

BAB III

METODOLOGI PENELITIAN

3.1 Objek dan Tempat Penelitian

Objek penelitian dalam skripsi ini adalah respon fisiologis dan mekanika gerak tubuh member gym, yang meliputi parameter detak jantung (BPM), saturasi oksigen (SpO₂), suhu tubuh, dan durasi lama set (*Time Under Tension*) yang diukur secara nyata pada set terakhir dari variasi latihan *Bicep Curl* atau *Tricep Pushdown*. Pengondisian ini bertujuan untuk menangkap titik jenuh kapabilitas kontraksi otot subjek akibat adanya akumulasi kelelahan metabolik sistemik dari keseluruhan sesi olahraga. Data tersebut diamati untuk mendeteksi ambang batas kelelahan (*fatigue detection*) dan memberikan *biofeedback* guna mencegah risiko cedera otot akibat *ego lifting*.

Penelitian ini dilaksanakan secara terpusat di Bomber Gym yang berlokasi di Margamekar, Kec. Pangalengan, Kabupaten Bandung, Jawa Barat 40378. Pemilihan lokasi ini dilakukan secara sengaja (*purposive*) karena merupakan tempat latihan rutin peneliti yang memiliki ekosistem pengguna aktif latihan beban (*resistance training*). Selain itu, adanya dukungan dari komunitas rekan gym di lokasi tersebut memberikan keuntungan strategis bagi peneliti dalam memudahkan proses perolehan data biometrik serta data gerakan yang valid untuk memenuhi kebutuhan pengujian perangkat secara akurat.

3.2 Metode Penelitian

Metode penelitian ini dirancang sebagai landasan sistematis dalam mengembangkan produk teknologi yang menggabungkan aspek perangkat keras dan perangkat lunak secara terintegrasi.

3.2.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam tugas akhir ini adalah Penelitian Eksperimental Berbasis Pengembangan Prototipe (*Pure Prototyping*). Pendekatan ini dipilih karena fokus utama penelitian adalah merancang,

merealisasikan, dan menguji coba sebuah versi awal (*prototipe*) dari perangkat fungsional berupa *wearable armband* berbasis IoT. Penelitian ini menitikberatkan pada validasi fungsionalitas teknis serta akurasi komputasi algoritma pada perangkat untuk mendeteksi kelelahan otot, tanpa memperluas cakupan ke tahap produksi massal atau komersialisasi produk.

1.2.2 Metode Pengembangan Sistem

Metode Prototype diterapkan pada penelitian ini karena sifat pengembangan wearable IoT yang menuntut siklus uji-coba-perbaiki berulang antara hardware dan software. Pendekatan ini memungkinkan kekurangan teknis seperti kesalahan kalibrasi sensor atau ketidaksesuaian payload BLE teridentifikasi sejak tahap awal, alih-alih setelah sistem final dioperasikan (Andini et al., 2023).

Tahapan pengembangan sistem dalam penelitian ini mengikuti alur adaptasi metode *Prototype* 5 langkah, yang dijabarkan secara aplikatif sebagai berikut:

1. Pengumpulan Kebutuhan (*Requirements Gathering*) Tahap identifikasi mendalam terhadap kebutuhan sistem Fit-Tick, meliputi struktur data biometrik (BPM, SpO2, suhu), alur transmisi data via BLE, serta parameter input-output pada *dashboard monitoring Personal Trainer*.
2. Membangun Prototipe (*Quick Design & Build*) Tahap implementasi teknis awal yang mencakup perancangan kelistrikan *hardware*, pengkodean *firmware* pada mikrokontroler ESP32, serta pembangunan *layout dashboard* menggunakan *framework* Laravel dan basis data MySQL.
3. Menguji Prototipe (*Prototype Testing*) Proses pengujian fungsionalitas internal (*Alpha Testing*) untuk menemukan dan memperbaiki masalah teknis awal pada sirkuit elektronika, *noise*, kestabilan pengiriman *payload* data, serta latensi transmisi Web Bluetooth API.

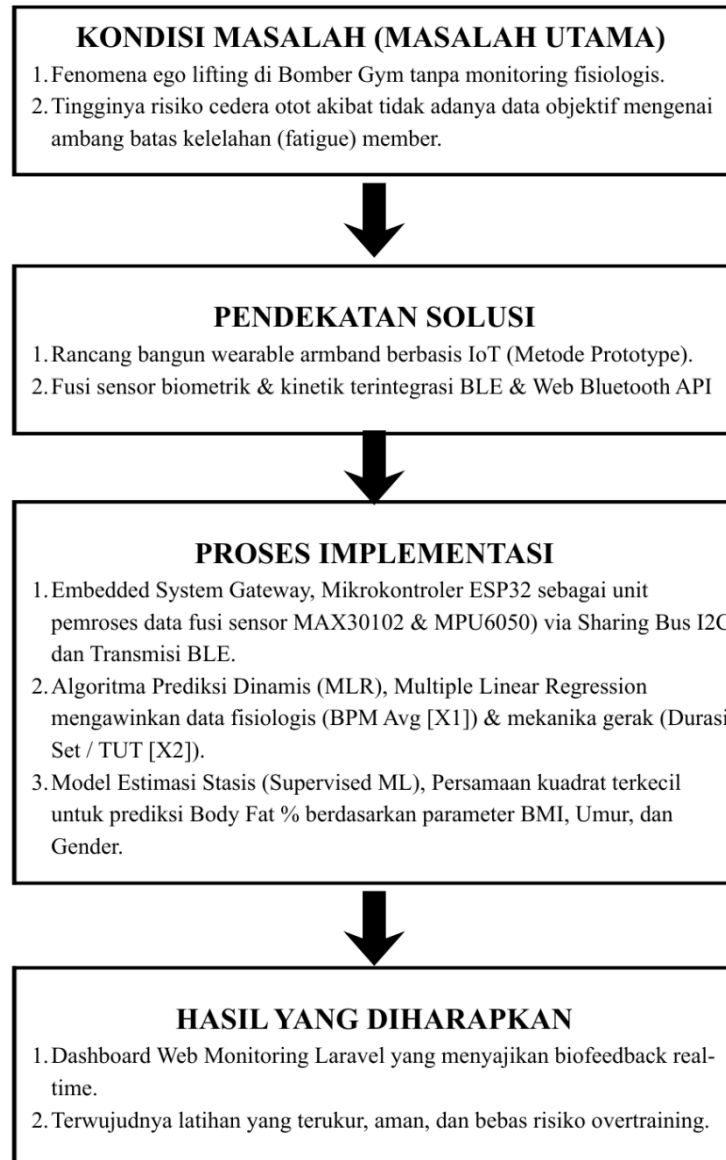
4. Evaluasi Prototipe (*Evaluation*) Evaluasi bersama pengguna di lingkungan *gym* (*Beta Testing*) untuk menguji ketepatan deteksi repetisi dan anomali data (seperti pembacaan detak jantung). Hasil evaluasi digunakan sebagai dasar rekayasa ulang algoritma filter komputasi pada *backend* Laravel.
5. Finalisasi Prototipe Teruji (*Finalizing Prototype*). Langkah terakhir untuk mengintegrasikan seluruh perbaikan secara permanen, termasuk mengunci rumus komputasi *Multiple Linear Regression* (MLR) dan fitur notasi audio (beep.mp3) ke dalam *source code*. Tahap ini berfokus untuk memastikan *wearable armband* dan *dashboard Live Monitor* Fit-Tick berada dalam kondisi paling stabil dan siap digunakan sebagai instrumen pengambilan data akhir penelitian.

3.3 Kerangka Berpikir

Kerangka berpikir dalam penelitian ini disusun untuk memberikan gambaran sistematis mengenai alur penalaran ilmiah dan pemecahan masalah yang dihadapi oleh member gym di Bomber Gym. Proses penelitian ini dirancang secara terstruktur, diawali dengan identifikasi masalah riil di lapangan, dilanjutkan dengan pendekatan solusi teknologis, perancangan model komputasi, hingga tahap validasi hasil akhir sistem.

Alur kerangka berpikir penelitian ini secara utuh direpresentasikan melalui bagan logis pada Gambar 3.1 berikut:

KERANGKA BERFIKIR



Gambar 3. 1 Bagan Kerangka Berpikir Sistem Monitoring

Berdasarkan alur pada Gambar 3.1, deskripsi operasional dari masing-masing tahapan kerangka berpikir dapat dijelaskan sebagai berikut:

1. Kondisi Masalah (Masalah Utama), Adanya fenomena *ego lifting* di Bomber Gym, di mana member sering memaksakan volume beban latihan

secara berlebihan tanpa memantau respon fisiologis internal tubuh. Hal ini diperparah oleh ketiadaan indikator objektif yang mampu mendeteksi batas kemampuan otot secara *real-time*, sehingga meningkatkan risiko cedera akut atau *overtraining*.

2. Pendekatan Solusi, Mengembangkan sebuah perangkat perangkat keras *wearable armband* berbasis IoT yang mengintegrasikan sensor biometrik (detak jantung dan Saturasi oksigen) serta sensor inersia kinetik (akselerometer). Metodologi pengembangan sistem menggunakan pendekatan metode *Prototype* agar siklus iterasi dari model awal hingga sistem final menjadi lebih efisien dan adaptif.
3. Proses Implementasi:
 - a. Implementasi Perangkat Tertanam (*Embedded System Gateway*): Fokus pada implementasi sirkuit elektronik menggunakan mikrokontroler ESP32 sebagai unit pemroses utama sekaligus *gateway* data lokal. ESP32 melakukan fusi data (*sensor fusion*) terhadap parameter dari sensor MAX30102 dan MPU6050 pada *armband*. Data mentah diekstraksi di memori lokal mikrokontroler sebelum ditransmisikan secara nirkabel menggunakan protokol *Bluetooth Low Energy* (BLE) demi efisiensi daya dan latensi minimal.
 - b. Penerapan Algoritma Regresi Linear Berganda (*Multiple Linear Regression*): Proses pengolahan data server menerapkan algoritma Multiple Linear Regression (MLR) untuk mendeteksi ambang batas indeks kelelahan (*Fatigue Index*) secara objektif. Algoritma *supervised learning* ini menganalisis hubungan linear simultan antara variabel respons (persentase kelelahan) dengan dua variabel prediktor numerik, yaitu fluktuasi rata-rata detak jantung (BPM Avg X_1) yang dievaluasi terhadap ambang batas Maximal

Heart Rate (MHR) berbasis rumus Dr. Samuel M. Fox ($MHR = 220 - \text{Usia}$), dan durasi lama set atau Time Under Tension (X_2) di akhir sesi latihan beban.

- c. Penerapan Model Estimasi Komposisi Tubuh Berbasis *Supervised Machine Learning*: Sistem mengintegrasikan model persamaan prediktif yang diadaptasi dari riset Xu et al. (2023) untuk mengestimasi persentase lemak tubuh (*Body Fat Percentage*) pada *dashboard monitoring Personal Trainer*. Pendekatan *ordinary least squares* ini bekerja dengan mengorelasikan data antropometri *Body Mass Index* (BMI) dengan variabel sosio-demografis (usia dan jenis kelamin) guna memberikan profil kebugaran dasar dalam mengevaluasi batas ketahanan fisik pengguna.
4. Hasil yang Diharapkan, Terciptanya sistem monitoring terpadu melalui *dashboard* web Laravel yang mampu menangkap transmisi data via *Web Bluetooth API* guna memberikan umpan balik klinis (*biofeedback*) secara *real-time* kepada member dan Personal Trainer, sehingga aktivitas latihan isolasi lengan bawah (*Bicep Curl* dan *Tricep Pushdown*) menjadi lebih aman, efektif, dan terukur untuk mendeteksi kelelahan.

3.4 Analisis Sistem

Analisis sistem merupakan tahap krusial untuk menentukan spesifikasi teknis dan kebutuhan operasional yang harus dipenuhi agar sistem *armband* IoT dapat berjalan sesuai dengan tujuan penelitian.

3.4.1 Alat dan Bahan

Komponen yang digunakan dalam penelitian ini dibagi menjadi dua kategori utama, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai berikut:

A. Perangkat Keras (*Hardware*)

1. ESP32-DevKit-V1: Berfungsi sebagai mikrokontroler utama untuk memproses data sensor dan mengirimkannya melalui koneksi Bluetooth Low Energy (BLE).
2. Sensor MAX30102: Digunakan untuk memperoleh data biometrik berupa detak jantung (BPM) dan saturasi oksigen (SpO₂) secara *real-time*.
3. Sensor MPU6050: Berfungsi sebagai akselerometer dan giroskop untuk memantau orientasi gerakan serta menghitung repetisi latihan.
4. Sensor DS18B20: Digunakan untuk mengukur suhu kulit pengguna sebagai indikator respon fisiologis tubuh.
5. Layar OLED 0.96 inci: Berfungsi sebagai antarmuka lokal untuk menampilkan data biometrik langsung pada perangkat *armband*.
6. Modul TP4056 & Baterai Li-ion: Digunakan sebagai sistem manajemen daya agar perangkat bersifat *portable*.

B. Perangkat Lunak (*Software*)

1. Laravel 12 Digunakan sebagai *framework* utama dalam membangun *dashboard* monitoring dan manajemen data member.
2. MySQL Berfungsi sebagai sistem manajemen basis data untuk menyimpan log latihan dan profil pengguna.
3. Arduino IDE / VS Code: Digunakan sebagai lingkungan pengembangan untuk mengkodekan logika pada mikrokontroler ESP32.
4. Figma Digunakan untuk merancang antarmuka pengguna (*User Interface*) sebelum diimplementasikan ke dalam kode.

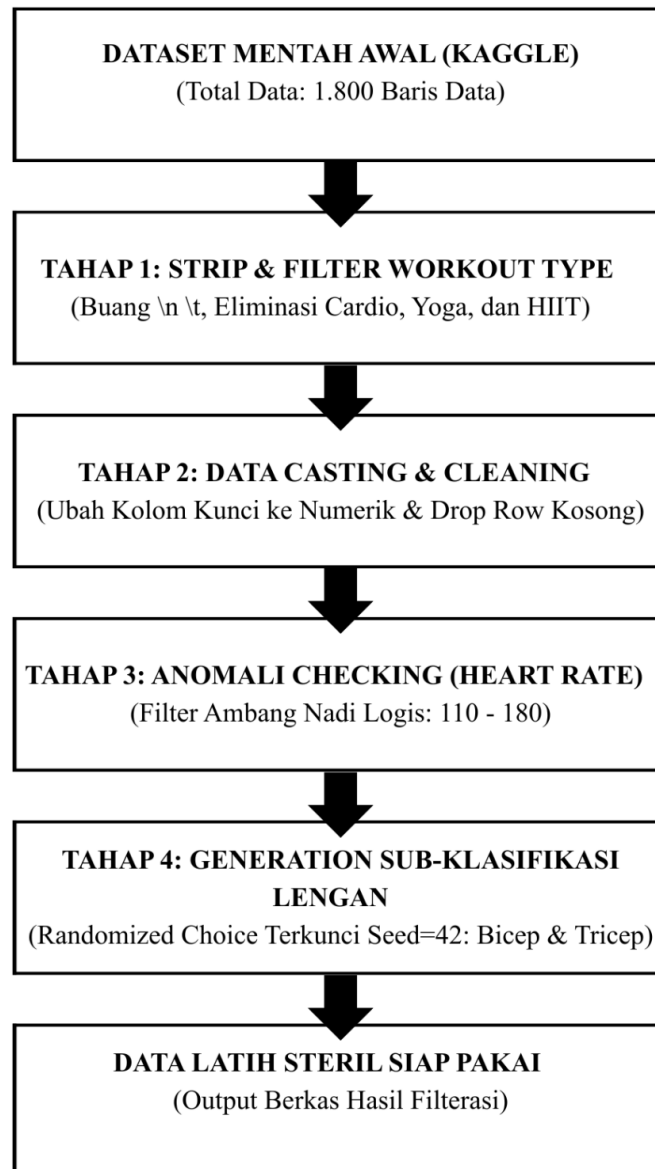
3.4.2 Analisis Kebutuhan Dataset Latih

Guna mendukung fungsionalitas modul komputasi statistik longitudinal, pemodelan regresi, dan pemetaan tren performa fisik pada aplikasi Fit-Tick,

sistem memerlukan ketersediaan data latih (*training data*) awal yang valid dan reliabel. Pemenuhan kebutuhan data historis ini dialihkan memanfaatkan data sekunder berupa *Fitness Tracker Dataset* yang diperoleh dari platform repositori global Kaggle (Choudhary, 2024). Berkas mentah (*raw dataset*) tersebut aslinya berisi 1.800 baris rekaman aktivitas fisik dari berbagai member gym heterogen. Namun, berdasarkan analisis kebutuhan data, berkas tersebut masih berada dalam kondisi kotor (*noisy data*) karena cakupan jenis aktivitas olahrahanya terlalu universal, seperti mencantumkan data olahraga *Cardio*, *Yoga*, dan *HIIT*, serta terdapat gangguan spasi liar (*whitespace noise*) pada penulisan string seperti karakter `\nStrength` dan `\tCardio`.

Aktivitas olahraga non-beban tersebut secara biomekanis tidak memicu kontraksi pekat (*time under tension*) pada area otot lengan atas, sehingga tidak sinkron dengan tujuan pemantauan perangkat keras *wearable armband* yang berfokus pada analisis otot Biceps Curl dan Triceps Pushdown. Oleh karena itu, dalam analisis kebutuhan sistem ini, dirancang sebuah pipa pemrosesan data (*data pipeline preprocessing*) menggunakan bahasa pemrograman Python dan pustaka *Pandas/NumPy* untuk memurnikan data mentah tersebut. Proses ini bertugas untuk mengeliminasi anomali, melakukan standardisasi tipe data, dan mengisolasi sampel data secara spesifik ke dalam dua kebutuhan variasi latihan otot lengan, yaitu: Biceps Curl dan Triceps Pushdown.

Alur tahapan pemurnian dari kondisi mentah hingga menjadi dataset steril siap pakai diilustrasikan secara runut pada Gambar 3.2.

PREPROCECING PEMURNIAN DATASET

Gambar 3. 2 Alur Pipeline Preprocessing Pemurnian Dataset

```
[INFO] Memproses analisis dataset mentah...
ANALISIS DISTRIBUSI SELURUH DATASET SEBELUM DISARING
=====
Total Keseluruhan Data Mentah: 1800 baris

Distribusi Tiap Jenis Latihan di Dataset Awal:
Workout_Type
Strength      464
Cardio        428
Yoga          425
HIIT          389
nan           61
\nStrength    12
\tCardio      11
\tYoga        10

=====
HASIL PURIFIKASI DATA LATIH LENGAN
=====
Dimensi Data Setelah Saringan : 447 baris x 15 kolom.
Jumlah Data Hasil Klasifikasi Biceps : 245 baris
Jumlah Data Hasil Klasifikasi Triceps: 202 baris
Data Noise Non-Arm Sukses Dibuang : 1353 baris.
```

Gambar 3. 3 Jumlah Baris Sebelum dan Sesudah Tahap Filtering

Mekanisme penapisan data mentah untuk memenuhi kebutuhan data makro sistem dirancang melalui empat tahapan analisis komputasi utama sebagai berikut:

- A. Ekstraksi Ragam Latihan Beban (*Strength Training*) Sistem memuat berkas CSV mentah berisi 1.800 record ke dalam memori komputer menjadi struktur *Pandas DataFrame*. Sebelum melakukan penyaringan, sistem membersihkan karakter spasi liar (`\n` dan `\t`) pada kolom `Workout_Type` menggunakan fungsi `.str.strip()`. Selanjutnya, menggunakan operator pencarian string dinamis, sistem memangkas secara massal sekitar 1.353 baris aktivitas non-beban (seperti data yoga, hiit, dan kardio) dan murni mengekstrak baris yang memiliki nilai kategori latihan beban (*Strength Training*) melalui perintah:

```

# 2. BACA DATASET
df = pd.read_csv(io.BytesIO(uploaded[nama_file_uploaded]))
df['Workout_Type'] = df['Workout_Type'].astype(str).str.strip()

print("ANALISIS DISTRIBUSI SELURUH DATASET SEBELUM DISARING")
print("=" * 100)
total_awal = len(df)
print(f"Total Keseluruhan Data Mentah: {total_awal} baris\n")
print("Distribusi Tiap Jenis Latihan di Dataset Awal:")
print(df['Workout_Type'].value_counts().to_string())

# 3. PROSES DATA CLEANING (AMBIL KATEGORI STRENGTH)
df_cleaned = df[df['Workout_Type'].str.contains('Strength', case=False, na=False)].copy()

# Konversi kolom fitur kunci menjadi angka numerik murni
fitur_kunci = ['Avg_BPM', 'Session_Duration (hours)', 'Calories_Burned', 'Fat_Percentage']
for col in fitur_kunci:
    if col in df_cleaned.columns:
        df_cleaned[col] = pd.to_numeric(df_cleaned[col], errors='coerce')

```

Gambar 3. 4 Skrip Pembersihan dan Kategorisasi Dataset

- B. Konversi Paksa Tipe Data (*Data Casting*) dan Eliminasi Nilai Kosong
- Untuk menjamin kelancaran komputasi algoritma statistik pada backend aplikasi tanpa memicu kegagalan eksekusi saat aplikasi dijalankan (*runtime error*), fitur-fitur biometrik kunci dari dataset Kaggle yang bertindak sebagai variabel prediktor yaitu Avg_BPM, Session_Duration (hours), Calories_Burned, dan Fat_Percentage dikonversi paksa menjadi tipe data numerik murni melalui metode `pd.to_numeric()`. Argumen `errors='coerce'` ditanamkan secara preventif agar jika terdapat karakter teks cacat (*missing value* atau tulisan kosong), karakter tersebut otomatis diubah menjadi nilai NaN (*Not a Number*). Sektor baris kotor yang mengandung nilai NaN tersebut kemudian dieliminasi total dari struktur array menggunakan fungsi `df_cleaned.dropna()`.
- C. Penapisan Batas Anomali Fisiologis (*Boundary Checking*) Sistem melakukan penyaringan terhadap rekaman detak jantung rata-rata pengguna (Avg_BPM) untuk menjamin bahwa data latih yang digunakan murni merepresentasikan kondisi fisik manusia saat melakukan aktivitas olahraga intensitas menengah hingga tinggi. Berkas data yang dipertahankan disaring secara ketat melalui batas operasi matematika

denyut nadi logis aktivitas olahraga intensitas tinggi, yaitu pada rentang nilai rentang:

$$110 \leq \text{Avg_BPM} \leq 180$$

- D. Pembentukan Sub-Klasifikasi Target Lengan Terikat Mengingat data awal dari Kaggle tidak mendefinisikan area anatomi otot secara spesifik (hanya tertulis kategori besar *Strength*), perancangan sistem membagi records data *Strength* yang telah lolos saringan secara proporsional ke dalam dua kelompok latihan lengan memanfaatkan penapis acak terikat pada pustaka *NumPy*:

```
df_cleaned = df_cleaned.dropna(subset=fitur_kunci)
df_cleaned = df_cleaned[(df_cleaned['Avg_BPM'] >= 110) & (df_cleaned['Avg_BPM'] <= 180)].copy()

# PROSES GENERATE SUB-KLASIFIKASI BICEPS & TRICEPS SECARA ILMIAH
np.random.seed(42)
pilihan_latihan = np.random.choice(['Biceps Curl', 'Triceps Extension'], size=len(df_cleaned), p=[0.55, 0.45])
df_cleaned['workout_Type'] = pilihan_latihan
```

Gambar 3. 5 Kode *Eliminasi Missing Value* dan Klasifikasi Latihan

Penerapan fungsi `np.random.seed(42)` bertindak sebagai kunci generator angka acak semu (*pseudo-random number generator lock*). Analisis logika ini sangat vital dalam rekayasa perangkat lunak agar variasi sebaran baris data hasil pembagian mutlak bersifat konsisten (konstan) dan menghasilkan keluaran data yang sama persis setiap kali skrip program dieksekusi ulang. Hal ini menjaga agar model data terhindar dari bias fluktuasi acak.

Hasil akhir dari pemrosesan ini berhasil mereduksi data mentah secara presisi dan menghasilkan berkas data steril bernama `cleaned_arm_workout_dataset.csv` dengan dimensi 447 baris data murni, yang terbagi secara proporsional ke dalam dua variasi latihan lengan:

1. Kategori *Biceps Curl* berjumlah 245 baris data.
2. Kategori *Triceps Pushdown* berjumlah 202 baris data.

- E. Ringkasan Karakteristik Deskriptif dan Matriks Korelasi Output Dataset
- Sebagai bentuk validasi formal terhadap pemenuhan kebutuhan data, proses eksekusi skrip pra-pemrosesan di atas menghasilkan keluaran berupa karakteristik statistik deskriptif nilai batas prediktor. Nilai numerik murni dari parameter biometrik yang berhasil disterilkan dirangkum secara ringkas pada Tabel 3.1.

Tabel 3. 1 Karakteristik Deskriptif Atribut Latihan Lengan Bersih

Parameter Statistik	Avg_BP M (Nadi)	Session_Duration (Jam)	Calories_Burned (Kalori)	Fat_Percentage (Lemak)
Mean (Rata-rata)	144.15	1.24	935.48	27.63
Std (Simpangan Baku)	14.16	0.43	422.39	8.24
Min (Nilai Batas Bawah)	120.00	0.50	204.00	11.50
Max (Nilai Batas Atas)	169.00	2.00	1795.00	44.90

Berdasarkan hasil eksekusi pada Tabel 3.1, nilai minimal detak jantung rata-rata berada pada angka 120.00 BPM dan batas maksimal berada pada 169.00 BPM, yang membuktikan bahwa data anomalus di luar batas latihan beban telah tereliminasi sepenuhnya.

Selain sebaran nilai deskriptif, sistem melakukan uji koefisien korelasi Pearson secara linear untuk memastikan tingkat keeratan hubungan antarfitur serta mendeteksi ada tidaknya gejala multikolinearitas ekstrem yang dapat mengganggu performa model statistik backend. Nilai matriks korelasi antarvariabel dari total 447 baris data hasil purifikasi ditunjukkan pada Tabel 3.2.

Tabel 3. 2 Matriks Korelasi Pearson Nilai Validitas Fitur Lengan

Atribut Fitur	Avg_B PM	Session_D uration	Calories_B urned	Fat_Percen tage
Avg_BPM	1.0000	-0.0159	0.4079	0.0537
Session_Duration	-0.0159	1.0000	0.5310	0.0469
Calories_Burned	0.4079	0.5310	1.0000	-0.0631
Fat_Percentage	0.0537	0.0469	-0.0631	1.0000

Nilai matriks korelasi pada Tabel 3.2 membuktikan secara komputasional bahwa seluruh nilai korelasi antarvariabel independen berada jauh di bawah ambang batas kritis 0.70. Indikator ini menegaskan bahwa prediktor data latih bersifat independen, bebas dari multikolinearitas ekstrem, serta memenuhi syarat legalitas kebutuhan data untuk ditransmisikan ke database utama.

3.4.3 Analisis Kebutuhan Dataset Latih & Telemetri Sensor

Seluruh variabel *input* yang dikonsumsi oleh model *Machine Learning* (Multiple Linear Regression) dan *Rule-Based System* pada *backend* murni bersumber dari data pengukuran sensor fisik (*real sensor measurement*) yang ditangkap saat eksekusi latihan. Pemetaannya disajikan pada Tabel 3.3 berikut:

Tabel 3.3 Pemetaan Parameter Input Sensor Fisik untuk Machine Learning

Parameter Telemetri	Sensor Fisik	Satuan	Rentang Nilai Riil	Tipe Data	Peran Output / pada Modul ML & Sistem Pakar
Sudut Kemiringan (θ)	MPU6050	Derajat ($^{\circ}$)	$0^{\circ} - 180^{\circ}$	Float	Klasifikasi Validasi Repetisi & Time Under

					<i>Tension</i> (TUT)
G-Force (G)	MPU6050	G (m/s^2)	8.0G – 12.0G	Float	Deteksi Akselerasi & Identifikasi <i>Failed Reps</i>
Heart Rate (BPM)	MAX30102	BPM	60 – 180 +	Integer	Predictor X_1 pada MLR & Evaluasi MHR ($\geq$ 85%)
Saturasi Oksigen (SpO₂)	MAX30102	%	90% – 100%	Integer	Parameter Kontrol Batas Darurat (<i>Emergency State</i> < 92%)
Suhu Kulit (T)	DS18B20	°C	30.0°C – 37.0°C	Float	Parameter Kontrol Kondisi Fisiologis Fusi Data

Mekanisme Transmisi dan Konsumsi ML:

Transmisi *payload* telemetri mentah dari ESP32 dikirim via BLE secara *real-time* untuk kalkulasi sudut dan gerakan kinetik. Selanjutnya, saat 1 set latihan diselesaikan, *frontend* mengagregasikan data tersebut menjadi 1 paket *payload* JSON (bpm_avg, temp_avg, spo2_avg, failed_reps, dan load_weight) yang dikirim ke *backend* untuk dievaluasi oleh model *Machine Learning*. Pembuktian hasil pemrosesan 10 skenario data riil sensor ini dibahas secara mendalam pada Subbab 4.4.4 (Tabel 10 Skenario Pengujian Lapangan).

3.4.4 Analisis Kebutuhan Fungsional

Kebutuhan fungsional mendeskripsikan layanan atau fitur yang harus disediakan oleh sistem, antara lain:

A. Perangkat Keras (*Hardware*)

1. Perangkat *wearable* harus mampu membaca detak jantung (BPM) dan saturasi oksigen (SpO₂) pengguna secara *real-time* melalui sensor biometrik.
2. Perangkat harus mampu mengukur perubahan akselerasi dan sudut orientasi lengan bawah (*forearm*) untuk mendeteksi repetisi latihan beban.
3. Perangkat harus mampu memproses data mentah sensor di dalam mikrokontroler sebelum ditransmisikan.
4. Perangkat harus mampu mengirimkan data biometrik dan mekanika secara nirkabel melalui protokol komunikasi lokal menuju server/database.
5. Perangkat harus memiliki indikator visual (layar OLED) pada fisik alat untuk menampilkan status koneksi dan daya baterai.

B. Perangkat Lunak (*Software*)

1. Sistem *dashboard* web harus menyediakan halaman autentikasi (*Login*) yang aman untuk member gym dan *personal trainer*.
2. Sistem harus mampu menerima data kiriman dari mikrokontroler dan menyimpannya ke dalam basis data secara terstruktur.
3. Sistem web harus mampu menampilkan visualisasi detak jantung, suhu permukaan kulit, Lemak pada tubuh, jumlah repetisi, dan durasi *Time Under Tension* (TUT) secara *real-time*.

4. Sistem web harus mampu menjalankan fungsi algoritma *Multiple Linear Regression* untuk menghasilkan prediksi indeks kelelahan (*fatigue index*).
5. Sistem harus mampu menampilkan notifikasi atau peringatan visual pada *dashboard* jika pengguna terdeteksi mengalami kelelahan berlebih (*overtraining*).

3.4.5 Analisis Kebutuhan Non-Fungsional

Kebutuhan non-fungsional berkaitan dengan properti sistem yang menunjang kualitas operasional:

A. Perangkat Keras (*Hardware*)

1. Dimensi dan Ergonomi, Perangkat *wearable* harus ringan, compact, dan nyaman digunakan di area *dorsal forearm* tanpa mengganggu mobilitas gerakan latihan beban.
2. Catu Daya, Perangkat harus menggunakan baterai baterai Li-ion yang dilengkapi modul manajemen daya (TP4056) agar dapat bertahan minimal 3–4 jam sesi latihan aktif.
3. Ketahanan, Casing komponen perangkat keras harus kokoh dan terlindungi dari cipratan keringat (*sweat-resistant*).

B. Perangkat Lunak (*Software*)

1. Portabilitas & Responsif, Antarmuka *dashboard* web yang dibangun dengan *framework* Laravel harus responsif dan dapat diakses dengan baik melalui berbagai peramban (*web browser*) di komputer maupun perangkat bergerak.
2. Keandalan (*Reliability*), Sistem harus memiliki latensi (*delay*) pengiriman data yang minimal (di bawah 2 detik) dari perangkat keras menuju *dashboard* web guna menjamin pemantauan secara langsung (*real-time*).

3. Skalabilitas, Struktur basis data MySQL harus mampu menangani *data logging* biometrik yang masuk secara intensif selama sesi latihan berlangsung.

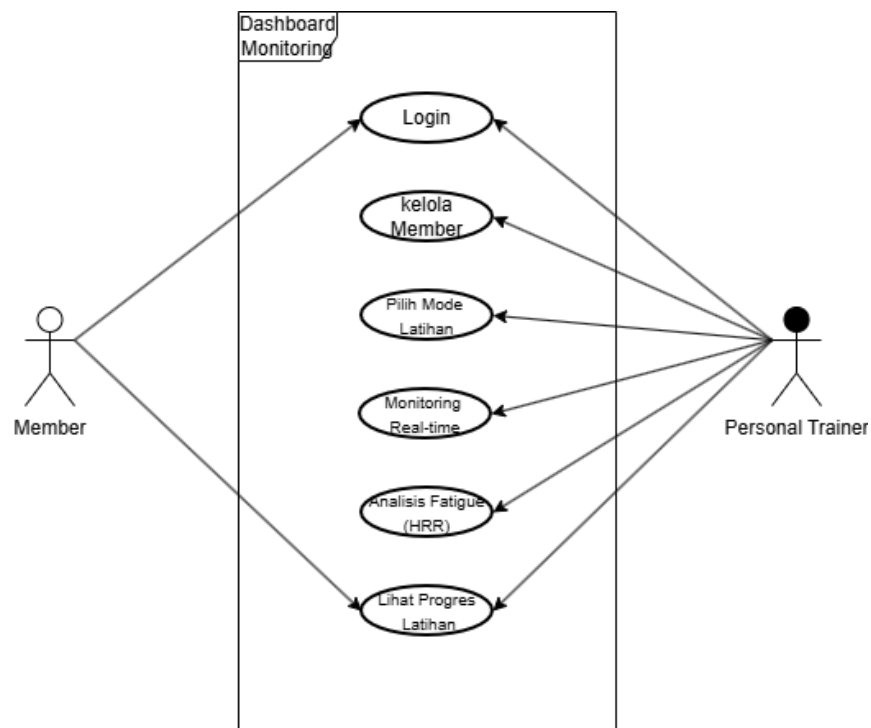
3.5 Perancangan Sistem

3.5.1 Perancangan Pemodelan Sistem (UML)

Perancangan pemodelan dilakukan untuk menggambarkan alur kerja, fungsionalitas, dan interaksi antara pengguna (*member gym* dan *personal trainer*) dengan sistem *dashboard* web Laravel serta perangkat *armband* ESP32. Pemodelan ini digambarkan menggunakan dua diagram UML, yaitu:

A. Use Case Diagram

Use Case Diagram digunakan untuk memetakan aktor-aktor yang terlibat di dalam sistem beserta batasan hak akses fungsional yang mereka miliki pada aplikasi *dashboard monitoring* dan ekosistem *wearable*.



Gambar 3. 6 Use Case Diagram

Untuk memperjelas interaksi dan batasan sistem yang digambarkan pada diagram di atas, berikut adalah detail mengenai definisi aktor serta deskripsi fungsional dari masing-masing *use case*:

1. Definisi Aktor Sistem

Tabel di bawah ini menjelaskan peran dan fungsi dari setiap aktor yang berinteraksi di dalam lingkungan sistem:

Tabel 3.4 Definisi Aktor Sistem

No	Nama Aktor	Jenis Aktor	Deskripsi Peran
1	<i>Member Gym</i>	Pengguna Utama (<i>User</i>)	Anggota pusat kebugaran yang menggunakan perangkat <i>wearable armband</i> saat latihan, melakukan kalibrasi instrumen busur gerak, serta mengakses <i>dashboard</i> web untuk melihat data evaluasi set dan rekomendasi kelelahan harian.
2	<i>Personal Trainer</i>	Pengguna Utama (<i>User</i>)	Instruktur yang memiliki hak akses lebih luas untuk memantau performa, riwayat log reps valid/gagal, serta status kelelahan (<i>fatigue history</i>) dari beberapa <i>member</i> di bawah bimbingannya.
3	<i>Embedded System</i>	Sistem Perangkat Keras	Komponen mikrokontroler (ESP32) beserta rangkaian sensor yang mendeteksi parameter gerak saat latihan aktif dan metrik biometrik pada fase istirahat secara berkala untuk dikirim via BLE Notification.

2. Deskripsi Fungsional *Use Case*

Tabel berikut merinci fungsi, hak akses aktor, serta penjelasan operasional dari setiap *use case* yang tertera pada diagram:

Tabel 3. 5Deskripsi fungsional *Use Case*

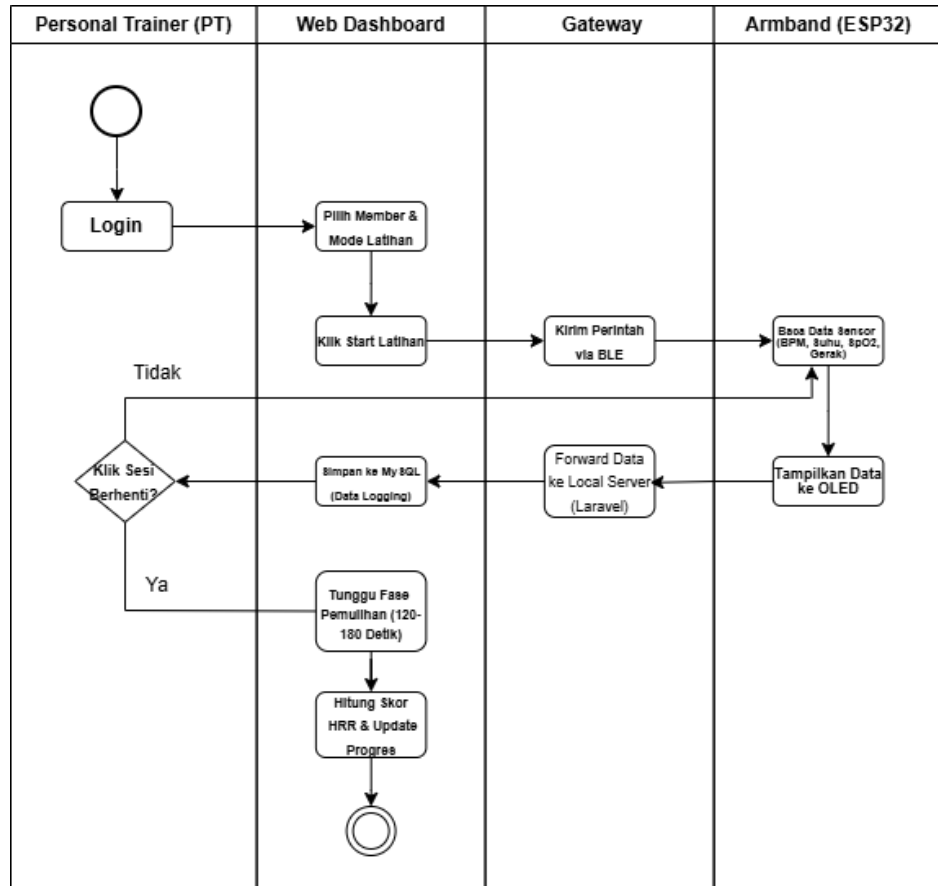
No	Nama Use Case	Aktor yang Terlibat	Deskripsi Fungsi Operasional Sistem
1	Registrasi Akun & Profil Fisik	Member Gym	Proses mendaftarkan akun baru dan menginputkan parameter antropometri beserta data

No	Nama Use Case	Aktor yang Terlibat	Deskripsi Fungsi Operasional Sistem
			pengalaman latihan (lama_gym dalam bulan) sebagai acuan dasar dataset personal.
2	Login Sistem	Member Gym, Personal Trainer	Proses autentikasi hak akses menggunakan <i>username</i> dan <i>password</i> untuk menjamin keamanan data log sebelum masuk ke halaman utama dasbor Laravel.
3	Mengelola Data Member	Personal Trainer	Fitur khusus pelatih untuk memetakan daftar member aktif, melihat perkembangan log reps gagal/sukses, serta mengevaluasi status rekomendasi <i>overtraining</i> .
4	Memulai Sesi Latihan	Member Gym	Aksi untuk mengaktifkan pelacakan data pada <i>dashboard</i> sebelum melakukan set gerakan gym, yang memicu kesiapan sistem menerima data nirkabel.
5	Akuisisi Data Sensor Berfase	Embedded System	Proses berkala di mana sirkuit armband membaca nilai akselerometer MPU6050 saat set berjalan aktif, serta membaca nilai biometrik (MAX30102 & DS18B20) secara diam (senyap) saat masa istirahat selesai.
6	Transmisi Data via BLE Uniform	Embedded System	Proses pengiriman gabungan paket data string terenkapsulasi (Sudut, BPM, Suhu, SpO2, Zona, Baterai) dari mikrokontroler menuju server lokal gateway.
7	Visualisasi Panel Resume Evaluasi	Member Gym, Personal Trainer	Penampilan kotak ringkasan biometrik pasca-latihan yang memuat angka reps valid, reps gagal, serta visualisasi grafik pulsa jantung senyap 5 detik secara otomatis pada akhir set.
8	Prediksi Cerdas Target & Recovery	Member Gym, Personal Trainer	Eksekusi internal web yang menjalankan algoritma <i>Multiple Linear Regression</i> ganda untuk memprediksi target reps dan

No	Nama Use Case	Aktor yang Terlibat	Deskripsi Fungsi Operasional Sistem
			waktu istirahat optimal set berikutnya.
9	Expert System Recommendation	Member Gym, Personal Trainer	Mesin pengambilan keputusan (<i>inference engine</i>) Laravel untuk mengeluarkan notifikasi peringatan tingkat kelelahan (<i>Fatigue</i>) dan saran olahraga hari berikutnya.

B. Activity Diagram

Activity Diagram digunakan untuk menggambarkan aliran kerja (*workflow*), urutan aktivitas proses, serta interaksi paralel yang terjadi antara pengguna, perangkat keras *armband*, dan *framework* Laravel pada sistem monitoring kelelahan gym.



Gambar 3.7 Activity Diagram

Berdasarkan *Activity Diagram* yang dirancang pada Gambar di atas, alur operasional sistem dibagi menjadi tiga entitas utama (*swimlane*) dengan penjelasan tahapan proses sebagai berikut:

1. Sesi Awal Pengguna (*Member / Personal Trainer*)

- Alur dimulai (*start state*) saat pengguna melakukan *Login* ke dalam sistem *dashboard* web Laravel.
- Pengguna masuk ke halaman utama, membuka modul *Hardware Gateway*, melakukan pairing BLE, dan mengunci kalibrasi ambang batas bawah (Lock A) serta ambang batas atas (Lock B) pada lengan. Setelah itu, pengguna memilih opsi "Memulai Sesi Latihan".

2. Pemrosesan Data pada Perangkat Keras (*Wearable Armband ESP32*)

- a. Aktivitas beralih ke perangkat keras ketika pengguna mulai melakukan gerakan latihan beban. Mikrokontroler ESP32 secara aktif mengeksekusi fungsi pemantauan berdasarkan fase kondisi sirkuit:
 1. Fase Set Aktif (*Recording*): Sensor MPU6050 secara mandiri mendeteksi nilai akselerometer dan giroskop lengan. Jika sudut tekukan sendi melewati ambang batas 70% lintasan, reps valid bertambah. Jika tekukan melampaui 25% namun ambruk sebelum 70%, sistem mengunci nilai *TUT Gerakan Gagal*. Pada fase ini, sensor MAX30102 dinonaktifkan dari pembacaan biometrik untuk mengeliminasi *motion artifacts*.
 2. Fase Jeda Istirahat (*Recovery*): Begitu target reps tercapai, status beralih ke *standby*. Pada 6 detik terakhir setelah waktu istirahat habis, sensor MAX30102 dan DS18B20 aktif secara senyap membaca denyut nadi (*bpm_avg*), saturasi (*spo2_avg*), dan suhu kulit (*temp_avg*) saat lengan pengguna dalam posisi diam.
- b. ESP32 merangkum seluruh variabel tersebut menjadi satu baris string payload uniform, lalu mengeksekusi fungsi Transmisi Data via BLE Notification ke Laravel gateway.

3. Pengolahan Data dan Visualisasi pada Perangkat Lunak (*Dashboard Web*)

- a. Server web Laravel menerima pancaran data dari protokol BLE secara *real-time* dan langsung melakukan operasi Penyimpanan ke Database MySQL (*data logging*) agar data latihan terekam secara longitudinal.

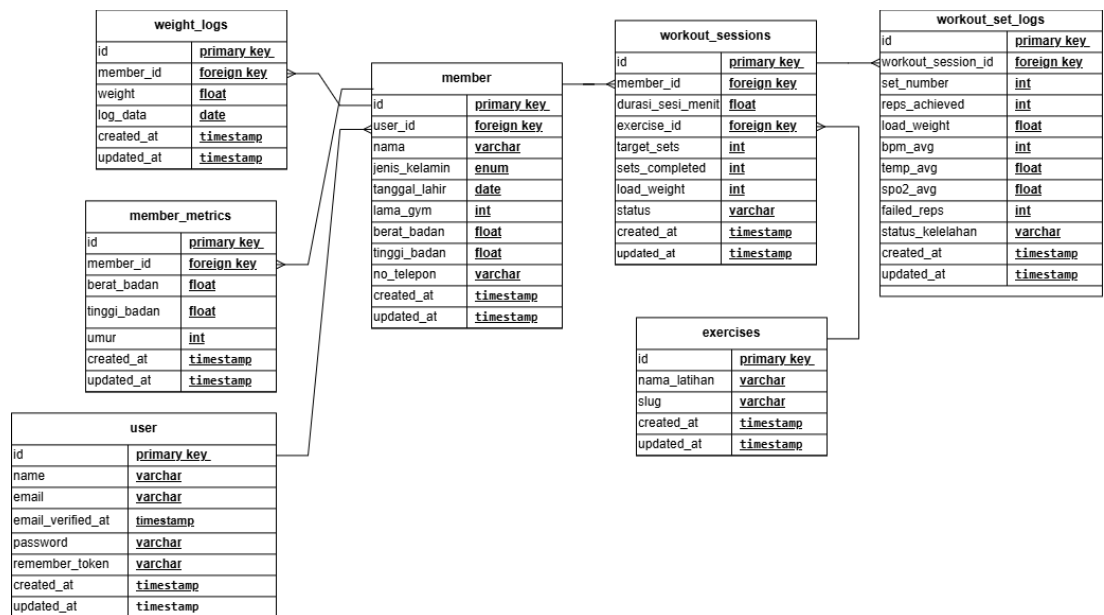
- b. Secara simultan, perangkat lunak menjalankan fungsi Visualisasi *Real-Time* untuk menampilkan fluktuasi parameter fisik (BPM, Suhu, SpO₂, Repetisi) dan menghitung nilai *Time Under Tension* (TUT) pada layar dasbor.
- c. Langkah berikutnya, sistem mengeksekusi Algoritma *Multiple Linear Regression* untuk memproses variabel prediktor detak jantung (X_1) dan durasi gerakan TUT (X_2) menjadi satu nilai *output* kontinu berupa Prediksi Indeks Kelelahan.
- d. Sistem kemudian melakukan percabangan keputusan (*decision node*) untuk mengevaluasi apakah hasil indeks kelelahan tersebut melebihi ambang batas aman (*overtraining*).
 Jika Ya (Melebihi Batas), Sistem web akan memicu dan menampilkan Notifikasi Peringatan Bahaya Visual (*alert notification*) pada layar *personal trainer*.
 Jika Tidak, Grafik pemantauan akan terus berjalan normal tanpa memicu *alert*.
- e. Alur aktivitas pemantauan ini akan terus berulang secara melingkar (*looping*) hingga pengguna secara manual menekan tombol "Akhir Sesi Latihan" (*end state*).

C. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) dirancang untuk memetakan struktur arsitektur basis data, entitas-entitas data yang terlibat, atribut spesifik dari masing-masing entitas, serta tipe hubungan relasional (*relationship*) antar tabel di dalam sistem aplikasi Fit-Tick. Perancangan ERD ini penting untuk memastikan kelancaran fusi data (*sensor fusion*)

dan pengarsipan data transaksional longitudinal yang dikirim secara intensif oleh perangkat keras *armband* IoT menuju database MySQL.

Notasi relasi dalam ERD ini digambarkan menggunakan standar *Crow's Foot Notation* (Notasi Kaki Gagak) untuk merepresentasikan hubungan *One-to-Many* (Satu ke Banyak) maupun *One-to-One* (Satu ke Satu) secara akurat. Gambar arsitektur relasi dari 7 tabel database utama Fit-Tick dapat dilihat pada Gambar 3.8 berikut:



Gambar 3. 8 *Entity Relationship Diagram* (ERD)

Berdasarkan diagram ERD pada Gambar 3.8 di atas, alur relasi dan logika ketergantungan data antar 7 entitas tersebut dapat dijabarkan secara rinci sebagai berikut:

1. Relasi pada Tabel *users* (*One-to-Many/Optional* ke *members*):

Entitas *users* bertindak sebagai penyimpan master data kredensial untuk hak akses otentikasi login pengguna. Kolom *id* pada tabel *users* dihubungkan ke kolom *user_id* pada tabel *members*. Hubungan ini memastikan bahwa satu akun pengguna yang terdaftar di dalam sistem

dapat memetakan, mengontrol, dan memiliki satu profil identitas serta data fisik keanggotaan gym secara spesifik.

2. Relasi pada Tabel *members* (*One-to-Many* ke *member_metrics*):

Kolom *id* yang menjadi *Primary Key* pada tabel *members* diturunkan sebagai kunci tamu atau *Foreign Key* (*member_id*) ke dalam tabel *member_metrics*. Relasi ini memungkinkan aplikasi untuk melakukan pencatatan banyak riwayat pembaruan parameter kondisi fisik (*metrics*) dinamis dari seorang member secara longitudinal dari waktu ke waktu.

3. Relasi pada Tabel *weight_logs* (*Many-to-One* dari *members*):

Tabel *weight_logs* berfungsi khusus merekam jejak data berat badan pengguna. Kolom *member_id* pada tabel ini merujuk kembali menuju kolom *parent id* pada tabel *members*. Hubungan transaksional ini memfasilitasi fungsionalitas pelacakan berat badan (*weight tracking*), di mana satu orang *member* dapat menyimpan banyak log rekam berat badannya secara periodik untuk diolah menjadi grafik tren perkembangan fisik.

4. Relasi pada Tabel *member_metrics* (*Many-to-One* dari *members*):

Tabel *member_metrics* berdiri sebagai entitas penampung data antropometri berkala (berat badan, tinggi badan, dan kalkulasi umur dinamis). Melalui relasi *Foreign Key member_id* yang terikat ke tabel *members*, sistem dapat mengelompokkan data metrik fisik harian agar menjadi dasar variabel input yang valid bagi fungsi pengolahan rumus prediksi komposisi tubuh (*Body Fat Estimator*).

5. Relasi pada Tabel *exercises* (*One-to-Many* ke *workout_sessions*):

Tabel master *exercises* bertindak sebagai penyedia katalog referensi jenis gerakan latihan beban yang terdaftar di dalam sistem. Kolom utama *id* pada tabel *exercises* didistribusikan menuju kolom *Foreign Key exercise_id* pada tabel *workout_sessions*. Hubungan relasional ini mengatur bahwa satu jenis menu latihan gerakan gym (seperti *Bicep Curl*

atau *Tricep Pushdown*) dapat dipilih dan digunakan di dalam banyak sesi latihan aktif yang berbeda oleh para pengguna.

6. Relasi pada Tabel *workout_sessions* (*Many-to-One* dari *members* & *exercises*):

Tabel *workout_sessions* mengikat relasi multidimensi antara aktivitas member dengan jenis latihan yang dipilih. Entitas ini menerima data kunci tamu *member_id* dari tabel *members* dan kunci *exercise_id* dari tabel *exercises* secara bersamaan. Struktur ini merepresentasikan logika bisnis pusat kebugaran, di mana ketika pengguna memulai latihan aktif di web *dashboard*, sistem akan menjadwalkan satu baris sesi latihan baru yang spesifik berdasarkan identitas member dan target jenis gerakannya.

7. Relasi pada Tabel *workout_set_logs* (*Many-to-One* dari *workout_sessions*):

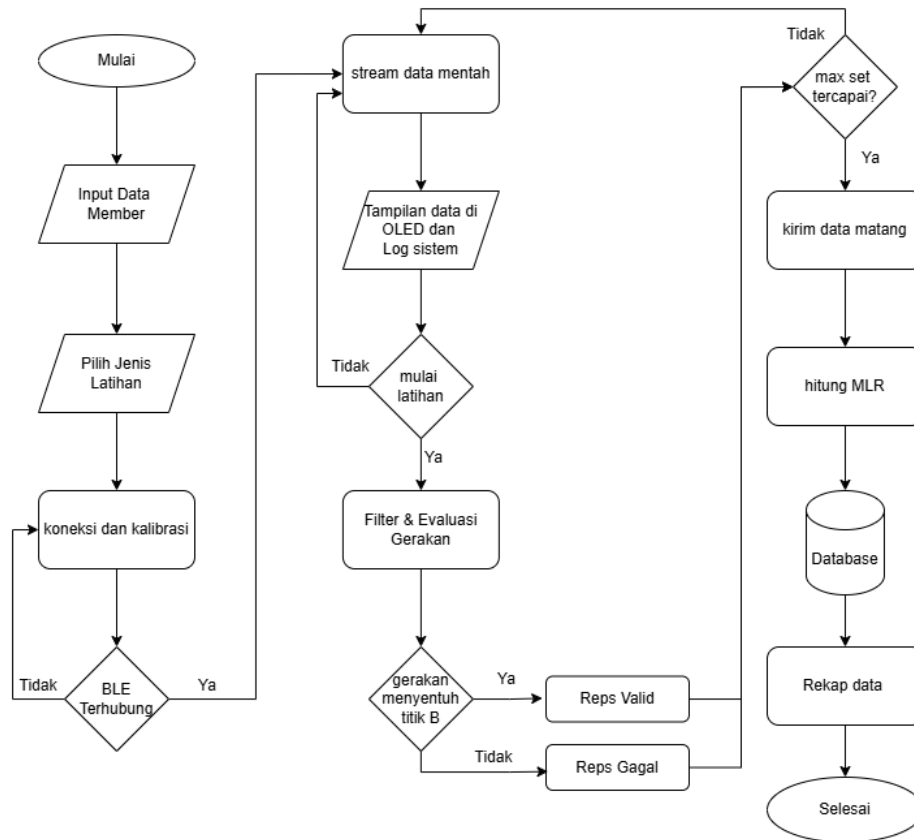
Tabel *workout_set_logs* merupakan muara akhir dari seluruh proses penampungan data transaksional longitudinal fusi sensor nirkabel per nomor set latihan. Kolom *Foreign Key workout_session_id* menghubungkan tabel ini ke tabel induknya yaitu *workout_sessions*.

Struktur *One-to-Many* ini memungkinkan satu sesi latihan aktif menampung banyak baris log data sensor, di mana setiap set latihan yang diselesaikan oleh *member* akan mendokumentasikan nilai rata-rata denyut jantung (*bpm_avg*), saturasi oksigen darah (*spo2_avg*), suhu kulit (*temp_avg*), jumlah repetisi sukses (*reps_achieved*), jumlah repetisi gagal (*failed_reps*), serta teks hasil kesimpulan klinis sistem berupa *status_kelelahan* sebagai bahan log evaluasi longitudinal jangka panjang.

D. Alur Logika Sistem (*Flowchart* Sistem)

Untuk memahami bagaimana perangkat keras *Smart Armband* dan perangkat lunak *Live Monitoring Dashboard* saling terintegrasi dan memproses data, dirancang sebuah alur logika sistem (*flowchart* sistem). Perancangan alur ini berfungsi untuk memetakan sekuensial interaksi pengguna, pemrosesan

sensor, hingga eksekusi sistem pakar di *backend*. Adapun visualisasi dari *flowchart* sistem ini ditunjukkan pada Gambar 3.9 berikut:



Gambar 3. 9 *Flowchart* Sistem

Secara rinci, penjelasan dari setiap tahapan proses yang mengalir pada diagram alur sistem di atas adalah sebagai berikut:

1. Tahap Inisialisasi dan Input Karakteristik Fisik. Sistem diawali dengan simbol *Mulai*, kemudian masuk ke dalam proses input data oleh pengguna. Di sini, *Personal Trainer* atau member gym memasukkan data profil karakteristik subjek melalui antarmuka web Laravel, yang meliputi variabel usia, jenis kelamin, dan akumulasi durasi latihan mingguan (dalam satuan menit), serta menentukan jenis latihan yang akan dieksekusi. Bersamaan dengan itu, sistem menjalankan inisialisasi

awal untuk modul sensor MPU6050, MAX30102, dan layar OLED lokal pada perangkat keras *armband*.

2. Tahap Konektivitas dan Sinkronisasi Bluetooth (BLE). Setelah inisialisasi selesai, aplikasi web akan mengaktifkan Web Bluetooth API untuk melakukan pemindaian dan penyelarasan nirkabel dengan komponen Bluetooth Low Energy (BLE) pada mikrokontroler ESP32. Sistem memiliki percabangan keputusan koneksi; jika BLE gagal terhubung, aliran sistem akan berputar kembali ke tahap inisialisasi untuk mencoba rekoneksi. Jika koneksi berhasil (*Yes*), sistem langsung masuk ke dalam fase penyiapan aliran data mentah (*stream raw data*).
3. Tahap Akuisisi Data Aktual dan *Visualizer Feedback*. Pada fase ini, ESP32 mulai membaca parameter data mentah dari sensor secara real-time. Hasil pembacaan biometrik berupa detak jantung (BPM) dan saturasi oksigen (SpO2) langsung ditampilkan secara lokal pada layar OLED perangkat sebagai feedback instan bagi subjek, sekaligus ditransmisikan bersama data sudut, akselerasi G-Force, dan suhu basal tubuh menuju browser untuk ditampilkan pada konsol telemetri halaman monitor. Aliran data mentah (*raw data stream*) dari perangkat armband ini sengaja dibiarkan berjalan secara kontinu sepanjang gerakan dilakukan murni untuk menyuplai visualisasi baris log aktivitas telemetri pada layar browser agar pergerakan data terlihat responsif. Namun, data biometrik yang mengalir pada fasa latihan aktif ini tidak langsung dicatat ke dalam database transaksi untuk menghindari derau gangguan gerak (*motion artifacts*) pada sensor optik MAX30102
4. Tahap Penentuan Sesi dan Evaluasi Kontraksi Otot. Sistem kemudian mengecek status sesi latihan. Jika status sesi belum diubah menjadi *Active* oleh pengguna, sistem akan tetap berada dalam mode *Standby/Idle* (melakukan *looping* pemantauan biometrik pasif). Namun, jika status

sesi telah aktif, mesin logika berbasis JavaScript pada browser akan mulai mengevaluasi pergerakan lengan subjek.

Proses evaluasi ini bercabang menjadi dua kondisi fungsional:

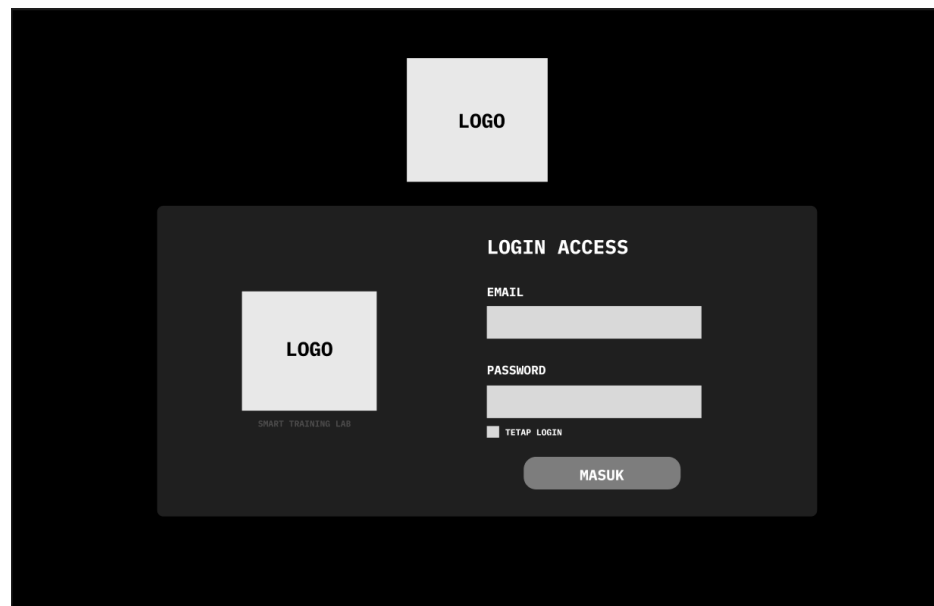
- a. Jalur *Yes (Reps Valid)*, Jika gerakan lengan subjek berhasil menyentuh ambang atas (akselerasi 3.4G atau jangkauan sudut minimal 65%) serta berhasil ditahan secara konstan selama 0.5 detik (guna memfilter *noise* akibat getaran otot atau *muscle tremor*), maka fungsi *addRep()* dieksekusi dan jumlah repetisi valid bertambah.
 - b. Jalur *No (Reps Gagal)*: Jika gerakan lengan subjek hanya bergerak setengah jalan dan berayun turun kembali tanpa pernah menyentuh batas ambang atas dalam durasi kunci yang ditentukan, sistem mendeteksinya sebagai anomali gerak, memicu fungsi *addFailedRep()*, dan menambah hitungan repetisi gagal.
5. Tahap Pengiriman Data dan Pemrosesan Sistem Pakar *Backend*. Setelah satu repetisi selesai diproses (baik valid maupun gagal), sistem memeriksa apakah target repetisi dalam satu set telah terpenuhi. Jika belum, aliran sistem berputar kembali untuk menerima stream data berikutnya. Jika target set telah terpenuhi (Yes), status sesi otomatis berpindah ke Standby/Recovery. Pada fasa Standby/Recovery inilah, browser akan menampung data pemulihan (*Heart Rate Recovery*) selama jeda istirahat dan secara spesifik mengunci sampling data biometrik steril berdurasi 5 detik tepat pasca-fase istirahat selesai dieksekusi. Selanjutnya, browser mengompilasi data agregat set dan data cuplikan pemulihan tersebut untuk dikirim ke server melalui protokol Ajax Fetch POST. Di sisi backend, server Laravel mengeksekusi formula matematika Regresi Linier Berganda (*Multiple Linear Regression*) untuk menghitung nilai Fatigue Index secara objektif dengan

mengombinasikan variabel fisiologis (BPM hasil sampling pemulihan), variabel performa mekanis (jumlah reps gagal), serta variabel kontrol profil subjek (usia, jenis kelamin, durasi latihan mingguan) Tahap Penyimpanan dan Terminasi Aplikasi: Hasil diagnosis status kondisi fisik (apakah subjek berada dalam kondisi 'Normal' atau 'Kelelahan') disimpan secara permanen ke dalam tabel `workout_set_logs` pada database MySQL. Setelah data tersimpan, sistem masuk ke dalam proses rekap data akhir untuk menyajikan resume latihan komprehensif pada komponen dashboard web, sebelum akhirnya aliran sistem mencapai titik terminasi atau *Selesai*.

3.5.2 Perancangan Antarmuka (*User Interface Design*)

Perancangan antarmuka (*User Interface Design*) dirancang untuk memberikan gambaran visual, alur interaksi, serta tata letak elemen fungsional (*layout*) aplikasi web Fit-Tick sebelum diimplementasikan ke dalam kode program menggunakan *framework Laravel 12*. Desain antarmuka ini berfokus pada kemudahan penggunaan (*usability*) dan aspek responsif agar personal trainer maupun *member gym* dapat memantau data biometrik serta mekanika gerak secara nyaman melalui berbagai perangkat (*desktop* maupun *smartphone*). Perancangan antarmuka pada sistem ini dibagi menjadi beberapa halaman utama yang mewakili kebutuhan fungsional sistem, sebagai berikut:

1. Halaman Login Access

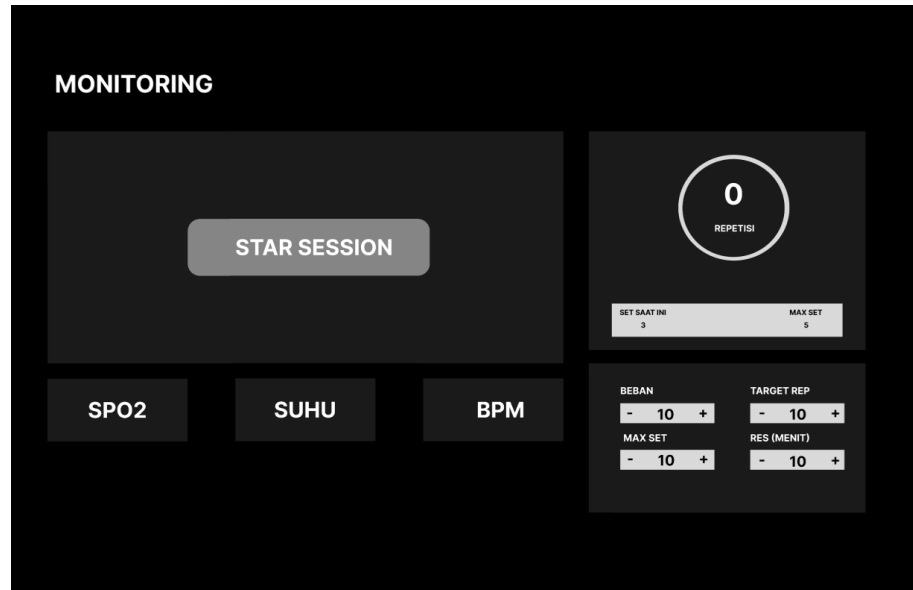


Gambar 3. 10 Halaman Login Sistem

Halaman ini merupakan gerbang utama untuk mengakses sistem. Mengusung desain minimalis, halaman ini fokus pada keamanan dan kemudahan navigasi.

- a. Fungsi: Melakukan autentikasi kredensial pengguna (Email dan *Password*).
- b. Elemen Utama: Terdapat logo sistem sebagai identitas visual, *form input* yang terhubung dengan tabel users, dan fitur "Tetap Login" untuk menjaga durasi sesi pengguna.

2. Halaman monitoring



Gambar 3. 11 Halaman Monitoring Sistem

Halaman ini merupakan *dashboard* utama yang menyajikan data longitudinal secara *real-time* dari perangkat *armband*.

- a. Fungsi, Memantau jalannya latihan dan kondisi fisik pengguna saat melakukan set gerakan.
- b. Indikator Utama:
 1. Panel Data Biometrik, Menampilkan fluktuasi nilai BPM, Saturasi Oksigen (SpO₂), dan Suhu Tubuh.
 2. Panel Mekanika Gerak, Menampilkan hitungan Repetisi otomatis hasil olah data sensor MPU6050.
 3. Input Konfigurasi Set, Panel pengaturan beban angkatan (*load weight*), target repetisi, batas maksimum set, waktu istirahat (*rest time*), serta menu pilihan jenis gerakan yang dikunci pada variasi latihan '*Bicep Curl*' atau '*Tricep Pushdown*'. Input ini krusial agar personal trainer dapat menandai jenis akumulasi kelelahan otot spesifik yang sedang diuji pada database server.

4. Kontrol Sesi, Tombol "Start Session" untuk memulai akuisisi data nirkabel dari ESP32.

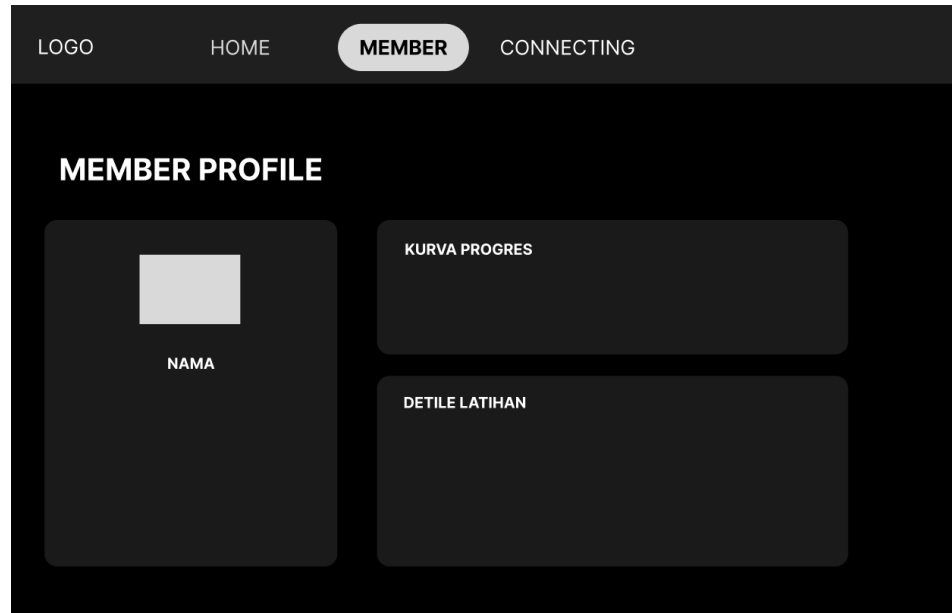
3. Halaman *Set Up* (Kalibrasi & Metrik)

Gambar 3. 12 Halaman *Set Up*

Halaman ini berfungsi sebagai pusat kendali teknis dan pembaruan data fisik pengguna.

- a. Kalibrasi Repetisi: Fitur Titik 1 dan Titik 2 digunakan untuk mengatur ambang batas rentang gerak (*Range of Motion*). Ini memastikan sensor mampu mendeteksi repetisi secara akurat meskipun setiap pengguna memiliki panjang lengan yang berbeda.
- b. Manajemen Latihan: Fitur pemilihan jenis gerakan melalui menu lungsur (*dropdown*) yang menarik data dari tabel exercises (Contoh: *Bicep Curl*).
- c. Update Physical Metrics: Form pembaruan Tanggal Lahir, Berat Badan, dan Tinggi Badan yang datanya akan disimpan ke dalam tabel *member_metrics* sebagai variabel input algoritma prediksi lemak tubuh.

4. Halaman *Member Profile Detail* (Rekapitulasi)



Gambar 3. 13 Halaman *Member Profile Detail*

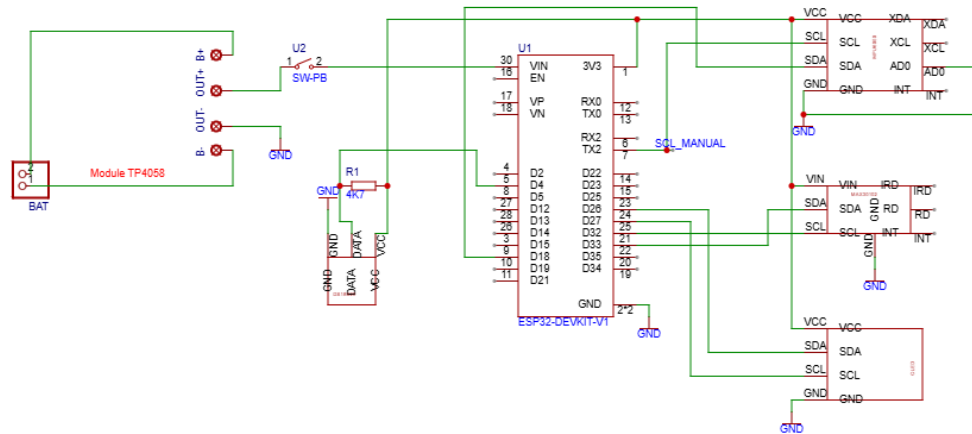
Halaman ini difokuskan pada analisis data historis untuk melihat perkembangan latihan jangka panjang.

- a. Kurva Progres: Visualisasi grafik tren peningkatan performa dari waktu ke waktu.
- b. Detail Latihan: Rekapitulasi mendalam dari setiap sesi yang tersimpan dalam tabel *workout_set_logs*, mencakup rata-rata biometrik dan total beban yang diangkat.

3.5.3 Perancangan Perangkat Keras (*Hardware Schema*)

Perancangan perangkat keras (*hardware schema*) pada sistem *wearable armband* Fit-Tick berbasis ESP32 dirancang untuk mengintegrasikan seluruh komponen sensor biometrik dan mekanika ke dalam satu kesatuan sistem tertanam (*embedded system*). Perancangan ini berfokus pada efisiensi tata letak komponen, optimalisasi fusi jalur komunikasi data, serta manajemen distribusi daya mandiri agar perangkat dapat digunakan secara nirkabel saat *member* melakukan latihan beban.

Skema interkoneksi sirkuit elektronik (*wiring diagram*) antar komponen pada perangkat keras Fit-Tick dapat dilihat pada Gambar 3.X berikut:



Gambar 3. 14 Diagram Wiring

Berdasarkan skema sirkuit elektronik pada Gambar 3.14, arsitektur perancangan perangkat keras ini dibagi menjadi tiga Subsistem Utama sebagai berikut:

1. Subsistem Komunikasi Data I2C Ganda (*Dual I2C Bus Management*)

Untuk menghindari konflik alamat (*address collision*) dan mengoptimalkan pembacaan fusi data sensor nirkabel secara longitudinal, sistem menggunakan dua jalur bus I2C yang terpisah pada mikrokontroler ESP32:

- Jalur I2C Master (Hardware): Digunakan secara khusus oleh sensor biometrik MAX30102 (detak jantung dan SpO₂) melalui pin default ESP32, yaitu SDA ke GPIO 32 dan SCL ke GPIO 33.
- Jalur I2C Sekunder (*Software/Manual*): Digunakan oleh sensor inersia MPU6050 (akselerometer dan giroskop) untuk pelacakan repetisi gerakan. Jalur ini dikonfigurasi secara manual (*custom I2C*) menggunakan pin SDA ke GPIO 18 dan SCL ke GPIO 17 (memanfaatkan jalur TX2) demi menjaga kelancaran *traffic data stream* dari kedua sensor yang bekerja simultan.

2. Subsistem Sensor Suhu Satu Jalur (*One-Wire Temperature Interface*)

Sensor suhu kulit DS18B20 dikonfigurasi menggunakan protokol komunikasi *One-Wire* (Satu Jalur Data). Jalur data sensor ini dihubungkan langsung ke GPIO 4 pada ESP32 dengan bantuan *pull-up resistor* eksternal untuk menstabilkan pembacaan data temperatur derajat Celcius ($^{\circ}\text{C}$) dari permukaan kulit pengguna.

3. Subsistem Manajemen Daya Mandiri (*Power Management Block*)

Sebagai perangkat *wearable*, sistem ini dirancang independen dari kabel catu daya statis dengan mengimplementasikan arsitektur daya mandiri:

- Sumber Daya: Menggunakan baterai Lithium Polymer (Li-Po) bertegangan basis 3.7V.
- Regulasi & Pengisian: Modul TP4056 digunakan sebagai unit manajemen pengisian daya (*charger module*) sekaligus proteksi baterai.
- Kontrol Sistem: Sebuah saklar mekanis (*slide switch*) ditempatkan tepat pada jalur keluaran positif (OUT+) modul TP4056 sebelum masuk ke pin VCC ESP32, berfungsi sebagai pemutus atau penyambung aliran daya utama (*On/Off Switch*) perangkat *armband*.

Tabel 3. 6 Pemetaan Pin Sensor dan ESP32 pada Sistem Armband Monitoring

Nama Komponen	Pin Sensor/Modul	Pin Mikrokontroler (ESP32)	Jenis Protokol / Fungsi
Sensor MAX30102	VCC	3.3V	Catu Daya
	GND	GND	Ground
	SDA	GPIO 32	I2C Data (Hardware)
	SCL	GPIO 33	I2C Clock (Hardware)
Sensor MPU6050	VCC	3.3V	Catu Daya
	GND	GND	Ground
	SDA	GPIO 18	I2C Data (Custom)
	SCL	GPIO 17 (TX2)	I2C Clock (Custom)
Sensor DS18B20	VCC	3.3V	Catu Daya
	GND	GND	Ground
	DATA	GPIO 4	One-Wire Protocol

Nama Komponen	Pin Sensor/Modul	Pin Mikrokontroler (ESP32)	Jenis Protokol / Fungsi
Modul TP4056 & Baterai Li-Po	B+ / B-	Kutub Baterai Li-Po	Jalur Pengisian Daya
	OUT-	GND ESP32	Ground Utama
	OUT+	Saklar (Switch) VIN ESP32	Jalur Distribusi VCC Utama

3.6 Rencana Pengujian Sistem

Pengujian sistem merupakan tahapan krusial dalam metode *Prototype* yang bertujuan untuk mengevaluasi fungsionalitas perangkat keras *wearable armband* maupun perangkat lunak web Laravel Fit-Tick. Proses pengujian ini dirancang untuk memastikan bahwa seluruh integrasi komponen, pembacaan data sensor, transmisi nirkabel, hingga kalkulasi algoritma prediktif dapat berjalan sesuai dengan analisis kebutuhan yang telah ditentukan sebelumnya.

Rencana pengujian pada sistem ini dibagi menjadi dua metode utama, yaitu:

3.6.1 Pengujian Black-Box Testing

Pengujian *Black-Box* berfokus pada evaluasi fungsionalitas eksternal aplikasi web Laravel tanpa menguji struktur kode internalnya. Pengujian ini dilakukan dengan cara memberikan sejumlah variasi data input pada antarmuka, lalu memeriksa apakah *output* respons yang dihasilkan sistem sudah sesuai dengan skenario yang diharapkan.

Beberapa skenario utama yang akan diuji dalam metode *Black-Box* ini meliputi:

- A. Fungsi Autentikasi, Menguji ketahanan formulir *login* saat dimasukkan kombinasi email dan *password* yang valid maupun tidak valid dari tabel *users*.
- B. Fungsi Manajemen Latihan, Memastikan menu luncur (*dropdown*) jenis gerakan dari tabel *exercises* dapat dipilih, disimpan, dan berhasil menginisiasi baris baru pada tabel *workout_sessions*.
- C. Fungsi Responsnotifikasi (*Alert*), Menguji apakah dasbor web Laravel mampu memicu pemicu visual (*pop-up alert*) secara otomatis tepat

ketika menerima kiriman log data kelelahan yang melewati ambang batas aman.

3.6.2 Pengujian Perangkat Keras dan Integrasi (*Hardware & Integration Testing*)

Pengujian ini dilakukan secara fisik untuk mengukur tingkat akurasi fusi data sensor (*sensor fusion*) pada mikrokontroler ESP32 serta stabilitas pengiriman data longitudinal menuju server basis data melalui konektivitas nirkabel.

Beberapa parameter kritis yang diuji pada tahapan ini meliputi:

- A. Pengujian Sensor Biometrik (MAX30102 & DS18B20), Membandingkan keakuratan pembacaan rata-rata denyut jantung (*bpm_avg*), saturasi oksigen (*spo2_avg*), dan suhu kulit (*temp_avg*) pada *armband* dengan alat ukur medis konvensional (seperti *Pulse Oximeter* pabrikan) untuk menghitung persentase nilai *error* pembacaan.
- B. Pengujian Sensor Mekanika (MPU6050): Menguji ketepatan algoritma state machine pada mikrokontroler dalam menghitung akumulasi jumlah repetisi gerakan gym secara *real-time* saat lengan *member* bergerak naik dan turun.
- C. Pengujian Latensi Transmisi BLE, Mengukur jeda waktu (latensi dalam satuan milidetik) yang dibutuhkan paket data log sensor dari saat diekstraksi di lengan pengguna hingga berhasil masuk ke dalam tabel transaksional *workout_set_logs* di database MySQL.

