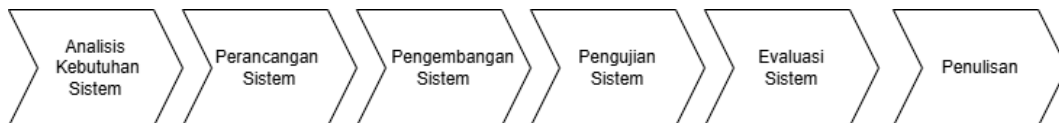


## BAB III METODE PENELITIAN

### 3.1 Analisis Kebutuhan Sistem

Metode penelitian pada penelitian ini menggunakan metode *Design and Development* (D&D) yang merupakan pendekatan yang sistematis dan iteratif dalam pengembangan dan pengujian produk atau sistem dalam konteks penelitian. Metode ini menggabungkan serangkaian langkah yang mencakup perencanaan, desain, pengembangan, dan evaluasi. Tujuan utama dari metode D&D adalah untuk menghasilkan produk atau sistem yang berfungsi penuh dan efektif serta dapat diimplementasikan dalam praktik nyata. Dalam penelitian ini, metodologi pengembangan D&D digunakan untuk menghasilkan teknologi baru, produk, atau proses yang mengintegrasikan penelitian sebagai bagian dari proses pengembangan. Menurut Darmayanti (2023) Pendekatan ini memastikan bahwa hasil penelitian langsung dapat diaplikasikan untuk memecahkan masalah praktis. Proses pembuatan sistem dalam penelitian ini dilakukan sesuai dengan langkah-langkah yang diilustrasikan pada Gambar 3.1 menunjukkan tahapan metode penelitian pada penelitian ini.



Gambar 3. 1 Tahapan Metode Penelitian D&D

Pada penelitian yang dilakukan, spesifikasi komputer milik peneliti yang dibutuhkan harus memadai untuk mendukung aplikasi yang digunakan dalam pengolahan data, pembuatan aplikasi android, pembuatan API, dan pelatihan kecerdasan buatan. Perancangan alat, pengembangan model, aplikasi Android, dan server dilakukan pada satu komputer yang sama dengan spesifikasi seperti pada Tabel 3.1 dibawah ini.

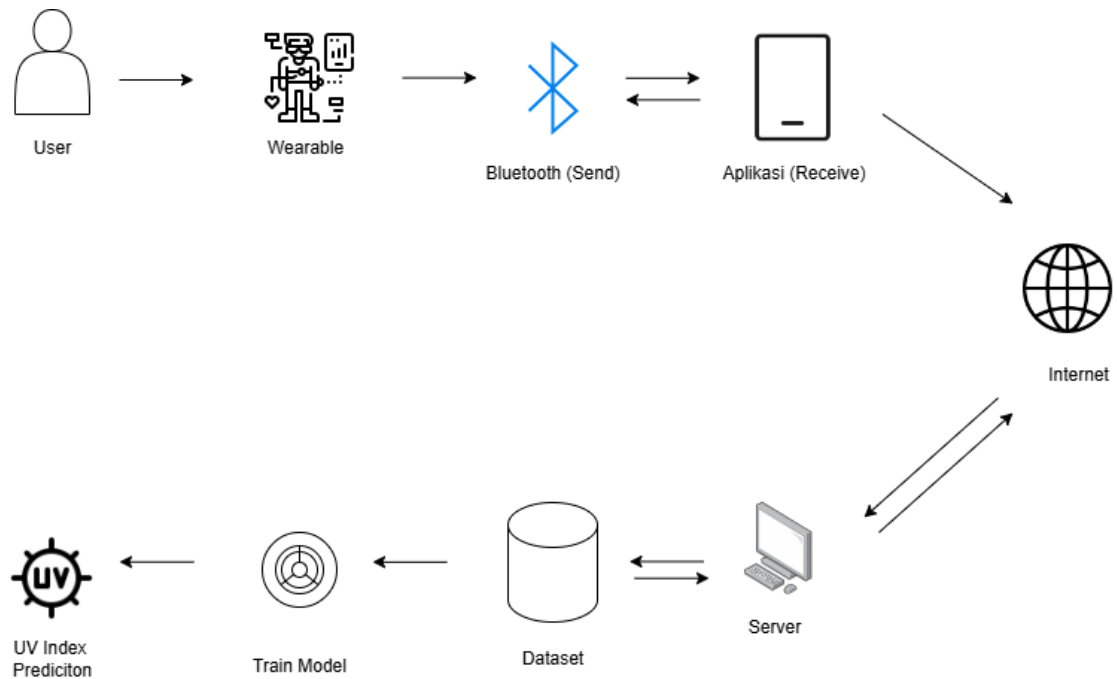
Tabel 3. 1 Spesifikasi Kebutuhan Pengembangan Sistem Perangkat Keras dan Perangkat Lunak

<b>Jenis</b>	<b>Nama</b>	<b>Detail</b>
<b>Hardware</b>	<i>Processor</i>	Intel i5 11400H
	<i>Memory</i>	16GB DDR4
	<i>Graphic Card</i>	RTX 3050 12GB
	<i>Storage</i>	SSD NVMe 250GB
	<b>Software</b>	<i>Operating System</i>
	Visual Studio Code	Versi 1.92.0
	Python	Versi 3.12.0
	Android Studio Hedgehog	Versi 17.0.6
<b>Library</b>	Statsmodels	Versi 0.14.1
<b>Python</b>		
<b>Library</b>	Retrofit	Versi 2.9.0
<b>Android</b>	Google Services Location	Versi 21.0.1
	Simple Gauge Android	Versi 0.3.1

### 3.2 Perancangan Sistem

Dalam perancangan sistem berdasarkan analisis kebutuhan diatas yang telah diuraikan sebelumnya, Proses perancangan sistem ini terdiri dari beberapa bagian utama seperti user, *wearable*, aplikasi, server, dan internet. Arsitektur perancangan sistem dapat dilihat pada Gambar 3.2 dimana *wearable* akan mengirimkan data menggunakan *Bluetooth* lalu data akan diterima oleh server untuk pemrosesan dan pengolahan data untuk membuat kecerdasan buatan, model yang digunakan

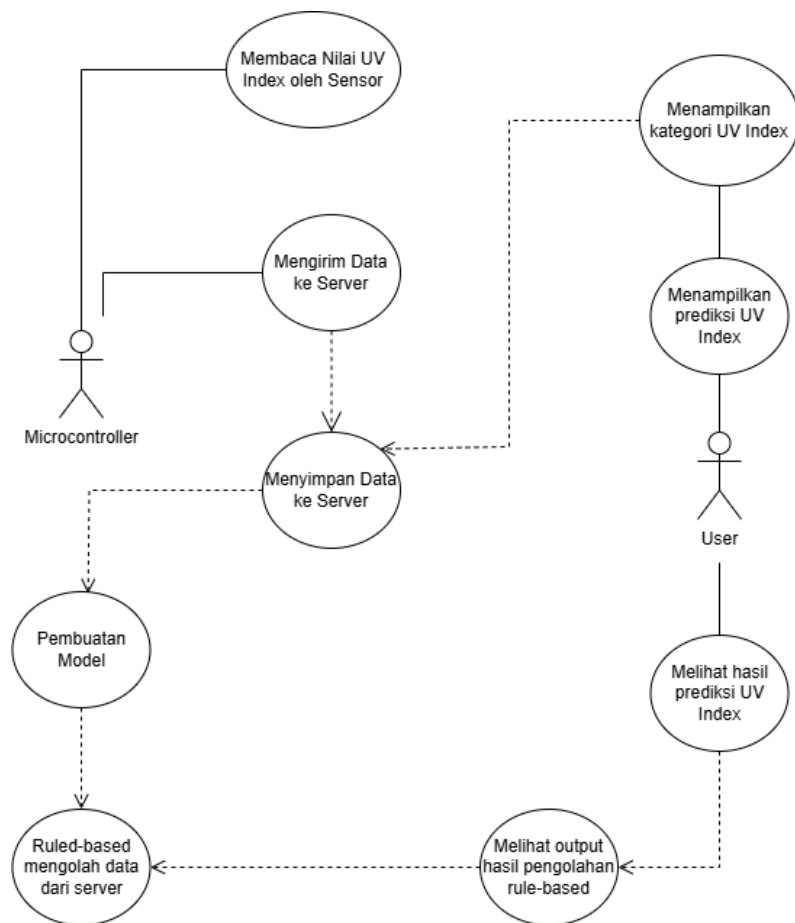
menggunakan ARIMA (*Autoggressive Integrated Moving Average*) untuk memprediksi nilai UVI (UV Indeks) 1 jam kedepan.



Gambar 3. 2 Desain Keseluruhan Sistem

### 3.2.1 Perancangan Sistem Pendeteksi Sinar UV

Setelah melakukan analisis kebutuhan aplikasi yang akan dibangun, langkah selanjutnya adalah proses desain aplikasi. Dalam tahap perancangan ini, hasil dari analisis kebutuhan sistem menjadi sebuah pemodelan menggunakan *Unified Modeling Language* (UML). Pemodelan UML ini mencakup pembuatan berbagai diagram seperti *use case*, dan *activity diagram* sehingga semua sistem berfungsi untuk menggambarkan interaksi antar komponen aplikasi serta alur kerja dari operasi yang mungkin terjadi. Tujuan dari penggunaan UML adalah untuk memvisualisasikan desain aplikasi secara struktural dan dinamis, memungkinkan para pengembang untuk memahami dan mengimplementasikan setiap fungsi dengan lebih efektif. *Use case diagram* pada aplikasi yang akan dibangun seperti pada Gambar 3.3 yang menunjukkan proses ketika *wearable device* mendapatkan nilai data melalui sensor selain itu gambar tersebut dapat dilihat dibawah ini.



Gambar 3. 3 *Use case Diagram* Sistem Pendeteksi sinar UV

### 3.2.2 Perancangan Definisi *Use Case Diagram*

*Use case* diagram berfungsi untuk menjelaskan sebuah fungsi interaksi yang ada pada sistem pendeteksi sinar UV. Untuk memperjelas maksud dari *use case* diagram yang telah dibuat, berikut merupakan tabel deskripsi dari *use case* diagram sistem pendeteksi sinar UV. Tabel tersebut dapat dilihat pada Tabel 3.2 dibawah ini.

Tabel 3. 2 Definisi *Use Case Diagram*

User Case ID	Use Case	Deskripsi
1	Membaca Nilai <i>UV Index</i> oleh Sensor	Sistem menerima nilai <i>UV Index</i> dari sensor yang terhubung ke ESP32 (microcontroller), berfungsi untuk membaca nilai <i>UV Index</i>
2	Mengirim Data ke Server	Sistem mengirim data ke Server melalui ESP32 dalam bentuk format JSON
3	Menyimpan data dari microcontroller	Sistem menerima data sensor yang dikirim oleh ESP32 dengan format JSON
4	Mengolah data dari server menggunakan rule-based method	Sistem mengolah data yang diambil dari Server berdasarkan nilai yang diterima oleh sensor secara <i>real-time</i>
5	Menampilkan kategori <i>UV Index</i>	User dapat melihat hasil kategori <i>UV Index</i> berdasarkan rule-based
6	Melihat hasil output hasil pengolahan rule-based	User dapat melihat hasil yang telah diolah sistem berupa kategori <i>UV Index</i> berdasar waktu terkini
7	Menampilkan prediksi <i>UV Index</i>	User dapat melihat hasil prediksi <i>UV Index</i> setelah pemrosesan data dan pembuatan model

### 3.2.3 Perancangan Skenario *Use Case Diagram*

Skenario *use case diagram* berfungsi untuk memaparkan bagaimana proses sistem bekerja dan menjelaskan jenis aksi maupun reaksi yang dilakukan oleh actor yang berperan pada sistem. Berikut merupakan tabel pemaparan skenario *use case diagram*, Tabel 3.3 merupakan skenario proses nilai *UV Index* terdapat dibawah.

Tabel 3. 3 Proses Membaca nilai *UV Index*

<b>User Case ID</b>	<b>01</b>
<b>User Case Name</b>	Membaca nilai <i>UV Index</i>
<b>Actors</b>	Microcontroller
<b>Goal</b>	Sistem membaca nilai <i>UV Index</i>
<b>Description</b>	1. Sensor terhubung dengan ESP32 2. Sensor terpapar sinar matahari baik itu di dalam maupun luar ruangan

Tabel 3. 4 Proses Mengirim data ke Server

<b>User Case ID</b>	<b>02</b>
<b>User Case Name</b>	Mengirim data ke Server
<b>Actors</b>	Microcontroller
<b>Goal</b>	Sistem membaca nilai <i>UV Index</i>
<b>Description</b>	1. Sensor terhubung dengan ESP32 menggunakan Bluetooth 2. Sensor terhubung dengan server lalu mengirim data dengan format JSON menggunakan method POST

Tabel 3. 5 Proses Menyimpan Data ke Server

<b>User Case ID</b>	<b>03</b>
<b>User Case Name</b>	Menyimpan data dari microcontroller
<b>Actors</b>	Server

<b>Goal</b>	Menerima data dengan format JSON dari ESP32
<b>Description</b>	<ol style="list-style-type: none"> <li>1. Menerima data dari microcontroller</li> <li>2. Menyimpan data sensor sebagai dataset</li> </ol>

Tabel 3. 6 Proses Mengolah Data

<b>User Case ID</b>	<b>04</b>
<b>User Case Name</b>	Mengolah data menggunakan metode rule-based
<b>Actors</b>	
<b>Goal</b>	Sistem mengolah data yang berhasil didapat melalui sensor lalu disimpan dalam bentuk dataset
<b>Description</b>	<ol style="list-style-type: none"> <li>1. Sistem mengambil data dari server</li> <li>2. Sistem memasukan data ke metode rule-based</li> </ol>

Tabel 3. 7 Proses Menampilkan Hasil

<b>User Case ID</b>	<b>05</b>
<b>User Case Name</b>	Melihat output hasil pengolahan rule-based
<b>Actors</b>	User
<b>Goal</b>	User dapat melihat informasi dari data yang sudah diolah menggunakan metode rule-based
<b>Description</b>	<ol style="list-style-type: none"> <li>1. Buka Aplikasi</li> <li>2. Sistem menampilkan <i>UV Index</i> waktu terkini beserta kategori dan rekomendasi penggunaan SPF</li> </ol>

Tabel 3. 8 Proses Menampilkan Hasil Kategori *UV Index*

<b>User Case ID</b>	<b>07</b>
<b>User Case Name</b>	Menampilkan kategori <i>UV Index</i>
<b>Actors</b>	
<b>Goal</b>	Menampilkan kategori <i>UV Index</i> dan rekomendasi penggunaan SPF

<b>Description</b>	1. Buka Aplikasi 2. Setelah User terhubung dengan microcontroller user dapat menampilkan <i>UV Index</i> waktu terkini beserta rekomendasi SPF
--------------------	---

Tabel 3. 1 Proses Menampikan Hasil Prediksi *UV Index*

<b>User Case ID</b>	<b>08</b>
<b>User Case Name</b>	Menampilkan prediksi <i>UV Index</i>
<b>Actors</b>	User
<b>Goal</b>	User dapat melihat informasi prediksi <i>UV Index</i>
<b>Description</b>	1. Buka Aplikasi 2. Sistem menampilkan <i>UV Index</i> waktu 4 jam kedepan

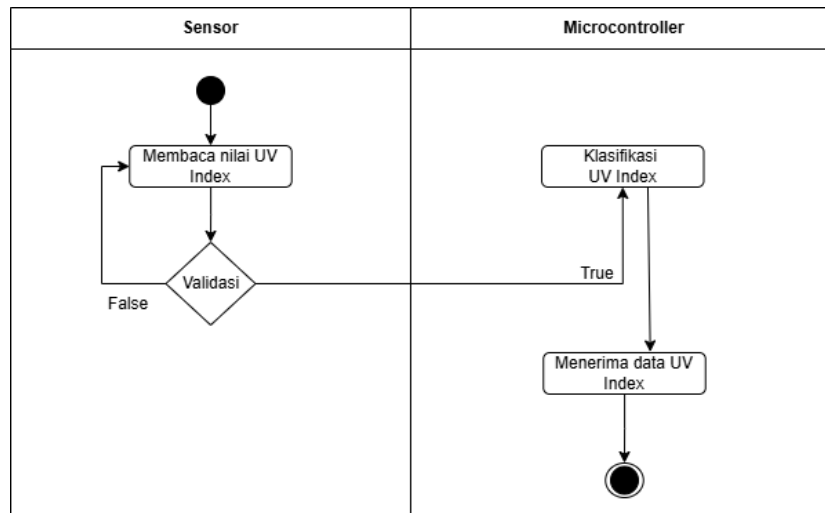
### 3.2.4 Perancangan *Activity Diagram*

*Activity diagram* menggambarkan aliran kerja (*work flow*) proses bisnis dan aktivitas – aktivitas pada suatu sistem. Perlu diperhatikan bahwa diagram ini menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, tetapi apa yang dapat dilakukan oleh sistem. *Activity diagram* memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah, mirip dengan flowchart karena memodelkan work flow dari satu aktivitas ke aktivitas lainnya. Diagram ini juga berguna untuk membantu memahami keseluruhan proses dan menggambarkan interaksi antara beberapa case.

#### 1. *Activity Membaca Nilai UV Index*

Pada proses ini, *activity* dilakukan untuk menangkap sinar ultraviolet oleh sensor dan kemudian dikirim ke ESP32 (*Microcontroller*) Gambar 3.4 menunjukkan kegiatan ketika proses membaca nilai *UV Index*, dimana sensor hanya membaca nilai *UV Index* sedangkan *microcontroller* bekerja sebagai klasifikasi *UV Index* dan menerima data tersebut.



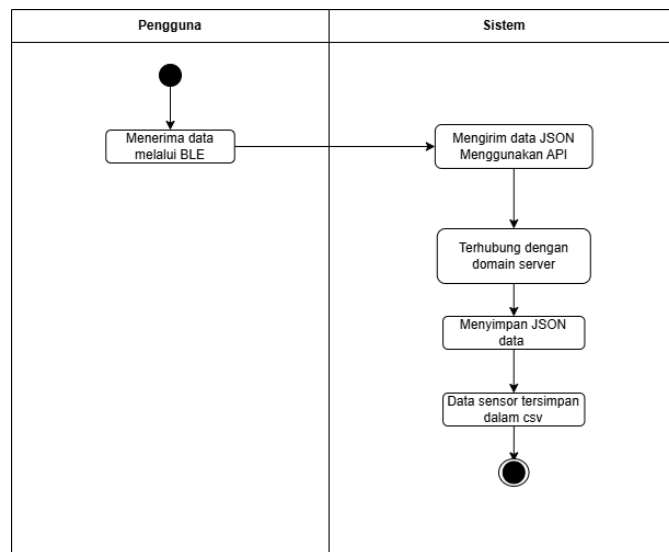


Gambar 3. 4 Activity diagram membaca nilai UV Index.

## 2. Activity Mengirim Data ke Server

Activity ini merupakan proses yang dilakukan oleh microcontroller terhadap server untuk pengiriman data. Gambar 3.5 menunjukkan proses ketika pengguna menerima data lalu sistem berfungsi mengirim data JSON lalu tersimpan di server sehingga data tersimpan dalam format dataset csv.

Gambar 3.5 merupakan *activity diagram* untuk pengiriman data ke server.

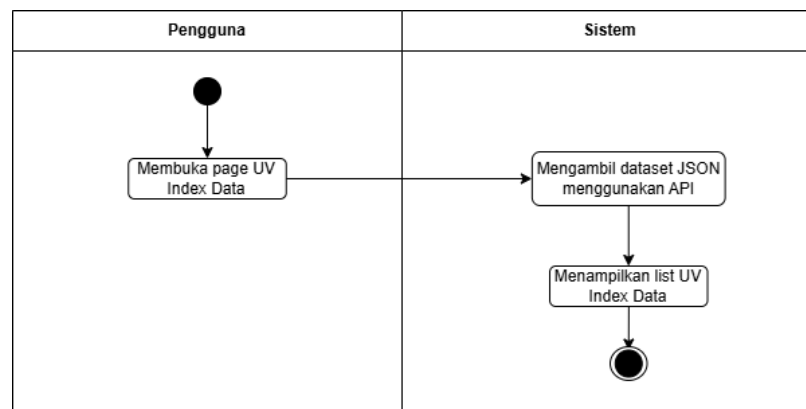


Gambar 3. 5 Activity Mengirim Data

### 3. *Activity diagram* Menampilkan data *UV Index* dan cuaca

Proses ini menjelaskan *activity* untuk menampilkan data *UV Index* dan cuaca menggunakan API dengan *method GET* untuk menampilkan data, data tersebut dapat dipilih oleh pengguna berdasarkan tanggal yang user inginkan, Gambar 3.6 menunjukkan kegiatan pengiriman data server.

Gambar 3.6 merupakan *activity diagram* untuk pengiriman data ke server.

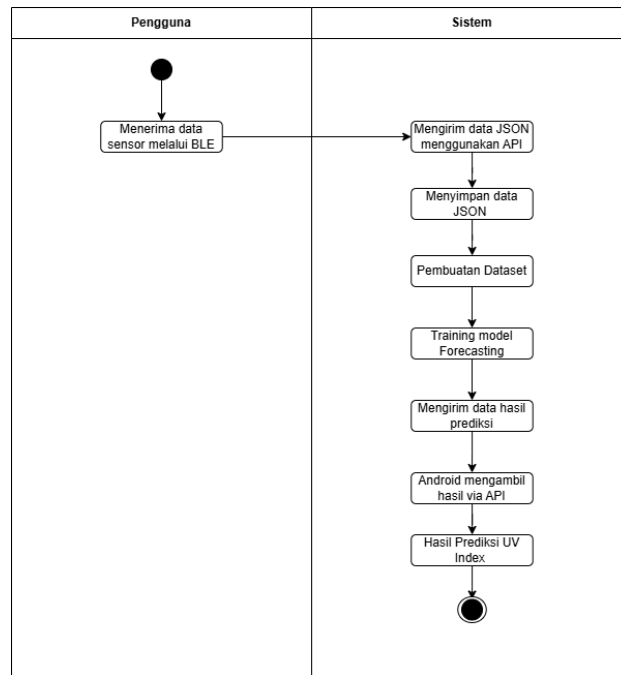


Gambar 3. 6 *Acitivity* Menampilkan Data UV Index dan Cuaca

### 4. *Activity diagram* Mengolah Data

Proses ini menjelaskan *activity* untuk mengelola data menggunakan *rule based*, dimana pengguna di sistem hanya bekerja sebagai penerima data melalui sensor BLE, sedangkan sistem bekerja untuk mengirim, menerima, lalu sistem akan menghasilkan nilai uv prediksi dengan cara menggunakan metode api *GET*. Gambar *Activity* tersebut dapat dilihat pada Gambar 3.7 dibawah ini.

Gambar 3.7 merupakan *activity diagram* untuk mengolah data dengan metode rule-based.

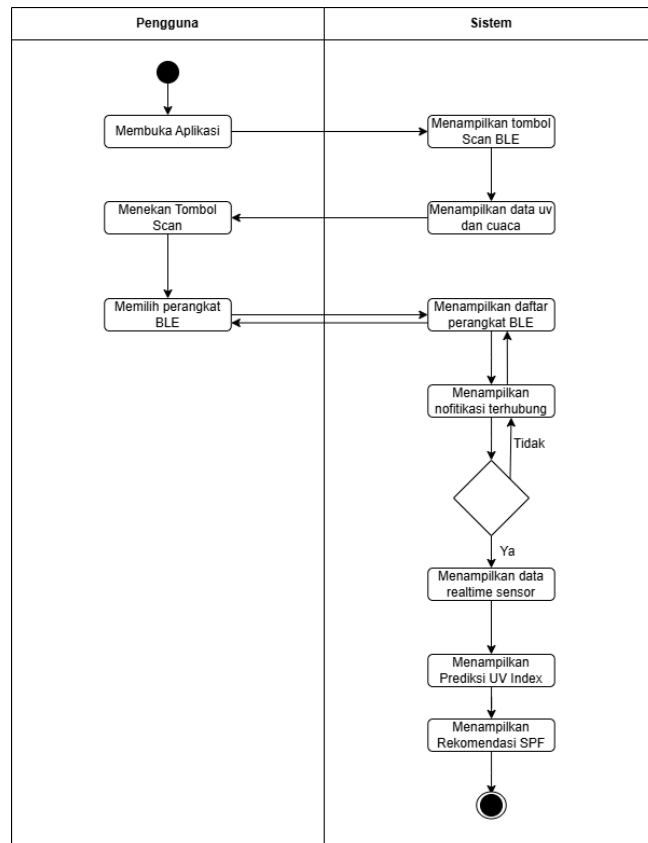


Gambar 3. 7 Activity Mengolah Data

## 5. Acitivity Diagram Keseluruhan Sistem

Proses ini menjelaskan *activity* untuk merancang keseluruhan sistem, dimana pengguna hanya dapat membuka aplikasi, menekan tombol scan BLE dan dikembalikan jika *user* memiliki *wearable*, Sistem dapat dilihat pada Gambar 3.8 yang menunjukkan sistem bekerja menampilkan data.

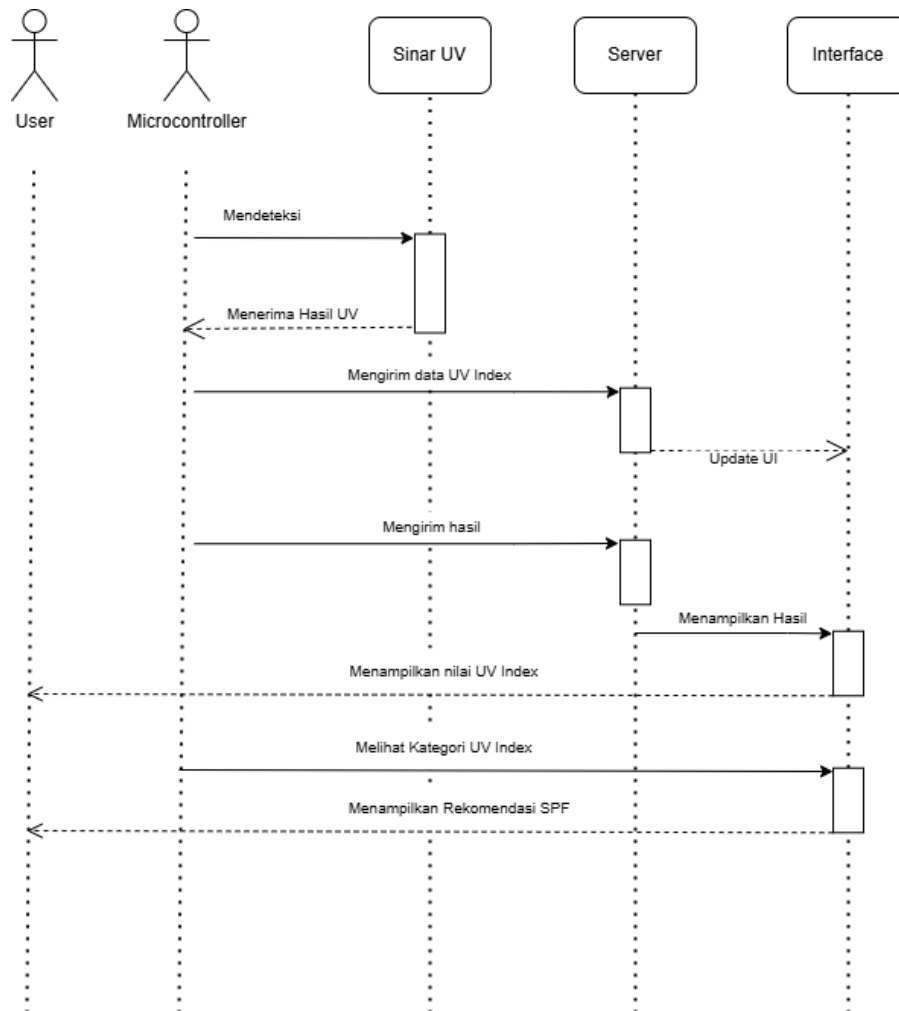
Gambar 3.8 merupakan *activity diagram* untuk merancang keseluruhan sistem.



Gambar 3. 8 Activity Keseluruhan Sistem

### 3.2.5 Perancangan Sequence Diagram

Skema alur kerja dari sistem digambarkan dengan sequence diagram. Diagram ini menggambarkan kolaborasi antara sejumlah objek dari suatu sistem. Sequence diagram bagian dari UML yang berguna untuk menampilkan rangkaian data yang dikirim antara objek dan interaksi antara objek, diagram juga meliputi alur kerja seluruh kegiatan sistem sehingga memberikan gambaran data yang mengalir pada sistem. Gambar 3.9 merupakan *sequence* diagram sistem pendeteksi sinar ultraviolet.



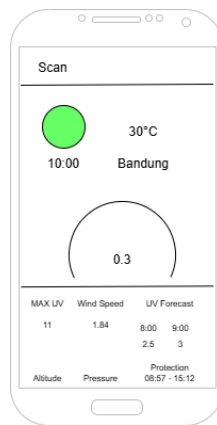
Gambar 3.9 *Sequence Diagram*

### 3.2.6 Perancangan Antarmuka

Rancangan antarmuka atau *interface* merupakan rancangan yang memberikan gambaran kasar bagaimana interaksi sistem pada entitas atau komunikasi antara pengguna (*user*) pada sistem. Tujuan dari perancangan antarmuka adalah merancang *interface* yang efektif untuk memberikan gambaran kasar sebagai tolak ukur untuk tampilan aplikasi yang akan dibuat, Penjelasan setiap antarmuka dapat dilihat pada halaman selanjutnya.

### 3.2.7 Perancangan *Design Page Home*

*Design home* pada sistem pendeteksi sinar UV ini merupakan tampilan utama yang menampilkan informasi data *UV Index* beserta kategori bahaya dan himbauan bagi kesehatan kulit. Data ditampilkan secara delay 5 detik. Gambar 3.10 merupakan design home aplikasi pendeteksi sinar UV.



Gambar 3. 10 Antarmuka *Home*

### 3.2.8 Perancangan *Design Page List Data*

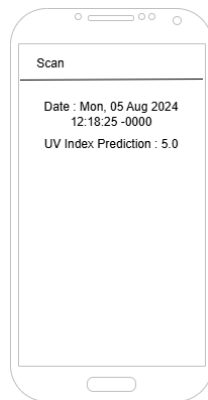
Tampilan list data berisi data-data *UV Index* yang telah ditampilkan pada halaman home yang berisikan tanggal, jam dan juga data *UV Index* yang telah diambil. Gambar 3.11 merupakan design list data aplikasi UV monitoring.



Gambar 3. 11 Antarmuka *List Data*

### 3.2.9 Perancangan *Design UV Index Prediction*

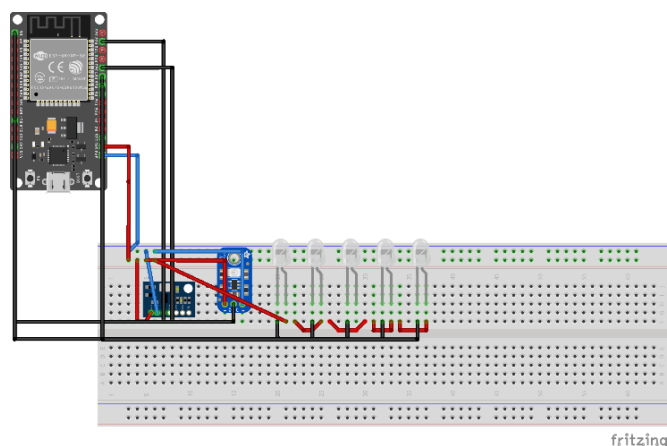
Tampilan *UV Index prediction* berisi data tanggal dan *UV Index* yang akan diprediksi selama 1 jam kedepan, data tersebut ditampilkan pada halaman *UV Index predict*. Gambar 3.12 merupakan design *UV Index Prediction*.



Gambar 3. 12 Antarmuka *UV Index Prediction*

### 3.2.10 Perancangan *Wiring Diagram*

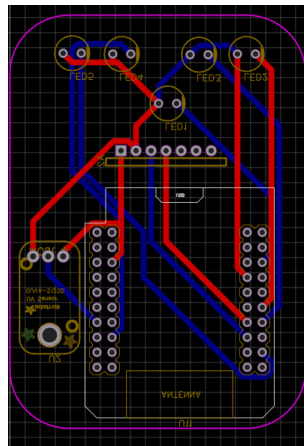
Pada perancangan sistem ini, *wiring diagram* diagram ini dirancang untuk memastikan bahwa proses pembuatan alat ketika proses prototype komponen terhubung dengan benar dan efisien, mengurangi risiko kesalahan koneksi yang dapat menyebabkan kerusakan atau kegagalan sistem. Gambar 3.13 menunjukkan gambaran koneksi antar pinout.



Gambar 3. 13 *Wiring Diagram*

### 3.2.11 Perancangan *Design PCB (Prototype Circuit Board)*

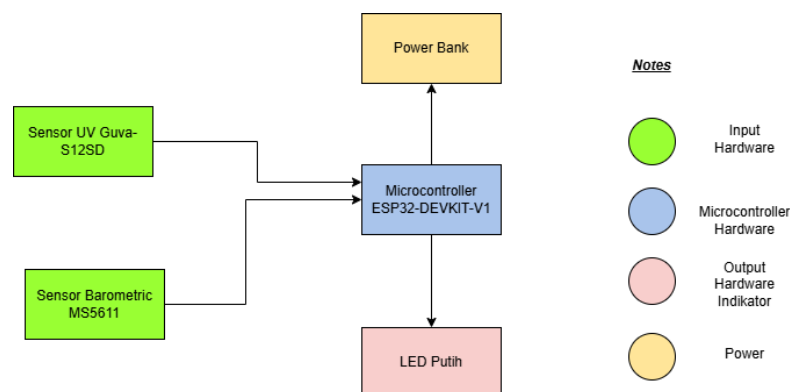
Perancangan *design PCB* yang digunakan dalam penelitian ini ditunjukkan melalui Gambar 3.14 yang memperlihatkan susunan dan hubungan antar komponen secara detail. Desain tersebut dikembangkan dengan tujuan untuk memaksimalkan efisiensi dan efektivitas sistem dalam kinerja penelitian. Dalam gambar tersebut, setiap komponen ditempatkan dengan pertimbangan khusus terhadap minimisasi gangguan elektrik dan memfasilitasi jalur sinyal yang lebih bersih. Gambar PCB dapat dilihat pada gambar dibawah ini.



Gambar 3. 14 Design PCB

### 3.2.12 Perancangan *Block Diagram*

Dalam *block diagram* pengembangan sistem ini terdapat beberapa komponen penting seperti modul sensor untuk input, mikrokontroler untuk pemrosesan, dan modul aktuator sebagai output, ditampilkan. Rincian lebih lanjut tentang komponen-komponen ini disajikan dalam Gambar 3.15



Gambar 3. 15 Block Diagram



### 3.2.13 Perancangan *Endpoint* API

Perancangan *Endpoint* API, dijelaskan rancangan dan fungsi dari setiap endpoint yang didefinisikan untuk aplikasi. Endpoint API ini dirancang untuk memfasilitasi pertukaran data antara aplikasi klien dan server. Berikut adalah rincian dari setiap endpoint pada penelitian kali ini dapat dilihat pada Tabel 3.2 untuk setiap *endpoint* yang tersedia.

Tabel 3. 2 Perancangan Endpoint API

No	Proses	API	
		Method	Endpoint
1	Mengambil data <i>UV Index</i>	GET	/api/v1/uvi
2	Mengambil data Cuaca	GET	/weather
3	Mengambil List data melalui dataset	GET	/api/getsensordata
4	Mengirim data dari ke server	POST	/api/sensordata
5	Menampilkan hasil prediksi UV	POST	/arima/prediction

Berikut adalah rincian fungsionalitas dari setiap endpoint yang terdapat dalam sistem, memberikan pemahaman mendalam tentang peran masing-masing dalam arsitektur, dapat dilihat pada halaman selanjutnya.

#### 1. Mengambil Data *UV Index*

Method: GET

Endpoint: /api/v1/uvi

Deskripsi: Endpoint ini digunakan untuk mengambil data indeks UV terkini dari server. Penggunaan metode GET menunjukkan bahwa operasi ini tidak mengubah state server dan hanya mengambil data.

#### 2. Mengambil Data Cuaca

Method: GET

Endpoint: /weather

Deskripsi: Endpoint ini bertujuan untuk mengambil data cuaca saat ini atau prakiraan cuaca dari server. Ini dapat digunakan oleh aplikasi untuk menampilkan kondisi cuaca bersamaan dengan data *UV Index*.

### **3. Mengambil List Data Melalui Dataset**

Method: GET

Endpoint: /api/getsensordata

Deskripsi: Endpoint ini digunakan untuk mengambil daftar data historis atau real-time yang tersimpan dalam dataset server. Ini memungkinkan pengguna untuk melihat data historis atau melakukan analisis tren dari data yang telah terkumpul.

### **4. Mengirim Data ke Server**

Method: POST

Endpoint: /api/sensordata

Deskripsi: Endpoint ini digunakan untuk mengirim data dari sensor ke server. Metode POST menunjukkan bahwa data dikirimkan ke server untuk disimpan atau diproses, yang mungkin mengubah state data di server.

### **5. Menampilkan Hasil Prediksi UV**

Method: POST

Endpoint: /arima/prediction

Deskripsi: Endpoint ini digunakan untuk mengirim data ke model prediksi ARIMA yang dihosting di server dan menerima hasil prediksi *UV Index*. Penggunaan POST di sini mengindikasikan bahwa data yang dikirimkan dapat memicu proses komputasi untuk generasi output.

#### **3.2.14 Perancangan Model ARIMA (Autogressive Integrated Moving Average)**

Perancangan model pada penelitian ini melibatkan pelatihan menggunakan model ARIMA sebagai prediksi *UV Index* beberapa menit kedepan sehingga pengguna dapat mengantisipasi penggunaan tabir surya sebelum beraktifitas diluar

ruangan. Proses perancangan model ini dimulai dengan pengumpulan dan pembersihan data, yang sangat penting untuk memastikan bahwa model mendapatkan data input yang akurat dan berkualitas. Perancangan ini memerlukan beberapa tahapan yaitu :

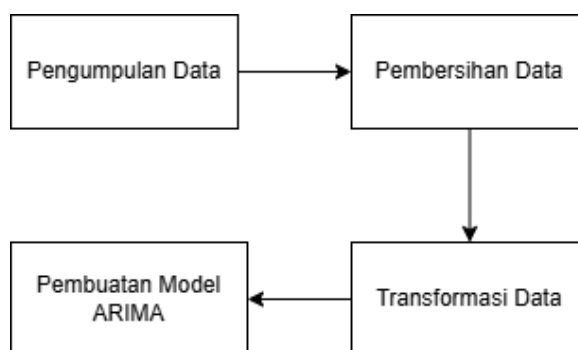
### **1. Pengumpulan Data**

Data yang digunakan dalam penelitian ini diperoleh melalui proses yang dimulai ketika pengguna memakai perangkat *wearable* yang terhubung ke internet. Setelah perangkat terhubung, data secara otomatis dikirim ke server dalam format JSON, memudahkan pengumpulan dan analisis selanjutnya. Informasi yang dikumpulkan dari perangkat ini mencakup berbagai parameter penting, termasuk tanggal, temperatur, ketinggian, tekanan atmosfer, dan *UV Index*.

Pengumpulan data yang efisien ini memastikan bahwa dataset yang dihasilkan mencerminkan kondisi lingkungan yang akurat, Dataset tersebut digunakan terdapat 70 data yang dikumpulkan di Cibiru Bandung dengan parameter yang telah disebutkan diatas sehingga dapat memudahkan untuk melakukan pembuatan model kecerdasan buatan.

### **2. Pengolahan Data**

Pengolahan data ini melibatkan serangkaian langkah yang sistematis untuk menyiapkan dataset *UV Index* untuk analisis lebih lanjut. Proses ini diilustrasikan secara rinci dalam Gambar 3.15, yang menunjukkan langkah-langkah yang diperlukan untuk mempersiapkan data sehingga dapat digunakan dalam model prediktif atau analisis statistik, dengan tujuan utama untuk meningkatkan akurasi prediksi yang dihasilkan dari dataset tersebut. Gambar alur pengolahan dapat dilihat pada Gambar 3.15 dibawah ini.



Gambar 3. 15 Proses Pengolahan Data

#### a. Proses Pengumpulan Data

Data dikumpulkan melalui perangkat *wearable* yang digunakan oleh pengguna dan terhubung ke internet. Setelah terhubung, data secara otomatis dikirim ke server dalam format JSON. Informasi yang dikumpulkan meliputi tanggal, temperatur, ketinggian, tekanan atmosfer, dan *UV Index*.

#### b. Pembersihan Data

Langkah selanjutnya adalah pembersihan data, yang melibatkan penghilangan kolom yang tidak diperlukan untuk analisis, seperti temperatur, ketinggian, dan tekanan. Proses ini memastikan bahwa hanya data yang relevan dan penting yang dipertahankan untuk analisis lebih lanjut.

#### c. Transformasi Data

Data yang telah dibersihkan kemudian diurutkan berdasarkan waktu, dari pagi hari hingga malam hari. Pengurutan ini penting untuk memastikan bahwa dataset mencerminkan kronologi harian yang akurat, yang sangat membantu dalam menganalisis tren harian *UV Index*.

#### d. Pembuatan Model ARIMA

Setelah data diurutkan dan siap, langkah selanjutnya adalah pembuatan model ARIMA. Proses ini melibatkan pemilihan parameter yang tepat ( $p$ ,  $d$ ,  $q$ ) berdasarkan data *UV Index*. Data kemudian dibuat stasioner untuk

memungkinkan pemodelan yang efektif. Model yang dikembangkan akan dilatih menggunakan data historis untuk memperkirakan nilai *UV Index*.

Melalui tahapan ini dataset *UV Index* dipersiapkan setelah diproses dan memungkinkan analisis dan prediksi dilakukan sehingga dapat menghasilkan sebuah model yang lebih akurat dan mendukung keputusan yang lebih tepat dan informasi yang lebih bermanfaat bagi pengguna.

### 3.3 Pengembangan Sistem

Dalam tahap pengembangan sistem ini, implementasi dilaksanakan berdasarkan desain yang telah dirancang sebelumnya. Pengembangan melibatkan pelatihan model ARIMA yang direncanakan untuk memprediksi *UV Index*. Model ini dilatih menggunakan dataset yang telah disiapkan dan diproses sesuai dengan tahapan pengolahan data yang telah dijelaskan.

Untuk pengembangan aplikasi Android, digunakan Android Studio sebagai IDE. Fitur-fitur yang telah dirinci dalam desain aplikasi akan diimplementasikan mengikuti spesifikasi UML (*Unified Model Language*) dan desain antarmuka pengguna yang telah disusun. Pada bagian server, pengembangan dilakukan menggunakan *micro framework* Flask dengan Python sebagai bahasa pemrogramannya. Server ini mengimplementasikan endpoint API sesuai dengan desain yang telah dibuat, menggunakan protokol HTTP/HTTPS dengan arsitektur RESTful API. Ini memungkinkan server untuk menerima dan mengelola permintaan dari aplikasi Android secara efisien.

Di sisi perangkat keras, perangkat *wearable* yang dikembangkan terintegrasi dengan sistem melalui konektivitas Bluetooth dan pengiriman data ke server melalui internet. Perangkat ini diprogram menggunakan Python, dan data yang dikumpulkan diolah dan ditransmisikan secara *real-time* untuk dimanfaatkan dalam prediksi *UV Index* yang berkelanjutan. Melalui Proses ini, sistem dikembangkan untuk menyediakan prediksi akurat mengenai *UV Index*, mendukung pengguna dalam mengambil keputusan yang tepat untuk menggunakan

SPF (*Sun Protection Factor*) yang tepat ketika beraktivitas luar ruang dan perlindungan terhadap paparan sinar UV.

### **3.4 Pengujian Sistem**

Pengujian sistem menggunakan metode *black box*, yang berfokus pada pengujian sistem tanpa mempertimbangkan proses internal dan implementasi kode secara mendetail (Fikri et al., 2024). Pengujian sistem dibagi menjadi beberapa bagian dari pengujian.

#### **3.4.1 Pengujian Fungsionalitas perangkat keras dan lunak**

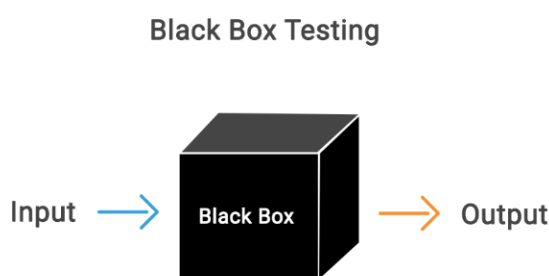
Pengujian ini bertujuan untuk memverifikasi kinerja dan reliabilitas pengiriman data dari perangkat *wearable* ke aplikasi smartphone melalui Bluetooth. Pengujian meliputi pengecekan stabilitas koneksi, kecepatan transfer data, serta kemampuan sistem untuk handle *reconnectivity* dan interferensi sinyal.

Proses ini sangat penting untuk memastikan bahwa data yang dikirim tidak hanya cepat tapi juga akurat, mengurangi kemungkinan data yang hilang atau rusak selama transmisi. Selain itu, pengujian ini juga mencakup keberhasilan komunikasi antara *microcontroller* dan juga Android.

#### **3.4.2 Pengujian Black Box**

Metode *Black Box* merupakan salah satu metode pengujian yang digunakan dalam penelitian ini untuk menguji perangkat lunak, yang berfokus pada pengujian fungsionalitas aplikasi tanpa memerlukan pengetahuan tentang struktur internal atau implementasi kode dari aplikasi tersebut. Dalam penelitian ini, metode pengujian *black box* telah digunakan secara intensif untuk mengidentifikasi dan mendemonstrasikan berbagai kesalahan yang mungkin terjadi dalam sistem perancangan yang telah dikembangkan, termasuk kesalahan pada fungsi aplikasi android, model kecerdasan buatan, API, dan komunikasi server. Pendekatan ini memungkinkan peneliti untuk berfokus pada input dan output sistem dan memverifikasi bahwa semua fitur beroperasi sesuai dengan spesifikasi yang telah ditetapkan. serta serta mengevaluasi respons sistem terhadap berbagai kondisi dan

skenario, untuk gambaran metode pengujian sistem *Black-Box Testing* dapat dilihat pada Gambar 3.16.



Gambar 3. 16 Ilustrasi Cara Kerja *Black-box Testing*

Pengujian ini *blackbox* menguji keakuratan dan kejelasan informasi *UV Index* yang ditampilkan di halaman utama aplikasi melalui *wearable* dan Halaman List Data (*History*) serta pengujian halaman *UV Index Prediction* Pengujian ini meliputi pemeriksaan tata letak, kecepatan respons saat interaksi pengguna, serta konsistensi data berbagai parameter yang ditampilkan. Oleh karena itu pengujian ini dapat mengidentifikasi masalah dalam fungsionalitas sistem yang tidak penulis sadari selama proses pengembangan, sehingga peneliti dapat memperbaiki kesalahan yang terdapat dalam penelitian. Dengan mengimplementasikan teknik pengujian *Black-box Testing*, peneliti dapat menilai kinerja sistem yang dibuat berdasarkan fungsi dapat berfungsi atau tidaknya fitur dan berapa lamanya waktu *delay* yang didapatkan dalam memenuhi kebutuhan pengguna tanpa bergantung pada pengetahuan internal tentang struktur atau logika internal sistem, Oleh karena itu pengujian ini dapat mengidentifikasi masalah atau kesenjangan dalam fungsionalitas sistem yang tidak penulis sadari selama proses pengembangan, Hal tersebut memungkinkan penulis untuk membuat perbaikan yang diperlukan sebelum diproduksi, sehingga dapat meningkatkan kualitas dan keandalan sistem yang telah dibuat. Adapun hasil temuan dan pembahasannya dalam pengujian sistem dapat dilihat pada BAB IV.

### 3.4.3 Pengujian API

Pengujian API melibatkan verifikasi bahwa semua endpoint berfungsi sesuai spesifikasi dan dapat menangani permintaan baik dari aplikasi mobile maupun dari server. Ini termasuk pengujian keamanan, kecepatan respons, dan kemampuan *error handling*.

Pengujian ini juga mencakup validasi format data yang dikirim dan diterima melalui API, memastikan bahwa semua pertukaran data dilakukan dengan parameter berhasil. Selain itu dengan pengujian API juga yang dapat mendukung sistem dalam kondisi penggunaan yang pengiriman data yang intensif.

### 3.4.4 Pengujian Kalibrasi Sensor

Tahap ini menguji keakuratan sensor setelah kalibrasi di berbagai kondisi lingkungan, tujuan pengujian ini untuk memastikan bahwa sensor menyediakan data yang akurat serta sesuai dengan kondisi klasifikasi sinar UV.

Pengujian ini tidak hanya memfokuskan pada keakuratan pembacaan sensor namun juga pada kemampuan sensor untuk mempertahankan kalibrasi tersebut dalam kondisi tertentu. Selain itu, proses ini mengevaluasi proses kalibrasi itu sendiri untuk memastikan bahwa sensor tersebut dapat menghasilkan hasil yang sesuai.

### 3.4.5 Pengujian Alat

Pengujian ini fokus pada pengujian lampu LED untuk menghasilkan output berdasarkan klasifikasi sinar UV dan sistem perangkat wearable dalam kondisi penggunaan nyata. Pengujian ini mencakup jumlah lampu LED yang berhasil menyala berdasarkan sinar UV dan data tersebut sesuai dengan nilai sensor atau tidak.

Pengujian ini penting untuk memastikan bahwa sistem dapat membaca sinar UV dalam berbagai kondisi lingkungan tanpa mengalami *error*. Pengujian juga mencakup pengujian antarmuka perangkat keras untuk memastikan bahwa sistem telah dirancang dengan sesuai dan mudah dimengerti oleh semua pengguna.



### 3.4.6 Pengujian Metode *Rule-Based*

Pengujian metode *rule-based* adalah merupakan tahap pengujian yang cukup penting untuk memastikan bahwa sistem berbasis aturan berfungsi sesuai dengan kebutuhan spesifik pengguna. Dalam penelitian ini, pengujian dilakukan untuk memverifikasi dan memvalidasi keakuratan aturan yang telah ditetapkan dalam mendeteksi indeks UV. Pengujian ini melibatkan penyusunan skenario yang spesifik, di mana setiap skenario berisi kondisi-kondisi tertentu yang harus dapat diidentifikasi oleh sistem. Tujuannya adalah untuk menjamin bahwa semua kondisi dan tindakan yang telah didefinisikan dalam sistem dapat dijalankan secara konsisten dan menghasilkan respons yang tepat sesuai dengan input yang diberikan.

Selain itu, pengujian ini juga mengevaluasi kinerja sistem dalam menangani berbagai situasi yang bisa terjadi dalam penggunaan sehari-hari. Pengujian ini tidak hanya mencakup verifikasi fungsionalitas dari aturan-aturan yang telah dibuat, tetapi juga penilaian terhadap performa sistem ketika dihadapkan pada *data real-time* atau simulasi.

### 3.4.7 Pengujian Hasil Model

Pengujian hasil model adalah tahap penting untuk memvalidasi model yang telah dikembangkan, bertujuan untuk menilai kemampuan model dalam memprediksi atau menghasilkan data yang sesuai dengan ekspektasi, berdasarkan data latih yang telah digunakan sebelumnya.

Pengujian tersebut melibatkan penggunaan model pada dataset yang telah dikumpulkan dari data sensor, yang digunakan selama fase pelatihan, tujuan dari pengujian ini adalah untuk mengukur kinerja model dengan membandingkan nilai prediksi yang dihasilkan oleh model terhadap nilai aktual dari data sensor. Sehingga memungkinkan evaluasi pada sistem tentang seberapa efektif model dalam menghasilkan prediksi yang akurat di lingkungan nyata.

## 3.5 Evaluasi Sistem

Evaluasi sistem merupakan tahapan penting yang dilakukan untuk memastikan bahwa seluruh komponen dan fungsi sistem bekerja sesuai dengan

spesifikasi yang telah ditetapkan dan memenuhi kebutuhan pengguna. Evaluasi ini melibatkan tes komprehensif yang dirancang untuk menilai setiap aspek sistem, mulai dari kinerja perangkat keras, perangkat lunak. Evaluasi ini tidak hanya mengukur kinerja teknis sistem tetapi juga bagaimana sistem tersebut dapat digunakan dalam kehidupan nyata. Hal ini memungkinkan penelitian ini untuk mendapatkan umpan balik yang baik dan memahami aspek-aspek yang mungkin memerlukan penyempurnaan lebih lanjut.

### **3.6 Penulisan Laporan**

Penulisan laporan adalah tahap terakhir dalam proses pengembangan sistem yang tidak hanya bertujuan untuk mendokumentasikan semua aktivitas dan hasil yang diperoleh selama penelitian, tetapi juga untuk menuliskannya dalam bentuk tertulis dari metodologi dan temuan penelitian.

Laporan ini memungkinkan peneliti untuk secara sistematis mencatat setiap aspek dari proses penelitian, mulai dari pengumpulan data awal hingga analisis hasil yang mendalam. Oleh karena itu, laporan ini berfungsi sebagai dokumentasi yang menggambarkan proses perkembangan penelitian, menguraikan metode yang digunakan, dan mendemonstrasikan bagaimana kesimpulan telah dicapai. Laporan yang dihasilkan dalam bentuk skripsi mencakup secara rinci setiap fase penelitian, termasuk konsep awal penelitian, pengembangan sistem, implementasi, pengujian dan evaluasi hasil.