalam aplikasi yang akan dikembangkan. *Product Backlog* dibuat berdasarkan dari analisis pada tahap sebelumnya yaitu *User Stories*. Setelah *Product Backlog* dibuat, langkah selanjutnya yaitu menentukan prioritas dan estimasi waktu dari setiap daftar *Product Backlog*.

**Tabel 3. 6** Menentukan *Product Backlog*

ID	Fitur
PB1	Pengembangan fungsi Login dan Register
PB2	Pengembangan fungsi Pembuatan Tiket ( <i>New Ticket</i> )
PB3	Pengembangan fungsi Daftar Tiket ( <i>List Ticket</i> )
PB4	Pengembangan fungsi Tiket yang telah disetujui ( <i>Approval Ticket</i> )
PB5	Pengembangan fungsi Tiket setiap <i>User</i> ( <i>My Ticket</i> )
PB6	Pengembangan fungsi tambah, ubah dan hapus pada master <i>User</i>
PB7	Pengembangan fungsi tambah, ubah dan hapus pada master Departemen
PB8	Pengembangan fungsi tambah, ubah dan hapus pada master Kategori
PB9	Pengembangan fungsi tambah, ubah dan hapus pada master Kondisi
PB10	Pengembangan fungsi tambah, ubah dan hapus pada master Informasi (Hanya dapat ditambahkan oleh admin)
PB11	Pengembangan fungsi menampilkan daftar Informasi
PB12	Pengembangan fungsi metode Naive Bayes pada klasifikasi teknisi
PB13	Pengembangan fungsi aksi pada menu <i>Approval Ticket</i>
PB14	Pengembangan fungsi validasi feedback
PB15	Pengembangan fungsi progress setiap pengerjaan teknisi
PB16	Pengembangan fungsi <i>Dashboard</i>

### 3. Menentukan Sprint Planning

Pada tahap ini, penulis akan menentukan estimasi waktu pengerjaan dari *Product Backlog* yang sudah dibuat pada tahap sebelumnya. Pada langkah ini, penulis juga menentukan batas waktu dalam satu *Sprint* yaitu satu bulan atau empat minggu (sabtu dan minggu libur).

**Tabel 3. 7** Menentukan *Sprint Planning*

<b>ID</b>	<b>Fitur</b>	<b>Estimasi</b>
PB1	Pengembangan fungsi Login dan Register	1 hari
PB2	Pengembangan fungsi Pembuatan Tiket ( <i>New Ticket</i> )	4 hari
PB3	Pengembangan fungsi Daftar Tiket ( <i>List Ticket</i> )	1 hari
PB4	Pengembangan fungsi Tiket yang telah disetujui ( <i>Approval Ticket</i> )	5 hari
PB5	Pengembangan fungsi Tiket setiap User ( <i>My Ticket</i> )	3 hari
PB6	Pengembangan fungsi tambah, ubah dan hapus pada master <i>User</i>	2 hari
PB7	Pengembangan fungsi tambah, ubah dan hapus pada master Departemen	2 hari
PB8	Pengembangan fungsi tambah, ubah dan hapus pada master Kategori	2 hari
PB9	Pengembangan fungsi tambah, ubah dan hapus pada master Kondisi	2 hari
PB10	Pengembangan fungsi tambah, ubah dan hapus pada master Informasi (Hanya dapat ditambahkan oleh admin)	2 hari
PB11	Pengembangan fungsi menampilkan daftar Informasi	1 hari
PB12	Pengembangan fungsi metode Naive Bayes untuk klasifikasi tekisi	2 hari
PB13	Pengembangan fungsi aksi pada menu <i>Approval Ticket</i>	3 hari
PB14	Pengembangan fungsi validasi feedback	2 hari
PB15	Pengembangan fungsi progress setiap pengerjaan teknisi	4 hari
PB16	Pengembangan fungsi <i>Dashboard</i>	2 hari

#### 4. Menentukan Sprint Backlog

Pada tahap ini, penulis akan menentukan *Product Backlog* mana yang akan dikerjakan terlebih dahulu pada *Sprint* tersebut. Karena *Product Backlog* yang



ditentukan cukup banyak, tahap ini dilakukan untuk memfokuskan penulis dalam pengerjaan aplikasi.

#### 5. Mengerjakan Aplikasi (coding)

Pada tahap ini, penulis akan membuat aplikasi atau biasa disebut *coding*. Penulis akan memantau kinerja selama *sprint* yang telah berlangsung. Tahap ini dipergunakan untuk mengetahui waktu penyelesaian dan mengetahui sisa *item backlog* yang dikerjakan pada *sprint* selanjutnya.

#### 6. Mengerjakan Sprint Review

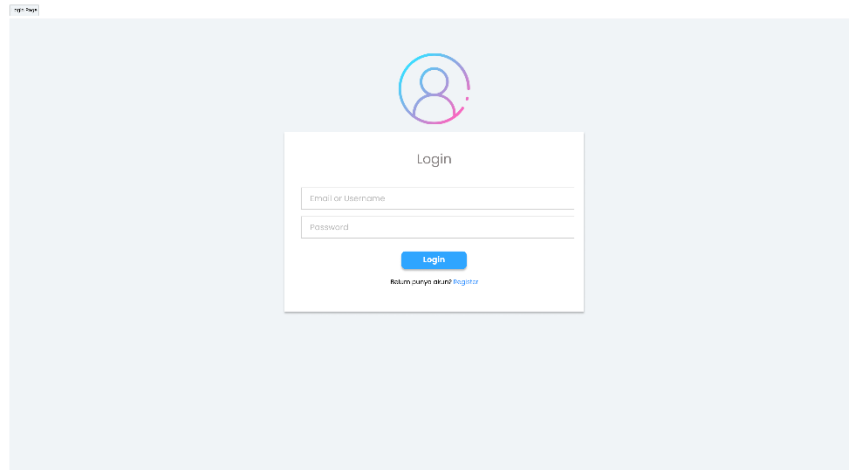
Pada tahap ini, penulis akan membahas hasil yang sudah dikerjakan pada kegiatan *sprint* sebelumnya. Penulis akan menjelaskan *Product Backlog* yang berhasil selesai dan sisa yang belum selesai. Pada tahap ini, memberi kesempatan pada seluruh *Scrum Team* masalah apa yang dihadapi pada saat *sprint* berjalan dan dapat menyampaikan solusi untuk masalah tersebut.

#### 7. Sprint Retrospective

Pada tahap ini, penulis akan meninjau pekerjaan yang telah selesai dilaksanakan. *Sprint* dapat dianggap selesai apabila semua *Product Backlog* telah selesai dikerjakan dan tidak ada penambahan *Product Backlog* baru. Namun, *sprint* akan berlanjut jika ada *Product Backlog* tambahan ataupun yang belum terselesaikan.

### 3.2.3 Perancangan Design User Interface / Mockup Aplikasi

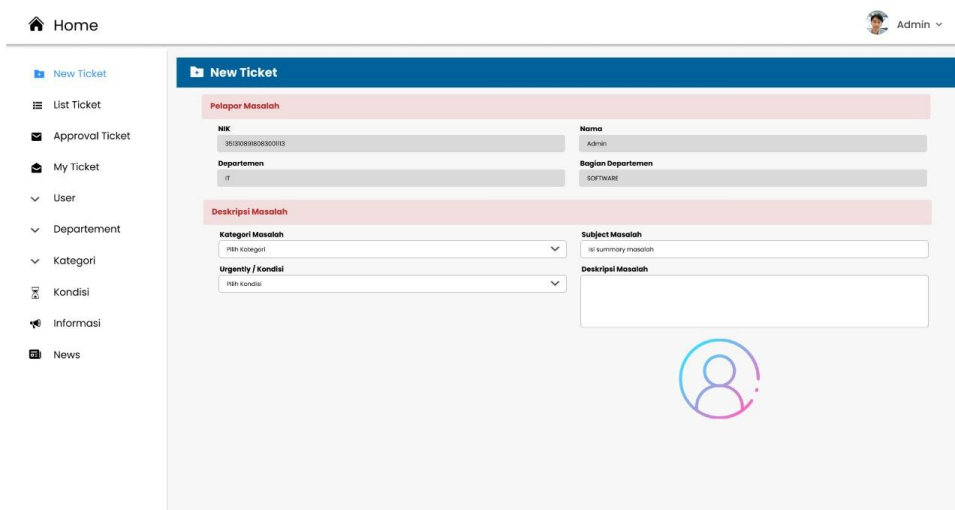
#### 1) Login



A login form mockup centered on a light blue background. At the top center is a circular icon with a person silhouette. Below it, the word "Login" is displayed. The form contains two input fields: "Email or Username" and "Password". A blue "Login" button is positioned below the fields. At the bottom of the form, there is a link that says "Belum punya akun? Registrasi".

**Gambar 3. 11** Rancangan Tampilan Login

#### 2) Halaman Open New Ticket



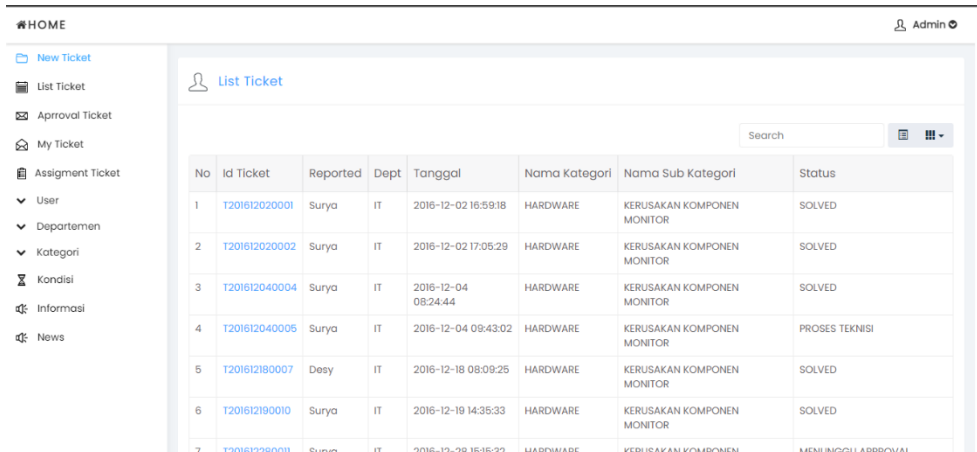
A screenshot of a web application interface for opening a new ticket. The top navigation bar includes a "Home" link and a user profile icon labeled "Admin". A sidebar on the left lists menu items: "New Ticket", "List Ticket", "Approval Ticket", "My Ticket", "User", "Departement", "Kategori", "Kondisi", "Informasi", and "News". The main content area is titled "New Ticket" and contains a form with the following sections:

- Pelapor Masalah**: Fields for "NIK" (value: 35.300819803630013), "Nama" (value: Admin), "Departemen" (value: IT), and "Bagian Departemen" (value: SOFTWARE).
- Deskripsi Masalah**: Fields for "Kategori Masalah" (value: IT), "Urgently / Kondisi" (value: IT), "Subject Masalah" (value: IT summary masalah), and "Deskripsi Masalah" (a large text area).

A circular icon with a person silhouette is located at the bottom right of the form area.

**Gambar 3. 12** Rancangan Tampilan New Ticket

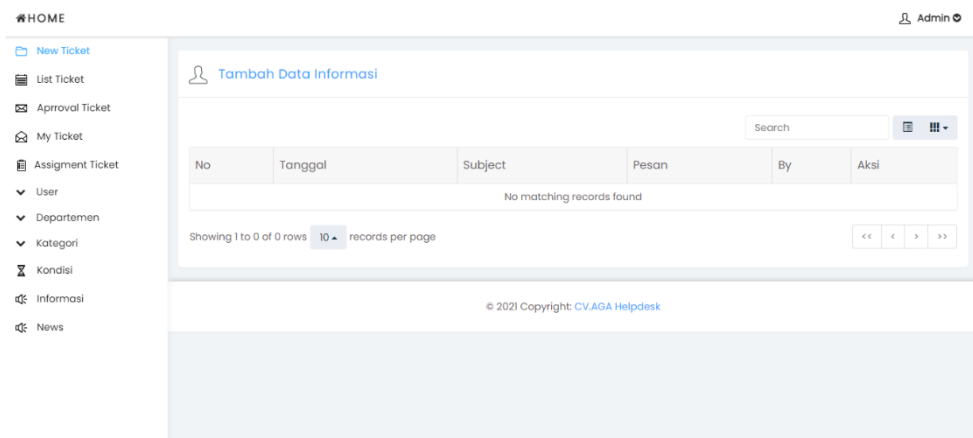
### 3) Halaman List Ticket



No	Id Ticket	Reported	Dept	Tanggal	Nama Kategori	Nama Sub Kategori	Status
1	T201612020001	Surya	IT	2016-12-02 16:59:18	HARDWARE	KERUSAKAN KOMPONEN MONITOR	SOLVED
2	T201612020002	Surya	IT	2016-12-02 17:05:29	HARDWARE	KERUSAKAN KOMPONEN MONITOR	SOLVED
3	T201612040004	Surya	IT	2016-12-04 08:24:44	HARDWARE	KERUSAKAN KOMPONEN MONITOR	SOLVED
4	T201612040005	Surya	IT	2016-12-04 08:43:02	HARDWARE	KERUSAKAN KOMPONEN MONITOR	PROSES TEKNSI
5	T201612180007	Desy	IT	2016-12-18 08:09:25	HARDWARE	KERUSAKAN KOMPONEN MONITOR	SOLVED
6	T201612190010	Surya	IT	2016-12-19 14:35:33	HARDWARE	KERUSAKAN KOMPONEN MONITOR	SOLVED
7	T201612280011	Surya	IT	2016-12-28 15:15:32	HARDWARE	KERUSAKAN KOMPONEN	MENUNGGU APPROVAL

**Gambar 3. 13** Rancangan Tampilan List Ticket

### 4) FAQ (Pertanyaan Umum)



No	Tanggal	Subject	Pesan	By	Aksi
No matching records found					

Showing 1 to 0 of 0 rows  records per page

© 2021 Copyright: CV.AGA Helpdesk

**Gambar 3. 14** Rancangan Tampilan FAQ (Data Informasi)

### 5) Halaman Master Karyawan

No	Nik	Nama	Email	Alamat	Jenis Kelamin	Departemen	Bagian	Jabatan	Aksi
1	K0007	Deden	deden@gmail.com	TES	LAKI-LAKI	IT	SOFTWARE	KEPALA BAGIAN	[Edit] [Delete]
2	K0009	Bobo	bobo@gmail.com	BANDUNG	LAKI-LAKI	IT	SOFTWARE	KEPALA BAGIAN	[Edit] [Delete]
3	K0001	Surya	surya@gmail.com	TANGERANG	LAKI-LAKI	IT	SOFTWARE	KEPALA DEPARTEMEN	[Edit] [Delete]
4	K0005	Rio	rio@gmail.com	TANGERANG	LAKI-LAKI	IT	SOFTWARE	KEPALA DEPARTEMEN	[Edit] [Delete]
5	K0008	Admin	admin@gmail.com	Admin	LAKI-LAKI	IT	SOFTWARE	KEPALA DEPARTEMEN	[Edit] [Delete]
6	K0002	Desy	desy@gmail.com	JAKARTA	PEREMPUAN	IT	SOFTWARE	KEPALA REGU	[Edit] [Delete]
7	K0003	Hendi	hendi@gmail.com	TANGERANG	LAKI-LAKI	IT	SOFTWARE	OPERATOR	[Edit] [Delete]
8	K0010	Jeje	jeje@gmail.com	BOGOR	PEREMPUAN	IT	SOFTWARE	OPERATOR	[Edit] [Delete]
9	K0011	Cory	cory@gmail.com	BANDUNG	LAKI-LAKI	IT	SOFTWARE	OPERATOR	[Edit] [Delete]

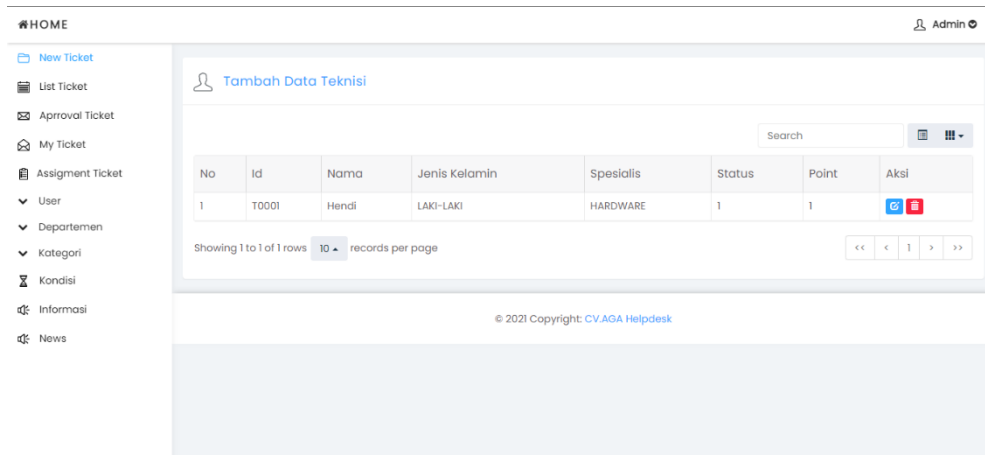
**Gambar 3. 15** Rancangan Tampilan Master Karyawan

### 6) Halaman Master User

No	Username	Email	Nama	Departemen	Level	Aksi
1	K0003	hendi@gmail.com	Hendi	IT	TEKNISI	[Edit] [Delete]
2	K0002	desy@gmail.com	Desy	IT	USER	[Edit] [Delete]
3	K0005	rio@gmail.com	Rio	IT	USER	[Edit] [Delete]
4	K0008	admin@gmail.com	Admin	IT	ADMIN	[Edit] [Delete]
5	K0009	bobo@gmail.com	Bobo	IT	USER	[Edit] [Delete]
6	K0010	jeje@gmail.com	Jeje	IT	TEKNISI	[Edit] [Delete]
7	K0014	user@gmail.com	user	IT	USER	[Edit] [Delete]
8	K0015	teknisi@gmail.com	teknisi	IT	TEKNISI	[Edit] [Delete]
9	K0012	david@gmail.com	David	PPIC	TEKNISI	[Edit] [Delete]

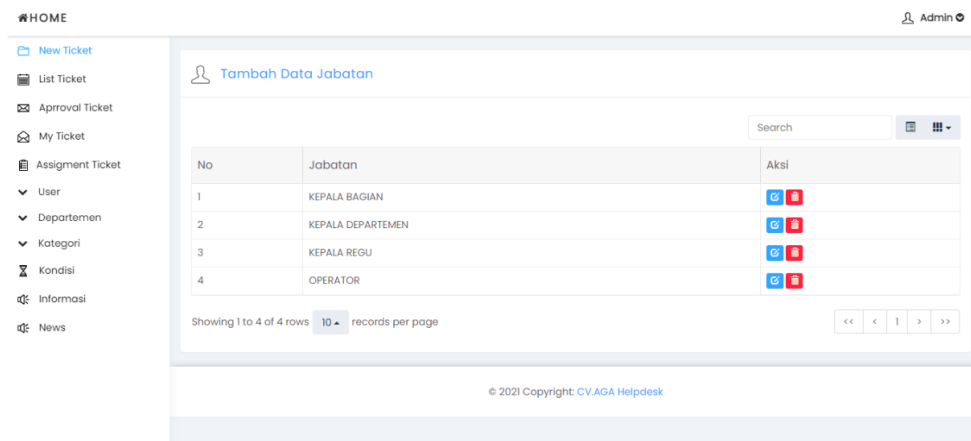
**Gambar 3. 16** Rancangan Tampilan Master User

## 7) Halaman Master Teknisi



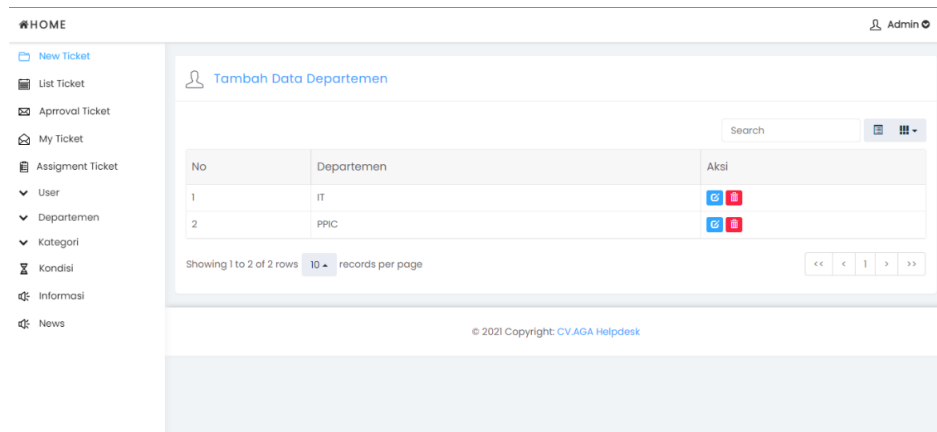
**Gambar 3. 17** Rancangan Tampilan Master Teknisi

## 8) Halaman Master Jabatan



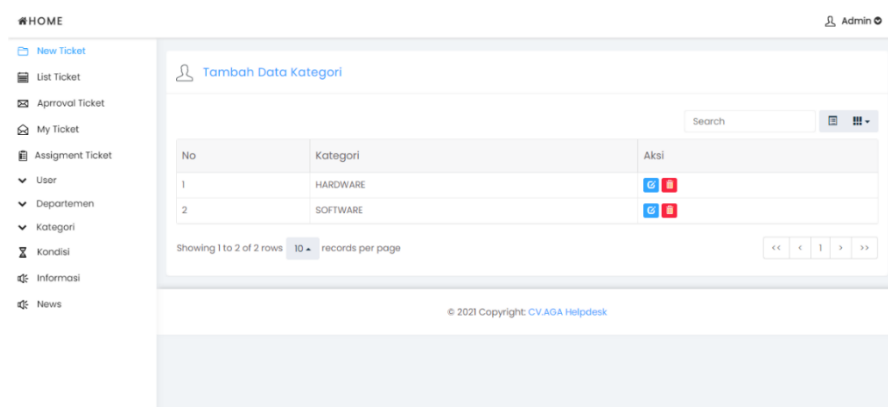
**Gambar 3. 18** Rancangan Tampilan Master Jabatan

## 9) Halaman Master Departemen



**Gambar 3. 19** Rancangan Tampilan Master Departemen

## 10) Halaman Master Data Kategori



**Gambar 3. 20** Rancangan Tampilan Master Kategori

### 3.2.4 Rancangan Pengujian

Dari pengembangan aplikasi yang dilakukan pada penelitian ini, maka dilakukan pengujian dari aplikasi yang dibangun dengan menggunakan metode Blackbox testing. Pengujian black box merupakan pengujian yang memastikan fungsionalitas dari aplikasi yang akan dikembangkan. Pengujian aplikasi ini dilakukan oleh pengembang dan pengguna dengan menguji berbagai fitur aplikasi ini.