

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi telah membawa dampak signifikan dalam berbagai aspek kehidupan manusia. Dalam beberapa dekade terakhir, teknologi telah mengalami lonjakan pesat yang membuka pintu bagi inovasi dan peningkatan efisiensi di berbagai sektor. Terobosan seperti komputer, *internet*, telepon seluler, dan kecerdasan buatan telah mengubah cara kita berkomunikasi, bekerja, dan hidup sehari-hari.

Salah satu bidang di mana kemajuan teknologi telah memberikan dampak positif adalah penghematan energi. Dengan adanya perangkat pintar dan sistem otomatisasi yang terhubung, kita dapat mengontrol penggunaan energi dengan lebih efisien. Misalnya, teknologi yang mengintegrasikan sensor cahaya dan pengaturan otomatis dapat mengoptimalkan penggunaan pencahayaan dalam ruangan berdasarkan tingkat cahaya alami yang tersedia. Selain itu, penggunaan perangkat hemat energi, seperti lampu LED yang efisien, dan sistem manajemen energi yang cerdas, memungkinkan kita untuk mengurangi konsumsi energi dan mengurangi dampak negatif terhadap lingkungan.

Internet of Things (IoT) adalah konsep yang menghubungkan perangkat dan objek di sekitar kita melalui internet. Ini memungkinkan pertukaran data dan komunikasi antara perangkat yang terhubung, yang memberikan kemampuan untuk memonitor dan mengontrol sistem secara jarak jauh. Dalam penggunaan IoT, kita

dapat menghubungkan berbagai perangkat seperti lampu, thermostat, kamera pengawas, dan perangkat lainnya, sehingga menciptakan rumah pintar yang terhubung. Dengan mengintegrasikan perangkat dan aplikasi yang terhubung dengan IoT, kita dapat meningkatkan kenyamanan, efisiensi, dan keamanan rumah atau lingkungan kita. Contohnya adalah pengendalian pencahayaan jarak jauh.

Penggunaan IoT juga meluas ke berbagai sektor lainnya, seperti industri, pertanian, kesehatan, dan transportasi. Dengan menghubungkan perangkat dan sistem, kita dapat mengumpulkan dan menganalisis data secara real-time, memperbaiki efisiensi operasional. Secara keseluruhan, kemajuan teknologi memberikan kontribusi yang signifikan dalam penghematan energi dan penggunaan IoT. Dengan penerapan yang cerdas dan bijak, kita dapat mencapai kehidupan yang lebih efisien, terhubung, dan berkelanjutan. Berdasarkan uraian latar belakang tersebut, maka penulis melakukan penelitian yang berjudul “Aplikai Pengatur Tingkat Pencahayaan Lampu pada Ruang Berbasis Android”.

B. Rumusan Masalah

Berdasarkan dari uraian latar belakang diatas, maka rumusan masalah yang akan dibahas adalah bagaimana merancang sebuah Aplikasi berbasis android yang dapat mengatur tingkat pencahayaan lampu?

C. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan masalah tersebut adalah sebagai berikut:

1. Alat yang dibangun ini bersifat prototype.
2. Prototype ini menggunakan lampu khusus yang bersifat *dimnable*.
3. Prototype ini menggunakan dua lampu.
4. Sistem ini adalah aplikasi berbasis android.

D. Tujuan Penelitian

Sesuai dengan permasalahan dan batasan masalah yang telah diuraikan diatas, maka tujuan penelitian ini adalah untuk membuat alat yang dapat mengatur tingkat pencahayaan lampu yang terintegrasi dengan aplikasi android.

E. Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk dapat mengetahui pemanfaatan teknologi mikrokontroller dalam penghematan energi listrik dengan menggunakan aplikasi berbasis android.

F. Sistematika Penulisan

Agar pembahasan lebih sistematis, maka tulisan ini dibuat dalam lima bab, yaitu:

BAB I PENDAHULUAN

Berisi tentang uraian latar belakang pemilihan judul aplikasi pengatur tingkat pencahayaan lampu pada ruangan berbasis android, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam bab ini berisikan tentang teori-teori yang berkaitan dengan mikrokontroller, Arduino, *dimmbable* led bulb, Bahasa pemrograman C++, Bahasa pemrograman kotlin, library, Arduino IDE.

BAB III METODE PENELITIAN

Dalam bab ini menjelaskan mengenai tahapan-tahapan yang dilalui dalam penyelesaian penelitian ini, yaitu tempat penelitian, waktu penelitian, metode pengumpulan data, alat, dan bahan penelitian, tahapan penelitian, metode pengujian serta gambaran desain sistem yang akan dirancang dan dibuat.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Dalam bab ini menjelaskan mengenai rancangan sistem yang telah dibuat, metode dan teknik pengujian.

BAB V KESIMPULAN DAN SARAN

Dalam bab ini dipaparkan mengenai kesimpulan dan saran dari hasil.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Penelitian Terdahulu

Penelitian ini dilakukan tidak lepas dari hasil penelitian terdahulu yang dilakukan sebagai perbandingan dan kajian. Pada penelitian Nano Sudin, Ilham Djufri dan M Kasyif G Umar, (Sudin et al., 2020). dengan judul rancang bangun sistem pengontrol lampu rumah berbasis mikrokontroler arduino uno menggunakan smartphone. Dalam penelitian ini, para penulis merancang dan membangun sebuah sistem yang mengintegrasikan teknologi mikrokontroler Arduino Uno dengan kemampuan komunikasi nirkabel menggunakan smartphone. Sistem ini memungkinkan pengguna untuk mengontrol lampu rumah secara nirkabel melalui aplikasi yang diinstal pada smartphone.

Penerapan mikrokontroler Arduino Uno sebagai otak dari sistem ini memungkinkan pengontrolan lampu rumah dengan mudah dan efisien. Pengguna dapat mengatur keadaan lampu, termasuk menghidupkan, mematikan, melalui antarmuka yang disediakan oleh aplikasi smartphone. Kekurangan dari sistem ini adalah kemampuannya untuk berkomunikasi dengan jarak yang dibatasi oleh kemampuan dari modul *bluetooth* terhadap smarthphone mengontrolnya.

Agus Mulyanto, Yebi Agus Nurhuda, dan Imam Khoirusid (Mulyanto et al., 2017). melakukan penelitian tentang pengembangan dan implementasi sistem pengendalian pencahayaan rumah menggunakan smartphone. Sistem ini

memungkinkan pengguna untuk mengontrol pencahayaan rumah mereka melalui smartphone yang mereka miliki. Sistem yang dikembangkan ini menggabungkan antarmuka pengguna yang intuitif pada aplikasi smartphone dengan perangkat keras mikrokontroler yang terhubung dengan sistem lampu rumah. Pengguna dapat dengan mudah mengatur keadaan lampu, seperti menghidupkan dan mematikan serta membuat jadwal waktu untuk mengontrol lampu secara otomatis. Kekurangan dari sistem ini adalah masih menggunakan *Bluetooth* dalam pengontrolan nirkabelnya.

Prototype sistem pengendalian pencahayaan lampu dengan menggunakan smartphone para penulis bertujuan untuk merancang dan mengimplementasikan sebuah prototipe yang memungkinkan pengguna mengontrol pencahayaan lampu menggunakan smartphone. Sistem ini dirancang dengan memanfaatkan komunikasi nirkabel antara smartphone dan perangkat keras mikrokontroler yang terhubung dengan lampu. Pengguna dapat mengatur keadaan lampu, seperti menghidupkan dan mematikan, mengatur tingkat kecerahan, serta membuat skenario pencahayaan yang disesuaikan dengan kebutuhan melalui antarmuka aplikasi di smartphone. Prototipe ini juga dilengkapi dengan fitur pengaturan jadwal waktu untuk mengontrol otomatis lampu sesuai dengan waktu yang ditentukan. Kelemahan dari sistem ini adalah menggunakan modul *Bluetooth* sebagai media komunikasi nirkabel untuk terhubung kepada aplikasi pada smartphone dimana *module bluetooth* tersebut memiliki keterbatasan jarak koneksi.

B. *Dimmable Bulb*

Lampu LED yang dapat diredupkan berarti Anda dapat mengatur tingkat kecerahan atau lumen sesuai kebutuhan Anda. Misalnya, Anda dapat menggunakan cahaya yang lebih terang (lumen lebih tinggi) saat Anda sedang bekerja, dan meredupkan ke lumen yang lebih rendah saat menonton film atau membaca. Lampu LED yang tidak dapat diredupkan dapat memberi Anda pengalaman pencahayaan yang lebih konsisten. (Ledvance, 2022). Lampu hemat energi yang dapat diatur kecerahannya, yang dikenal sebagai "*dimnable bulb*" atau lampu yang dapat diatur intensitasnya, telah menjadi solusi populer dalam pengaturan pencahayaan modern. Lampu-lampu ini dirancang dengan kemampuan untuk mengubah tingkat kecerahan cahaya yang dihasilkan, memungkinkan pengguna untuk menyesuaikan suasana ruangan sesuai dengan preferensi dan kebutuhan mereka. Dengan menggunakan teknologi yang inovatif, *dimnable bulb* dapat menghemat energi dengan mengurangi tingkat kecerahan ketika cahaya yang lebih terang tidak diperlukan. Ini adalah solusi yang efisien dalam mengurangi konsumsi energi dan biaya listrik secara keseluruhan.

Salah satu keunggulan *dimnable bulb* adalah kemampuannya untuk menciptakan suasana yang nyaman dan fleksibel di dalam ruangan. Dengan memungkinkan pengguna untuk mengatur kecerahan lampu sesuai dengan kegiatan atau suasana yang diinginkan, *dimnable bulb* dapat memberikan pengalaman pencahayaan yang lebih personal dan menyesuaikan suasana ruangan dengan mudah. Misalnya, pengguna dapat meningkatkan kecerahan saat membaca atau bekerja, lalu mengurangi tingkat kecerahan untuk menciptakan suasana yang lebih

hangat dan relaks saat bersantai atau menghabiskan waktu bersama keluarga. Dalam hal ini, *dimmbable* bulb memberikan fleksibilitas dalam menciptakan suasana yang diinginkan dalam ruangan yang berbeda.



Gambar 2. 1. *Dimmable light bulb*

Sumber: <https://m.indiamart.com/proddetail/10w-dc-dimming-emergency-led-bulb-2850594255397.html>

Dimmable bulb telah mengalami perkembangan pesat dengan penerapan teknologi LED (Light Emitting Diode). Teknologi LED memberikan keunggulan dalam hal efisiensi energi, umur panjang, dan kecerahan yang dapat diatur dengan presisi. Lampu LED dimmable memanfaatkan teknologi pemutusan dan peningkatan arus listrik untuk mengatur kecerahan lampu. Dengan menggunakan *dimmer switch* atau sistem kontrol yang sesuai, pengguna dapat dengan mudah mengatur kecerahan lampu sesuai keinginan mereka. Selain itu, lampu LED *dimmbable* juga memiliki kemampuan untuk mempertahankan kecerahan yang konsisten saat diatur, sehingga memberikan kualitas pencahayaan yang stabil dan dapat diandalkan.

C. Lampu Pijar

Lampu pijar adalah lampu yang menghasilkan cahaya dengan cara memanaskan logam filament sampai ke suhu tinggi sehingga menghasilkan sinar. Lampu pijar bekerja melalui prinsip pemijaran dimana saat lampu pijar dinyalakan, maka akan terdapat arus yang mengalir menuju filament (Husnayain et al., 2023). Ketika arus listrik mengalir melalui filamen, panas yang dihasilkan membuatnya bersinar dan memancarkan cahaya yang terlihat. Lampu pijar ditemukan oleh Thomas Edison pada akhir abad ke-19 dan telah menjadi salah satu bentuk pencahayaan yang paling umum digunakan di seluruh dunia.

Lampu pijar memiliki desain yang sederhana namun efektif. Sebuah bola kaca transparan atau translusen melindungi filamen dan mencegah oksidasi yang dapat merusak kawat. Di dalam bola, udara dihisap keluar dan digantikan oleh gas inert, seperti argon, untuk memperpanjang umur filamen. Bagian bawah bola biasanya memiliki soket logam yang dapat disekrupkan ke dalam dudukan lampu, memungkinkan lampu pijar untuk dengan mudah dipasang dan diganti. Meskipun teknologi pencahayaan telah berkembang pesat, lampu pijar tetap populer karena menghasilkan cahaya yang hangat dan alami.

Namun, lampu pijar tidak efisien dalam penggunaan energi dibandingkan dengan teknologi pencahayaan modern seperti lampu fluoresen atau lampu LED. Hanya sekitar 10% dari energi yang digunakan oleh lampu pijar diubah menjadi cahaya, sementara sisanya terbuang sebagai panas. Karena alasan ini, banyak negara telah mulai mengurangi penggunaan lampu pijar dan beralih ke alternatif yang lebih hemat energi. Meskipun demikian, lampu pijar masih digunakan dalam

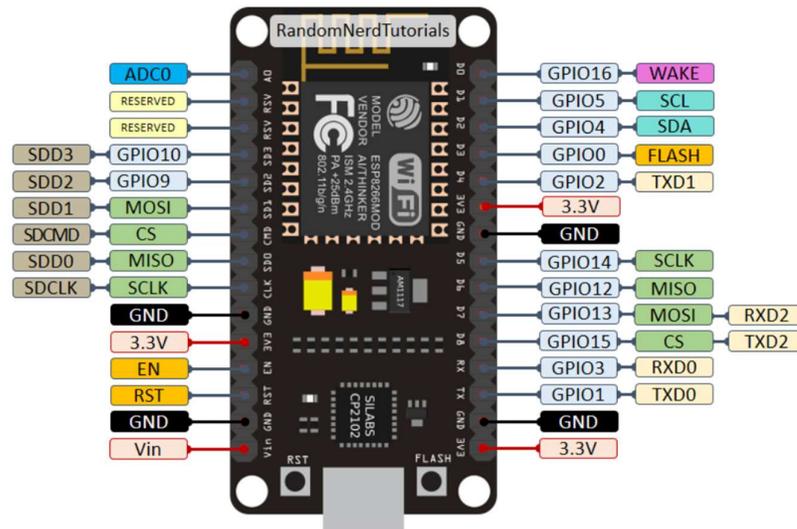
beberapa aplikasi khusus di mana kualitas cahaya yang dihasilkan diinginkan, seperti dalam fotografi atau seni teater.



Gambar 2. 2. Lampu pijar

D. NodeMCU ESP8266

NodeMCU adalah sebuah board elektronik yang berbasis chip ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi internet (*WiFi*) (Satria, 2022). NodeMCU ESP8266 adalah modul WiFi yang populer dan terjangkau yang digunakan dalam berbagai proyek *Internet of Things* (IoT). Dengan ukuran kecil dan konsumsi daya yang rendah, ESP8266 mengintegrasikan mikrokontroler dan modul WiFi dalam satu chip, memberikan kemampuan konektivitas nirkabel kepada perangkat elektronik. Modul ini mendukung protokol komunikasi seperti TCP/IP, UDP, dan MQTT, memungkinkan pengguna untuk menghubungkan perangkat mereka ke jaringan WiFi dan berkomunikasi dengan server atau perangkat lainnya melalui internet. Kebanyakan para pengembang memilih menggunakan ESP8266 dalam mengembangkan solusi IoT yang terhubung secara nirkabel.



Gambar 2. 3. NodeMCU Esp8266

(Sumber: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>)

Salah satu kelebihan NodeMCU ESP8266 adalah fleksibilitasnya dalam pemrograman. Modul ini dapat diprogram menggunakan berbagai bahasa pemrograman seperti Arduino IDE, Python, Lua, dan NodeMCU. Dengan dukungan komunitas yang luas, terdapat banyak sumber daya yang tersedia secara online untuk mempelajari dan memperluas penggunaan NodeMCU ESP8266. Selain itu, modul ini juga kompatibel dengan berbagai sensor dan perangkat elektronik lainnya, sehingga memungkinkan pengembang untuk mengintegrasikan dan mengendalikan berbagai komponen dalam proyek IoT.

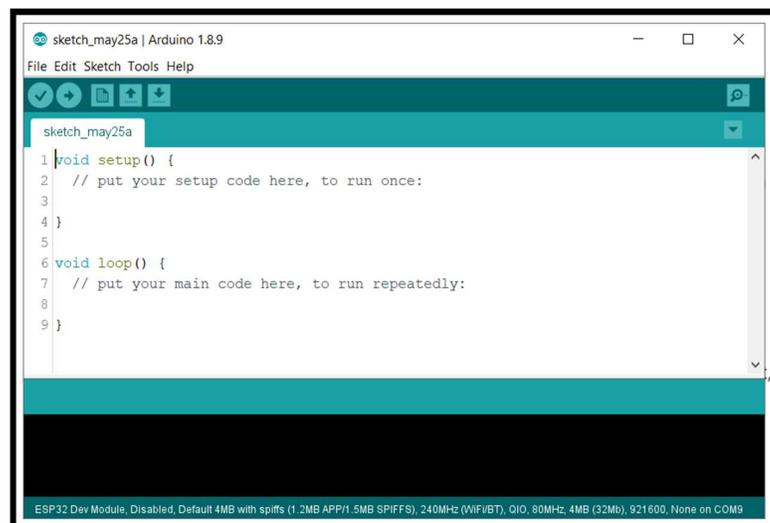
E. Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah sebuah program atau perangkat lunak yang digunakan untuk membuat program Arduino. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara sederhana merupakan lingkungan terintegrasi yang digunakan untuk

melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsifungsi yang dibenamkan melalui sintaks pemrograman. (Septian & Aji, 2018).

Arduino IDE menyediakan berbagai macam library atau perpustakaan untuk memudahkan pengguna dalam mengembangkan program untuk Arduino. Library-library tersebut mencakup fungsi-fungsi dasar seperti pembacaan input dari sensor, pengendalian output, dan komunikasi serial. Dalam mengembangkan program dengan Arduino IDE, pengguna juga dapat memanfaatkan kode sumber dari library yang tersedia atau membuat library kustom sesuai dengan kebutuhan.

Arduino IDE juga menyediakan fitur-fitur seperti syntax highlighting dan autocompletion, yang memudahkan pengguna dalam menulis kode program. Selain itu, Arduino IDE juga menyediakan fitur *debugging* sederhana yang memungkinkan pengguna untuk melacak kesalahan atau bug pada kode program.



Gambar 2. 4. Arduino IDE

Sumber: (<https://docs.thinger.io/sdk-setup/arduino-ide>)

Arduino IDE telah banyak digunakan oleh komunitas pengembang Arduino di seluruh dunia. Komunitas pengembang tersebut juga menyediakan berbagai macam tutorial dan dokumentasi untuk memudahkan pengguna dalam mempelajari dan mengembangkan program dengan Arduino IDE. Dalam menggunakan Arduino IDE, pengguna dapat memanfaatkan sumber daya dan dukungan dari komunitas pengembang untuk mengembangkan program yang lebih kompleks dan inovatif.

F. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middlewaredan aplikasi. Dalam pengembangan aplikasi Android menyediakan AndroidSDK yang menyediakan toolsdan API untuk para pengembang aplikasi dengan platform Android (Wiharto & Budihartanti, 2017). Android merupakan sistem berbasis Linux yang dirancang khusus untuk digunakan pada ponsel pintar, tablet, dan jam pintar. Sistem operasi Android, yang dikembangkan oleh Google, kini menjadi salah satu platform yang paling banyak digunakan secara global. Android menawarkan berbagai fitur dan kemampuan yang mencakup antarmuka pengguna yang intuitif, kemampuan multitasking, akses ke jutaan aplikasi melalui Google Play Store, dan integrasi yang erat dengan layanan Google.

Salah satu daya tarik utama dari Android adalah fleksibilitasnya. Sistem operasi ini memiliki sifat open-source, yang berarti kode sumbernya dapat diakses oleh pengembang dan dimodifikasi sesuai kebutuhan. Hal ini memungkinkan adanya berbagai varian Android yang dikembangkan oleh produsen perangkat seluler, sehingga pengguna dapat memilih perangkat dengan tampilan dan fitur

yang sesuai dengan preferensi mereka. Fleksibilitas ini juga memungkinkan pengembangan aplikasi yang kaya dan inovatif untuk memenuhi kebutuhan pengguna.

Dengan popularitas dan adopsi yang luas, Android telah menjadi ekosistem yang dinamis dan berkembang pesat. Pengguna Android dapat menikmati berbagai fitur canggih, mengakses ribuan aplikasi yang beragam, dan menjelajahi dunia digital dengan mudah. Keberadaan Android telah mengubah cara kita berkomunikasi, bekerja, dan bersosialisasi, serta memberikan pengalaman yang terus berkembang dalam dunia perangkat seluler. Dalam beberapa tahun terakhir, Android juga telah memperluas dukungan untuk teknologi baru seperti kecerdasan buatan, augmented reality (AR), dan virtual reality (VR). Dengan integrasi yang lebih dalam dengan teknologi ini, Android terus menghadirkan pengalaman yang semakin imersif dan interaktif bagi pengguna.



Gambar 2. 5. Logo android

Sumber: https://www.android.com/intl/id_id/

Secara keseluruhan, perkembangan Android telah memainkan peran yang signifikan dalam mengubah cara kita menggunakan perangkat seluler. Dengan inovasi terus menerus, kolaborasi yang luas, dan kemampuan yang berkembang,

Android terus memperkaya dan memperluas ekosistemnya, memberikan pengalaman yang lebih baik dan membuka peluang baru dalam dunia teknologi.

G. Android Studio IDE

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android. Berbasis editor kode dan alat developer yang andal dari *IntelliJ IDEA* (Developer Android, 2024). Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk membuat aplikasi berbasis Android yang tersedia secara gratis atau sebagai open source. Pada konferensi Google I/O 2013 pada 16 Mei 2013, Google mengumumkan peluncuran Android Studio. Sejak saat itu, Android Studio menjadi IDE resmi untuk pengembangan aplikasi Android menggantikan Eclipse.

Berbeda dengan Eclipse, Android Studio didasarkan pada IntelliJ IDEA dan dilengkapi dengan plugin ADT (Android Development Tools). Android Studio menawarkan lebih banyak fitur yang mampu meningkatkan produktivitas Anda saat mem-build aplikasi Android, seperti:

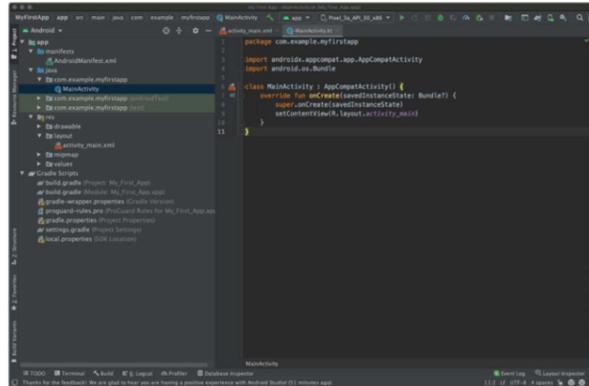
1. Sistem build berbasis *Gradle* yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat *Android*
4. *Edit Live* untuk mengupdate composable di emulator dan perangkat fisik secara real time

5. Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
6. *Framework* dan alat pengujian yang lengkap
7. Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*



Gambar 2. 6. Tampilan awal adroid studio

Sumber:<https://developer.android.com/training/basics/firstapp/images/studio-editor.png?hl=id>



Gambar 2. 7. Halaman kerja android studio

Sumber:<https://developer.android.com/training/basics/firstapp/images/studio-editor.png?hl=id>

H. Kotlin

Bahasa Kotlin adalah bahasa pemrograman yang dikembangkan oleh JetBrains, perusahaan yang juga mengembangkan IDE Android Studio. Bahasa Kotlin adalah pengembangan dari bahasa Java yang sudah populer sebelumnya. Bahasa Kotlin memiliki fitur-fitur bahasa modern yang lebih dibandingkan bahasa Java (Sibarani et al., 2018). Kotlin adalah bahasa pemrograman lintas platform yang sepenuhnya kompatibel dengan Java. Versi dari *Java Virtual Machine (JVM)* pada pustaka standar Kotlin didasarkan pada Pustaka Kelas Java, namun penggunaan inferensi tipe membuat sintaksnya lebih mudah. Pada 7 Mei 2019, Google mengumumkan bahwa Kotlin sekarang menjadi bahasa yang lebih disukai bagi para pengembang aplikasi Android. Sejak dirilisnya Android Studio 3.0 pada Oktober 2017, Kotlin telah menjadi compiler Java alternatif.



Gambar 2. 8. Logo kotlin

Sumber: <https://www.mailslurp.com/blog/end-to-end-test-apis-kotlin-retrofit-junit/>

Kompiler Android Kotlin menghasilkan bytecode Java 8 secara default (yang berjalan di JVM yang lebih baru), tetapi memungkinkan pemrogram memilih untuk menargetkan Java 9 hingga 18, untuk pengoptimalan, atau mengizinkan lebih banyak fitur; memiliki dukungan interoperabilitas kelas rekaman dua arah untuk JVM, diperkenalkan di Java 16, dianggap stabil sejak Kotlin 1.5. Kotlin memiliki dukungan untuk web; dengan mengompilasi ke JavaScript (yaitu, Kotlin/JS dengan back-end klasik, dinyatakan stabil sejak versi 1.3), sedangkan Kotlin/JS (berbasis IR) yang lebih baru dalam versi beta pada versi 1.5.30. Kotlin/*Native* dianggap beta sejak versi 1.3.

Tabel 2. 1. Penulisan sintak pada kotlin

```
fun main() {  
    val scope = "World"  
    println("Hello, $scope!")  
}  
  
fun main(args: Array<String>) {  
    for (arg in args)  
        println(arg)  
}
```

Kotlin melonggarkan batasan Java yang mengizinkan metode dan variable statis hanya ada di dalam badan kelas. Objek dan fungsi statis dapat didefinisikan di tingkat atas paket tanpa memerlukan tingkat kelas yang berlebihan. Untuk

kompatibilitas dengan Java, Kotlin menyediakan anotasi `JvmName` yang menentukan nama kelas yang digunakan saat paket dilihat dari proyek Java. Seperti pada *C*, *C++*, *C#*, *Java*, dan *Go*, titik masuk ke program Kotlin adalah fungsi bernama "main", yang dapat dilewatkan array yang berisi argumen baris perintah apa pun. Ini opsional karena Kotlin 1.3, Perl, PHP dan interpolasi *string* gaya *shell Unix* didukung. Jenis inferensi juga didukung.

Kotlin memungkinkan penambahan fungsi ekstensi ke kelas mana pun tanpa formalitas membuat kelas turunan dengan fungsi baru. Fungsi ekstensi memiliki akses ke semua antarmuka publik kelas, yang dapat digunakan untuk membuat antarmuka fungsi baru ke kelas target. Fungsi ekstensi akan muncul persis seperti fungsi kelas dan akan ditampilkan dalam pemeriksaan penyelesaian kode fungsi kelas.

Kotlin juga mendukung pemrograman berorientasi objek (OOP) dan pemrograman fungsional secara bersamaan. Hal ini memungkinkan pengembang untuk menggunakan paradigma yang paling sesuai dengan kebutuhan proyek mereka. Kotlin mendukung konsep-konsep seperti kelas abstrak, antarmuka, warisan, dan polimorfisme, yang umum dalam pemrograman berorientasi objek. Selain itu, Kotlin memungkinkan penggunaan fungsi tingkat tinggi, lambda, dan fungsi tanpa nama yang sangat berguna dalam pemrograman fungsional. Kemampuan ini menjadikan Kotlin fleksibel dalam berbagai skenario pengembangan, baik untuk aplikasi Android, server, maupun aplikasi berbasis web.

Tabel 2. 2. Sintaks *OOP* pada kotlin

```

open class Animal(val name: String, var age: Int) {

    // Fungsi metode
    open fun sound() {
        println("$name makes a sound")
    }
}

// Kelas Dog mewarisi dari kelas Animal
class Dog(name: String, age: Int, val breed: String) : Animal(name, age) {

    // Override fungsi metode sound
    override fun sound() {
        println("$name, a $breed, barks")
    }
}

fun main() {
    val dog = Dog("Buddy", 3, "Golden Retriever")
    dog.sound() // Output: Buddy, a Golden Retriever, barks
}

```

Salah satu keunggulan utama Kotlin adalah kemampuannya untuk mencegah *NullPointerException*, salah satu masalah yang paling umum ditemui dalam pemrograman Java. Kotlin memperkenalkan sistem penanganan *null* yang lebih aman dengan tipe nullable dan non-nullable. Dengan demikian, variabel tidak dapat memiliki nilai null kecuali secara eksplisit dideklarasikan sebagai nullable. Ini memaksa pengembang untuk menangani nilai null dengan cara yang lebih terkontrol dan aman, mengurangi risiko terjadinya kesalahan yang disebabkan oleh nilai null.

Tabel 2. 3. Sintaks penggunaan *NullPointerException*

```

fun main() {
    // Variabel nullable yang bisa bernilai null
    var name: String? = null
    println(name?.length) // Output: null
}

```

```

// Menggunakan elvis operator '?' untuk memberikan nilai default
val length = name?.length ?: 0
println("Panjang nama: $length") // Output: Panjang nama: 0

// Jika ingin secara eksplisit mengizinkan NullPointerException, gunakan '!!'
try {
    println(name!!.length) // Ini akan melempar NullPointerException
} catch (e: NullPointerException) {
    println("Terjadi NullPointerException: ${e.message}")
}
}

```

Kotlin juga menawarkan fitur *coroutines*, yang memungkinkan penanganan tugas-tugas asynchronous secara lebih efisien. Dengan *coroutines*, pengembang dapat menulis kode yang lebih bersih dan mudah dibaca saat mengelola tugas-tugas seperti pemrosesan jaringan atau I/O yang berjalan di latar belakang. Ini sangat berguna dalam pengembangan aplikasi Android, di mana penanganan tugas yang memerlukan waktu lama tanpa mengganggu antarmuka pengguna sangat penting. *Coroutines* memungkinkan pengembang untuk mengelola proses-proses ini secara lebih sederhana dan terstruktur dibandingkan dengan penggunaan thread manual atau callback.

Tabel 2. 4. Sintaks penggunaan *coroutines*

```

import kotlinx.coroutines.*

fun main() = runBlocking {
    // Memulai coroutine yang berjalan secara asynchronous
    launch {
        delay(1000L) // Simulasi tugas yang memakan waktu
        println("Tugas dalam coroutine selesai")
    }

    // Coroutine lainnya berjalan secara paralel
    launch {
        println("Coroutine lain mulai")
        delay(500L)
    }
}

```

```
println("Coroutine lain selesai")
}

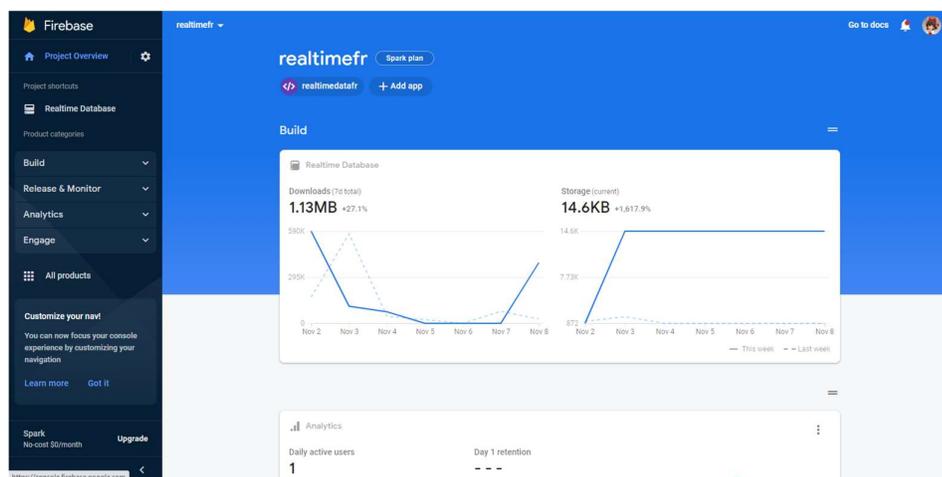
// Tugas utama menunggu sampai coroutine selesai
println("Menunggu coroutine...")
delay(1500L)
println("Program selesai")
}
```

I. Firebase

Firebase adalah *Cloud Service Provider* dan *backend as a service* yang dimiliki *Google*. *Firebase* merupakan solusi yang ditawarkan oleh *Google* untuk mempermudah dalam pengembangan aplikasi *mobile* maupun *web* dan bersifat *Realtime Database*. *Firebase Database* merupakan penyimpanan basis data *nonSQL* yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String*, *Long*, dan *Boolean*. (Andrianto & Munandar, 2022). *Firebase* adalah layanan dari *Google* yang memudahkan pengembang dalam membuat aplikasi mereka. *Firebase* membuat pengembang aplikasi untuk sepenuhnya fokus pada pembuatan aplikasi mereka. *Firebase Remote Config* dan *Firebase Realtime Database* adalah dua fitur unggulan dari *Firebase*. Selain itu, *Firebase Notification*, sebuah fitur yang penting untuk aplikasi, juga didukung.

Layanan *database* dan *backend* tersedia di *Firebase*. Layanan ini menyediakan antarmuka pengguna grafis (GUI) kepada pengembang untuk mengembangkan aplikasi, menyinkronkan semua data aplikasi mereka dengan akun *Firebase Cloud*. Perusahaan ini menyediakan perpustakaan untuk integrasi dengan *Android*, *iOS*, *JavaScript*, *Java*, *Swift*, dan *Node.js*.

Aturan keamanan yang telah ditetapkan oleh server suatu organisasi memastikan bahwa para pengembang dapat melindungi data mereka secara real-time. Generasi mendatang dari *Firebase Realtime Database*, yaitu Cloud Firestore, telah dirilis untuk versi beta. *Firebase Realtime Database* merupakan database yang dihosting di cloud. Setiap klien yang terhubung menerima data dalam format *JSON* dan disinkronkan secara real-time.



Gambar 2. 9. Tampilan *firebase realtime database*

J. Robotdyn AC Dimmer

Dimmer adalah perangkat yang terhubung ke perlengkapan perangkat variabel dan digunakan untuk menurunkan kecerahan perangkat variabel. Dengan mengubah bentuk gelombang tegangan yang diterapkan ke perangkat tersebut, intensitas keluaran perangkat dapat diturunkan. Meskipun perangkat tegangan variabel digunakan untuk berbagai keperluan, istilah *dimmer* secara khusus ditujukan untuk perangkat yang dimaksudkan untuk mengendalikan keluaran cahaya dari lampu pijar resistif dan lampu neon kompak (CFL) (Singh et al., 2022).

RobotDyn AC Dimmer adalah perangkat yang digunakan untuk mengatur intensitas cahaya atau kecepatan motor yang menggunakan daya listrik *AC*. Prinsip kerjanya didasarkan pada pengaturan fase, di mana *dimmer* mengontrol waktu penyediaan tegangan ke beban selama setiap siklus *AC*. Dengan memanipulasi sudut fase, *RobotDyn AC Dimmer* dapat mengatur jumlah daya yang diberikan ke perangkat terhubung, sehingga memungkinkan pengguna untuk mengontrol tingkat kecerahan lampu atau kecepatan motor. Beberapa versi *RobotDyn AC Dimmer* dapat diintegrasikan dengan mikrokontroler atau sistem otomatisasi rumah pintar, memberikan fleksibilitas tambahan dalam penerapannya.



Gambar 2. 10. *AC dimmer*

Modul *RobotDyn AC Dimmer* ini sering digunakan bersama mikrokontroler seperti *Arduino* atau *ESP8266/ESP32*, yang memungkinkan pengguna untuk mengontrol intensitas cahaya atau kecepatan motor melalui software. Berikut adalah beberapa komponen utama yang terdapat pada *RobotDyn AC Dimmer*:

1. *TRIAC*: Komponen utama yang bertindak sebagai saklar elektronik, memungkinkan pengendalian arus *AC* melalui teknik phase-cut.
2. Optoisolator (*Optocoupler*): Digunakan untuk memisahkan sirkuit daya tinggi (*AC*) dari sirkuit kontrol tegangan rendah (*DC*, seperti mikrokontroler). Komponen ini menjaga keamanan sistem dan mencegah interferensi listrik.
3. *Zero-cross Detection Circuit*: Sebuah rangkaian yang mendeteksi titik nol dari gelombang *AC* (di mana gelombang melintasi sumbu nol). Deteksi ini penting untuk mengetahui kapan memicu *TRIAC* agar kontrol phase-cut bisa dilakukan dengan tepat.
4. Resistor dan Kapasitor: Komponen pasif ini digunakan untuk mengatur arus, tegangan, dan waktu dalam rangkaian. Resistor dan kapasitor juga membantu dalam pembentukan jaringan *RC* untuk mengurangi lonjakan tegangan atau osilasi.
5. Diode: Dioda digunakan untuk melindungi komponen lain dalam rangkaian dari tegangan berlebihan yang mungkin terjadi saat switching atau perubahan dalam arus *AC*.
6. *Heatsink* (Pendingin): *TRIAC* sering kali disertai dengan heatsink untuk mengelola panas yang dihasilkan selama pengoperasian, terutama ketika beban yang dikontrol cukup besar.
7. *Pin Header* atau Terminal: Tempat untuk menghubungkan modul dimmer dengan sirkuit mikrokontroler atau dengan perangkat *AC* yang ingin dikontrol.

K. Dimmable dan Non Dimmable Bulb

(Ledvance, 2022) Sebelum ada lampu LED, lampu pijar jaman dulu sangat mudah diredupkan. Selama saklar dan sirkuit yang sesuai tersedia untuk perlengkapan tersebut, bohlam yang kompatibel dapat menciptakan tingkat pencahayaan yang berbeda. Dengan perkembangan teknologi LED, banyak hal telah berubah. Meskipun kita masih dapat menemukan bohlam yang dapat diredupkan dan tidak dapat diredupkan, proses pemasangannya menjadi sedikit lebih rumit. Tidak selalu mungkin untuk mengontrol tingkat kecerahan, terutama jika Anda tidak memiliki kompatibilitas peredup yang disertakan dengan pengaturan.

Lampu LED yang dapat diredupkan berarti kita dapat mengatur tingkat kecerahan atau lumennya sesuai dengan kebutuhan kita. Lampu led yang tidak dapat diredupkan dapat memberi kita pengalaman pencahayaan yang lebih konsisten. Tingkat lumen akan selalu sama saat kita menyalakannya dan tidak ada kemampuan untuk mengontrolnya. Setiap jenis lampu LED mempunyai kelebihan dan kekurangannya masing-masing, Anda bisa membelinya sesuai kebutuhan. Misalnya, lampu yang dapat diredupkan lebih cocok digunakan di sekolah dan rumah, sedangkan lampu yang tidak dapat diredupkan cocok digunakan di gudang.

LED dapat diredupkan melalui CCR (pengurangan arus konstan) atau modulasi lebar pulsa (PWM), sehingga menghasilkan lebih banyak potensi penghematan energi setelah dipasang dengan benar. Peredupan CCR adalah versi analog dari teknologi ini. Arus maju yang disalurkan ke bohlam dikontrol,

mengurangi levelnya untuk menghasilkan cahaya yang dapat diredupkan. Dengan teknologi peredupan PWM, LED dimatikan dan dihidupkan dengan kecepatan tinggi. Siklus tepat waktu dibagi menjadi interval milidetik untuk menciptakan efek peredupan. Ia berkedip sangat cepat sehingga mata manusia tidak menyadari perubahannya. Selama lajunya tetap cukup tinggi, denyutnya tidak terlihat.

L. Internet of Things (IOT)

Internet of Thing atau IoT adalah sebuah istilah yang dimaksudkan dalam penggunaan internet yang lebih besar, mengadopsi komputasi yang bersifat mobile dan konektivitas kemudian menggabungkannya kedalam kehidupan sehari-hari. IoT berkaitan dengan DoT (*Disruption of Things*) dan sebagai pengantar perubahan atau transformasi penggunaan internet dari sebelumnya Internet of People menjadi Internet of M2M (*Maching-to-Machine*) (Natsir et al., 2019).

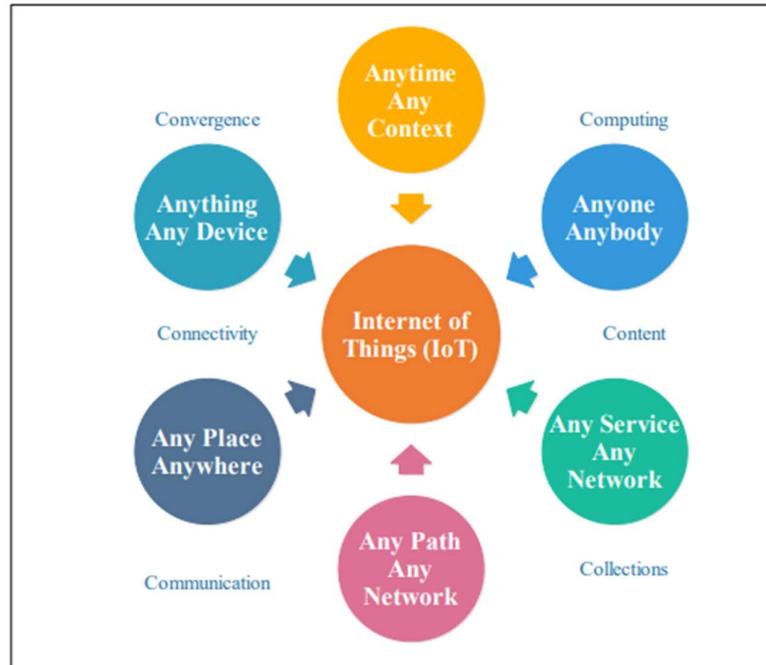
Internet of Things (IoT) merupakan konsep yang menghubungkan berbagai perangkat fisik melalui internet, memungkinkan mereka untuk saling berkomunikasi dan bertukar data. Dalam pengembangan IoT, berbagai objek seperti perangkat rumah tangga, kendaraan, dan alat kesehatan dapat dilengkapi dengan sensor, perangkat lunak, dan teknologi lainnya untuk meningkatkan fungsionalitasnya. Perangkat-perangkat ini dapat bekerja secara otomatis, mengambil keputusan berdasarkan data yang dikumpulkan, dan mengurangi kebutuhan intervensi manusia. IoT juga memungkinkan pemantauan jarak jauh dan memberikan efisiensi dalam berbagai aspek kehidupan sehari-hari.

Salah satu bidang penerapan IoT yang semakin berkembang adalah dalam industri, di mana konsep ini dikenal sebagai Industrial Internet of Things (IIoT). IIoT memungkinkan perusahaan untuk meningkatkan produktivitas dan efisiensi melalui pemantauan dan pengendalian otomatis mesin dan peralatan. Data yang dikumpulkan dari berbagai sensor di mesin dapat digunakan untuk melakukan prediksi pemeliharaan, mengurangi waktu henti, serta meningkatkan kualitas produk. Hal ini mengubah cara perusahaan mengelola rantai pasok dan operasional pabrik, memberikan dampak yang signifikan terhadap biaya dan keuntungan.

Selain di bidang industri, IoT juga memainkan peran penting dalam pengembangan kota pintar (*smart cities*). Di kota pintar, berbagai infrastruktur seperti transportasi, utilitas, dan keamanan dipantau dan dioptimalkan secara real-time. Sistem transportasi cerdas, misalnya, dapat menyesuaikan lampu lalu lintas untuk mengurangi kemacetan berdasarkan data dari sensor yang terpasang di jalan. Sementara itu, penggunaan energi dapat dioptimalkan melalui pengaturan otomatis pada jaringan listrik yang terhubung dengan IoT, yang memungkinkan penghematan energi dan pengurangan dampak lingkungan.

Berbagai jenis perangkat dapat termasuk dalam ekosistem *Internet of Things* (IoT), yang umumnya dikategorikan berdasarkan fungsi dan kegunaannya. Perangkat rumah tangga pintar (*smart home devices*) seperti lampu pintar, termostat pintar, dan kunci pintu otomatis adalah contoh alat IoT yang banyak digunakan di rumah. Lampu pintar dapat dikendalikan dari jarak jauh melalui aplikasi, sementara termostat pintar dapat mengatur suhu ruangan secara otomatis berdasarkan preferensi pengguna. Selain itu, ada perangkat keamanan rumah seperti kamera

pengawas dan detektor gerak yang terhubung ke jaringan internet untuk memberikan perlindungan lebih efisien.



Gambar 2. 11. Konsep iot

Sumber: (Joseph et al., 2020)

M. *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak (Pressman). UML bisa saja digunakan untuk visualisasi, spesifikasi, konstruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak. Dalam kata lain, seperti halnya seorang arsitek dalam membuat dokumen cetak biru yang digunakan oleh perusahaan konstruksi untuk membangun sebuah bangunan, arsitek perangkat lunak membuat diagram-diagram UML untuk membantu programmer/ developer membangun

perangkat lunak. Untuk selanjutnya, semakin kita mengetahui beberapa kosakata yang digunakan UML, kita akan semakin mudah dalam memahami spesifik (Sumiati et al., 2021). Unified Modelling Language (UML) menyediakan beberapa jenis diagram yang dapat dikelompokkan berdasar sifatnya antara lain:

1. Diagram *Use Case*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Dharwiyanti & Wahono, 2003).

Tabel 2. 5. Simbol *use case diagram*

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).

No	Gambar	Nama	Keterangan
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		Association	menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemenelemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

2. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. (Dharwiyanti & Wahono, 2003).

Tabel 2. 6. Simbol *Activity diagram*

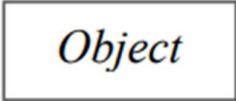
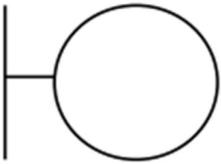
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

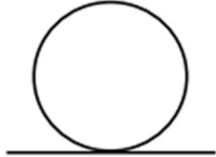
3. *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. (Dharwiyanti & Wahono, 2003).

Tabel 2. 7. Simbol *sequence diagram*

No	Gambar	Nama	Keterangan
1		Object	Object atau biasa juga disebut partisipan merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma.
2		Actor	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom.
3		Life line	Life line mengidentifikasi keberadaan sebuah object dalam basis waktu. Notasi untuk lifeline adalah garis putus-putus vertical yang di tarik oleh sebuah object
4		Collaborations	Activication dinotasikan sebagai kotak segi empat yang digambar pada sebuah lifeline action mengidentifikasi sebuah object yang akan beraksi.
5		Boundary	Boundary terletak diantara system dengan dunia di sekelilingnya. Semua form, laporan-laporan, antara muka ke perangkat keras seperti printer atau scenner dan antara muka ke system lainnya adalah termasuk dalam kategori.

No	Gambar	Nama	Keterangan
6		Control	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		Entity	Entity digunakan untuk menengani informasi yang akan disimpan secara permanen.

4. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Dharwiyanti & Wahono, 2003).

Tabel 2. 8. Simbol *class diagram*

No	Gambar	Nama	Keterangan
1		Assosiation	Hubungan statis antar kelas. Asosiasi menggambarkan kelas yang memiliki atribut berupa kelas lain, atau kelas yang harus mengetahui eksistensi kelas lain. Asosiasi biasanya disertai dengan multiplicity

No	Gambar	Nama	Keterangan
2		Directed Assosiation	asosiasi dengan makna kelas yang satu digunakan oleh kelas yang lain. Asosiasi bearah juga biasanya disertai dengan multiplicity
3		Generalization	relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) atau untuk menyatakan hubungan inheritance
4		Dependency	Relasi antar kelas dengan makna kebergantungan antar kelas
5		Agregation	Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain
6		Composition	Bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi whole. Misalnya kelas whole dihapus, maka kelas yang menjadi part ikut musnah
7		Realization	Hubungan antar kelas dimana sebuah kelas memiliki keharusan untuk mengikuti aturan yang ditetapkan oleh kelas lainnya.

N. Flowchart

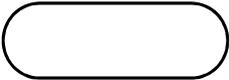
Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan symbol, dimana masing-masing symbol merepresentasikan suatu kegiatan tertentu (Hasyim, 2023). *Flowchart* dapat mempermudah pengembangan program karena dapat membantu pengembang dalam memahami alur kerja dari program yang dibangun.

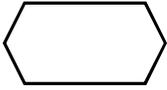
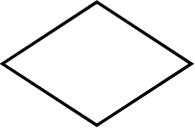
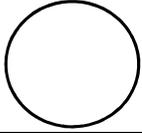
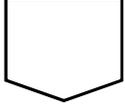
Menurut buku "Information Systems Development: Reflections, Challenges and New Directions" karya Robert D. Galliers dan Wendy L. Currie, *flowchart* merupakan alat bantu yang sangat penting dalam pengembangan sistem informasi. *Flowchart* dapat digunakan untuk merepresentasikan proses bisnis atau alur kerja dari suatu sistem informasi. Selain itu, *flowchart* juga dapat digunakan sebagai alat untuk mengevaluasi dan mengoptimalkan proses bisnis atau sistem informasi yang sudah ada.

Flowchart juga dapat digunakan untuk memudahkan komunikasi antara pengembang program dengan stakeholder atau pengguna akhir. Dalam buku "*System Analysis and Design: An Object-Oriented Approach with UML*" karya Alan Dennis, Barbara Haley Wixom, dan David Tegarden, disebutkan bahwa *flowchart* dapat digunakan untuk memudahkan komunikasi antara pengembang program dengan pengguna akhir. *Flowchart* dapat membantu pengguna akhir dalam memahami alur kerja dari program yang dibangun sehingga dapat memberikan masukan yang lebih baik dalam proses pengembangan program.

Berikut adalah beberapa simbol yang digunakan dalam membuat suatu diagram alur.

Tabel 2. 9. Simbol *flowchart*

Simbol	Nama	Fungsi
	<i>Terminator</i>	Menandai awal dan akhir dari suatu <i>flowchart</i>

Simbol	Nama	Fungsi
	Garis alir (<i>flow line</i>)	Menghubungkan setiap langkah dalam flowchart dan menunjukkan kemana arah aliran diagram.
	<i>Preparation</i> atau Persiapan	Menandai proses inialisasi atau pemberian nilai awal
	Proses	Melambangkan kegiatan pemrosesan input.
	<i>Input / output data</i>	Digunakan untuk menunjukkan proses input/output data, parameter, atau informasi.
	prosedur	Berperan sebagai blok pembangun dari suatu program
	<i>Decision</i> (Keputusan)	Digunakan untuk melambangkan percabangan, yaitu pemeriksaan suatu kondisi.
	<i>On page connector</i> (Penghubung pada Halaman)	Menghubungkan suatu langkah dengan langkah lain dari flowchart dalam satu halaman
	<i>Off page connector</i> (Penghubung di luar halaman)	Menghubungkan suatu langkah dengan langkah lain dari flowchart dalam halaman yang berbeda

O. Kerangka Pikir

Pencahayaan yang tepat dalam ruangan sangat penting untuk kenyamanan, namun pencahayaan yang tidak diatur dengan baik yang menyebabkan ketidaknyamanan terutama dalam situasi yang berbeda, seperti siang hari atau malam hari yang mempengaruhi tingkat cahaya yang diperlukan.



Pencahayaan yang tidak sesuai dan tidak diatur dengan baik dalam ruangan dapat menimbulkan ketidaknyamanan bagi penghuninya.



Membuat aplikasi pengatur tingkat pencahayaan dalam suatu ruangan dengan menggunakan *NodeMCU esp8266*, *AC Dimmer* dan lampu *dimmable*.



Telah dihasilkan sebuah aplikasi pengatur tingkat pencahayaan lampu pada ruangan berbasis android.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Metode yang diterapkan dalam penelitian ini adalah metode eksperimental. Peneliti akan merancang dan membuat prototype alat pengatur tingkat kecerahan dan aplikasi androidnya. Setelah itu, dilakukan pengujian dan evaluasi terhadap performa alat pengatur tersebut.

B. Lokasi dan Waktu

1. Lokasi

Penelitian ini dilakukan di lingkup Universitas Muhammadiyah Parepare, dengan durasi pelaksanaan penelitian selama kurang lebih satu bulan.

2. Waktu

Penelitian ini dilaksanakan selama satu bulan.

C. Alat dan Bahan

Dalam melaksanakan penelitian untuk pembuatan aplikasi, dibutuhkan penggunaan perangkat keras dan perangkat lunak sebagai pendukung kegiatan penelitian tersebut. Berikut merupakan beberapa hardware dan software yang digunakan.

1. Perangkat keras

Perangkat keras yang digunakan dalam pembuatan aplikasi terdapat dalam tabel berikut:

Tabel 3.1 Spesifikasi perangkat keras

No	Spesifikasi	
1	Merk laptop	DELL Latitude E6440
2	<i>Processor</i> laptop	Intel(R) Core(TM) i5-430M CPU @ 2.60 GHz (4CPUs), ~2.6GHz
3	<i>RAM</i> laptop	8 (8192) GB
4	Mikrokontroler	<i>NodeMCU Esp8266, Robotdyn AC Dimmer Module</i>
5	Komponen tambahan	<i>Dimmable AC LED Bulb, LED, kabel jumper, fitting lampu, kabel listrik, kabel mikro usb.</i>

2. Perangkat lunak

Perangkat lunak yang digunakan untuk pembuatan aplikasi dan mengunggah kode ke dalam mikrokontrol dapat dilihat sebagai berikut:

Tabel 3.2 Spesifikasi perangkat lunak

No	Spesifikasi	
1	Sistem operasi	Windows 10
2	Tool pemrograman	Arduino IDE, Android Studio, Fritzing
3	Bahasa pemrograman	C / C++, Kotlin

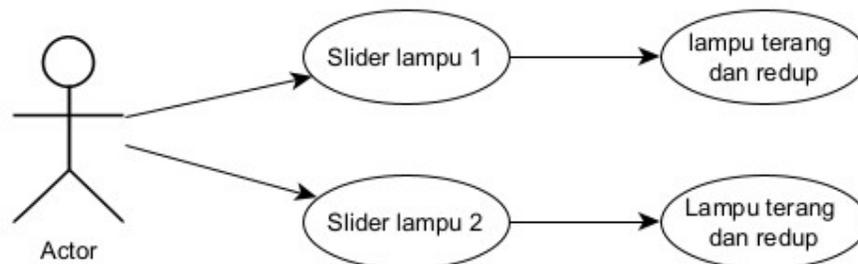
D. Desain Sistem

Perencanaan sistem dilakukan untuk memperoleh pemahaman yang jelas dan mempermudah implementasi serta evaluasi dari sistem yang direncanakan. Dalam merencanakan suatu sistem, diperlukan penggunaan flowchart, diagram use

case, diagram aktivitas, desain antarmuka pengguna (UI), sistem-sistem yang sudah ada, serta ide-ide yang relevan.

Use case merupakan alat bantu grafis untuk merepresentasikan model kebutuhan sehingga dapat menjelaskan aktivitas-aktivitas apa saja yang harus dikerjakan oleh sistem dan menjelaskan perilaku dari komponen-komponen sistem. (Astuti, 2009).

Dalam membuat diagram *use case*, perlu ditentukan pengguna mana yang akan menggunakan sistem tersebut. Aktor adalah nama pengguna, dan aktor harus berkomunikasi dengan sistem. Hubungan antara aktor dan *Use Cases* bersifat linear. Gambar 3.1 menampilkan diagram use-case dari sistem tersebut.



Gambar 3.1 *Use case diagram*

Pada gambar 3.1 *Actor* melakukan inialisasi dan menjalankan mikrokontroler tersebut, kemudian actor mengatru tingkat kecerahan melalui aplikasi dan sistem akan mendeteksi sinyal cahaya lalu *AC dimmer* melakukan peredupan dan penerangan lampu.

E. Metode Pengumpulan Data

Penelitian ini akan difokuskan pada perancangan dan pembuatan prototipe alat pengatur tingkat pencahayaan lampu berbasis android menggunakan *AC Light Dimmer Module* dan NodeMCU Esp8266. Kemudian pengujian dan evaluasi akan dilakukan di ruangan menggunakan metode *whitebox* dan *blackbox*.

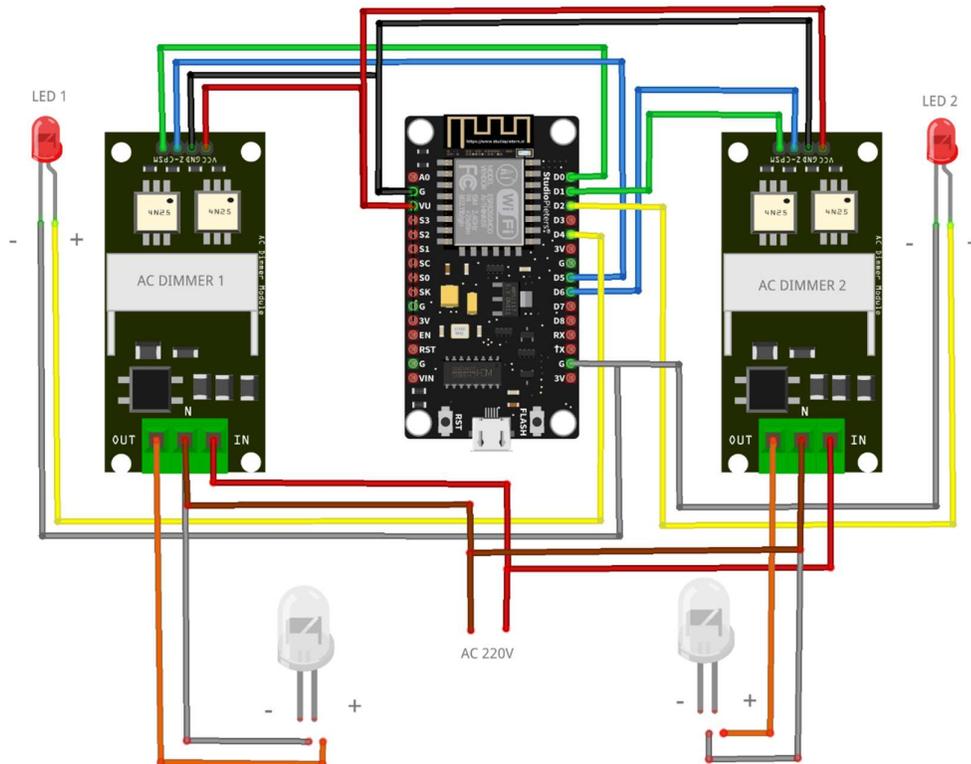
BAB IV

HASIL DAN PEMBAHASAN

A. Rancangan Sistem

Pada dasarnya hasil penelitian ini menghasilkan 2 unsur utama yaitu perangkat keras dan perangkat lunak.

1. Perangkat Keras (Hardware)



Gambar 4. 1. Diagram Pengkabelan

Pada gambar 4.1 menjelaskan *NodeMCU esp8266* dihubungkan pada dua *AC Dimmer*, *AC Dimmer 1* untuk mengontrol lampu dimmable 1 dan *AC Dimmer 2* untuk mengontrol lampu dimmable 2. *NodeMCU esp8266* dihubungkan pada wifi

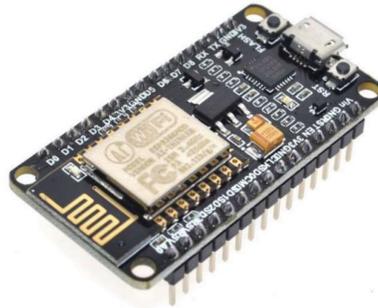
untuk mendapatkan konektivitas internet, lalu *NodeMCU* mengambil nilai berupa integer dari database *Firebase* kemudian diteruskan kepada kedua *AC Dimmer*.

Tabel 4.1 Pengoneksian pin *NodeMCU Esp8266*

No	Pin	Keterangan
1	D2	Dihubungkan ke Anoda <i>LED 1</i>
2	D4	Dihubungkan ke Anoda <i>LED 2</i>
3	GND	Dihubungkan ke GND <i>AC Dimmer</i> , Katoda <i>LED 1</i> dan <i>LED 2</i>
4	D5	Dihubungkan ke zero-cross pin <i>AC Dimmer</i>
5	D0	Dihubungkan ke pwm pin <i>AC Dimmer 1</i>
6	D6	Dihubungkan ke zero-cross pin <i>AC Dimmer</i>
7	D1	Dihubungkan ke pwm pin <i>AC Dimmer 2</i>
8	VU	Dihubungkan ke VCC <i>AC Dimmer 1</i> dan <i>AC Dimmer 2</i>

a. *NodeMCU Esp8266*

NodeMCU Esp8266 digunakan untuk mengambil value berupa nilai integer dari *firebase* untuk diteruskan ke *AC Dimmer*, value tersebut digunakan untuk mengatur tingkat pencahayaan lampu. *NodeMCU* dihubungkan pada dua *AC Dimmer*, yaitu *AC Dimmer 1* dan *AC Dimmer 2*, masing-masing *AC Dimmer* dihubungkan pada lampu *dimmable*.

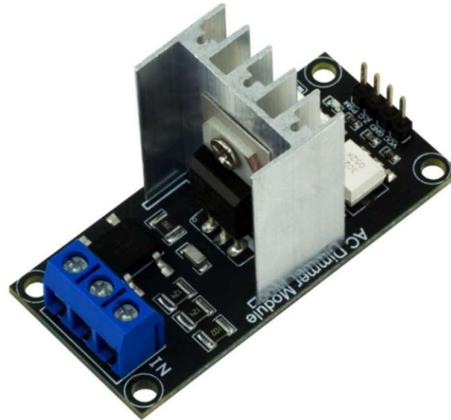


Gambar 4. 2. *NodeMCU Esp8266*

b. *AC Dimmer*

RBD AC Light Dimmer adalah modul yang digunakan untuk mengontrol intensitas cahaya lampu AC dengan mengatur tegangan yang diterima oleh lampu. Prinsip kerjanya didasarkan pada teknik *phase-cut dimming*, di mana bagian tertentu dari gelombang sinusoida arus AC dipotong untuk mengurangi daya yang diterima lampu. Arus listrik yang dialirkan ke lampu adalah arus bolak-balik (AC) berbentuk gelombang sinusoida, yang memiliki dua siklus, yaitu positif dan negatif. Dalam pengoperasiannya, *dimmer* ini menggunakan *triac* atau *thyristor* untuk memotong gelombang pada fase tertentu. Semakin awal fase dipotong, semakin sedikit daya yang diterima lampu, sehingga intensitas cahayanya menurun. *Triac* berfungsi sebagai saklar elektronik untuk mengatur kapan arus diaktifkan atau dinonaktifkan berdasarkan timing yang diatur oleh pengguna. Kontrol *dimmer* dilakukan melalui potensiometer atau pengaturan elektronik lain yang mengubah waktu pemicuan *triac*, sehingga cahaya dapat diatur naik atau turun sesuai kebutuhan. Selain itu, *dimmer* juga sering menggunakan

zero-cross detection, yakni mendeteksi titik ketika gelombang *AC* melewati nol, untuk menentukan waktu yang tepat dalam memicu triac dan memastikan pengaturan fase yang akurat.



Gambar 4. 3. *AC Dimmer*

c. *TRIAC (Triode for Alternating Current)*

TRIAC, atau *triode for alternating current* adalah komponen semikonduktor yang berfungsi sebagai saklar elektronik untuk mengontrol arus *AC* (arus bolak-balik) (Widiastuti, 2023). *TRIAC* dapat menghantarkan arus di kedua arah, baik pada setengah siklus positif maupun negatif dari gelombang *AC*, sehingga cocok digunakan dalam pengaturan daya pada perangkat yang menggunakan listrik *AC*, seperti lampu, motor, dan pemanas. *TRIAC* mampu bekerja pada kedua siklus dan memungkinkan kontrol yang lebih efisien. *TRIAC* sering digunakan dalam aplikasi *dimmer*, kontrol kecepatan motor, dan regulator daya karena kemampuannya untuk mengendalikan arus dengan mengatur fase gelombang *AC*.



Gambar 4. 4. TRIAC (*triode for alternating current*)

d. *Dimmable Bulb* (Lampu Peredupan)

Dimmable Bulb adalah lampu yang dapat diredupkan, lampu pijar dapat diredupkan karena cara kerjanya bergantung pada tegangan yang diterimanya. Lampu ini menghasilkan cahaya dengan memanaskan filamen logam, biasanya tungsten, hingga bersinar ketika arus listrik mengalir melaluinya. Intensitas cahaya yang dihasilkan berbanding lurus dengan jumlah arus listrik atau daya yang diberikan ke filamen. Ketika tegangan yang dialirkan dikurangi, seperti melalui penggunaan dimmer, arus yang mengalir melalui filamen juga berkurang, menyebabkan filamen memanaskan lebih sedikit dan menghasilkan cahaya yang lebih redup. Lampu pijar memiliki respon linear terhadap perubahan tegangan, sehingga mudah diredupkan dibandingkan jenis lampu lain seperti LED atau fluorescent, yang memerlukan kontrol khusus. Kemampuan ini berasal dari sifat filamen yang secara langsung menyesuaikan intensitas cahayanya berdasarkan perubahan tegangan listrik yang diterima. Lampu *LED* Hyderson dapat diredupkan karena dilengkapi dengan teknologi *dimmable LED driver*, yang memungkinkan kontrol arus listrik ke dioda pemancar cahaya, sehingga

intensitas cahaya dapat diatur. Teknologi ini memastikan pengaturan cahaya yang presisi, di mana arus yang dikurangi menghasilkan cahaya lebih rendah tanpa mengubah warna atau efisiensi energi, sehingga lampu dapat diredupkan secara halus tanpa *flicker* atau penurunan kualitas cahaya.



Gambar 4. 5. (a) Lampu pijar (b) led dimmable

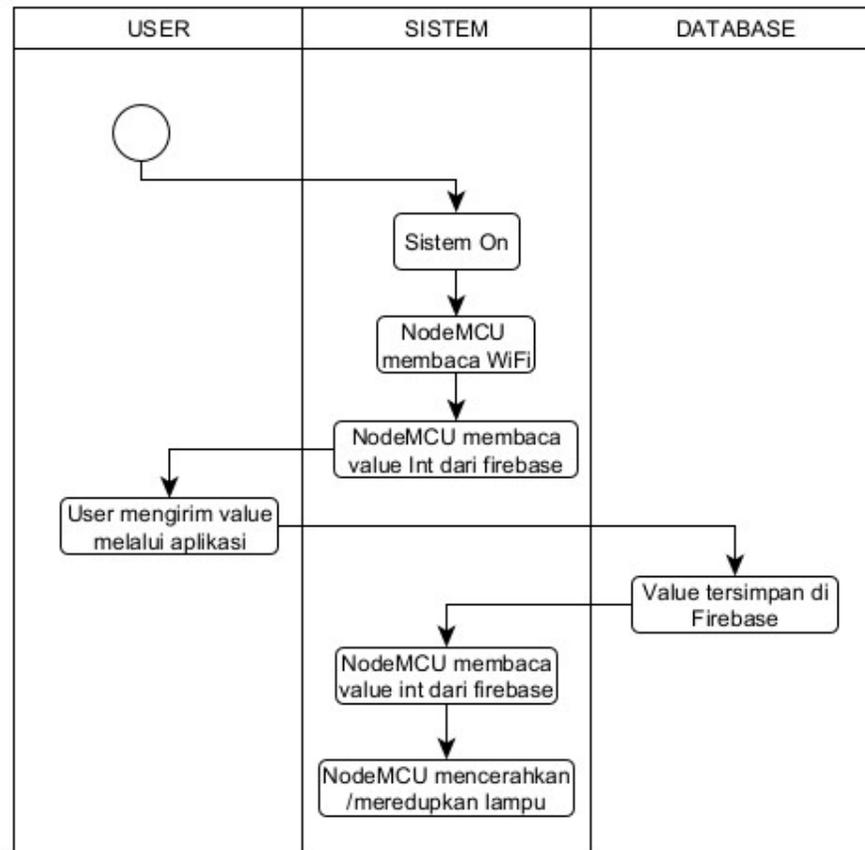
2. Perangkat Lunak (*Software*)

- a. Interface yang dibuat adalah sebagai berikut



Gambar 4. 6. Tampilan Aplikasi

b. Activity Diagram

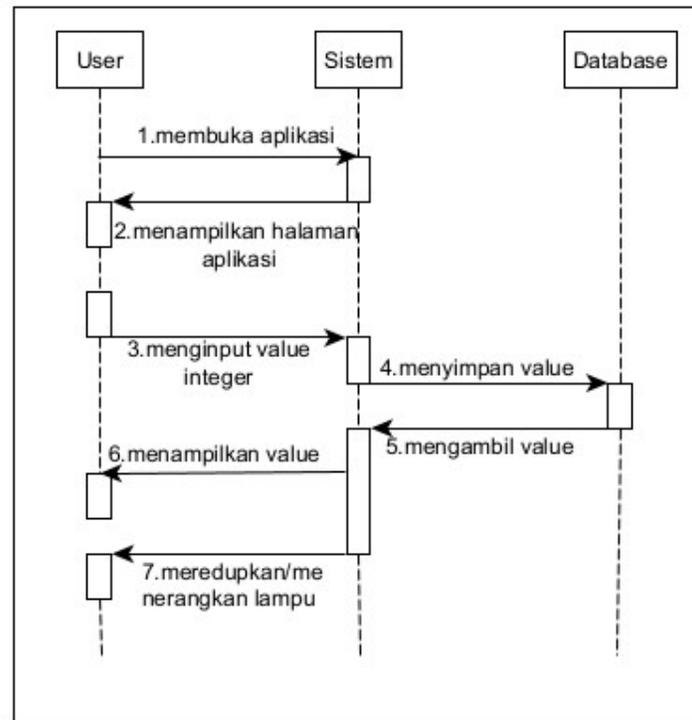


Gambar 4. 7. Activity Diagram

Pada gambar diatas menjelaskan hubungan antara *user*, sistem dan *database*. *Database* digunakan untuk menyimpan nilai integer yang akan digunakan oleh sistem dan di input oleh user melalui aplikasi. Awal mulanya sistem dinyalakan akan membaca nilai dari database yang kemudian sistem (*NodeMCU*) akan meneruskan nilai ini ke *AC Dimmer* agar *AC Dimmer* dapat menyesuaikan tingkat kecerahan lampu, kemudian user mengirim nilai integer ini melalui aplikasi android, nilai ini disimpan

di *database firebase* yang kemudian diambil lagi oleh *NodeMCU* untuk diteruskan kembali ke *AC Dimmer*.

c. Sequence Diagram



Gambar 4. 8. Sequence diagram

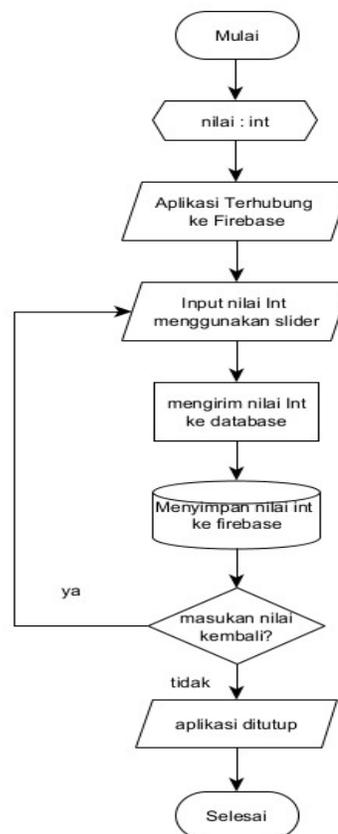
Pada gambar 4.8. menjelaskan urutan proses yang terjadi pada aplikasi pengatur tingkat pencahayaan lampu, mulanya user membuka aplikasi, kemudian sistem menampilkan halaman pengatur yang memiliki slider sebagai metode input value, value tersebut kemudian oleh sistem disimpan kedalam database yang selanjutnya digunakan oleh sistem pada hal ini yaitu alat untuk melakukan peredupan dan peningkatan cayaha lampu.

B. Pengujian Sistem

1. Pengujian *Whitebox*

Pengujian *whitebox* dilakukan untuk melihat hasil kontrol yang dilakukan NodeMCU Esp8266 berdasarkan perhitungan dan logika program untuk mengetahui efektifitas sistem yang telah dibangun.

a. *Whitebox* Aplikasi

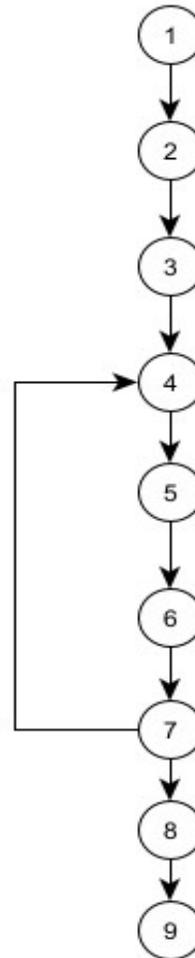


Gambar 4. 9. *Flowchart* Aplikasi

Pada flowchart diatas, ketika aplikasi dijalankan langkah pertama yaitu aplikasi melakukan konektivitas terhadap *cloud database* firebase. Kemudian aplikasi menyimpan nilai integer sesuai *input slider* yang

kemudian disimpan di dalam *cloud database* firebase. Nilai integer ini yang akan di ambil (retrieve) oleh NodeMCU untuk mengontrol *dimnable bulb*.

1) *Flowgraph* aplikasi



Gambar 4. 10. *Flowgraph* aplikasi

2) *Cyclomatic Complexity*

Menghitung cyclomatic complexity $V(G)$ dari Edge dan Node:

Menggunakan rumus $V(G) = E - N + 2$

E (Edge) = 9

$$N (\text{Node}) = 9$$

$$\begin{aligned} \text{Penyelesaian: } V(G) &= E - N + 2 \\ &= 9 - 9 + 2 \\ &= 2 \end{aligned}$$

3) Berdasarkan perhitungan *kompleksitas cyclomatic* dari *flowgraph* di atas, terdapat 2 region.

4) *Independent path* pada *flowgraph* di atas adalah:

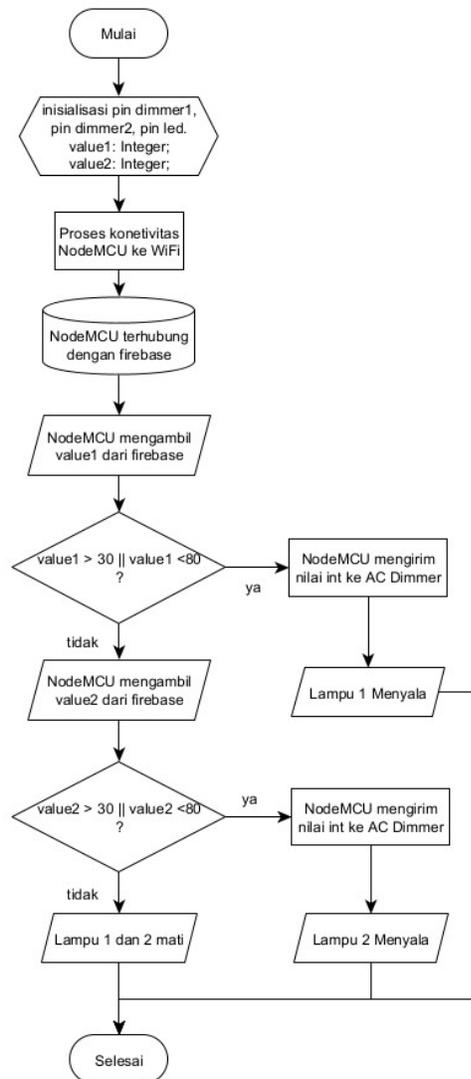
$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 4$$

5) Grafik matriks aktivasi *flowgraph*

Tabel 4.2 grafik matriks aktivasi *flowgraph* aplikasi

	1	2	3	4	5	6	7	8	9	E - 1
1		1								$1 - 1 = 0$
2			1							$1 - 1 = 0$
3				1						$1 - 1 = 0$
4					1					$1 - 1 = 0$
5						1				$1 - 1 = 0$
6							1			$1 - 1 = 1$
7				1				1		$2 - 1 = 1$
8									1	$1 - 1 = 0$
9										0
Sum (E) + 1										$1 + 1 = 2$

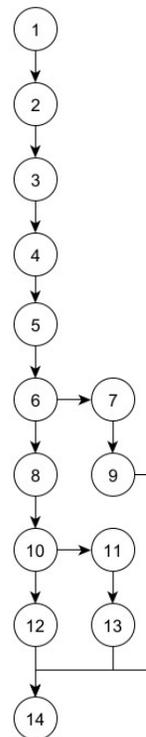
b. *Whitebox* Alat

Gambar 4. 11. *Flowchart* alat

Pada *flowchart* diatas, langkah pertama alat tersebut adalah NodeMCU menginisialisasi *pin out* yang akan digunakan dalam mengatur *AC Dimmer* module dan sebuah led, kemudian NodeMCU menghubungkan WiFi untuk mengakses *cloud database* firebase.

Setelah inisialisasi selesai selanjutnya NodeMCU mengambil *value1* yang bertipe *integer* untuk diteruskan pada modul *AC Dimmer 1*, dan mengambil *value2* yang bertipe *integer* untuk diteruskan pada modul *AC Dimmer 2*. Kemudian modul *AC Dimmer* mengatur tingkat kecerahan lampu sesuai nilai yang diterima dari NodeMCU.

1) *Flowgraph* alat



Gambar 4. 12. *Flowgraph* alat

2) *Cyclomatic Complexity*

Menghitung cyclomatic complexity $V(G)$ dari Edge dan Node:

Menggunakan rumus $V(G) = E - N + 2$

E (Edge) = 15

N (Node) = 14

$$\begin{aligned}
 \text{Penyelesaian: } V(G) &= E - N + 2 \\
 &= 15 - 14 + 2 \\
 &= 3
 \end{aligned}$$

3) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* di atas mempunyai *region* = 3

4) *Independent path* pada *flowgraph* di atas adalah:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 8 - 10 - 12 - 14$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 9 - 14$$

$$\text{Path 3} = 1 - 2 - 3 - 4 - 5 - 6 - 8 - 10 - 11 - 13 - 14$$

5) Grafik matriks aktivasi *flowgraph*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E - 1
1		1													1 - 1 = 0
2			1												1 - 1 = 0
3				1											1 - 1 = 0
4					1										1 - 1 = 0
5						1									1 - 1 = 0
6							1	1							2 - 1 = 1
7									1						1 - 1 = 0
8										1					1 - 1 = 0
9														1	1 - 1 = 0
10											1	1			2 - 1 = 1
11													1		1 - 1 = 0
12														1	1 - 1 = 0
13														1	1 - 1 = 0
14															0
Sum (E) + 1															2 + 1 = 3

2. Pengujian *Blackbox*

a. *Blackbox* aplikasi

Pengujian *blackbox* aplikasi bertujuan untuk mengetahui berfungsi atau tidaknya aplikasi yang telah dibuat. Pengujian dilakukan untuk melihat hasil kontrol yang dilakukan NodeMCU Esp8266 berdasarkan intruksi nilai yang diberikan oleh aplikasi android.

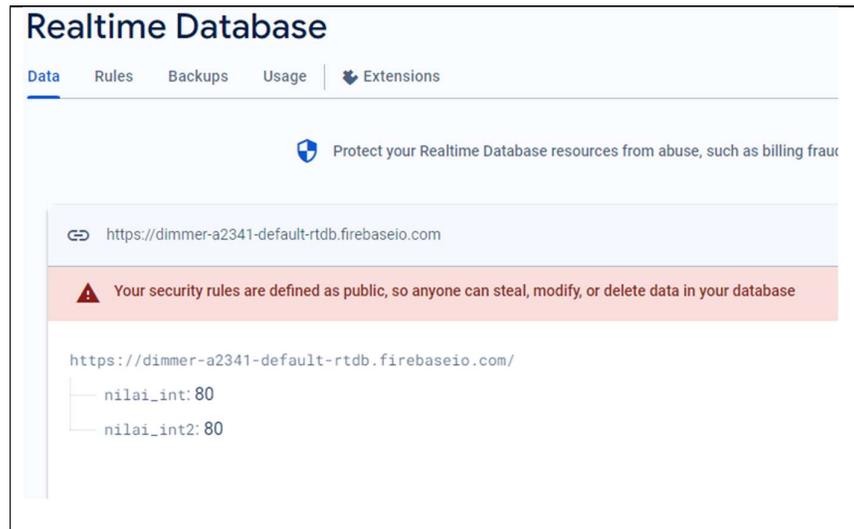
1) Detail aplikasi

Halaman home adalah tampilan aplikasi saat aplikasi dijalankan, halaman ini menampilkan *slider* yang digunakan untuk memberikan nilai integer yang kemudian akan disimpan pada *database* firebase.



Gambar 4. 13. Tampilan aplikasi

Pengujian ini dilakukan untuk melihat apakah aplikasi yang telah dibangun sudah berjalan dengan semestinya tanpa adanya *error* maupun *bug* yang terjadi. Pengujian ini juga dilakukan untuk melihat konektifitas antara aplikasi dan firebase cloud database.



Gambar 4.14. *Firebase*

2) Pengujian aplikasi

Pengujian aplikasi dilakukan untuk mengetahui jalannya aplikasi dan *database*.

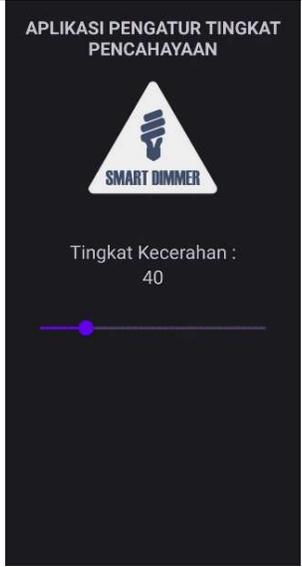
Tabel 4.3 *Black Box* tampilan input 30

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 30	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		

Tabel 4.4 *Black Box* tampilan input 35

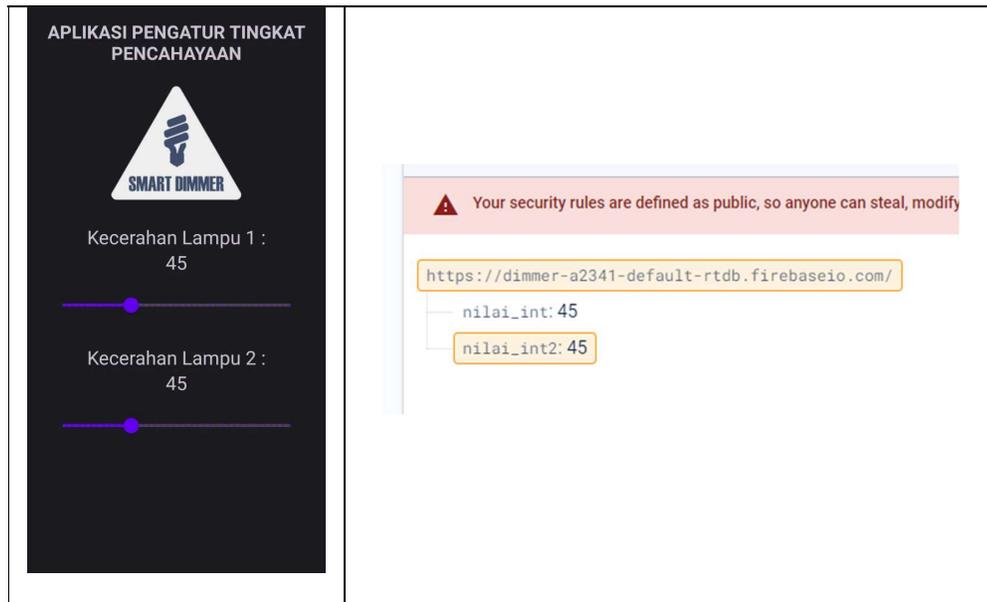
Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 35	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
<p>The screenshot shows the application interface with the title 'APLIKASI PENGATUR TINGKAT PENCAHAYAAN' and 'SMART DIMMER' logo. The sliders for 'Kecerahan Lampu 1' and 'Kecerahan Lampu 2' are now set to 35.</p>	<p>The screenshot shows the Realtime Database view with the same URL. The JSON object now contains: <code>{ "nilai_int": 35, "nilai_int2": 35 }</code>.</p>	

Tabel 4.5 *Black Box* tampilan input 40

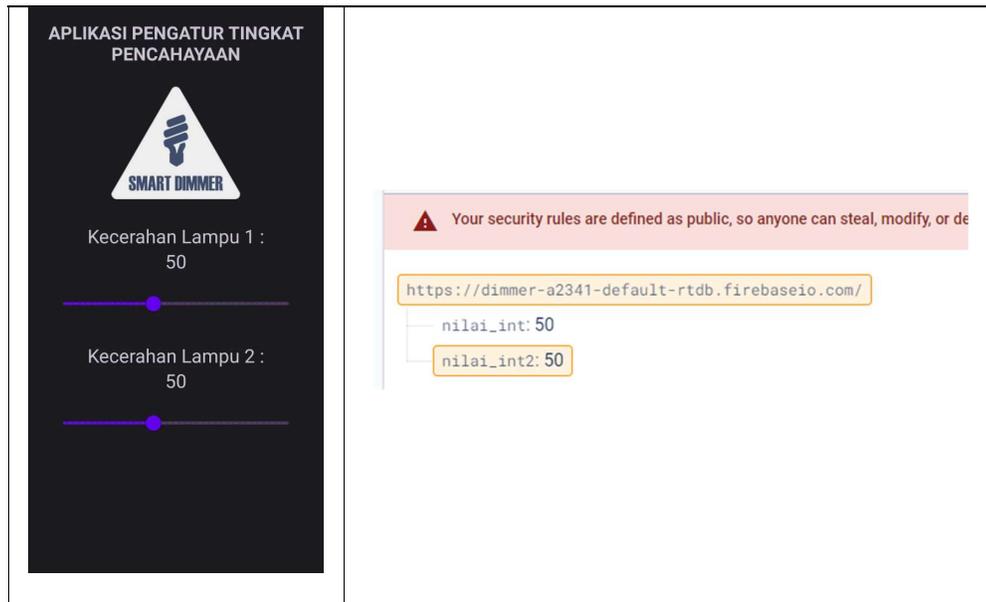
Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 40	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
		

Tabel 4.6 *Black Box* tampilan input 45

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 45	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		

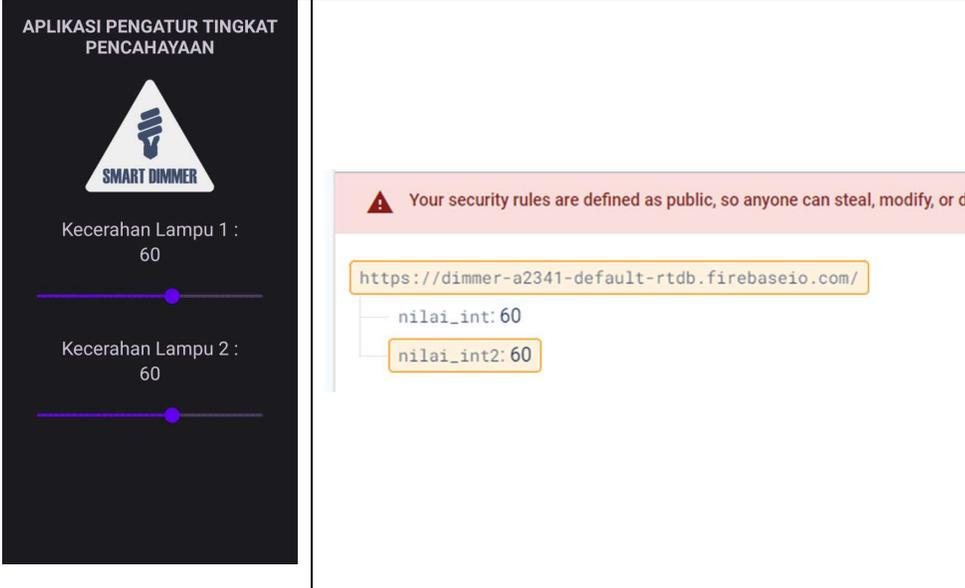
Tabel 4.7 *Black Box* tampilan input 50

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 50	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		

Tabel 4.8 *Black Box* tampilan input 55

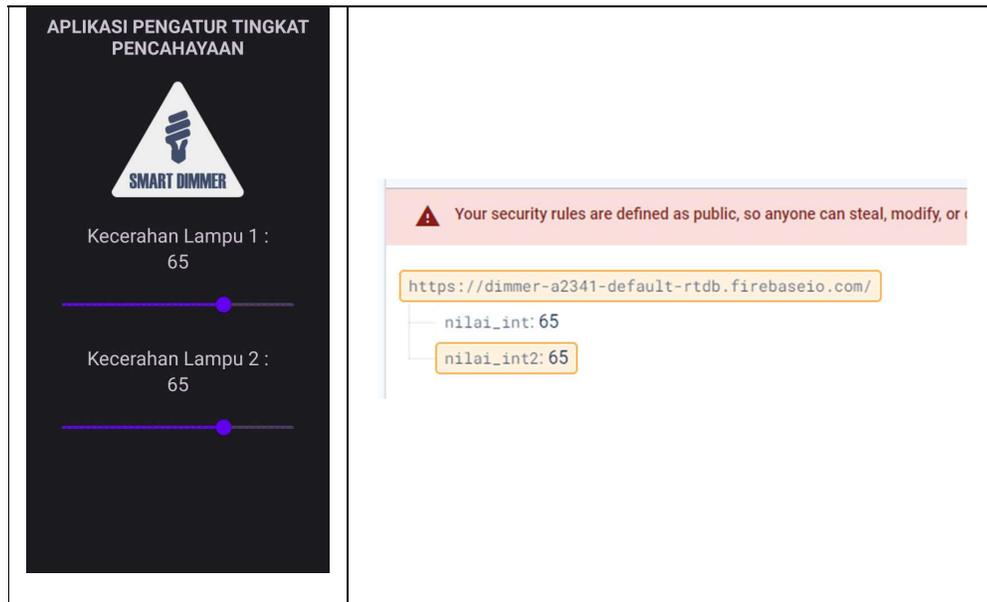
Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 55	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
<p>The image shows a side-by-side comparison. On the left is the application interface titled 'APLIKASI PENGATUR TINGKAT PENCAHAYAAN' (Light Level Control Application). It features a 'SMART DIMMER' logo and two sliders. The first slider is labeled 'Kecerahan Lampu 1 : 55' and the second is 'Kecerahan Lampu 2 : 55'. On the right is a screenshot of a database console. It displays a warning: 'Your security rules are defined as public, so anyone can steal, modify, or delete your data.' Below the warning, the URL 'https://dimmer-a2341-default-rtdb.firebaseio.com/' is shown, followed by a JSON object: { 'nilai_int': 55, 'nilai_int2': 55 }.</p>		

Tabel 4.9 *Black Box* tampilan input 60

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 60	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
 <p>The screenshot displays the 'SMART DIMMER' application interface on the left, which includes a title 'APLIKASI PENGATUR TINGKAT PENCAHAYAAN', a lightbulb icon, and two sliders for 'Kecerahan Lampu 1' and 'Kecerahan Lampu 2', both set to 60. On the right, a database view shows a JSON object with the following structure: <code>{ "nilai_int": 60, "nilai_int2": 60 }</code>. A warning message at the top of the database view states: 'Your security rules are defined as public, so anyone can steal, modify, or delete your data.'</p>		

Tabel 4.10 *Black Box* tampilan input 65

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 65	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		



Tabel 4 11 *Black Box* tampilan input 70

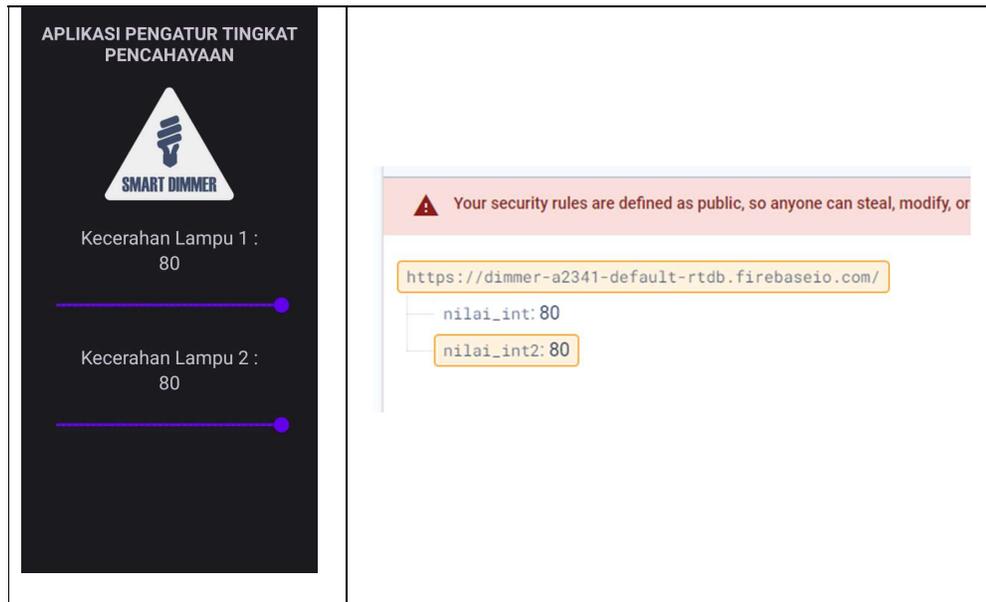
Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 70	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
<p>The image shows two side-by-side screenshots. The left screenshot is the application interface titled 'APLIKASI PENGATUR TINGKAT PENCAHAYAAN' (Lighting Level Control Application). It features a 'SMART DIMMER' logo and two sliders. The first slider is labeled 'Kecerahan Lampu 1 : 70' and the second is 'Kecerahan Lampu 2 : 70'. The right screenshot shows a database view with a warning message: 'Your security rules are defined as public, so anyone can steal, modify, or delete your data.' Below the warning, a JSON object is displayed: <code>{ "nilai_int": 70, "nilai_int2": 70 }</code>.</p>		

Tabel 4.12 *Black Box* tampilan input 75

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 75	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		
		

Tabel 4.13 *Black Box* tampilan input 80

Text Faktor	Hasil	Kesimpulan
Aplikasi menginput nilai 80	✓	Berhasil, nilai berhasil disimpan di database
<i>Screenshot</i>		



Aplikasi pengatur tingkat pencahayaan ini menggunakan dibangun menggunakan bahasa program kotlin. Aplikasi ini menggunakan slider yang berfungsi sebagai input yang kemudian input tersebut akan menyimpan nilai bertipe integer kedalam *cloud database* firebase. Nilai tersebut akan diambil oleh mikrokontroller NodeMCU Esp8266 dan digunakan untuk kontroller ac dimmer module dalam mengatur tingkat pencahayaan lampu *dimmable*.

Tabel 4.14 Pengujian Aplikasi dan database

Nilai	Respon Aplikasi	Respon Firebase	Keterangan
30	slider posisi 30	value bernilai 30	berhasil
35	slider posisi 35	value bernilai 35	berhasil
40	slider posisi 40	value bernilai 40	berhasil
45	slider posisi 45	value bernilai 45	berhasil
50	slider posisi 50	value bernilai 50	berhasil
55	slider posisi 55	value bernilai 55	berhasil
60	slider posisi 60	value bernilai 60	berhasil
65	slider posisi 65	value bernilai 65	berhasil
70	slider posisi 70	value bernilai 70	berhasil
75	slider posisi 75	value bernilai 75	berhasil

Nilai	Respon Aplikasi	Respon Firebase	Keterangan
80	slider posisi 80	value bernilai 80	berhasil

Berdasarkan pengujian diatas, aplikasi ini berhasil mengirimkan nilai ke dalam database firebase dengan tingkat akurasi 100%.

b. *Blackbox* Alat

1) Detail alat



Gambar 4. 15. Alat Pengatur Pencahayaan Lampu

2) Pengujian alat

Pengujian alat dilakukan untuk mengetahui berhasil tidaknya alat yang telah dibuat dalam mengambil nilai dari database dan diteruskan sebagai nilai penercahan lampu.

Tabel 4.15 Pengujian alat dan kecerahan lampu

Poin	Nilai Firebase	Respon Alat	Keterangan
30	value bernilai 30	Lampu menyala	berhasil
35	value bernilai 35	Lampu menyala	berhasil
40	value bernilai 40	Lampu menyala	berhasil
45	value bernilai 45	Lampu menyala	berhasil
50	value bernilai 50	Lampu menyala	berhasil
55	value bernilai 55	Lampu menyala	berhasil
60	value bernilai 60	Lampu menyala	berhasil
65	value bernilai 65	Lampu menyala	berhasil
70	value bernilai 70	Lampu menyala	berhasil
75	value bernilai 75	Lampu menyala	berhasil
80	value bernilai 80	Lampu menyala	berhasil

Berdasarkan pengujian diatas, alat berhasil menyalakan lampu dan mengubah kecerahannya berdasarkan nilai yang diambil dari database.

**Gambar 4. 16.** Kecerahan Lampu

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis maka dapat menarik kesimpulan sebagai berikut:

Berdasarkan penelitian yang dilakukan, aplikasi pengatur tingkat pencahayaan lampu berbasis Android telah berhasil dikembangkan dengan memanfaatkan teknologi *Internet of Things* (IoT). Sistem ini menggunakan *NodeMCU ESP8266* dan modul *AC Dimmer* untuk mengontrol intensitas cahaya pada lampu yang dapat diatur tingkat kecerahannya. Aplikasi Android berfungsi sebagai antarmuka pengguna yang memungkinkan kontrol pencahayaan dari jarak jauh dengan memanfaatkan *Firebase* sebagai platform penyimpanan data.

Pengujian sistem dilakukan melalui *whitebox* dan *blackbox* testing untuk memastikan fungsionalitas dan efektivitas sistem. Hasil pengujian menunjukkan bahwa aplikasi dapat berfungsi dengan baik, mampu mengirimkan dan menerima data dari *Firebase* dengan akurasi yang tinggi. Pengujian alat menunjukkan bahwa sistem berhasil mengontrol intensitas cahaya lampu sesuai dengan nilai yang dikirimkan dari aplikasi, yang diambil dari *database Firebase*.

Secara keseluruhan, penelitian ini berhasil menunjukkan bahwa dengan memanfaatkan teknologi *IoT* dan perangkat keras serta perangkat lunak yang tepat, sistem pengatur pencahayaan berbasis Android dapat diimplementasikan dengan

efektif. Penggunaan aplikasi ini dapat mendukung efisiensi energi dengan mengoptimalkan penggunaan pencahayaan dalam ruangan.

B. Saran

Dalam penelitian ini tentu saja peneliti menemukan kelemahan kelemahan pada alat yang dibuat, dengan demikian saran yang kami berikan untuk penelitian lebih lanjut terhadap penelitian ini adalah sebagai berikut:

1. Sistem ini masih menggunakan dua lampu *dimnable*, yang mana harapan kedepannya adalah pengembang mampu mendambahkan lampu, baik kontroller di dalam aplikasi android maupun lampu fisik lagi agar dikemudian hari penelitian ini benar-benar dapat diimplementasikan pada banyak ruangan secara terus-menerus.
2. Pengujian sistem ini masih menggunakan *NodeMCU Esp8266* dengan *Mobile Wi-Fi* sebagai perangkat jaringan internet, saran penulis ialah pengembang dapat menggunakan mikrokontroller yang lebih baik dari *NodeMCU Esp8266* dari segi pemrosesan maupun penanganan jaringan.

DAFTAR PUSTAKA

- Andrianto, R., & Munandar, M. H. (2022). Aplikasi E-Commerce Penjualan Pakaian Berbasis Android Menggunakan Firebase Realtime Database. *Journal Computer Science and Information Technology (Jcoint)*, 3(1), 20–29. <https://jurnal.ulb.ac.id/index.php/JCoInT/article/view/2478>
- Astuti, R. (2009). Pemodelan Analisis Berorientasi Objek dengan Use Case. *Media Informatika*, 8(2), 73–81. https://jurnal.likmi.ac.id/Jurnal/7_2009/Pemodelan_Analisis_rini_.pdf
- Developer Android. (2024). *Mengenal Android Studio*. <https://developer.android.com/studio/intro?hl=id>
- Dharwiyanti, S., & Wahono, R. S. (2003). Pengantar Unified Modeling Language (UML). *IlmuKomputer.Com*, 1–13. <http://www.unej.ac.id/pdf/yanti-uml.pdf>
- Hasyim, A. H. (2023). *Dasar Pemrograman* (3rd ed.). CV. Bangun Bumitama.
- Husnayain, F., Himawan, D. S., Utomo, A. R., Ardita, I. M., & Sudiarto, B. (2023). Analisis Perbandingan Kinerja Lampu LED, CFL, dan Pijar Pada Sistem Penerangan Kantor. *Cyclotron*, 6(01), 78–83.
- Joseph, S. B., Dada, E. G., & Abdullahi, M. S. (2020). Development of Internet of Things (IoT) Based Energy Consumption Monitoring and Device Control System. *NIPES Journal of Science and Technology Research*, 2(3), 85. <https://doi.org/10.37933/nipes/2.3.2020.9>
- Ledvance. (2022). *What is the Difference: Dimmable vs Non-Dimmable Bulbs*. <https://www.ledvanceus.com/blog/Pages/Dimmable-vs-Non-Dimmable.aspx>
- Mulyanto, A., Nurhuda, Y. A., & Khoirusid, I. (2017). Sistem Kendali Lampu Rumah Menggunakan Smartphone Android. *Jurnal Teknoinfo*, 11(2), 48. <https://doi.org/10.33365/jti.v11i2.28>
- Natsir, M., Rendra, D. B., & Anggara, A. D. Y. (2019). Implementasi IOT Untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya. *Jurnal PROSISKO (Pengembangan Riset Dan Observasi Rekayasa Sistem Komputer)*, 6(1), 69–72.
- Satria, B. (2022). IoT Monitoring Suhu dan Kelembaban Udara dengan Node MCU ESP8266. *Sudo Jurnal Teknik Informatika*, 1(3), 136–144. <https://doi.org/10.56211/sudo.v1i3.95>
- Septian, O. :, & Aji, P. (2018). *Alat Monitoring Tetesan Infus...(Septian Prastyo Aji) 78 INFUSING MONITORING TOOLS USING WEB ONLINE BASED ESP8266 WITH ARDUINO IDE PROGRAMMING*. 7(1), 78–86.

- Sibarani, N. S., Munawar, G., & Wisnuadhi, B. (2018). Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java Dan Kotlin. *9th Industrial Research Workshop And National Seminar, July*, 319–324.
- Singh, B., Bir, M., & Gulshan, S. (2022). *AC Dimmer*.
- Sudin, N., Djufri, I., & Umar, M. K. G. (2020). Rancang Bangun Sistem Pengontrol Lampu Rumah Berbasis Mikrokontroler Arduino Uno Menggunakan Smartphone. *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*, 3(2), 52–61. <https://doi.org/10.47324/ilkominfo.v3i2.102>
- Sumiati, M., Abdillah, R., & Cahyo, A. (2021). *Uml 21. 11(2)*, 79–86.
- Widiastuti, S. (2023). Rancang Bangun Alat Pengatur Intensitas Cahaya Lampu Pijar Menggunakan Potensiometer. *Elektriase: Jurnal Sains Dan Teknologi Elektro*, 12(02), 141–150. <https://doi.org/10.47709/elektriase.v12i02.1658>
- Wiharto, A., & Budihartanti, C. (2017). Aplikasi Mobile Augmented Reality sebagai Media Pembelajaran Pengenalan Hardware Komputer Berbasis Android. *Jurnal PROSISKO*, 4(2), 17–24. <https://e-jurnal.lppmu.nsera.org/index.php/PROSISKO/article/view/387>