

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

#### **3.1 Analisis**

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

Tidur merupakan bagian penting untuk setiap orang karena tidur yang baik dapat menjaga kesehatan tubuh tetap optimal. Namun yang seringkali terjadi seseorang tidak memperhatikan kebutuhan tidur mereka karena kebutuhan lain seperti jam kerja yang lebih panjang atau memiliki aktivitas lainnya. Padahal kurang tidur pada manusia jika dibiarkan akan sangat berdampak bagi aktivitas sehari-hari dan juga dapat memicu penyakit gangguan tidur. Penyakit gangguan tidur biasanya baru ditangani ketika berada di tahap kronis yang mana membutuhkan waktu yang lama untuk penyembuhan jika dibandingkan penanganan sejak dini. Pada tahap kronis penyembuhan bukan hanya tentang penggunaan obat tetapi juga tentang perubahan gaya hidup yang tentunya membutuhkan waktu yang tidak sebentar. Berdasarkan permasalahan di atas maka diperlukan alat bantu pengklasifikasian penyakit gangguan tidur guna mengklasifikasi penyakit secara dini guna menghindari kerugian yang tidak diinginkan.

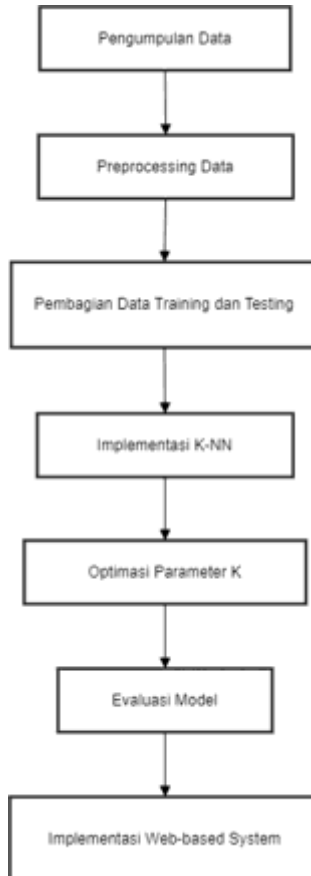
##### **3.1.2 Pemecahan Masalah**

Peneliti membuat sebuah website yang bernama sleep.in dengan bantuan algoritma *K-Nearest Neighbor* yang dinilai mampu membantu pengklasifikasian penyakit gangguan tidur secara efektif. Seseorang dapat diprediksi apakah memiliki penyakit gangguan tidur atau tidak dengan berdasarkan variabel jenis kelamin, usia, durasi tidur, kualitas tidur, tingkat stres, kategori BMI, denyut jantung, langkah harian, tekanan darah sistolik dan diastolik.

#### **3.2 Perancangan**

### 3.2.1 Perancangan Sistem

#### 3.2.1.1 Diagram Alir



Gambar 3. 1 Diagram Alir

Gambar 3.1 menjelaskan langkah demi langkah untuk pembangunan sebuah sistem berbasis web untuk mengklasifikasi gangguan tidur dengan metode *K-Nearest Neighbors* (K-NN). Berikut adalah penjelasan dari setiap langkah dalam diagram alir tersebut:

### a. Pengumpulan Data

Pengumpulan data akan diambil dari dataset *Sleep Health and Lifestyle Dataset* yang tersedia di platform *Kaggle*, yang merupakan sumber data terbuka yang banyak digunakan dalam penelitian dan pengembangan model pembelajaran mesin. Dataset ini berisi informasi tentang 374 individu dengan variabel jenis kelamin, usia, durasi tidur, kualitas tidur, tingkat stres, kategori BMI, denyut jantung, langkah harian, tekanan darah sistolik, tekanan darah diastolik dan gangguan tidur. Berikut rincian dataset :

Tabel 3. 1 Dataset Kategori Klasifikasi

Kategori	Jumlah Dataset
<i>Insomnia</i>	77
<i>Sleep Apnea</i>	78
Normal	219

Berikut merupakan daftar *dataset* ditampilkan melalui Tabel 3.2 :

Tabel 3. 2 Daftar *Dataset*

ID	Jenis Kelamin	Usia	Durasi Tidur	Kualitas Tidur	...	Gangguan Tidur
1	Perempuan	58	8	9	...	<i>Sleep Apnea</i>
2	Perempuan	51	7.1	7	...	Normal
3	Laki-laki	35	7.2	8	...	Normal
4	Laki-laki	43	7.8	8	...	<i>Insomnia</i>
5	Perempuan	45	6.6	7	...	<i>Insomnia</i>
6	Perempuan	36	7.1	8	...	Normal
7	Perempuan	45	6.8	7	...	<i>Insomnia</i>
8	Laki-laki	31	7.7	7	...	Normal
...	...	...	...	...	...	...
374	Laki-laki	28	6.2	6	8	Normal

## b. *Preprocessing Data*

Setelah melakukan pengumpulan data, tahapan selanjutnya yakni tahap *preprocessing data*. Tahapan *preprocessing data* yaitu sebagai berikut:

### 1. *Handling Missing Value* (Mengatasi Nilai yang Hilang)

Data yang mempunyai nilai kosong atau *missing values* perlu ditangani. Caranya bisa dengan mengisi nilai yang hilang menggunakan metode tertentu atau menghapus data yang hilang, agar algoritma dapat bekerja dengan baik.

### 2. *Encoding Data Kategorikal*

Beberapa kolom di dataset terdapat data kategorikal seperti jenis kelamin, kategori BMI, dan gangguan tidur yang berupa teks atau label. Algoritma KNN hanya bisa bekerja dengan data numerik, sehingga data kategorikal perlu dikonversi ke dalam format angka. Misalnya, kolom jenis kelamin (Laki-laki/Perempuan) bisa diubah menjadi 0 (Perempuan) dan 1 (Laki-laki).

### 3. *Normalisasi Data*

Metode normalisasi yang akan digunakan adalah *min-max scaling*, dimana nilai-nilai variabel dikonversi ke rentang 0-1 berdasarkan nilai minimum dan maksimum yang ada dalam dataset.

Rumus *Min-Max Scaling*:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Gambar 3. 2 Rumus Min-Max Scaling

Penjelasan Rumus *Min-Max Scaling*:

- $X_{scaled}$  adalah nilai setelah di-scaling
- $X$  adalah nilai asli dari data
- $X_{min}$  adalah nilai minimum dalam dataset
- $X_{max}$  adalah nilai maksimum dalam dataset

#### c. Pembagian *Data Training* dan *Testing*

*Data training* bertujuan untuk mengajarkan pola kepada model K-NN, sedangkan *data testing* digunakan guna mengukur kemampuan model dalam mengklasifikasikan data baru.

#### d. Implementasi K-NN

Berikut merupakan tahapan algoritma K-NN yang bisa digunakan dalam pelaksanaannya (Nacher & Akutsu, 2013):

1. Menentukan Nilai K, Memilih K yang nantinya menjadi dasar dalam penentuan kelas pada model klasifikasi K-NN.
2. Perhitungan Jarak, Menghitung jarak semua data dalam set pelatihan.
3. Penentuan Tetangga Terdekat, Memilih k tetangga terdekat berdasarkan jarak paling kecil.
4. Voting atau Penghitungan Rata-rata, Untuk klasifikasi, menentukan kategori mayoritas dari tetangga terdekat.

Dalam penyelesaian algoritma tersebut digunakan rumus K-NN yaitu:

Rumus jarak *Euclidean*:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Gambar 3. 3 Rumus *Euclidean*

Penjelasan Rumus Jarak *Euclidean*:

- $d(x, y)$  adalah jarak antara dua titik  $x$  dan  $y$
- $x_i$  adalah nilai fitur  $i$  dari titik  $x$
- $y_i$  adalah nilai fitur  $i$  dari titik  $y$
- $n$  adalah jumlah fitur

#### e. Optimasi Parameter K

Langkah selanjutnya adalah mengoptimalkan parameter  $K$ . Nilai  $K$  yang akan diuji adalah angka dalam rentang 1 – 20. Misalnya dalam uji optimasi parameter  $K$  didapatkan hasil akurasi tertinggi yaitu 94% jika menggunakan nilai  $K = 13$ . Maka dari itu, nilai  $K = 13$  yang akan dipilih untuk model final.

#### f. Metrik Evaluasi dan Validasi Model

Model yang telah dioptimalkan kemudian dievaluasi menggunakan *data testing*. Evaluasi ini bertujuan untuk menilai akurasi dan performa model dalam mendeteksi gangguan tidur. Performa model akan dievaluasi menggunakan metrik berikut:

- *Accuracy*: untuk mengukur keseluruhan ketepatan klasifikasi
- *Precision*: untuk mengukur ketepatan prediksi positif
- *Recall*: untuk mengevaluasi efektivitas model dalam mengenali data yang termasuk kasus positif
- *F1-score*: untuk memberikan skor keseimbangan antara *precision* dan *recall*

#### g. Implementasi Sistem Web

Tahap terakhir adalah mengimplementasikan sistem berbasis web. Sistem ini akan memanfaatkan model  $K$ -NN yang sudah dievaluasi untuk memberikan diagnosis gangguan tidur kepada pengguna melalui antarmuka web.

### 3.2.1.2 Diagram *Use Case*



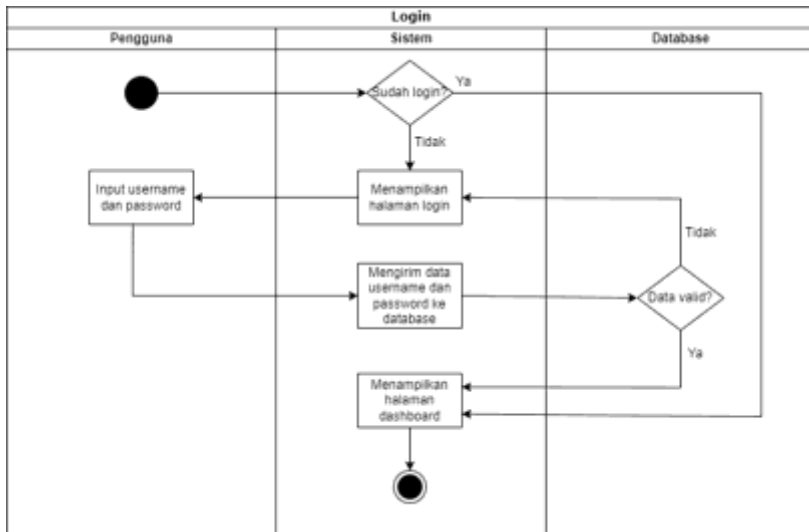
Gambar 3. 4 Diagram *Use Case*

Fitur untuk masing-masing pengguna dijelaskan sebagai berikut:

- User dapat melakukan input data, lihat hasil klasifikasi dan lihat riwayat klasifikasi.
- Admin dapat mengelola data training dan melihat statistik klasifikasi.

### 3.2.1.3 Activity Diagram

#### a. Activity Diagram Login

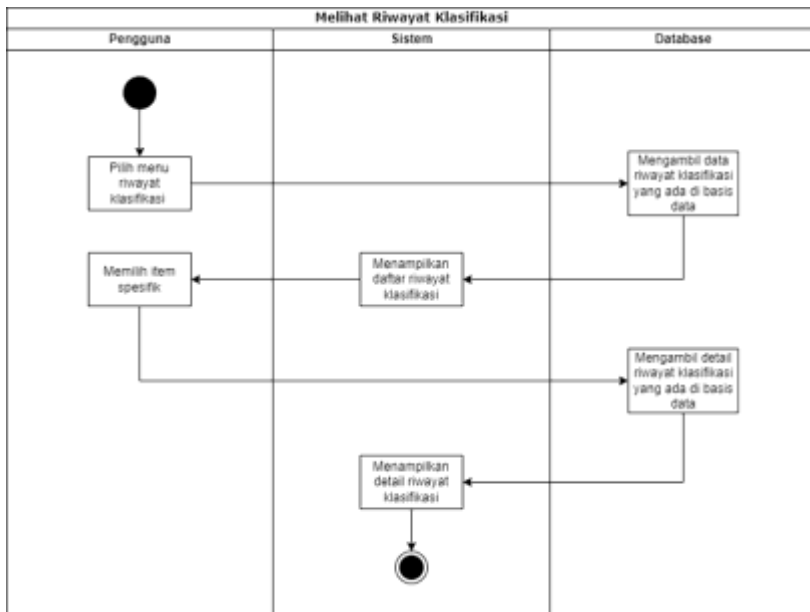


Gambar 3. 5 Activity Diagram Login

Untuk dapat menggunakan sistem, pengguna perlu *login* dengan memasukkan *username* dan *password* kemudian akan dicek benarkah data yang diinputkan pengguna. Apabila data yang digunakan benar maka pengguna dapat masuk ke halaman *dashboard*.



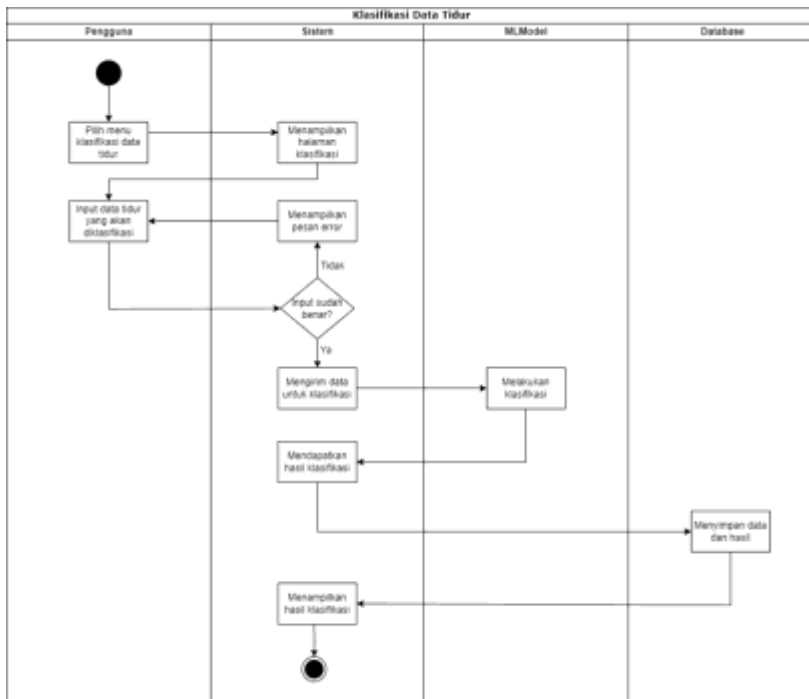
b. **Activity Diagram** Melihat Riwayat Klasifikasi



Gambar 3. 6 *Activity Diagram* Melihat Riwayat Klasifikasi

Pengguna dapat melihat data riwayat klasifikasi. Pada halaman data riwayat klasifikas maka pengguna dapat melihat riwayat klasifikasi yang pengguna lakukan dan pengguna juga dapat melihat detail riwayat klasifikasi yang dipilih.

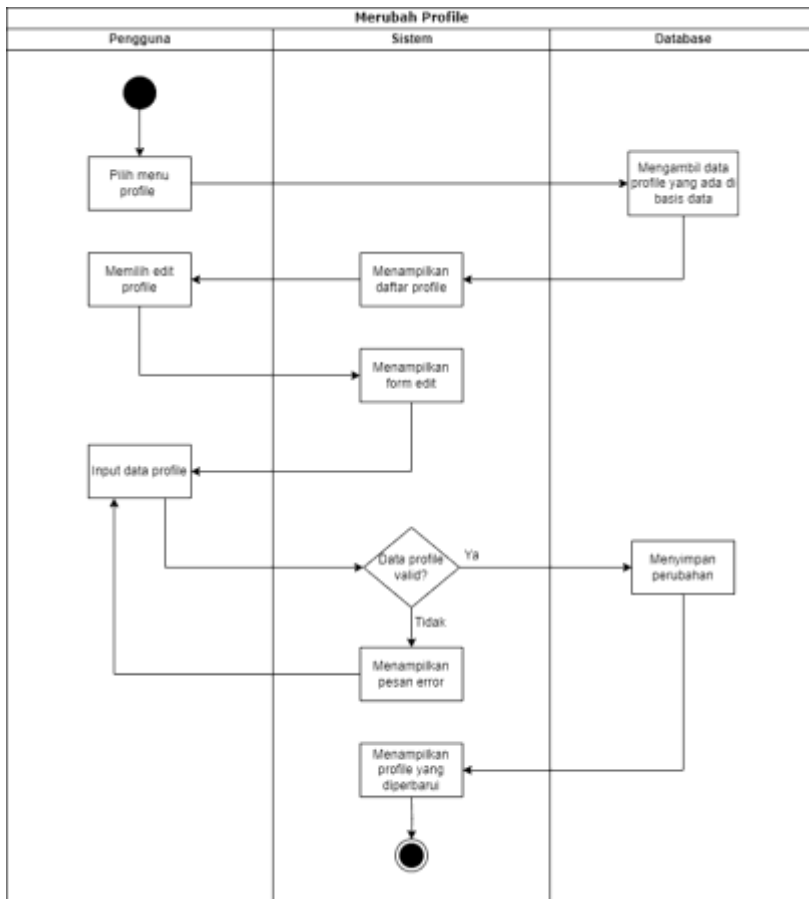
c. **Activity Diagram** Klasifikasi Data Tidur



Gambar 3. 7 *Activity Diagram* Klasifikasi Data Tidur

Pengklasifikasian penyakit dapat dilakukan dengan syarat yang sudah ditentukan yaitu dengan mengisi form yang sudah disediakan oleh sistem. Jika data yang pengguna masukkan sudah benar maka dapat dilakukan klasifikasi.

d. **Activity Diagram Merubah Profile**

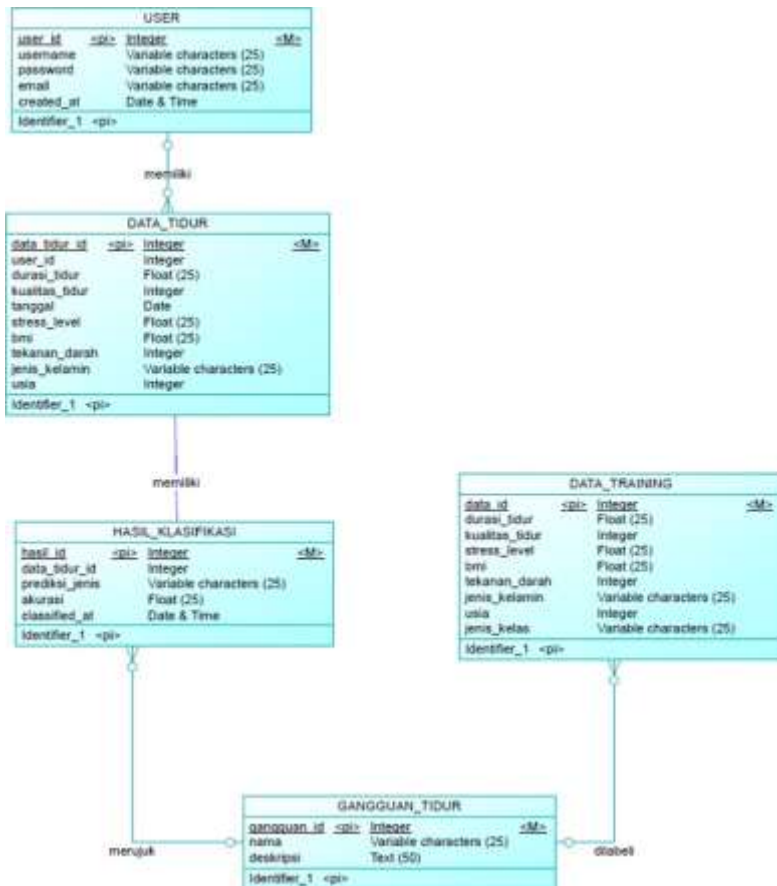


Gambar 3. 8 Activity Diagram Merubah Profile

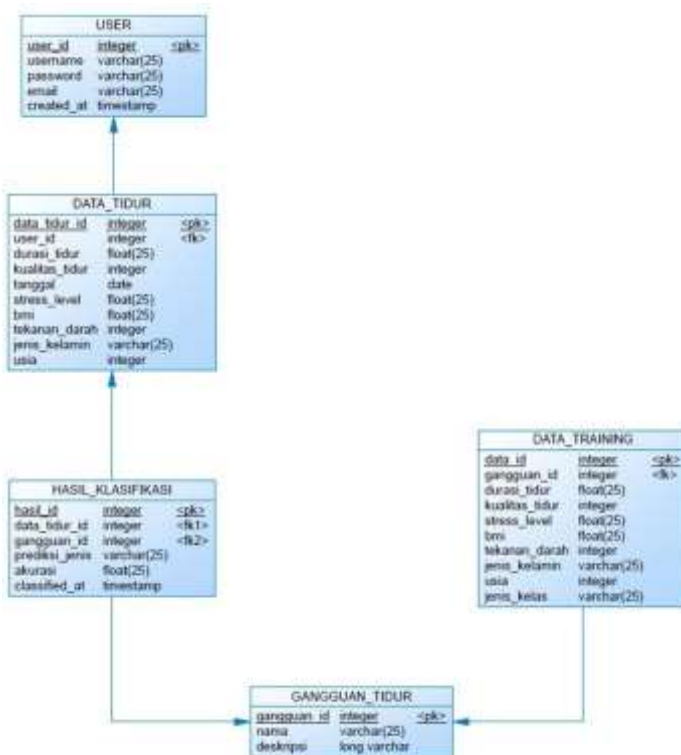
Pengguna dapat melakukan perubahan profile dengan syarat yang sudah ditentukan yaitu dengan mengisi form yang sudah disediakan oleh sistem. Sistem nantinya akan menyimpan hasil perubahan data yang baru kedalam *database*.

### 3.2.2 Perancangan Data

#### 3.2.2.1 Desain Database



Gambar 3. 9 Conceptual Data Model



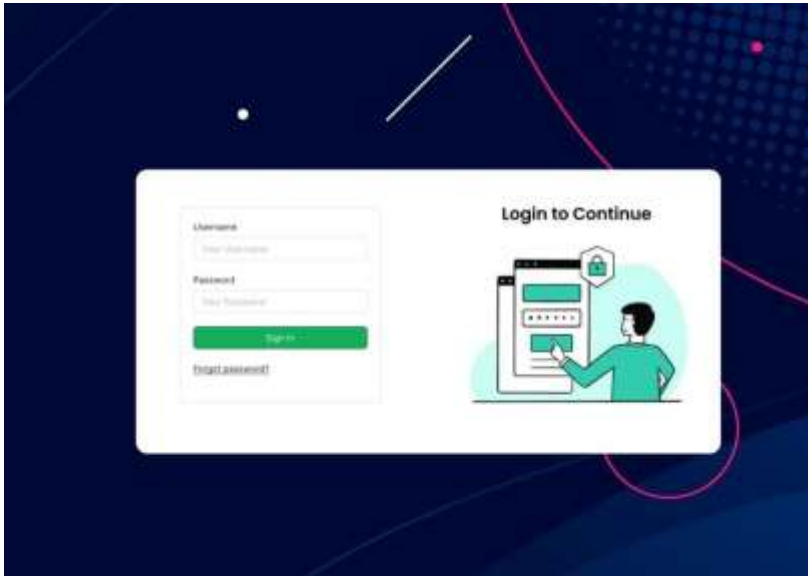
Gambar 3. 10 *Physical Data Model*

Dari desain *database* terdapat 5 tabel yaitu USER, DATA\_TIDUR, HASIL\_KLASIFIKASI, DATA\_TRAINING, dan GANGGUAN\_TIDUR. Penjelasan tabel-tabel dalam skema *database*:

- USER: Menyimpan informasi pengguna sistem.
- DATA\_TIDUR: Menyimpan data tidur yang diinput oleh pengguna.
- HASIL\_KLASIFIKASI: Menyimpan hasil klasifikasi dari algoritma K-NN.
- DATA\_TRAINING: Menyimpan *data training* untuk algoritma K-NN.
- GANGGUAN\_TIDUR: Menyimpan informasi tentang jenis-jenis gangguan tidur.

### 3.2.3 Perancangan User Interface / *Mock-up* aplikasi

#### 3.2.3.1 *User Interface Login*



Gambar 3. 11 Perancangan Halaman *Login*

Pada halaman ini akan disediakan form untuk menginputkan data yang diperlukan. Pengguna dapat berpindah ke halaman selanjutnya atau tidak bergantung dengan data yang dimasukkan.

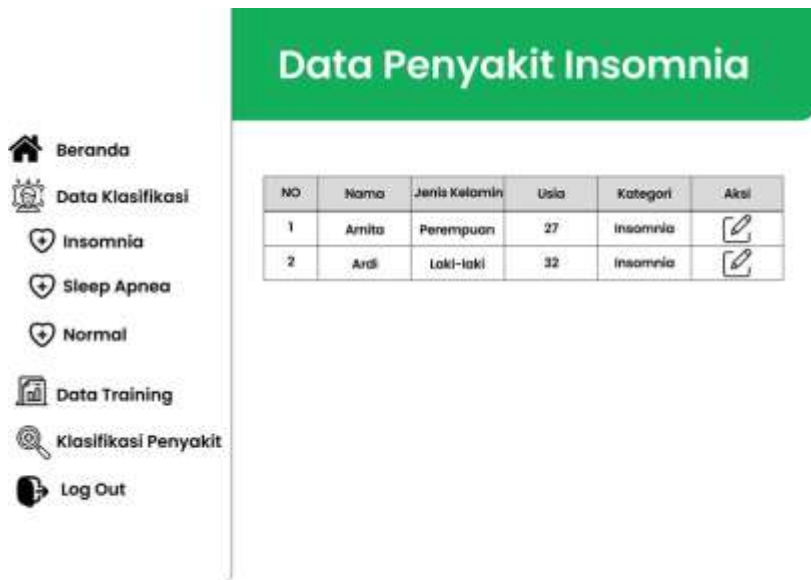
### 3.2.3.2 User Interface Beranda



Gambar 3. 12 Perancangan Halaman Beranda

Pada halaman beranda akan ditampilkan jumlah data yang ada di sistem yang mencakup *data training* serta klasifikasi. Selain menampilkan jumlah data, pada halaman beranda juga akan menampilkan tingkat akurasi klasifikasi.

### 3.2.3.3 User Interface Data Penyakit *Insomnia*

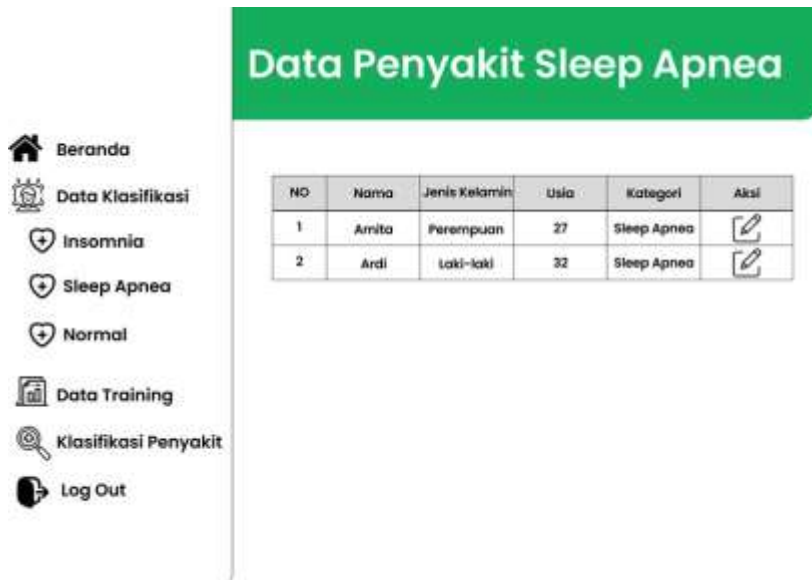


Gambar 3. 13 Perancangan Halaman Data Penyakit *Insomnia*

Pada halaman ini akan ditampilkan daftar data yang berhasil diklasifikasikan sebagai kategori penyakit *insomnia*. Daftar ditampilkan dalam bentuk tabel dan pengguna dapat mengubah data tersebut.



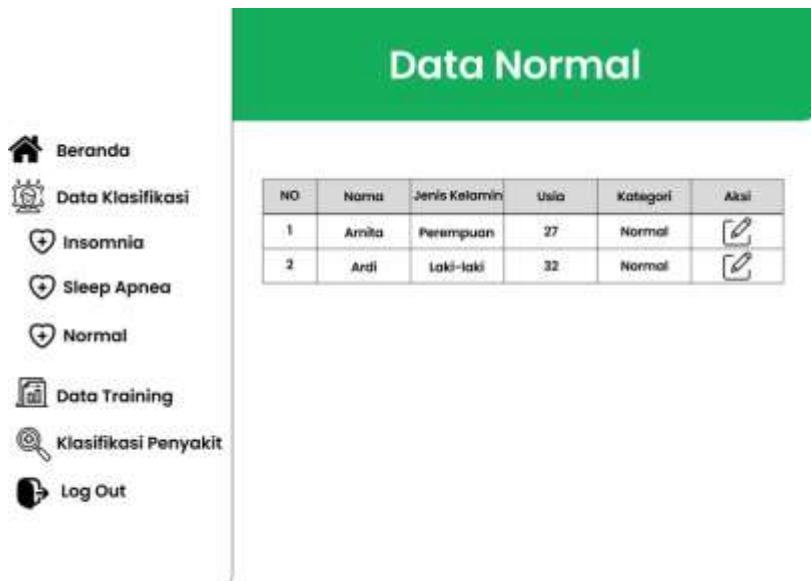
### 3.2.3.4 User Interface Data Penyakit *Sleep Apnea*



Gambar 3. 14 Perancangan Halaman Data Penyakit *Sleep Apnea*

Pada halaman ini akan ditampilkan daftar data yang berhasil diklasifikasikan sebagai kategori penyakit *sleep apnea*. Daftar ditampilkan dalam bentuk tabel dan pengguna dapat mengubah data tersebut.

### 3.2.3.5 *User Interface Data Normal*



Gambar 3. 15 Perancangan Halaman Data Normal

Pada halaman data normal akan ditampilkan daftar data yang berhasil diklasifikasikan sebagai kategori normal atau tidak memiliki penyakit gangguan tidur. Daftar ditampilkan dalam bentuk tabel dan pengguna dapat mengubah data tersebut.

### 3.2.3.6 User Interface Data Training



Gambar 3. 16 Perancangan Halaman *Data Training*

Pada halaman *data training* akan ditampilkan daftar *data training*. Daftar ditampilkan dalam bentuk tabel dan pengguna dapat mengubah data tersebut. Selain melihat daftar data terdapat juga tombol untuk menambahkan data yang mana penambahan *data training* harus dalam bentuk file .csv.

### 3.2.3.7 User Interface Klasifikasi Penyakit

Gambar 3. 17 Perancangan Halaman Klasifikasi Penyakit

Pada halaman klasifikasi penyakit akan ditampilkan form yang mana harus pengguna isi jika pengguna ingin mengklasifikasi penyakit. Data yang diperlukan yaitu nama, jenis kelamin, usia, durasi tidur, level stres, dll.

### 3.3 Rancangan Pengujian

Rancangan pengujian bertujuan untuk mengevaluasi kinerja dan keandalan sistem yang dikembangkan. Penelitian ini menggunakan dua pengujian yakni pengujian *Black Box* dan *Confusion Matrix*.

#### 1. *Black Box*

Metode *Black Box* akan digunakan untuk pengujian aplikasi. Dengan metode *Black Box*, penguji menentukan sejumlah kondisi input untuk kemudian menguji apakah keluaran yang dihasilkan sesuai dengan ketentuan yang telah direncanakan.

## 2. *Confusion Matrix*

*Confusion Matrix* merupakan salah satu alat visualisasi yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan cara membandingkan antara hasil prediksi dan data aktual. Matriks ini memberikan gambaran yang jelas mengenai seberapa baik model dalam mengklasifikasikan data, serta menunjukkan secara rinci pada bagian mana model melakukan prediksi yang benar maupun salah. Untuk masalah klasifikasi, matriks ini memiliki empat komponen utama yakni:

- *True Positive (TP)*: Seseorang yang benar-benar menderita penyakit gangguan tidur dan model memprediksinya dengan benar sebagai penderita penyakit gangguan tidur.
- *True Negative (TN)*: Seseorang yang tidak menderita penyakit gangguan tidur dan model memprediksinya dengan benar sebagai bukan penderita.
- *False Positive (FP)* (Error Tipe I): Seseorang yang sebenarnya tidak menderita penyakit gangguan tidur, tetapi model salah memprediksinya sebagai penderita.
- *False Negative (FN)* (Error Tipe II): Seseorang yang sebenarnya menderita penyakit gangguan tidur, tetapi model salah memprediksinya sebagai bukan penderita.

Tabel 3. 3 Tabel *Confusion Matrix*

	<b>Prediksi: Tidak Ada Gangguan Tidur</b>	<b>Prediksi: Insomnia</b>	<b>Prediksi: Sleep Apnea</b>
<b>Aktual: Tidak Ada Gangguan Tidur</b>	TN	FP	FP
<b>Aktual: Insomnia</b>	FN	TP	FP
<b>Aktual: Sleep Apnea</b>	FN	FP	TP

Berdasarkan nilai-nilai *Confusion Matrix* tersebut maka dapat dihitung metrik-metrik berikut:

- Akurasi (*Accuracy*): Seberapa sering model membuat prediksi yang benar secara keseluruhan.

Rumus:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Gambar 3. 18 Rumus Akurasi

- Presisi (*Precision*): Dari semua yang diprediksi sebagai Positif, berapa persen yang benar-benar Positif.

Rumus:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Gambar 3. 19 Rumus Presisi

- Recall (*Sensitivity*): Dari semua kasus yang sebenarnya Positif, berapa persen yang berhasil diprediksi sebagai Positif oleh model.

Rumus:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Gambar 3. 20 Rumus Recall

- F-1 Score: Rata-rata harmonik dari *Precision* dan *Recall*.

Rumus:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Gambar 3. 21 Rumus F1-Score