

BAB III

METODE PENELITIAN

3.1 ALAT DAN BAHAN PENELITIAN

a. Alat

Peralatan yang digunakan pada penelitian ini adalah :

1. Perangkat Keras :

a. Laptop, dengan spesifikasi :

- Intel Core i5 2,60 GHz
- RAM 8 Gb
- SSD 120 Gb
- HDD 320 Gb
- Windows 10 Pro

b. Smartphone, dengan spesifikasi :

- Octa Core 2 GHz
- RAM 4 Gb
- Storage 64 Gb
- Android 11

2. Perangkat Lunak :

- a. Android Studio versi Dolphin 2021.3.1
- b. Figma
- c. YAT Terminal 2.4.1
- d. Apptim 0.9.2

b. Bahan

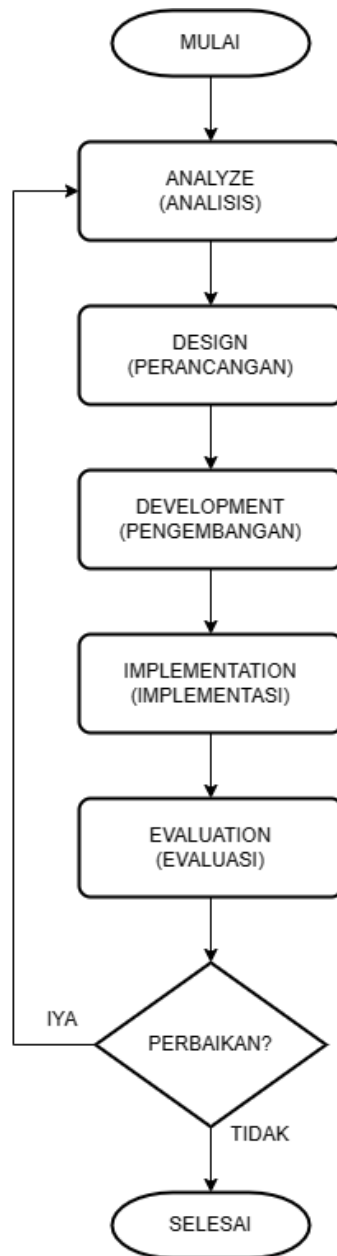
Bahan – bahan yang digunakan untuk melakukan penelitian ini antara lain :

1. Data protokol komunikasi yang digunakan modul monita
2. Data spesifikasi monita

3.2 METODE YANG DIGUNAKAN

Dalam pengembangan aplikasi ini digunakan metode ADDIE. Adapun tahapan dalam metode ADDIE yaitu *Analyze* (Analisis), *Design* (Perancangan),

Development (Pengembangan), *Implementation* (Implementasi), dan *Evaluation* (Evaluasi). Metode ini dipilih karena melakukan pendekatan secara sistematis.



Gambar 3.1 *Flowchart* Penelitian Dengan Metode ADDIE

3.2.1 ANALYZE (ANALISIS)

3.2.1.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data yang dibutuhkan dalam pembuatan sebuah program aplikasi. Dalam penelitian ini teknik pengumpulan data yang dilakukan dengan studi literatur. Metode ini merupakan

teknik pengumpulan data berdasarkan referensi dan pustaka pendukung sebagai bahan acuan yang berhubungan dengan objek yang diteliti dengan cara mempelajari metode, teknik, konsep, teori dari berbagai jurnal terkait dengan informasi dari pembuatan aplikasi yang dibuat pada tahap analisa data. Adapun referensi yang digunakan untuk data pada penelitian ini:

- Ovidiu Constantin Novac, “*Development of a Weather Mobile Application in Android Operating System*”, *Electronics, Computers and Artificial Intelligence (ECAI)*, 2019.
- Arushi Singh, Dr. Shilpi Sharma, “*Implement Android Application For Book Donation*”, *International Conference on Intelligent Engineering and Management (ICIEM)*, 2020, p.138-141.
- F. Perdana, S. Madanijah, I. Ekayanti, “*Pengembangan Media Edukasi Gizi Berbasis Android Dan Website Serta Pengaruhnya Terhadap Perilaku Tentang Gizi*”, *Jurnal Gizi dan Pangan*, 2017, p.169-178.
- R. Sovia, M. Yanto, J. Nursam, “*Perancangan Aplikasi E-News Berbasis Android Dan Web (Studi Kasus RRI Stasiun Padang)*, *Explore*, 2020, p.42-48.

3.2.1.2 Analisis Kebutuhan Pengguna

Penyampaian informasi dan kepuasan pengguna merupakan tujuan utama dari aplikasi konfigurasi berbasis *mobile* ini. Oleh karena itu dibutuhkan spesifikasi kebutuhan pemakai pada umumnya. Ada beberapa hal yang diperhatikan dalam memenuhi kebutuhan pengguna yaitu :

1. *User Friendly* yaitu pemakai lebih memilih aplikasi yang mudah digunakan dan dioperasikan.
2. Tampilan sederhana namun tetap menarik untuk dilihat yakni menggunakan elemen – elemen seperti gambar, teks, dan tombol.
3. Aplikasi yang dihasilkan dapat digunakan untuk mengkonfigurasi modul monita sesuai fungsinya.

3.2.1.3 Analisis Kebutuhan Fungsional

Adapun kebutuhan fungsional dari aplikasi ini supaya dapat berjalan sesuai dengan apa yang diinginkan :

1. Tampilan halaman utama dari aplikasi akan menampilkan tiga menu *navigation* yaitu *information*, *monitor*, dan *data register* serta satu tombol menu *setting* dibagian *toolbar*.
2. Aplikasi dilengkapi dengan *guide tour* agar mempermudah pengguna dalam menggunakan aplikasi.
3. Aplikasi ini nantinya dapat digunakan oleh teknisi maupun konsumen yang memakai produk Monita.

3.2.1.4 Analisis Kebutuhan Sistem Aplikasi

1. Perangkat keras, untuk aplikasi ini memerlukan perangkat keras *smartphone* bersistem operasi android.
2. Perangkat lunak, untuk menjalankan aplikasi ini diperlukan perangkat lunak sistem operasi android dari versi 5.0 : *Lollipop* sampai versi 12.

3.2.2 DESIGN (PERANCANGAN)

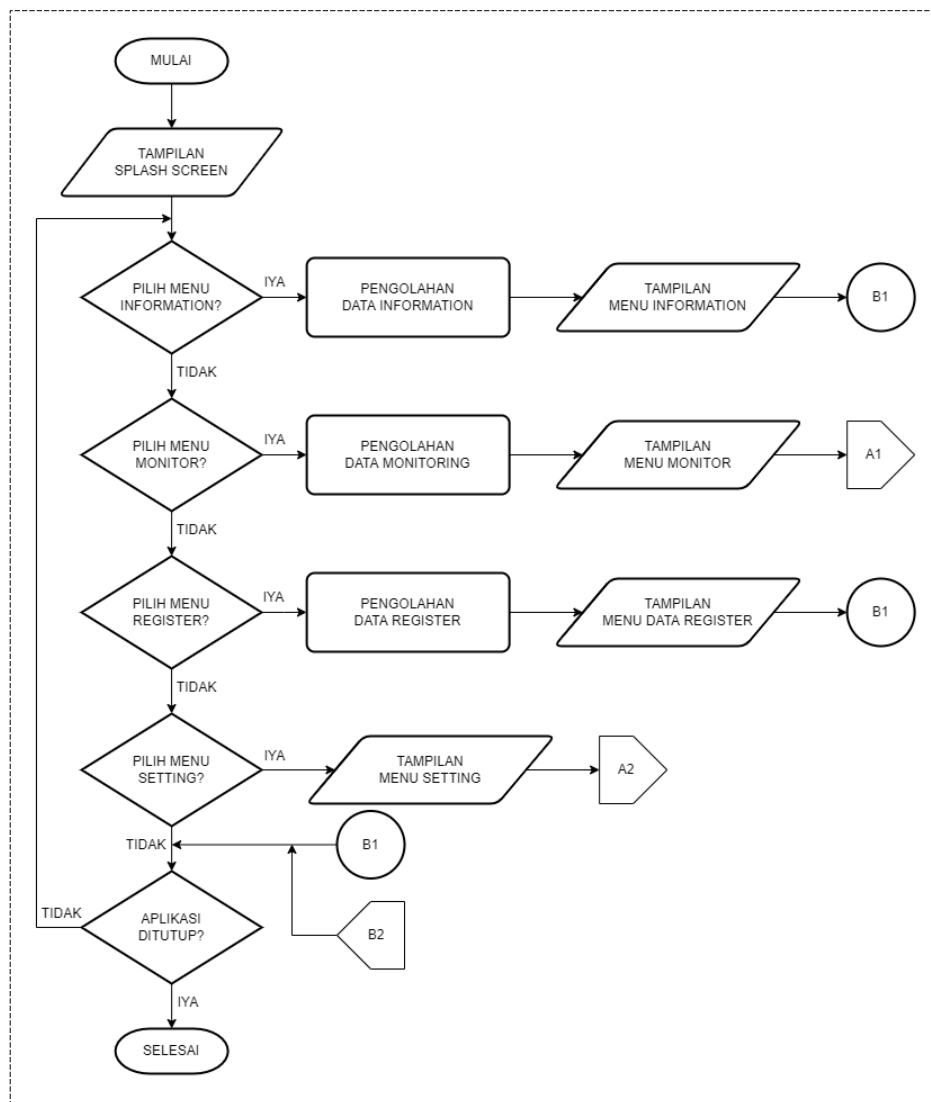
Rancangan aplikasi “Monita *Bluetooth*” akan dimulai dari *Splash Screen*, setelah itu masuk ke menu utama terdapat empat buah menu yaitu menu *information* yang berisi informasi tentang modul monita yang sedang terhubung dengan aplikasi, menu *monitor* yang berisi tabel data dan grafik data *real-time* dari modul monita, *data register* berisi data *register* yang dapat dipilih, dan menu *setting* berisi menu *setting bluetooth*, *setting device information*, *setting communication*, *setting protocol*, dan *setting data source*.

3.2.2.1 Flowchart (Alur Aplikasi)

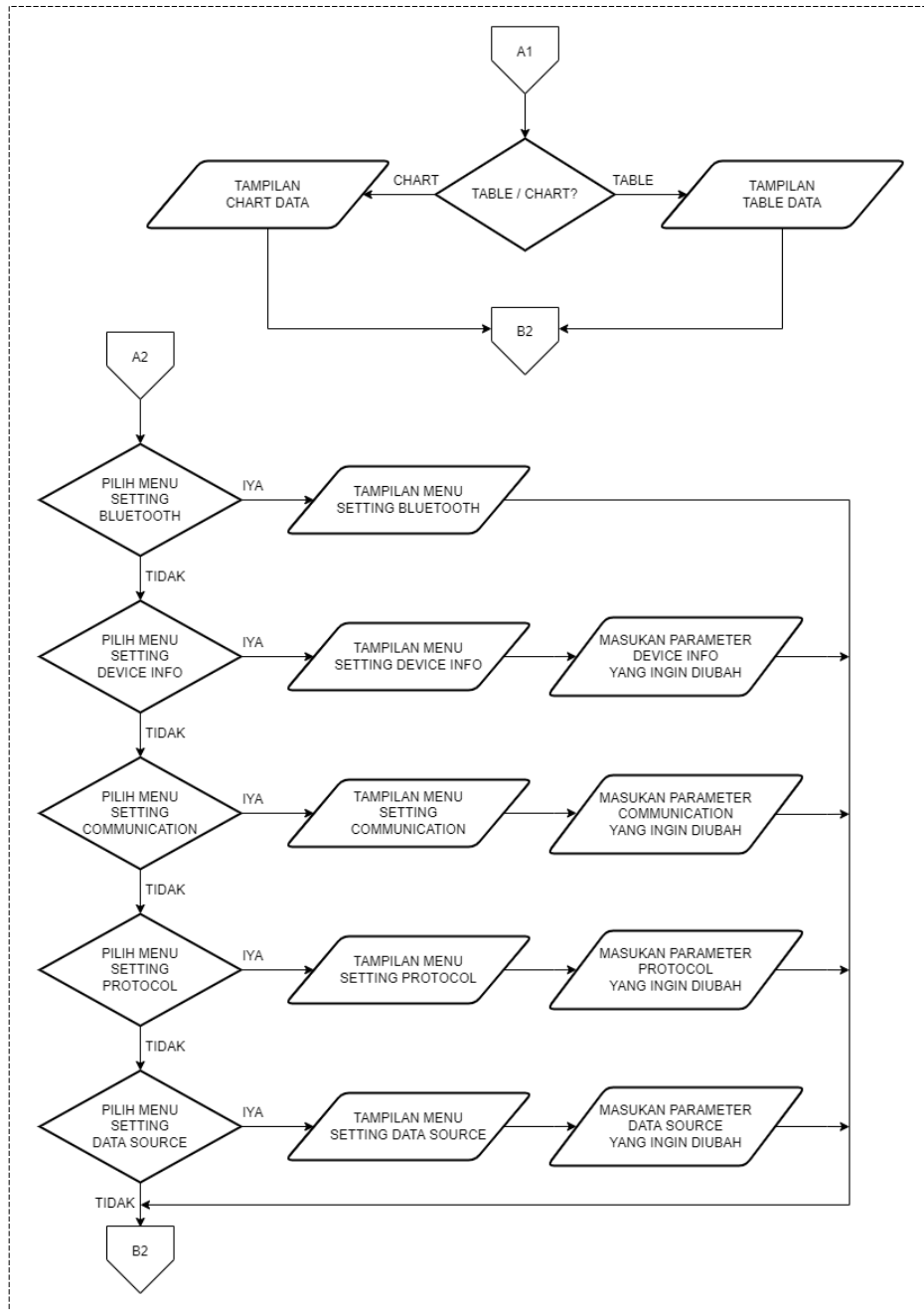
Alur proses jalannya aplikasi ditunjukkan pada gambar 3.2 dan 3.3, berawal dari mulai yang artinya aplikasi telah dibuka oleh pengguna kemudian aplikasi menampilkan *splash screen* yang berisi tampilan logo *Monita Bluetooth* dengan durasi 1 detik. Setelah itu, aplikasi menampilkan menu utama yang berisi tiga *bottom navigation* yaitu *information*, *monitor*, dan *data register* untuk memilih menu dan satu tombol pada *toolbar* untuk masuk ke menu *setting*.

Jika memilih menu *information*, maka akan dimulai pengolahan data *information* kemudian data hasil pengolahan tersebut ditampilkan pada menu tersebut dalam bentuk *list* data. Jika memilih menu *monitor*, maka akan dimulai pengolahan data *monitoring* kemudian data hasil pengolahan tersebut ditampilkan dalam bentuk tabel dan grafik data. Jika memilih menu *data register*,

maka akan dimulai pengolahan data *register* kemudian data hasil pengolahan tersebut ditampilkan dalam bentuk tabel yang dapat diubah nilai – nilainya. Jika menekan tombol *setting* pada *toolbar*, maka akan tampil menu *setting* yang terdiri dari *setting bluetooth* berfungsi untuk menyambungkan aplikasi dengan modul monita via komunikasi *bluetooth*, *setting device* info berfungsi untuk mengubah nama modul yang terhubung, *setting communication* berfungsi untuk mengubah data – data *communication* modul yang terhubung, *setting protocol* berfungsi untuk mengubah dan memilih *protocol* yang akan digunakan oleh modul, *setting data source* berfungsi untuk menampilkan dan mengubah data sumber dari modul. Alur proses tersebut akan berulang terus – menerus hingga aplikasi ditutup oleh pengguna.



Gambar 3.2 Flowchart (Alur Aplikasi) Bagian 1



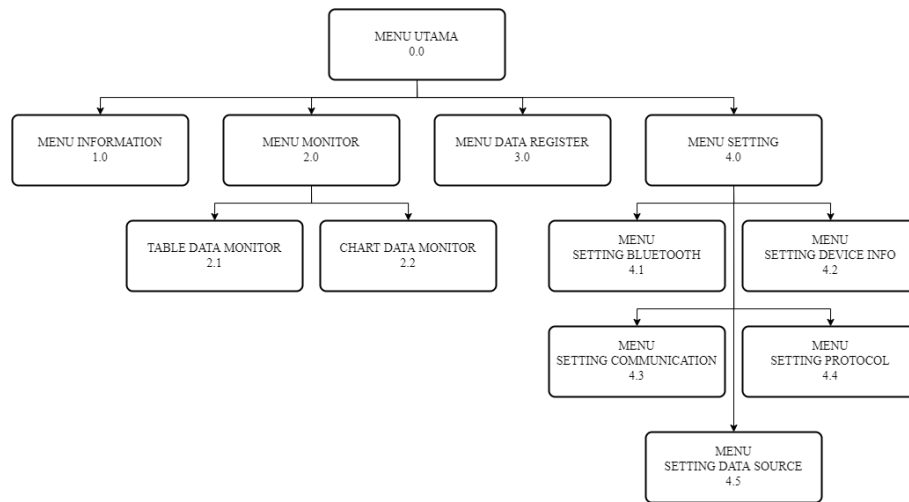
Gambar 3.3 Flowchart (Alur Aplikasi) Bagian 2

3.2.2.2 Hierarchy Plus Input Process Output (HIPO)

Aplikasi akan dikembangkan dengan metode perancangan HIPO (*Hierarchy Plus Input Process Output*) untuk fungsional sistem dan perancangan antarmuka untuk visualisasi sistem. Tingkatan HIPO yaitu VTOC (*Visual Table Of Content*) dan Diagram Ringkas (*Overview Diagram*).

1. VTOC (*Visual Tabel of Content*)

VTOC yang dibuat terdiri dari 11 proses, yaitu Menu Utama, Menu *Information*, Menu Monitor, Menu *Table Monitor*, Menu *Chart Monitor*, Menu *Data Register*, Menu *Setting*, Menu *Setting Bluetooth*, Menu *Setting Device Info*, Menu *Setting Communication*, Menu *Setting Protocol*, dan Menu *Data Source*. VTOC dari aplikasi yang dikembangkan dapat dilihat pada gambar 3.3.



Gambar 3.4 VTOC

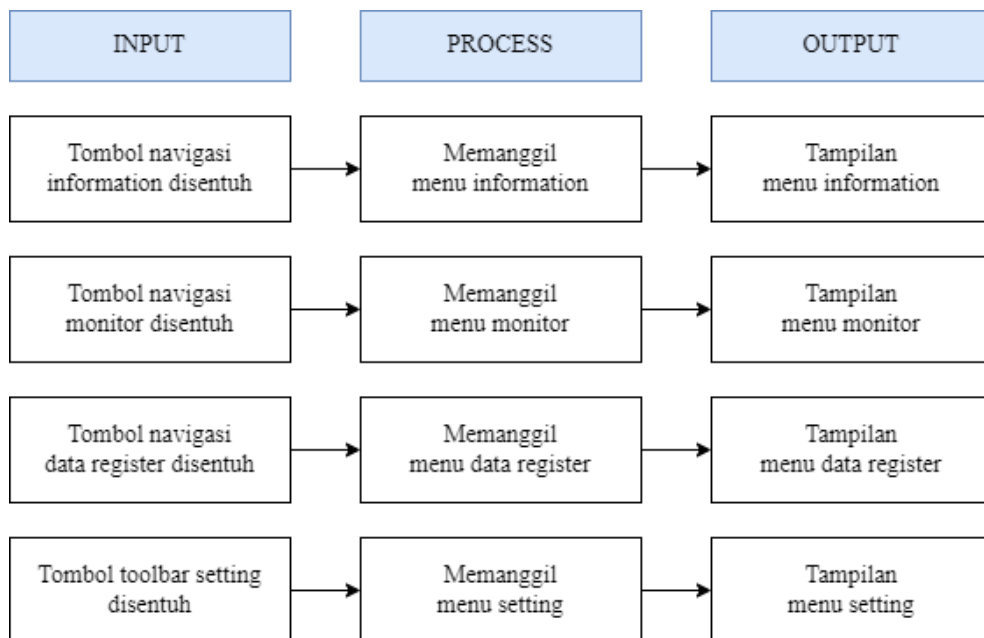
Untuk penjelasan VTOC di atas, dapat dilihat pada tabel 3.1 di bawah ini.

Tabel 3.1 Penjelasan VTOC

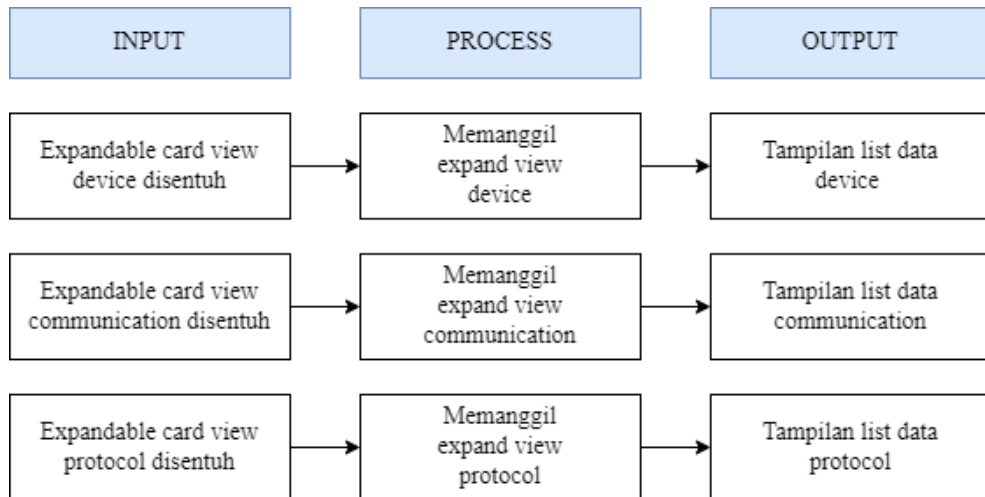
Proses	Penjelasan
Menu Utama 0.0	Proses awal pada saat aplikasi dibuka.
Menu <i>Information</i> 1.0	Proses ini akan menampilkan data <i>information</i> modul monita yang terhubung.
Menu Monitor 2.0	Proses ini akan menampilkan data <i>monitoring</i> dari modul monita yang terhubung secara <i>real-time</i> dalam bentuk tabel dan grafik.
Menu <i>Data Register</i> 3.0	Proses ini akan menampilkan data <i>register</i> dari modul monita yang terhubung.
Menu <i>Setting</i> 4.0	Proses ini akan menampilkan menu – menu <i>setting</i> yang ada di aplikasi.

Menu <i>Setting Bluetooth</i> 4.1	Proses ini untuk menghubungkan antara aplikasi dan modul monita yang tersedia.
Menu <i>Setting Device Info</i> 4.2	Proses ini untuk mengubah nama modul monita yang terhubung.
Menu <i>Setting Communication</i> 4.3	Proses ini untuk mengubah data – data <i>communication</i> modul monita yang terhubung.
Menu <i>Setting Protocol</i> 4.4	Proses ini untuk mengubah dan memilih <i>protocol</i> modul monita yang terhubung.
Menu <i>Setting Data Source</i> 4.5	Proses ini untuk memilih data <i>source</i> modul monita yang terhubung.

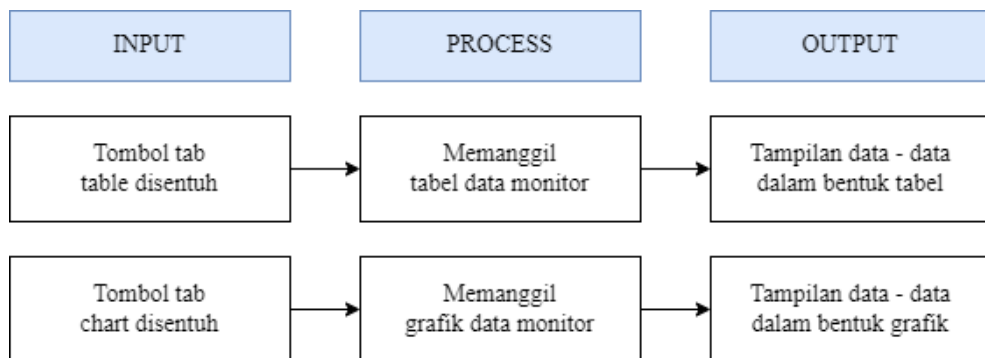
2. Diagram Ringkas (*Overview Diagram*)



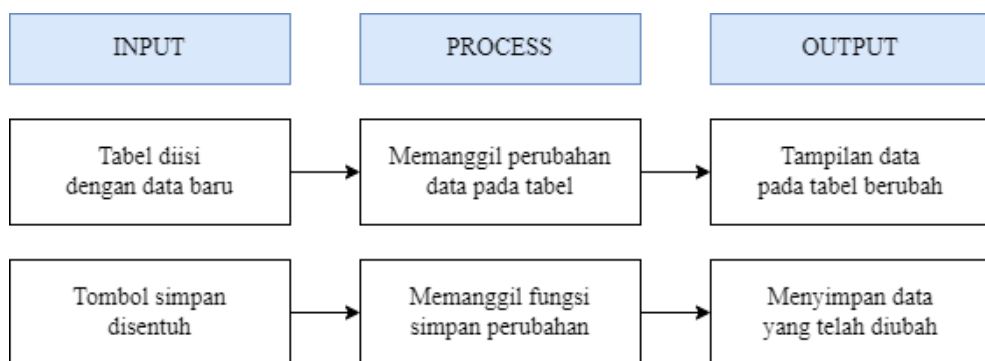
Gambar 3.5 Diagram Ringkas Menu Utama



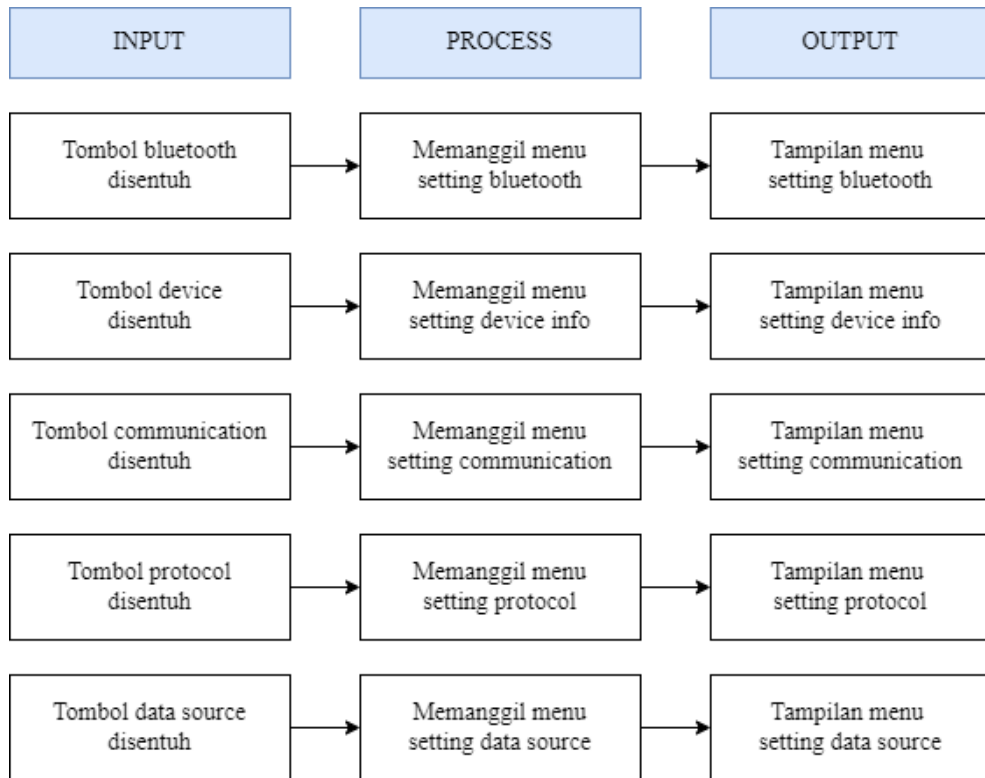
Gambar 3.6 Diagram Ringkas Menu *Information*



Gambar 3.7 Diagram Ringkas Menu *Monitor*



Gambar 3.8 Diagram Ringkas Menu *Data Register*

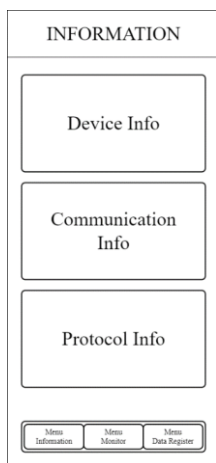


Gambar 3.9 Diagram Ringkas Menu Setting

Diagram ringkas pada gambar 3.5, 3.6, 3.7, 3.8, dan 3.9 di atas menjelaskan secara detail tentang alur mulai dari *input*, proses sampai *output* yang terjadi di setiap menu aplikasi.

3.2.2.3 Rancangan Antarmuka Aplikasi

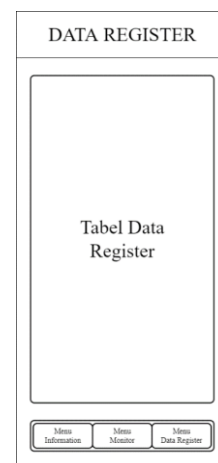
Storyboard aplikasi yang dikembangkan dapat dilihat pada gambar 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, dan 3.17.



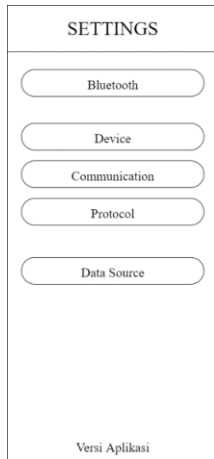
Gambar 3.10 Menu Information



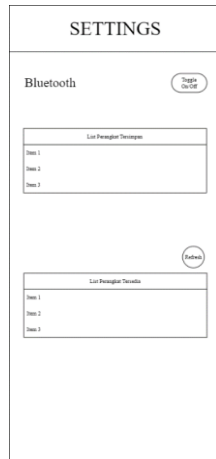
Gambar 3.11 Menu Monitor



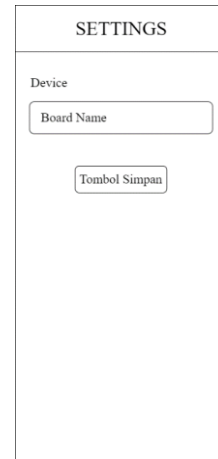
Gambar 3.12 Menu Data Register



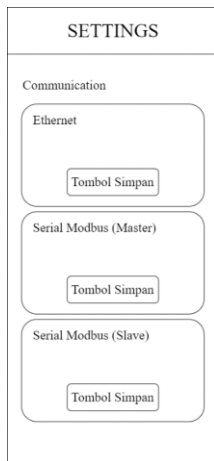
Gambar 3.13 Menu *Setting*



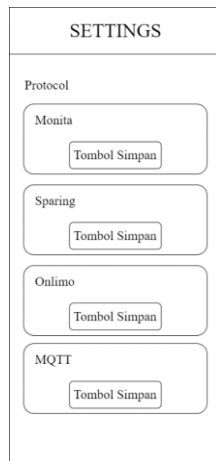
Gambar 3.14 Menu *Setting Bluetooth*



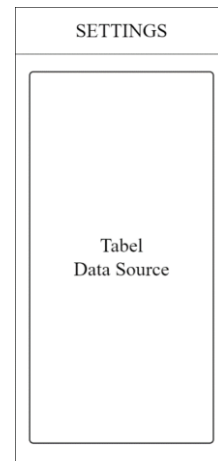
Gambar 3.15 Menu *Setting Device Info*



Gambar 3.16 Menu *Setting Communication*



Gambar 3.17 Menu *Setting Protocol*



Gambar 3.18 Menu *Setting Data Source*

Setiap menu yang ditampilkan pada *storyboard* di atas memiliki fungsi yang berbeda – beda. Fungsi menu – menu tersebut dijelaskan pada tabel 3.2 di bawah.

Tabel 3.2 Fungsi Menu – Menu Pada Aplikasi

Menu	Fungsi
<i>Information</i>	Untuk menampilkan informasi – informasi mengenai modul monita yang sedang terhubung.
Monitor	Untuk menampilkan data secara <i>realtime</i> yang diterima dari modul monita dalam bentuk tabel dan grafik.

<i>Data Register</i>	Untuk konfigurasi sensor – sensor yang terhubung dengan modul monita.
<i>Setting</i>	Menampilkan menu – menu setting,
<i>Setting Bluetooth</i>	Untuk memindai modul monita dan menghubungkan aplikasi dengan modul monita.
<i>Setting Device Info</i>	Untuk konfigurasi data yang terdapat di menu <i>device information</i> .
<i>Setting Communication</i>	Untuk konfigurasi data yang terdapat di menu <i>communication</i>
<i>Setting Protocol</i>	Untuk konfigurasi data yang terdapat di menu <i>protocol</i> .
<i>Data Source</i>	Untuk melihat sumber data yang diterima oleh modul monita.

3.2.2.4 Persepsi Teknisi Terhadap Aplikasi

Alat ukur untuk mengetahui persepsi dan implementasi teknisi dan konsumen tentang aplikasi adalah dengan menggunakan *feedback form* yang dibuat oleh peneliti.

3.2.3 DEVELOPMENT (PENGEMBANGAN)

Tahap ini adalah proses mewujudkan tahap desain menjadi kenyataan berdasarkan *storyboard* dan *flowchart*. Untuk pembuatannya dilakukan penggabungan seluruh bahan, materi, grafis, teks, dan gambar. Monita *Bluetooth* ini dikembangkan dengan menggunakan *software Figma* dan *software Android Studio*. *Software Figma* digunakan untuk mendesain *layout User Interface* atau biasa dikenal dengan istilah *front-end*, sedangkan *Android Studio* digunakan untuk memprogram seluruh algoritma aplikasi atau biasa dikenal dengan istilah *back-end*.

3.2.4 IMPLEMENTATION (IMPLEMENTASI)

Pada tahap ini, semua rancangan aplikasi yang telah dikembangkan akan dilakukan uji coba pada teknisi. Uji coba ini dilakukan kepada teknisi PT. Daun Biru Engineering. Proses uji coba dilakukan dengan melakukan tes konfigurasi modul monita melalui aplikasi serta menampilkan *monitoring* melalui aplikasi di lapangan agar terlihat apakah aplikasi berfungsi semestinya. Pengujian juga akan dilakukan dengan membandingkan konfigurasi menggunakan aplikasi

dengan konfigurasi menggunakan CLI (*Command Line Interface*) serta dihitung interval waktu dari masing – masing.

3.2.5 EVALUATION (EVALUASI)

Evaluasi merupakan tahap terakhir dari model pengembangan ADDIE. Pada tahap evaluasi ini, akan dilakukan analisis data yang telah dikumpulkan dari proses tahap *implementation*.

3.3 PENGUJIAN

Untuk keberhasilan dalam pengembangan aplikasi android, diperlukan pengujian dan analisa terhadap aplikasi yang sedang dikembangkan. Sehingga, dalam pengembangan aplikasi tersebut dapat menghasilkan aplikasi yang berjalan dengan baik sesuai dengan fungsinya.

Pada penelitian ini penulis menggunakan metode *Use Case Technique* pada *Black Box Testing* secara otomatis yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi, dan kesesuaian alur fungsi dengan fungsi yang diinginkan. Penulis menggunakan framework Espresso pada Android Studio Dolphin 2021.3.1 untuk pengujian *user interface, software* Apptim 0.9.2 untuk pengujian performa, stopwatch *online* untuk menghitung selisih durasi konfigurasi modul monita menggunakan CLI dengan menggunakan aplikasi android, dan mengkombinasikan debug pada Android Studio Dolphin 2021.3.1 dengan YAT Terminal 2.4.1 untuk menganalisa lalu lintas paket data yang melewati *bluetooth*.

3.3.1 Parameter Pengujian

Parameter pada pengujian aplikasi android untuk konfigurasi modul monita yang dikembangkan meliputi:

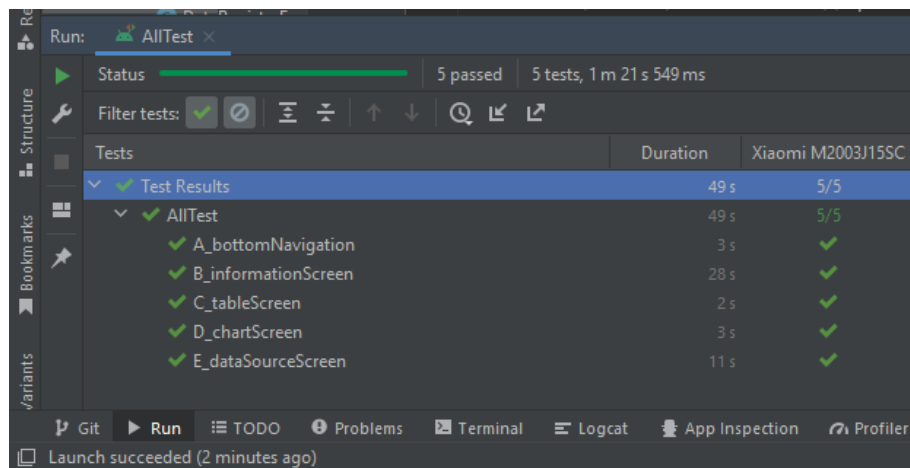
1. Validitas pada pengujian *user interface* aplikasi android.
2. *CPU Usage, Memory Usage*, dan FPS pada pengujian performa aplikasi android.
3. Selisih durasi pada pengujian konfigurasi.
4. Selisih durasi antara komunikasi kabel USB dengan *bluetooth*.

3.3.2 Skenario Pengujian

Skenario pengujian pada penelitian ini meliputi skenario pada masing – masing pengujian, yaitu ;

1. Skenario pengujian *user interface* aplikasi android.

Pada pengujian *user interface* aplikasi android dilakukan dengan cara menjalankan aplikasi android secara otomatis melalui *framework* espresso pada android studio dengan menghubungkan *smartphone* ke laptop menggunakan kabel USB. *Framework* espresso akan mengecek satu per satu fungsi *widget* yang terdapat pada aplikasi android secara otomatis. Setelah selesai, *framework* espresso akan mengeluarkan hasil dari pengujian tersebut berupa validitas.

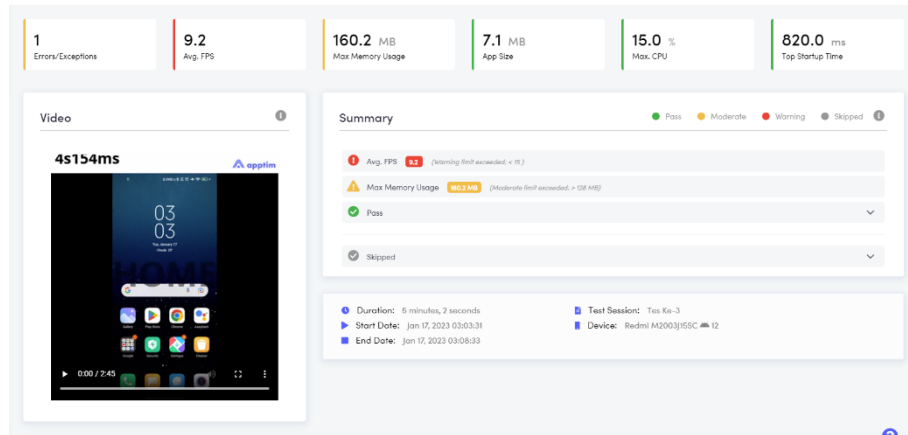


Gambar 4.1 Pengujian User Interface Aplikasi Android di Framework Espresso Android Studio

Gambar 4.1 menunjukkan contoh proses pengujian user interface aplikasi android pada android studio. Bagian tests merupakan menu – menu yang diuji user interface nya, duration merupakan durasi pengujian setiap user interface pada masing – masing menu, tanda ceklis menunjukkan bahwa user interface yang diuji bekerja dengan baik atau tidak terjadi *error*.

2. Skenario pengujian performa aplikasi android.

Pada pengujian performa aplikasi android dilakukan dengan cara menghubungkan *smartphone* dengan laptop dan aplikasi digunakan selama durasi tertentu. Untuk merekam performa aplikasi android selama digunakan, penulis menggunakan *software* apptim. Setelah selesai, *software* apptim akan menampilkan rincian performa dari aplikasi android yang diuji.



Gambar 4.2 Pengujian Performa Aplikasi Android di Software Apptim

Gambar 4.2 menunjukkan dashboard software apptim yang berisi contoh ringkasan hasil dari pengujian performa aplikasi android. Data yang ditampilkan antara lain, keterangan *error*, FPS rata – rata, penggunaan memori rata – rata, penggunaan CPU rata – rata, ukuran aplikasi, dan durasi aplikasi dibuka.

3. Skenario pengujian konfigurasi modul monita.

Pada pengujian konfigurasi modul monita dilakukan 2 tahap, yaitu :

a. Pengujian konfigurasi modul monita menggunakan CLI.

Pengujian konfigurasi modul monita menggunakan CLI dilakukan dengan cara menghubungkan modul monita ke laptop dengan menggunakan kabel USB, *software* CLI yang digunakan adalah YAT Terminal yang berfungsi untuk menuliskan command – command : cek_env, cek_data, cek_sumber. Untuk mengetahui durasi pengerjaan konfigurasi modul monita menggunakan CLI, penulis menggunakan stopwatch yang perhitungannya dimulai dari awal menghubungkan modul monita ke laptop hingga YAT Terminal menampilkan data informasi dari modul monita.

b. Pengujian konfigurasi modul monita menggunakan aplikasi android.

Pengujian konfigurasi modul monita menggunakan aplikasi android dilakukan dengan cara menghubungkan modul monita ke smartphone dengan menggunakan *bluetooth*, Untuk mengetahui durasi pengerjaan konfigurasi modul monita menggunakan aplikasi android, penulis menggunakan stopwatch yang perhitungannya dimulai dari awal

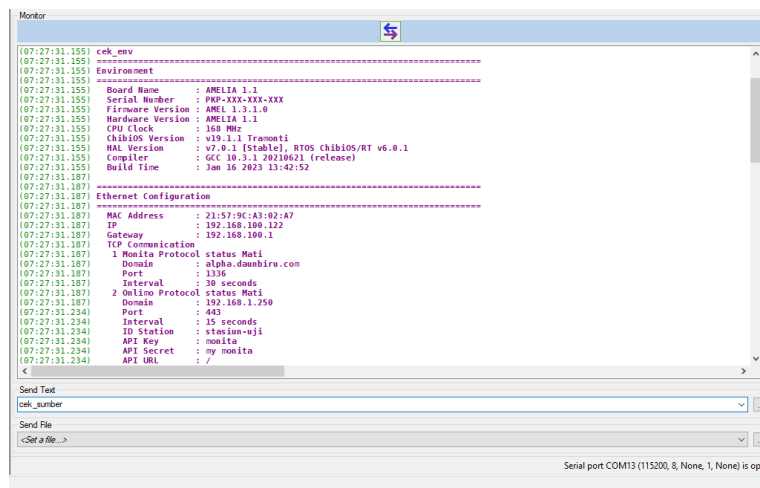
menghubungkan modul monita ke smartphone hingga aplikasi android menampilkan data informasi dari modul monita.

4. Skenario pengujian kecepatan komunikasi kabel USB dan *bluetooth*.

Pada pengujian kecepatan komunikasi kabel USB dan *bluetooth* dipisah menjadi 2, yaitu :

a. Pengujian kecepatan komunikasi kabel USB.

Pengujian kecepatan komunikasi kabel USB dilakukan dengan cara menghitung selisih waktu penerimaan awal paket data dengan penerimaan akhir paket data selama modul monita dan laptop berkomunikasi, yang dilihat menggunakan *timestamp* YAT Terminal.



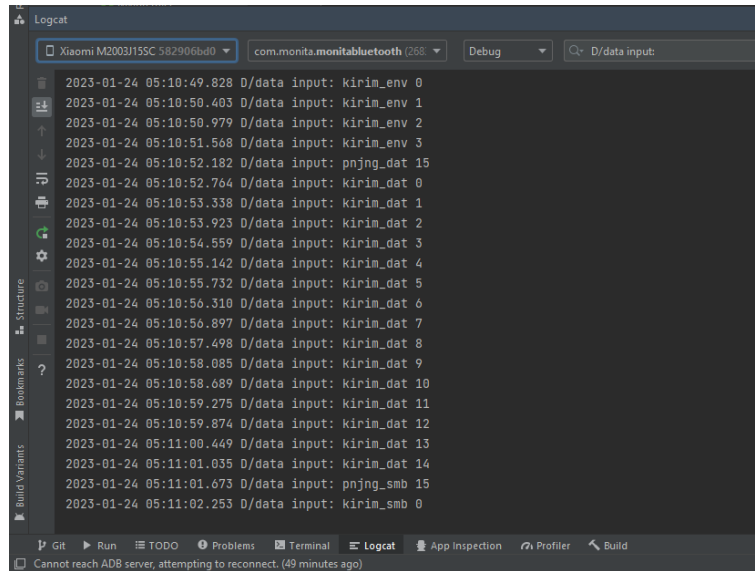
```
Monitor
(07:27:31.155) cek_env
(07:27:31.155) =====
(07:27:31.155) Environment
(07:27:31.155) =====
(07:27:31.155) Board Name      : AMELIA 1.1
(07:27:31.155) Serial Number   : PHP-XXX-XXX-XXX
(07:27:31.155) Firmware Version : AMELIA 1.1.0
(07:27:31.155) Hardware Version : AMELIA 1.1
(07:27:31.155) CPU Clock       : 100 MHz
(07:27:31.155) ChibiOS Version : v19.1.1 Tramonti
(07:27:31.155) HAL Version    : v7.0.1 [Stable], RTOS chibiOS/RT v6.0.1
(07:27:31.155) Compiler      : GCC 10.2.1 20210621 (release)
(07:27:31.155) Build Time   : Jan 16 2023 13:42:52
(07:27:31.187) =====
(07:27:31.187) Ethernet Configuration
(07:27:31.187) =====
(07:27:31.187) MAC Address   : 21:57:9C:A3:02:A7
(07:27:31.187) IP           : 192.168.100.122
(07:27:31.187) Gateway      : 192.168.100.1
(07:27:31.187) TCP Communication
(07:27:31.187) 1 Monita Protocol status Mat1
(07:27:31.187)   Domain     : alpha.daanbiru.com
(07:27:31.187)   Port      : 1336
(07:27:31.187)   Interval  : 30 seconds
(07:27:31.187) 2 OnLine Protocol status Mat2
(07:27:31.187)   Domain     : 192.168.1.250
(07:27:31.187)   Port      : 443
(07:27:31.234)   Interval  : 15 seconds
(07:27:31.234)   ID Station : stasius-uji
(07:27:31.234)   API Key   : monita
(07:27:31.234)   API Secret : my monita
(07:27:31.234)   API URL   : /
<
Send Text
cek_number
Send File
<-Set a file->
Serial port COM13 (115200, 8, None, 1, None) is open
```

Gambar 4.3 Pengujian Kecepatan Komunikasi Kabel USB

Gambar 4.3 menunjukkan tampilan software YAT Terminal yang berisi contoh proses perhitungan durasi komunikasi kabel USB. Perhitungan durasi komunikasi dilihat dari awal mulai tampil command hingga selesai. Durasi didapatkan dari selisih waktu mulai dengan waktu selesai command tampil di YAT Terminal.

b. Pengujian kecepatan komunikasi *bluetooth*.

Pengujian kecepatan komunikasi *bluetooth* dilakukan dengan cara menghitung selisih waktu penerimaan awal paket data dengan penerimaan akhir paket data selama modul monita dan aplikasi android berkomunikasi, yang dilihat menggunakan *debug time* pada android studio.



Gambar 4.4 Pengujian Kecepatan Komunikasi *Bluetooth*

Gambar 4.4 menunjukkan tampilan debug pada software android studio yang berisi contoh perhitungan durasi komunikasi *bluetooth*. Perhitungan durasi komunikasi *bluetooth* dilihat dari awal mulai tampil debug hingga selesai. Durasi didapatkan dari selisih waktu mulai dengan waktu selesai debug ditampilkan pada kolom debug.