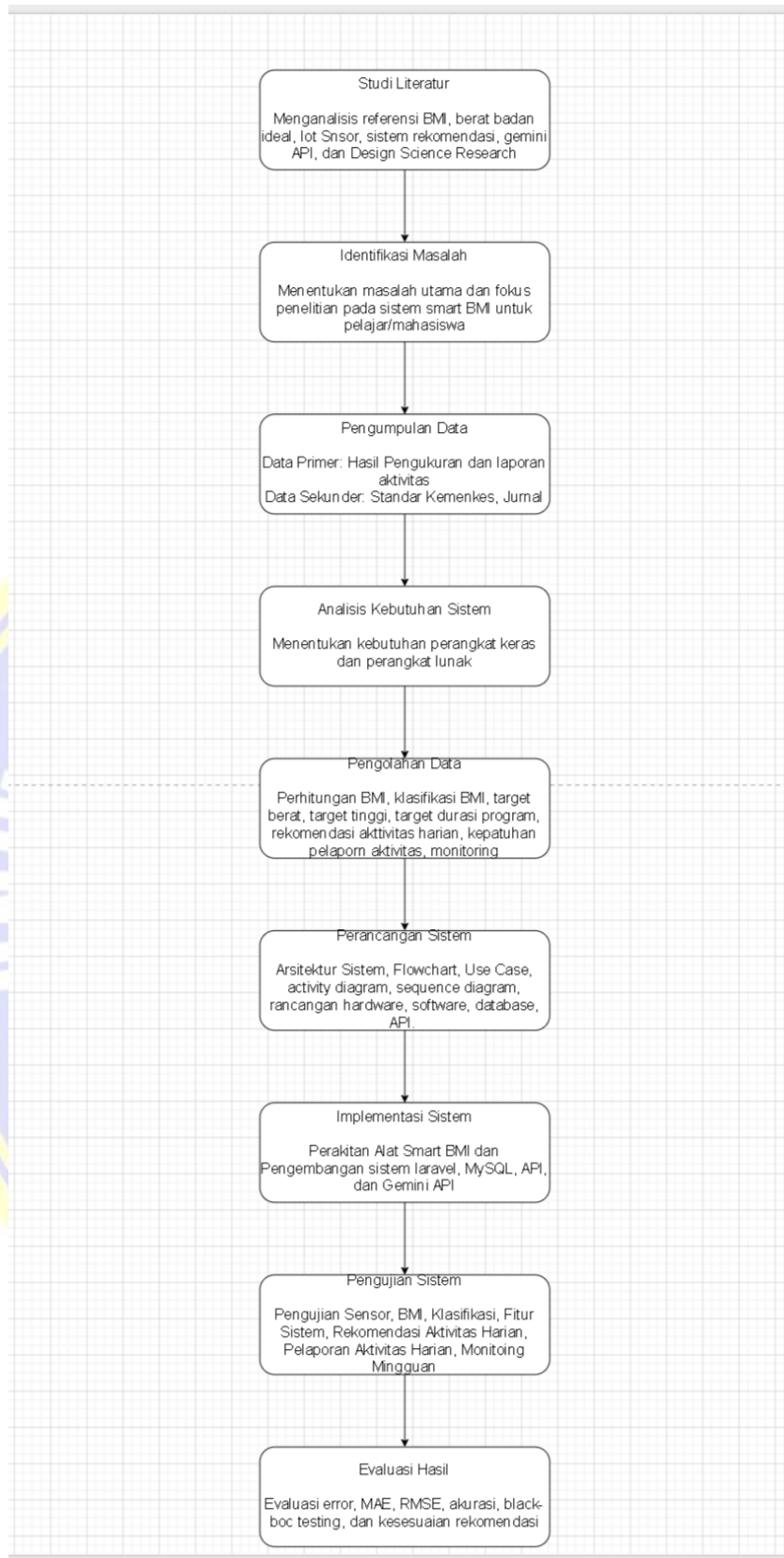


BAB III

METODOLOGI PENELITIAN

Bab ini memaparkan tahapan dan metode yang digunakan dalam penelitian dan pengembangan sistem Smart BMI berbasis mikrokontroler untuk membantu pengguna usia produktif mencapai berat badan ideal mereka. Penelitian ini dilakukan secara sistematis mulai dari tinjauan pustaka, identifikasi masalah, pengumpulan data, analisis persyaratan sistem, pemrosesan data, perancangan sistem, implementasi sistem, pengujian sistem, evaluasi hasil, hingga kesimpulan dan rekomendasi. Tahapan-tahapan ini digunakan untuk memvalidasi sistem yang dikembangkan agar memenuhi persyaratan penelitian dan mampu memecahkan masalah penelitian yang telah ditetapkan.

Proses penelitian dimulai dengan tinjauan pustaka untuk membangun landasan teoretis dan mengumpulkan referensi mengenai BMI, berat badan ideal, sistem pemantauan kesehatan, Internet of Things, sensor, sistem rekomendasi, Gemini API, dan Design Science Research. Selanjutnya, masalah didefinisikan dan data dikumpulkan. Hal ini menjadi dasar untuk pemrosesan data dan perancangan sistem. Sistem yang dirancang kemudian diimplementasikan, diuji, dan dievaluasi untuk menentukan kesesuaian fungsinya dalam pengukuran, perhitungan BMI, klasifikasi BMI, rekomendasi aktivitas harian, pelaporan aktivitas, serta pemantauan BMI mingguan.



Gambar 3. 1 Tahapan Penelitian

3.1 Metode Penelitian

Penelitian ini menggunakan metode Design Science Research (DSR) karena penelitian berfokus pada perancangan, pengembangan, dan evaluasi artefak teknologi berupa sistem Smart BMI berbasis mikrokontroler. Metode ini dipilih karena sesuai dengan tujuan penelitian yang tidak hanya mengkaji permasalahan BMI tidak ideal pada pengguna usia produktif, tetapi juga menghasilkan solusi berbasis sistem yang dapat digunakan untuk melakukan pengukuran berat badan dan tinggi badan, perhitungan BMI, klasifikasi status BMI, pemberian rekomendasi tugas harian, pelaporan aktivitas, serta monitoring perkembangan BMI secara mingguan. Dengan pendekatan DSR, proses penelitian diarahkan untuk menghasilkan artefak yang memiliki fungsi praktis dan dapat dievaluasi berdasarkan kesesuaian antara rancangan sistem, kebutuhan pengguna, dan tujuan penelitian.

Artefak yang dikembangkan dalam penelitian ini adalah sistem Smart BMI berbasis mikrokontroler yang mengintegrasikan perangkat keras, perangkat lunak, serta sistem rekomendasi. Perangkat keras terdiri dari sensor loadcell untuk mengukur berat badan, sensor ultrasonik untuk mengukur tinggi badan, dan mikrokontroler sebagai pusat pemrosesan data serta perhitungan BMI. Data hasil pengukuran kemudian dikirimkan ke sistem untuk ditampilkan dan diolah lebih lanjut. Pada sisi perangkat lunak, sistem dirancang untuk menampilkan hasil pengukuran berat badan, tinggi badan, nilai BMI, kategori BMI, target berat badan ideal, estimasi durasi program, rekomendasi tugas harian, pelaporan aktivitas pengguna, dan monitoring perkembangan BMI mingguan. Sistem rekomendasi dalam penelitian ini menggunakan Gemini API alat utama untuk membuat rekomendasi harian berdasarkan data yang diperoleh dari pengguna. Keputusan utama seperti target tinggi badan, berat badan dan durasi program semua ditentukan oleh pengguna, sedangkan Gemini API digunakan untuk membantu menyusun variasi tugas harian berdasarkan data yang diperoleh pengguna.

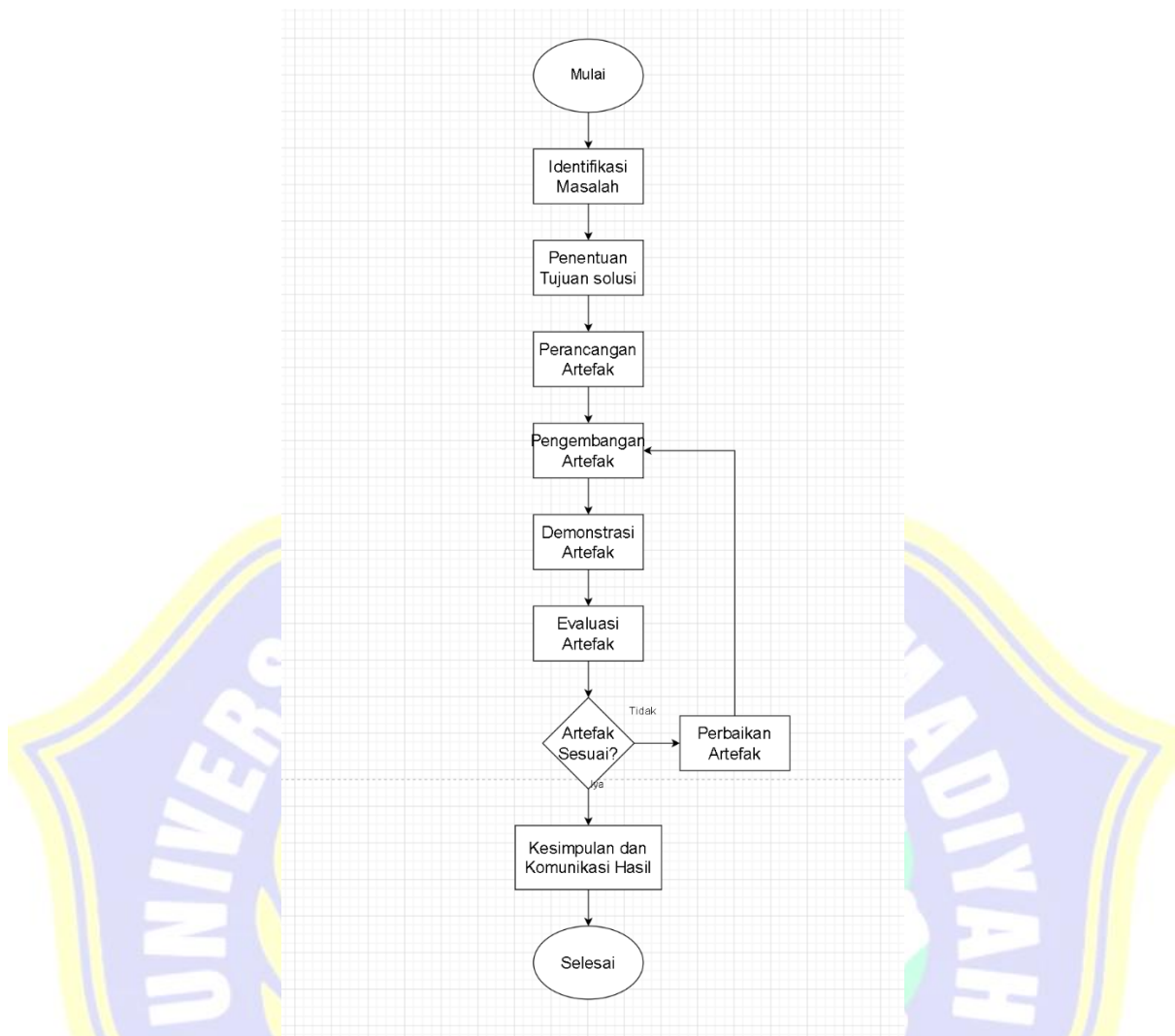
Tahapan penelitian dengan pendekatan Design Science Research meliputi identifikasi masalah, penentuan tujuan solusi, perancangan dan pengembangan artefak, demonstrasi artefak, evaluasi artefak, dan komunikasi hasil. Identifikasi masalah dilakukan untuk memahami kebutuhan sistem dalam membantu pengguna

memantau dan mencapai BMI ideal. Penentuan tujuan solusi digunakan untuk merumuskan fungsi utama sistem. Perancangan dan pengembangan artefak dilakukan melalui desain perangkat keras, perangkat lunak, basis data, alur pengolahan data, mekanisme rekomendasi berbasis Gemini API sebagai mesin utama untuk membuat tugas harian. Demonstrasi dilakukan untuk menunjukkan alur kerja sistem secara menyeluruh, mulai dari pengukuran berat dan tinggi badan, perhitungan BMI, pemberian rekomendasi, pelaporan aktivitas, hingga monitoring BMI mingguan. Evaluasi dilakukan untuk menilai akurasi pengukuran, kesesuaian perhitungan dan klasifikasi BMI, ketepatan target berat badan ideal, kesesuaian rekomendasi, fungsi pelaporan aktivitas, serta monitoring perkembangan BMI. Seluruh tahapan tersebut kemudian didokumentasikan sebagai bagian dari laporan penelitian.

3.1.1 Flowchart Metode Penelitian

Flowchart metode penelitian digunakan untuk menggambarkan tahapan penelitian secara sistematis dari awal hingga akhir. Alur penelitian dimulai dari identifikasi masalah, kemudian dilanjutkan dengan studi literatur, analisis kebutuhan sistem, perancangan sistem, pengembangan artefak, implementasi, pengujian, evaluasi, analisis hasil, dan penyusunan laporan.

Alur flowchart metode penelitian yang digunakan dalam penelitian ini adalah sebagai berikut:



Gambar 3. 2 Flowchart Metode Penelitian

Alur penelitian ini menggunakan pendekatan Design Science Research (DSR) yang berjalan secara sistematis dan iteratif (berulang). Merujuk pada diagram tersebut, tahapan diawali dengan Identifikasi Masalah untuk menggali dan merumuskan masalah spesifik yang melatarbelakangi penelitian. Setelah masalah dipahami, peneliti melanjutkan ke tahap Penentuan Tujuan Solusi, yakni menetapkan target serta kriteria solusi seperti apa yang diharapkan mampu menjawab permasalahan tadi.

Selanjutnya, penelitian masuk ke fase pembuatan solusi. Langkah ini dimulai dengan Perancangan Artefak untuk merumuskan konsep atau desain solusi. Desain tersebut kemudian direalisasikan bentuk nyatanya pada tahap

Pengembangan Artefak. Untuk memastikan solusi ini benar-benar bisa bekerja, dilakukanlah Demonstrasi Artefak dalam kondisi yang menyerupai permasalahan aslinya.

Setelah itu, hasil demonstrasi diukur melalui Evaluasi Artefak guna melihat apakah solusi tersebut sudah mencapai tujuan awal atau belum. Di sinilah letak proses iterasinya. Pada poin keputusan Artefak Sesuai?, jika ternyata hasilnya belum memuaskan ("Tidak"), maka peneliti harus melakukan Perbaikan Artefak dan mengulangi proses dari tahap pengembangan. Namun, jika artefak terbukti berhasil ("Iya"), rangkaian berlanjut ke penarikan Kesimpulan dan Komunikasi Hasil untuk melaporkan kontribusi penelitian, dan penelitian pun dinyatakan selesai.

3.1.2 Identifikasi Permasalahan

Tahap pertama adalah identifikasi permasalahan yang menjadi dasar pengembangan sistem. Permasalahan utama dalam penelitian ini adalah masih terbatasnya sistem pemantauan BMI yang mampu mengintegrasikan pengukuran otomatis, rekomendasi aktivitas harian, pelaporan kegiatan pengguna, dan evaluasi perkembangan BMI secara berkala. Pengguna umumnya hanya mengetahui berat badan atau nilai BMI pada waktu tertentu tanpa adanya pencatatan histori, target berat badan ideal, maupun rekomendasi tindakan yang dapat membantu mencapai BMI ideal. Kondisi tersebut menjadi dasar perlunya pengembangan sistem Smart BMI yang tidak hanya berfungsi sebagai alat ukur, tetapi juga sebagai sistem pendamping menuju berat badan ideal.

3.1.3 Penentuan Tujuan Solusi

Tahap kedua adalah penentuan tujuan solusi. Tujuan solusi dalam penelitian ini adalah mengembangkan sistem Smart BMI berbasis mikrokontroler yang mampu mengukur berat badan dan tinggi badan pengguna secara otomatis, menghitung BMI, mengklasifikasikan status BMI, menentukan target berat badan ideal, memberikan estimasi durasi program, menghasilkan rekomendasi tugas harian, serta memfasilitasi pelaporan aktivitas harian dan monitoring BMI mingguan. Tujuan ini disusun agar sistem yang dikembangkan tidak hanya memberikan informasi kondisi BMI pengguna, tetapi juga memberikan arahan yang dapat dilakukan pengguna secara bertahap.

3.1.4 Perancangan dan Pengembangan Artefak

Tahap ketiga adalah perancangan dan pengembangan artefak. Pada tahap ini dilakukan perancangan sistem yang mencakup perangkat keras, perangkat lunak, basis data, alur pengolahan data, serta mekanisme rekomendasi. Perangkat keras dirancang menggunakan sensor loadcell untuk mengukur berat badan dan sensor ultrasonik untuk mengukur tinggi badan. Data dari sensor kemudian diproses oleh mikrokontroler untuk menghitung BMI. Selanjutnya, data hasil pengukuran dikirimkan ke sistem untuk disimpan, ditampilkan, dan digunakan sebagai dasar klasifikasi BMI serta pemberian rekomendasi. Pada sisi perangkat lunak, sistem dirancang untuk menampilkan hasil pengukuran, target berat badan ideal, estimasi durasi program, tugas harian, laporan aktivitas, dan progres BMI mingguan.

3.1.5 Demonstrasi Artefak

Tahap keempat adalah demonstrasi artefak. Pada tahap ini, sistem Smart BMI yang telah dirancang dan dikembangkan akan digunakan untuk menunjukkan alur kerja sistem secara keseluruhan. Demonstrasi dilakukan dengan skenario pengguna melakukan pengukuran berat badan dan tinggi badan menggunakan alat, kemudian sistem menghitung BMI, menampilkan kategori BMI, menentukan target berat badan ideal, menghasilkan rekomendasi tugas harian, serta menerima laporan aktivitas pengguna. Demonstrasi ini bertujuan untuk memastikan bahwa setiap komponen sistem dapat bekerja sesuai dengan rancangan yang telah dibuat.

3.1.6 Evaluasi Artefak

Tahap terakhir adalah komunikasi hasil. Pada tahap ini, seluruh proses penelitian, mulai dari identifikasi masalah, perancangan sistem, pengembangan artefak, demonstrasi, hingga evaluasi, didokumentasikan dalam bentuk laporan proposal dan laporan skripsi. Hasil penelitian diharapkan dapat memberikan kontribusi dalam pengembangan sistem monitoring kesehatan berbasis teknologi, khususnya sistem Smart BMI yang mengintegrasikan pengukuran otomatis, rekomendasi tugas harian, pelaporan aktivitas, dan monitoring perkembangan BMI.

3.1.7 Komunikasi Hasil

Tahap Communication merupakan tahap akhir dalam metode Design Science Research (DSR) yang bertujuan untuk mengomunikasikan hasil penelitian

kepada pihak yang berkepentingan. Pada tahap ini, seluruh proses penelitian mulai dari identifikasi permasalahan, perancangan dan pengembangan artefak, implementasi sistem, hingga hasil evaluasi didokumentasikan secara sistematis dalam bentuk laporan penelitian. Penyusunan laporan dilakukan agar proses pengembangan sistem, metode yang digunakan, serta hasil yang diperoleh dapat dipahami, dipertanggungjawabkan, dan dijadikan referensi bagi penelitian selanjutnya.

Dalam penelitian ini, hasil pengembangan sistem Smart BMI berbasis mikrokontroler disusun dalam bentuk skripsi yang memuat landasan teori, metodologi penelitian, implementasi sistem, hasil pengujian, serta pembahasan mengenai kinerja sistem. Selain sebagai bentuk dokumentasi ilmiah, penyusunan laporan penelitian juga bertujuan untuk menyebarluaskan pengetahuan mengenai pengembangan sistem monitoring Body Mass Index (BMI) yang dilengkapi dengan fitur rekomendasi program pencapaian berat badan ideal berbasis Gemini API.

3.2 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan untuk memperoleh informasi yang dibutuhkan dalam pengembangan dan pengujian sistem Smart BMI berbasis mikrokontroler. Data yang dikumpulkan digunakan sebagai dasar dalam proses pengukuran berat badan dan tinggi badan, perhitungan BMI, klasifikasi status BMI, penentuan target berat badan ideal, pemberian rekomendasi tugas harian, serta monitoring perkembangan pengguna. Secara umum, data dalam penelitian ini dibagi menjadi dua jenis, yaitu data primer yang diperoleh langsung dari sistem dan pengguna, serta data sekunder yang diperoleh dari standar dan literatur pendukung yang relevan.

3.2.1 Data Primer

Data primer dalam penelitian ini merupakan data yang diperoleh secara langsung dari proses pengukuran sistem Smart BMI dan input aktivitas pengguna selama menjalankan program. Data primer digunakan sebagai dasar utama dalam proses perhitungan BMI, klasifikasi status BMI, penentuan target berat badan ideal, pemberian rekomendasi tugas harian, serta monitoring perkembangan pengguna. Data ini diperoleh melalui dua sumber utama, yaitu perangkat pengukuran berbasis

mikrokontroler dan laporan aktivitas harian yang diisi oleh pengguna melalui sistem.

Data primer pertama adalah data berat badan dan data tinggi badan pengguna. Data berat badan diperoleh melalui sensor loadcell yang terpasang pada alat Smart BMI, sedangkan data tinggi badan diperoleh melalui sensor ultrasonik. Kedua data tersebut diproses oleh mikrokontroler untuk menghasilkan nilai BMI pengguna. Data hasil pengukuran ini menjadi dasar awal bagi sistem dalam menentukan kategori BMI, target berat badan ideal, selisih berat badan terhadap target, serta estimasi durasi program yang diperlukan untuk mencapai BMI ideal.

Selain data pengukuran tubuh, penelitian ini juga menggunakan data pelaporan aktivitas harian sebagai data primer. Data ini diperoleh dari pengguna setelah sistem memberikan tugas harian yang mencakup aktivitas fisik, pola makan, pola tidur, dan kebiasaan sehat. Pengguna melaporkan apakah tugas yang diberikan telah dilakukan atau belum, serta dapat menambahkan informasi pendukung seperti durasi aktivitas, jumlah langkah, catatan makanan, jam tidur, dan kendala yang dialami. Data pelaporan aktivitas harian digunakan untuk menghitung tingkat kepatuhan pengguna terhadap rekomendasi sistem serta menjadi bahan evaluasi dalam monitoring perkembangan BMI secara mingguan.

Secara umum, data primer dalam penelitian ini meliputi:

1. Berat badan pengguna, diperoleh dari sensor loadcell.
2. Tinggi badan pengguna, diperoleh dari sensor ultrasonik.
3. Nilai BMI pengguna, diperoleh dari hasil perhitungan berdasarkan berat badan dan tinggi badan.
4. Kategori BMI pengguna, diperoleh dari hasil klasifikasi berdasarkan standar Kemenkes.
5. Data pelaporan aktivitas harian, diperoleh dari input pengguna terhadap tugas harian yang diberikan sistem.
6. Data monitoring mingguan, diperoleh dari pengukuran ulang berat badan, tinggi badan, dan BMI setiap bulan.

3.2.2 Data Sekunder

Data sekunder dalam penelitian ini merupakan data pendukung yang diperoleh dari standar, pedoman, dokumentasi teknologi, dan literatur ilmiah yang relevan dengan pengembangan sistem Smart BMI berbasis mikrokontroler. Data sekunder digunakan sebagai acuan dalam proses klasifikasi BMI, penentuan target berat badan ideal, estimasi durasi program, penyusunan batasan rekomendasi, serta validasi keluaran rekomendasi tugas harian yang dihasilkan oleh sistem. Dengan demikian, data sekunder tidak digunakan sebagai dataset pelatihan, melainkan sebagai dasar aturan dan referensi agar sistem dapat menghasilkan rekomendasi yang realistis, aman, dan sesuai dengan tujuan penelitian.

Data sekunder utama yang digunakan adalah standar klasifikasi BMI dari Kemenkes. Standar ini digunakan untuk mengelompokkan nilai BMI pengguna ke dalam kategori *underweight*, *normal*, *overweight*, dan *obesitas*. Kategori BMI tersebut menjadi dasar bagi sistem dalam menentukan kondisi awal pengguna, target berat badan ideal, jenis program yang diberikan, serta batas intensitas rekomendasi aktivitas. Selain itu, rentang BMI *normal* digunakan untuk menghitung berat badan ideal minimum dan maksimum berdasarkan tinggi badan pengguna, sehingga target yang diberikan sistem tetap mengacu pada standar yang dapat dipertanggungjawabkan.

Selain standar BMI, penelitian ini juga menggunakan literatur mengenai aktivitas fisik, pola makan sehat, pola tidur, kebiasaan hidup sehat, sistem monitoring kesehatan, dan sistem rekomendasi berbasis aturan. Literatur tersebut digunakan sebagai dasar dalam menyusun batasan rekomendasi dan aturan sistem, khususnya agar rekomendasi yang diberikan tidak bersifat ekstrem, tidak mengandung klaim medis, serta tidak menggantikan konsultasi dengan tenaga kesehatan. Dalam penelitian ini, Gemini API digunakan mesin utama untuk menentukan rekomendasi harian berdasarkan data yang diperoleh dari pengguna, dari berat badan, tinggi badan, usia, jenis kelamin, target berat badan, target tinggi badan dan durasi program. Data tersebut nantinya akan menjadi dasar data untuk pembuatan rekomendasi harian

3.3 Pengolahan Data

Pengolahan data dalam penelitian ini dilakukan untuk mengubah data hasil pengukuran dan laporan aktivitas pengguna menjadi informasi yang dapat digunakan oleh sistem Smart BMI. Data berat badan dan tinggi badan diolah untuk menghasilkan nilai BMI, kategori BMI, target berat badan ideal, estimasi durasi program, dan dasar rekomendasi tugas harian. Selain itu, data pelaporan aktivitas harian digunakan untuk mengetahui tingkat kepatuhan pengguna, sedangkan hasil pengukuran ulang mingguan digunakan untuk memonitor perkembangan BMI pengguna menuju berat badan ideal.

3.3.1 Pengolahan Data Berat Badan dan Tinggi Badan

Data berat badan dan tinggi badan merupakan data utama yang digunakan dalam perhitungan Body Mass Index (BMI). Data berat badan diperoleh dari sensor loadcell dalam satuan kilogram, sedangkan data tinggi badan diperoleh dari sensor ultrasonik dalam satuan sentimeter. Kedua data tersebut dibaca oleh mikrokontroler untuk diproses sebagai dasar perhitungan BMI pengguna.

Sebelum dilakukan perhitungan BMI, data tinggi badan dikonversi terlebih dahulu dari sentimeter ke meter karena rumus BMI menggunakan satuan meter. Konversi dilakukan dengan membagi nilai tinggi badan dalam sentimeter dengan 100. Setelah data berat badan dan tinggi badan berada dalam satuan yang sesuai, data tersebut digunakan untuk menghitung nilai BMI dan disimpan ke dalam sistem sebagai histori pengukuran pengguna.

Rumus konversi tinggi badan adalah sebagai berikut:

$$\left(TB_m = \frac{TB_{cm}}{100} \right) \quad (1)$$

Keterangan:

TB_m = Tinggi Badan Dalam meter

TB_{cm} = Tinggi Badan dalam sentimeter

3.3.2 Perhitungan BMI

Perhitungan Body Mass Index (BMI) dilakukan setelah data berat badan dan tinggi badan pengguna diperoleh serta tinggi badan telah dikonversi ke dalam satuan meter. Nilai BMI dihitung dengan membandingkan berat badan dalam kilogram terhadap kuadrat tinggi badan dalam meter. Hasil perhitungan BMI digunakan sebagai dasar untuk menentukan kategori status berat badan pengguna, seperti underweight, normal, overweight, atau obesitas

$$\left(BMI = \frac{BB}{TB_m^2} \right) \quad (2)$$

Keterangan

BMI = Body Mass Index

BB = Berat Badan (Kg)

TB = Tinggi Badan (m)

3.3.3 Klasifikasi BMI

Klasifikasi BMI dilakukan setelah sistem memperoleh nilai BMI dari hasil perhitungan berat badan dan tinggi badan pengguna. Nilai BMI tersebut digunakan untuk menentukan status berat badan pengguna berdasarkan kategori yang telah ditetapkan. Dalam penelitian ini, klasifikasi BMI mengacu pada standar Kemenkes karena penelitian dilakukan dalam konteks pengguna di Indonesia. Hasil klasifikasi ini menjadi dasar bagi sistem dalam menentukan kondisi awal pengguna, target berat badan ideal, estimasi durasi program, serta batas rekomendasi tugas harian. Kategori BMI yang digunakan dalam sistem dijelaskan pada tabel 3.1

Tabel 3. 1 Klasifikasi BMI Berdasarkan Kemenkes

Nilai BMI	Kategori
$BMI < 18,5$	Underweight / Kurus
$18,5 \leq BMI \leq 25,0$	Normal
$25,0 < BMI \leq 27,0$	Overweight / Gemuk
$BMI > 27,0$	Obesitas

Berdasarkan klasifikasi tabel 3.1, sistem menentukan jenis program pengguna. Pengguna dengan kategori underweight diarahkan pada program peningkatan berat badan secara sehat, pengguna dengan kategori normal diarahkan pada program pemeliharaan berat badan, sedangkan pengguna dapat memilih berapa target tinggi badan dan berat badan yang ingin pengguna capai, dan berapa durasi untuk melakukan target tersebut. Pengguna memiliki kebebasan untuk menentukan target tinggi badan, berat badan serta durasi program. Kategori BMI ini juga digunakan untuk melakukan generate untuk membuat aktivitas harian berdasarkan data pengguna

3.3.4 Penentuan Target Berat Badan Ideal

Penentuan target berat badan ideal dilakukan setelah sistem memperoleh nilai BMI dan kategori BMI pengguna. Target berat badan ideal ditentukan berdasarkan tinggi badan pengguna dan rentang BMI normal menurut standar Kemenkes. Dalam penelitian ini, rentang BMI normal digunakan sebagai acuan karena target sistem adalah membantu pengguna mencapai atau mempertahankan berat badan yang berada pada kategori normal.

Target berat badan ideal dihitung dengan mengalikan batas bawah dan batas atas BMI normal dengan kuadrat tinggi badan pengguna dalam satuan meter. Dengan demikian, sistem tidak hanya mempertimbangkan berat badan pengguna, tetapi juga menyesuaikannya dengan tinggi badan agar target yang diberikan lebih proporsional.

Rumus penentuan rentang berat badan ideal adalah sebagai berikut:

$$(BBI_{min} = 18,5 \times TB_m^2) \quad (3)$$

$$(BBI_{max} = 25,0 \times TB_m^2) \quad (4)$$

Keterangan:

BBI min = batas bawah berat badan ideal (kg)

BBI max = batas atas berat badan ideal (kg)

TBm = Tinggi Badan (m)

3.3.5 Penentuan Selisih Berat Badan

Penentuan selisih berat badan dilakukan untuk mengetahui jarak antara berat badan aktual pengguna dengan target berat badan ideal yang telah ditentukan sistem. Nilai selisih ini digunakan sebagai dasar untuk menentukan arah program, yaitu menaikkan berat badan, menurunkan berat badan, atau mempertahankan berat badan. Selain itu, selisih berat badan juga menjadi acuan dalam menentukan estimasi durasi program dan tingkat rekomendasi tugas harian yang diberikan kepada pengguna.

Pada pengguna dengan kategori underweight, selisih berat badan dihitung dari batas bawah berat badan ideal dikurangi berat badan aktual. Sementara itu, pada pengguna dengan kategori overweight atau obesitas, selisih berat badan dihitung dari berat badan aktual dikurangi batas atas berat badan ideal. Untuk pengguna dengan kategori normal, sistem tidak menetapkan target kenaikan atau penurunan berat badan, melainkan mengarahkan pengguna untuk mempertahankan berat badan dalam rentang ideal.

Rumus penentuan selisih berat badan adalah sebagai berikut:

Untuk pengguna Underweight Kurus

$$(SB = BBI_{min} - BB) \quad (5)$$

Untuk pengguna Overweight / Obesitas

$$(SB = BB - BBI_{max}) \quad (6)$$

Keterangan:

SB = Selisih Berat Badan

BB = Berat Badan Aktual Pengguna

BBmin = Batas Bawah Berat Badan Ideal

BBmax = Batas Atas Berat Badan Ideal

Hasil selisih berat badan tersebut digunakan pada tahap berikutnya untuk menentukan estimasi durasi program agar target yang diberikan sistem bersifat bertahap, realistis, dan sesuai dengan kondisi awal pengguna.

3.3.6 Penentuan Estimasi Durasi Program

Penentuan estimasi durasi program dilakukan setelah sistem memperoleh nilai selisih berat badan pengguna terhadap target berat badan ideal. Estimasi durasi digunakan untuk menentukan jangka waktu yang dibutuhkan pengguna dalam mencapai target secara bertahap. Dalam penelitian ini, durasi program tidak ditentukan secara tetap, tetapi dihitung berdasarkan selisih berat badan dan target perubahan berat badan per bulan agar hasil yang diberikan lebih realistis dan tidak bersifat ekstrem.

Rumus estimasi durasi program yang digunakan adalah sebagai berikut:

$$\left(EDP = \left\lceil \frac{SB}{TPB} \right\rceil\right) \quad (7)$$

Keterangan:

EDP = Estimasi Durasi Program

SB = Selisih Berat Badan

TPB = Target Perubahan Berat Badan

Dalam sistem ini, nilai TPB dapat ditentukan berdasarkan arah program pengguna. Pada program penurunan berat badan, target perubahan dapat dibuat secara bertahap, misalnya sekitar 2 kg per bulan, agar program tidak memberikan target yang terlalu cepat. Pada program peningkatan berat badan untuk pengguna underweight, target perubahan juga dibuat bertahap agar kenaikan berat badan tetap realistis. Jika selisih berat badan pengguna cukup besar, maka estimasi durasi program akan otomatis menjadi lebih panjang dan tetap dievaluasi secara berkala melalui monitoring BMI mingguan.

Dengan pendekatan tersebut, pengguna dengan selisih berat badan kecil akan memperoleh durasi program yang lebih pendek, sedangkan pengguna dengan selisih berat badan besar akan memperoleh durasi program yang lebih panjang. Hasil estimasi durasi ini kemudian digunakan oleh sistem sebagai dasar dalam

menyusun rekomendasi tugas harian, menentukan tingkat intensitas aktivitas, serta mengevaluasi progres pengguna setiap bulan.

3.3.7 Pengolahan Rekomendasi Tugas Harian

Pengolahan rekomendasi aktivitas harian dilakukan setelah sistem memperoleh data profil pengguna yang meliputi umur, jenis kelamin, berat badan, tinggi badan, nilai BMI, target berat badan, target tinggi badan, dan durasi program. Berbeda dengan penelitian sebelumnya yang menentukan target program secara otomatis, pada penelitian ini pengguna diberikan kebebasan untuk menentukan target berat badan, target tinggi badan, serta durasi program sesuai dengan kebutuhannya. Seluruh data tersebut kemudian digunakan sebagai masukan (input) bagi Gemini API untuk menghasilkan rekomendasi aktivitas harian yang dipersonalisasi sesuai kondisi dan tujuan masing-masing pengguna.

Rekomendasi yang dihasilkan oleh Gemini API mencakup beberapa aspek utama, yaitu aktivitas fisik, pola makan, pola tidur, dan kebiasaan hidup sehat yang mendukung pencapaian target pengguna. Dengan memanfaatkan informasi profil pengguna secara menyeluruh, Gemini API dapat menghasilkan rekomendasi yang lebih adaptif dan bervariasi dibandingkan rekomendasi yang bersifat statis, sehingga pengguna memperoleh saran yang lebih relevan dengan kondisi dan target yang telah ditentukan.

Sebelum rekomendasi ditampilkan kepada pengguna, sistem melakukan proses validasi terhadap keluaran (output) Gemini API. Validasi ini bertujuan untuk memastikan bahwa rekomendasi yang diberikan memiliki struktur yang sesuai dengan format yang telah ditentukan serta tidak mengandung respons yang tidak relevan, kosong, atau berada di luar konteks penelitian. Selain itu, sistem juga melakukan penyaringan terhadap rekomendasi yang berpotensi mengandung saran ekstrem, seperti pembatasan makan yang berlebihan, aktivitas fisik yang tidak proporsional, penggunaan obat atau suplemen tanpa indikasi medis, maupun klaim kesehatan yang tidak dapat dipertanggungjawabkan. Apabila keluaran Gemini API tidak memenuhi kriteria tersebut, sistem akan meminta proses generate ulang (re-generation) hingga diperoleh rekomendasi yang sesuai dengan kebutuhan penelitian.

3.3.8 Pengolahan Laporan Aktivitas Harian

Pengolahan pelaporan aktivitas harian dilakukan untuk mengetahui sejauh mana pengguna menjalankan tugas yang telah diberikan oleh sistem. Setelah rekomendasi tugas harian ditampilkan, pengguna diminta untuk melaporkan status pelaksanaan tugas tersebut melalui sistem. Data yang dilaporkan dapat berupa status selesai atau belum selesai, durasi aktivitas fisik, jumlah langkah, catatan pola makan, jam tidur, serta catatan tambahan terkait kendala yang dialami pengguna.

Data laporan aktivitas harian digunakan untuk menghitung tingkat kepatuhan pengguna terhadap tugas yang diberikan. Skor kepatuhan ini menjadi salah satu indikator dalam monitoring progres pengguna, karena sistem tidak hanya menilai perubahan BMI secara mingguan, tetapi juga melihat konsistensi pengguna dalam menjalankan aktivitas harian yang direkomendasikan. Semakin tinggi skor kepatuhan, maka semakin baik tingkat konsistensi pengguna dalam mengikuti program yang diberikan oleh sistem.

Rumus perhitungan skor kepatuhan adalah sebagai berikut:

$$\left(SK = \frac{TS}{TD} \times 100\% \right) \quad (8)$$

Keterangan:

SK: Skor Kepatuhan Pengguna

TS: Jumlah Tugas Diselesaikan

TD: Jumlah Tugas Diberikan

Hasil dari pengolahan laporan aktivitas harian disimpan sebagai histori aktivitas pengguna. Data tersebut digunakan oleh sistem untuk menampilkan status pelaporan harian, menghitung skor kepatuhan, serta menjadi bahan evaluasi pada monitoring BMI mingguan. Dengan demikian, pelaporan aktivitas harian berfungsi sebagai penghubung antara rekomendasi tugas harian dan evaluasi perkembangan pengguna menuju BMI ideal.

3.3.9 Monitoring BMI Mingguan

Monitoring BMI mingguan dilakukan untuk mengevaluasi perkembangan pengguna setelah menjalankan rekomendasi tugas harian yang diberikan oleh sistem. Pengukuran BMI tidak dilakukan setiap hari, melainkan secara berkala

setiap minggu menggunakan sistem Smart BMI. Pada tahap ini, pengguna melakukan pengukuran ulang berat badan dan tinggi badan, kemudian sistem menghitung kembali nilai BMI dan membandingkannya dengan data pengukuran sebelumnya.

Data yang digunakan dalam monitoring BMI mingguan meliputi berat badan awal, berat badan terbaru, tinggi badan, BMI awal, BMI terbaru, kategori BMI, target berat badan ideal, selisih berat badan, serta skor kepatuhan aktivitas harian. Perbandingan data tersebut digunakan untuk mengetahui apakah pengguna mengalami perubahan menuju target yang telah ditentukan. Hasil monitoring ini juga menjadi dasar bagi sistem untuk menampilkan progres pengguna, baik dalam bentuk perubahan berat badan, perubahan nilai BMI, maupun persentase pencapaian target.

Rumus perhitungan progres target adalah sebagai berikut:

$$\left(PT = \frac{BT}{SB} \times 100\% \right) \quad (9)$$

PT: Presentase Progress Target

BT: Berat Badan yyang telah Berubah Menuju Target

SB: Selisih Berat Badan Awal Terhadap Target Berat Badan Ideal

Monitoring BMI mingguan berfungsi sebagai evaluasi berkala terhadap efektivitas program yang sedang dijalankan pengguna. Jika progres pengguna sesuai dengan target, sistem dapat mempertahankan pola rekomendasi yang diberikan. Namun, jika progres belum sesuai, sistem dapat menyesuaikan kembali rekomendasi tugas harian berdasarkan kondisi terbaru pengguna. Dengan demikian, monitoring BMI mingguan menjadi bagian penting dalam memastikan bahwa proses pencapaian BMI ideal berjalan secara terarah, bertahap, dan terdokumentasi.

3.4 Rancangan atau Design yang diusulkan

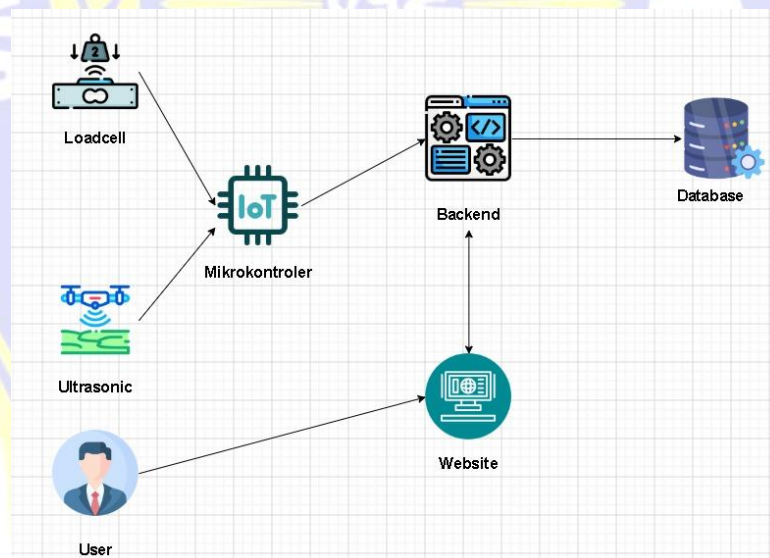
Penelitian ini menjelaskan arsitektur dan alur operasional sistem monitoring pencapaian BMI ideal berbasis ESP32, yang digunakan untuk mengumpulkan data antropometri balita dan mengintegrasikannya ke dalam platform pemantauan berbasis web. Sistem ini dirancang untuk memudahkan pengukuran berat dan tinggi balita, memproses data antropometri, serta menampilkan hasil pemantauan dan evaluasi risiko stunting.

3.4.1 Gambaran Umum Sistem

Sistem Smart BMI yang diusulkan merupakan sistem berbasis mikrokontroler yang dirancang untuk membantu pengguna dalam melakukan pengukuran berat badan, tinggi badan, perhitungan BMI, klasifikasi status BMI, pemberian rekomendasi tugas harian, pelaporan aktivitas, serta monitoring perkembangan BMI secara mingguan. Sistem ini mengintegrasikan sensor loadcell sebagai pengukur berat badan, sensor ultrasonik sebagai pengukur tinggi badan, mikrokontroler sebagai pusat pemrosesan data, serta antarmuka sistem sebagai media penyajian informasi kepada pengguna.

Secara umum, sistem bekerja melalui beberapa tahap utama, yaitu pengukuran berat badan dan tinggi badan, perhitungan BMI, klasifikasi BMI berdasarkan standar Kemenkes, penentuan target berat badan ideal, estimasi durasi program, pemberian rekomendasi tugas harian, pelaporan aktivitas harian, dan monitoring BMI mingguan. Alur kerja sistem secara rinci disajikan pada subbab flowchart system

3.4.2 Arsitektur Sistem



Tabel 3. 2 Arsitektur Diagram

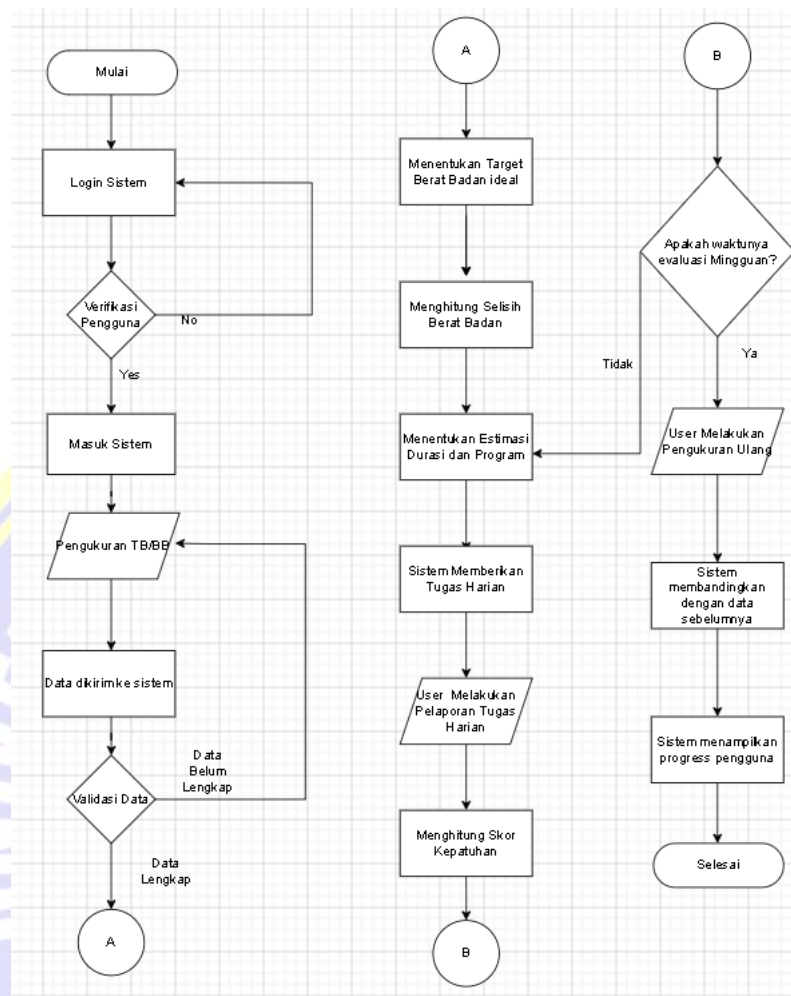
Arsitektur sistem Smart BMI dirancang untuk menghubungkan perangkat pengukuran, backend, database, dan website dalam satu alur sistem yang terintegrasi. Pada bagian perangkat keras, sensor loadcell digunakan untuk

membaca berat badan pengguna, sedangkan sensor ultrasonik digunakan untuk membaca tinggi badan pengguna. Kedua data tersebut dikirimkan ke mikrokontroler untuk dilakukan pembacaan awal dan perhitungan BMI. Setelah data berat badan, tinggi badan, dan BMI diperoleh, mikrokontroler mengirimkan data tersebut ke backend agar dapat diproses lebih lanjut.

Backend berperan sebagai pusat pengolahan data dalam sistem. Data yang diterima dari mikrokontroler akan disimpan ke database dan digunakan untuk proses klasifikasi BMI, penentuan target berat badan ideal, estimasi durasi program, pengolahan rekomendasi tugas harian, pelaporan aktivitas, serta monitoring BMI mingguan. Selain menerima data dari mikrokontroler, backend juga mengolah data yang dimasukkan pengguna melalui website, seperti laporan aktivitas harian dan data pendukung lainnya.

Website berfungsi sebagai antarmuka pengguna untuk menampilkan hasil pengukuran dan menerima input dari pengguna. Melalui website, pengguna dapat melihat berat badan, tinggi badan, nilai BMI, kategori BMI, target berat badan ideal, rekomendasi tugas harian, laporan aktivitas, dan progres BMI mingguan. Data yang dimasukkan melalui website akan dikirim ke backend, disimpan ke database, kemudian ditampilkan kembali sebagai informasi yang dapat digunakan pengguna untuk memantau perkembangan menuju BMI ideal.

3.4.3 Flowchart Sistem



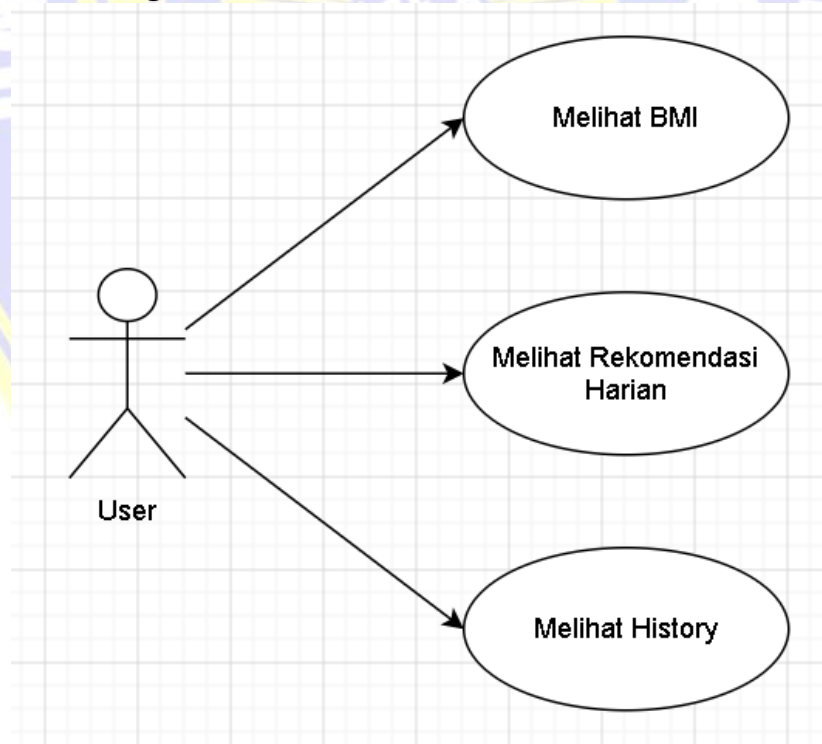
Tabel 3. 3 Flowchart Sistem

Flowchart sistem menggambarkan alur kerja sistem Smart BMI mulai dari pengguna masuk ke sistem hingga proses monitoring perkembangan BMI. Proses diawali dengan pengguna melakukan login, kemudian sistem melakukan verifikasi akun. Apabila verifikasi berhasil, pengguna dapat masuk ke sistem dan melakukan pengukuran tinggi badan serta berat badan melalui alat Smart BMI. Data hasil pengukuran kemudian dikirimkan ke sistem untuk divalidasi. Jika data belum lengkap, pengguna diminta melakukan pengukuran ulang. Namun, apabila data sudah lengkap, sistem melanjutkan proses ke tahap penentuan target berat badan ideal.

Setelah data pengukuran dinyatakan lengkap, sistem menentukan target berat badan ideal berdasarkan hasil BMI pengguna. Selanjutnya, sistem menghitung selisih berat badan antara berat badan aktual dan target berat badan ideal, kemudian menentukan estimasi durasi program yang dibutuhkan. Berdasarkan hasil tersebut, sistem memberikan tugas harian kepada pengguna. Pengguna kemudian menjalankan tugas harian yang diberikan dan melaporkan pelaksanaannya melalui sistem. Laporan tersebut digunakan untuk menghitung skor kepatuhan pengguna terhadap program yang dijalankan.

Tahap berikutnya adalah evaluasi mingguan. Sistem memeriksa apakah sudah waktunya dilakukan evaluasi mingguan. Jika belum, sistem tetap melanjutkan pemberian tugas harian sesuai program. Jika sudah waktunya evaluasi, pengguna melakukan pengukuran ulang tinggi badan dan berat badan. Hasil pengukuran ulang tersebut dibandingkan dengan data sebelumnya untuk mengetahui perkembangan pengguna. Setelah itu, sistem menampilkan progres pengguna menuju berat badan ideal.

3.4.4 Use Case Diagram



Tabel 3. 4 Use Case Diagram

3.4.5 Use Case Deskripsi

Tabel 3. 5 Use Case Deskripsi Melihat BMI

Nama Use Case	Melihat BMI	
ID Use Case	UC-001	
Aktor	User	
Deskripsi	User melakukan pengukuran berat badan dan tinggi badan menggunakan sistem Smart BMI untuk memperoleh nilai BMI beserta kategori BMI berdasarkan standar Kementerian Kesehatan Republik Indonesia. Hasil pengukuran disimpan ke dalam database sebagai dasar penyusunan rekomendasi aktivitas harian dan evaluasi perkembangan BMI.	
Normal Course	Aktor	Sistem
	1. User Membuka Menu Pengukuran 2. User Menekan Tombol Pengukuran 3. User Melakukan Pengukuran	1. Sistem menampilkan halaman Pengukuran. 2. Sistem mengirimkan perintah kepada ESP32 untuk memulai proses pengukuran.. 3. Sensor Loadcell dan HC-SR04 mengukur berat badan dan tinggi badan, dan diolah ESP32, kemudian mengirimkan data hasil pengukuran ke server. 4. Sistem menghitung nilai BMI dan menentukan kategori BMI berdasarkan standar Kemenkes RI. 5. Sistem menyimpan hasil pengukuran ke database 6. Sistem menampilkan hasil pengukuran berupa berat badan, tinggi badan, nilai BMI, dan kategori BMI.
Pre-Condition	1. User telah berhasil login ke dalam sistem 2. ESP32 terhubung dengan sistem	
Post-Condition	1. User berhasil login dan mendapatkan akses ke sistem. 2. User dapat melanjutkan ke use case lainnya sesuai hak akses.	

Tabel 3. 6 Use Case Deskripsi Melihat Rekomendasi Aktivitas Harian

Nama Use Case	Melihat Rekomendasi Aktivitas Harian	
ID Use Case	UC-002	
Aktor	User	
Deskripsi	User melihat rekomendasi aktivitas harian yang dihasilkan berdasarkan hasil pengukuran BMI, target berat badan, target tinggi badan, serta durasi program. Rekomendasi dihasilkan menggunakan Gemini API sesuai dengan kondisi pengguna.	
Normal Course	Aktor	Sistem
	1. User membuka menu rekomendasi aktivitas harian	1. Sistem memeriksa apakah data hasil pengukuran BMI tersedia 2. Jika data pengukuran belum tersedia, sistem meminta user melakukan pengukuran terlebih dahulu. 3. Sistem menyimpan data program pengguna. 4. Sistem mengambil data profil pengguna, hasil BMI, target berat badan, target tinggi badan, dan durasi program. 5. Sistem mengirimkan data ke Gemini API untuk menghasilkan rekomendasi aktivitas harian. 6. Sistem menyimpan hasil rekomendasi ke database. 7. Sistem menampilkan rekomendasi aktivitas harian kepada user.
Pre-Condition	1. User telah berhasil login ke dalam sistem. 2. User telah memiliki hasil pengukuran BMI	

Post-Condition	1. 1. Data Program berhasil disimpan 2. rekomendasi aktivitas harian berhasil ditampilkan kepada user
----------------	--

Tabel 3. 7 Use Case Melihat Histori/Evaluasi Mingguan

Nama Use Case	Melihat Histori/Evaluasi Mingguan	
ID Use Case	UC-003	
Aktor	User	
Deskripsi	User melihat riwayat hasil pengukuran berat badan, tinggi badan, nilai BMI, kategori BMI, serta perkembangan program yang telah dijalankan. Informasi tersebut digunakan untuk memantau perubahan kondisi pengguna berdasarkan data pengukuran yang tersimpan pada sistem.	
Normal Course	Aktor	Sistem
	1. User membuka menu History/Evaluasi Mingguan	1. Sistem menerima permintaan untuk menampilkan riwayat pengukuran. 2. Sistem mengambil data riwayat pengukuran dari database berdasarkan akun user. 3. Sistem mengurutkan data berdasarkan tanggal pengukuran. 4. Sistem menampilkan riwayat pengukuran yang meliputi tanggal, berat badan, tinggi badan, nilai BMI, kategori BMI, dan perkembangan program.
Pre-Condition	1. User telah berhasil login ke dalam sistem. 2. User telah memiliki hasil pengukuran BMI	

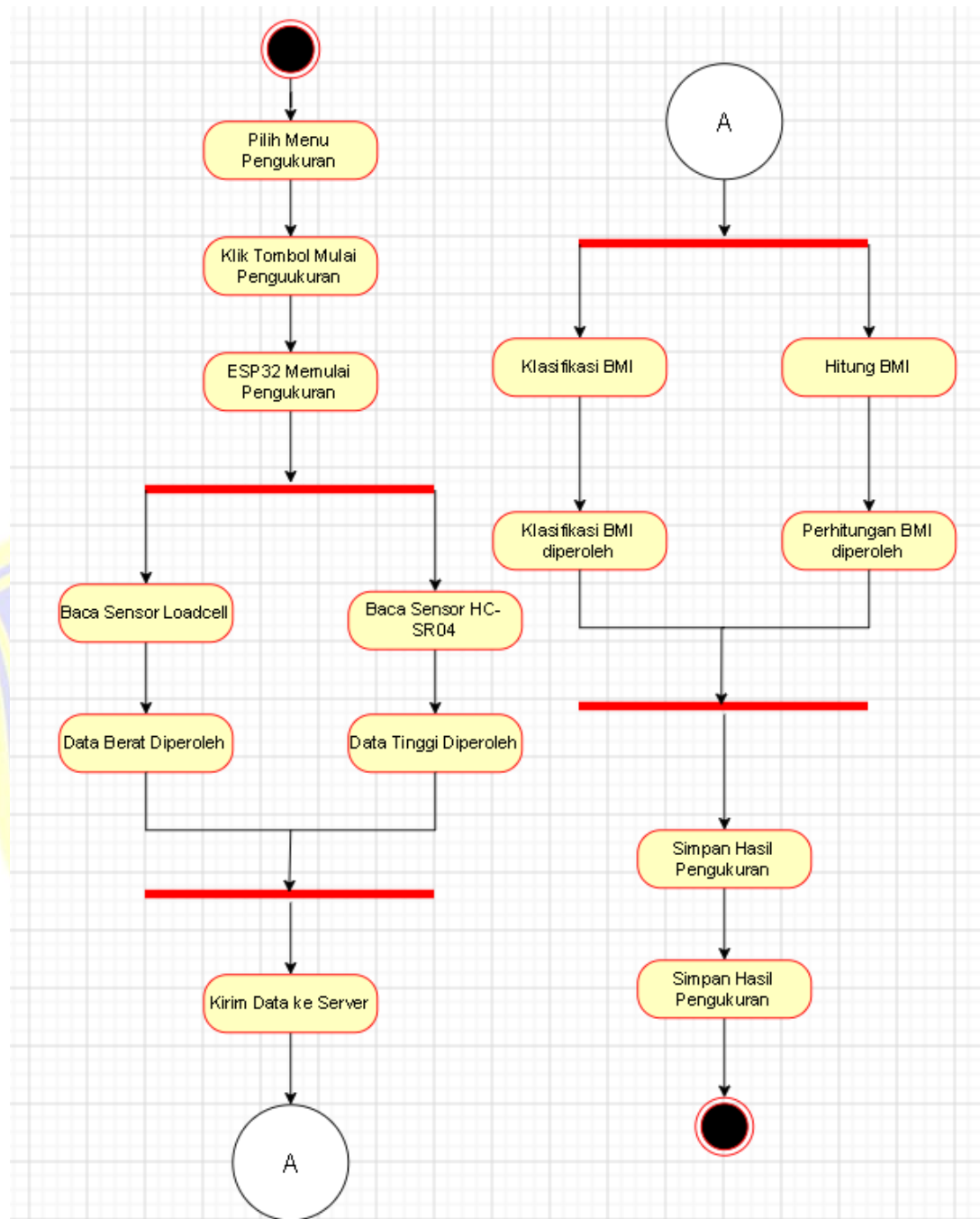
Post-Condition

1. 1. Data Program berhasil disimpan
2. rekomendasi aktivitas harian berhasil ditampilkan kepada user



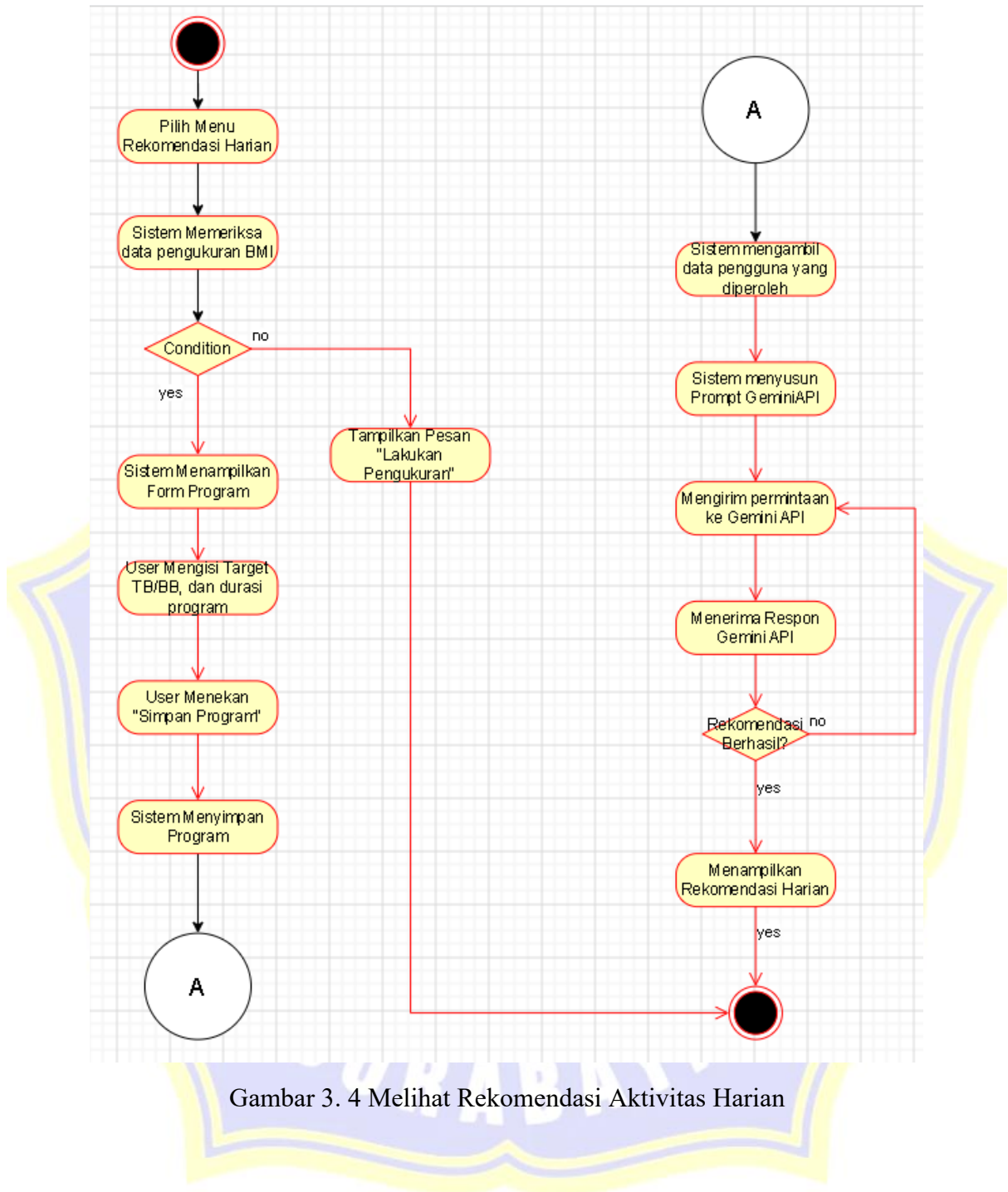
3.4.6 Activity Diagram

1. Melihat BMI



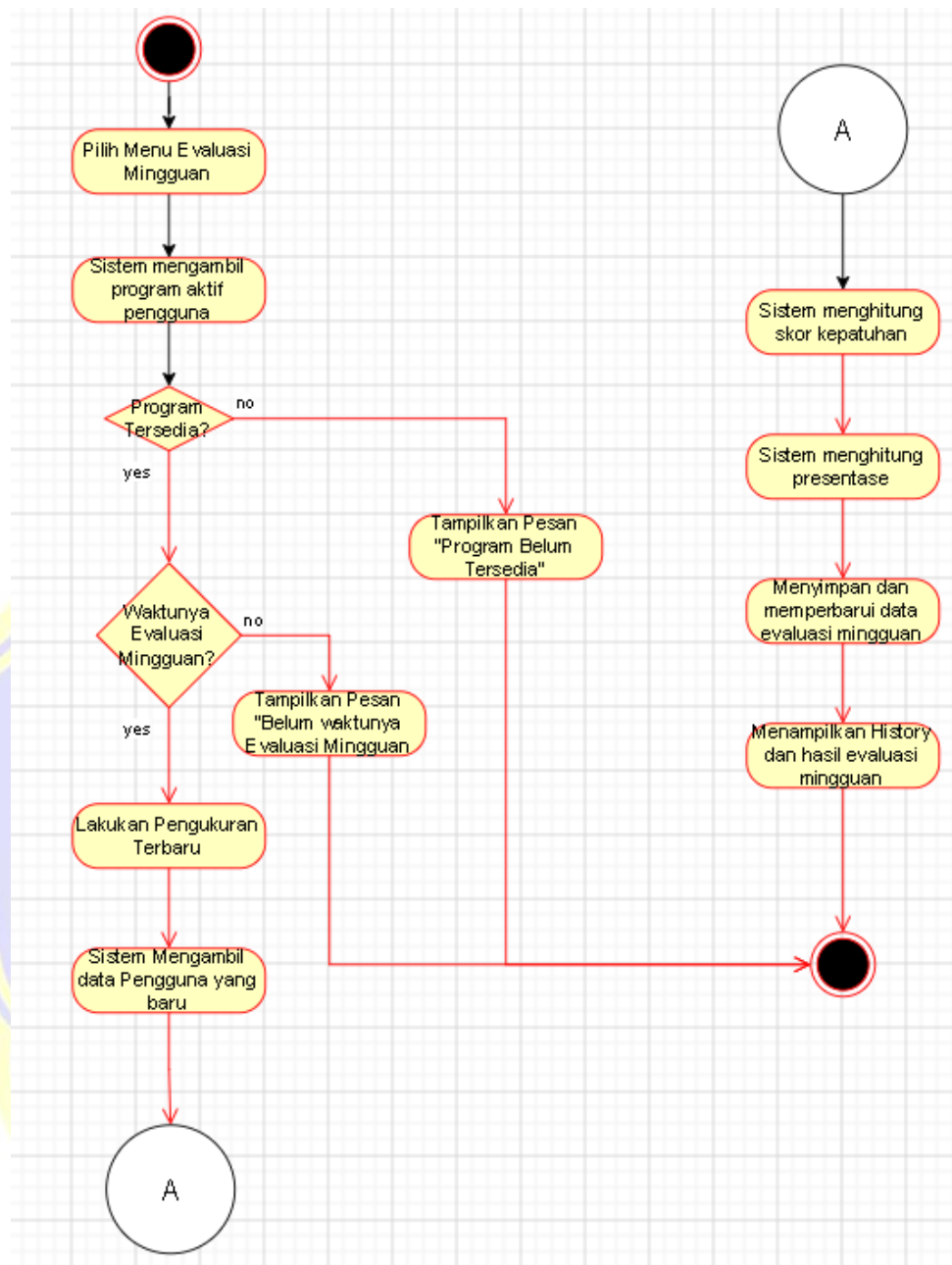
Gambar 3. 3 Activity Diagram Melihat BMI

2. Melihat Aktivitas Harian



Gambar 3. 4 Melihat Rekomendasi Aktivitas Harian

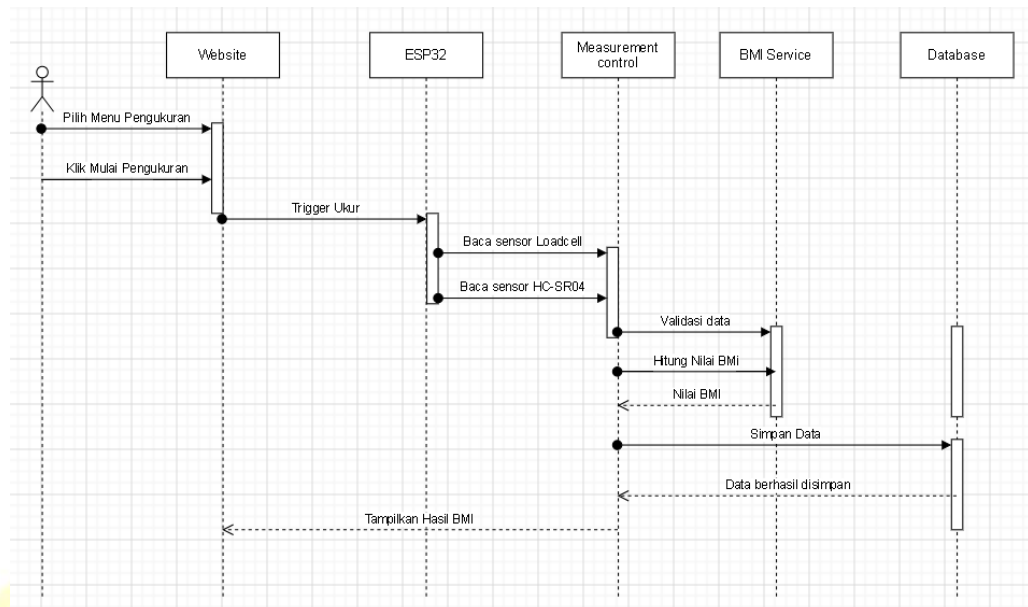
3. Melihat History



Gambar 3. 5 Activity Diagram Melihat History

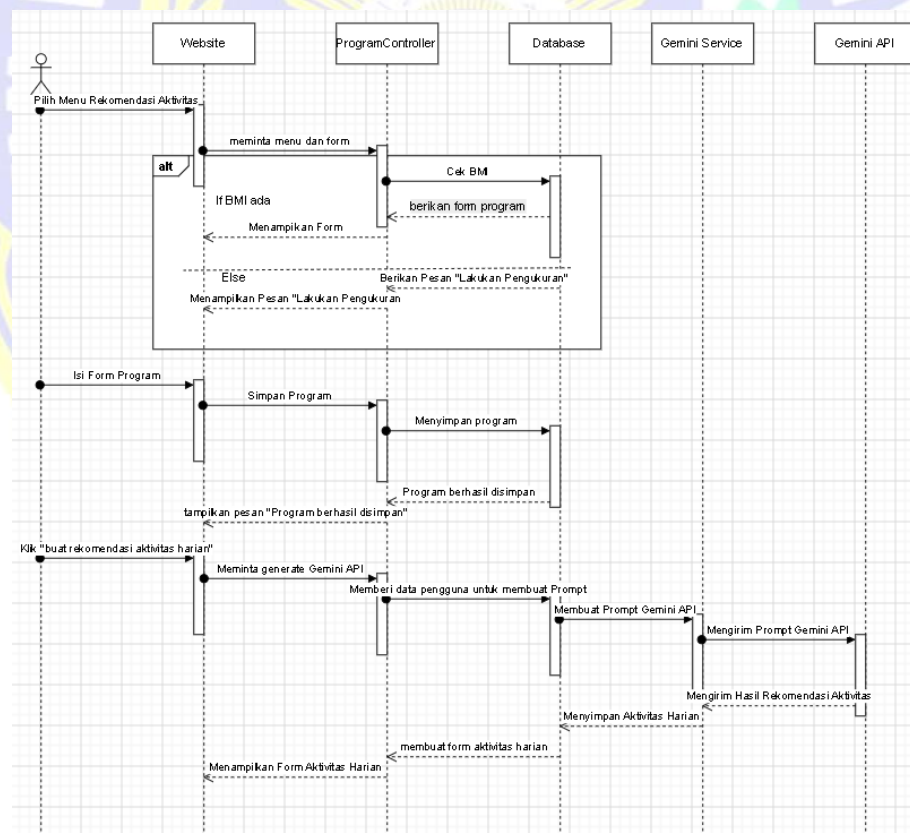
3.4.7 Sequence Diagram

1. Melihat BMI



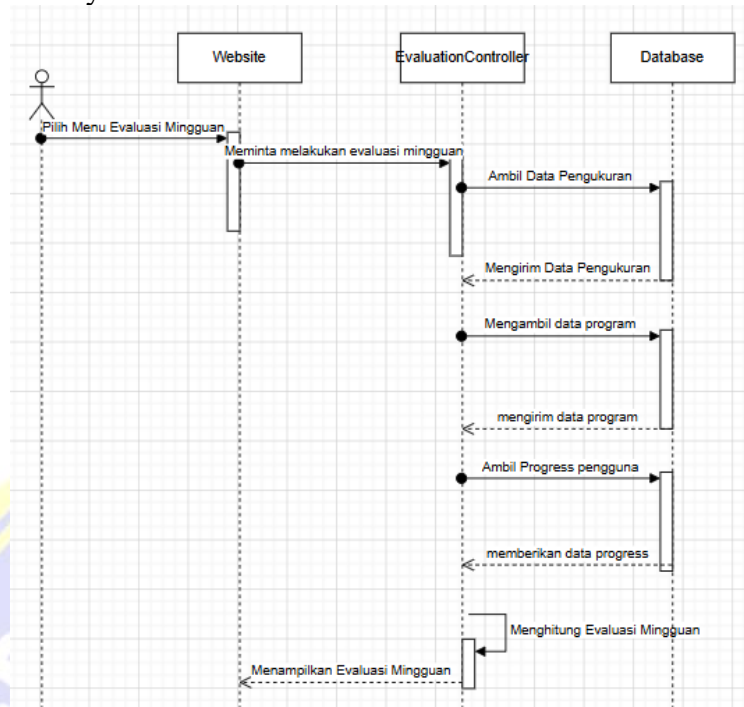
Gambar 3. 6 Sequence Diagram Melihat BMI

2. Melihat Aktivitas Harian



Gambar 3. 7 Sequence Diagram Melihat Rekomendasi Aktivitas Harian

3. Melihat History



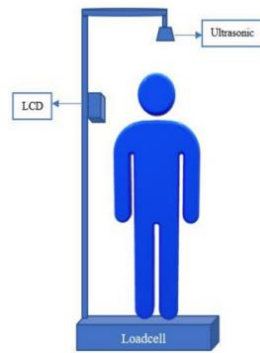
Gambar 3. 8 Sequence Diagram Melihat History

3.4.8 Rancangan Perangkat Keras (Hardware)

Sistem ini terdiri dari sensor berat, sensor tinggi, mikrokontroler, modul pendukung, dan unit tampilan. Sensor sel beban mengukur berat Badan, sedangkan sensor ultrasonik mengukur tinggi Badan. Data dari sensor sel beban diproses terlebih dahulu oleh modul konverter sebelum mencapai mikrokontroler, sedangkan data dari sensor ultrasonik langsung dikirim ke mikrokontroler.

1. Desain Alat BMI

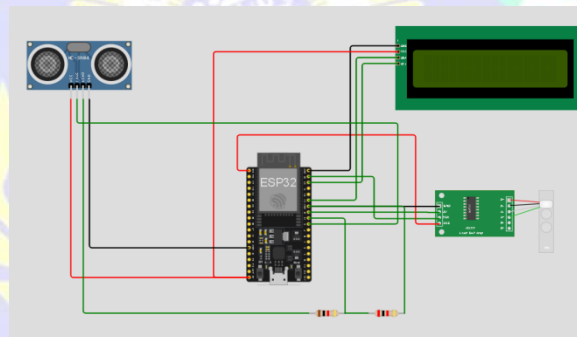
Mikroprosesor berfungsi sebagai pusat kendali sistem, mengumpulkan data dari semua sensor, melakukan pemrosesan awal, dan menampilkan hasil pengukuran pada layar LCD sebagai informasi real-time di perangkat.



Gambar 3. 9 Desain Alat BMI

Gambar 3.9 menunjukkan desain fisik alat pengukuran BMI. Sensor ditempatkan pada bagian atas dan bawah alat untuk mendukung proses pengukuran tinggi badan dan berat badan. Seluruh komponen sensor terhubung ke mikrokontroler yang berfungsi sebagai pusat pengendali sistem. Desain alat ini memungkinkan proses pengukuran dilakukan secara terintegrasi dalam satu perangkat.

1. Rangkaian Mikrokontroler

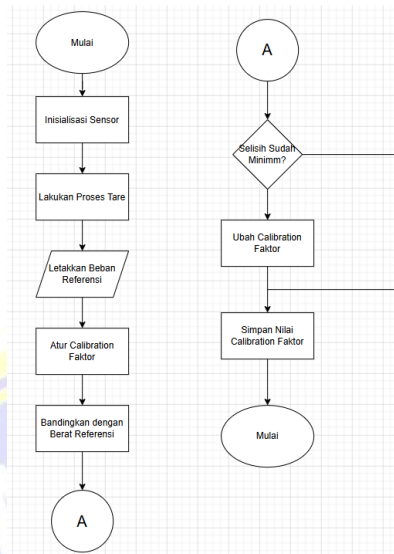


Gambar 3. 10 Rangkaian Mikrokontroler

Gambar 3.10 memperlihatkan rangkaian mikrokontroler yang digunakan dalam sistem pengukuran BMI. Sensor loadcell terhubung ke mikrokontroler melalui modul konverter untuk mengubah sinyal analog menjadi data digital. Sensor ultrasonik terhubung langsung ke pin input mikrokontroler untuk membaca jarak pengukuran tinggi badan. LCD digunakan sebagai media tampilan hasil pengukuran dan terhubung langsung ke mikrokontroler. Rangkaian ini memungkinkan seluruh komponen bekerja secara terintegrasi dalam proses pengukuran BMI.

3.4.8.3 Kalibrasi Sensor

1. Sensor Loadcell



Gambar 3. 11 Flowchart Kalibrasi Sensor Loadcell

Pada Gambar 3.11 kalibrasi sensor Loadcell dilakukan dengan memberikan beban yang telah diketahui nilainya sebagai acuan pengukuran. Sebelum proses kalibrasi dimulai, sensor diatur pada kondisi nol (tare) untuk menghilangkan pengaruh berat awal. Selanjutnya nilai calibration factor diubah secara bertahap dan dilakukan pengujian berulang hingga diperoleh hasil pengukuran yang paling mendekati berat referensi. Berdasarkan proses kalibrasi tersebut diperoleh nilai calibration factor sebesar 12750,0, yang kemudian digunakan pada seluruh proses pengujian dalam penelitian ini.

2. Sensor HC-SR04

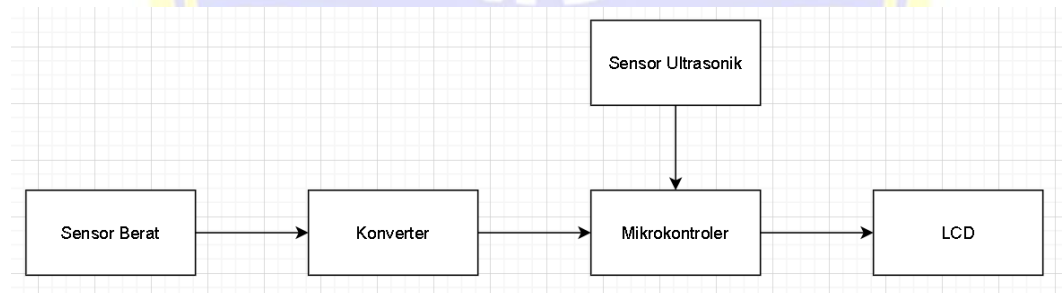


Gambar 3. 12 Flowchart Kalibrasi Sensor HC-SR04

Pada gambar 3.12 kalibrasi sensor HC-SR04 dilakukan dengan mengukur jarak antara posisi sensor ultrasonik dan titik pijakan pengguna menggunakan meteran sebagai alat ukur referensi. Hasil pengukuran menunjukkan tinggi alat sebesar 212 cm, yang selanjutnya digunakan sebagai nilai konstanta pada proses perhitungan tinggi badan. Tinggi badan pengguna dihitung dengan mengurangi nilai konstanta tersebut terhadap jarak yang dibaca oleh sensor HC-SR04 sehingga diperoleh hasil pengukuran tinggi badan secara otomatis

3.4.9 Alur Kerja Sistem Pengukuran Antropometri

Alur kerja sistem pengukuran antropometri balita menggambarkan tahapan proses kerja alat mulai dari pembacaan sensor hingga penampilan hasil pengukuran. Alur ini bertujuan untuk memastikan bahwa proses pengukuran berat badan dan tinggi badan pengguna berjalan secara terstruktur.



Gambar 3. 13 Alur Kerja Pengukuran Antropometri

Gambar 3.13 menunjukkan flowchart kerja alat pengukuran antropometri balita. Proses dimulai dari pembacaan data berat badan oleh sensor berat dan data tinggi badan oleh sensor ultrasonik. Data berat badan terlebih dahulu diproses melalui modul konverter sebelum diterima oleh mikrokontroler. Selanjutnya, mikrokontroler memproses data hasil pengukuran dan menampilkan informasi berat badan dan tinggi badan pada LCD sebagai keluaran awal sistem.

3.4.10 Rancangan Sistem Software

Rancangan perangkat lunak pada sistem Smart BMI berfungsi sebagai penghubung antara perangkat pengukuran, backend, database, dan antarmuka pengguna. Perangkat lunak dirancang untuk menerima data berat badan, tinggi badan, dan BMI dari mikrokontroler melalui API, kemudian mengolah data tersebut untuk proses klasifikasi BMI, penentuan target berat badan ideal, estimasi durasi program, rekomendasi tugas harian, pelaporan aktivitas, serta monitoring BMI mingguan.

Selain sebagai media pengolahan data, perangkat lunak juga berfungsi untuk mengatur hak akses pengguna, menyimpan histori pengukuran, menampilkan hasil pengolahan data, serta mendukung proses evaluasi perkembangan pengguna. Dengan demikian, perangkat lunak menjadi komponen utama yang mengintegrasikan hasil pengukuran otomatis dengan fitur monitoring dan rekomendasi pada sistem Smart BMI.

3.4.10.1 Framework dan Teknologi Pengembangan

Perangkat lunak sistem Smart BMI dikembangkan menggunakan framework Laravel karena mendukung pengembangan sistem berbasis web yang terstruktur, memiliki fitur routing, validasi, middleware, autentikasi, serta pengelolaan database. Laravel digunakan sebagai backend utama untuk menerima data dari mikrokontroler, mengolah data BMI, mengatur hak akses pengguna, serta menjalankan fitur rekomendasi, pelaporan aktivitas, dan monitoring BMI mingguan.

Basis data yang digunakan adalah MySQL yang dijalankan melalui XAMPP. MySQL digunakan untuk menyimpan data pengguna, hasil pengukuran,

program, tugas harian, laporan aktivitas, dan progres BMI mingguan. Sistem juga menggunakan API sebagai penghubung antara ESP32 dan backend, sedangkan antarmuka pengguna dapat dikembangkan menggunakan Blade atau Livewire. Untuk menampilkan grafik perkembangan berat badan dan BMI, sistem dapat memanfaatkan Chart.js agar progres pengguna lebih mudah dipahami

3.4.10.2 Hak Akses Pengguna

Hak akses pengguna pada sistem Smart BMI dibagi menjadi dua aktor utama, yaitu Admin dan User. Pembagian hak akses ini bertujuan agar setiap aktor hanya dapat menggunakan fitur sesuai dengan perannya. Admin memiliki akses untuk mengelola data user, seperti melihat, menambahkan, memperbarui, dan menghapus data user. Selain itu, Admin juga dapat memantau data pengukuran, laporan aktivitas harian, serta perkembangan BMI pengguna yang tersimpan dalam sistem.

User memiliki hak akses untuk menggunakan fitur yang berkaitan dengan data dirinya sendiri. User dapat melihat data diri, melakukan pengukuran tinggi badan dan berat badan melalui alat Smart BMI, melihat hasil BMI, melaporkan aktivitas harian, melakukan evaluasi mingguan, serta melihat progres perkembangan BMI. User tidak memiliki akses untuk menambah, menghapus, atau mengubah data user lain. Pembaruan data pada sisi User terjadi melalui aktivitas penggunaan sistem, seperti pengukuran TB dan BB, pelaporan aktivitas harian, serta evaluasi mingguan.

3.4.10.3 Rancangan Backend Sistem

Backend pada sistem Smart BMI dirancang sebagai pusat pengolahan data dan pengatur logika utama sistem. Backend menerima data berat badan, tinggi badan, dan BMI dari ESP32 melalui API, kemudian melakukan validasi data sebelum disimpan ke database MySQL. Setelah data pengukuran tersimpan, backend mengolah data tersebut untuk melakukan klasifikasi BMI berdasarkan standar Kemenkes, menentukan target berat badan ideal, menghitung selisih berat badan, serta menentukan estimasi durasi program pengguna.

Selain mengolah data pengukuran, backend juga mengatur proses rekomendasi tugas harian, pelaporan aktivitas, dan monitoring BMI mingguan. Rekomendasi tugas harian diberikan berdasarkan kategori BMI, target berat badan

ideal, selisih berat badan, estimasi durasi program, dan progres pengguna. Data laporan aktivitas harian yang dikirim oleh user akan diproses oleh backend untuk menghitung skor kepatuhan. Selanjutnya, pada evaluasi mingguan, backend membandingkan data pengukuran terbaru dengan data sebelumnya untuk mengetahui perubahan berat badan, perubahan BMI, dan progres pengguna terhadap target.

Dalam rancangan ini, backend juga bertanggung jawab mengatur hak akses antara Admin dan User. Admin dapat mengelola data user dan memantau perkembangan pengguna, sedangkan User hanya dapat mengakses data miliknya sendiri. Dengan demikian, backend tidak hanya berfungsi sebagai penghubung antara database dan antarmuka sistem, tetapi juga sebagai pusat validasi, pengolahan data, pengaturan akses, dan pengendali proses utama dalam sistem Smart BMI.

3.4.10.4 Rancangan Database

Rancangan database pada sistem Smart BMI menggunakan database relasional MySQL yang dijalankan melalui XAMPP. Database digunakan untuk menyimpan seluruh data yang dibutuhkan sistem, mulai dari data pengguna, hasil pengukuran berat badan dan tinggi badan, nilai BMI, program pengguna, tugas harian, laporan aktivitas, progres mingguan, hingga log interaksi dengan Gemini API apabila fitur tersebut digunakan. Penggunaan database relasional bertujuan agar setiap data dapat tersimpan secara terstruktur dan saling terhubung sesuai kebutuhan sistem.

Tabel utama yang digunakan dalam sistem Smart BMI meliputi users, measurement_sessions, measurements, programs, task_library, daily_tasks, activity_reports, progress_logs, dan gemini_logs. Tabel users digunakan untuk menyimpan data akun Admin dan User. Tabel measurement_sessions digunakan untuk mengatur sesi pengukuran antara sistem dan mikrokontroler, sedangkan tabel measurements menyimpan histori hasil pengukuran berat badan, tinggi badan, BMI, kategori BMI, serta batas berat badan ideal. Tabel programs menyimpan data program pengguna, seperti target berat badan, jenis program, selisih berat badan, estimasi durasi, dan status program.

Selain itu, tabel `task_library` digunakan sebagai pustaka tugas harian yang mencakup aktivitas fisik, pola makan, pola tidur, dan kebiasaan sehat. Tabel `daily_tasks` menyimpan tugas harian yang diberikan kepada pengguna, sedangkan tabel `activity_reports` menyimpan laporan aktivitas harian yang diisi oleh pengguna. Tabel `progress_logs` digunakan untuk menyimpan hasil evaluasi mingguan, seperti berat badan terbaru, BMI, skor kepatuhan, dan progres terhadap target. Adapun tabel `gemini_logs` digunakan untuk menyimpan riwayat permintaan dan respons Gemini API jika sistem menggunakan bantuan AI dalam menghasilkan variasi rekomendasi tugas harian.

Relasi utama dalam database ini adalah satu user dapat memiliki banyak data pengukuran, sesi pengukuran, program, tugas harian, laporan aktivitas, dan catatan progres mingguan. Satu program dapat memiliki banyak tugas harian dan progress log. Satu data pada `task_library` dapat digunakan pada banyak `daily_tasks`, sedangkan satu `daily_task` hanya memiliki satu `activity_report`. Dengan rancangan database tersebut, sistem Smart BMI dapat mengelola data pengukuran, rekomendasi, pelaporan aktivitas, dan monitoring BMI mingguan secara terstruktur dan saling terintegrasi.

3.4.10.5 Rancangan Antarmuka Sistem

Rancangan antarmuka sistem Smart BMI dibuat untuk memudahkan Admin dan User dalam mengakses fitur sesuai dengan hak akses masing-masing. Antarmuka sistem dirancang sederhana, responsif, dan mudah dipahami agar pengguna dapat melihat hasil pengukuran, rekomendasi tugas harian, laporan aktivitas, serta perkembangan BMI secara jelas. Setiap tampilan disusun berdasarkan kebutuhan utama sistem, sehingga informasi yang ditampilkan tidak hanya berupa data mentah, tetapi juga hasil pengolahan seperti kategori BMI, target berat badan ideal, estimasi durasi program, skor kepatuhan, dan progres mingguan.

Pada sisi User, antarmuka sistem terdiri dari halaman dashboard, data diri, hasil pengukuran, tugas harian, laporan aktivitas harian, evaluasi mingguan, dan progres BMI. Dashboard User menampilkan informasi utama seperti berat badan terbaru, tinggi badan terbaru, nilai BMI, kategori BMI, target berat badan ideal, estimasi durasi program, tugas harian, status laporan aktivitas, dan grafik

perkembangan BMI. Sementara itu, pada sisi Admin, antarmuka sistem digunakan untuk mengelola data user, melihat data pengukuran, memperbarui data user, menghapus data user, serta memantau laporan aktivitas dan progres pengguna

3.4.10.6 Rancangan API Mikrokontroler

Rancangan API mikrokontroler digunakan sebagai penghubung antara alat Smart BMI berbasis ESP32 dan sistem backend. API ini berfungsi untuk menerima data hasil pengukuran dari mikrokontroler, seperti berat badan, tinggi badan, dan nilai BMI. Data yang dikirim oleh ESP32 kemudian diterima oleh backend, divalidasi, disimpan ke dalam database, dan digunakan untuk proses pengolahan lanjutan seperti klasifikasi BMI, penentuan target berat badan ideal, estimasi durasi program, rekomendasi tugas harian, serta monitoring BMI mingguan.

Pada sistem Smart BMI, proses pengukuran dapat dimulai melalui sesi pengukuran yang dibuat oleh sistem. Ketika pengguna menekan tombol mulai pengukuran pada antarmuka sistem, backend membuat data sesi pengukuran dengan status pending. ESP32 kemudian membaca status sesi tersebut, melakukan pengukuran berat badan menggunakan loadcell dan tinggi badan menggunakan sensor ultrasonik, lalu mengirimkan hasil pengukuran ke API. Setelah data diterima, sistem memperbarui status sesi menjadi completed apabila pengukuran berhasil, atau failed apabila data tidak valid.

3.4.10.7 Rancangan Rekomendasi dan Monitoring

Rancangan rekomendasi pada sistem Smart BMI digunakan untuk memberikan tugas harian kepada pengguna berdasarkan hasil pengolahan BMI. Rekomendasi disusun berdasarkan kategori BMI, target berat badan ideal, jenis program, selisih berat badan, estimasi durasi program, dan skor kepatuhan pengguna. Tugas harian yang diberikan mencakup aktivitas fisik, pola makan, pola tidur, dan kebiasaan sehat. Apabila sistem menggunakan Gemini API, maka layanan tersebut berperan sebagai pendukung untuk menghasilkan variasi rekomendasi tugas harian, sedangkan keputusan utama tetap ditentukan oleh sistem berbasis aturan.

Setiap tugas harian yang diberikan kepada pengguna disimpan dalam tabel `daily_tasks`, sedangkan daftar tugas dasar disimpan dalam tabel `task_library`. Setelah menjalankan tugas, pengguna melaporkan aktivitas harian melalui sistem.

Data laporan tersebut disimpan dalam tabel `activity_reports` dan digunakan untuk menghitung skor kepatuhan pengguna. Skor kepatuhan dihitung berdasarkan perbandingan antara jumlah tugas yang diselesaikan dengan jumlah tugas yang diberikan. Hasil skor tersebut menjadi salah satu indikator dalam proses monitoring perkembangan pengguna.

Monitoring BMI dilakukan secara mingguan melalui pengukuran ulang berat badan dan tinggi badan menggunakan alat Smart BMI. Data hasil pengukuran terbaru dibandingkan dengan data pengukuran sebelumnya untuk mengetahui perubahan berat badan, perubahan BMI, dan progres terhadap target berat badan ideal. Hasil monitoring disimpan dalam tabel `progress_logs` dan ditampilkan kepada pengguna dalam bentuk informasi progres atau grafik perkembangan. Dengan rancangan ini, sistem tidak hanya memberikan rekomendasi hari

3.5 Rencana Evaluasi

Rencana evaluasi dalam penelitian ini dilakukan untuk menilai apakah sistem Smart BMI berbasis mikrokontroler dapat bekerja sesuai dengan tujuan yang telah ditetapkan. Evaluasi difokuskan pada komponen yang dikembangkan dalam penelitian, yaitu alat pengukuran berat badan dan tinggi badan, proses perhitungan serta klasifikasi BMI, penentuan target berat badan ideal, estimasi durasi program, rekomendasi tugas harian, pelaporan aktivitas, monitoring BMI mingguan, serta kemudahan penggunaan sistem oleh pengguna. Setiap bagian dievaluasi menggunakan metode pengujian yang sesuai agar hasil pengujian dapat menunjukkan tingkat keberhasilan sistem secara terukur.

3.5.1 Evaluasi Perangkat Keras

Evaluasi perangkat keras dilakukan untuk mengetahui tingkat akurasi sensor loadcell dalam mengukur berat badan dan sensor ultrasonik dalam mengukur tinggi badan. Pengujian dilakukan dengan membandingkan hasil pembacaan sensor dengan alat ukur pembanding, yaitu timbangan digital untuk berat badan dan meteran/stadiometer untuk tinggi badan. Setiap pengujian dilakukan beberapa kali pada objek atau pengguna yang sama untuk mengetahui kestabilan hasil pembacaan sensor.

Metrik yang digunakan pada evaluasi perangkat keras adalah persentase error, Mean Absolute Error (MAE), dan Root Mean Square Error (RMSE).

Persentase Error

$$\text{Error}(\%) = \left| \frac{X_{\text{sensor}} - X_{\text{referensi}}}{X_{\text{referensi}}} \right| \times 100\% \quad (10)$$

X_{sensor} = Hasil pengukuran sensor

$X_{\text{referensi}}$ = hasil pengukuran alat pembanding

MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |X_i - Y_i| \quad (11)$$

Keterangan

(X_i) = hasil pengukuran sensor

(Y_i) = hasil pengukuran alat pembanding

(n) = jumlah data pengujian

RMSE:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - Y_i)^2} \quad (12)$$

Keterangan:

(X_i) = hasil pengukuran sensor

(Y_i) = hasil pengukuran alat pembanding

(n) = jumlah data pengujian

Pengujian ini dilakukan untuk mengevaluasi tingkat akurasi sensor dengan membandingkan hasil pengukuran perangkat yang dikembangkan terhadap alat ukur referensi. Evaluasi dilakukan menggunakan parameter persentase error, *Mean Absolute Error* (MAE), dan *Root Mean Square Error* (RMSE) untuk mengetahui tingkat penyimpangan hasil pengukuran. Semakin rendah nilai error, MAE, dan RMSE yang diperoleh, maka semakin tinggi tingkat akurasi sistem pengukuran. Pada penelitian ini, kinerja sensor dinyatakan memenuhi kriteria apabila memiliki

rata-rata persentase error tidak melebihi 5% atau setara dengan tingkat akurasi minimal 95%, mengacu pada standar internasional IEC No. 13B-23 mengenai ketelitian alat ukur sebagaimana telah dijelaskan pada Bab II (Fatimah & Hidayat, 2024). Selain itu, hasil pengujian sensor loadcell juga dibandingkan dengan batas Maximum Permissible Error (MPE) berdasarkan standar OIML R76-1:2006 sesuai kelas akurasi timbangan, sedangkan hasil pengukuran sensor HC-SR04 dievaluasi terhadap spesifikasi akurasi bawaan pabrikan sebesar ± 3 mm pada kondisi pengukuran ideal sebagai referensi pendukung.

3.5.2 Evaluasi White Box Testing

White Box Testing merupakan metode pengujian perangkat lunak yang dilakukan dengan memeriksa struktur internal program, logika algoritma, percabangan, dan jalur eksekusi program. Berbeda dengan Black Box Testing yang berfokus pada kesesuaian keluaran terhadap masukan tanpa memperhatikan proses di dalam program, White Box Testing mengevaluasi apakah setiap bagian logika telah berjalan sesuai dengan rancangan algoritma yang dikembangkan. Melalui pengujian ini, setiap jalur keputusan pada program dapat dianalisis sehingga kesalahan logika maupun percabangan yang tidak sesuai dapat diidentifikasi sejak tahap evaluasi.

Pada penelitian ini, White Box Testing digunakan untuk mengevaluasi logika internal sistem Smart BMI yang dikembangkan. Pengujian tidak difokuskan pada seluruh fitur aplikasi, melainkan pada modul-modul utama yang merepresentasikan proses pengolahan data dan implementasi metode yang digunakan dalam penelitian. Pemilihan modul tersebut didasarkan pada kontribusinya terhadap tujuan penelitian, yaitu menghasilkan sistem yang mampu melakukan pengukuran data pengguna, menghitung nilai Body Mass Index (BMI), mengklasifikasikan status BMI berdasarkan standar Kementerian Kesehatan Republik Indonesia, menentukan target program, menghasilkan rekomendasi aktivitas harian menggunakan Gemini API, serta melakukan evaluasi perkembangan BMI secara mingguan.

Tabel 3. 8 Tabel Rencana Pengujian White Box

Kode	Modul yang Diuji	Tujuan Pengujian
WB-01	Proses Pengukuran Data Pengguna	Memastikan logika proses pengambilan data dari ESP32, Loadcell, dan HC-SR04 hingga data diterima oleh sistem berjalan sesuai rancangan.
WB-02	Proses Perhitungan BMI	Memastikan algoritma perhitungan BMI menghasilkan nilai yang sesuai dengan rumus yang digunakan.
WB-03	Proses Klasifikasi BMI	Memastikan seluruh percabangan klasifikasi BMI berdasarkan threshold Kementerian Kesehatan RI dieksekusi dengan benar.
WB-04	Proses Penentuan Target dan Estimasi Program	Memastikan logika penentuan target pengguna serta estimasi durasi program berjalan sesuai algoritma yang dirancang.
WB-05	Proses Pembangkitan Rekomendasi Gemini API	Memastikan proses penyusunan prompt, pengiriman data ke Gemini API, serta penerimaan dan penyimpanan hasil rekomendasi berjalan sesuai alur sistem.
WB-06	Proses Evaluasi Mingguan	Memastikan logika evaluasi mingguan, perbandingan perkembangan BMI, perhitungan kepatuhan, serta pembaruan rekomendasi ketika target berubah berjalan sesuai rancangan.

Pelaksanaan White Box Testing pada setiap modul dilakukan melalui beberapa tahapan. Tahap pertama adalah mengidentifikasi algoritma yang digunakan pada masing-masing modul. Algoritma tersebut kemudian direpresentasikan dalam bentuk flowchart untuk menggambarkan alur logika program secara visual. Selanjutnya flowchart dikonversi menjadi flow graph yang terdiri atas sejumlah node dan edge sebagai dasar analisis jalur program.

Berdasarkan flow graph yang telah diperoleh, dilakukan perhitungan Cyclomatic Complexity untuk menentukan jumlah jalur independen (independent path) yang harus diuji. Nilai Cyclomatic Complexity dihitung menggunakan Persamaan (13) .

$$V(G) = E - N + 2P \quad (13)$$

Keterangan

$V(G)$ = Cyclomatic Complexity;

E = jumlah edge pada flow graph;

N = jumlah node pada flow graph;

P = jumlah komponen terhubung (*connected component*), yang pada penelitian ini bernilai 1.

Setelah jumlah jalur independen diperoleh, setiap jalur disusun menjadi skenario pengujian (test case) yang mewakili seluruh kemungkinan alur eksekusi program. Selanjutnya pengujian dilakukan dengan mengeksekusi setiap jalur tersebut untuk memastikan seluruh logika program berjalan sesuai dengan rancangan sistem. Hasil White Box Testing kemudian dianalisis berdasarkan keberhasilan setiap jalur independen dalam mengeksekusi proses yang dirancang tanpa menghasilkan kesalahan logika maupun percabangan.

3.5.3 Evaluasi Perhitungan dan Klasifikasi BMI

Evaluasi perhitungan BMI dilakukan untuk memastikan bahwa sistem dapat menghitung nilai BMI dengan benar berdasarkan data berat badan dan tinggi badan pengguna. Pengujian dilakukan dengan membandingkan hasil perhitungan BMI dari sistem dengan hasil perhitungan manual menggunakan rumus BMI. Jika hasil perhitungan sistem sesuai dengan hasil manual, maka fungsi perhitungan BMI dinyatakan berjalan dengan baik.

$$BMI = \frac{BB}{TB_m^2} \quad (14)$$

Keterangan:

(BB) = berat badan dalam kilogram

(TB_m) = tinggi badan dalam meter

Evaluasi klasifikasi BMI dilakukan dengan membandingkan kategori BMI yang dihasilkan sistem dengan kategori BMI berdasarkan standar Kemenkes. Pengujian ini menggunakan beberapa skenario nilai BMI, seperti underweight, normal, overweight, dan obesitas.

Metrik yang digunakan adalah accuracy, karena klasifikasi BMI pada sistem bersifat berbasis aturan, bukan model machine learning.

Accuracy:

$$Accuracy = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Seluruh\ Data\ Uji} \times 100\% \quad (15)$$

Dalam penelitian ini, prediksi dianggap benar apabila kategori BMI yang dihasilkan sistem sama dengan kategori BMI berdasarkan standar Kemenkes.

3.5.4 Evaluasi Rekomendasi Tugas Harian

Evaluasi rekomendasi tugas harian dilakukan untuk memastikan bahwa tugas yang diberikan sistem sesuai dengan kondisi BMI pengguna. Pengujian dilakukan dengan membuat beberapa skenario pengguna, misalnya pengguna underweight, normal, overweight, dan obesitas. Setiap skenario diuji untuk melihat apakah sistem memberikan rekomendasi yang sesuai dengan kategori BMI, arah program, estimasi durasi, dan batas intensitas aktivitas.

Metrik yang dapat digunakan adalah accuracy berbasis skenario uji, yaitu membandingkan jumlah rekomendasi yang sesuai dengan aturan terhadap seluruh skenario yang diuji.

Accuracy rekomendasi

$$Accuracy = \frac{Jumlah\ Rekomendasi\ Sesuai}{Jumlah\ Seluruh\ Skenario\ Uji} \times 100\% \quad (16)$$

:

Jika sistem menggunakan Gemini API, maka evaluasi tambahan dilakukan terhadap validasi output Gemini API. Output dinilai berdasarkan format, kesesuaian kategori tugas, keamanan rekomendasi, dan tidak adanya saran ekstrem seperti diet ketat, olahraga berlebihan, obat pelangsing, suplemen, atau klaim medis.

Untuk penelitian ini, precision dan recall tidak wajib digunakan, karena rekomendasi sistem bukan klasifikasi multi-label berbasis dataset training. Precision dan recall baru relevan jika kamu memiliki dataset ground truth rekomendasi dan ingin menguji apakah sistem berhasil memilih rekomendasi yang benar dari banyak kemungkinan label.

3.5.5 Evaluasi Pelaporan Aktivitas Harian

Evaluasi pelaporan aktivitas harian dilakukan untuk memastikan sistem dapat menerima laporan aktivitas pengguna dan menghitung skor kepatuhan dengan benar. Pengujian dilakukan dengan membandingkan hasil skor kepatuhan dari sistem dengan perhitungan manual berdasarkan jumlah tugas yang diselesaikan dan jumlah tugas yang diberikan.

Rumus skor kepatuhan:

$$SK = \frac{TS}{TD} \times 100\% \quad (17)$$

Keterangan:

SK = Skor Kepatuhan

TS = Jumlah Tugas Selesai

TD = Jumlah Tugas Diberikan

Sistem dinyatakan sesuai apabila skor kepatuhan yang dihitung sistem sama dengan hasil perhitungan manual.

3.5.6 Evaluasi Monitoring BMI Mingguan

Evaluasi monitoring BMI mingguan dilakukan untuk memastikan sistem dapat membandingkan data pengukuran awal dengan data pengukuran terbaru. Pengujian dilakukan dengan memasukkan data pengukuran awal dan data evaluasi mingguan, kemudian sistem menghitung perubahan berat badan, perubahan BMI, dan progres terhadap target berat badan ideal.

Rumus progres target:

$$PT = \frac{BT}{SB} \times 100\% \quad (18)$$

Keterangan:

PT = Presentasi Progres Target

BT = Berat Badan yang telah berubah menuju target

SB == Selisih berat badan awal terhadap target berat badan ideal

Hasil sistem dibandingkan dengan perhitungan manual untuk memastikan bahwa proses monitoring berjalan dengan benar