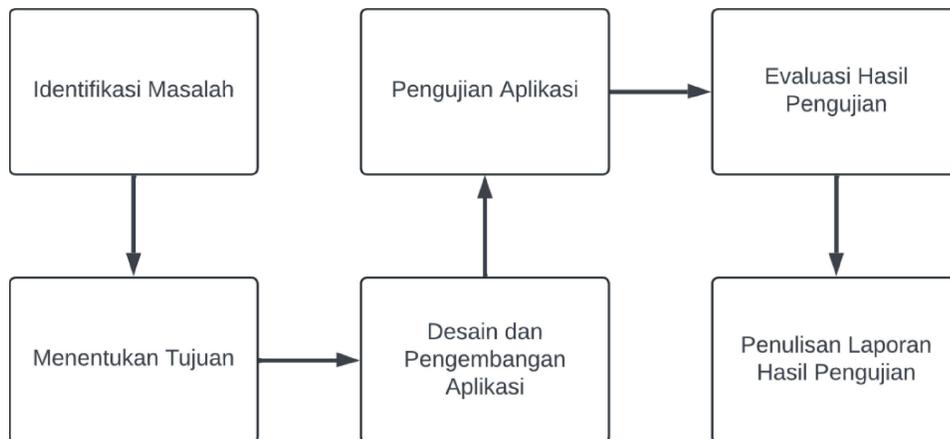


Bab III

METODE PENELITIAN

3.1. Desain Penelitian

Desain Penelitian berfungsi untuk menggambarkan tahapan-tahapan yang dilakukan dalam penelitian ini. Desain penelitian yang akan digunakan pada penelitian ini adalah metode *Design and Development* (D&D). Menurut Richey dan Klein (2014) menyatakan bahwa desain penelitian *Design and Development* (D&D) adalah penelitian yang bersifat sistematis mencakup desain, pengembangan, dan evaluasi sebuah produk. Oleh karena itu metode *Design and Development* (D&D) digunakan ketika penelitian bermaksud untuk menghasilkan produk tertentu dan kemudian dilakukan pengujian untuk mengetahui efektivitas produk tersebut. Tahapan metode penelitian *Design and Development* dapat dilihat pada Gambar 3.1.



Gambar 3.1 Tahapan Metode *Design and Development* (Richey & Klein, 2014)

Berikut adalah penjelasan dari setiap tahap dari Desain Penelitian D&D:

1. Identifikasi Masalah

Tahap ini bertujuan untuk menemukan fokus dari masalah yang akan dibahas dan diatasi oleh aplikasi yang akan dikembangkan. Untuk mengidentifikasi masalah yang pertama harus dilakukan adalah mencari berbagai referensi dan studi literatur yang berasal dari jurnal, skripsi, dan sumber lainnya. Pada penelitian ini, masalah yang diidentifikasi adalah banyaknya orang yang mendalami bidang *Internet of Things* (IoT) dan menciptakan berbagai alat untuk berbagai keperluan. Mayoritas dari mereka menggunakan platform IoT yang menyediakan berbagai alat untuk keperluan

monitoring dan kontrol perangkat yang mereka buat. Namun banyak juga orang yang memanfaatkan penyimpanan *cloud* untuk menyimpan data dari perangkat IoT yang mereka buat salah satu yang paling populer adalah *Firebase*. Akan tetapi, *Firebase* tidak dirancang sebagai platform IoT pada umumnya, sehingga tidak memiliki alat atau fitur yang dibutuhkan untuk monitoring dan kontrol perangkat IoT yang dibuat.

2. Menentukan Tujuan

Tahap ini dilakukan berdasarkan pada masalah yang telah diidentifikasi. Pada penelitian ini, tujuan yang ingin dicapai adalah merancang dan membangun sebuah aplikasi manajemen perangkat IoT berbasis *Firebase* yang dapat menyediakan fitur - fitur yang dibutuhkan untuk monitoring dan kontrol perangkat IoT pengguna. Aplikasi ini diharapkan dapat memudahkan pengguna dalam mengelola perangkat IoT mereka dengan lebih efektif dan efisien.

3. Desain dan Pengembangan Sistem

Tahap ini bertujuan untuk merancang dan membangun sistem aplikasi manajemen perangkat IoT. Proses desain mencakup pembuatan diagram alir, dan arsitektur sistem yang akan digunakan. Pengembangan sistem melibatkan pembuatan kode program menggunakan Flutter untuk pengembangan aplikasi web, *Firebase* untuk penyimpanan data perangkat, *Firebase Authentication* untuk autentikasi pengguna, dan *Firebase Functions* untuk menangani pembaruan status perangkat dan merekan log perangkat.

4. Pengujian Aplikasi

Tahap ini bertujuan untuk memastikan bahwa aplikasi yang telah dikembangkan berfungsi dengan baik. Pengujian dilakukan dengan dua metode, yaitu pengujian fungsional dan pengujian keandalan. Pengujian fungsional dilakukan untuk memastikan bahwa setiap fitur dalam aplikasi bekerja sesuai dengan yang diharapkan pengembang. Pengujian keandalan dilakukan untuk mengevaluasi seberapa baik aplikasi dalam mengambil data dari perangkat IoT.

5. Evaluasi Hasil Pengujian

Pada tahap ini, hasil dari pengujian fungsionalitas dan keandalan aplikasi dianalisis untuk mengevaluasi kinerja dan fungsionalitas aplikasi. Berdasarkan

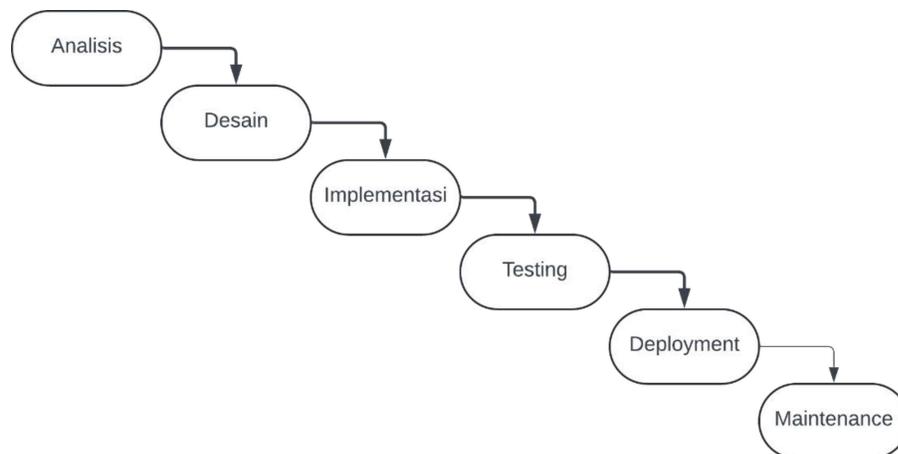
hasil evaluasi ini, dilakukan perbaikan dan penyempurnaan pada aplikasi agar dapat memenuhi kebutuhan pengguna.

6. Penulisan Laporan Hasil Pengujian

Pada tahap ini, seluruh proses dan hasil penelitian di dokumentasikan ke dalam bentuk laporan. Laporan ini berisi latar belakang penelitian, identifikasi masalah, batasan masalah, tujuan penelitian, metode penelitian, desain dan pengembangan sistem, hasil pengujian, evaluasi hasil pengujian, dan kesimpulan dan rekomendasi untuk penelitian selanjutnya.

3.2. Metode Pengembangan Perangkat Lunak

Dalam penelitian ini pengembangan aplikasi akan dilakukan dengan metode pengembangan *waterfall*. Metode *waterfall* dipilih karena setiap tahapnya memberikan terstruktur dengan jelas dan terdefinisi dengan baik, sehingga memudahkan perencanaan dan pelaksanaan pada setiap tahap pengembangan (Badrul, 2021). Adapun tahapan dalam metode *waterfall* dapat digambarkan pada Gambar 3.2.



Gambar 3.2 Metode Pengembangan Perangkat Lunak *Waterfall* (Badrul, 2021)

1. Analisis

Tahap pertama dalam metode *waterfall* adalah analisis kebutuhan, baik kebutuhan pengguna ataupun sistem. Mengidentifikasi kebutuhan pengguna meliputi kebutuhan fitur – fitur yang diperlukan di dalam sebuah aplikasi manajemen perangkat IoT pada umumnya seperti pendaftaran perangkat, monitoring status perangkat, dan kontrol perangkat. Sedangkan identifikasi

untuk kebutuhan sistem meliputi kebutuhan teknis seperti pemilihan *framework* dan *database*.

2. Desain

Setelah memahami kebutuhan selanjutnya adalah tahap desain. Tahap ini melibatkan pembuatan diagram alir dan arsitektur sistem dari aplikasi yang akan dibuat berdasarkan kebutuhan pengguna dan kebutuhan sistem yang telah diidentifikasi sebelumnya.

3. Implementasi

Tahap implementasi bermaksud untuk membuat kode program sesuai dengan desain sistem yang telah dibuat. Kode ditulis dengan menggunakan bahasa pemrograman Dart untuk integrasi *database*, antarmuka, dan logika aplikasi serta Javascript untuk beberapa logika sistem pada Firebase Functions di belakang layar.

4. Testing

Tahap pengujian bertujuan untuk memastikan bahwa aplikasi yang telah dikembangkan berfungsi dengan baik dan sesuai dengan yang diharapkan. Pengujian melibatkan pengujian fungsionalitas dan pengujian keandalan.

5. Deployment

Setelah tahap pengujian aplikasi selanjutnya adalah tahap *deployment*. Aplikasi web akan di *deploy* menggunakan Firebase Hosting sehingga website dapat di akses oleh siapa saja melalui web browser.

6. Maintenance

Tahap terakhir adalah *maintenance* atau pemeliharaan. Aplikasi web yang telah di *deploy* akan dilakukan pemeliharaan untuk memastikan aplikasi berjalan dengan baik.

3.3. Alat dan Bahan Penelitian

Untuk mengembangkan aplikasi yang sesuai dengan tujuan penelitian, peneliti perlu memilih alat dan bahan penelitian yang sesuai dengan kebutuhan. Pemilihan ini mencakup perangkat keras dan perangkat lunak yang akan digunakan dalam proses pengembangan dan pengujian aplikasi. Pemilihan alat dan bahan yang

tepat akan memastikan bahwa pengembangan aplikasi dapat berjalan dengan lancar tanpa ada gangguan teknis yang mengganggu jalannya pengembangan aplikasi.

3.3.1. Analisis Kebutuhan Perangkat Lunak

Setelah membuat rancangan maka peneliti perlu memutuskan perangkat lunak yang harus digunakan dalam mengembangkan aplikasi pada penelitian ini. Berikut adalah perangkat lunak yang digunakan pengembang pada penelitian ini.

1. Sistem Operasi Windows 11 Home

Lingkungan pengembangan aplikasi akan dijalankan pada sistem Operasi Windows 11 Home dari Microsoft. Sistem operasi ini merupakan sistem operasi untuk komersial yang dikembangkan oleh Microsoft.

2. Visual Studio Code

Visual Studio Code (VSCode) adalah sebuah *Integrated Development Environment* (IDE) yang sangat populer dalam pengembangan perangkat lunak. VSCode banyak digunakan pengembang karena memiliki banyak ekstensi dan mendukung banyak bahasa pemrograman.

3. Flutter

Flutter merupakan *framework* aplikasi *multiplatform* yang dikembangkan oleh *Google*. Flutter dikenal dapat membuat aplikasi untuk Android, iOS, web, dan desktop hanya dengan satu basis kode.

4. Google Chrome

Google Chrome adalah web browser yang banyak dipakai oleh pengguna internet di dunia. Chrome juga digunakan sebagai browser untuk pengembangan dan pengujian aplikasi web oleh pengembang.

5. Microsoft Office

Microsoft Office adalah paket perangkat lunak produktivitas yang mencakup aplikasi seperti Word, Excel, dan PowerPoint. Pada konteks penelitian Microsoft Office digunakan untuk menyusun berbagai pendukung penelitian seperti dokumen, presentasi, dan analisis data.

6. Figma

Figma merupakan perangkat lunak untuk mendesain segala jenis aset yang dibutuhkan pada aplikasi, seperti logo aplikasi, aset ilustrasi, dan beberapa rancangan antarmuka pengguna.

3.3.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras berperan penting dalam proses jalannya penelitian agar aplikasi yang dikembangkan dapat berjalan sesuai rancangan penelitian. Tanpa adanya perangkat keras yang memadai tentu akan membuat pengembangan aplikasi menjadi lebih lambat dan menghambat jalannya penelitian. Adapun spesifikasi perangkat keras yang digunakan oleh pengembang untuk merancang dan membangun aplikasi pada penelitian ini adalah sebagai berikut.

- Intel Core i7-9750H @4.50 GHz
- Nvidia GeForce GTX 1650Ti 4GB
- RAM 16GB DDR4
- SSD 512 GB M.2
- HDD 1 TB

3.4. Perancangan Sistem

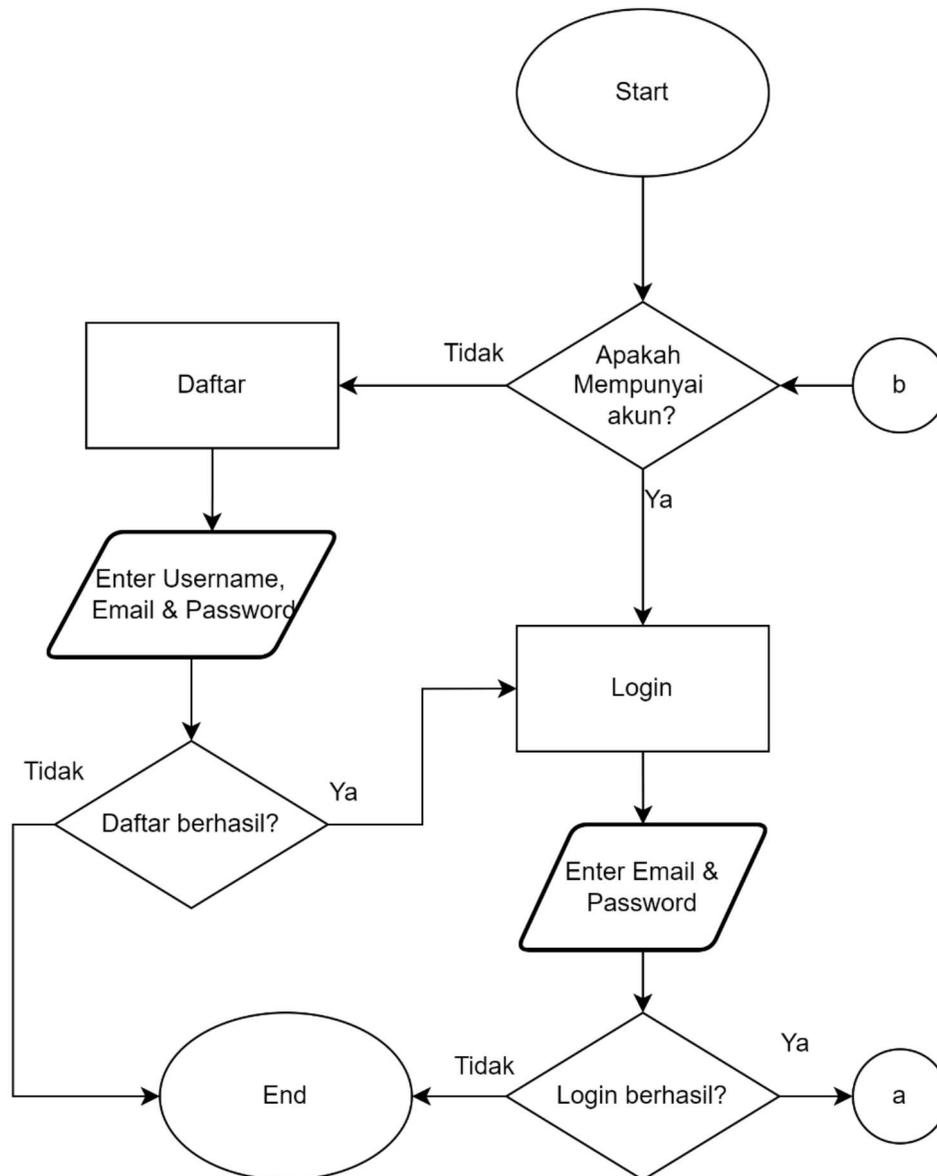
Perancangan sistem berguna untuk menentukan bagaimana alur dari aplikasi bekerja. Pada tahap ini peneliti memberikan gambaran sistem aplikasi yang akan dikembangkan. Perancangan sistem akan dijabarkan menggunakan menggunakan diagram *Undefined Model Language* (UML) yaitu diagram alir(*flowchart*) dan *activity diagram*. Untuk gambaran cara kerja sistem secara keseluruhan akan digambarkan menggunakan diagram alir sementara untuk detail cara kerja dari fitur pada sistem akan digambarkan menggunakan *activity diagram*.

3.4.1. Diagram Alir Sistem

Diagram alir bertujuan untuk menggambarkan bagaimana alur dari sistem yang dirancang bekerja. Untuk mempermudah penggambaran alur sistem diagram alir akan dibagi menjadi 3 bagian fitur utama aplikasi yang terdiri dari autentikasi pengguna, dashboard pengguna, dan perangkat pengguna. Berikut adalah gambaran alur keseluruhan rancangan sistem yang dibuat.

1. Autentikasi Pengguna

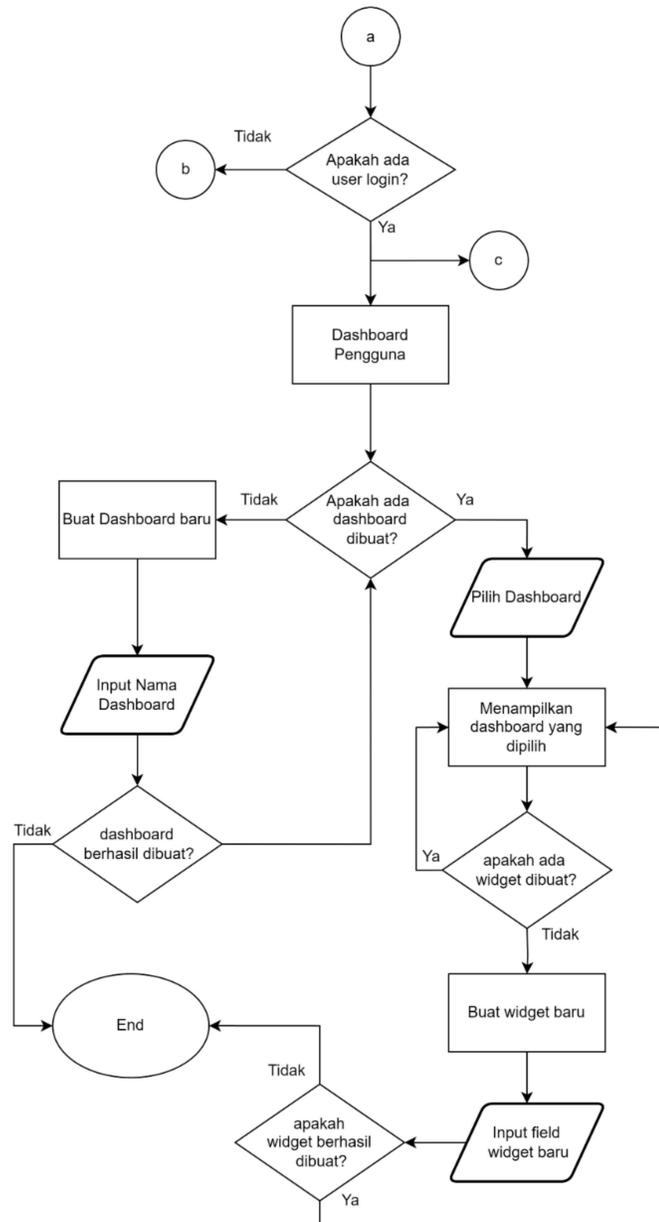
Diagram alir autentikasi pengguna bertujuan untuk menggambarkan alur proses masuk dan pendaftaran pengguna baru saat pengguna pertama kali membuka aplikasi. Langkah-langkah dalam diagram alir autentikasi pengguna digambarkan pada Gambar 3.3 seperti berikut.



Gambar 3.3 Diagram Alir Sistem Bagian Autentikasi Pengguna

2. Dashboard Pengguna

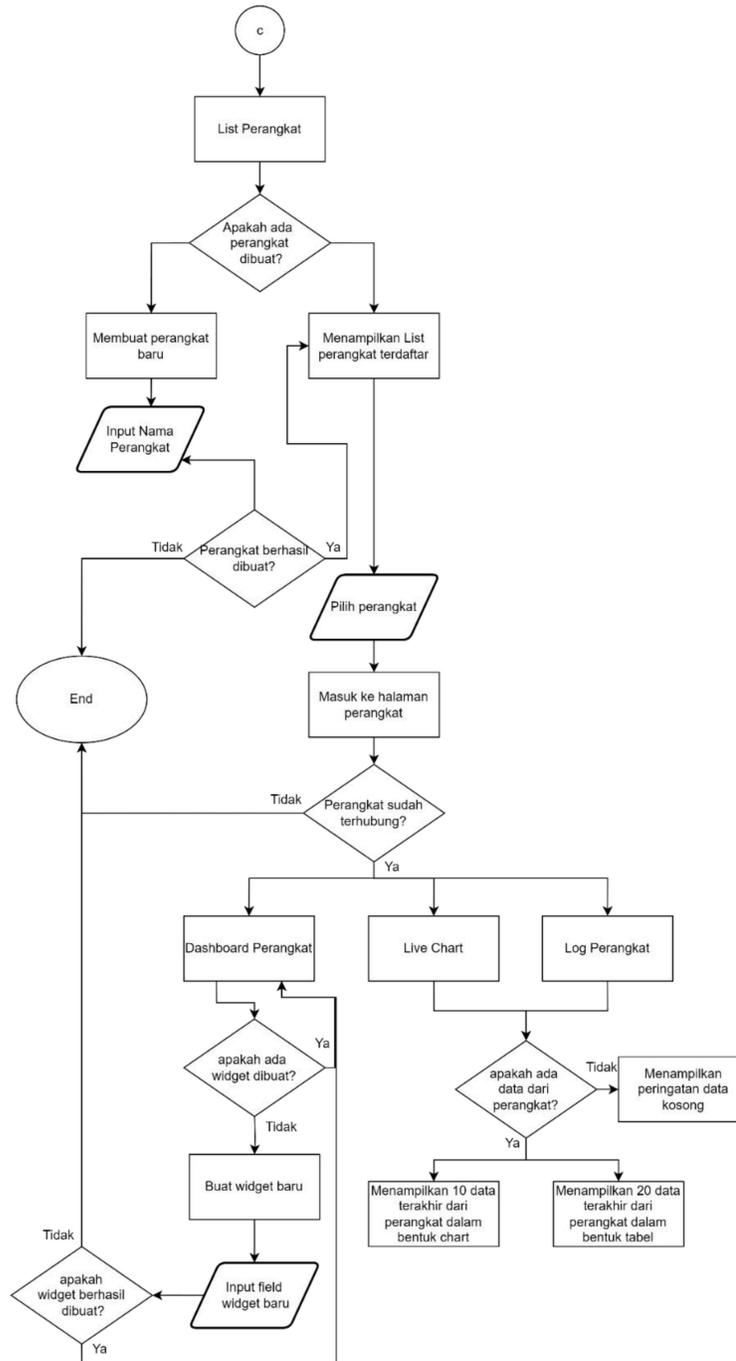
Diagram alir pada dashboard pengguna bertujuan untuk menggambarkan alur proses pembuatan widget pada fitur dashboard pengguna yang berguna untuk memonitoring dan mengontrol beberapa data perangkat IoT secara bersamaan. Langkah-langkah dalam diagram alir dashboard pengguna digambarkan pada Gambar 3.4 seperti berikut.



Gambar 3.4 Diagram Alir Sistem Bagian Dashboard Pengguna

3. Perangkat Pengguna

Diagram alir perangkat pengguna bertujuan untuk menggambarkan alur fitur-fitur yang dapat digunakan pengguna ketika perangkat terhubung pada aplikasi. Alur halaman perangkat pengguna dapat dilihat pada Gambar 3.5.



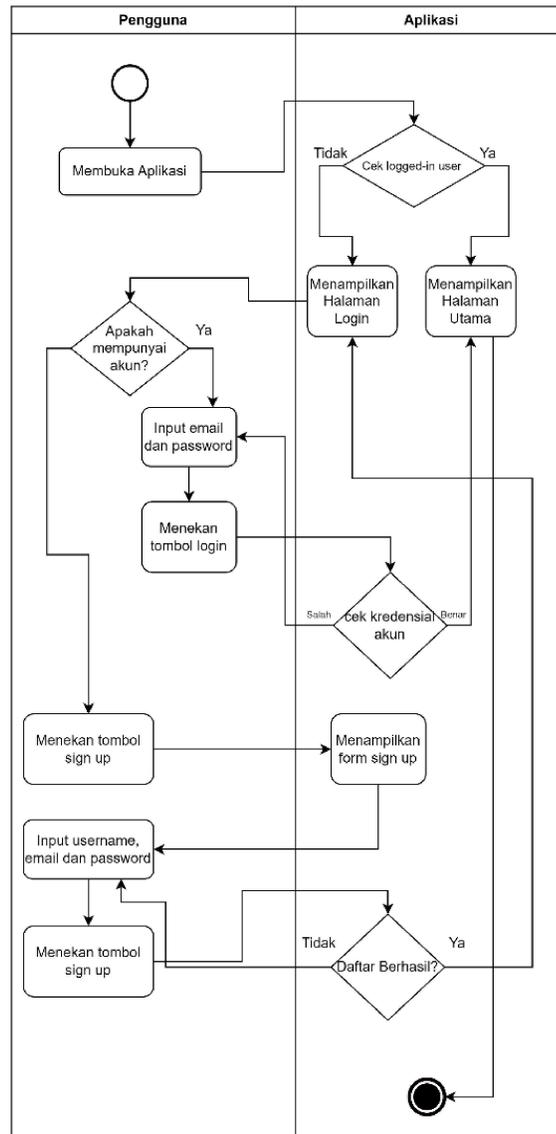
Gambar 3.5 Diagram Alir Sistem Bagian Perangkat Pengguna

3.4.2. Activity Diagram Fitur Aplikasi

Beberapa *activity diagram* berikut akan menggambarkan bagaimana alur fitur – fitur pada aplikasi bekerja. Berikut adalah *activity diagram* dari cara kerja aplikasi yang akan dikembangkan.

1. Pengguna Masuk ke Aplikasi

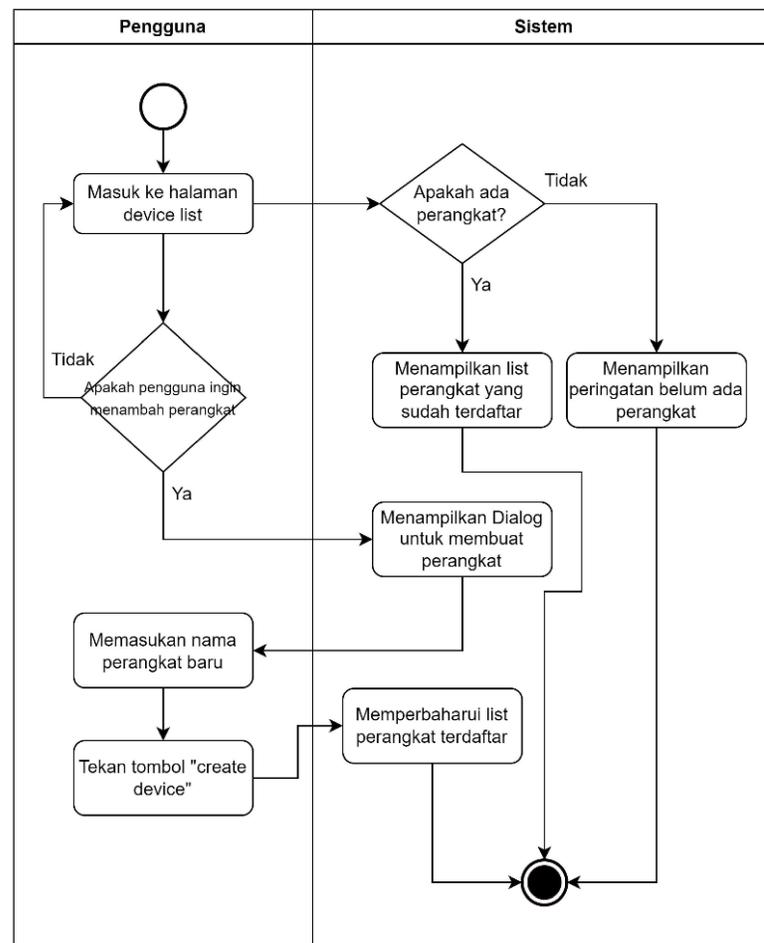
Saat pengguna masuk membuka aplikasi pengguna perlu melakukan autentikasi terlebih dahulu. *Activity diagram* ketika pengguna ingin masuk ke aplikasi dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Activity Diagram* Masuk ke Aplikasi

2. Menambahkan Alat

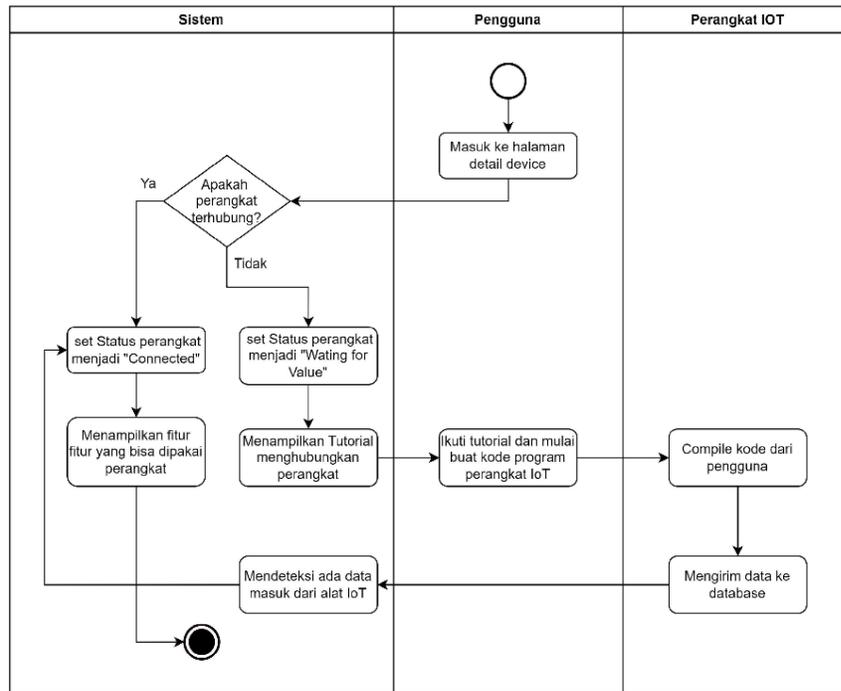
Agar dapat menggunakan berbagai fitur pada aplikasi pengguna perlu mendaftarkan alat yang dibuat oleh pengguna untuk bisa di dihubungkan dengan aplikasi. Gambaran alur pengguna dan sistem untuk mendaftarkan alat dapat dilihat pada gambar 3.7.



Gambar 3.7 Activity Diagram Membuat Perangkat Baru

3. Menghubungkan Perangkat Baru

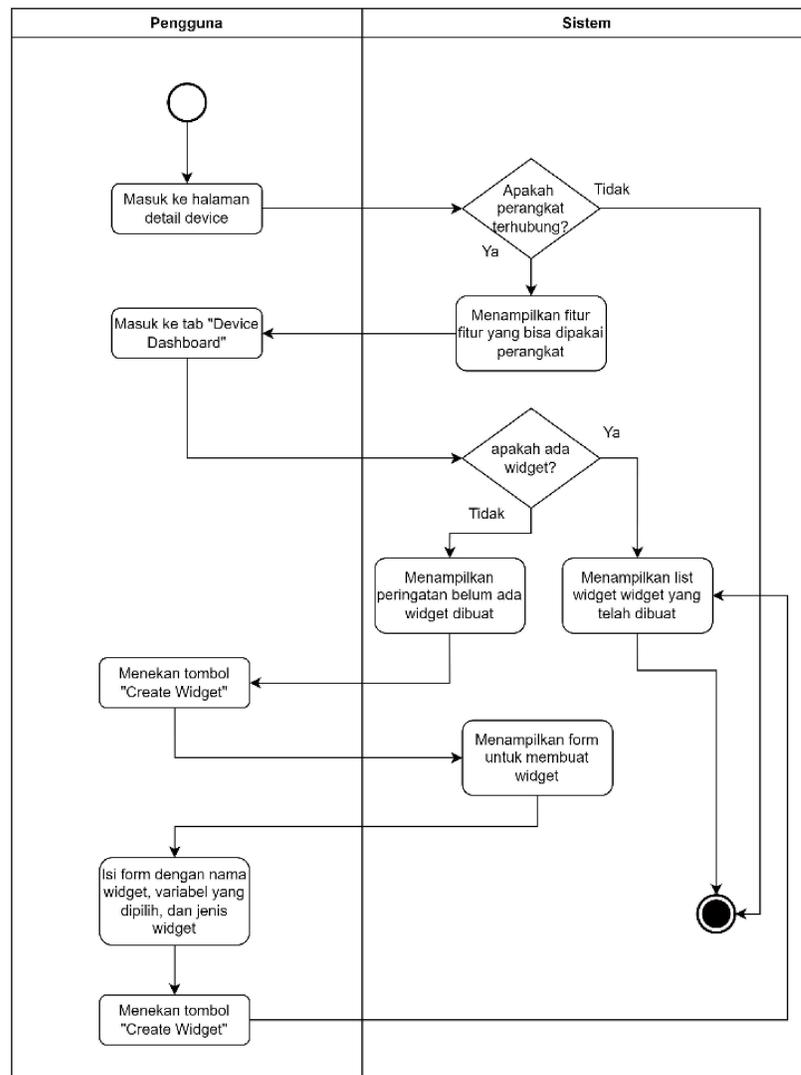
Untuk dapat memonitor data - data dari perangkat IoT aplikasi harus menerima data dari perangkat IoT yang telah didaftarkan sebelumnya. Alur dari menghubungkan perangkat ke aplikasi dapat dilihat pada Gambar 3.8.



Gambar 3.8 *Activity Diagram* Menghubungkan Perangkat

4. Membuat *Widget* untuk *Dashboard* Perangkat

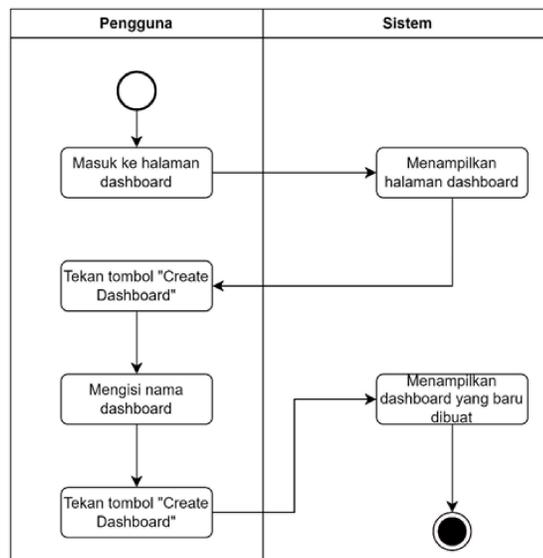
Monitoring data dari perangkat IoT baru dapat dilakukan setelah perangkat berhasil mengirim data ke aplikasi. Untuk dapat melihat data dari perangkat tersebut pengguna dapat memanfaatkan fitur *dashboard* perangkat dan membuat *widget* yang menampilkan nilai dari variabel yang dipilih pengguna saat membuat *widget* tersebut. Pada Gambar 3.9 menggambarkan bagaimana alur pengguna dan sistem saat membuat *widget* untuk menampilkan data perangkat.



Gambar 3.9 Activity Diagram Membuat Widget

5. Membuat Dashboard Pengguna

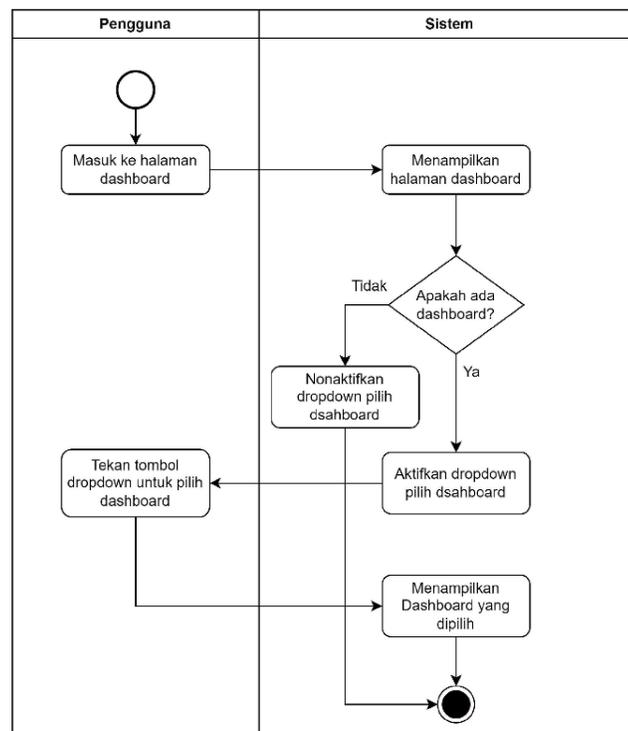
Saat pengguna telah mendaftarkan dan menghubungkan lebih dari satu perangkat IoT pengguna dapat melakukan monitoring secara sekaligus pada beberapa perangkat IoT yang pengguna daftarkan melalui fitur dashboard pengguna. Pengguna dapat membuat dan memilih dashboard yang pengguna buat untuk kustomisasi variabel dari perangkat mana saja yang ingin ditampilkan. Alur bagaimana fitur untuk membuat dashboard dapat dilihat pada Gambar 3.10.



Gambar 3.10 *Activity Diagram* Membuat Dashboard Pengguna

6. Memilih Dashboard Pengguna

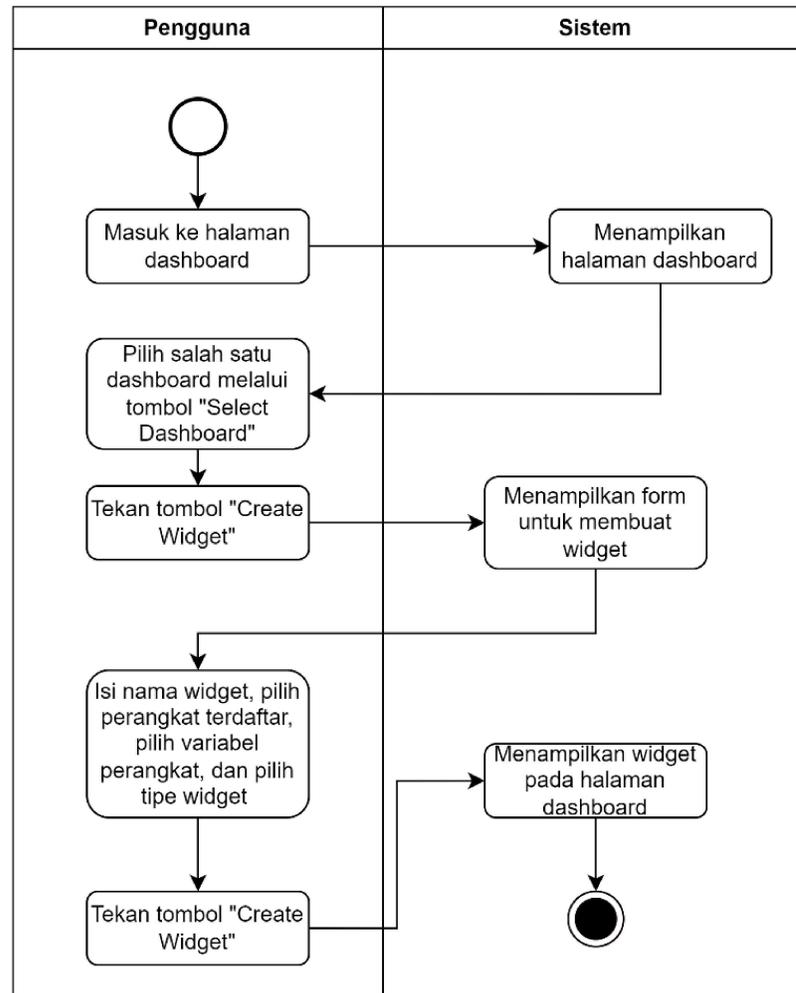
Setelah membuat dashboard pengguna dapat memilih dashboard yang telah dibuat untuk menampilkan widget – widget yang dibuat pengguna yang menampilkan berbagai data perangkat IoT seperti pada gambar 3.11.



Gambar 3.11 *Activity Diagram* Memilih Dashboard

7. Membuat *Widget* pada Dashboard Pengguna

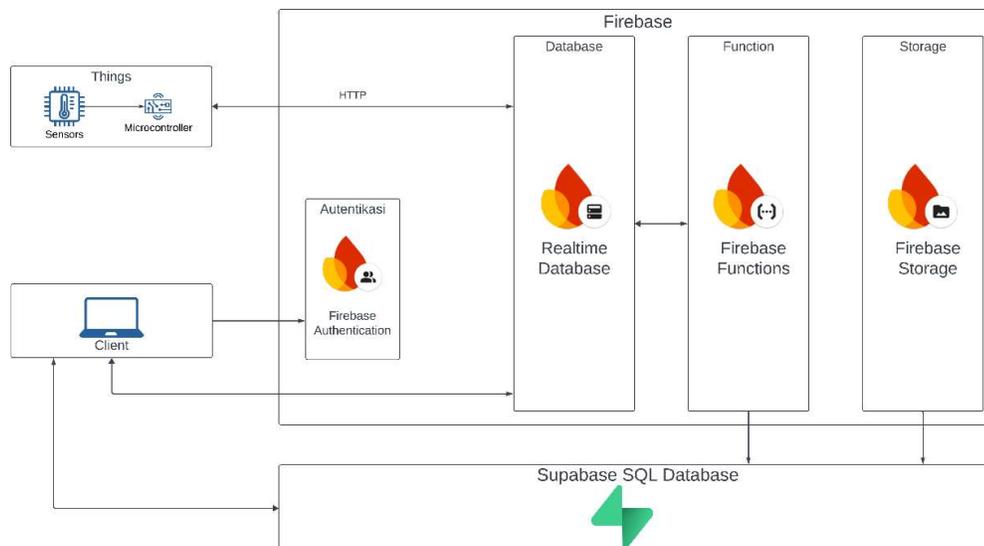
Pengguna dapat membuat *widget* untuk melakukan monitoring ke berbagai perangkat IoT yang pengguna pilih pada dashboard yang pengguna pilih. Adapun untuk alur rancangan sistem untuk membuat *widget* pada dashboard pengguna dapat dilihat pada Gambar 3.12 di bawah ini.



Gambar 3.12 *Activity Diagram* Membuat *Widget*

3.4.3. Diagram Arsitektur Sistem

Diagram arsitektur sistem adalah representasi visual dari komponen – komponen pada sistem perangkat lunak. Arsitektur sistem mengacu pada berbagai fungsi, implementasi, dan interaksinya antara satu dengan yang lain (Amazon Web Services, n.d.). Rancangan arsitektur sistem dari penelitian ini dapat dilihat dari Gambar 3.13.



Gambar 3.13 Diagram Arsitektur Sistem

Sistem yang dibuat akan menggunakan 2 *database*, *Firebase Realtime Database* untuk menyimpan data-data yang dikirim dari perangkat dan *Supabase SQL Database* untuk menyimpan data - data pengguna. Kemudian *Firebase Functions* akan berperan untuk menjalankan fungsi – fungsi di latar belakang untuk keperluan beberapa fitur aplikasi.

Fungsi pertama yang berjalan di *Firebase Functions* akan memantau perubahan data untuk setiap perangkat. Setiap kali terjadi perubahan nilai, perubahan tersebut akan dicatat ke salah satu tabel di *database SQL*. Kemudian Fungsi kedua akan berperan untuk perubahan status setiap perangkat, fungsi akan membandingkan waktu perangkat mengirimkan data dan waktu saat ini ketika perbandingan waktu kurang dari 5 menit maka status perangkat IoT akan berubah menjadi “*Online*” dan sebaliknya jika hasil perbandingan waktu terakhir perangkat mengirimkan data lebih dari waktu saat fungsi dijalankan maka status perangkat akan menjadi “*Offline*”.

Firebase Storage akan berperan untuk menampung foto profil milik pengguna dan log perangkat. Ketika pengguna mengubah foto profil mereka di aplikasi maka foto tersebut akan di unggah ke *Firebase Storage* dan URL dari foto tersebut akan disimpan ke dalam sebuah tabel di *Supabase SQL Database*. Log perangkat akan disimpan secara otomatis melalui sebuah fungsi yang dieksekusi di latar belakang menggunakan *Firebase Functions*.

3.5. Instrumen Penelitian

Pengujian sistem untuk penelitian ini akan dilakukan dengan tiga tahap. Pertama adalah pengujian fungsionalitas menggunakan metode *blackbox* yang akan menguji semua fitur pada aplikasi yang dibangun berjalan sesuai rencana. Kedua adalah pengujian keandalan *Firebase* untuk menerima data dari perangkat IoT menggunakan metode pengujian *QoS (Quality of Service)* menggunakan standarisasi *TIPHON* untuk menguji keandalan *Firebase* dalam menerima data dari perangkat IoT. Terakhir adalah pengujian platform yang bertujuan untuk mengetahui latensi aplikasi dalam mengambil data dari perangkat *database*.

3.5.1. Pengujian Blackbox

Pengujian *Blackbox* merupakan cara pengujian perangkat lunak untuk mengevaluasi fungsionalitas sistem tanpa mengetahui bagaimana struktur kode yang ditulis untuk membangun perangkat lunak tersebut (Beizer, 1996). Untuk melakukan pengujian *blackbox* pengujian akan membuat beberapa kasus uji untuk menguji fitur-fitur perangkat lunak. Kasus uji akan berisi skenario – skenario yang akan dilakukan oleh pengguna dan apa yang seharusnya dilakukan oleh sistem untuk setiap skenario tersebut. Kemudian pengujian akan membandingkan antara hasil yang dari pengujian dan hasil yang diharapkan untuk mengetahui bahwa semua fitur berjalan dengan baik atau tidak.

Pada penelitian ini pengujian *blackbox* akan dibagi menjadi beberapa fitur utama dari aplikasi, yaitu autentikasi pengguna, daftar perangkat, kontrol dan monitoring perangkat, dan dashboard pengguna untuk kontrol dan monitoring berbagai perangkat IoT. Berikut adalah kasus uji untuk pengujian *blackbox* pada penelitian ini.

1. Autentikasi Pengguna

Pengujian fitur autentikasi mencakup semua tahapan agar pengguna dapat mengakses dan keluar dari aplikasi. Pengujian fitur autentikasi mencakup fitur *sign-in* untuk memberikan akses pengguna terdaftar menggunakan aplikasi, *sign-up* untuk memberikan akses kepada pengguna baru untuk menggunakan aplikasi, dan *sign-out* untuk membuat pengguna dapat keluar dari aplikasi. Adapun kasus uji untuk pengujian autentikasi dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kasus Uji Autentikasi Pengguna

No	Komponen yang Diuji	Skenario	Hasil Yang Diharapkan
1	<i>Sign In</i>	Memasukan Email dan password yang terdaftar	Pengguna masuk ke halaman utama
		Memasukan Email dan password yang salah	Menampilkan peringatan email atau password salah
		Mengosongkan field email atau password	Menampilkan peringatan field harus diisi
2	<i>Sign Up</i>	Memasukan semua field dengan benar	Muncul dialog akun berhasil dibuat
		Mengosongkan salah field username, email atau password	Menampilkan peringatan semua field harus diisi
		Membedakan isi field ulang password	Menampilkan peringatan password tidak sama
3	<i>Sign Out</i>	Keluar dari aplikasi	Pengguna keluar dari aplikasi dan kembali ke halaman login

2. Daftar Perangkat

Fitur daftar perangkat berfungsi untuk mengelola perangkat-perangkat terdaftar milik pengguna. Pengujian ini mencakup semua kasus pengelolaan perangkat seperti mengganti nama perangkat, menambah perangkat baru, dan menghapus perangkat. Skenario pengujian untuk fitur daftar perangkat tertera pada Tabel 3.2 di bawah ini.

Tabel 3.2 Kasus uji daftar perangkat

No	Komponen yang Diuji	Skenario	Hasil yang Diharapkan
1	List Perangkat	Saat memasuki laman list perangkat menampilkan semua perangkat milik pengguna	Semua perangkat milik pengguna ditampilkan melalui sebuah tabel
		Mengganti nama perangkat	Nama Perangkat pada tabel berubah
		Menghapus Perangkat	Perangkat yang dihapus tidak ditampilkan pada tabel perangkat
		Menambah Perangkat	Perangkat yang ditambahkan akan ditampilkan pada tabel

3. Monitoring dan Kontrol Perangkat

Pengujian ini mencakup semua kasus ketika perangkat telah terhubung dan belum terhubung dengan aplikasi. Saat perangkat terhubung pengujian akan melibatkan fitur *dashboard* perangkat, *chart* data, dan log perangkat. Tabel 3.3 menunjukkan kasus uji untuk monitoring dan kontrol perangkat.

Tabel 3.3 Kasus Uji Monitoring dan Kontrol Perangkat

No	Komponen yang Diuji	Skenario	Hasil yang Diharapkan
1	Menghubungkan Perangkat	Perangkat belum terhubung dengan platform	Menampilkan halaman tutorial untuk menghubungkan perangkat

		Perangkat telah terhubung dengan platform	Menampilkan 3 tab halaman fitur perangkat, yaitu <i>Dashboard</i> Perangkat, <i>Chart Data</i> , dan Log Perangkat
2	<i>Dashboard</i> Perangkat	Menambahkan <i>widget</i> baru untuk <i>dashboard</i>	<i>Widget</i> yang di tambahkan muncul pada tab <i>dashboard</i> perangkat dihalaman detail perangkat
		Menghapus <i>widget</i> pada <i>dashboard</i>	<i>Widget</i> yang dihapus menghilang dari tab <i>dashboard</i> perangkat dihalaman detail perangkat
		Mengganti nama <i>widget</i> pada <i>dashboard</i>	Nama <i>widget</i> pada <i>dashboard</i> berubah setelah diedit
		Menampilkan Nilai dari sensor pada <i>widget</i> bertipe <i>Text</i>	Nilai dari sensor ditampilkan secara <i>realtime</i> dalam bentuk <i>text</i> didalam <i>widget</i>
		Menampilkan Nilai dari sensor pada <i>widget</i> bertipe <i>Gauge</i>	Nilai dari sensor ditampilkan secara <i>realtime</i> dalam bentuk <i>gauge</i> didalam <i>widget</i>
		Mengubah nilai dari suatu variabel melalui <i>widget</i> bertipe <i>switch</i>	Varibel bernilai 0 saat <i>switch</i> dalam kondisi <i>off</i> dan bernilai 1 saat <i>switch</i> dalam kondisi <i>on</i>
		Mengubah nilai dari suatu variabel melalui <i>widget</i> bertipe <i>Input</i>	Varibel bernilai sesuai dengan apa yang dimasukan pengguna pada <i>widget</i>
3	Chart Data	Menampilkan log histori ke dalam bentuk grafik <i>cartesian</i>	Grafik menampilkan 10 data terakhir yang terekam dalam log histori

4	Log Perangkat	Menampilkan log histori ke dalam bentuk tabel	Tabel menampilkan 20 data terakhir perangkat yang terekam
5	Catatan	Menampilkan semua catatan yang dibuat pengguna	Catatan ditampilkan pada bagian atas kanan halaman perangkat
		Membuat Catatan Baru	Catatan yang baru dibuat akan langsung ditampilkan pada <i>list</i> catatan yang tersedia
		Menampilkan isi dari catatan dan lampiran pada catatan	Catatan ditampilkan pada sebuah dialog menampilkan judul, isi, dan semua lampiran pada catatan tersebut
		Menghapus Catatan yang telah dibuat	Catatan dan lampiran yang dihapus akan hilang dari <i>list</i>
		Mengedit catatan yang telah dibuat	Catatan akan menampilkan data terbaru setelah diedit

4. Dashboard Pengguna

Pengujian Dashboard Pengguna berfungsi untuk menguji semua fungsionalitas fitur Dashboard Pengguna yang mencakup membuat *widget*, menghapus *widget*, dan menguji semua jenis *widget* dalam menampilkan data dari perangkat IoT. Skenario pengujian fitur dashboard pengguna dapat dilihat melalui Tabel 3.4 dibawah ini.

Tabel 3.4 Kasus uji *Dashboard* Pengguna.

No	Komponen yang Diuji	Skenario	Hasil yang Diharapkan
1	Membuat <i>Dashboard</i> Baru	Membuat <i>Dashboard</i> baru yang dapat berisi <i>widget</i> dari berbagai perangkat IoT milik pengguna	<i>Dashboard</i> yang dibuat dapat dipilih melalui <i>dropdown</i> pada halaman kemudian ketika dipilih akan

			menampilkan semua <i>widget</i> yang telah dibuat
2	<i>Widget Dashboard</i> Pengguna	Membuat <i>widget</i> baru untuk <i>dashboard</i> yang dipilih	<i>Widget</i> yang di tambahkan muncul pada <i>dashboard</i> yang dipilih di halaman <i>dashboard</i> pengguna
		Menghapus <i>widget</i> pada <i>dashboard</i>	<i>Widget</i> yang dihapus menghilang dari <i>dashboard</i> yang dipilih oleh pengguna
		Mengganti nama <i>widget</i> pada <i>dashboard</i>	Nama <i>widget</i> pada <i>dashboard</i> berubah setelah diedit
		Menampilkan Nilai dari sensor pada <i>widget</i> bertipe <i>Text</i>	Nilai dari sensor ditampilkan secara <i>realtime</i> dalam bentuk <i>text</i> di dalam <i>widget</i>
		Menampilkan Nilai dari sensor pada <i>widget</i> bertipe <i>Gauge</i>	Nilai dari sensor ditampilkan secara <i>realtime</i> dalam bentuk <i>gauge</i> di dalam <i>widget</i>
		Mengubah nilai dari suatu variabel melalui <i>widget</i> bertipe <i>switch</i>	Variabel bernilai 0 saat <i>switch</i> dalam kondisi <i>off</i> dan bernilai 1 saat <i>switch</i> dalam kondisi <i>on</i>
		Mengubah nilai dari suatu variabel melalui <i>widget</i> bertipe <i>Input</i>	Variabel bernilai sesuai dengan apa yang di masukan pengguna <i>pengguna</i> pada <i>widget</i>

5. Profil Pengguna

Fitur profil pengguna berfungsi untuk menampilkan informasi pengguna saat pengguna mendaftar pada aplikasi. Tabel 3.5 menunjukkan kasus uji untuk fitur profil pengguna.

Tabel 3.5 Kasus uji profil pengguna

No	Komponen yang Diuji	Skenario	Hasil yang Diharapkan
1	Profil	Mengganti foto profil	Foto profil berubah setelah diganti
2	<i>Username</i>	Mengganti <i>username</i>	<i>Username</i> berubah setelah diganti oleh pengguna

3.5.2. Pengujian Quality of Service

Pengujian QoS (*Quality of Service*) bermaksud untuk menguji seberapa baik Firebase dalam memproses data-data perangkat IoT. Pengujian akan dilakukan dengan menggunakan metode QoS menggunakan standarisasi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*). Terdapat beberapa parameter QoS seperti *throughput*, *delay*, *jitter*, *packet loss* dan *packet delivery* (Mulyani & Oktawati, 2022). Namun penelitian ini hanya akan melakukan tiga jenis pengujian QoS yaitu pengujian *packet delivery*, *pengujian delay*, dan *packet loss*. *Packet delivery* yaitu menguji apakah jumlah paket yang tersimpan sama dengan jumlah paket yang dikirim. Pengujian *packet delivery* akan dilakukan dengan cara mengirim beberapa data dari perangkat IoT dan memastikan semua data yang dikirim tersimpan di database. Menurut TIPHON kategori *packet delivery* terbagi menjadi 4 kategori seperti pada Tabel 3.6 berikut.

Tabel 3.6 Kategori *Packet Delivery* Menurut TIPHON

Kategori	Persentase	Indeks
Sangat Bagus	100%	4
Bagus	97%	3
Sedang	85%	2
Jelek	75%	1

Pengujian *delay* berarti menguji waktu penundaan paket saat proses pengiriman dari satu titik ke titik lain. *Delay* menggunakan satuan milidetik (ms). Pengujian *delay* akan dilakukan dengan menghitung rata-rata dari semua *delay* dari keseluruhan data yang telah dikirim. Menurut standar TIPHON kategori pengujian *delay* dapat dilihat pada Tabel 3.7.

Tabel 3.7 Kategori *Delay* Menurut TIPHON

Kategori	Nilai Delay	Indeks
Sangat Bagus	<150 ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Jelek	>450 ms	1

Pengujian *packet loss* bertujuan untuk mengetahui persentase berapa banyak data yang gagal masuk ke dalam *database* dari banyak permintaan ke *database*. Pengujian *packet loss* akan dilakukan dengan menggunakan aplikasi Apache JMeter untuk melihat berapa banyak persentase permintaan yang gagal dikirim pada saat jangka waktu pengujian. Klasifikasi *packet loss* menurut TIPHON dapat dilihat pada Tabel 3.8.

Tabel 3.8 Kategori *Packet Loss* Menurut TIPHON

Kategori	Persentase	Indeks
Sangat Bagus	0 – 2%	4
Bagus	3 – 14%	3
Sedang	15 – 24%	2
Jelek	>25%	1

Hasil dari perhitungan pengujian *packet delivery*, *packet loss*, dan *delay* akan dijumlahkan dan dihitung nilai rata-ratanya. Kemudian hasil rata-rata tersebut akan menentukan kategori dari QoS. Hasil QoS menurut standarisasi TIPHON dapat dilihat pada Tabel 3.9 (Mulyani & Oktawati, 2022).

Tabel 3.9 Kategori QoS

Kategori	Persentase	Indeks
3.8 – 4	95 – 100 %	Sangat Memuaskan
3 – 3,79	75 – 94,75 %	Memuaskan
2 – 2,99	50 – 74,75 %	Kurang Memuaskan
1 – 1,99	25 – 49,75 %	Buruk

3.5.3. Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk mengukur seberapa baik platform IoT dalam mengelola, monitoring, dan mengontrol perangkat IoT yang terhubung pada platform. Menurut Salami & Yari (2018) pengujian platform IoT dapat dibagi menjadi 5 kategori yaitu *Security*, *Usages*, *Data*, *Communication*, dan *Platform*. Adapun untuk detail tiap kategori pengujian dapat dilihat pada Gambar 3.14.

Security			Platform
<ul style="list-style-type: none"> • Authentication • Confidentially • Data Storing Protection • Threat Protection 	Usages	<ul style="list-style-type: none"> • Performance • Correctness • Accesability • Predictability • Usability • Comprehensibility 	<ul style="list-style-type: none"> • Testability • Scalability • Impovability • Reuseability • Portability • Reliability • Flexibility • Latency of Receiving Data
	Data	<ul style="list-style-type: none"> • Visibility • Data Visualization • Data Storing • Data Processing • Data Sharing 	
	Communications	<ul style="list-style-type: none"> • Interfaces • API • Supporting for Heterogeneous Device 	

Gambar 3.14 *Test Framework* Untuk Platform IoT (Salami & Yari, 2018)

Seperti pada gambar di atas pengujian platform IoT ada 8 jenis yaitu *Testability*, *Scalability*, *Improvability*, *Reuseability*, *Portability*, *Reliability*, *Flexibility*, dan *Latency of Receiving Data*. Namun pada penelitian ini hanya akan dilakukan pengujian *Latency of Receiving Data* untuk mengukur kemampuan aplikasi dalam mengambil data perangkat IoT dari Firebase, *Authentication* untuk menunjukkan bagaimana pengguna dapat mengakses aplikasi, *Data Storing* untuk menguji sistem saat menyimpan data, API untuk menguji API saat menarik data, dan *Accessability* untuk menguji aksesibilitas antarmuka yang dibuat.