

**APLIKASI *SMARTHOME* DENGAN MEMANFAATKAN *IOT*
MENGUNAKAN ANDROID STUDIO**

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Sarjana Terapan/Diploma IV Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Ibnu Kurniawan

NIM: 1061809

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

APLIKASI SMARTHOME DENGAN MEMANFAATKAN IOT
MENGUNAKAN ANDROID STUDIO

Oleh :

Ibnu Kurniawan / 1061809

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur negeri Bangka Belitung

Menyetujui,

Pembimbing 1



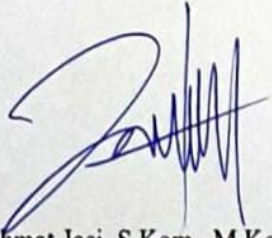
Sidhiq Andriyanto, S.T., M.Kom.
NIP. 199007182019031011

Pembimbing 2



Indra Dwisaputra, M.T.
NIP. 198811102014041002

Penguji 1



Ahmat Josi, S.Kom., M.Kom.
NIP. 198908202019031015

Penguji 2



Yang Agita Rindri, S.Kom., M.Eng.
NIP. 198609282022032003

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa : Ibnu Kurniawan

NIM : 1061809

Dengan Judul : Aplikasi Smarthome Dengan Memanfaatkan Iot
Menggunakan Android Studio

Menyatakan bahwa laporan akhir ini adalah hasil kerja saya sendiri dan bukan merupakan plagiat. Pernyataan ini saya buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Sungailiat, 11 Januari 2025

Nama Mahasiswa

Tanda Tangan

Ibnu Kurniawan



ABSTRAK

Internet of Things (IoT) memungkinkan integrasi perangkat keras dengan jaringan internet untuk kontrol dan pengaturan jarak jauh. IoT merupakan teknologi yang memungkinkan berbagai perangkat untuk saling terhubung dan berkomunikasi melalui internet, sehingga memungkinkan otomatisasi yang efisien. Penelitian ini mengembangkan aplikasi Smarthome bernama EDITH yang beroperasi berbasis IoT. Aplikasi ini memungkinkan kontrol perangkat lampu melalui smart phone dengan fitur tambahan seperti penjadwalan, monitoring energi, dan riwayat penggunaan. Sistem ini menggunakan NodeMCU ESP32 sebagai mikrokontroler yang terintegrasi dengan server Adafruit.io. Hasil pengujian menunjukkan bahwa EDITH dapat berfungsi secara efektif dalam mengontrol lampu hingga jarak jauh, dengan tingkat kepuasan pengguna sebesar 84%.

Kata Kunci : *Internet of Things, Android Studio, Smarthome, EDITH, ESP32*

ABSTRACT

The Internet of Things (IoT) allows the integration of hardware with internet networks for remote control and regulation. IoT is a technology that allows various devices to connect and communicate with each other over the internet, thus enabling efficient automation. This research develops a Smarthome application called EDITH that operates based on IoT. The app allows control of the lighting device through a smartphone with additional features such as scheduling, energy monitoring, and usage history. The system uses the NodeMCU ESP32 as a microcontroller that is integrated with the Adafruit.io server. The test results show that EDITH can function effectively in controlling the lights remotely, with a user satisfaction rate of 84%.

Keywords : *Internet of Things, Android Studio, Smarthome, EDITH, ESP32*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh. Segala puji bagi Allah SWT., yang telah melimpahkan taufiq dan hidayah-Nya kepada penulis, sehingga penulis dapat menyusun laporan proyek akhir ini dengan Judul “APLIKASI SMARTHOME DENGAN MEMANFAATKAN IOT MENGGUNAKAN ANDROID STUDIO” dan dapat menyelesaikan Program Studi Diploma IV Teknologi Rekayasa Perangkat Lunak di Politeknik Manufaktur Negeri Bangka Belitung. Penulis menyadari sepenuhnya, bahwa dalam penyusunan laporan proyek akhir ini banyak terdapat kekurangan mengingat terbatasnya kemampuan penulis, namun berkat rahmat Allah SWT, serta pengarahan dari berbagai pihak. akhirnya laporan proyek akhir ini dapat diselesaikan. Harapan penulis semoga laporan proyek akhir ini dapat bermanfaat untuk kepentingan bersama. Sehubungan dengan itu, penulis menyampaikan ucapan terima kasih kepada:

1. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Bapak Irwan, M.Sc, Ph.D selaku Wakil Direktur I Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Subhan, M.T selaku Wakil Direktur II Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Eko Sulistyono, M.T selaku Wakil Direktur III Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Sidhiq Andriyanto, S.T., M.Kom selaku Ko. Prodi D4 Teknologi Rekayasa Perangkat Lunak dan juga Dosen Pembimbing Utama Proyek Akhir.
7. Bapak Indra Dwisaputra, S.ST., M.T. selaku Dosen Pembimbing kedua Proyek Akhir.
8. Teman-teman seperjuangan yang telah memberikan motivasi dan dukungannya.

9. Ibu dan Ayah tercinta serta seluruh keluarga yang dengan penuh keikhlasan untuk dukungannya.

10. Bos Besar Nurdiah yang memberi motivasi dan dukungan untuk menyelesaikan.

Setelah melalui proses yang panjang dan penuh tantangan, akhirnya penulis dapat menyelesaikan pembuatan alat dan laporan proyek akhir ini. Penulis menyadari bahwa penulisan makalah proyek akhir ini masih terdapat banyak kekurangan baik dari segi penulisan maupun isi makalah. Oleh karena itu, penulis mengharapkan segala petunjuk, kritik, dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulisan selanjutnya. Akhir kata, penulis mohon maaf atas kekurangan dalam penulisan makalah ini dan penulis dengan senang hati menerima saran dan kritik yang membangun dari pembaca. Semoga makalah ini dapat berguna untuk menambah wawasan dan wacana bagi rekan-rekan mahasiswa. Waalaikumsalam warahmarullahi wabarakatuh.

Sungailiat, 11 Januari 2025



Ibnu Kurniawan

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Masalah	2
1.4. Batasan Masalah	3
BAB II LANDASAN TEORI.....	4
2.1. Tinjauan Pustaka	4
2.2. Keunggulan Aplikasi EDITH.....	8
2.3. <i>Internet of Things (IoT)</i>	9
2.4. Aplikasi <i>Smarthome</i>	10
2.5. Adafruit.IO	10
2.6. Arduino IDE.....	11
BAB III METODE PELAKSANAAN	12
3.1. Model Rancangan.....	12
3.1.1. Pengumpulan Data.....	12

3.1.2. Analisis data kebutuhan	13
3.1.3. Perancangan <i>Prototype</i>	13
3.1.4. Pembuatan Program Dan Alat	13
3.1.5. Uji Coba	14
3.2. Metode <i>Rapid Application Development</i>	14
3.3. Use Case Diagram	15
3.4. Analisis Kebutuhan	15
3.4.1. <i>Hardware</i>	16
3.4.2. <i>Software</i>	16
3.5. Perancangan Sistem	17
3.6. Rancangan <i>Hardware</i>	18
3.7. Pengumpulan dan Pengujian Data	18
3.7.1. Fungsionalitas	19
3.7.2. Kuesioner	19
BAB IV HASIL DAN PEMBAHASAN	20
4.1. Pembuatan Aplikasi <i>FrontEnd</i>	20
4.1.1. <i>FrontEnd</i> : Menu Utama	20
4.1.2. <i>FrontEnd</i> : Menu Lampu	22
4.1.3. <i>FrontEnd</i> : Menu Monitor	23
4.1.4. <i>FrontEnd</i> : Menu Riwayat	25
4.1.5. <i>FrontEnd</i> : Menu Jadwal	27
4.1.6. <i>FrontEnd</i> : Menu Bantuan	29
4.2. Pembuatan Program <i>Back End</i>	31
4.2.1. <i>Back End</i> : Menu Utama	31
4.2.2. <i>Back End</i> : Menu Lampu	32
4.2.3. <i>Back End</i> : Menu Monitor	34

4.2.4. <i>Back End</i> : Menu Riwayat	37
4.2.5. <i>Back End</i> : Menu Jadwal	37
4.2.6. <i>Back End</i> : Menu Bantuan	39
4.3. Server Adafuit.IO	40
4.4. Pembuatan Perangkat Keras (<i>Hardware</i>)	41
4.4.1. Rancangan Model Alat	41
4.4.2. Perakitan Alat	41
4.4.3. Pemrograman Arduino IDE	41
4.5. Pengujian Sistem Aplikasi <i>Smarthome</i>	44
4.5.1. Pengujian jarak alat dan wifi.....	44
4.5.2. Pengujian jarak alat dan aplikasi	45
4.5.3. Kesimpulan Pengujian	46
4.6. Kuesioner	46
BAB V KESIMPULAN DAN SARAN.....	50
5.1. Kesimpulan	50
5.2. Saran.....	50
DAFTAR PUSTAKA.....	51
LAMPIRAN	52

DAFTAR GAMBAR

Gambar 3 1 Model Rancangan	12
Gambar 3 2 Perancangan Prototype.....	13
Gambar 3 3 Rapid Application Development	14
Gambar 3 4 Use Case Diagram	15
Gambar 3 5 Perancangan Sistem	17
Gambar 3 6 Rancangan Hardware	18
Gambar 4 1 FrontEnd Menu Utama (b)	20
Gambar 4 2 FrontEnd Menu Utama (e)	21
Gambar 4 3 FrontEnd Menu Lampu (b)	22
Gambar 4 4 FrontEnd Menu Lampu (e).....	23
Gambar 4 5 FrontEnd Menu Monitoring (b).....	24
Gambar 4 6 FrontEnd Menu Monitoring (e)	25
Gambar 4 7 FrontEnd Menu Riwayat (b)	26
Gambar 4 8 FrontEnd Menu Riwayat (e).....	27
Gambar 4 9 FrontEnd Menu Jadwal (b).....	28
Gambar 4 10 FrontEnd Menu Jadwal (e)	29
Gambar 4 11 FrontEnd Menu Bantuan (b).....	29
Gambar 4 12 FrontEnd Menu Bantuan (e).....	30
Gambar 4 13 Dashboard Adafruit.IO.....	40
Gambar 4 14 Feed Adafruit.io	40
Gambar 4 15 Perakitan Alat	41
Gambar 4 16 Diagram Hasil Kuesioner	48

DAFTAR TABEL

Tabel 2 1 Penelitian Terdahulu	4
Tabel 4 1 Pengujian Jarak Alat Dengan Wifi Ponsel	45
Tabel 4 2 Pengujian Jarak Alat Dengan Wifi Router	45
Tabel 4 3 Pengujian Jarak Alat Dengan Aplikasi	46
Tabel 4 4 Kuesioner Kepuasan Pengguna	47
Tabel 4 5 Hasil Kuesioner Kepuasan Pengguna	48



BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi yang semakin maju sangat membantu sibuknya para kaum urban yang produktif, salah satunya adalah teknologi rumah pintar. *Smarthome* adalah integritas dari teknologi dan layanan Masyarakat melalui jaringan internet untuk meningkatkan kenyamanan Masyarakat dirumah. *Smarthome* memiliki banyak keunggulan yang dapat membantu anda dalam menghemat waktu dan tenaga saat berada dirumah. Lebih jelasnya *Smarthome* yang dimaksud adalah otomatisasi fungsi rumah yang membuat anda dapat mengontrol rumah dengan system online melalui platform. Dan tidak dapat dipungkiri hal ini dapat memberikan kesan lebih modern dan efisien.

Dengan adanya Android Studio sebagai perangkat lunak *Integrated Development Environment* (IDE) yang digunakan untuk mengembangkan aplikasi Android. Android Studio memiliki beberapa fitur yang cocok untuk pengembangan aplikasi *Smarthome*. Android Studio menggunakan Java dan Kotlin yang sama dengan yang digunakan pada system operasi android. Sehingga bisa membuat aplikasi yang memberikan fitur yang diinginkan yang diberikan nama EDITH.

EDITH merupakan aplikasi *Smarthome* dengan sistem kendali berupa menghidupkan dan mematikan lampu, menghitung dan memonitoring pengeluaran daya, dan juga penjadwalan lampu. Dengan adanya EDITH yang memanfaatkan *IoT* dengan basis android memungkinkan pengguna atau konsumen mampu mengendalikan rumah dari jarak jauh dan juga didalam rumah. EDITH sebagai aplikasi android mampu mengirimkan informasi terhadap ESP32 yang berfungsi sebagai mikrokontroler yang dirancang untuk membuat sistem aplikasi *Internet of Things* (*IoT*) dengan Adafruit.io sebagai server.

Adafruit.io adalah platform *cloud* yang memungkinkan pengguna untuk memantau, merespon, dan berinteraksi dengan input atau output sensor secara *real-time*. Adafruit.io juga merupakan API (*Application Programming Interface*) yang dirancang untuk membangun aplikasi *Internet of Things (IoT)*.

Latar belakang pembuatan aplikasi smarthome EDITH, yakni dari pengalaman pribadi dan juga mimpi penulis. Dari pengalaman pribadi sendiri disebabkan sering lupa mematikan lampu ruangan saat bepergian, hal ini menyebabkannya pengeluaran juga meningkat. Tidak hanya saat bepergian, saat pulang kerumah terkadang kita sudah mager untuk menghidupkan lampu jika sudah tertidur. Mimpi dari penulis sendiri didasari latar belakang dan dari ilmu yang didapatkan sebelumnya. Dengan adanya mata kuliah pembelajaran pembuatan aplikasi dan mikrokontroller, penulis memutuskan untuk membuat sebuah aplikasi smarthome yang diberi nama EDITH. EDITH sendiri dibuat dengan aplikasi perangkat lunak Android Studio sebagai wadah pembuatan. Dan sistem aplikasi dari alat untuk penyambungan aplikasi digunakan Board ESP-32 sebagai mikrokontroller dan diprogram dengan aplikasi perangkat lunak Arduino IDE.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, terdapat beberapa masalah yang perlu dipecahkan, yaitu:

1. Bagaimana merancang dan membuat aplikasi *Smarthome* dengan berbasis *IoT*?
2. Bagaimana membuat aplikasi yang mudah digunakan?
3. Bagaimana merancang aplikasi dengan fitur penjadwalan dan penghitungan energi

1.3. Tujuan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dijelaskan, sehingga dapat disimpulkan tujuan permasalahan, yaitu :

1. Membuat aplikasi Smarthome dengan berbasis IoT sesuai perancangan

2. Membuat sistem alat lampu dengan Board-ESP 32 sehingga terhubung dengan Adafruit.io
3. Membuat aplikasi smarthome dengan fitur menghidupkan dan mematikan lampu jarak jauh, melihat status lampu, mengetahui pengeluaran lampu, dan penjadwalan lampu.

1.4. Batasan Masalah

Batasan masalah yang harus di garis bawahi yakni :

1. Aplikasi smarthome hanya berfokus sistem alat lampu didalam rumah.
2. Aplikasi yang dibuat menggunakan aplikasi Android Studio.
3. *Mikrokontroler* diprogram menggunakan Android IDE.

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

Penulis membuat tinjauan pustaka dengan merujuk pada penelitian sebelumnya yang relevan dengan topik penelitian ini. Di bawah ini terdapat hasil dari penelitian yang telah dilakukan sebelumnya:

Tabel 2 1 Penelitian Terdahulu

No.	Judul	Hasil	Kelemahan
1	Sistem Kendali Smart home berbasis <i>Internet of Things</i> (IoT) dengan menggunakan mikrokontroller sederhana dan belum ada sistem penghitungan dan penjadwalan.	Smart home berbasis <i>Internet of Things</i> (IoT) dengan menggunakan mikrokontroller Arduino, <i>Ethernet shield</i> , modul relay, sensor api, sensor suhu (LM35), sensor gas (MQ6), dan sensor magnetik mampu dikendalikan dan dimonitor dari jarak jauh menggunakan smartphone android selama terkoneksi dengan internet.	Tampilan masih terlihat sederhana dan belum ada sistem penghitungan dan penjadwalan.
2	Sistem Kendali Lampu Berbasis <i>IoT</i> Menggunakan Aplikasi Blynk 2.0 Dengan Modul <i>Nodemcu Esp8266</i>	Dari hasil pengujian sistem dapat diketahui bahwa sistem secara keseluruhan bekerja dengan baik. Sistem kendali berfungsi dengan baik pada jangkauan internet dengan jarak maksimal 25 meter. Saat aplikasi dijalankan lampu menyala. Sistem monitoring dapat membaca tegangan dengan	Memiliki jangkauan sejauh 25 meter hal ini membuat aplikasi hanya bisa digunakan dalam keadaan dekat dan tidak bisa digunakan dalam keadaan jarak jauh

baik di mana saat input aplikasi Blynk 2.0 dimulai (on), aplikasi dapat membaca output tegangan sebesar 12V dan ditampilkan dilayar *smart phone*. Respon yang dihasilkan cukup cepat walaupun terkadang terdapat jeda waktu atau delay saat sistem merespon pada saat aplikasi sistem kendali dijalankan. Hal ini salah satunya dipengaruhi oleh tingkat stabilitas koneksi internet.

3	Penerapan <i>Internet of Things (IoT)</i> Pada Sistem Kendali Lampu Berbasis Mobile	Berdasarkan uraian analisa dan pembahasan, maka disimpulkan dengan memanfaatkan jaringan internet berbasis <i>Internet of Things (IoT)</i> sistem ini berhasil mengontrol perangkat lampu melalui koneksi jaringan internet. <i>Internet of Things (IoT)</i> dari rancang bangun kendali lampu ini telah berhasil dilakukan dengan dua kondisi kendali tombol satu digunakan untuk menghidupkan satu lampu dan tombol dua digunakan untuk menghidupkan lampu secara bersamaan.	Penelitian ini masih belum adanya sistem penjelasan tentang bagaimana sistem ini mempertimbangkan efisiensi energi.
---	--	---	--

4	Perancangan pengendali Lampu Kantor Berbasis Internet of Thing	Berdasarkan kesimpulan dari penulis, lampu otomatis berbasis <i>nodemcu esp8266</i> ini memiliki Hasil Setelah menghubungkan semua komponen dan memprogram, langkah selanjutnya adalah melakukan percobaan. Dari percobaan di dapat sebuah hasil bahwasannya perangkat keras dan perangkat lunak bekerja serta saling merespon dengan baik.	Tidak menggunakan aplikasi tetapi menggunakan <i>google assistant</i> .
5	Prototype Smart Street Light System Berbasis Arduino Menggunakan Metode Fuzzy Logic	Berdasarkan penelitian maka Metode Fuzzy Logic berhasil diterapkan dan terintegrasi dengan modul arduino untuk menentukan tingkat intensitas cahaya pada <i>smart street light system</i> .	Belum adanya sistem monitoring keadaan lampu.
6	Perancangan Aplikasi Smart Home Menggunakan Esp32 Berbasis Android	Pada penelitian ini didapatkan kesimpulan bahwa, fitur pembacaan tegangan, arus, dan daya pada Aplikasi Android dapat membaca tegangan, arus, dan daya pada alat melalui komunikasi <i>Bluetooth</i> .	Rancangan masih menggunakan komunikasi <i>Bluetooth</i> dan aplikasi yang digunakan terlihat sederhana.
7	Rancangan Bangun Smart Home Berbasis	kesimpulan yang dapat diambil dari Rancangan Bangun Smart Home Berbasis <i>IoT</i>	Konsep IFTTT (<i>If This Then That</i>) Dengan ESP8266 dan Google

<i>IoT</i> Menggunakan Konsep IFTTT (If This Then That) Dengan Konsep Ifttt (If This Then That) Dengan <i>Esp8266</i> Dan <i>Google Assistant</i>	Menggunakan Konsep IFTTT (If This Then That) Dengan <i>ESP8266</i> dan <i>Google Assistant</i> yakni untuk menyalakan dan mematikan lampu berhasil di bangun dan berjalan dengan baik dan sangat efektif untuk membantu mengatasi pengguna dalam aktifitas sehari hari untuk menyalakan dan mematikan lampu tentu hanya dengan <i>smart phone</i> yang ada pada genggamannya.	Assistant yang dirancang ini ada fitur menyalakan dan mematikan lampu, namun belum terdapat fitur monitoring lampu apakah dalam kondisi nyala atau mati dan menambah fitur timing yaitu mengatur kapan lampu untuk menyala dan kapan juga untuk mati.
<i>Prototype Smart Home Berbasis IoT Dengan Handphone Android Menggunakan Nodemcu Esp32</i>	Berdasarkan hasil maka dapat diperoleh kesimpulan bahwa Lampu dapat dikontrol dari jarak jauh dan terdapat dua sistem pengontrolan yaitu secara <i>IoT</i> dan secara manual, pengontrolan secara manual hanya bisa digunakan ketika mengaktifkan mode <i>Hardware</i> / mode manual. Ketika wifi terputus pengontrolan secara <i>IoT</i> atau lewat Blynk tidak bisa digunakan, maka disarankan untuk selalu mengaktifkan mode <i>hardware</i> agar <i>prototype smarthome</i> ini dapat dikontrol secara manual.	belum terdapat fitur monitoring lampu apakah dalam kondisi nyala atau mati.

Kesimpulan yang saya ambil dari hasil kajian pustaka yang telah mencakup berbagai penelitian tentang sistem *Smarthome* berbasis *IoT*, terdapat beberapa fitur unggulan yang ditemukan. Di antaranya adalah keberhasilan penggunaan mikrokontroler untuk menghidupkan dan mematikan lampu, pemanfaatan sistem *IoT* dalam pengembangan proyek, serta pembuatan aplikasi *Smarthome* yang berhasil. Namun, penelitian-penelitian tersebut juga memiliki beberapa kekurangan yang dapat diidentifikasi.

Kelemahan yang dapat diidentifikasi dari tinjauan pustaka atau penelitian terdahulu meliputi:

- Beberapa penelitian mungkin masih memiliki jangkauan jarak untuk bisa mengoperasikan aplikasi *Smarthome* yang mengakibatkan kurang efisien dan terbatas
- Beberapa penelitian mungkin kurang memfokuskan interface pengguna aplikasi android *Smarthome*, sehingga membuat pengalaman pengguna yang baik dan mampu memudahkan pengguna
- Beberapa penelitian belum menyediakan layanan pemantauan energi daya, sehingga pengguna tidak menyadari berapa biaya atau daya KWh yang telah digunakan oleh lampu tersebut. Beberapa penelitian belum menyediakan layanan riwayat penggunaan lampu, sehingga pengguna tidak menyadari berapa lama lampu telah digunakan dan tidak mengetahui kapan lampu hidup dan mati.
- Beberapa penelitian belum menyediakan sistem penjadwalan lampu, sehingga pengguna tidak perlu lagi memadamkan lampu secara manual jika adanya sistem penjadwalan lampu maka lampu akan otomatis hidup dan mati dengan aturan jam dan menit yang telah ditentukan sebelumnya.

2.2. Keunggulan Aplikasi EDITH

Penelitian ini menawarkan beberapa keunggulan yang menonjol dibandingkan penelitian sebelumnya, dengan memberikan nilai tambah signifikan bagi pengguna. Keunggulan-keunggulan tersebut meliputi:

- Penghitungan energi yang terpakai, aplikasi mampu menghitung jumlah energi listrik yang digunakan oleh lampu, membantu pengguna dalam mengelola konsumsi daya secara efisien.
- Penghitungan durasi lampu menyala, menyediakan informasi tentang berapa lama lampu telah menyala, sehingga pengguna dapat memantau durasi penggunaan secara real-time.
- Riwayat penggunaan lampu, mencatat kapan lampu dinyalakan dan dimatikan, memberikan transparansi penggunaan dan data historis yang dapat diakses kapan saja.
- Pengalaman antarmuka pengguna (*User Interface*) yang lebih baik, menawarkan desain antarmuka yang lebih intuitif, memudahkan pengguna dalam mengoperasikan aplikasi dengan nyaman.
- Akses dan pengendalian jarak jauh, memungkinkan pengguna untuk mengontrol semua fitur aplikasi tanpa batasan jarak, memberikan fleksibilitas tinggi dalam pengelolaan rumah pintar.
- Fitur penjadwalan otomatis, mendukung penjadwalan waktu lampu hidup dan mati secara otomatis, sehingga pengguna tidak perlu melakukannya secara manual.

Dengan kombinasi fitur unggulan ini, aplikasi Smarthome EDITH menawarkan inovasi dan fungsi yang lebih baik dibandingkan penelitian atau aplikasi Smarthome sebelumnya.

2.3. *Internet of Things (IoT)*

Internet of Things (IoT) merupakan teknologi untuk mengendalikan, mengatur, mentransfer data dari perangkat keras (*hardware*) melalui jaringan internet. Teknologi ini memungkinkan manusia melakukan kegiatan tersebut secara otomatis. Dalam implementasinya, pada *IoT* terdapat konektivitas antara perangkat keras, *IoT* platform/*IoT software*, data, dan koneksi internet sebagai media komunikasi dan kontrol pada perangkat tersebut. Perangkat keras pada *IoT* terdiri dari mikrokontroler sebagai otak, perangkat yang dikendalikan dan perangkat lain sebagai alat pengendali.

Internet of Things (IoT) berupa perangkat keras (Raspberry Pi) yang dapat tersambung dengan internet dengan tujuan untuk memperluas jaringan internet yang terhubung secara menyeluruh pada *hardware*. *Internet of Things (IoT)* bisa dimanfaatkan pada gedung untuk mengendalikan peralatan elektronik seperti lampu ruangan yang dapat dioperasikan dari jarak jauh melalui jaringan komputer, tidak dapat dipungkiri kemajuan teknologi yang sedemikian cepat harus bisa dimanfaatkan, dipelajari serta diterapkan dalam kehidupan sehari-hari.

2.4. Aplikasi Smarthome

Smarthome adalah aplikasi gabungan antara teknologi dan pelayanan yang dikhususkan pada lingkungan rumah dengan fungsi tertentu yang bertujuan meningkatkan keamanan, efisiensi dan kenyamanan penghuninya. Pembuatan *Smarthome* menggunakan android studio, karena android studio memiliki beberapa fitur yang cocok untuk pengembangan aplikasi *Smarthome*. Android Studio menggunakan Java dan Kotlin yang sama dengan yang digunakan pada sistem operasi android.

Pemilihan aplikasi perangkat lunak Android studio sebagai media tempat pembuatan aplikasi *Smarthome* EDITH dikarenakan android studio memiliki banyak fitur yang diperlukan dalam pembuatan dan pengembangan aplikasi baru. Android Studio juga mampu untuk mempercepat proses pengembangan aplikasi *mobile* sesuai dengan kebutuhan pihak pengembang. Adanya *tools* yang tersedia sangat mudah dipahami dan dipelajari bagi pemula sekalipun. Bahasa pemrograman yang *native* untuk melakukan penulisan kode program atau *coding* dari awal hingga akhir sesuai dengan kebutuhan dari pihak pengembang.

2.5. Adafruit.IO

Adafruit IO adalah platform *IoT* berbasis *cloud* yang dikembangkan oleh Adafruit untuk menghubungkan perangkat, menyimpan data, dan berinteraksi dengan perangkat melalui internet. Platform ini memungkinkan pengguna untuk mengirim, menyimpan, dan memvisualisasikan data sensor serta mengendalikan perangkat dari jarak jauh melalui antarmuka web atau aplikasi mobile. Adafruit IO mendukung protokol komunikasi MQTT (*Message Queuing Telemetry Transport*)

dan HTTP (*Hypertext Transfer–Transfer Protocol*), yang memudahkan integrasi dengan perangkat *IoT* seperti ESP32, Raspberry Pi, dan lainnya.

Pemilihan Adafruit.IO sebagai media yang menyediakan basis *cloud* untuk membantu pembuatan aplikasi EDITH dikarenakan Adafruit.IO memiliki *dashboard visual* yang mudah untuk membuat *dashboard* interaktif untuk memantau data perangkat dan komunikasi perangkat yang *real-time* dan efisien. Kompatibilitas dengan banyak perangkat mikrokontroler seperti ESP32, Arduino, Raspberry-Pi, dan perangkat IoT lainnya.

2.6. Arduino IDE

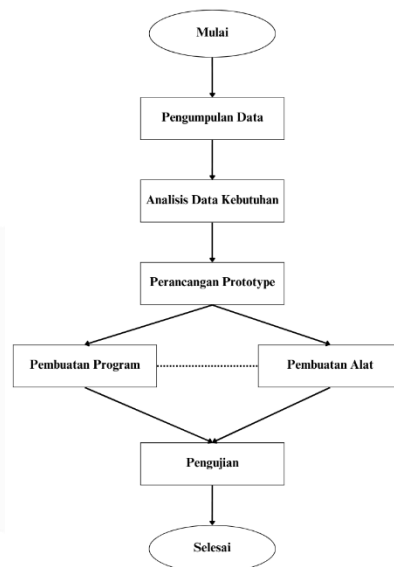
Arduino IDE (*Integrated Development Environment*) merupakan software yang khusus digunakan untuk merancang program melalui Arduino, dengan bahasa lain Arduino IDE menjadi sebuah media untuk melakukan program board Arduino. Arduino IDE berfungsi untuk *editor text* guna mengedit, membuat, dan memvalidasi sebuah kode program.

Pemilihan Arduino IDE dalam pembuatan banyak sebuah *software board* pemrograman yang tersedia di banyak situs dan fungsinya sama. Akan tetapi, penggunaan Arduino dipilih oleh banyak kalangan dikarenakan mempunyai alasan yang meningkatkan mikrokontroler menjadi lebih istimewa.

BAB III METODE PELAKSANAAN

3.1. Model Rancangan

Tahapan penelitian terdiri dari empat tahapan dimulai dari analisis sistem, perancangan *prototype Smarthome*, pembuatan program dan alat, dan implementasi dan pengujian.



Gambar 3 1 Model Rancangan

3.1.1. Pengumpulan Data

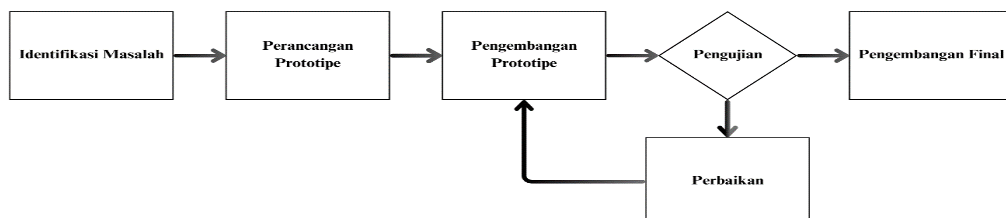
Pengumpulan data yang digunakan yakni mengumpulkan hasil teknologi sebelumnya dari laporan maupun jurnal serta dengan mengumpulkan data dari pengamatan langsung. Hal ini bertujuan agar mampu memberikan data yang kongkrit dan cepat dari teknologi sebelumnya. Agar teknologi sebelumnya dapat dikembangkan dan juga memberikan sentuhan lebih modern dan efisien dan juga melihat kelemahan kelemahan dari teknologi sebelumnya. Serta, mendapatkan informasi bahkan data yang bermanfaat untuk penelitian ini.

3.1.2. Analisis data kebutuhan

Setelah data terkumpul, tahap analisis kebutuhan dilakukan untuk mengidentifikasi masalah atau kebutuhan yang ada dalam sistem. Analisis sistem akan menganalisis data yang telah dikumpulkan dan memahami kebutuhan pengguna serta tujuan organisasi. Dalam tahap ini, analisis sistem juga akan mengidentifikasi kesenjangan antara sistem yang ada dengan kebutuhan yang sebenarnya. Hal ini akan menjadi dasar untuk merancang solusi yang tepat guna.

3.1.3. Perancangan *Prototype*

Perancangan *prototype* memberikan kita gambaran fitur-fitur yang digunakan dalam sebuah aplikasi dari rangkuman analisa data sebelumnya. Dengan metode *prototype*, tim pengembang dapat menciptakan prototipe awal yang mencerminkan konsep dan fitur produk yang diinginkan. Berikut adalah tahapan metode *prototype* secara umum:



Gambar 3 2 Perancangan *Prototype*

3.1.4. Pembuatan Program Dan Alat

Pembuatan program aplikasi membutuhkan 2 aplikasi Software yakni Android Studio untuk membuat aplikasi *Smarthome* dan Arduino IDE untuk pemrograman NodeMCU ESP32 yang digunakan sebagai mikrokontroler penerapan sistem lampu. Tahap ini harus saling terikat, walau terpisah tapi tetaplah harus sama. Dalam pembuatan sistem automasi

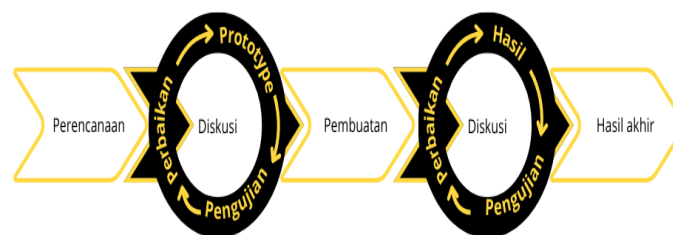
lampu rumah diperlukan perangkat keras dan perangkat lunak agar sistem dapat bekerja sesuai kebutuhan dan fungsinya. Perangkat keras nantinya akan dirakit sedemikian rupa dan akan dihubungkan dengan perangkat lunak sebagai media pengontrol sistem.

3.1.5. Uji Coba

Pada proses akhir ini sebagai bukti akan terlaksananya tujuan dalam program dan fungsi fungsi fitur yang di rancang sedemikian rupa. Dalam uji coba tentu memiliki syarat yakni berupa dokumentasi dan juga laporan dari hasil uji coba agar hasil dari rancangan valid dan terbukti.

3.2. Metode *Rapid Application Development*

Metode pelaksanaan menggunakan terapkan dalam pembuatan aplikasi pada penelitian ini didasarkan pada prinsip-prinsip metode *Rapid Application Development*. Metode *Rapid Application Development* (RAD) yakni metode pengembangan yang mengutamakan kecepatan dalam pembuatan ataupun pengembangan sebuah aplikasi dengan kualitas yang baik. Hal ini dikarenakan penulis sekaligus *developer* atau pihak pengembang akan bekerja sama dengan pihak kedua untuk memberikan masukan dan permintaan sehingga tercipta aplikasi yang cepat dan terarah. Metode ini harus mengutamakan kedua pihak dengan satu tujuan dan akan andil secara bersamaan pada pengembangan sehingga komunikasi antar keduanya akan lebih cepat dan baik. Berikut skema alur metode *Rapid Application Development* yang digunakan dalam metode penelitian dalam pembuatan aplikasi *Smarthome* EDITH.

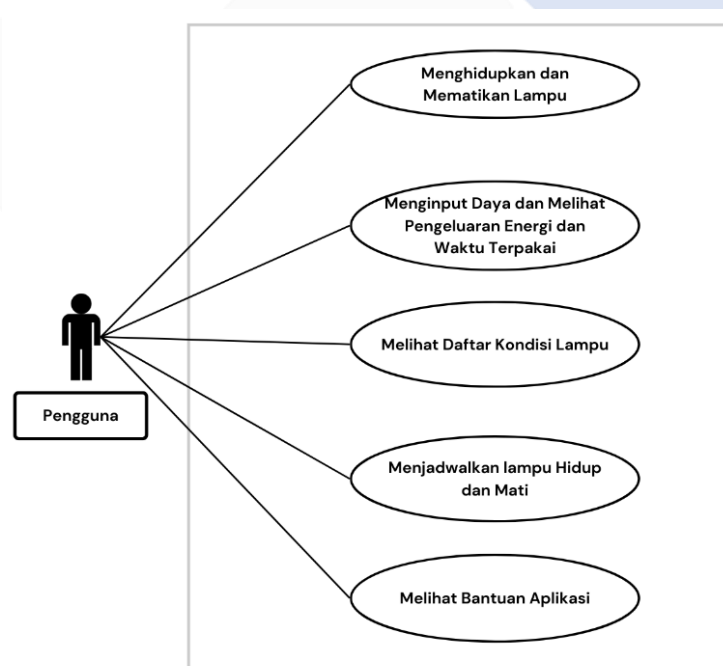


Gambar 3 3 *Rapid Application Development*

Pada proses metode di atas kita dapat melihat kolaborasi dan andil dalam 2 pihak yakni, melalui perencanaan aplikasi dan memasuki diskusi dengan melihat kelayakan dan fungsi dari *prototype*, jika sekiranya telah mendapatkan kesempatan maka pihak pengembang akan membuat atau merancang dari hasil kesempatan tersebut. Setelah hasil kesempatan telah dibuat maka terdapat diskusi lagi guna untuk pengecekan dan diskusi akhir, jika terjadi kekurangan akan diperbaiki dan jika terjadi kesepakatan maka hasil akhir telah didapatkan dengan baik.

3.3. Use Case Diagram

Pemodelan diagram *use case* untuk pengguna pada aplikasi *Smarthome* dengan memanfaatkan IoT, untuk mengidentifikasi interaksi antara pengguna dan sistem aplikasi.



Gambar 3 4 Use Case Diagram

3.4. Analisis Kebutuhan

Terdapat beberapa *hardware* dan *software* yang digunakan untuk membuat aplikasi *Smarthome* berbasis android, berikut uraian *hardware* dan softwarennya.

3.4.1. *Hardware*

Aplikasi smarhome memerlukan beberapa komponen *hardware* atau perangkat keras, antara lain:

a) Lampu

Komponen ini digunakan sebagai media pembuktian sistem aplikasi *Smarthome*.

b) Kabel

Komponene ini digunakan untuk menghubungkan komponen satu dengan komponen lainnya.

c) NodeMCU ESP 32

Komponen ini digunakan sebagai mikrokontroler sistem alat.

d) Fitting lampu

Komponen ini sebagai tempat wadah lampu.

e) Downloader USB

Komponen ini digunakan sebagai tempat sambungan untuk mengupload program Arduino IDE ke dalam ESP-32.

f) Power Suply

Komponen untuk memasokkan daya Listrik ke komponen.

3.4.2. *Software*

Aplikasi smarhome memerlukan beberapa komponen *hardware* atau perangkat keras, antara lain:

a) *Integrated Development Environment*(IDE)

IDE yang digunakan untuk membuat aplikasi ini adalah Android Studio, karena Android Studio sendiri menyediakan lingkungan pengembangan yang memfasilitasi pembuatan aplikasi *Smarthome* menggunakan bahasa pemrograman Kotlin.

b) Bahasa pemrograman Kotlin

Bahasa pemrograman Kotlin digunakan untuk membuat dan mengembangkan aplikasi smartlock dengan sidik jari.

c) Sistem operasi android

Aplikasi *Smarthome* dirancang untuk berjalan pada sistem operasi Android.

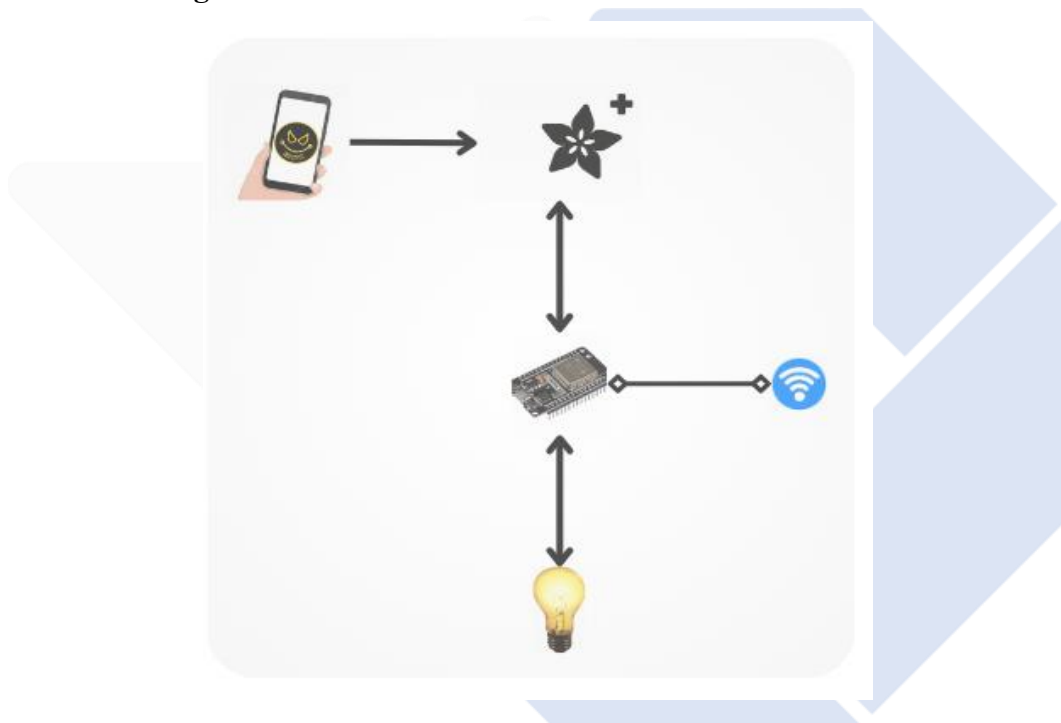
d) Android Software Development Kit (SDK)

SDK Android menyediakan alat dan API yang diperlukan untuk mengembangkan aplikasi Android, termasuk aplikasi *Smarthome*. Untuk aplikasi kami menggunakan API versi 34.

e) Server *cloud*

Sistem aplikasi *Smarthome* dihubungkan dengan Adafruit.io sebagai tempat informasi dan pengolahan data.

3.5. Perancangan Sistem



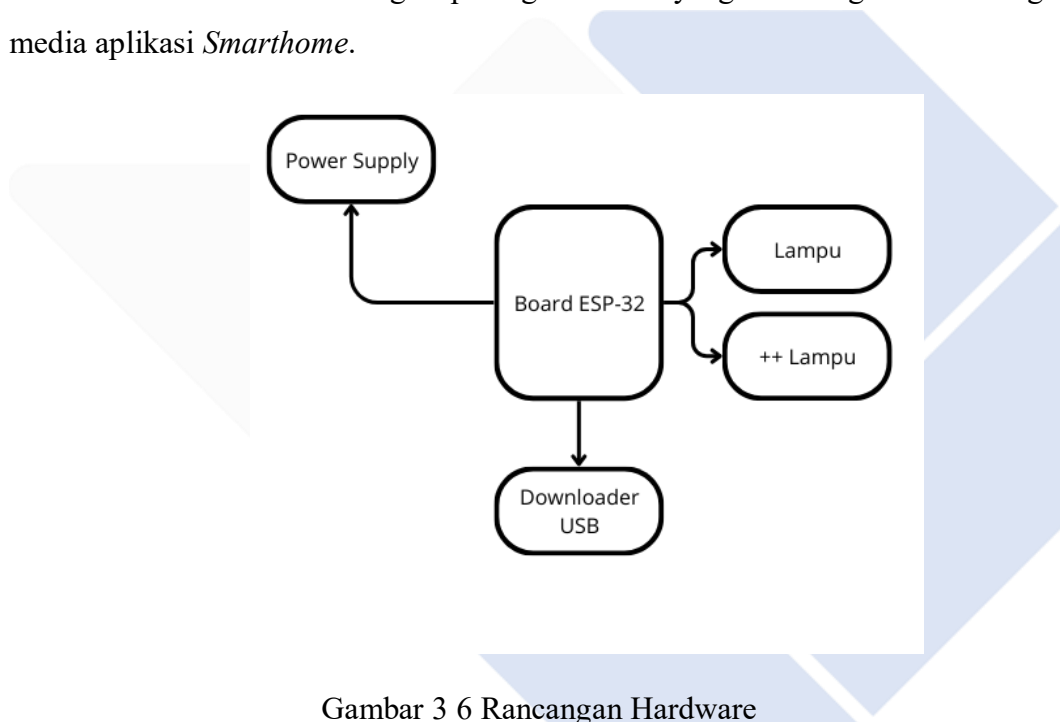
Gambar 3 5 Perancangan Sistem

Pada tahap ini berisi tentang rancangan sistem *Smarthome* berbasis *IoT*. Diawali dengan Aplikasi *Smarthome* yang diberi nama EDITH. Dimana pengguna akan meroperasikan EDITH maka EDITH akan terhubung dengan Adafruit.io yang bertugas sebagai server sebagai tempat penyimpanan. Pengguna akan memberikan perintah kepada EDITH yang akan dikirimkan ke Adafruit.io dan diteruskan ke NodeMCU ESP-32 yang online melalui wifi yang telah terhubung sebelumnya. ESP-32 akan mengoperasikan apa saja perintah dari EDITH yang dikirimkan oleh Adafruit.io. kemudian sebagai timbal balik dari proses ini akan ditampilkan di layar

handphone di aplikasi EDITH dan juga pop up. EDITH memiliki akses antara lain mampu melihat riwayat lampu beroperasi sehingga mampu mengetahui kapan hidup dan mati, EDITH dapat mengetahui energi Kwh yang digunakan dan waktu lampu dihidupkan, EDITH dapat mengatur jadwal perhari hidup dan mati lampu, dan juga dapat memberikan cara kerja operasi sistem EDITH itu sendiri kepada pengguna

3.6. Rancangan *Hardware*

Perancangan *Hardware* dan alur kerja dari sistem *Smarthome* menggunakan berbasis *IoT*. Berikut rancangan perangkat keras yang akan digunakan sebagai media aplikasi *Smarthome*.



Gambar 3 6 Rancangan Hardware

3.7. Pengumpulan dan Pengujian Data

Jenis metode pengujian data yang digunakan adalah metode gabungan. Untuk pengujian alat kami menggunakan metode kuantitatif berupa uji coba keberhasilan penghidupan lampu serta mematikan lampu dan untuk metode aplikasi kami menggunakan metode kualitatif berupa survey kuesioner. Berikut merupakan metode dari pengumpulan dan pengujian data yang diperlukan untuk aplikasi *Smarthome*.

3.7.1. Fungsionalitas

Pengujian fungsionalitas yang kami lakukan adalah suatu proses evaluasi yang bertujuan untuk memastikan bahwa alat yang diteliti beroperasi sesuai dengan fungsinya dan memenuhi standar yang ditetapkan.

3.7.2. Kuesioner

Pengujian kuesioner yang kami lakukan melibatkan evaluasi terhadap pertanyaan-pertanyaan yang diajukan kepada responden dan analisis terhadap jawaban yang diberikan. Dalam metode penelitian ini, terlihat bahwa kuesioner digunakan untuk mengukur kepuasan pengguna terhadap alat dan aplikasi *Smarthome* EDITH.



BAB IV

HASIL DAN PEMBAHASAN

4.1. Pembuatan Aplikasi *FrontEnd*

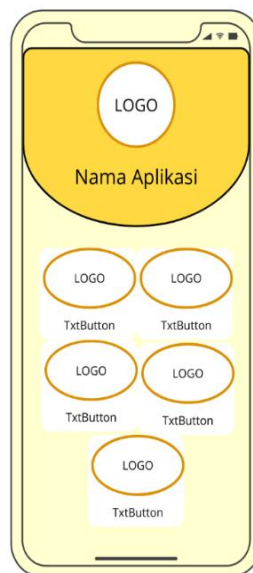
Pada kali ini peneliti akan melakukan pembuatan fitur pada aplikasi *Smarthome*, berikut pengembangan fitur

4.1.1. *FrontEnd* : Menu Utama

a) Rencana

Pada tahap ini yaitu tahap perancangan menu utama memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu utama mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu utama menampilkan menu fitur yang ada dalam aplikas.

b) Desain



Gambar 4 1 *FrontEnd* Menu Utama (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan 5 *icon* dengan 5 *textButton*. Pada *textButton* menampilkan fitur apa saja disajikan dalam aplikasi EDITH.

c) Pengembangan

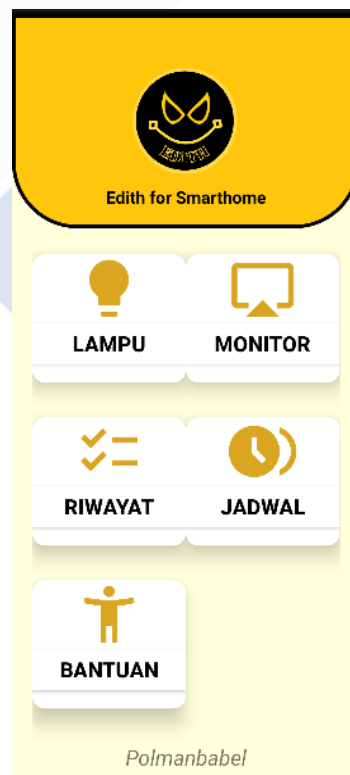
Di tahap *develop* atau pengembangan ini Dimana tahap pengkodean tampilan menu utama di rancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek *software* Android Studio. Dengan bahasa pemrograman Kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat error atau *force close* saat menjalankan menu utama.

e) Peluncuran

Sesi terakhir dari pembuatan menu utama ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu utama siap untuk digunakan dengan baik.



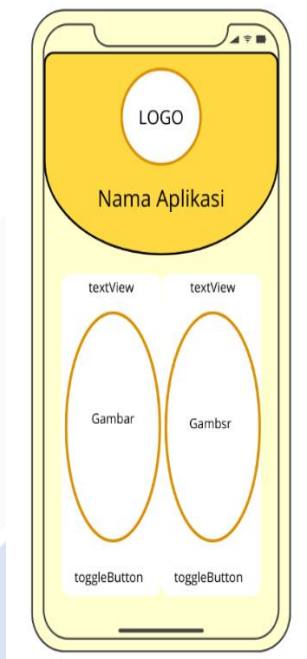
Gambar 4 2 *FrontEnd* Menu Utama (e)

4.1.2. *FrontEnd* : Menu Lampu

a) Rencana

Pada tahap ini yaitu tahap perancangan menu lampu memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu lampu mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu lampu menampilkan tempat menghidupkan dan mematikan lampu.

b) Desain



Gambar 4 3 *FrontEnd* Menu Lampu (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan 2 gambar dengan 2 *toggleButton*. Pada *toggleButton* akan membuat tampilan lampu jika toglleButton ditekan saat “Hidupkan” maka gambar akan berubah, dan begitu sebaliknya.

c) Pengembangan

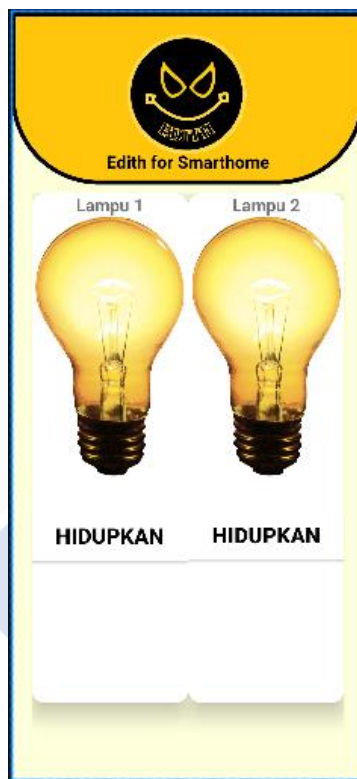
Di tahap *develop* atau pengembangan ini Dimana tahap pengkodean tampilan menu lampu di rancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek *software* android studio. Dengan Bahasa pemrograman kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat *error* atau *force close* saat menjalankan menu lampu.

e) Peluncuran

Sesi terakhir dari pembuatan menu lampu ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu monitor siap untuk digunakan dengan baik.



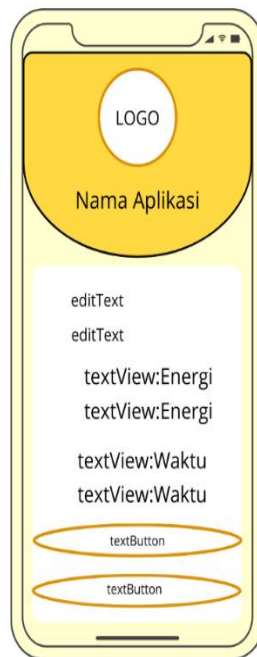
Gambar 4 4 *FrontEnd* Menu Lampu (e)

4.1.3. *FrontEnd* : Menu Monitor

a) Rencana

Pada tahap ini yaitu tahap perancangan menu monitor memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu monitor mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu monitor menampilkan input daya lampu sehingga mampu menghitung energi dan menampilkan waktu lamanya penggunaan lampu.

b) Desain



Gambar 4 5 *FrontEnd* Menu Monitoring (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan 2 inputan data dengan 4 textview dengan fungsi menampilkan energi dan waktu dan juga 2 tombol *textButton*. Pada textView Energi dipengaruhi dari inputan yang diberikan serta waktu yang tertera. Pada tombol *textButton* berfungsi sebagai tempat reset waktu nanti.

c) Pengembangan

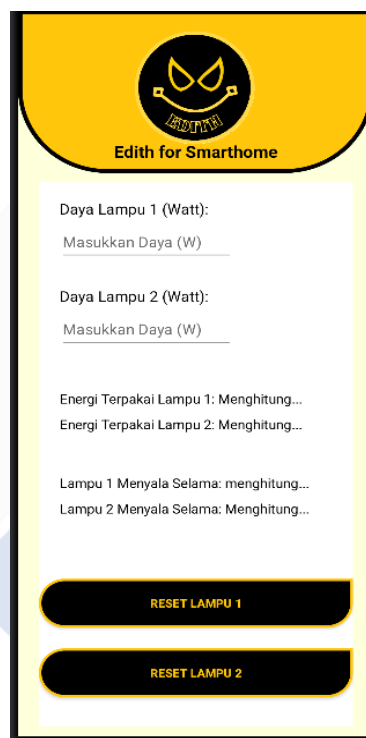
Di tahap *develop* atau pengembangan ini Dimana tahap pengkodean tampilan menu monitor dirancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek software android studio. Dengan Bahasa pemrograman Kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat *error* atau *force close* saat menjalankan menu monitor.

e) Peluncuran

Sesi terakhir dari pembuatan menu monitor ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu monitor siap untuk digunakan dengan baik.



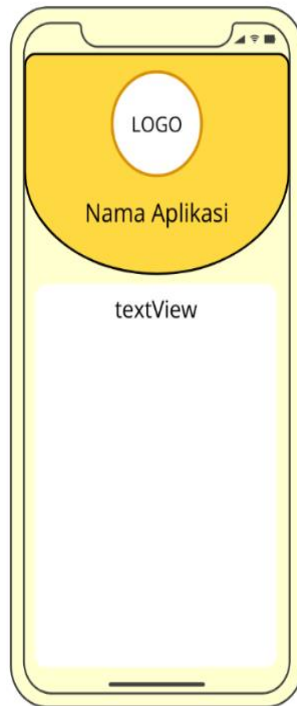
Gambar 4 6 *FrontEnd* Menu Monitoring (e)

4.1.4. *FrontEnd* : Menu Riwayat

a) Rencana

Pada tahap ini yaitu tahap perancangan menu riwayat memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu riwayat mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu riwayat menampilkan Riwayat kapan lampu hidup dan kapan lampu mati.

b) Desain



Gambar 4 7 *FrontEnd* Menu Riwayat (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan *textView* dengan *scrollView*. Dengan *scrollView* tampilan ini bisa memberikan daftar Riwayat lampu baik sedikit maupun banyak nantinya.

c) Pengembangan

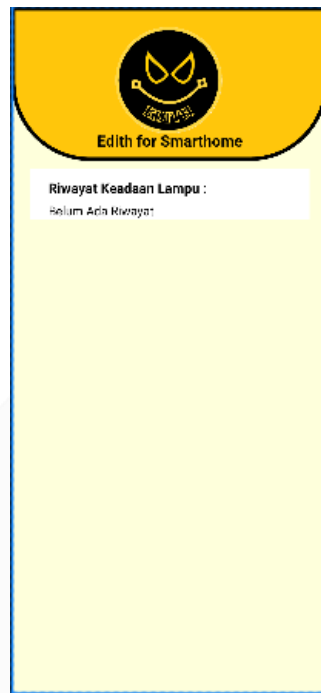
Di tahap *develop* atau pengembangan ini Dimana tahap pengkodean tampilan menu riwayat di rancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek software Android Studio. Dengan Bahasa pemrograman Kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat error atau *force close* saat menjalankan menu riwayat.

e) Peluncuran

Sesi terakhir dari pembuatan menu jadwal ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu riwayat siap untuk digunakan dengan baik.



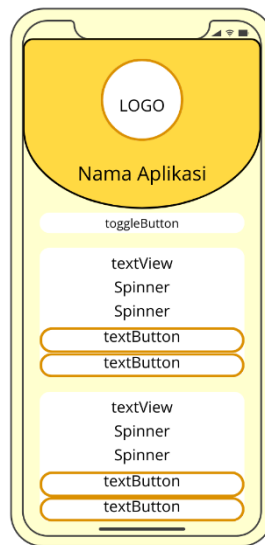
Gambar 4 8 FrontEnd Menu Riwayat (e)

4.1.5. FrontEnd : Menu Jadwal

a) Rencana

Pada tahap ini yaitu tahap perancangan menu jadwal memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu jadwal mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu jadwal menampilkan settingan penjadwalan lampu kapan matinya dan hidupnya.

b) Desain



Gambar 4 9 *FrontEnd* Menu Jadwal (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan 2 blok dengan 2 spinner, 2 *textButton* disetiap blok tersebut, dan 1 *toggleButton*.

c) Pengembangan

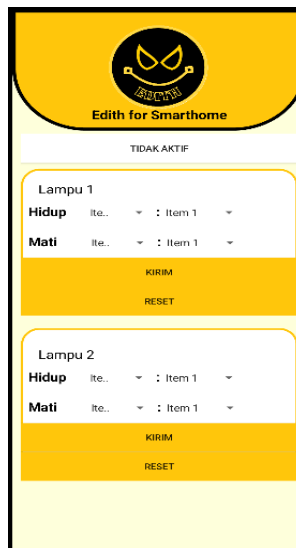
Di tahap *develop* atau pengembangan ini dimana tahap pengkodean tampilan menu jadwal di rancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek *software* android studio. Dengan Bahasa pemrograman kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat error atau *force close* saat menjalankan menu jadwal.

e) Peluncuran

Sesi terakhir dari pembuatan menu jadwal ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu jadwal siap untuk digunakan dengan baik.



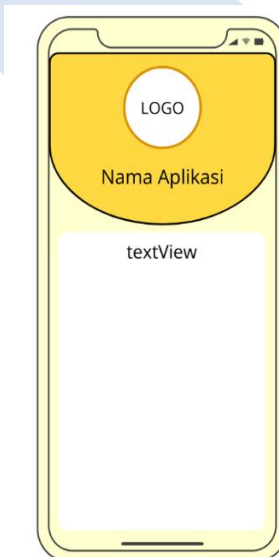
Gambar 4 10 *FrontEnd* Menu Jadwal (e)

4.1.6. *FrontEnd* : Menu Bantuan

a) Rencana

Pada tahap ini yaitu tahap perancangan menu bantuan memfokuskan fitur apa saja yang diberikan dalam aplikasi. Menu bantuan mencakupi tampilan yang berisikan 2 bagian yakni *Header* dan juga *Body*. Menu bantuan menampilkan panduan fitur aplikasi.

b) Desain



Gambar 4 11 *FrontEnd* Menu Bantuan (b)

Desain antar muka dalam tampilan *Header* menampilkan logo EDITH sedangkan di tampilan *Body* menampilkan *textView* dengan *scrollView*. Dengan *scrollView* tampilan ini bisa memberikan daftar bantuan fitur dalam aplikasi.

c) Pengembangan

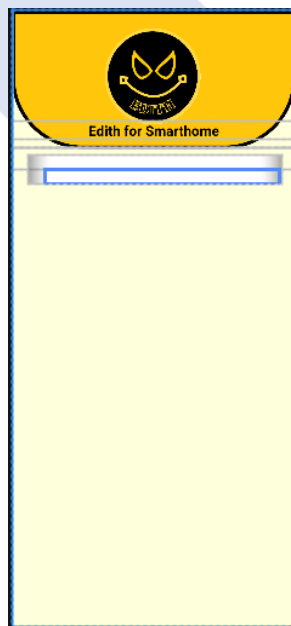
Di tahap *develop* atau pengembangan ini dimana tahap pengkodean tampilan menu bantuan di rancang mengikuti desain *wireframe* sebelumnya. Pembuatan desain ditulis dalam kode xml dalam proyek software Android Studio. Dengan Bahasa pemrograman Kotlin untuk menjalankan *FrontEnd* program.

d) Pengujian

Pada sesi pengujian kami memastikan bahwa sistem dari aplikasi berfungsi dengan baik dan lancar. Dengan menguji apabila terdapat error atau *force close* saat menjalankan menu bantuan.

e) Peluncuran

Sesi terakhir dari pembuatan menu bantuan ini. menetapkan dan luncurkan untuk menambahkan fitur ini ke aplikasi. dan akan pastikan lagi bahwa fitur menu bantuan siap untuk digunakan dengan baik.



Gambar 4 12 *FrontEnd* Menu Bantuan (e)

4.2. Pembuatan Program *Back End*

4.2.1. *Back End* : Menu Utama

a) Tujuan

Menu utama merupakan tampilan awal dengan tujuan memberikan Gambaran menu lainnya dan sebagai gerbang ke menu lainnya. Menu utama mempunyai 5 menu di dalamnya yakni lampu, monitor, riwayat, jadwal, dan bantuan.

b) Program

Menu utama memiliki tugas utama yakni sebagai tempatnya fitur yang disediakan.

```
val light = findViewById<Button>(R.id.btnlight)
light.setOnClickListener {
    intent = Intent (this,menulight::class.java)
    startActivity(intent)
}

val monitoring = findViewById<Button>(R.id.btnmonitor)
monitoring.setOnClickListener {
    intent = Intent (this,menumonitoring::class.java)
    startActivity(intent)
}

val help = findViewById<Button>(R.id.btnhelp)
help.setOnClickListener {
    intent = Intent (this,menuhelp::class.java)
    startActivity(intent)
}

val history = findViewById<Button>(R.id.btnhistory)
history.setOnClickListener {
    intent = Intent (this,menuhistory::class.java)
    startActivity(intent)
}

val schedule = findViewById<Button>(R.id.btnschedule)
schedule.setOnClickListener {
    intent = Intent (this,menuschedule::class.java)
    startActivity(intent)
}
```

4.2.2. Back End : Menu Lampu

a) Tujuan

Menu lampu merupakan tampilan dengan tujuan memberikan kendali dengan tombol lampu untuk mematikan dan menghidupkan secara manual. Menu lampu mempunyai 2 gambar dan terikat dengan button, sehingga memberikan kesan menampilkan keadaan lampu sendiri dan memiliki pop up.

b) Program

Menu lampu memiliki 2 inti program yakni program perubahan foto dengan *toggleButton* dan respon Adafruit.io.

```
tglbtn0.setOnCheckedChangeListener { _, isChecked ->
    if (isChecked) {
        lampu0.setImageResource(R.drawable.onlight)
        Toast.makeText(applicationContext, "Switch ON Lampu 1",
        Toast.LENGTH_SHORT).show()
        sendToAdafruitIO("relay1", "Lamp1 On")
    } else {
        lampu0.setImageResource(R.drawable.offlight)
        Toast.makeText(applicationContext, "Switch OFF Lampu 1",
        Toast.LENGTH_SHORT).show()
        sendToAdafruitIO("relay1", "Lamp1 Off")
    }
}

tglbtn1.setOnCheckedChangeListener { _, isChecked ->
    if (isChecked) {
        lampu1.setImageResource(R.drawable.onlight)
        Toast.makeText(applicationContext, "Switch ON Lampu 2",
        Toast.LENGTH_SHORT).show()
        sendToAdafruitIO("relay2", "Lamp2 On")
    } else {
        lampu1.setImageResource(R.drawable.offlight)
        Toast.makeText(applicationContext, "Switch OFF Lampu 2",
        Toast.LENGTH_SHORT).show()
        sendToAdafruitIO("relay2", "Lamp2 Off")
    }
}
```

```
override fun onResponse(call: Call, response: Response) {
    response.use {
        if (response.isSuccessful) {
```

```

val responseBody = response.Body?.string()
responseBody?.let {
    val jsonObject = JSONObject(it)
    val value = jsonObject.getString("value")

    runOnUiThread {
        when (value) {
            "Lamp1 On" -> {
                toggleButton.isChecked = true
                imageView.setImageResource(onDrawable)
            }
            "Lamp1 Off" -> {
                toggleButton.isChecked = false
                imageView.setImageResource(offDrawable)
            }
            "Lamp2 On" -> {
                toggleButton.isChecked = true
                imageView.setImageResource(onDrawable)
            }
            "Lamp2 Off" -> {
                toggleButton.isChecked = false
                imageView.setImageResource(offDrawable)
            }
            else -> {
                Log.e("AdafruitIO", "Unexpected value: $value")
            }
        }
    }
} else {
    val errorBody = response.Body?.string()
    runOnUiThread {
        Toast.makeText(this@menulight, "Failed to fetch data:
        ${response.code}", Toast.LENGTH_SHORT).show()
        Log.e("AdafruitIO", "Fetch failed with status code:
        ${response.code}, Error: $errorBody")
    }
}
}
}

```

4.2.3. Back End : Menu Monitor

a) Tujuan

Menu monitor merupakan tampilan dengan tujuan memberikan gambaran energi yang digunakan oleh lampu dengan memasukkan daya dan berapa lama lampu digunakan. Menu monitor juga memberi tombol reset lampu yang bertujuan untuk mereset ulang waktu lampu jika ingin menggantikan daya lampu.

b) Program

Menu Monitor memiliki 3 fungsi utama yakni penghitungan energi, penghitungan jangka lama lampu, dan tombol reset.

```
if (lastState != isOn) {
    val currentTime = getCurrentFormattedTime()
    when (lampNumber) {
        1 -> {
            lastLamp1State = isOn
            if (isOn) {
                // Save the start time when the lamp turns ON
                lamp1StartTime = System.currentTimeMillis()
                sharedPreferences.edit().putLong("lamp1_start_time",
lamp1StartTime!!).apply()
                addToLampHistory("Lamp 1 ON at $currentTime")
            } else {
                // When the lamp turns OFF, calculate and save the total ON
time
                lamp1TotalOnTime += System.currentTimeMillis() -
(lamp1StartTime ?: 0)
                sharedPreferences.edit().putLong("lamp1_total_on_time",
lamp1TotalOnTime).apply()
                lamp1StartTime = null
                addToLampHistory("Lamp 1 OFF at $currentTime")
            }
            updateTimeDisplayAndEnergy(lampNumber, isOn)
        }
        2 -> {
            lastLamp2State = isOn
            if (isOn) {
                // Save the start time when the lamp turns ON
                lamp2StartTime = System.currentTimeMillis()
                sharedPreferences.edit().putLong("lamp2_start_time",
lamp2StartTime!!).apply()
                addToLampHistory("Lamp 2 ON at $currentTime")
            }
        }
    }
}
```

```

    } else {
        // When the lamp turns OFF, calculate and save the total ON
time
        lamp2TotalOnTime += System.currentTimeMillis() -
(lamp2StartTime ?: 0)
        sharedPreferences.edit().putLong("lamp2_total_on_time",
lamp2TotalOnTime).apply()
        lamp2StartTime = null
        addToLampHistory("Lamp 2 OFF at $currentTime")
    }
    updateTimeDisplayAndEnergy(lampNumber, isOn)
}
}

// Ensure the energy usage and time are updated regularly when the
lamp is on
if (isOn) {
    updateTimeDisplayAndEnergy(lampNumber, isOn)
}
}

private fun updateTimeDisplayAndEnergy(lampNumber: Int, isOn:
Boolean) {
    val startTime = when (lampNumber) {
        1 -> lamp1StartTime
        2 -> lamp2StartTime
        else -> null
    }

    // Calculate the total ON time
    val totalOnTime = when (lampNumber) {
        1 -> lamp1TotalOnTime + if (isOn && startTime != null)
System.currentTimeMillis() - startTime else 0
        2 -> lamp2TotalOnTime + if (isOn && startTime != null)
System.currentTimeMillis() - startTime else 0
        else -> 0
    }

    val seconds = (totalOnTime / 1000) % 60
    val minutes = (totalOnTime / (1000 * 60)) % 60
    val hours = (totalOnTime / (1000 * 60 * 60))

    val timeText = String.format("%02d:%02d:%02d", hours, minutes,
seconds)

```

```

val powerInput = when (lampNumber) {
    1 -> powerInputLamp1.text.toString().toDoubleOrNull() ?: 0.0
    2 -> powerInputLamp2.text.toString().toDoubleOrNull() ?: 0.0
    else -> 0.0
}

// Calculate energy usage (kWh) and cost in Rupiah
val energyUsage = (powerInput / 1000) * (totalOnTime / (1000 * 60 * 60).toDouble())
val cost = energyUsage * PRICE_PER_KWH

// Display energy usage and cost
when (lampNumber) {
    1 -> {
        timeOnTextViewLamp1.text = "Waktu Terpakai Lampu 1: $timeText"
        energyUsageTextViewLamp1.text = "Energi Digunakan 1: %.4f kWh, Harga: Rp %.2f".format(energyUsage, cost)
    }
    2 -> {
        timeOnTextViewLamp2.text = "Waktu Terpakai Lampu 2: $timeText"
        energyUsageTextViewLamp2.text = "Energi Digunakan 2: %.4f kWh, harga: Rp %.2f".format(energyUsage, cost)
    }
}

private fun resetLampTime(lampNumber: Int) {
    when (lampNumber) {
        1 -> {
            lamp1TotalOnTime = 0
            lamp1StartTime = System.currentTimeMillis()
            sharedPreferences.edit().putLong("lamp1_total_on_time", lamp1TotalOnTime).apply()
            Toast.makeText(this, "Lamp 1 time reset", Toast.LENGTH_SHORT).show()
        }
        2 -> {
            lamp2TotalOnTime = 0
            lamp2StartTime = System.currentTimeMillis()
            sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()
            Toast.makeText(this, "Lamp 2 time reset", Toast.LENGTH_SHORT).show()
        }
    }
}

```

```

    }
  }
}

```

4.2.4. *Back End* : Menu Riwayat

a) Tujuan

Menu riwayat merupakan tampilan dengan tujuan memberikan informasi tentang kapan lampu dihidupkan dan kapan lampu dimatikan. Pada *Body* menu riwayat menggunakan *scrollView*, guna untuk memberikan layanan informasi yang menyeluruh tanpa suatu kebatasan.

b) Program

Menu riwayat memiliki fitur menambahkan informasi Riwayat menyeluruh terus menerus daripada menghidupkan atau mematikan lampu.

```

private fun saveHistoryToSharedPreferences() {
    val editor = sharedPreferences.edit()
    val historySet = lampHistory.toSet() // Convert history to a Set

    editor.putStringSet("lamp_history", historySet)
    editor.apply()
}

private fun restoreHistoryFromSharedPreferences() {
    val savedHistory = sharedPreferences.getStringSet("lamp_history",
setOf()) ?: setOf()
    lampHistory.clear()
    lampHistory.addAll(savedHistory)

    val displayedHistory = lampHistory.joinToString(separator = "\n")
    historyTextView.text = displayedHistory
}

```

4.2.5. *Back End* : Menu Jadwal

a) Tujuan

Menu jadwal merupakan tampilan dengan tujuan memberikan opsi untuk menjadwalkan lampu secara otomatis. Penjadwalan memiliki akses untuk menghidupkan dan mematikan lampu secara harian.

b) Program

Menu jadwal memiliki 3 fitur yakni mengirimkan penjadwalan atau mereset dan juga mengaktifkan fungsi penjadwalan.

```
toggleButton.setOnCheckedChangeListener { _, isChecked ->
    enableDisableFeatures(isChecked, kirimLampu0, resetLampu0,
    kirimLampu1, resetLampu1)
    val toggleState = if (isChecked) "On" else "Off"
    sendToFeed("schedule", toggleState) // Kirim status On/Off ke Adafruit
    IO
}
```

```
private fun sendToFeed(feedKey: String, data: String) {
    val url = "$baseUrl/$feedKey/data"
    val client = OkHttpClient()
    val json = JSONObject().put("value", data).toString()
    val Body =
    RequestBody.create("application/json".toMediaTypeOrNull(), json)
    val request = Request.Builder()
        .url(url)
        .addHeader("X-AIO-Key", ioKey)
        .post(Body)
        .build()
    client.newCall(request).enqueue(object : Callback {
        override fun onFailure(call: Call, e: IOException) {
            runOnUiThread {
                Toast.makeText(this@menuschedule, "Failed to send data",
                Toast.LENGTH_SHORT).show()
            }
        }
        override fun onResponse(call: Call, response: Response) {
            runOnUiThread {
                Toast.makeText(this@menuschedule, "Data sent successfully",
                Toast.LENGTH_SHORT).show()
            }
        }
    })
}
```

```
private fun clearFeedData(feedKey: String) {
    val url = "$baseUrl/$feedKey/data"
    val client = OkHttpClient()
    val request = Request.Builder()
```

```

        .url(url)
        .addHeader("X-AIO-Key", ioKey)
        .delete()
        .build()
        client.newCall(request).enqueue(object : Callback {
            override fun onFailure(call: Call, e: IOException) {
                runOnUiThread {
                    Toast.makeText(this@menuschedule, "Failed to reset data",
                        Toast.LENGTH_SHORT).show()
                }
            }
            override fun onResponse(call: Call, response: Response) {
                runOnUiThread {
                    Toast.makeText(this@menuschedule, "Data reset
                    successfully", Toast.LENGTH_SHORT).show()
                }
            }
        })
    }

    private fun resetSpinners(vararg spinners: Spinner) {
        spinners.forEach { it.setSelection(0) }
    }
}

```

4.2.6. Back End : Menu Bantuan

a) Tujuan

Menu bantuan merupakan tampilan dengan tujuan memberikan informasi fitur – fitur didalam aplikasi EDITH. Bantuan informasi akan sesimple mungkin, guna memudahkan pengguna untuk mengetahui kinerja sistem aplikasi EDITH

b) Program

Menu bantuan memiliki fitur informasi sehingga menggunakan *scrollView*.

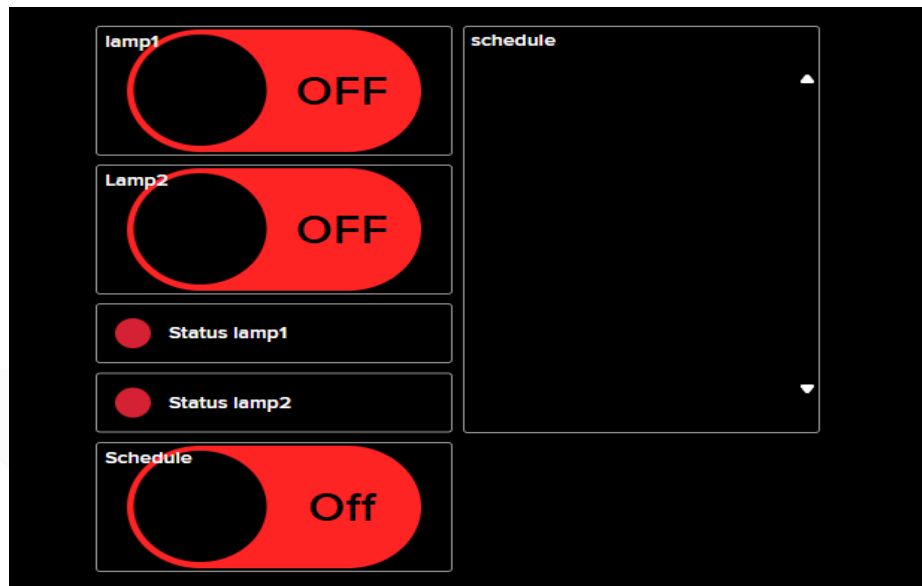
```
val helpText = findViewById<TextView>(R.id.helpTextView)
```

```
helpText.text = ""
```

Selamat datang di Aplikasi Smarthome! Berikut adalah panduan untuk membantu Anda menggunakan aplikasi:

4.3. Server Adafuit.IO

Adafruit IO yakni platform *IoT* yang dibangun menggunakan prinsip Protokol MQTT (*Message Queue Telemetry Transport*). Sedangkan MQTT yakni protokol ringan yang memungkinkan beberapa perangkat untuk terhubung ke server.



Gambar 4 13 *Dashboard* Adafruit.IO

Default				
Feed Name	Key	Last value	Recorded	
☐ lamp1	lamp1	0	less than a minute ago	
☐ lamp1_schedule	lamp1-schedule	02:14-02:30	34 minutes ago	
☐ lamp2	lamp2	0	less than a minute ago	
☐ lamp2_schedule	lamp2-schedule	02:28-02:31	34 minutes ago	
☐ relay1	relay1	Lamp1 Off	29 minutes ago	
☐ relay2	relay2	Lamp2 Off	29 minutes ago	
☐ schedule	schedule	Off	31 minutes ago	

Gambar 4 14 *Feed* Adafruit.io

4.4. Pembuatan Perangkat Keras (*Hardware*)

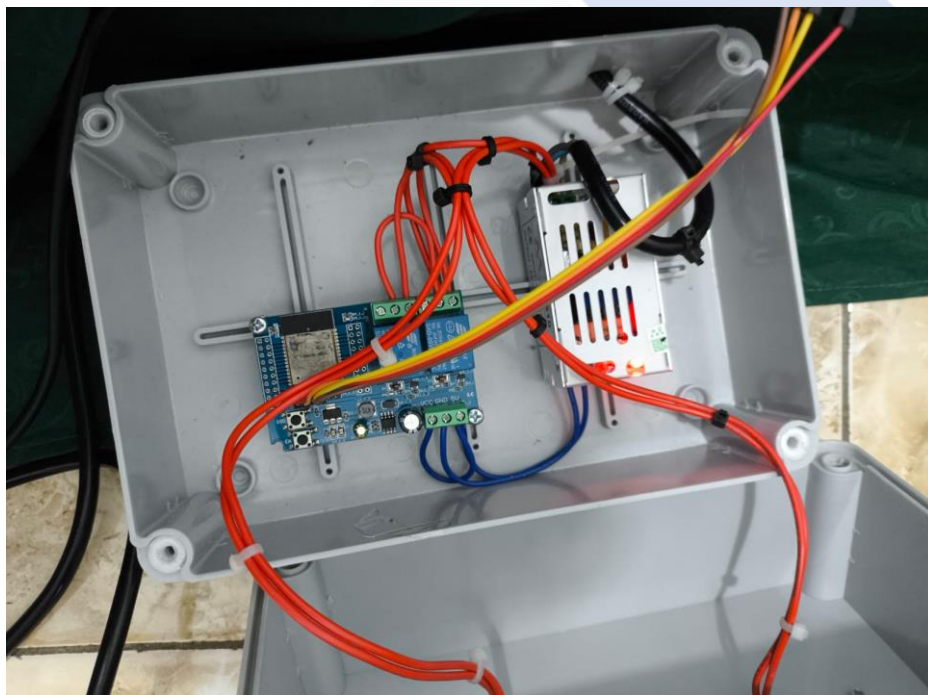
Pembuatan perangkat keras dimulai dari rancangan, perakitan, pemrograman, uji coba, dan peluncuran

4.4.1. Rancangan Model Alat

Sistem aplikasi *Smarthome* EDITH tentunya memiliki *hardware* ataupun perangkat keras. Desain tampilan alat secara keseluruhan dibuat menyerupai *prototype* simple terlebih dahulu dengan 2 lampu.

4.4.2. Perakitan Alat

Pembuatan rakitan alat mengikuti rangkaian perancangan sebelumnya yang telah di tentukan.



Gambar 4 15 Perakitan Alat

4.4.3. Pemrograman Arduino IDE

Pemrograman mikrokontroler ESP 32 menggunakan Arduino IDE, pemrograman ini bertujuan dengan fungsi utama yakni

1. Koneksi terhadap wifi

```
#define WIFI_SSID "POCO F6"  
#define WIFI_PASS "12345678"
```

2. Koneksi akun Adafruit.io

```
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883 // MQTT port
#define AIO_USERNAME "PnD"
#define AIO_KEY "aio_LjiC48U6JXWz3q7MJuFoR9Xt2aW0"
```

3. Pengecekan status penjadwalan atau tidak

```
void handleLampMode(String modeFeed, String scheduleFeed, int
relayPin, Adafruit_MQTT_Publish &statusFeed,
Adafruit_MQTT_Subscribe &relayFeed) {
    String mode = fetchFeedMode(modeFeed);
    if (mode == "On") {
        Serial.println("Mode otomatis aktif.");
        processSchedule(scheduleFeed, relayPin, statusFeed);
    } else {
        Serial.println("Mode manual aktif.");
        String relayCommand = fetchLastRelayCommand(relayFeed);
        if (relayCommand != "") {
            handleRelayCommand(relayCommand.c_str(), relayPin,
statusFeed);
        }
    }
}
```

4. Sistem alat penjadwalan

```
void processSchedule(String scheduleFeed, int relayPin,
Adafruit_MQTT_Publish &statusFeed) {
    String schedule = fetchSchedule(scheduleFeed);
    if (schedule != "") {
        int startTime, endTime;
        if (parseSchedule(schedule, startTime, endTime)) {
            time_t now = time(nullptr);
            struct tm timeinfo;
            localtime_r(&now, &timeinfo);

            int currentHour = timeinfo.tm_hour;
            int currentMinute = timeinfo.tm_min;
            int currentTime = currentHour * 60 + currentMinute;

            if (currentTime >= startTime && currentTime < endTime) {
                digitalWrite(relayPin, HIGH);
                statusFeed.publish("1");
            } else {
                digitalWrite(relayPin, LOW);
            }
        }
    }
}
```

```

        statusFeed.publish("0");
    }
}
}
}

String fetchSchedule(String feedName) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "https://io.adafruit.com/api/v2/" +
String(AIO_USERNAME) + "/feeds/" + feedName + "/data/last";
        http.begin(url);
        http.addHeader("X-AIO-Key", AIO_KEY);
        int httpStatusCode = http.GET();
        if (httpStatusCode > 0) {
            String payload = http.getString();
            Serial.println("Response: " + payload);
            DynamicJsonDocument doc(1024);
            deserializeJson(doc, payload);
            return doc["value"].as<String>(); // Extracting value
        }
        http.end();
    }
    return "";
}

bool parseSchedule(String schedule, int &startTime, int &endTime) {
    int separatorIndex = schedule.indexOf('-');
    if (separatorIndex > 0) {
        String startStr = schedule.substring(0, separatorIndex);
        String endStr = schedule.substring(separatorIndex + 1);

        startTime = parseTimeToMinutes(startStr);
        endTime = parseTimeToMinutes(endStr);
        return true;
    }
    return false;
}

int parseTimeToMinutes(String timeStr) {
    int separatorIndex = timeStr.indexOf(':');
    if (separatorIndex > 0) {
        int hours = timeStr.substring(0, separatorIndex).toInt();
        int minutes = timeStr.substring(separatorIndex + 1).toInt();
        return hours * 60 + minutes;
    }
}

```

```

    }
    return -1;
}

void handleRelayCommand(const char *command, int relayPin,
Adafruit_MQTT_Publish &statusFeed) {
    if (String(command).indexOf("On") >= 0) {
        digitalWrite(relayPin, HIGH);
        statusFeed.publish("1");
    } else if (String(command).indexOf("Off") >= 0) {
        digitalWrite(relayPin, LOW);
        statusFeed.publish("0");
    }
}

String fetchLastRelayCommand(Adafruit_MQTT_Subscribe &relayFeed)
{
    if (relayFeed.lastread != nullptr) {
        return String((char *)relayFeed.lastread);
    }

    return "";
}

```

4.5. Pengujian Sistem Aplikasi *Smarthome*

Pengujian sistem aplikasi *Smarthome* dilakukan agar dapat melihat dari aplikasi *Smarthome* EDITH, server Adafruit.io, dan juga alat lampu bisa tersambung sehingga mampu menjalankan tugasnya masing masing dan juga dapat melihat kegagalan ataupun kesalahan. Pengujian ini dilakukan dengan beberapa tahap yakni jarak wifi dengan alat dan juga jarak aplikasi dengan alat.

4.5.1. Pengujian jarak alat dan wifi

Pengujian ini menggunakan wifi dengan alat sehingga kita dapat melihat jarak dari *NodeMCU esp-32* dapat menangkap sinyal seberapa jauh dengan sinyal dari wifi. Pengujian akan dilakukan dengan metode yakni wifi ponsel dengan wifi router. Adapun jaringan yang digunakan yakni jaringan provider Telkomsel dan router yang digunakan yakni wifi router advan. Status yang ditampilkan terdapat di tampilan *pop up* di aplikasi.

Berikut table hasil dari pengujian jarak alat dan wifi ponsel.

Tabel 4 1 Pengujian Jarak Alat Dengan Wifi Ponsel

No.	Jarak	Status
1	5 Meter	Terkoneksi
2	10 Meter	Terkoneksi
3	15 Meter	Tidak Terkoneksi
4	20 Meter	Tidak Terkoneksi

Berikut table hasil dari pengujian jarak alat dan wifi router.

Tabel 4 2 Pengujian Jarak Alat Dengan Wifi Router

No.	Jarak	Status
1	5 Meter	Terkoneksi
2	10 Meter	Terkoneksi
3	15 Meter	Terkoneksi
4	20 Meter	Terkoneksi
5	25 Meter	Terkoneksi
6	30 Meter	Tidak Terkoneksi

Kesimpulan dari 2 pengujian jarak alat dengan wifi ponsel maupun router yakni, wifi router lebih diutamakan karena jarak lebih jauh sehingga lebih efisien untuk suatu rumah pintar mendatang.

4.5.2. Pengujian jarak alat dan aplikasi

Pengujian jarak alat dan aplikasi bertujuan untuk sejauh mana efektifitas dan sebagai inti permasalahan dari alat sebelumnya yang memiliki jarak. Adapun status yang digunakan akan ditampilkan dalam tampilan aplikasi dan pernyataan waktu delay kami menggunakan *timer stopwatch* dengan komunikasi 2 arah.

Berikut table hasil dari pengujian jarak alat dan aplikasi.

Tabel 4 3 Pengujian Jarak Alat Dengan Aplikasi

No.	Jarak	Status	Delay
1	5 Meter	Terkoneksi	1 Detik
2	50 Meter	Terkoneksi	1 Detik
3	1 Kilo Meter	Terkoneksi	1 Detik
4	3 Kilo Meter	Terkoneksi	1 Detik
5	7 Kilo Meter	Terkoneksi	2 Detik
6	900 Kilo Meter	Terkoneksi	1 Detik

Kesimpulan dari hasil tabel pengujian jarak alat dengan aplikasi yakni, aplikasi EDITH mampu mengakses dan mengirimkan perintah kepada alat lampu.

4.5.3. Kesimpulan Pengujian

Pengujian jarak alat menggunakan wifi ponsel maupun router lebih diutamakan wifi router karena memiliki jarak yang cukup luas. Sedangkan, pengujian jarak alat dengan aplikasi dapat diakses melalui jarak yang jauh. Sehingga inti permasalahan dari penelitian ini dianggap cukup berhasil karena mampu terkoneksi antara aplikasi dengan alat. Delay yang terjadi disebabkan Lokasi tempat pengujian cukup sulit sinyal sehingga memiliki jarak rentan waktu yang 1 cukup mengganggu. Jadi kuat sinyal mampu memengaruhi kinerja sistem aplikasi *Smarthome* EDITH,

4.6. Kuesioner

Kuesioner digunakan untuk mendapatkan sample pengalaman eksperimen pengguna, guna memberikan Gambaran kepuasan kinerja dan juga tampilan dari aplikasi dan sistem *Smarthome* EDITH. Berikut tabel kuesioner yang diberikan :

Tabel 4 4 Kuesioner Kepuasan Pengguna

No	Pertanyaan
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?
	1 (Sangat Tidak Setuju)
	2 (Tidak Setuju)
	3 (Netral)
	4 (Setuju)
	5 (Sangat Setuju)

Dari pertanyaan kuesioner diatas, diperoleh responden sebanyak 5 orang. Jawaban responden dapat dilihat pada lampiran dan tabel dibawah ini :

Tabel 4 5 Hasil Kuesioner Kepuasan Pengguna

No.	Responden	Hasil Jawaban							Hasil
		1	2	3	4	5	6	7	
1	Responden 1	4	5	4	5	5	4	5	32
2	Responden 2	5	5	3	3	4	4	4	28
3	Responden 3	4	5	3	5	5	4	4	30
4	Responden 4	3	5	4	5	4	3	3	27
5	Responden 5	3	5	5	4	5	4	4	30

Hasil perhitungan presentase hasil jawaban:

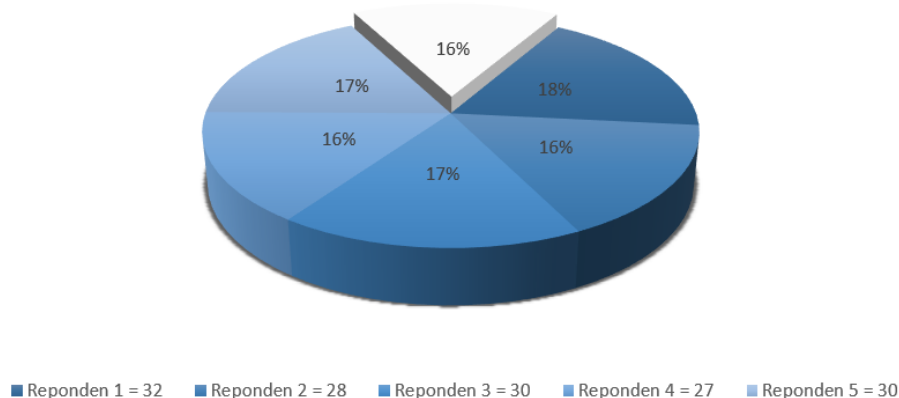
Total = 147

Nilai Presentase = $147/175 * 100 \%$

= 84%

Berikut diagram lingkaran hasil kuesioner dari table kepuasan pengguna diatas.

Diagram Hasil Kuesioner



Gambar 4 16 Diagram Hasil Kuesioner

Berdasarkan diagram lingkaran hasil kuesioner, responden terbagi ke dalam lima kategori dengan rincian sebagai berikut. Responden 1 (18%), Responden 2 (16%), Responden 3 (17%), Responden 4 (16%), dan Responden 5 (17%). Hasil keseluruhan menunjukkan bahwa terdapat nilai 3 ("Netral") sebanyak 7, nilai 4

("Puas") sebanyak 14, dan nilai 5 ("Sangat Puas") sebanyak 14, sedangkan nilai 1 dan 2 tidak ditemukan dalam data. Dengan demikian, tingkat kepuasan tinggi tercapai, di mana 84% responden memberikan tingkat kepuasan "Puas" atau lebih. Hal ini mencerminkan mayoritas responden memiliki persepsi yang positif terhadap aspek yang diukur dalam kuesioner.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Aplikasi *Smarthome* EDITH berbasis IoT berhasil dikembangkan menggunakan Android Studio dengan tingkat kepuasan pengguna mencapai 84%. Aplikasi ini menawarkan fungsi inovatif seperti pengendalian lampu jarak jauh, pemantauan energi yang terpakai, penghitungan durasi lampu menyala, serta riwayat penggunaan yang mencatat kapan lampu menyala dan dimatikan. Selain itu, fitur penjadwalan otomatis memberikan kemudahan bagi pengguna untuk mengatur jadwal lampu hidup dan mati sesuai kebutuhan harian. Keunggulan utama EDITH terletak pada kemampuannya diakses dari jarak jauh tanpa batas fisik dengan mikrokontroler.

5.2. Saran

- a) Aplikasi *Smarthome* EDITH masih belum mempunyai sistem penambahan lampu jadi jika ingin menambah lampu harus manual dan merubah pengkodean. Jadi diharapkan penemuan selanjutnya mampu memberikan fitur penambahan dan penghapusan lampu dari aplikasi langsung.
- b) Sistem alat *Smarthome* EDITH masih belum adanya sensor arus balik, sehingga jika lampu rusak atau Listrik padam waktu tetap berjalan diaplikasi. Jadi diharapkan penemuan selanjutnya mampu memberikan dan menambahkan sensor arus balik guna terjadi masalah yang sama.

DAFTAR PUSTAKA

- [1] Rizal, R., & Karyana, I. (2019). Sistem Kendali dan Monitoring pada Smart Home Berbasis Internet of Things (IoT). Volume 1, Nomor 2, halaman 43–50.
- [2] Herlina, A., Irfan Syahbana, M., Adi Gunawan, M., & Miftahul Rizqi, M. (2022). Sistem Kendali Lampu Berbasis Iot Menggunakan Aplikasi Blynk 2.0 Dengan Modul Nodemcu Esp8266. In Sains Teknik Elektro (Vol. 3, Issue 2).
- [3] Muzawi, R., & Kurniawan, W. J. (2018). Penerapan Internet of Things (IoT) Pada Sistem Kendali Lampu Berbasis Mobile. J-SAKTI (Jurnal Sains Komputer Dan Informatika), 2(2), 115, <https://doi.org/10.30645/j-sakti.v2i2.75>
- [4] Shafitri, A., & Mashuri, A. (2023). PERANCANGAN PENGENDALI LAMPU KANTOR BERBASIS INTERNET OF THING, <https://doi.org/10.30656/prosisko.v9i1.4672>.
- [5] Bimantoro, F. N., Subagio, R. T., & Sulhan, Muh. A. (2023). PROTOTYPE SMART STREET LIGHT SYSTEM BERBASIS ARDUINO MENGGUNAKAN METODE FUZZY LOGIC. JIKA (Jurnal Informatika), 7(3), 281, <https://doi.org/10.31000/jika.v7i3.7651>
- [6] Rupianto, I. A., Astutik, R. P., & Surya, Y. A. (2023). Perancangan Aplikasi Smart Home Menggunakan Esp32 Berbasis Android. Jurnal POLEKTRO: Jurnal Power Elektronik, Volume 12, Nomor 3, tahun 2019.
- [7] Haryanto, S., & Ramdhani Raharjo, M. (2022). RANCANGAN BANGUN SMART HOME BERBASIS IOT MENGGUNAKAN KONSEP IFTTT (IF THIS THEN THAT) DENGAN ESP8266 DAN GOOGLE ASSISTANT, <https://doi.org/10.32520/stmsi.v12i1.2118>
- [8] Rombekila, A., & Luoukelay Entamoing, B. (2022). PROTOTYPE SMART HOME BERBASIS IoT DENGAN HANDPHONE ANDROID MENGGUNAKAN NODEMCU ESP32. Jurnal Teknik AMATA, Volume 3, Nomor 1, tahun 2022.
- [9] Hidayatullah, R. N., Ramdhan, A., & Khamid, A. (2024). PENGEMBANGAN KENDALI LAMPU MENGGUNAKAN MIKROKONTROLER NODEMCU ESP32 DAN ARDUINO IDE BERBASIS INTERNET OF THINGS (IOT). In Jurnal Mahasiswa Teknik Informatika (Vol. 8, Issue 4).
- [10] Yoganatha, R., & Zakariyah, M. (2023). APLIKASI KONTROL LAMPU JARAK JAUH BERBASIS ANDROID. SemanTIK : Teknik Informasi, 9(2), 149. <https://doi.org/10.55679/semantik.v9i2.45360>
- [11] *Develop* Android apps with Kotlin, Android Studio [Online], diakses pada 10 oktober 2024, Available : Kotlin and Android | Android Developers .
- [12] Explore and Learn, Adafruit.io [Online], diakses pada 12 oktober 2024, Available : Adafruit Learning System.

LAMPIRAN

LAMPIRAN I : DAFTAR RIWAYAT HIDUP PENULIS

1. Data Pribadi

Nama Lengkap : Ibnu Kurniawan
Tempat & Tanggal Lahir : Penagan, 30 Maret 2000
Alamat Rumah : Desa Penagan, gang Bukit Besar
No *Handphone* : 0852 6722 5424
Email : ibnukurniawan345@gmail.com
Jenis Kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

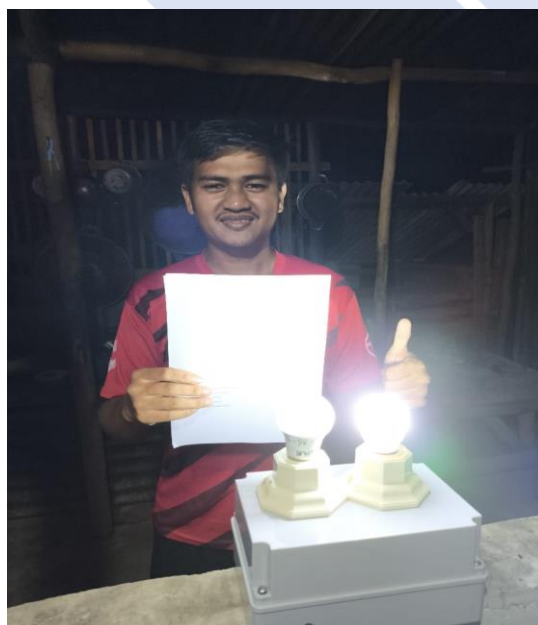
SD : 2006 – 2012
SMP : 2012 – 2015
SMA : 2015 – 2018

LAMPIRAN II : FOTO DAN HASIL KUESIONER

Nama : *R. Yandi*
Pendidikan Terakhir : *Sarjana Pertanian*
Pekerjaan : *Petani*

No	Pertanyaan	Nilai
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?	4
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?	5
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?	3
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?	5
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya	5
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?	4
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?	4

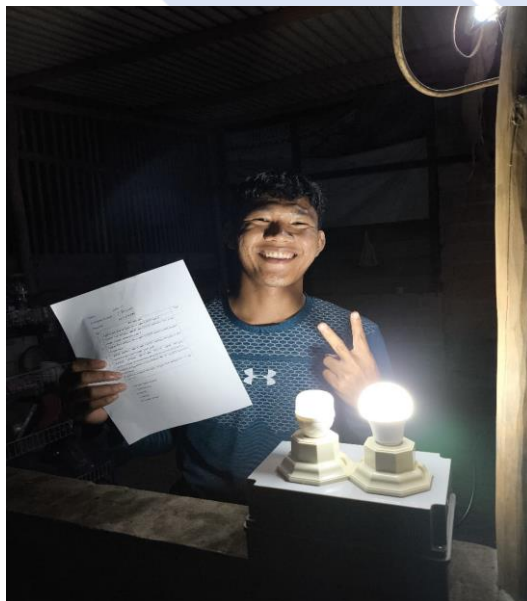
1 (Sangat Tidak Setuju)
2 (Tidak Setuju)
3 (Netral)
4 (Setuju)
5 (Sangat Setuju)



Nama : Sobirin
Pendidikan Terakhir : S1 Pertanian
Pekerjaan : wiraswasta

No	Pertanyaan	Nilai
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?	5
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?	5
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?	3
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?	3
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya	4
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?	4
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?	4

- 1 (Sangat Tidak Setuju)
2 (Tidak Setuju)
3 (Netral)
4 (Setuju)
5 (Sangat Setuju)



Nama : Gunawan
 Pendidikan Terakhir : S1
 Pekerjaan : Wirasaha

No	Pertanyaan	Nilai
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?	4
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?	5
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?	4
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?	5
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya	5
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?	4
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?	5

- 1 (Sangat Tidak Setuju)
- 2 (Tidak Setuju)
- 3 (Netral)
- 4 (Setuju)
- 5 (Sangat Setuju)



Nama : MARIAJI
 Pendidikan Terakhir : ST
 Pekerjaan : Operator Sekolah

No	Pertanyaan	Nilai
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?	5
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?	5
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?	4
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?	5
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya	4
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?	3
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?	3

- 1 (Sangat Tidak Setuju)
 2 (Tidak Setuju)
 3 (Netral)
 4 (Setuju)
 5 (Sangat Setuju)



Nama : VERTIKA MEIYANA
 Pendidikan Terakhir : S1
 Pekerjaan : PNS

No	Pertanyaan	Nilai
1	Apakah tampilan aplikasi <i>Smarthome</i> EDITH baik ?	3
2	Apakah fitur aplikasi <i>Smarthome</i> EDITH memiliki fitur yang sesuai dengan rancangan awal ?	5
3	Apakah fitur aplikasi <i>Smarthome</i> EDITH dapat berjalan dengan dengan baik ?	5
4	Apakah aplikasi <i>Smarthome</i> EDITH mampu mengakses dan mengirim perintah dengan jarak jauh ?	4
5	Apakah aplikasi <i>Smarthome</i> EDITH dapat mengatasi permasalahan permasalahan yang belum selesai sebelumnya	5
6	Apakah penggunaan sistem aplikasi <i>Smarthome</i> EDITH mudah digunakan?	4
7	Bagaimana tingkat kepuasan dengan fitur aplikasi <i>Smarthome</i> EDITH ?	4

- 1 (Sangat Tidak Setuju)
- 2 (Tidak Setuju)
- 3 (Netral)
- 4 (Setuju)
- 5 (Sangat Setuju)



LAMPIRAN KODE PROGRAM

1. Aplikasi Smarthome EDITH

- Activity-main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      android:id="@+id/main"
7      android:background="#feffdb"
8      android:layout_width="match_parent"
9      android:layout_height="match_parent"
10     tools:context=".MainActivity">
11
12     <View
13         android:id="@+id/top_view"
14         android:layout_width="match_parent"
15         android:layout_height="250dp"
16         android:background="@drawable/bg_view"
17     />
18     <ImageView
19         android:layout_width="match_parent"
20         android:layout_height="245dp"
21         android:src="@drawable/edith"
22         android:id="@+id/logo"
23     ></ImageView>
24     <TextView
25         android:layout_width="match_parent"
26         android:layout_height="match_parent"
27         android:text="Edith for Smarthome"
28         android:textSize="20sp"
29         android:textStyle="bold"
30         android:layout_marginTop="200dp"
31         android:gravity="center_horizontal"
32         android:textColor="@color/black">
33     </TextView>
34
35     <GridLayout
36         android:layout_width="match_parent"
37         android:layout_height="match_parent"
38         android:rowCount="3"
```

```

38         android:rowCount="3"
39         android:columnCount="2"
40         android:layout_marginTop="10dp"
41         android:layout_marginLeft="24dp"
42         android:layout_marginRight="24dp"
43         android:layout_below="@+id/top_view">
44
45         <androidx.cardview.widget.CardView
46             android:layout_width="150dp"
47             android:layout_height="150dp"
48             app:cardCornerRadius="15dp"
49             app:cardElevation="12dp"
50             android:layout_marginTop="20dp"
51             android:layout_columnWeight="1"
52             android:layout_row="0"
53             android:layout_column="0">
54             <LinearLayout
55                 android:layout_width="match_parent"
56                 android:layout_height="match_parent"
57                 android:orientation="vertical">
58                 <ImageView
59                     android:layout_width="80dp"
60                     android:layout_height="80dp"
61                     android:src="@drawable/light"
62                     android:id="@+id/light"
63                     android:layout_gravity="center_horizontal">
64                 </ImageView>
65                 <Button
66                     android:layout_width="match_parent"
67                     android:layout_height="wrap_content"
68                     android:text="Lampu"
69                     android:id="@+id/btnlight"
70                     android:background="@drawable/bg_btn"
71                     android:layout_gravity="center_horizontal"
72                     android:textSize="25sp"
73                     android:textStyle="bold"

```

```

73         android:textStyle="bold"
74         android:textColor="#000000">
75     </Button>
76 </LinearLayout>
77 </androidx.cardview.widget.CardView>
78 <androidx.cardview.widget.CardView
79     android:layout_width="150dp"
80     android:layout_height="150dp"
81     app:cardCornerRadius="15dp"
82     app:cardElevation="12dp"
83     android:layout_marginTop="20dp"
84     android:layout_columnWeight="1"
85     android:layout_row="0"
86     android:layout_column="1">
87     <LinearLayout
88         android:layout_width="match_parent"
89         android:layout_height="match_parent"
90         android:orientation="vertical">
91         <ImageView
92             android:layout_width="80dp"
93             android:layout_height="80dp"
94             android:src="@drawable/monitor"
95             android:layout_gravity="center_horizontal">
96         </ImageView>
97         <Button
98             android:layout_width="match_parent"
99             android:layout_height="wrap_content"
100             android:id="@+id/btnmonitor"
101             android:text="Monitor"
102             android:layout_gravity="center_horizontal"
103             android:textSize="25sp"
104             android:textStyle="bold"
105             android:background="@drawable/bg_btn"

```

```

103         android:textSize="25sp"
104         android:textStyle="bold"
105         android:background="@drawable/bg_btn"
106         android:textColor="#000000">
107     </Button>
108 </LinearLayout>
109 </androidx.cardview.widget.CardView>
110 <androidx.cardview.widget.CardView
111     android:layout_width="150dp"
112     android:layout_height="150dp"
113     app:cardCornerRadius="15dp"
114     app:cardElevation="12dp"
115     android:layout_marginTop="40dp"
116     android:layout_columnWeight="1"
117     android:layout_row="1"
118     android:layout_column="0">
119     <LinearLayout
120         android:layout_width="match_parent"
121         android:layout_height="match_parent"
122         android:orientation="vertical">
123         <ImageView
124             android:layout_width="80dp"
125             android:layout_height="80dp"
126             android:src="@drawable/history"
127             android:layout_gravity="center_horizontal">
128         </ImageView>
129         <Button
130             android:layout_width="match_parent"
131             android:layout_height="wrap_content"
132             android:id="@+id/btnhistory"
133             android:text="Riwayat"
134             android:layout_gravity="center_horizontal"
135             android:textSize="25sp"

```

```

134         android:layout_gravity="center_horizontal"
135         android:textSize="25sp"
136         android:background="@drawable/bg_btn"
137         android:textStyle="bold"
138         android:textColor="@color/black">
139     </Button>
140 </LinearLayout>
141 </androidx.cardview.widget.CardView>
142 <androidx.cardview.widget.CardView
143     android:layout_width="150dp"
144     android:layout_height="150dp"
145     app:cardCornerRadius="15dp"
146     app:cardElevation="12dp"
147     android:layout_marginTop="40dp"
148     android:layout_columnWeight="1"
149     android:layout_row="1"
150     android:layout_column="1">
151     <LinearLayout
152         android:layout_width="match_parent"
153         android:layout_height="match_parent"
154         android:orientation="vertical">
155         <ImageView
156             android:layout_width="80dp"
157             android:layout_height="80dp"
158             android:src="@drawable/schedule"
159             android:layout_gravity="center_horizontal">
160         </ImageView>
161         <Button
162             android:layout_width="match_parent"
163             android:layout_height="wrap_content"
164             android:id="@+id/btnschedule"
165             android:text="Jadwal"
166             android:layout_gravity="center_horizontal"
167             android:textSize="25sp"

```

```

167         android:textSize="25sp"
168         android:background="@drawable/bg_btn"
169         android:textStyle="bold"
170         android:textColor="@color/black">
171     </Button>
172 </LinearLayout>
173 </androidx.cardview.widget.CardView>
174 <androidx.cardview.widget.CardView
175     android:layout_width="150dp"
176     android:layout_height="150dp"
177     app:cardCornerRadius="15dp"
178     app:cardElevation="12dp"
179     android:layout_marginTop="40dp"
180     android:layout_columnWeight="1"
181     android:layout_row="2"
182     android:layout_column="0">
183     <LinearLayout
184         android:layout_width="match_parent"
185         android:layout_height="match_parent"
186         android:orientation="vertical">
187         <ImageView
188             android:layout_width="80dp"
189             android:layout_height="80dp"
190             android:src="@drawable/help"
191             android:layout_gravity="center_horizontal">
192         </ImageView>
193         <Button
194             android:layout_width="match_parent"
195             android:layout_height="wrap_content"
196             android:id="@+id/btnhelp"
197             android:text="Bantuan"
198             android:layout_gravity="center_horizontal"
199             android:textSize="25sp"

```

```

182         android:layout_column="0">
183         <LinearLayout
184             android:layout_width="match_parent"
185             android:layout_height="match_parent"
186             android:orientation="vertical">
187             <ImageView
188                 android:layout_width="80dp"
189                 android:layout_height="80dp"
190                 android:src="@drawable/help"
191                 android:layout_gravity="center_horizontal">
192             </ImageView>
193             <Button
194                 android:layout_width="match_parent"
195                 android:layout_height="wrap_content"
196                 android:id="@+id/btnhelp"
197                 android:text="Bantuan"
198                 android:layout_gravity="center_horizontal"
199                 android:textSize="25sp"
200                 android:background="@drawable/bg_btn"
201                 android:textStyle="bold"
202                 android:textColor="@color/black">
203             </Button>
204         </LinearLayout>
205     </androidx.cardview.widget.CardView>
206 </GridLayout>
207 <TextView
208     android:layout_width="match_parent"
209     android:layout_height="wrap_content"
210     android:layout_marginTop="850dp"
211     android:text="Polmanbabel"
212     android:textSize="25sp"
213     android:textStyle="italic"
214     android:gravity="center_horizontal">
215 </TextView>
216 </RelativeLayout>

```

- MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        enableEdgeToEdge()  
        Thread.sleep(millis: 3000)  
        installSplashScreen()  
        setContentView(R.layout.activity_main)  
        val light = findViewById<Button>(R.id.btnlight)  
        light.setOnClickListener {  
            intent = Intent ( packageContext: this,menulight::class.java)  
            startActivity(intent)  
        }  
        val monitoring = findViewById<Button>(R.id.btnmonitor)  
        monitoring.setOnClickListener {  
            intent = Intent ( packageContext: this,menumonitoring::class.java)  
            startActivity(intent)  
        }  
        val help = findViewById<Button>(R.id.btnhelp)  
        help.setOnClickListener {  
            intent = Intent ( packageContext: this,menuhelp::class.java)  
            startActivity(intent)  
        }  
        val history = findViewById<Button>(R.id.bthnhistory)  
        history.setOnClickListener {  
            intent = Intent ( packageContext: this,menuhistory::class.java)  
            startActivity(intent)  
        }  
        val schedule = findViewById<Button>(R.id.bthnschedule)  
        schedule.setOnClickListener {  
            intent = Intent ( packageContext: this,menuschedule::class.java)  
            startActivity(intent)  
        }  
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
            insets  
        }  
    }  
}
```

- activity_menulight.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:id="@+id/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:background="#feffdb"
10     tools:context=".menulight">
11     <View
12         android:id="@+id/top_view"
13         android:layout_width="match_parent"
14         android:layout_height="200dp"
15         android:background="@drawable/bg_viewmenu"
16     />
17     <ImageView
18         android:layout_width="match_parent"
19         android:layout_height="275dp"
20         android:layout_marginTop="-45dp"
21         android:src="@drawable/edith"
22         android:id="@+id/logo"
23     >
24     </ImageView>
25     <TextView
26         android:layout_width="match_parent"
27         android:layout_height="match_parent"
28         android:text="Edith for Smarthome"
29         android:textSize="20sp"
30         android:textStyle="bold"
31         android:layout_marginTop="160dp"
32         android:gravity="center_horizontal"
33         android:textColor="@color/black">
34     </TextView>
35     <GridLayout
36         android:layout_width="387dp"
37         android:layout_height="631dp"
38         android:layout_below="@+id/top_view"

```

```

38     android:layout_below="@+id/top_view"
39     android:layout_marginLeft="24dp"
40     android:layout_marginTop="10dp"
41     android:layout_marginRight="24dp"
42     android:columnCount="2"
43     android:rowCount="1">
44     <androidx.cardview.widget.CardView
45         android:layout_width="150dp"
46         android:layout_height="match_parent"
47         android:layout_row="0"
48         android:layout_column="0"
49         android:layout_columnWeight="1"
50         android:layout_marginBottom="40dp"
51         app:cardCornerRadius="15dp"
52         app:cardElevation="12dp">
53         <LinearLayout
54             android:layout_width="match_parent"
55             android:layout_height="match_parent"
56             android:orientation="vertical">
57             <TextView
58                 android:layout_width="match_parent"
59                 android:layout_height="wrap_content"
60                 android:gravity="center_horizontal"
61                 android:text="Lampu 1"
62                 android:textSize="20sp"
63                 android:textStyle="bold">
64             </TextView>
65             <ImageView
66                 android:id="@+id/lampu0"
67                 android:layout_width="match_parent"
68                 android:layout_height="300dp"
69                 android:layout_gravity="center_horizontal"
70                 android:src="@drawable/onlight"></ImageView>
71             <ToggleButton
72                 android:id="@+id/tglbtn0"
73                 android:layout_width="match_parent"

```

```

73         android:layout_width="match_parent"
74         android:layout_height="59dp"
75         android:layout_gravity="center_horizontal"
76         android:layout_marginTop="40dp"
77         android:background="#ffffff"
78         android:textColor="@color/black"
79         android:textOff="Hidupkan"
80         android:textOn="Matikan"
81         android:textSize="25sp"
82         android:textStyle="bold">
83     </ToggleButton>
84 </LinearLayout>
85 </androidx.cardview.widget.CardView>
86 <androidx.cardview.widget.CardView
87     android:layout_width="150dp"
88     android:layout_height="match_parent"
89     android:layout_row="0"
90     android:layout_column="1"
91     android:layout_columnWeight="1"
92     android:layout_marginBottom="40dp"
93     app:cardCornerRadius="15dp"
94     app:cardElevation="12dp">
95     <LinearLayout
96         android:layout_width="match_parent"
97         android:layout_height="match_parent"
98         android:orientation="vertical">
99         <TextView
100             android:layout_width="match_parent"
101             android:layout_height="wrap_content"
102             android:gravity="center_horizontal"
103             android:text="Lampu 2"
104             android:textSize="20sp"
105             android:textStyle="bold">
106     </TextView>

```

```

95     <LinearLayout
96         android:layout_width="match_parent"
97         android:layout_height="match_parent"
98         android:orientation="vertical">
99         <TextView
100             android:layout_width="match_parent"
101             android:layout_height="wrap_content"
102             android:gravity="center_horizontal"
103             android:text="Lampu 2"
104             android:textSize="20sp"
105             android:textStyle="bold">
106         </TextView>
107         <ImageView
108             android:id="@+id/lampu1"
109             android:layout_width="match_parent"
110             android:layout_height="300dp"
111             android:layout_gravity="center_horizontal"
112             android:src="@drawable/onlight"></ImageView>
113         <ToggleButton
114             android:id="@+id/tglbtn1"
115             android:layout_width="match_parent"
116             android:layout_height="57dp"
117             android:layout_gravity="center_horizontal"
118             android:layout_marginTop="40dp"
119             android:background="#ffffff"
120             android:textColor="@color/black"
121             android:textOff="Hidupkan"
122             android:textOn="Matikan"
123             android:textSize="25sp"
124             android:textStyle="bold">
125         </ToggleButton>
126     </LinearLayout>
127 </androidx.cardview.widget.CardView>
128 </GridLayout>
129 </RelativeLayout>

```

- menulight.kt

```
package com.example.program

import ...

class menulight : AppCompatActivity() {

    companion object {
        const val IO_USERNAME = "PnD"
        const val IO_KEY = "aio_LjiC48U6JXWz3q7MJvFoR9Xt2aW0"
    }

    // Initialize OkHttpClient with LoggingInterceptor
    private val client: OkHttpClient by lazy {
        val logging = HttpLoggingInterceptor()
        logging.setLevel(HttpLoggingInterceptor.Level.BODY)
        OkHttpClient.Builder()
            .addInterceptor(logging)
            .build()
    }

    private lateinit var lampu0: ImageView
    private lateinit var tglbtn0: ToggleButton
    private lateinit var lampu1: ImageView
    private lateinit var tglbtn1: ToggleButton
    @SuppressWarnings("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_menulight)
        lampu0 = findViewById(R.id.lampu0)
        tglbtn0 = findViewById(R.id.tglbtn0)
        lampu1 = findViewById(R.id.lampu1)
        tglbtn1 = findViewById(R.id.tglbtn1)
        // Set up toggle button listeners
        tglbtn0.setOnCheckedChangeListener { _, isChecked ->
            if (isChecked) {
                lampu0.setImageResource(R.drawable.onlight)
                Toast.makeText(applicationContext, text: "Switch ON Lampu 1", Toast.LENGTH_SHORT).show()
                sendToAdafruitIO( feedKey: "relay1", value: "Lamp1 On")
            } else {
                lampu0.setImageResource(R.drawable.offlight)
                Toast.makeText(applicationContext, text: "Switch OFF Lampu 1", Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

```

52     } else {
53         lampu0.setImageResource(R.drawable.offlight)
54         Toast.makeText(applicationContext, text: "Switch OFF Lampu 1", Toast.LENGTH_SHORT).show()
55         sendToAdafruitIO( feedKey: "relay1", value: "Lamp1 Off")
56     }
57 }
58 toggle1.setOnCheckedChangeListener { _, isChecked ->
59     if (isChecked) {
60         lampu1.setImageResource(R.drawable.onlight)
61         Toast.makeText(applicationContext, text: "Switch ON Lampu 2", Toast.LENGTH_SHORT).show()
62         sendToAdafruitIO( feedKey: "relay2", value: "Lamp2 On")
63     } else {
64         lampu1.setImageResource(R.drawable.offlight)
65         Toast.makeText(applicationContext, text: "Switch OFF Lampu 2", Toast.LENGTH_SHORT).show()
66         sendToAdafruitIO( feedKey: "relay2", value: "Lamp2 Off")
67     }
68 }
69
70 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
71     val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
72     v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
73     insets
74 }
75
76 // Fetch initial lamp states from Adafruit IO
77 fetchLampState( feedKey: "relay1", toggle1, lampu0, R.drawable.onlight, R.drawable.offlight)
78 fetchLampState( feedKey: "relay2", toggle1, lampu1, R.drawable.onlight, R.drawable.offlight)
79
80 // Method to send data to Adafruit IO
81 private fun sendToAdafruitIO(feedKey: String, value: String) {
82     Log.d( tag: "AdafruitIO", msg: "sendToAdafruitIO function triggered in menuLight")
83
84     val url = "https://io.adafruit.com/api/v2/$IO_USERNAME/feeds/$feedKey/data"
85     val json = "{\"value\":\"$value\"}"
86     val body: RequestBody = RequestBody.create("application/json; charset=utf-8".toMediaTypeOrNull(), json)
87     val request = Request.Builder()
88         .url(url)

```

```

87     val request = Request.Builder()
88         .url(url)
89         .post(body)
90         .addHeader("name: "X-AIO-Key", IO_KEY)
91         .build()
92
93     client.newCall(request).enqueue(object : Callback {
94         override fun onFailure(call: Call, e: IOException) {
95             runOnUiThread {
96                 Toast.makeText(context: this@menuLight, text: "Failed to send data to Adafruit IO", Toast.LENGTH_SHORT).show()
97                 Log.e(tag: "AdafruitIO", msg: "Data send failed: ${e.message}")
98             }
99         }
100
101         override fun onResponse(call: Call, response: Response) {
102             response.use {
103                 if (response.isSuccessful) {
104                     val responseBody = response.body?.string()
105                     runOnUiThread {
106                         Toast.makeText(context: this@menuLight, text: "Data sent to Adafruit IO", Toast.LENGTH_SHORT).show()
107                         Log.d(tag: "AdafruitIO", msg: "Data sent successfully, Response: $responseBody")
108                     }
109                 } else {
110                     val errorBody = response.body?.string()
111                     runOnUiThread {
112                         Toast.makeText(context: this@menuLight, text: "Failed to send data: ${response.code}", Toast.LENGTH_SHORT).show()
113                         Log.e(tag: "AdafruitIO", msg: "Data send failed with status code: ${response.code}, Error: $errorBody")
114                     }
115                 }
116             }
117         }
118     })
119 }
120
121 // Method to fetch lamp state from Adafruit IO
122 private fun fetchLampState(feedKey: String, toggleButton: ToggleButton, imageView: ImageView, onDrawable: Int, offDrawable: Int) {
123     val url = "https://io.adafruit.com/api/v2/$IO_USERNAME/feeds/$feedKey/data/last"
124

```

```

120 // Method to fetch lamp state from Adafruit IO
121 private fun fetchLampState(feedKey: String, toggleButton: ToggleButton, imageView: ImageView, onDrawable: Int, offDrawable: Int) {
122     val url = "https://io.adafruit.com/api/v2/$IO_USERNAME/feeds/$feedKey/data/last"
123     val request = Request.Builder()
124         .url(url)
125         .get()
126         .addHeader("X-AIO-Key", IO_KEY)
127         .build()
128     client.newCall(request).enqueue(object : Callback {
129         @↑
130         override fun onFailure(call: Call, e: IOException) {
131             runOnUiThread {
132                 Toast.makeText(context: this@menuLight, text: "Failed to fetch lamp state", Toast.LENGTH_SHORT).show()
133                 Log.e(tag: "AdafruitIO", msg: "Fetch lamp state failed: ${e.message}")
134             }
135         }
136         @↑
137         override fun onResponse(call: Call, response: Response) {
138             response.use {
139                 if (response.isSuccessful) {
140                     val responseBody = response.body?.string()
141                     responseBody?.let {
142                         val jsonObject = JSONObject(it)
143                         val value = jsonObject.getString(name: "value")
144                         runOnUiThread {
145                             when (value) {
146                                 "Lamp1 On" -> {
147                                     toggleButton.isChecked = true
148                                     imageView.setImageResource(onDrawable)
149                                 }
150                                 "Lamp1 Off" -> {
151                                     toggleButton.isChecked = false
152                                     imageView.setImageResource(offDrawable)
153                                 }
154                                 "Lamp2 On" -> {
155                                     toggleButton.isChecked = true
156                                     imageView.setImageResource(onDrawable)
157                                 }
158                                 "Lamp2 Off" -> {
159                                     toggleButton.isChecked = false
160                                     imageView.setImageResource(offDrawable)
161                                 }
162                             }
163                         }
164                     }
165                 }
166             }
167         }
168     })
169 }

```

```

144         "Lamp1 On" -> {
145             toggleButton.isChecked = true
146             imageView.setImageResource(onDrawable)
147         }
148         "Lamp1 Off" -> {
149             toggleButton.isChecked = false
150             imageView.setImageResource(offDrawable)
151         }
152         "Lamp2 On" -> {
153             toggleButton.isChecked = true
154             imageView.setImageResource(onDrawable)
155         }
156         "Lamp2 Off" -> {
157             toggleButton.isChecked = false
158             imageView.setImageResource(offDrawable)
159         }
160         else -> {
161             Log.e(tag, "AdafruitIO", msg, "Unexpected value: $value")
162         }
163     }
164 }
165 }
166 } else {
167     val responseBody = response.body?.string()
168     runOnUiThread {
169         Toast.makeText(context, this@menuLight, text, "Failed to fetch data: ${response.code}", Toast.LENGTH_SHORT).show()
170         Log.e(tag, "AdafruitIO", msg, "Fetch failed with status code: ${response.code}, Error: $errorBody")
171     }
172 }
173 }
174 }
175 })
176 }
177 }
178

```

- activity_menumonitoring.xml

```

1  <RelativeLayout
2      xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:id="@+id/main"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="#feffdb"
8      tools:context=".menumonitoring">
9      <View
10         android:id="@+id/top_view"
11         android:layout_width="match_parent"
12         android:layout_height="200dp"
13         android:background="@drawable/bg_viewmenu" />
14      <RelativeLayout
15         android:layout_width="wrap_content"
16         android:layout_height="627dp"
17         android:layout_below="@+id/textView"
18         android:layout_alignParentBottom="true"
19         android:layout_marginLeft="24dp"
20         android:layout_marginTop="-3dp"
21         android:layout_marginRight="24dp"
22         android:layout_marginBottom="13dp"
23         android:background="#ffffff">
24          <!-- Input for Lamp 1 Power (Watts) -->
25          <TextView
26             android:id="@+id/powerLabelLamp1"
27             android:layout_width="wrap_content"
28             android:layout_height="wrap_content"
29             android:layout_marginLeft="24dp"
30             android:layout_marginTop="20dp"
31             android:text="Daya Lampu 1 (Watt):"
32             android:textColor="@color/black"
33             android:textSize="18sp" />
34          <EditText
35             android:id="@+id/powerInputLamp1"
36             android:layout_width="200dp"
37             android:layout_height="48dp"
38             android:layout_below="@+id/powerLabelLamp1"

```

```

38         android:layout_below="@+id/powerLabelLamp1"
39         android:layout_marginLeft="24dp"
40         android:layout_marginTop="5dp"
41         android:hint="Masukkan Daya (W)"
42         android:inputType="numberDecimal" />
43     <!-- Input for Lamp 2 Power (Watts) -->
44     <TextView
45         android:id="@+id/powerLabelLamp2"
46         android:layout_width="wrap_content"
47         android:layout_height="wrap_content"
48         android:layout_below="@+id/powerInputLamp1"
49         android:layout_marginLeft="24dp"
50         android:layout_marginTop="30dp"
51         android:text="Daya Lampu 2 (Watt):"
52         android:textColor="@color/black"
53         android:textSize="18sp" />
54     <EditText
55         android:id="@+id/powerInputLamp2"
56         android:layout_width="200dp"
57         android:layout_height="48dp"
58         android:layout_below="@+id/powerLabelLamp2"
59         android:layout_marginLeft="24dp"
60         android:layout_marginTop="5dp"
61         android:hint="Masukkan Daya (W)"
62         android:inputType="numberDecimal" />
63     <!-- Energy Usage and Price for Lamp 1 -->
64     <TextView
65         android:id="@+id/energyUsageTextViewLamp1"
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"
68         android:layout_below="@+id/powerInputLamp2"
69         android:layout_marginLeft="24dp"
70         android:layout_marginTop="50dp"
71         android:text="Energi Terpakai Lampu 1: Menghitung..."
72         android:textColor="@color/black"

```

```

71         android:text="Energi Terpakai Lampu 1: Menghitung..."
72         android:textColor="@color/black"
73         android:textSize="16sp" />
74     <!-- Energy Usage and Price for Lamp 2 -->
75     <TextView
76         android:id="@+id/energyUsageTextViewLamp2"
77         android:layout_width="wrap_content"
78         android:layout_height="wrap_content"
79         android:layout_below="@+id/energyUsageTextViewLamp1"
80         android:layout_marginLeft="24dp"
81         android:layout_marginTop="10dp"
82         android:text="Energi Terpakai Lampu 2: Menghitung..."
83         android:textColor="@color/black"
84         android:textSize="16sp" />
85     <!-- Time On for Lamp 1 -->
86     <TextView
87         android:id="@+id/timeOnTextViewLamp1"
88         android:layout_width="wrap_content"
89         android:layout_height="wrap_content"
90         android:layout_below="@+id/energyUsageTextViewLamp2"
91         android:layout_marginLeft="24dp"
92         android:layout_marginTop="50dp"
93         android:text="Lampu 1 Menyala Selama: menghitung..."
94         android:textColor="@color/black"
95         android:textSize="16sp" />
96     <!-- Time On for Lamp 2 -->
97     <TextView
98         android:id="@+id/timeOnTextViewLamp2"
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:layout_below="@+id/timeOnTextViewLamp1"
102        android:layout_marginLeft="24dp"
103        android:layout_marginTop="10dp"
104        android:text="Lampu 2 Menyala Selama: Menghitung..."
105        android:textColor="@color/black"

```

```

105     android:textColor="@color/black"
106     android:textSize="16sp" />
107     <!-- Title for Lamp State History -->
108     <Button
109         android:id="@+id/resetButtonLamp1"
110         android:layout_width="match_parent"
111         android:layout_height="60dp"
112         android:layout_below="@id/timeOnTextViewLamp2"
113         android:layout_marginTop="75dp"
114         android:background="@drawable/bg_btnm"
115         android:text="Reset Lampu 1"
116         android:textColor="#FFC608"
117         android:textStyle="bold" />
118     <Button
119         android:id="@+id/resetButtonLamp2"
120         android:layout_width="match_parent"
121         android:layout_height="60dp"
122         android:layout_below="@id/resetButtonLamp1"
123         android:layout_marginTop="25dp"
124         android:background="@drawable/bg_btnm"
125         android:text="Reset Lampu 2"
126         android:textColor="#FFC608"
127         android:textStyle="bold" />
128 </RelativeLayout>
129 <TextView
130     android:id="@+id/textView"
131     android:layout_width="match_parent"
132     android:layout_height="54dp"
133     android:layout_marginTop="160dp"
134     android:gravity="center_horizontal"
135     android:text="Edith for Smarthome"
136     android:textColor="@color/black"
137     android:textSize="20sp"
138     android:textStyle="bold" />
139 <ImageView

```

```

139 <ImageView
140     android:id="@+id/logo"
141     android:layout_width="match_parent"
142     android:layout_height="275dp"
143     android:layout_marginTop="-45dp"
144     android:src="@drawable/edith" />
145 </RelativeLayout>

```

- Menumonitoring.kt

```

1 package com.example.program
2
3 > import ...
19
20 class menumonitoring : AppCompatActivity() {
21
22     companion object {
23         const val IO_USERNAME = "PnD"
24         const val IO_KEY = "aio_LjiC48U6JXWz3q7MJuFoR9Xt2aW0"
25         const val PRICE_PER_KWH = 1444.7 // Price in Rupiah per kWh
26     }
27
28     private lateinit var timeOnTextViewLamp1: TextView
29     private lateinit var timeOnTextViewLamp2: TextView
30     private lateinit var powerInputLamp1: EditText
31     private lateinit var powerInputLamp2: EditText
32     private lateinit var energyUsageTextViewLamp1: TextView
33     private lateinit var energyUsageTextViewLamp2: TextView
34     private lateinit var resetButtonLamp1: Button
35     private lateinit var resetButtonLamp2: Button
36
37     private var lamp1StartTime: Long? = null
38     private var lamp2StartTime: Long? = null
39     private var lamp1TotalOnTime: Long = 0
40     private var lamp2TotalOnTime: Long = 0
41     private var lastLamp1State: Boolean = false
42     private var lastLamp2State: Boolean = false
43     private val client: OkHttpClient = OkHttpClient()
44     private val handler = Handler(Looper.getMainLooper())
45     private val updateInterval = TimeUnit.SECONDS.toMillis(1) // Update every second
46     private lateinit var sharedPreferences: SharedPreferences
47     // Lamp state history (stores last 5 states)
48     private val lampHistory: LinkedList<String> = LinkedList()
49
50     override fun onCreate(savedInstanceState: Bundle?) {
51         super.onCreate(savedInstanceState)
52         setContentView(R.layout.activity_menumonitoring)
53         // Initialize UI components
54         timeOnTextViewLamp1 = findViewById(R.id.timeOnTextViewLamp1)
55         timeOnTextViewLamp2 = findViewById(R.id.timeOnTextViewLamp2)
56         powerInputLamp1 = findViewById(R.id.powerInputLamp1)
57         powerInputLamp2 = findViewById(R.id.powerInputLamp2)
58         energyUsageTextViewLamp1 = findViewById(R.id.energyUsageTextViewLamp1)
59         energyUsageTextViewLamp2 = findViewById(R.id.energyUsageTextViewLamp2)
60         resetButtonLamp1 = findViewById(R.id.resetButtonLamp1)
61         resetButtonLamp2 = findViewById(R.id.resetButtonLamp2)
62     }
63 }

```

```

55     energyUsageTextViewLamp1 = findViewById(R.id.energyUsageTextViewLamp1)
56     energyUsageTextViewLamp2 = findViewById(R.id.energyUsageTextViewLamp2)
57     resetButtonLamp1 = findViewById(R.id.resetButtonLamp1)
58     resetButtonLamp2 = findViewById(R.id.resetButtonLamp2)
59     sharedPreferences = getSharedPreferences("lamp_preferences", Context.MODE_PRIVATE)
60     // Load saved values when the screen is created
61     restoreSavedData()
62     // Update the display with the restored data
63     updateTimeDisplayAndEnergy(lampNumber: 1, lastLamp1State)
64     updateTimeDisplayAndEnergy(lampNumber: 2, lastLamp2State)
65     // Reset buttons for resetting lamp time
66     resetButtonLamp1.setOnClickListener { resetLampTime(lampNumber: 1) }
67     resetButtonLamp2.setOnClickListener { resetLampTime(lampNumber: 2) }
68 }
69 @Override fun onResume() {
70     super.onResume()
71     // Start lamp monitoring when the activity resumes
72     startLampMonitoring()
73 }
74 @Override fun onPause() {
75     super.onPause()
76     // Stop the handler when the activity is paused
77     handler.removeCallbacksAndMessages(null)
78     // Save data
79     saveData()
80 }
81 private fun startLampMonitoring() {
82     handler.post(object : Runnable {
83         override fun run() {
84             // Fetch the current state of the lamp1 and lamp2
85             fetchLampState(feedKey: "lamp1", lampNumber: 1)
86             fetchLampState(feedKey: "lamp2", lampNumber: 2)
87             // Repeat the check every 1 second
88             handler.postDelayed(this, updateInterval)
89         }
90     })
91 }
92 private fun fetchLampState(feedKey: String, lampNumber: Int) {

```

```

91     }
92     private fun fetchLampState(feedKey: String, lampNumber: Int) {
93         val url = "https://io.adafruit.com/api/v2/$IO_USERNAME/feeds/$feedKey/data/last"
94         val request = Request.Builder()
95             .url(url)
96             .get()
97             .addHeader("name: X-AIO-Key", IO_KEY)
98             .build()
99         client.newCall(request).enqueue(object : Callback {
100             override fun onFailure(call: Call, e: IOException) {
101                 runOnUiThread {
102                     Toast.makeText(context, this@menumonitoring, text: "Failed to fetch lamp state", Toast.LENGTH_SHORT).show()
103                 }
104             }
105             override fun onResponse(call: Call, response: Response) {
106                 response.use {
107                     if (response.isSuccessful) {
108                         val responseBody = response.body?.string()
109                         responseBody?.let {
110                             val jsonObject = JSONObject(it)
111                             val value = jsonObject.getString("name: value")
112                             runOnUiThread {
113                                 when (lampNumber) {
114                                     1 -> updateLampState(isOn = value == "1", lampNumber)
115                                     2 -> updateLampState(isOn = value == "1", lampNumber)
116                                 }
117                             }
118                         }
119                     } else {
120                         runOnUiThread {
121                             Toast.makeText(context, this@menumonitoring, text: "Failed to fetch lamp state: ${response.code}", Toast.LENGTH_SHORT).show()
122                         }
123                     }
124                 }
125             }
126         })
127     }
128     private fun updateLampState(isOn: Boolean, lampNumber: Int) {

```

```

127     }
128     private fun updateLampState(isOn: Boolean, lampNumber: Int) {
129         val lastState = when (lampNumber) {
130             1 -> lastLamp1State
131             2 -> lastLamp2State
132             else -> null
133         }
134         // If the state has changed, update the time and status
135         if (lastState != isOn) {
136             val currentTime = getCurrentFormattedTime()
137             when (lampNumber) {
138                 1 -> {
139                     lastLamp1State = isOn
140                     if (isOn) {
141                         // Save the start time when the lamp turns ON
142                         lamp1StartTime = System.currentTimeMillis()
143                         sharedPreferences.edit().putLong("lamp1_start_time", lamp1StartTime!!).apply()
144                         //addToLampHistory("Lamp 1 ON at $currentTime")
145                     } else {
146                         // When the lamp turns OFF, calculate and save the total ON time
147                         lamp1TotalOnTime += System.currentTimeMillis() - (lamp1StartTime ?: 0)
148                         sharedPreferences.edit().putLong("lamp1_total_on_time", lamp1TotalOnTime).apply()
149                         lamp1StartTime = null
150                         //addToLampHistory("Lamp 1 OFF at $currentTime")
151                     }
152                     updateTimeDisplayAndEnergy(lampNumber, isOn)
153                 }
154                 2 -> {
155                     lastLamp2State = isOn
156                     if (isOn) {
157                         // Save the start time when the lamp turns ON
158                         lamp2StartTime = System.currentTimeMillis()
159                         sharedPreferences.edit().putLong("lamp2_start_time", lamp2StartTime!!).apply()
160                         //addToLampHistory("Lamp 2 ON at $currentTime")
161                     } else {
162                         // When the lamp turns OFF, calculate and save the total ON time
163                         lamp2TotalOnTime += System.currentTimeMillis() - (lamp2StartTime ?: 0)
164                         sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()

```

```

20      class menumonitoring : AppCompatActivity() {
128          private fun updateLampState(isOn: Boolean, lampNumber: Int) {
164              sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()
165              lamp2StartTime = null
166              // addToLampHistory("Lamp 2 OFF at $currentTime")
167              }
168              updateTimeDisplayAndEnergy(lampNumber, isOn)
169          }
170      }
171  }
172  // Ensure the energy usage and time are updated regularly when the lamp is on
173  if (isOn) {
174      updateTimeDisplayAndEnergy(lampNumber, isOn)
175  }
176  }
177  private fun updateTimeDisplayAndEnergy(lampNumber: Int, isOn: Boolean) {
178      val startTime = when (lampNumber) {
179          1 -> lamp1StartTime
180          2 -> lamp2StartTime
181          else -> null
182      }
183      // Calculate the total ON time
184      val totalOnTime = when (lampNumber) {
185          1 -> lamp1TotalOnTime + if (isOn && startTime != null) System.currentTimeMillis() - startTime else 0
186          2 -> lamp2TotalOnTime + if (isOn && startTime != null) System.currentTimeMillis() - startTime else 0
187          else -> 0
188      }
189      val seconds = (totalOnTime / 1000) % 60
190      val minutes = (totalOnTime / (1000 * 60)) % 60
191      val hours = (totalOnTime / (1000 * 60 * 60))
192      val timeText = String.format("%02d:%02d:%02d", hours, minutes, seconds)
193      val powerInput = when (lampNumber) {
194          1 -> powerInputLamp1.text.toString().toDoubleOrNull() ?: 0.0
195          2 -> powerInputLamp2.text.toString().toDoubleOrNull() ?: 0.0
196          else -> 0.0
197      }
198      // Calculate energy usage (kWh) and cost in Rupiah
199      val energyUsage = (powerInput / 1000) * (totalOnTime / (1000 * 60 * 60).toDouble())
200      val cost = energyUsage * PRICE_PER_KWH
201      // Display energy usage and cost

```

```

201 // Display energy usage and cost
202 when (lampNumber) {
203     1 -> {
204         timeOnTextViewLamp1.text = "Waktu Terpakai Lampu 1: $timeText"
205         energyUsageTextViewLamp1.text = "Energi Digunakan 1: %.4f kWh, Harga: Rp %.2f".format(energyUsage, cost)
206     }
207     2 -> {
208         timeOnTextViewLamp2.text = "Waktu Terpakai Lampu 2: $timeText"
209         energyUsageTextViewLamp2.text = "Energi Digunakan 2: %.4f kWh, harga: Rp %.2f".format(energyUsage, cost)
210     }
211 }
212 }
213
214 private fun resetLampTime(lampNumber: Int) {
215     when (lampNumber) {
216         1 -> {
217             lamp1TotalOnTime = 0
218             lamp1StartTime = System.currentTimeMillis()
219             sharedPreferences.edit().putLong("lamp1_total_on_time", lamp1TotalOnTime).apply()
220             Toast.makeText(context: this, text: "Lamp 1 time reset", Toast.LENGTH_SHORT).show()
221         }
222         2 -> {
223             lamp2TotalOnTime = 0
224             lamp2StartTime = System.currentTimeMillis()
225             sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()
226             Toast.makeText(context: this, text: "Lamp 2 time reset", Toast.LENGTH_SHORT).show()
227         }
228     }
229 }
230
231 private fun saveData() {
232     val editor = sharedPreferences.edit()
233     // Save the total ON time for lamp1 and lamp2
234     editor.putLong("lamp1_total_on_time", lamp1TotalOnTime)
235     editor.putLong("lamp2_total_on_time", lamp2TotalOnTime)
236     // Save the start time for lamp1 and lamp2 if the lamp is ON
237     lamp1StartTime?.let { editor.putLong("lamp1_start_time", it) }
238     lamp2StartTime?.let { editor.putLong("lamp2_start_time", it) }
239     // Save the lamp states

```

```

20     class menumonitoring : AppCompatActivity() {
230         private fun saveData() {
234             editor.putLong("lamp2_total_on_time", lamp2TotalOnTime)
235             // Save the start time for lamp1 and lamp2 if the lamp is ON
236             lamp1StartTime?.let { editor.putLong("lamp1_start_time", it) }
237             lamp2StartTime?.let { editor.putLong("lamp2_start_time", it) }
238             // Save the lamp states
239             editor.putBoolean("lastLamp1State", lastLamp1State)
240             editor.putBoolean("lastLamp2State", lastLamp2State)
241             // Save the power input for lamp1 and lamp2
242             editor.putString("power_lamp1", powerInputLamp1.text.toString())
243             editor.putString("power_lamp2", powerInputLamp2.text.toString())
244             editor.apply()
245         }
246         private fun restoreSavedData() {
247             // Restore the power input values
248             powerInputLamp1.setText(sharedPreferences.getString("power_lamp1", ""))
249             powerInputLamp2.setText(sharedPreferences.getString("power_lamp2", ""))
250             // Restore the total ON time for lamp1 and lamp2
251             lamp1TotalOnTime = sharedPreferences.getLong("lamp1_total_on_time", 0)
252             lamp2TotalOnTime = sharedPreferences.getLong("lamp2_total_on_time", 0)
253             // Restore the start times
254             lamp1StartTime = sharedPreferences.getLong("lamp1_start_time", 0).takeIf { it > 0 }
255             lamp2StartTime = sharedPreferences.getLong("lamp2_start_time", 0).takeIf { it > 0 }
256             // Restore the lamp states
257             lastLamp1State = sharedPreferences.getBoolean("lastLamp1State", false)
258             lastLamp2State = sharedPreferences.getBoolean("lastLamp2State", false)
259             // Restore the history from SharedPreferences
260         }
261     }
262     @Override fun onDestroy() {
263         super.onDestroy()
264         handler.removeCallbacksAndMessages(null) // Stop the handler when the activity is destroyed
265     }
266     // Helper function to format the current time
267     private fun getCurrentFormattedTime(): String {
268         val dateFormat = SimpleDateFormat(pattern = "dd-MM-yyyy HH:mm:ss", Locale.getDefault())
269         return dateFormat.format(Date())
270     }
271 }

```

- Menu_history.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:id="@+id/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:background="#feffdb"
10     tools:context=".menu.light">
11     <View
12         android:id="@+id/top_view"
13         android:layout_width="match_parent"
14         android:layout_height="200dp"
15         android:background="@drawable/bg_viewmenu"
16     />
17     <ImageView
18         android:layout_width="match_parent"
19         android:layout_height="275dp"
20         android:layout_marginTop="-45dp"
21         android:src="@drawable/edith"
22         android:id="@+id/logo"
23     >
24     </ImageView>
25     <TextView
26         android:layout_width="match_parent"
27         android:layout_height="match_parent"
28         android:text="Edith for Smarthome"
29         android:textSize="20sp"
30         android:textStyle="bold"
31         android:layout_marginTop="160dp"
32         android:gravity="center_horizontal"
33         android:textColor="@color/black">
34     </TextView>
35     <RelativeLayout
36         android:layout_width="match_parent"
37         android:layout_height="match_parent"

```

```

37         android:layout_height="match_parent"
38         android:layout_below="@id/top_view">
39         <ScrollView
40             android:layout_width="match_parent"
41             android:layout_height="match_parent">
42             <RelativeLayout
43                 android:layout_width="match_parent"
44                 android:layout_height="match_parent"
45                 android:layout_below="@id/top_view"
46                 android:layout_marginTop="10dp"
47                 android:layout_marginLeft="24dp"
48                 android:layout_marginRight="24dp"
49                 android:background="#ffffff"
50             >
51                 <TextView
52                     android:id="@+id/historyTitleTextView"
53                     android:layout_width="wrap_content"
54                     android:layout_height="wrap_content"
55                     android:layout_marginLeft="24dp"
56                     android:layout_marginTop="13dp"
57                     android:text="Riwayat Keadaan Lampu :"
58                     android:textColor="@color/black"
59                     android:textSize="18sp"
60                     android:textStyle="bold" />
61                 <TextView
62                     android:id="@+id/historyTextView"
63                     android:layout_width="wrap_content"
64                     android:layout_height="wrap_content"
65                     android:layout_below="@+id/historyTitleTextView"
66                     android:layout_marginLeft="24dp"
67                     android:layout_marginTop="7dp"
68                     android:text="Belum Ada Riwayat"
69                     android:textColor="@color/black"
70                     android:textSize="16sp" />
71             </RelativeLayout>
72         </ScrollView>
73     </RelativeLayout>
74 </RelativeLayout>
75

```

- Historymenu

```

1 package com.example.program
2
3 > import ...
12
13 class menuhistory : AppCompatActivity() {
14     private lateinit var historyTextView: TextView
15     private val client = OkHttpClient()
16     private val feedKeys = listOf("lamp1", "lamp2")
17     private val aioKey = "aio_LjiC48U6JXWz3q7MJuFoR9Xt2aW0"
18     private val username = "Pn0"
19     private val lastStatus = mutableMapOf<String, String>()
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContentView(R.layout.activity_menuhistory)
23         historyTextView = findViewById(R.id.historyTextView)
24         fetchLampHistory()
25     }
26     private fun fetchLampHistory() {
27         val historyData = mutableListOf<String>()
28         // Fetch data from each feed
29         feedKeys.forEach { feedKey ->
30             val url = "https://io.adafruit.com/api/v2/$username/feeds/$feedKey/data?limit=20"
31             val request = Request.Builder()
32                 .url(url)
33                 .addHeader("X-AIO-Key", aioKey)
34                 .build()
35             client.newCall(request).enqueue(object : Callback {
36                 override fun onFailure(call: Call, e: IOException) {
37                     runOnUiThread {
38                         Toast.makeText(
39                             context: this@menuhistory,
40                             text: "Failed to fetch history for $feedKey: ${e.message}",
41                             Toast.LENGTH_SHORT
42                         ).show()
43                     }
44                 }
45                 override fun onResponse(call: Call, response: Response) {
46                     if (response.isSuccessful) {
47                         response.body?.string()?.let { responseData ->
48                             val jsonArray = JSONArray(responseData)

```

```

48     val jsonArray = JSONArray(responseData)
49     for (i in 0 until jsonArray.length()) {
50         val dataObj = jsonArray.getJSONObject(i)
51         val rawValue = dataObj.getString(name: "value")
52         val value = translateValue(rawValue) // Translate 1/0 to Hidup/Mati
53         val timestamp = dataObj.getString(name: "created_at")
54         // Check if the status has changed
55         if (lastStatus[feedKey] != value) {
56             lastStatus[feedKey] = value // Update last known status
57             // Format the timestamp
58             val formattedTimestamp = formatTimestamp(timestamp)
59             // Combine feedKey, value, and formatted timestamp
60             val historyEntry = "$feedKey $value, $formattedTimestamp"
61             historyData.add(historyEntry)
62         }
63     }
64     runOnUiThread {
65         updateHistoryView(historyData)
66     }
67 }
68 } else {
69     runOnUiThread {
70         Toast.makeText(
71             context: this@menuhistory,
72             text: "Error fetching history for $feedKey: ${response.message}",
73             Toast.LENGTH_SHORT
74         ).show()
75     }
76 }
77 }
78 })
79 }
80 }
81 private fun translateValue(rawValue: String): String {
82     return when (rawValue) {
83         "1" -> "Hidup"

```

```

71         context: this@menuhistory,
72         text: "Error fetching history for $feedKey: ${response.message}",
73         Toast.LENGTH_SHORT
74     ).show()
75     }
76 }
77 }
78 })
79 }
80 }
81 private fun translateValue(rawValue: String): String {
82     return when (rawValue) {
83         "1" -> "Hidup"
84         "0" -> "Mati"
85         else -> rawValue // Handle other unexpected values if needed
86     }
87 }
88 private fun formatTimestamp(timestamp: String): String {
89     // Parse the Adafruit timestamp into a Date object
90     val inputFormat = SimpleDateFormat(pattern: "yyyy-MM-dd'T'HH:mm:ss'Z'", Locale.getDefault())
91     inputFormat.timeZone = TimeZone.getTimeZone(id: "UTC")
92     val date = inputFormat.parse(timestamp)
93     // Format the date into "HH:mm, dd/MM/yyyy"
94     val outputFormat = SimpleDateFormat(pattern: "HH:mm, dd/MM/yyyy", Locale.getDefault())
95     return outputFormat.format(date!!)
96 }
97 private fun updateHistoryView(historyData: List<String>) {
98     if (historyData.isNotEmpty()) {
99         val formattedHistory = historyData.joinToString(separator = "\n")
100         historyTextView.text = formattedHistory
101     } else {
102         historyTextView.text = "Belum Ada Riwayat"
103     }
104 }
105 }

```

- Activity_meuschedule.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:id="@+id/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:background="#feffdb"
10     tools:context=".menu.light">
11     <View
12         android:id="@+id/top_view"
13         android:layout_width="match_parent"
14         android:layout_height="200dp"
15         android:background="@drawable/bg_viewmenu"
16     />
17     <ImageView
18         android:layout_width="match_parent"
19         android:layout_height="275dp"
20         android:layout_marginTop="-45dp"
21         android:src="@drawable/edith"
22         android:id="@+id/logo"
23     >
24     </ImageView>
25     <TextView
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
28         android:text="Edith for Smarthome"
29         android:textSize="20sp"
30         android:textStyle="bold"
31         android:layout_marginTop="160dp"
32         android:id="@+id/edith"
33         android:gravity="center_horizontal"
34         android:textColor="@color/black">
35     </TextView>
36     <RelativeLayout
37         android:layout_width="match_parent"
38         android:layout_height="wrap_content"

```

```

38     android:layout_height="wrap_content"
39     android:layout_alignParentBottom="true"
40     android:layout_marginLeft="12dp"
41     android:layout_marginTop="5dp"
42     android:layout_marginRight="12dp"
43     android:layout_below="@id/top_view"
44 >
45 <TextView
46     android:id="@+id/Judul"
47     android:layout_width="wrap_content"
48     android:layout_height="wrap_content"
49     android:layout_marginLeft="45dp"
50     android:layout_marginTop="5dp"
51     android:text="Pendjadwalan Lampu"
52     android:textColor="@color/black"
53     android:gravity="center_horizontal"
54     android:textSize="30sp" />
55 <ToggleButton
56     android:layout_width="match_parent"
57     android:layout_height="wrap_content"
58     android:layout_below="@id/Judul"
59     android:id="@+id/togle"
60     android:background="@drawable/bg_btn"
61     android:textOn="Aktif"
62     android:textOff="Tidak Aktif"
63     android:layout_alignParentTop="true"
64
65     />
66 <RelativeLayout
67     android:id="@+id/Barlampu0"
68     android:layout_width="wrap_content"
69     android:layout_height="wrap_content"
70     android:layout_marginTop="10dp"
71     android:layout_below="@+id/togle"
72     android:background="@drawable/barlampujadwal"
73 >

```

```

73     >
74     <TextView
75         android:id="@+id/lampu0"
76         android:layout_width="wrap_content"
77         android:layout_height="wrap_content"
78         android:layout_marginLeft="24dp"
79         android:layout_marginTop="20dp"
80         android:text="Lampu 1"
81         android:textColor="@color/black"
82         android:textSize="20sp" />
83     <LinearLayout
84         android:layout_width="match_parent"
85         android:layout_height="wrap_content"
86         android:id="@+id/lampu0hidup"
87         android:layout_below="@+id/lampu0">
88         <TextView
89             android:layout_width="wrap_content"
90             android:layout_height="wrap_content"
91             android:text="Hidup"
92             android:textStyle="bold"
93             android:textSize="20sp"
94             android:textColor="@color/black"/>
95         <Spinner
96             android:id="@+id/jam0hidup"
97             android:layout_width="100dp"
98             android:layout_height="50dp"
99             />
100        <TextView
101            android:layout_width="wrap_content"
102            android:layout_height="wrap_content"
103            android:text="."
104            android:textStyle="bold"
105            android:textSize="20sp"
106            android:textColor="@color/black"/>
107        <Spinner

```

```

127     }
128     private fun updateLampState(isOn: Boolean, lampNumber: Int) {
129         val lastState = when (lampNumber) {
130             1 -> lastLamp1State
131             2 -> lastLamp2State
132             else -> null
133         }
134         // If the state has changed, update the time and status
135         if (lastState != isOn) {
136             val currentTime = getCurrentFormattedTime()
137             when (lampNumber) {
138                 1 -> {
139                     lastLamp1State = isOn
140                     if (isOn) {
141                         // Save the start time when the lamp turns ON
142                         lamp1StartTime = System.currentTimeMillis()
143                         sharedPreferences.edit().putLong("lamp1_start_time", lamp1StartTime!!).apply()
144                         //addToLampHistory("Lamp 1 ON at $currentTime")
145                     } else {
146                         // When the lamp turns OFF, calculate and save the total ON time
147                         lamp1TotalOnTime += System.currentTimeMillis() - (lamp1StartTime ?: 0)
148                         sharedPreferences.edit().putLong("lamp1_total_on_time", lamp1TotalOnTime).apply()
149                         lamp1StartTime = null
150                         //addToLampHistory("Lamp 1 OFF at $currentTime")
151                     }
152                     updateTimeDisplayAndEnergy(lampNumber, isOn)
153                 }
154                 2 -> {
155                     lastLamp2State = isOn
156                     if (isOn) {
157                         // Save the start time when the lamp turns ON
158                         lamp2StartTime = System.currentTimeMillis()
159                         sharedPreferences.edit().putLong("lamp2_start_time", lamp2StartTime!!).apply()
160                         //addToLampHistory("Lamp 2 ON at $currentTime")
161                     } else {
162                         // When the lamp turns OFF, calculate and save the total ON time
163                         lamp2TotalOnTime += System.currentTimeMillis() - (lamp2StartTime ?: 0)
164                         sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()

```

```

20      class menuMonitoring : AppCompatActivity() {
128          private fun updateLampState(isOn: Boolean, lampNumber: Int) {
164              sharedPreferences.edit().putLong("lamp2_total_on_time", lamp2TotalOnTime).apply()
165              lamp2StartTime = null
166              // addToLampHistory("Lamp 2 OFF at $currentTime")
167          }
168          updateTimeDisplayAndEnergy(lampNumber, isOn)
169      }
170  }
171  }
172  // Ensure the energy usage and time are updated regularly when the lamp is on
173  if (isOn) {
174      updateTimeDisplayAndEnergy(lampNumber, isOn)
175  }
176  }
177  private fun updateTimeDisplayAndEnergy(lampNumber: Int, isOn: Boolean) {
178      val startTime = when (lampNumber) {
179          1 -> lamp1StartTime
180          2 -> lamp2StartTime
181          else -> null
182      }
183      // Calculate the total ON time
184      val totalOnTime = when (lampNumber) {
185          1 -> lamp1TotalOnTime + if (isOn && startTime != null) System.currentTimeMillis() - startTime else 0
186          2 -> lamp2TotalOnTime + if (isOn && startTime != null) System.currentTimeMillis() - startTime else 0
187          else -> 0
188      }
189      val seconds = (totalOnTime / 1000) % 60
190      val minutes = (totalOnTime / (1000 * 60)) % 60
191      val hours = (totalOnTime / (1000 * 60 * 60))
192      val timeText = String.format("%02d:%02d:%02d", hours, minutes, seconds)
193      val powerInput = when (lampNumber) {
194          1 -> powerInputLamp1.text.toString().toDoubleOrNull() ?: 0.0
195          2 -> powerInputLamp2.text.toString().toDoubleOrNull() ?: 0.0
196          else -> 0.0
197      }
198      // Calculate energy usage (kWh) and cost in Rupiah
199      val energyUsage = (powerInput / 1000) * (totalOnTime / (1000 * 60 * 60).toDouble())
200      val cost = energyUsage * PRICE_PER_KWH
201      // Display energy usage and cost

```

```

140         android:layout_height="50dp"
141     />
142 </LinearLayout>
143 <Button
144     android:layout_width="match_parent"
145     android:layout_height="wrap_content"
146     android:id="@+id/kirimlampu0"
147     android:layout_below="@id/lampu0mati"
148     android:text="Kirim"
149     android:background="@color/Gold"/>
150 <Button
151     android:layout_width="match_parent"
152     android:layout_height="wrap_content"
153     android:layout_below="@id/kirimlampu0"
154     android:id="@+id/resetlampu0"
155     android:text="Reset"
156     android:background="@color/Gold"/>
157 </RelativeLayout>
158 <RelativeLayout
159     android:id="@+id/Barlampu1"
160     android:layout_width="wrap_content"
161     android:layout_height="wrap_content"
162     android:layout_below="@id/Barlampu0"
163     android:layout_marginTop="20dp"
164     android:background="@drawable/barlampujadwal"
165 >
166     <TextView
167         android:id="@+id/lampu1"
168         android:layout_width="wrap_content"
169         android:layout_height="wrap_content"
170         android:layout_marginLeft="24dp"
171         android:layout_marginTop="30dp"
172         android:text="Lampu 2"
173         android:textColor="@color/black"

```

```

173     android:textColor="@color/black"
174     android:textSize="20sp" />
175     <LinearLayout
176         android:layout_width="match_parent"
177         android:layout_height="wrap_content"
178         android:id="@+id/lampu1hidup"
179         android:layout_below="@+id/lampu1">
180         <TextView
181             android:layout_width="wrap_content"
182             android:layout_height="wrap_content"
183             android:text=" Hidup "
184             android:textStyle="bold"
185             android:textSize="20sp"
186             android:textColor="@color/black" />
187         <Spinner
188             android:id="@+id/jam1hidup"
189             android:layout_width="100dp"
190             android:layout_height="50dp"
191             />
192         <TextView
193             android:layout_width="wrap_content"
194             android:layout_height="wrap_content"
195             android:text="."
196             android:textStyle="bold"
197             android:textSize="20sp"
198             android:textColor="@color/black" />
199         <Spinner
200             android:id="@+id/menit1hidup"
201             android:layout_width="120dp"
202             android:layout_height="50dp"
203             />
204     </LinearLayout>
205     <LinearLayout
206         android:layout_width="match_parent"

```

```

206         android:layout_width="match_parent"
207         android:layout_height="wrap_content"
208         android:id="@+id/lampu1mati"
209         android:layout_below="@+id/lampu1hidup">
210         <TextView
211             android:layout_width="wrap_content"
212             android:layout_height="wrap_content"
213             android:text="Mati"
214             android:textStyle="bold"
215             android:textSize="20sp"
216             android:textColor="@color/black"/>
217         <Spinner
218             android:id="@+id/jam1mati"
219             android:layout_width="100dp"
220             android:layout_height="50dp"
221             />
222         <TextView
223             android:layout_width="wrap_content"
224             android:layout_height="wrap_content"
225             android:text="."
226             android:textStyle="bold"
227             android:textSize="20sp"
228             android:textColor="@color/black"/>
229         <Spinner
230             android:id="@+id/menit1mati"
231             android:layout_width="120dp"
232             android:layout_height="50dp"
233             />
234     </LinearLayout>
235     <Button
236         android:layout_width="match_parent"
237         android:layout_height="wrap_content"
238         android:id="@+id/kirimlampu1"
239         android:layout_below="@+id/lampu1mati"
240         android:text="Kirim"
241         android:background="@color/Gold"/>
242     <Button
243         android:layout_width="match_parent"
244         android:layout_height="wrap_content"
245         android:layout_below="@+id/kirimlampu1"
246         android:id="@+id/resetlampu1"
247         android:text="Reset"
248         android:background="@color/Gold"/>
249 </RelativeLayout>
250 </RelativeLayout>
251 </RelativeLayout>

```

- menuschedule.kt

```

1 package com.example.program
2
3 > import ...
10
11 class menuschedule : AppCompatActivity() {
12     private val ioUsername = "PnD"
13     private val ioKey = "aio_LjiC48U6JXWz3q7MJuFoR9Xt2aW0"
14     private val baseUrl = "https://io.adafruit.com/api/v2/$ioUsername/feeds"
15     private lateinit var toggleButton: ToggleButton
16
17     override fun onCreate(savedInstanceState: Bundle?) {
18         super.onCreate(savedInstanceState)
19         setContentView(R.layout.activity_menuschedule)
20
21         // Inisialisasi ToggleButton dan komponen lainnya
22         toggleButton = findViewById(R.id.toggle)
23         val jam0Hidup: Spinner = findViewById(R.id.jam0hidup)
24         val menit0Hidup: Spinner = findViewById(R.id.menit0hidup)
25         val jam0Mati: Spinner = findViewById(R.id.jam0mati)
26         val menit0Mati: Spinner = findViewById(R.id.menit0mati)
27         val kirimLampu0: Button = findViewById(R.id.kirimlampu0)
28         val resetLampu0: Button = findViewById(R.id.resetlampu0)
29         val jam1Hidup: Spinner = findViewById(R.id.jam1hidup)
30         val menit1Hidup: Spinner = findViewById(R.id.menit1hidup)
31         val jam1Mati: Spinner = findViewById(R.id.jam1mati)
32         val menit1Mati: Spinner = findViewById(R.id.menit1mati)
33         val kirimLampu1: Button = findViewById(R.id.kirimlampu1)
34         val resetLampu1: Button = findViewById(R.id.resetlampu1)
35
36         setupSpinners(jam0Hidup, menit0Hidup, jam0Mati, menit0Mati)
37         setupSpinners(jam1Hidup, menit1Hidup, jam1Mati, menit1Mati)
38         loadFeedData(feedKey: "lamp1-schedule", jam0Hidup, menit0Hidup, jam0Mati, menit0Mati)
39         loadFeedData(feedKey: "lamp2-schedule", jam1Hidup, menit1Hidup, jam1Mati, menit1Mati)
40
41         // Set listener untuk ToggleButton
42         toggleButton.setOnCheckedChangeListener { _, isChecked ->
43             enableDisableFeatures(isChecked, kirimLampu0, resetLampu0, kirimLampu1, resetLampu1)
44             val toggleState = if (isChecked) "On" else "Off"
45             sendToFeed(feedKey: "schedule", toggleState) // Kirim status On/Off ke Adafruit IO
46         }

```

```

47
48     kirimLampu0.setOnClickListener {
49         val data = "${jam0Hidup.selectedItem}:${menit0Hidup.selectedItem}-${jam0Mati.selectedItem}:${menit0Mati.selectedItem}"
50         sendToFeed( feedKey: "lamp1-schedule", data)
51     }
52
53     resetLampu0.setOnClickListener {
54         clearFeedData( feedKey: "lamp1-schedule")
55         resetSpinners(jam0Hidup, menit0Hidup, jam0Mati, menit0Mati)
56     }
57
58     kirimLampu1.setOnClickListener {
59         val data = "${jam1Hidup.selectedItem}:${menit1Hidup.selectedItem}-${jam1Mati.selectedItem}:${menit1Mati.selectedItem}"
60         sendToFeed( feedKey: "lamp2-schedule", data)
61     }
62
63     resetLampu1.setOnClickListener {
64         clearFeedData( feedKey: "lamp2-schedule")
65         resetSpinners(jam1Hidup, menit1Hidup, jam1Mati, menit1Mati)
66     }
67 }
68
69 private fun enableDisableFeatures(isEnabled: Boolean, vararg buttons: Button) {
70     buttons.forEach { it.isEnabled = isEnabled }
71 }
72
73 private fun setupSpinners(vararg spinners: Spinner) {
74     val hours = (0 ≤ .. ≤ 23).map { String.format("%02d", it) }
75     val minutes = (0 ≤ .. ≤ 59).map { String.format("%02d", it) }
76     val hourAdapter = ArrayAdapter( context: this, android.R.layout.simple_spinner_item, hours)
77     hourAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
78     val minuteAdapter = ArrayAdapter( context: this, android.R.layout.simple_spinner_item, minutes)
79     minuteAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
80     spinners.forEachIndexed { index, spinner ->
81         spinner.adapter = if (index % 2 == 0) hourAdapter else minuteAdapter
82     }
83 }
84

```

```

83     }
84
85     private fun loadFeedData(feedKey: String, jamHidup: Spinner, menitHidup: Spinner, jamMati: Spinner, menitMati: Spinner) {
86         val url = "$baseUrl/$feedKey/data/last"
87         val client = OkHttpClient()
88         val request = Request.Builder()
89             .url(url)
90             .addHeader("X-AIO-Key", ioKey)
91             .build()
92         client.newCall(request).enqueue(object : Callback {
93             override fun onFailure(call: Call, e: IOException) {
94                 runOnUiThread {
95                     Toast.makeText(context: this@menuschedule, text: "Failed to load data", Toast.LENGTH_SHORT).show()
96                 }
97             }
98             override fun onResponse(call: Call, response: Response) {
99                 response.body?.string()?.let {
100                     val data = JSONObject(it).getString("value")
101                     val parts = data.split(...delimiters: "-", ":")
102                     runOnUiThread {
103                         jamHidup.setSelection(parts[0].toInt())
104                         menitHidup.setSelection(parts[1].toInt())
105                         jamMati.setSelection(parts[2].toInt())
106                         menitMati.setSelection(parts[3].toInt())
107                     }
108                 }
109             }
110         })
111     }
112
113     private fun sendToFeed(feedKey: String, data: String) {
114         val url = "$baseUrl/$feedKey/data"
115         val client = OkHttpClient()
116         val json = JSONObject().put("value", data).toString()
117         val body = RequestBody.create("application/json".toMediaTypeOrNull(), json)
118         val request = Request.Builder()
119             .url(url)
120             .addHeader("X-AIO-Key", ioKey)
121             .post(body)

```

```

120         .addHeader( name: "X-AIO-Key", ioKey)
121         .post(body)
122         .build()
123     client.newCall(request).enqueue(object : Callback {
124         override fun onFailure(call: Call, e: IOException) {
125             runOnUiThread {
126                 Toast.makeText( context: this@menuschedule, text: "Failed to send data", Toast.LENGTH_SHORT).show()
127             }
128         }
129         override fun onResponse(call: Call, response: Response) {
130             runOnUiThread {
131                 Toast.makeText( context: this@menuschedule, text: "Data sent successfully", Toast.LENGTH_SHORT).show()
132             }
133         }
134     })
135 }
136
137 private fun clearFeedData(feedKey: String) {
138     val url = "$baseUrl/$feedKey/data"
139     val client = OkHttpClient()
140     val request = Request.Builder()
141         .url(url)
142         .addHeader( name: "X-AIO-Key", ioKey)
143         .delete()
144         .build()
145     client.newCall(request).enqueue(object : Callback {
146         override fun onFailure(call: Call, e: IOException) {
147             runOnUiThread {
148                 Toast.makeText( context: this@menuschedule, text: "Failed to reset data", Toast.LENGTH_SHORT).show()
149             }
150         }
151         override fun onResponse(call: Call, response: Response) {
152             runOnUiThread {
153                 Toast.makeText( context: this@menuschedule, text: "Data reset successfully", Toast.LENGTH_SHORT).show()
154             }
155         }
156     })
157 }

```

```

158
159     private fun resetSpinners(vararg spinners: Spinner) {
160         spinners.forEach { it.setSelection(0) }
161     }
162 }
163

```

- activity_menuhelp.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:id="@+id/main"
7      android:layout_width="match_parent"
8      android:layout_height="match_parent"
9      android:background="#feffdb"
10     tools:context=".menuhelp">
11     <View
12         android:id="@+id/top_view"
13         android:layout_width="match_parent"
14         android:layout_height="200dp"
15         android:background="@drawable/bg_viewmenu"
16     />
17     <ImageView
18         android:layout_width="match_parent"
19         android:layout_height="275dp"
20         android:layout_marginTop="-45dp"
21         android:src="@drawable/edith"
22         android:id="@+id/logo"
23     />
24     <TextView
25         android:layout_width="match_parent"
26         android:layout_height="wrap_content"
27         android:text="Edith for Smarthome"
28         android:textSize="20sp"
29         android:textStyle="bold"
30         android:layout_marginTop="160dp"
31         android:gravity="center_horizontal"
32         android:textColor="@color/black" />
33     <RelativeLayout
34         android:layout_width="match_parent"
35         android:layout_height="match_parent"
36         android:layout_below="@id/top_view">
37         <ScrollView
38             android:layout_width="match_parent"

```

```

27         android:text="Edith for Smarthome"
28         android:textSize="20sp"
29         android:textStyle="bold"
30         android:layout_marginTop="160dp"
31         android:gravity="center_horizontal"
32         android:textColor="@color/black" />
33     <RelativeLayout
34         android:layout_width="match_parent"
35         android:layout_height="match_parent"
36         android:layout_below="@id/top_view">
37         <ScrollView
38             android:layout_width="match_parent"
39             android:layout_height="match_parent">
40             <LinearLayout
41                 android:layout_width="match_parent"
42                 android:layout_height="match_parent"
43                 android:layout_below="@+id/top_view"
44                 android:layout_marginTop="10dp"
45                 android:layout_marginLeft="24dp"
46                 android:layout_marginRight="24dp"
47                 android:background="#ffffff"
48             >
49             <!-- Add the helpTextView for the help content -->
50             <TextView
51                 android:id="@+id/helpTextView"
52                 android:layout_width="match_parent"
53                 android:layout_height="wrap_content"
54                 android:layout_below="@+id/top_view"
55                 android:layout_marginTop="20dp"
56                 android:layout_marginLeft="24dp"
57                 android:textSize="16sp"
58                 android:textColor="@android:color/black" />
59             </LinearLayout>
60         </ScrollView>
61     </RelativeLayout>
62 </RelativeLayout>

```

- Menuhelp.kt

```

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_menuhelp)
    val helpText = findViewById<TextView>(R.id.helpTextView)
    helpText.text = """
Selamat datang di Aplikasi Smarthome! Berikut adalah panduan untuk membantu Anda menggunakan aplikasi:
1. Lamp Control :
    - Anda dapat mengontrol dua lampu dari layar Lamp Control.
    - Setiap lampu memiliki sakelar sakelar ON/OFF.
    - Ketuk sakelar untuk menghidupkan atau mematikan lampu.
    - Gambar lampu akan diperbarui untuk mencerminkan kondisinya saat ini.
    - Status setiap lampu disinkronkan dengan platform Adafruit IO.
2. Pemantauan Lampu:
    - Anda dapat melihat penggunaan daya dan riwayat status setiap lampu.
    - Masukkan konsumsi daya (dalam watt) untuk setiap lampu.
    - Aplikasi menghitung penggunaan energi dan menampilkan biaya dalam Rupiah.
    - Anda dapat mengatur ulang waktu ON untuk setiap lampu menggunakan tombol reset.
3. Riwayat Lampu
    - Anda dapat melihat kapan lampu dinyalakan dan dimatikan
    - Riwayat akan diperbarui setiap 5 sampai 10 detik sekali
    - Terdapat jam dan menit yang tertera dan juga tanggal
    - Dapat discroll guna memberikan informasi yang lebih luas
4. Penjadwalan Lampu
    - Memberikan opsi pemilihan jam dan menit penjadwalan
    - Pilih jam dan menit penjadwalan daripada hidup dan matinya lampu
    - Tombol kirim berfungsi mengirimkan waktu ke adafruit
    - Tombol reset berfungsi menghapus atau mngatur ulang penjadwalan, tapi belum menghapus jadwal yang telah
    dikirimkan sebelumnya.
5. Catatan Umum:
    - Aplikasi mengambil status sebenarnya dari lampu dari platform Adafruit IO.
    - Pastikan Anda memiliki koneksi internet yang stabil untuk menyinkronkan data dengan benar.
    - Untuk memantau penggunaan energi, pastikan input daya untuk setiap lampu akurat.
Jika Anda memerlukan bantuan lebih lanjut, jangan ragu untuk menghubungi dukungan.
    """
    helpText.text = helpText.text.trimIndent()
}

```

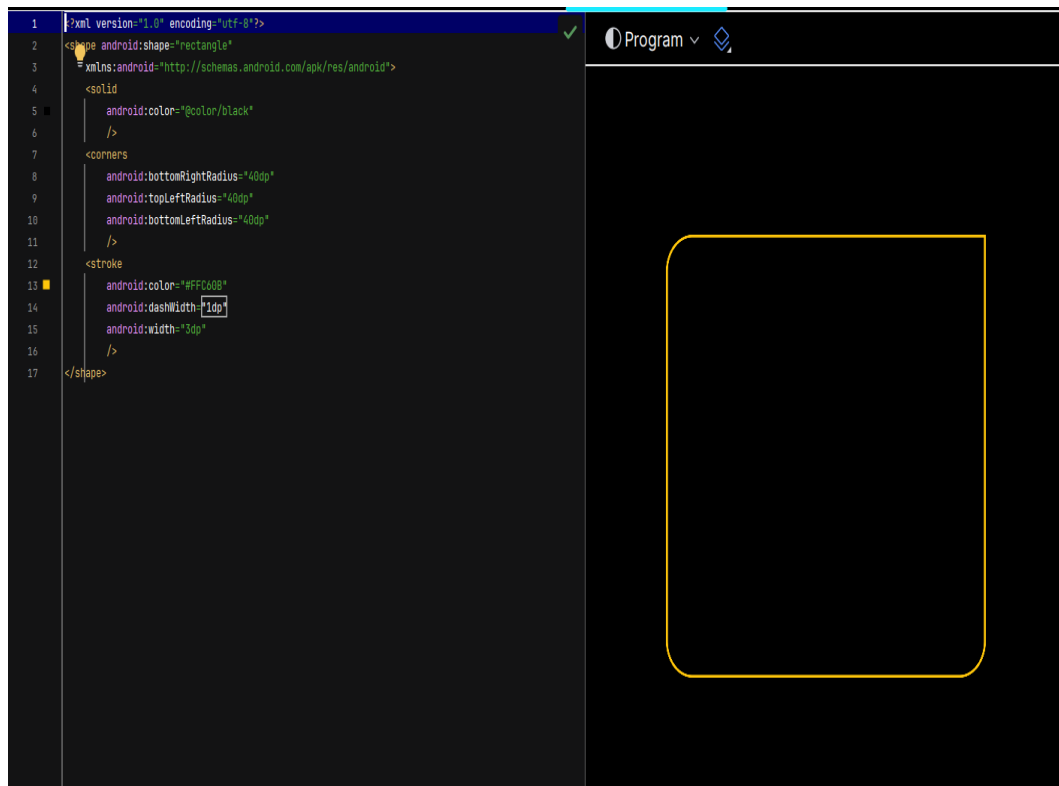
- Manifest

```

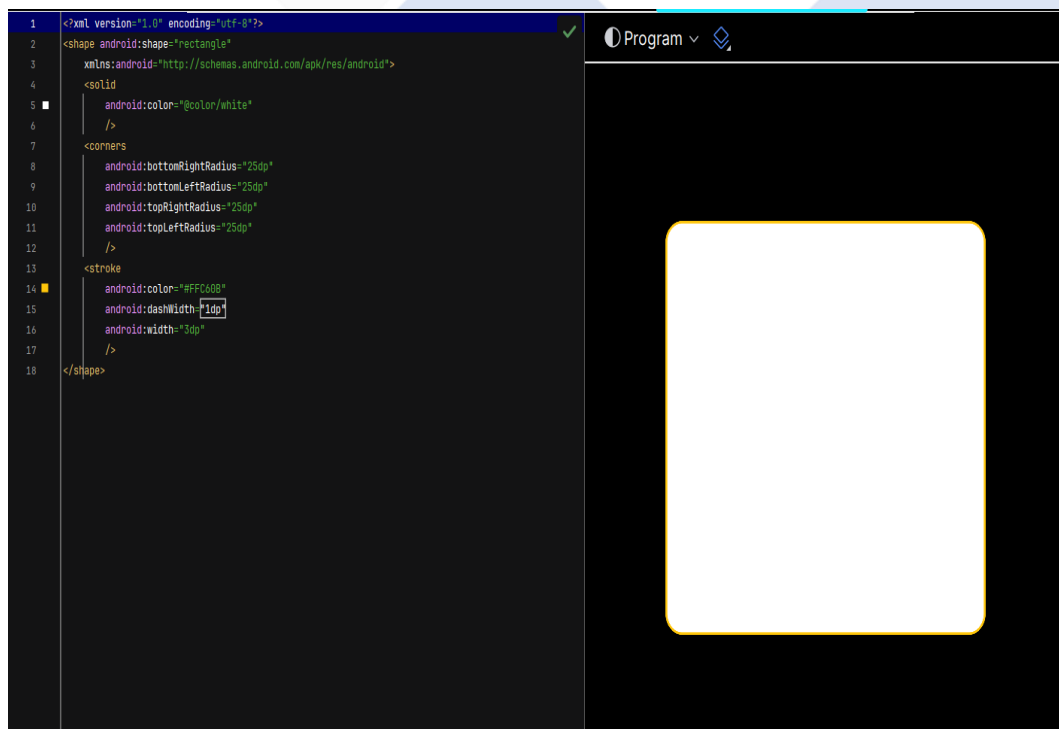
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools">
4      <!-- Required permissions for internet access -->
5      <uses-permission android:name="android.permission.INTERNET" />
6      <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
7      <application
8          android:allowBackup="true"
9          android:dataExtractionRules="@xml/data_extraction_rules"
10         android:fullBackupContent="@xml/backup_rules"
11         android:icon="@mipmap/ic_launcher"
12         android:label="EDITH"
13         android:roundIcon="@mipmap/ic_launcher_round"
14         android:supportRtl="true"
15         android:theme="@style/Theme.Program"
16         tools:targetApi="31">
17         <activity
18             android:name=".menuhistory"
19             android:exported="false" />
20         <activity
21             android:name=".menuschedule"
22             android:exported="false" />
23         <activity
24             android:name=".menuhelp"
25             android:exported="false" />
26         <activity
27             android:name=".menumonitoring"
28             android:exported="false" />
29         <activity
30             android:name=".menulight"
31             android:exported="false" />
32         <activity
33             android:name=".MainActivity"
34             android:exported="true">
35             <intent-filter>
36                 <action android:name="android.intent.action.MAIN" />
37                 <category android:name="android.intent.category.LAUNCHER" />
38             </intent-filter>
39         </activity>
40     </application>
41 </manifest>

```

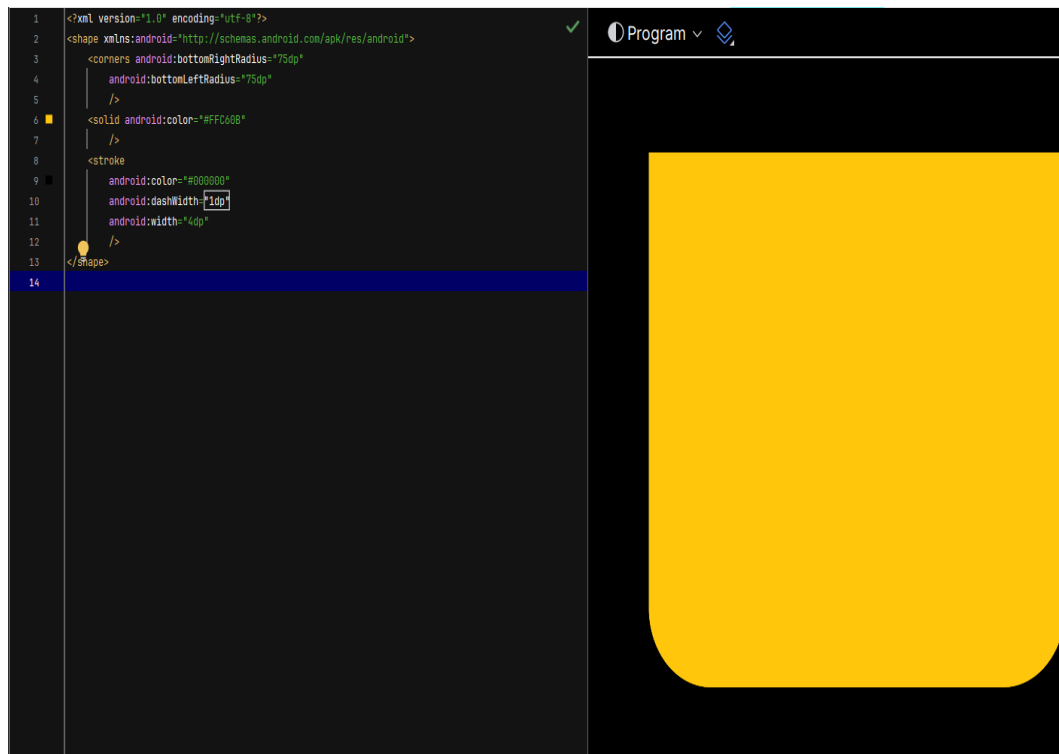
- Btnmonitor



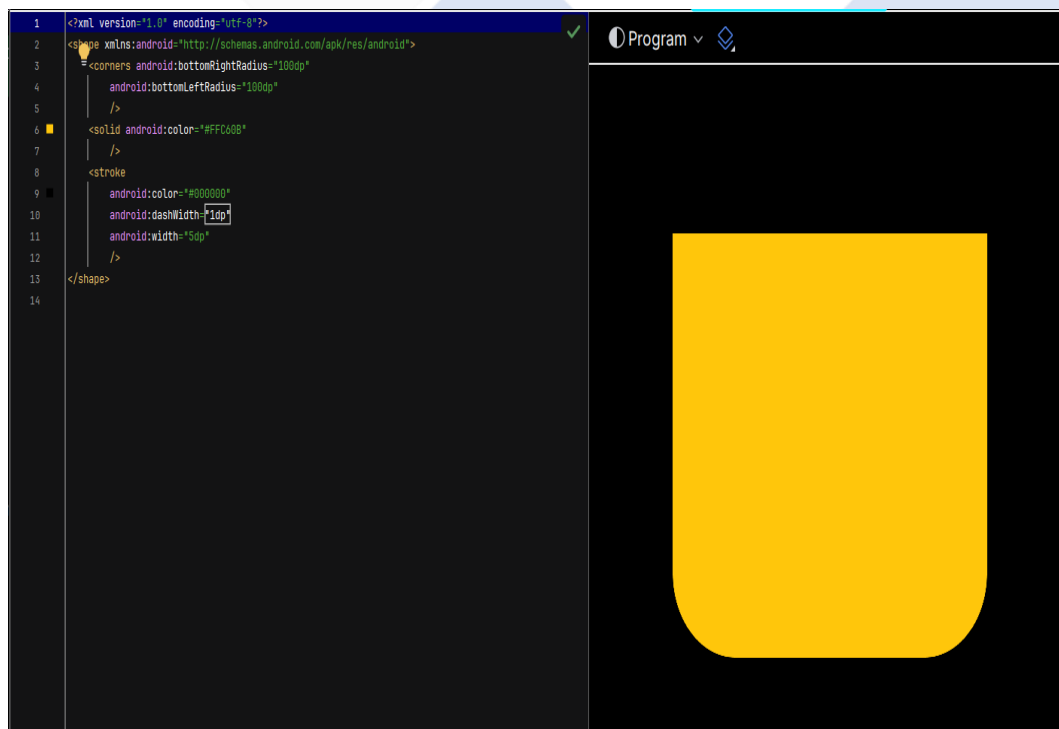
- Barlampujadwal



- view



- viewmenu



2. Lampiran progrsm Arduino IDE

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <time.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
#include <ArduinoJson.h> // Library untuk parsing JSON

// Ganti dengan informasi jaringan WiFi Anda
#define WIFI_SSID "POCO F6"
#define WIFI_PASS "12345678"

// Ganti dengan username dan key dari akun Adafruit IO Anda
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883
#define AIO_USERNAME "PnD"
#define AIO_KEY "aio_LjiC48U6JXWz3q7MJuFoR9Xt2aW0"

// Pin relay
#define RELAY1_PIN 16
#define RELAY2_PIN 17

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

// Feed untuk relay1 dan relay2
Adafruit_MQTT_Subscribe relay1Feed = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/relay1");
Adafruit_MQTT_Subscribe relay2Feed = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/relay2");

// Feed untuk status lamp1 dan lamp2
Adafruit_MQTT_Publish lamp1Status = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/lamp1");
Adafruit_MQTT_Publish lamp2Status = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/lamp2");

unsigned long lastCheck = 0;
const unsigned long interval = 1000; // 1 detik interval

void initializeNTP() {
  configTime(7 * 3600, 0, "pool.ntp.org", "time.nist.gov"); // WIB (UTC+7)
  Serial.println("Sinkronisasi waktu dengan NTP...");
  while (time(nullptr) < 8 * 3600 * 2) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("\nWaktu berhasil disinkronisasi!");
}

void setup() {
  Serial.begin(115200);
  pinMode(RELAY1_PIN, OUTPUT);
  pinMode(RELAY2_PIN, OUTPUT);
  digitalWrite(RELAY1_PIN, LOW);
}
```

```

50     digitalWrite(RELAY1_PIN, LOW);
51     digitalWrite(RELAY2_PIN, LOW);
52
53     Serial.print("Menghubungkan ke WiFi...");
54     WiFi.begin(WIFI_SSID, WIFI_PASS);
55     while (WiFi.status() != WL_CONNECTED) {
56         delay(500);
57         Serial.print(".");
58     }
59     Serial.println("Terhubung!");
60
61     initializeNTP();
62     mqtt.subscribe(&relay1Feed);
63     mqtt.subscribe(&relay2Feed);
64 }
65
66 void loop() {
67     if (!mqtt.connected()) {
68         reconnectMQTT();
69     }
70     mqtt.processPackets(100);
71     mqtt.ping();
72
73     if (millis() - lastCheck >= interval) {
74         lastCheck = millis();
75         handleLampMode("schedule", "lamp1-schedule", RELAY1_PIN, lamp1Status, relay1Feed);
76         handleLampMode("schedule", "lamp2-schedule", RELAY2_PIN, lamp2Status, relay2Feed);
77     }
78
79     Adafruit_MQTT_Subscribe *subscription;
80     while ((subscription = mqtt.readSubscription(5000))) {
81         if (subscription == &relay1Feed) {
82             handleRelayCommand((char *)relay1Feed.lastread, RELAY1_PIN, lamp1Status);
83         }
84         if (subscription == &relay2Feed) {
85             handleRelayCommand((char *)relay2Feed.lastread, RELAY2_PIN, lamp2Status);
86         }
87     }
88 }
89
90 void reconnectMQTT() {
91     while (!mqtt.connected()) {
92         Serial.println("Menyambungkan ke Adafruit IO...");
93         if (mqtt.connect() == 0) {
94             Serial.println("Terhubung ke Adafruit IO!");
95             mqtt.subscribe(&relay1Feed);
96             mqtt.subscribe(&relay2Feed);
97         } else {
98             Serial.println("Gagal menyambung, mencoba lagi...");
99             delay(5000);
100         }

```

```

101     }
102 }
103
104 void handleLampMode(String modeFeed, String scheduleFeed, int relayPin, Adafruit_MQTT_Publish &statusFeed, Adafruit_MQTT_Subscribe &relayFeed) {
105     String mode = fetchFeedMode(modeFeed);
106     if (mode == "On") {
107         Serial.println("Mode otomatis aktif.");
108         processSchedule(scheduleFeed, relayPin, statusFeed);
109     } else {
110         Serial.println("Mode manual aktif.");
111         String relayCommand = fetchLastRelayCommand(relayFeed);
112         if (relayCommand != "") {
113             handleRelayCommand(relayCommand.c_str(), relayPin, statusFeed);
114         }
115     }
116 }
117
118 String fetchFeedMode(String feedName) {
119     if (WiFi.status() == WL_CONNECTED) {
120         HTTPClient http;
121         String url = "https://io.adafruit.com/api/v2/" + String(AIO_USERNAME) + "/feeds/" + feedName + "/data/last";
122         http.begin(url);
123         http.addHeader("X-AIO-Key", AIO_KEY);
124         int httpStatusCode = http.GET();
125         if (httpStatusCode > 0) {
126             String payload = http.getString();
127             if (httpStatusCode > 0) {
128                 String payload = http.getString();
129                 return (payload.indexOf("On") >= 0) ? "On" : "Off";
130             }
131             http.end();
132         }
133         return "Off";
134     }
135 }
136
137 void processSchedule(String scheduleFeed, int relayPin, Adafruit_MQTT_Publish &statusFeed) {
138     String schedule = fetchSchedule(scheduleFeed);
139     if (schedule != "") {
140         int startTime, endTime;
141         if (parseSchedule(schedule, startTime, endTime)) {
142             time_t now = time(nullptr);
143             struct tm timeinfo;
144             localtime_r(&now, &timeinfo);
145
146             int currentHour = timeinfo.tm_hour;
147             int currentMinute = timeinfo.tm_min;
148             int currentTime = currentHour * 60 + currentMinute;
149
150             if (currentTime >= startTime && currentTime < endTime) {
151                 digitalWrite(relayPin, HIGH);
152                 statusFeed.publish("1");
153             } else {

```

```

150         } else {
151             digitalWrite(relayPin, LOW);
152             statusFeed.publish("0");
153         }
154     }
155 }
156 }
157
158 String fetchSchedule(String feedName) {
159     if (WiFi.status() == WL_CONNECTED) {
160         HTTPClient http;
161         String url = "https://io.adafruit.com/api/v2/" + String(AIO_USERNAME) + "/feeds/" + feedName + "/data/last";
162         http.begin(url);
163         http.addHeader("X-AIO-Key", AIO_KEY);
164         int httpResponseCode = http.GET();
165         if (httpResponseCode > 0) {
166             String payload = http.getString();
167             Serial.println("Response: " + payload);
168             DynamicJsonDocument doc(1024);
169             deserializeJson(doc, payload);
170             return doc["value"].as<String>(); // Extracting value
171         }
172         http.end();
173     }
174     return "";
175 }

```

```

176
177 bool parseSchedule(String schedule, int &startTime, int &endTime) {
178     int separatorIndex = schedule.indexOf('-');
179     if (separatorIndex > 0) {
180         String startStr = schedule.substring(0, separatorIndex);
181         String endStr = schedule.substring(separatorIndex + 1);
182
183         startTime = parseTimeToMinutes(startStr);
184         endTime = parseTimeToMinutes(endStr);
185         return true;
186     }
187     return false;
188 }
189
190 int parseTimeToMinutes(String timeStr) {
191     int separatorIndex = timeStr.indexOf(':');
192     if (separatorIndex > 0) {
193         int hours = timeStr.substring(0, separatorIndex).toInt();
194         int minutes = timeStr.substring(separatorIndex + 1).toInt();
195         return hours * 60 + minutes;
196     }
197     return -1;
198 }
199
200 void handleRelayCommand(const char *command, int relayPin, Adafruit_MQTT_Publish &statusFeed) {

```

```
200 void handleRelayCommand(const char *command, int relayPin, Adafruit_MQTT_Publish &statusFeed) {
201     if (String(command).indexOf("On") >= 0) {
202         digitalWrite(relayPin, HIGH);
203         statusFeed.publish("1");
204     } else if (String(command).indexOf("Off") >= 0) {
205         digitalWrite(relayPin, LOW);
206         statusFeed.publish("0");
207     }
208 }
209
210 String fetchLastRelayCommand(Adafruit_MQTT_Subscribe &relayFeed) {
211     if (relayFeed.lastread != nullptr) {
212         return String((char *)relayFeed.lastread);
213     }
214     return "";
215 }
216
```