

BAB IV PERANCANGAN

Bab ini akan membahas mengenai perancangan yang dilakukan oleh peneliti dalam melakukan penelitian.

4.1 Analisis Terdahulu

Peneliti menggunakan analisis penelitian terdahulu sebagai referensi untuk membuat sebuah sistem prediksi. Pada tahap analisis ini, peneliti menggunakan penelitian terdahulu untuk menentukan sistem yang dikembangkan serta kebutuhan yang diperlukan. Analisis ini berguna untuk menemukan kelebihan, dan kekurangan dari penelitian terdahulu. Hasil dari analisis ini akan berguna sebagai kesimpulan untuk melakukan pengembangan dari sistem sebelumnya.

Penelitian yang dilakukan oleh Putu Aditya Pratama dan Putu Satya Saputra telah menghasilkan sistem IoT (Internet of Things) untuk mengukur suhu dan kelembaban di lingkungan Universitas Panji Sakti Singaraja (Pratama & Saputra, 2023). Sistem ini telah melakukan pengukuran dan menunjukkan suhu 25,3 derajat Celsius dan kelembaban 54%, yang merupakan kondisi yang sangat baik. Selain itu, suhu dan kelembaban dapat dipantau secara online melalui platform cloud Blynk, baik melalui website maupun aplikasi Android. Sistem juga akan memberikan notifikasi apabila suhu dan kelembaban melewati batas normal di suatu ruangan. Hasil penelitian ini dapat bermanfaat dalam sektor pertanian dan peternakan guna mengembangkan jenis tanaman atau hewan yang membutuhkan suhu dan kelembaban tertentu pada ruangan.

4.2 Spesifikasi Kebutuhan Sistem Baru

Spesifikasi yang diperlukan untuk melakukan pengembangan sistem algoritma Random Forest untuk memprediksi suhu kabin mobil mencakup perangkat lunak dan perangkat keras. Tujuannya adalah memastikan sistem dapat berfungsi dengan baik, akurat, dan efisien. Dengan adanya spesifikasi yang memadai, diharapkan sistem yang akan dikembangkan akan menghasilkan sistem yang efisien dan optimal. Berikut adalah rincian spesifikasi yang dibutuhkan :

4.2.1 Spesifikasi Perangkat Lunak

Sistem yang dikembangkan memerlukan beberapa spesifikasi yang perlu untuk dipenuhi. Spesifikasi yang sesuai sangat diperlukan sehingga tidak ada kendala saat mengembangkan sistem. Berikut adalah spesifikasi yang digunakan dalam pengembangan sistem.

Tabel 4. 1 Spesifikasi Perangkat Lunak

No.	Perangkat	Keterangan
1.	Windows 10 (32 bit)	Windows 10 merupakan minimal versi sistem operasi yang digunakan dalam pengembangan sistem
2.	Visual Studio Code	Aplikasi <i>Code editor</i> untuk menulis kode program
3.	Pycharm	Python IDE dalam membaca data dan melakukan prediksi
4.	Laragon	<i>Universal development environment</i>
5.	MySQL	Basis data untuk data yang disimpan, dan data yang akan digunakan oleh algoritma

Tabel diatas adalah daftar dari perangkat lunak yang diperlukan dalam melakukan pengembangan sistem. Penjelasan detail dari setiap perangkat lunak adalah berikut:

1) Windows 10 (32 bit)

Sistem operasi Windows 10 (32 bit) adalah sistem operasi yang dikembangkan oleh Microsoft, dirilis pada 29 Juli 2015. Sistem operasi ini merupakan sistem operasi yang stabil sampai dengan sistem dikembangkan dan laporan ini dirancang oleh peneliti.

2) Visual Studio Code

Merupakan aplikasi *text editor* yang efisien dan relatif mudah digunakan dalam berbagai bahasa. Peneliti menggunakan *text editor* ini dalam membuat website berbasis PHP.

3) Pycharm

Pycharm merupakan sebuah python IDE yang digunakan peneliti untuk membuat algoritma membaca data dan melakukan prediksi dengan *random forest*.

4) Laragon

Laragon merupakan sebuah perangkat lunak yang membantu peneliti dalam membuat server web lokal untuk uji coba. Laragon dapat membuat serta menguji website yang sudah dikembangkan tanpa terhubung ke internet.

5) MySQL

MySQL merupakan sistem manajemen basis data relasional. MySQL merupakan basis data yang penting karena berisikan data yang diperlukan untuk melakukan sebuah prediksi.

4.2.2 Spesifikasi Perangkat Keras

Kehadiran perangkat keras yang mendukung dalam pengembangan sistem, sangatlah penting untuk dipenuhi. Adapun spesifikasi minimum perangkat keras adalah berikut :

Tabel 4. 2 Spesifikasi Minimum Perangkat Keras

No.	Perangkat	Keterangan
1.	Prosesor	Multi-core CPU
2.	Memori	8 GB RAM
3.	Penyimpanan	128 GB SSD

Tabel 4.2 merupakan daftar perangkat keras yang digunakan peneliti untuk membangun sistem prediksi suhu kabin mobil. Adapun maksud dari spesifikasi minimum yang disebutkan adalah berikut:

1) Prosesor

Prosesor yang memiliki lebih dari satu *core* diperlukan untuk dapat menjalankan aplikasi python. *Core* merupakan bagian inti CPU yang bertugas menerima instruksi dan melakukan tindakan sesuai dengan instruksi.

2) Memori

Memori yang diperlukan untuk menjalankan sistem prediksi suhu kabin mobil adalah 8 GB atau lebih karena program akan dijalankan terus menerus. Maka dari itu memori yang besar diperlukan sehingga tidak ada kendala pada perangkat keras.

3) Penyimpanan

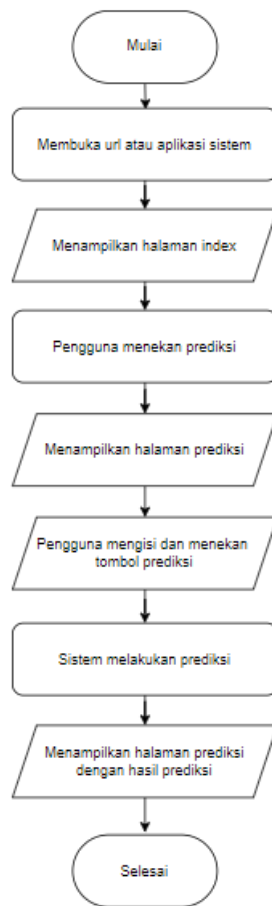
Penyimpanan yang optimal diperlukan sehingga tidak terdapat kendala saat mengembangkan sistem. Penyimpanan yang besar diperlukan untuk menampung data yang dimiliki dan program yang dikembangkan.

4.3 Perancangan dan Pembuatan Sistem

Perancangan dan pembuatan sistem merupakan langkah yang dilakukan peneliti untuk menjelaskan sistem yang akan dikembangkan. Berikut merupakan proses dari perancangan dan pembuatan sistem yang dilakukan peneliti.

4.3.1 Prinsip Kerja Sistem

Sistem yang dikembangkan oleh peneliti diawali dengan data yang diperoleh dari sistem tertanam yang telah dibangun oleh peneliti untuk mendapatkan data temperatur, co2, dan waktu diperolehnya data. Berikut adalah gambaran dari prinsip kerja yang akan dikembangkan oleh peneliti.

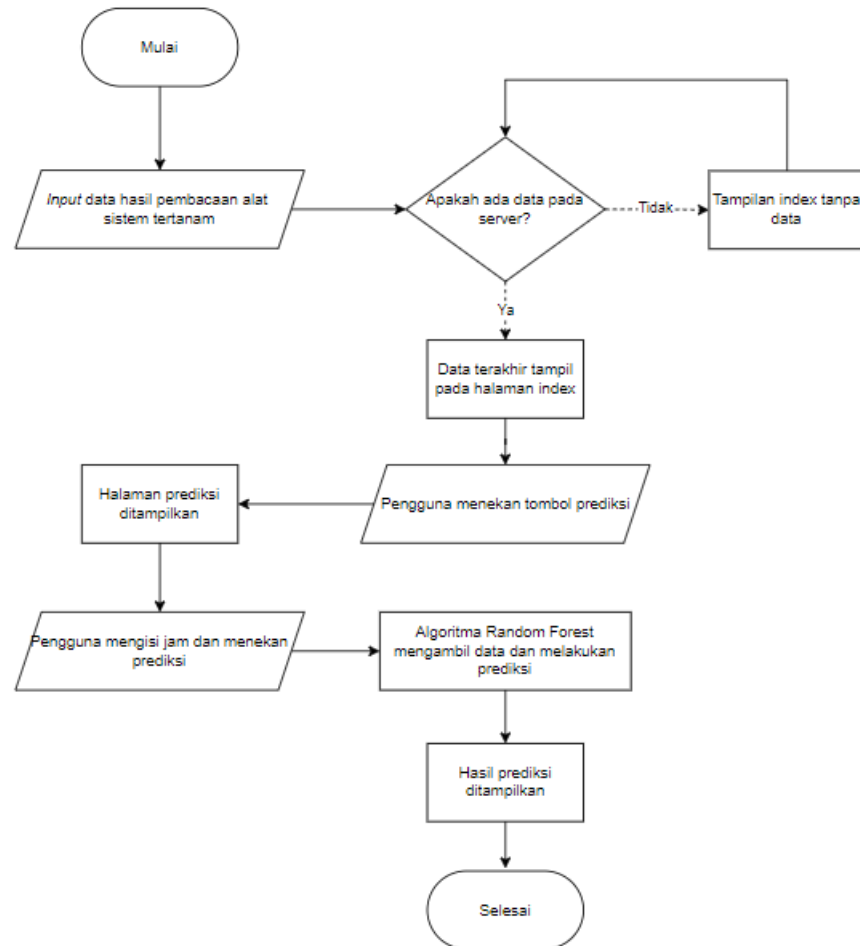


Gambar 4. 1 Prinsip Kerja Sistem

Gambar 4.1 merupakan prinsip kerja dari sistem prediksi suhu kabin mobil menggunakan algoritma *Random Forest*. Sistem ini akan bekerja ketika dataset berhasil masuk ke dalam server. Setelah server memiliki dataset, maka dataset yang masuk akan digunakan oleh algoritma *Random Forest* sebagai pelatihan sistem. Halaman index berguna sebagai halaman utama pada *website* yang menampilkan data terbaru dataset yang didapat. Data yang ditampilkan adalah temperatur dan waktu temperatur yang diambil. Ketika pengguna ingin melakukan prediksi temperatur pada jam tertentu, pengguna bisa memasuki halaman prediksi. Halaman prediksi berisikan jam yang akan ditentukan oleh pengguna, dan hasil prediksi yang dilakukan oleh algoritma *Random Forest*. Jam yang dimasukkan akan digunakan oleh algoritma untuk memprediksi temperatur berdasarkan jam yang dimasukkan pengguna dan jam terakhir data yang masuk. Hasil prediksi dari algoritma akan ditampilkan pada halaman prediksi sesudah pengguna mengisi jam yang akan diprediksi.

4.3.2 Diagram Alir Sistem

Diagram alir sistem yang dikembangkan peneliti akan digambarkan seperti pada gambar 4.2



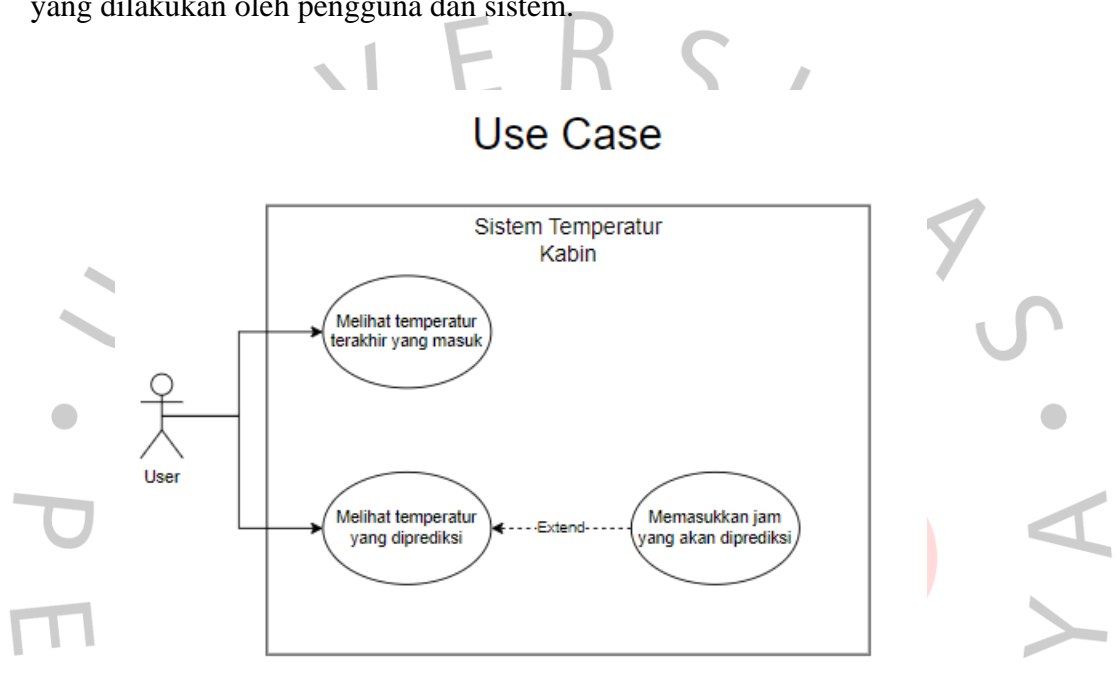
Gambar 4. 2 Diagram Alir Sistem

Proses alur sistem digambarkan pada gambar 4.2 yang merupakan diagram alir sistem. Awal kerja sistem bermula dengan sistem cerdas memasukkan data pada basis data yang dirancang oleh peneliti. Sistem akan melakukan pemeriksaan apakah ada data tersimpan pada server. Jika tidak terdapat data pada sistem, maka sistem memberikan tampilan index tanpa data, tetapi jika terdapat data pada server, maka sistem akan menampilkan data terbaru pada halaman index. Ketika pengguna menekan tombol prediksi, maka sistem akan memberikan halaman prediksi. Pengguna diminta untuk mengisi jam dan menekan tombol prediksi untuk melakukan prediksi suhu kabin mobil. Setelah pengguna menekan tombol

prediksi, sistem akan mengirimkan data yang didapat dari pengguna kepada algoritma untuk melakukan prediksi, dan ketika hasilnya didapatkan, sistem akan menampilkan hasilnya di halaman yang sama.

4.3.3 Diagram Kasus Pengguna

Diagram kasus pengguna berguna sebagai gambaran informasi interaksi yang dilakukan oleh pengguna dan sistem.



Gambar 4. 3 Diagram Kasus Pengguna

Diagram kasus pengguna dalam sistem yang dikembangkan oleh peneliti tergambar pada gambar 4.3. Terdapat 2 kasus pengguna yaitu untuk melihat temperatur terakhir yang masuk, dan melihat temperatur yang diprediksi dengan memasukkan jam yang akan diprediksi.

4.3.4 Skenario Kasus Pengguna

Skenarion kasus pengguna merupakan gambaran dari proses yang dilakukan oleh pengguna dalam menjalankan sistem. Berikut penjelasan dari skenario kasus pengguna yang telah dibuat peneliti.

Tabel 4. 3 Skenario Kasus Pengguna Melihat Temperatur Terakhir yang Masuk

Nama Kasus Pengguna	Melihat Temperatur Terakhir
Aktor	Pengguna

Deskripsi	Melihat temperatur terakhir yang masuk
Pre-Condition	-
Tahapan	<ol style="list-style-type: none"> 1. Pengguna mengakses halaman index 2. Halaman index terbuka dengan data terbaru yang masuk
Post-Condition	Berhasil masuk ke halaman index

Tabel 4.3 merupakan tabel kasus pengguna untuk melihat temperatur terakhir yang masuk ke dalam sistem.

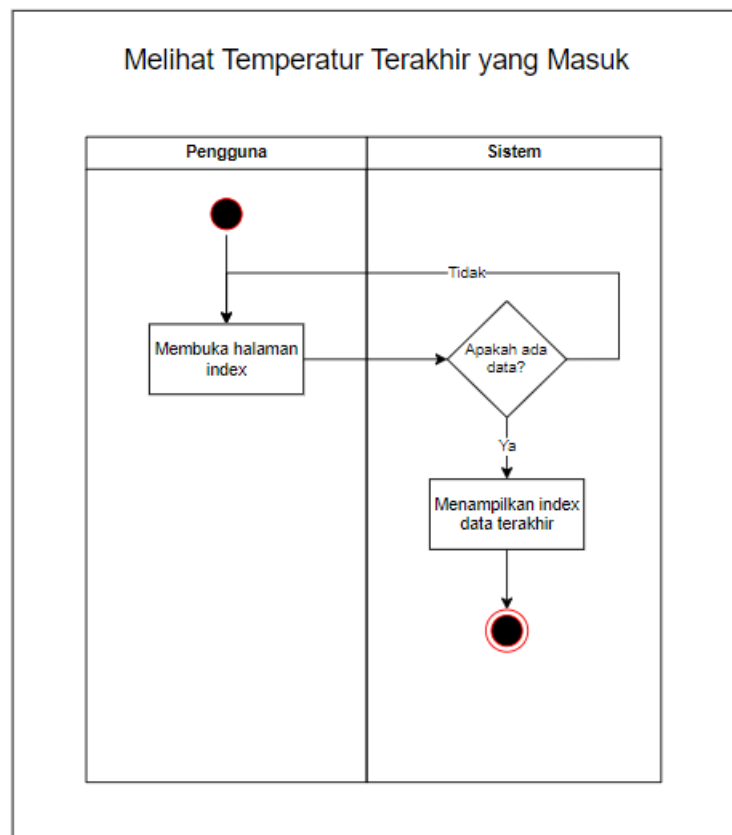
Tabel 4. 4 Skenario Kasus Pengguna Melihat Temperatur yang Diprediksi

Nama Kasus Pengguna	Melihat Temperatur Diprediksi
Aktor	Pengguna
Deskripsi	Melihat temperatur yang diprediksi
Pre-Condition	Memasukkan jam yang akan diprediksi
Tahapan	<ol style="list-style-type: none"> 1. Pengguna memasukkan jam yang akan diprediksi 2. Menunggu hasil prediksi 3. Hasil ditampilkan
Post-Condition	Hasil prediksi ditampilkan pada halaman prediksi

Tabel 4.4 merupakan tabel kasus pengguna untuk melihat temperatur yang diprediksi oleh sistem.

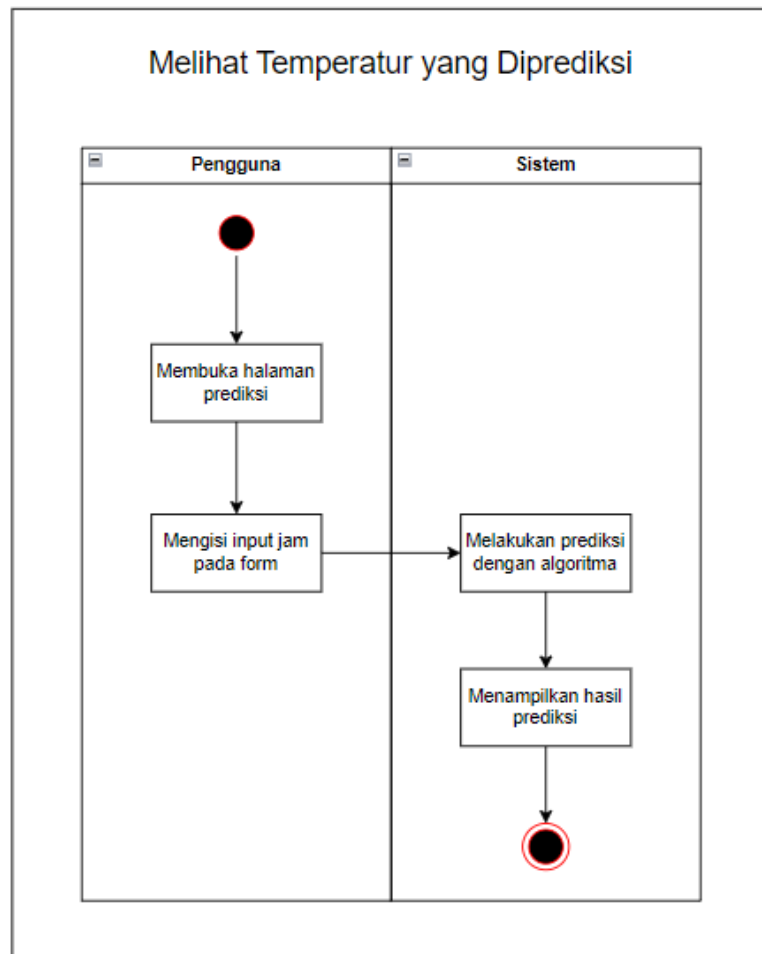
4.3.5 Diagram Aktivitas

Berdasarkan diagram kasus pengguna yang ada pada gambar 4.3, terdapat dua aktivitas yang bisa dilakukan oleh pengguna.



Gambar 4. 4 Diagram Aktivitas Melihat Temperatur Terakhir

Diagram aktivitas untuk melihat temperatur terakhir pada kabin mobil dimulai ketika pengguna berada pada halaman index. Ketika pengguna sudah membuka halaman index, sistem akan memeriksa pada basis data apakah ada data yang tersimpan pada basis data. Jika belum atau tidak ada data pada basis data, maka pengguna akan berada pada halaman index tanpa ada data temperatur terakhir. Jika data ditemukan oleh sistem pada basis data, maka data terbaru yang didapat oleh sistem akan ditampilkan pada halaman index sehingga pengguna bisa melihat data temperatur terakhir yang masuk pada kabin mobil.



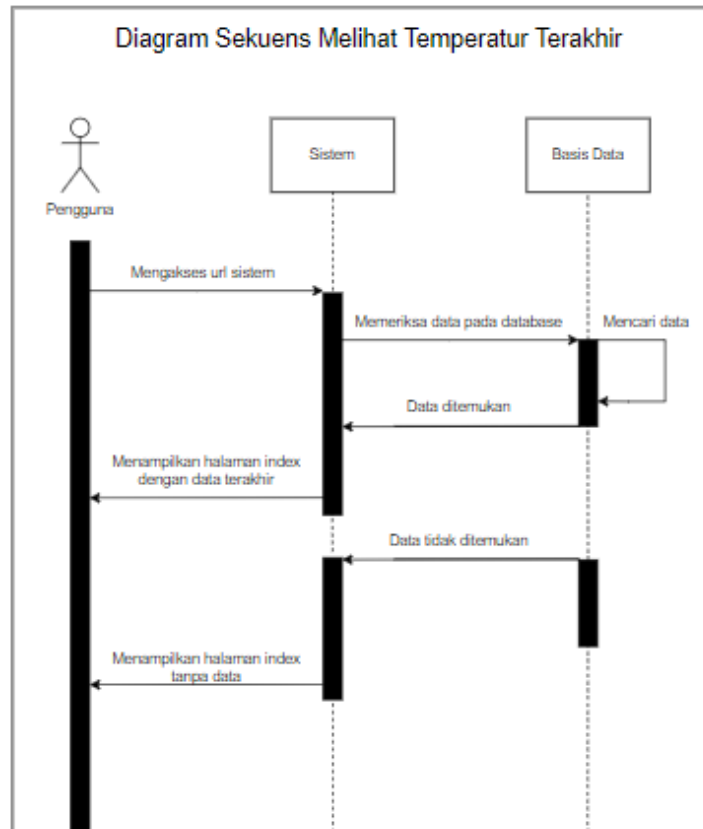
Gambar 4. 5 Diagram Aktivitas Melihat Temperatur Diprediksi

Diagram aktivitas untuk melihat hasil prediksi temperatur dimulai ketika pengguna membuka halaman prediksi dengan menekan tombol prediksi pada halaman index. Ketika pengguna membuka halaman prediksi, maka pengguna diminta untuk mengisi *input*. Setelah *input* dimasukkan dan pengguna menekan prediksi, maka sistem akan melakukan prediksi temperatur dengan menggunakan algoritma yang dirancang oleh peneliti. Hasil dari prediksi yang dilakukan sistem akan ditampilkan pada halaman prediksi.

4.3.6 Diagram Sekuens

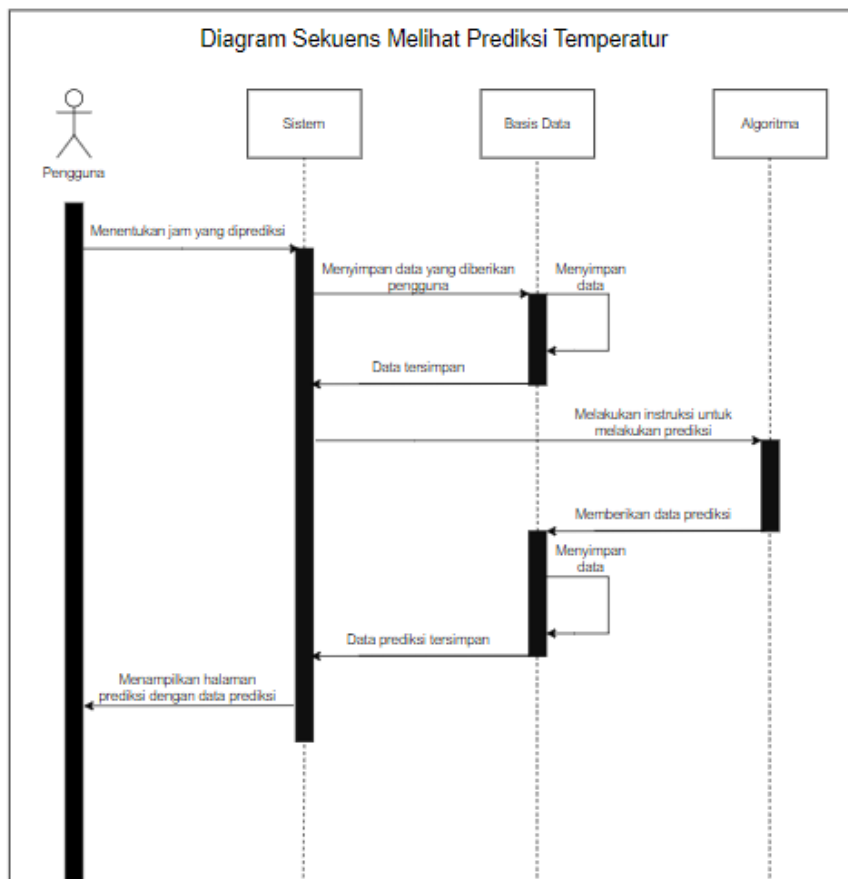
Diagram sekuens disusun oleh peneliti untuk menggambarkan skenario berjalannya sistem yang dikembangkan peneliti. Skenario diagram sekuens merupakan skenario respon yang dilakukan oleh sistem ketika melakukan

interaksi dengan pengguna. Berikut adalah gambaran diagram sekuens yang dirancang oleh peneliti.



Gambar 4. 6 Diagram Sekuens Melihat Temperatur Terakhir

Gambar 4.6 merupakan gambaran diagram sekuens pengguna untuk melihat temperatur terakhir pada sistem yang dikembangkan peneliti. Diagram bermula ketika pengguna mengakses url sistem untuk membuka halaman index. Sistem akan melakukan pemeriksaan data pada basis data terlebih dahulu sebelum menampilkan halaman index. Jika data ditemukan pada basis data, maka halaman index muncul dengan menampilkan data terbaru yang didapatkan oleh basis data. Apabila basis data tidak menemukan data apapun, maka sistem akan menampilkan halaman index tanpa adanya data.



Gambar 4. 7 Diagram Sekuens Melihat Prediksi Temperatur

Gambar 4.7 merupakan diagram sekuens pengguna untuk melihat hasil prediksi temperatur yang dilakukan. Diagram bermula ketika pengguna telah menentukan jam yang akan diprediksi. Data yang diberikan oleh pengguna akan diteruskan sistem kepada basis data untuk menyimpan data jam yang ditentukan. Setelah melakukan penyimpanan, sistem akan melakukan instruksi kepada algoritma untuk melakukan prediksi berdasarkan jam yang telah disimpan. Algoritma akan mengirimkan data prediksi yang telah dilakukan kepada basis data untuk disimpan. Setelah data tersimpan, sistem akan meneruskan data yang tersimpan tersebut kepada halaman prediksi untuk ditampilkan hasilnya.

4.3.7 Perancangan Basis Data

Basis data memainkan peran penting dalam sistem yang dikembangkan oleh peneliti. Tanpa adanya basis data maka sistem tidak akan berjalan seperti

yang seharusnya. Perancangan basis data bertujuan untuk menentukan data apa saja yang perlu untuk disimpan dan digunakan oleh sistem.

Tabel 4. 5 Basis Data Tabel Datasets

No.	Field	Type	Length	Keterangan
1.	id	INT	11	Id data yang masuk ke basis data
2.	temperature	FLOAT		Data temperatur
3.	co2	FLOAT		Data co2
4.	created_at	TIMESTAMP		Waktu data masuk
5.	updated_at	TIMESTAMP		Waktu data diubah

Tabel 4.5 merupakan tabel dari basis data tabel datasets yang dirancang peneliti. Tabel ini berfungsi untuk menampung data yang diberikan oleh sistem tertanam yang telah dibangun oleh peneliti dan juga berfungsi untuk digunakan oleh algoritma *random forest* untuk melakukan prediksi suhu. Sistem akan melakukan pemeriksaan terlebih dahulu pada tabel datasets ini untuk memeriksa ada atau tidaknya data pada tabel ini untuk menampilkannya kepada pengguna di halaman index. Data yang ditampilkan adalah data terbaru yang masuk ke dalam tabel datasets.

Tabel 4. 6 Basis Data Tabel Input_user

No.	Field	Type	Length	Keterangan
1.	id	INT	11	Id data yang masuk ke basis data
2.	input_user	INT	11	Data yang dimasukkan pengguna
3.	created_at	TIMESTAMP		Waktu data masuk
4.	updated_at	TIMESTAMP		Waktu data diubah

Tabel 4.6 merupakan tabel dari basis data tabel input_user yang dirancang peneliti untuk menyimpan data yang dimasukkan oleh pengguna. Data yang dimasukkan pengguna berupa integer yang menunjukkan angka jam yang dimasukkan pengguna untuk melakukan prediksi. Tabel ini akan digunakan oleh

algoritma *random forest* untuk menentukan jam yang akan diprediksi sesuai dengan keinginan pengguna.

Tabel 4. 7 Basis Data Tabel Settings

No.	Field	Type	Length	Keterangan
1.	id	INT	11	Id data yang masuk ke basis data
2.	input_id	INT	11	Data yang dimasukkan pengguna
3.	need_run	TinyINT	1	Data instruksi algoritma
4.	periode	INT	11	Data yang dimasukkan pengguna
5.	created_at	TIMESTAMP		Waktu data masuk
6.	updated_at	TIMESTAMP		Waktu data diubah

Tabel 4.7 merupakan basis data tabel settings yang berfungsi sebagai instruksi kepada algoritma *random forest* untuk melakukan prediksi atau tidak. Kolom *need_run* berisikan angka 1 atau 0 yang berperan untuk memerintahkan algoritma *random forest* melakukan prediksi atau tidak. Kolom *periode* merupakan kolom yang berfungsi sebagai instruksi jam yang ditentukan oleh pengguna untuk melakukan prediksi. Kolom ini berkaitan dengan kolom *input_user* yang ada pada tabel *input_user* karena berisikan data yang diberikan pengguna kepada sistem. Data yang ada pada tabel settings akan selalu terbaru ketika pengguna mengirimkan data baru kepada sistem.

Tabel 4. 8 Tabel Basis Data Tabel Prediction

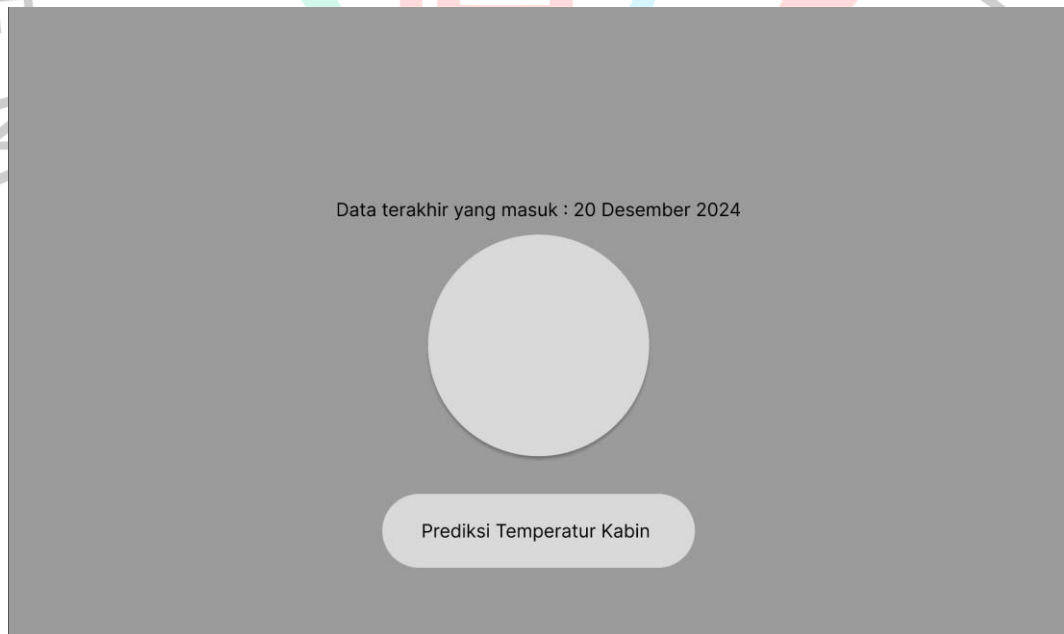
No.	Field	Type	Length	Keterangan
1.	id	INT	11	Id data yang masuk
2.	time_start	DATETIME		Waktu terakhir data masuk
3.	periode	INT	11	Data jam yang akan diprediksi
4.	prediction	FLOAT		Data prediksi <i>random forest</i>
5.	time	DATETIME		Data waktu yang diprediksi

6.	created_at	TIMESTAMP	Waktu data masuk
7.	updated_at	TIMESTAMP	Waktu data diubah

Tabel 4.8 merupakan tabel basis data tabel prediction yang berfungsi sebagai penyimpanan hasil prediksi yang dilakukan algoritma *random forest*. Hasil prediksi disimpan pada tabel ini dan nantinya hasilnya akan ditampilkan pada halaman prediksi.

4.3.8 Perancangan Antarmuka

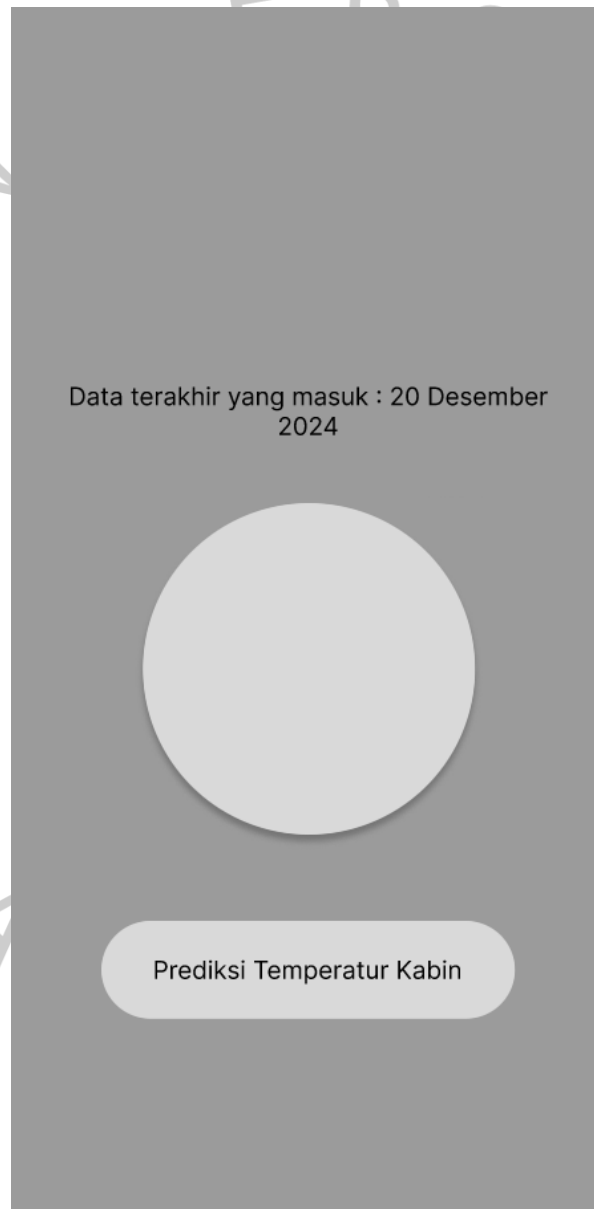
Antarmuka diperlukan dalam sistem yang dikembangkan peneliti sebagai pembantu pengguna untuk menggunakan sistem. Desain antarmuka yang ramah pengguna penting untuk memastikan pengalaman pengguna yang optimal. Antarmuka yang dirancang tidak hanya memudahkan navigasi, tetapi juga memastikan bahwa pengguna dapat mengakses semua fitur dan fungsi dengan mudah dan efisien. Berikut adalah beberapa gambar tampilan antarmuka yang telah dirancang peneliti:



Gambar 4. 8 Tampilan Mockup Antarmuka Halaman Index

Tampilan pada gambar 4.9 merupakan *mockup* antarmuka halaman index yang dikembangkan peneliti. Peneliti menggunakan tampilan web, tetapi web yang dirancang peneliti bisa diubah menjadi sebuah aplikasi *desktop* ataupun

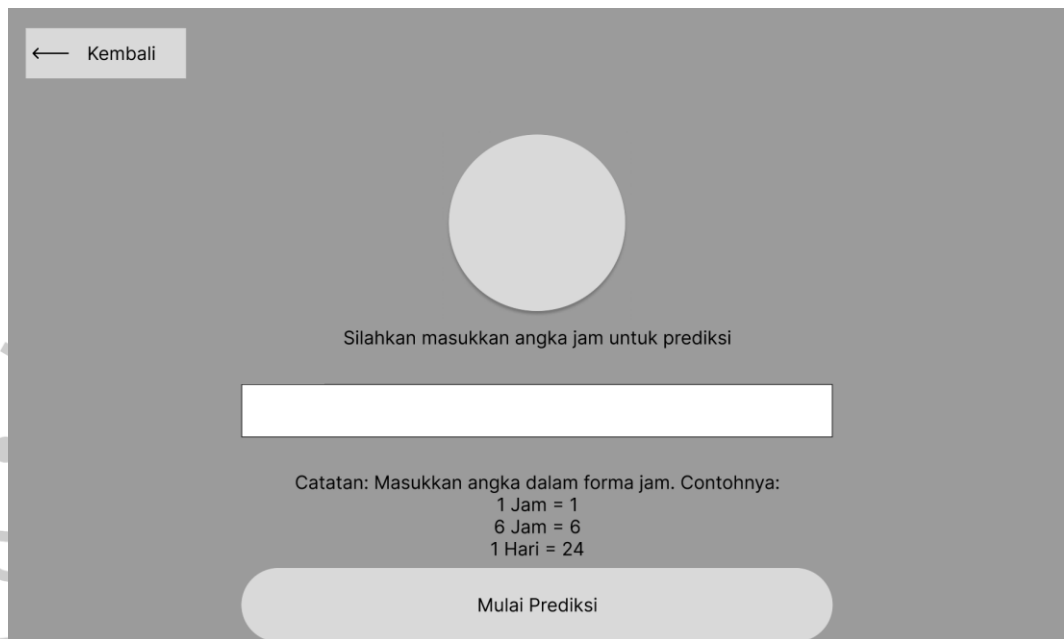
mobile karena peneliti menggunakan teknologi *Progressive Web Application* (PWA). Dengan PWA antarmuka web ini dapat berfungsi seperti aplikasi pada umumnya sehingga dapat memberikan pengalaman yang mudah digunakan dan responsif.



Gambar 4. 9 Tampilan Mockup Antarmuka Seluler Halaman Index

Peneliti menggunakan teknologi PWA untuk mempermudah pengguna dalam menggunakan sistem sehingga web yang dikembangkan bisa menjadi sebuah aplikasi seluler seperti yang tergambarkan pada gambar 4.10. Selain untuk

mempermudah pengguna, tujuan dibuatnya web dengan teknologi PWA adalah untuk membuat notifikasi layaknya aplikasi seluler umumnya sehingga membuat pengguna lebih waspada terhadap notifikasi perubahan suhu kabin mobil.



Gambar 4. 10 Tampilan Mockup Antarmuka Halaman Prediksi

Antarmuka halaman prediksi yang dikembangkan peneliti tergambar pada gambar 4.11. *Mockup* antarmuka halaman prediksi dirancang peneliti sedemikian rupa untuk memudahkan pengguna untuk melakukan prediksi. Pada halaman ini, pengguna dapat memasukkan angka jam yang diinginkan untuk prediksi dengan format yang jelas dan sederhana. Terdapat instruksi yang jelas mengenai format angka yang harus dimasukkan sehingga pengguna tidak salah memasukkan format angka yang diperlukan.

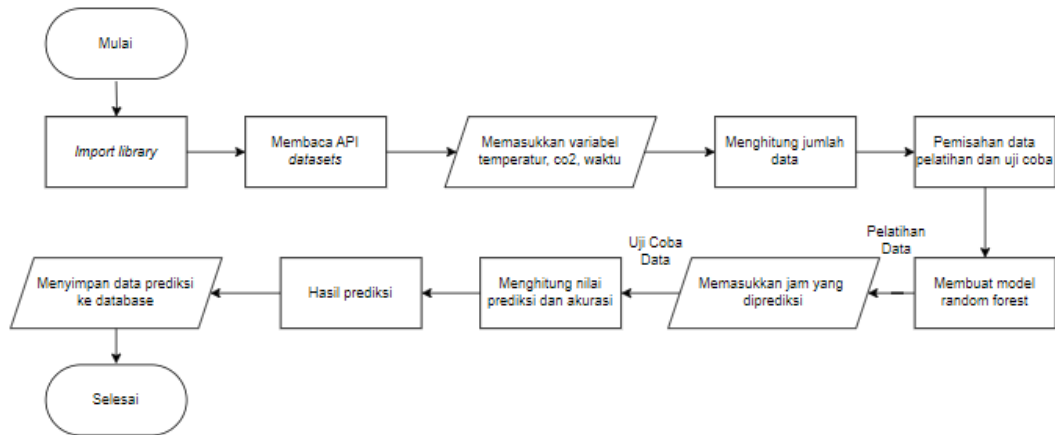


Gambar 4. 11 Tampilan Mockup Antarmuka Seluler Halaman Prediksi

Gambar 4.12 merupakan gambar dari tampilan *mockup* antarmuka halaman prediksi ketika sudah menjadi aplikasi seluler. Tampilan dibuat dengan sederhana dan efisien sehingga pengguna tidak kesulitan dalam menggunakan aplikasi.

4.3.9 Perancangan Algoritma *Random Forest*

Rancangan algoritma *Random Forest* dibuat untuk menunjukkan proses bekerja algoritma dari melakukan pengambilan data, lalu melakukan pelatihan data, melakukan prediksi, dan menyimpan data pada basis data.



Gambar 4. 12 Perancangan Algoritma Random Forest

Dibawah ini merupakan penjabaran dari setiap tahap pada perancangan algoritma *random forest* untuk memperjelas langkahnya:

1) *Import library*

Tahap ini merupakan tahap untuk mengimpor pustaka yang dibutuhkan dalam sistem.

2) *Membaca API datasets*

Tahap ini melibatkan pengambilan data dari API yang menyediakan dataset.

3) Memasukkan variabel temperatur, co2, dan waktu

Setelah data diambil, tiga variabel penting yang akan digunakan untuk melakukan pelatihan data akan diekstraksi dan disiapkan untuk pelatihan.

4) Menghitung jumlah data

Jumlah total data yang dikumpulkan akan dihitung terlebih dahulu untuk meyakinkan kumpulan data yang digunakan sesuai dengan data yang diambil dari API.

5) Pemisahan data pelatihan dan data uji coba

Data kemudian dipisah untuk dibagi data pelatihan dan data uji coba gunanya demi menentukan model terbaik yang digunakan dalam melakukan prediksi.

6) Membuat model *random forest*

Pada tahap ini, peneliti mengembangkan dan melatih model Random Forest menggunakan data pelatihan. Random Forest merupakan algoritma pembelajaran ensemble yang memanfaatkan beberapa pohon keputusan untuk meningkatkan akurasi prediksi.

7) Memasukkan jam yang akan diprediksi

Model yang telah dilatih kemudian digunakan untuk memprediksi variabel jam. Prediksi ini dilakukan terhadap data test yang telah disiapkan sebelumnya.

8) Menghitung nilai prediksi dan akurasi

Hasil prediksi dibandingkan dengan nilai sebenarnya untuk menghitung akurasi model. Akurasi ini menunjukkan seberapa baik model dalam memprediksi data.

9) Hasil prediksi

Hasil prediksi yang telah diperoleh dari model kemudian disiapkan untuk langkah selanjutnya.

10) Menyimpan data prediksi ke basis data

Prediksi yang telah dihasilkan disimpan ke dalam basis data untuk keperluan aplikasi.

4.4 Rancangan Pengujian

Untuk memastikan sistem yang dikembangkan berfungsi optimal dan sesuai dengan hasil yang diharapkan, peneliti merancang serangkaian pengujian. Mereka menerapkan metode pengujian white box dan black box sebagai standar pengujian. Pengujian white box berfokus pada struktur internal dan logika kode yang dibuat, sementara pengujian black box mengevaluasi fungsionalitas sistem tanpa memperhatikan struktur internal. Dengan metode pengujian yang terstruktur

ini, diharapkan sistem yang dikembangkan dapat memenuhi standar kualitas, dan siap diimplementasikan.

4.4.1 Rancangan Pengujian *White Box*

Perancangan pengujian *white box* bertujuan untuk menguji kode program agar sistem berjalan dengan baik. Rancangan pengujian *white box* akan ditampilkan seperti pada tabel berikut:

Tabel 4.9 Pengujian *White Box*

No.	Kode Sumber	Hasil yang Hasil diharapkan
1.	<pre>\$(document).ready(function() { reloadData(); setInterval(reloadData, 5000); // Adjust the interval time as needed (5000 ms = 5 seconds) });</pre>	<p>Melakukan pemanggilan fungsi reloadData() secara otomatis</p>
2.	<pre>function reloadData() { let notificationHighSent = false; let notificationMidSent = false; let notificationLowSent = false; \$.ajax({ url: 'api/received/latest/data', method: 'GET', success: function(response) { var date = new Date(response.created_at); var formattedDate = date.getFullYear() + '-' + ('0' + (date.getMonth() + 1)).slice(-2) + '-' + ('0' + date.getDate()).slice(-2) + '-' + ('0' + date.getHours()).slice(-2) + ':' +</pre>	<p>Melakukan refresh pada data terbaru yang didapatkan dan menyiapkan notifikasi untuk fungsi sendNotification</p>

```
        ('0' + date.getMinutes()).slice(-2)
    + ':' +
        ('0' + date.getSeconds()).slice(-2);
```

```
    $('#timeReceived').text('Data
terakhir yang diterima : ' + formattedDate);
```

```
    $('#tempReceived').text(response.temperature +
    '°C');
```

```
    if (response.temperature >= 40) {
```

```
        $('#tempReceived').css('background-color',
        'red');
```

```
        if (!notificationHighSent) {
```

```
            sendNotification("Suhu mobil
sudah mencapai 40C", "Segera pindahkan
barang yang menggunakan baterai lithium");
```

```
            notificationHighSent = true;
```

```
            notificationMidSent = false;
```

```
            notificationLowSent = false; //
```

```
            Reset notifikasi suhu rendah
```

```
        }
```

```
    } else if (response.temperature >=
    35) {
```

```
        $('#tempReceived').css('background-color',
        '#ffA500');
```

```
        if (!notificationHighSent) {
```

```
            sendNotification("Suhu mobil
sudah mencapai 35C", "Waspada, suhu sedikit
berbahaya terhadap barang yang menggunakan
baterai lithium");
```

```
            notificationHighSent = true;
```

```
            notificationMidSent = true;
```

```

        notificationLowSent = false; //
    Reset notifikasi suhu rendah
    }
    } else {

```

```

    $('#tempReceived').css('background-color',
    '#007bff');
        if (!notificationLowSent &&
    notificationHighSent) {
            sendNotification("Suhu mobil
    sudah turun di bawah 35C", "Suhu sudah normal
    kembali.");
            notificationLowSent = true;
            notificationMidSent = false;
            notificationHighSent = false; //

```

```

    Reset notifikasi suhu tinggi
    }
    }
        console.log(response);
    },
    error: function(xhr, status, error) {
        alert("Terjadi kesalahan, segera
    lakukan refresh halaman ini.");
    }
    });
}

```

- function sendNotification(title, body) { Memberikan notifikasi kepada pengguna dengan pesan

```

        if ('serviceWorker' in navigator &&
    'Notification' in window) {

    navigator.serviceWorker.ready.then(function(reg
    istration) {
        registration.showNotification(title, {
            body: body,

```

```
        icon: '/temperature_car.png', //
```

Gantilah path ini dengan path ke ikon notifikasi

Anda

```
        tag: 'temperature-notification'
```

```
    });
```

```
  });
```

```
  }
```

```
}
```

4. `if ('Notification' in window && Notification.permission !== 'granted') {` Meminta pengguna untuk memperbolehkan sistem memberikan notifikasi
- ```
 Notification.requestPermission().then(function(permission) {
 if (permission === 'granted') {
 console.log('Notification permission granted.');
```

```
 }
```

```
 });
```

```
}
```



```

$('#loadingtemp').text(response.message);
 setTimeout(function() {
 loadData();
 }, 10000);
},
error: function(xhr, status, error) {
 alert('Terjadi kesalahan, coba
kembali.');
```

```

$('#inputUser').removeAttr('disabled');
```

```

$('#submitButton').removeAttr('disabled');
```

```
 }
```

```
});
```

```
}
```

6.

```
function loadData() {
```

```
 $.ajax({
```

```
 url: 'api/prediction/latest',
```

```
 method: 'GET',
```

```
 success: function(response) {
```

```
 $('#loadingtemp').removeClass('d-
block').addClass('d-none');
```

```

$('#inputUser').removeAttr('disabled');
```

```

$('#submitButton').removeAttr('disabled');
```

```

 $('#timePred').text('Prediksi pada ' +
response.time);
```

```

 $('#temperature').text(response.prediction
+
°C');
```

Melakukan pengambilan data prediksi dan membuat permissalan ketika data melebihi 35 dan 40

```
 if (response.prediction >= 40) {
 $('#noteLabel').text('Suhu kabin
mobil berbahaya bagi barang yang
menggunakan baterai lithium pada jam
tersebut!');

 $('#temperature').css('background-color', 'red');
 } else if (response.prediction >= 35)
 {
 $('#noteLabel').text('Suhu kabin
mobil mendekati suhu berbahaya bagi barang
yang menggunakan baterai lithium pada jam
tersebut.');
```

UNIVERSITAS  
PEMBANGUNAN JAYA

```
 $('#temperature').css('background-color',
'ffa500');
 } else {
 $('#temperature').css('background-color',
'#007bff');
 }
 },
 error: function(xhr, status, error) {
 alert('Gagal untuk mendapatkan
data prediksi.');
```

BANGUNAN

```
 $('#loadingtemp').removeClass('d-
block').addClass('d-none');

 $('#inputUser').removeAttr('disabled');

 $('#submitButton').removeAttr('disabled');
 }
});
}
```

---

#### 4.4.2 Rancangan Pengujian *Black Box*

Perancangan pengujian *black box* bertujuan untuk menguji fitur yang ada pada sistem. Pengujian ini melibatkan web dan juga algoritma untuk memastikan keberhasilan sistem. Rancangan pengujian *black box* akan ditampilkan pada tabel berikut:

Tabel 4.10 Pengujian *Black Box*

| No. | Skenario Pengujian                                              | Hasil yang diharapkan                                               | Hasil |
|-----|-----------------------------------------------------------------|---------------------------------------------------------------------|-------|
| 1.  | Pengguna menekan tombol prediksi pada halaman index             | Membuka halaman prediksi                                            |       |
| 2.  | Pengguna mengisi jam dan menekan prediksi pada halaman prediksi | Sistem melakukan prediksi dan menampilkan hasilnya                  |       |
| 3.  | Data suhu terakhir yang didapatkan melebihi angka 35            | Sistem memberikan notifikasi kepada pengguna bahwa suhu mencapai 35 |       |
| 4.  | Data suhu terakhir yang didapatkan melebihi angka 40            | Sistem memberikan notifikasi kepada pengguna bahwa suhu mencapai 40 |       |