

**RANCANG BANGUN SISTEM PEMANTAUAN SUHU DAN KELEMBABAN GUDANG TEMBAKAU BERBASIS APLIKASI ANDROID**

**TUGAS AKHIR**

**Elyas Agrianto**

**18033228**

**PROGRAM STUDI TEKNIK TELEKOMUNIKASI**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI JAKARTA**

**2023**



**RANCANG BANGUN SISTEM PEMANTAUAN SUHU DAN KELEMBABAN GUDANG TEMBAKAU BERBASIS APLIKASI ANDROID**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma Tiga Politeknik**

**Elyas Agrianto**

**18033228**

**PROGRAM STUDI TEKNIK TELEKOMUNIKASI**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI JAKARTA**

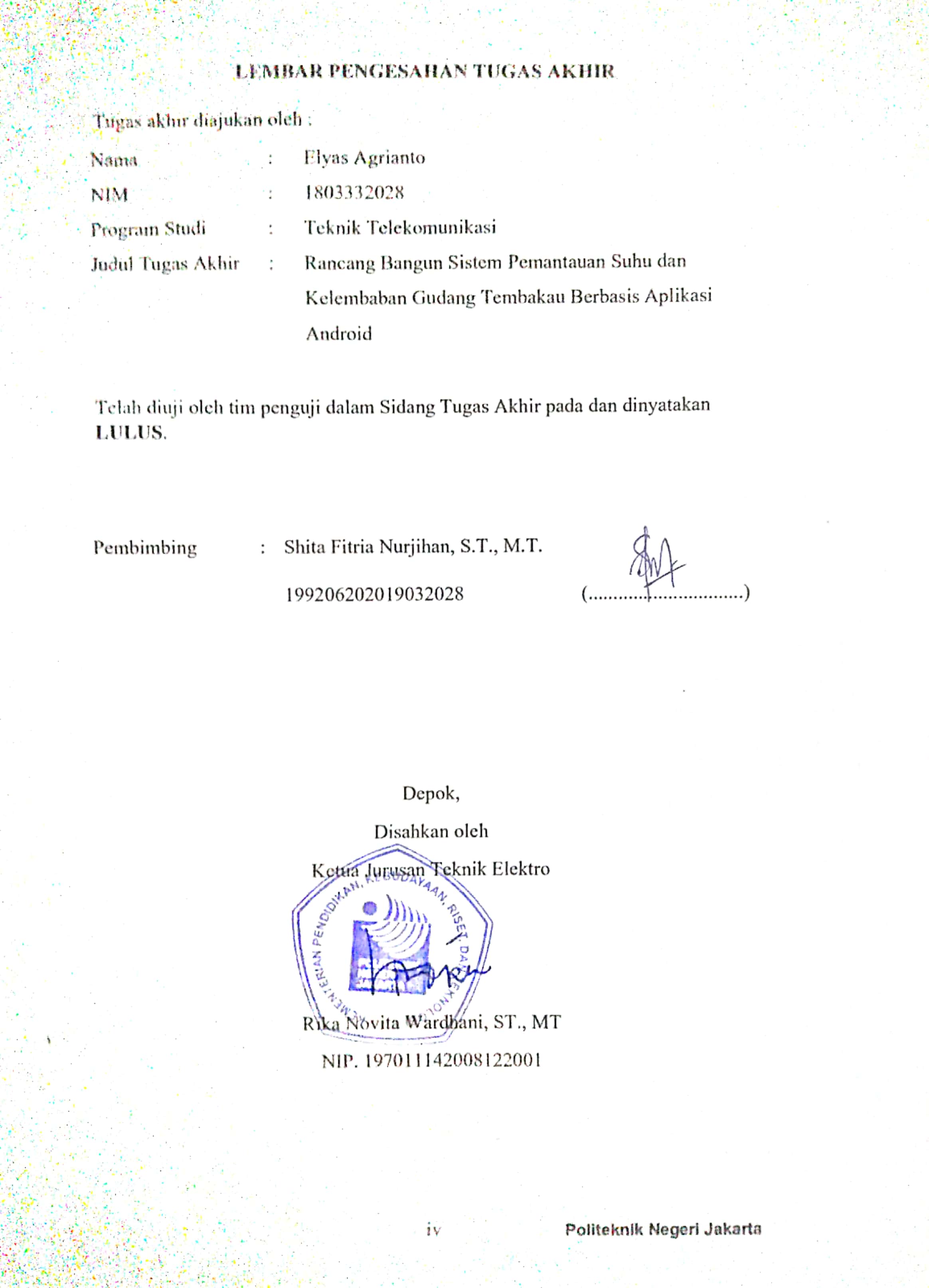
**2023**

# HALAMAN PERNYATAAN ORISINALITAS

**Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.**

|  |  |
| --- | --- |
| **Nama** | **: Elyas Agrianto** |
| **NIM** | **: 18033228** |
| **Tanda Tangan** | **:** |
| **Tanggal** | **:** |

# LEMBAR PENGESAHAN TUGAS AKHIR



# KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan laporan Tugas Akhir. Penulisan laporan dilakukan dalam rangka memnuhi salah satu syarat untuk mencapai gelar diploma Politeknik.

Penulisan laporan tugas akhir rancang bangun sistem pemantauan suhu dan kelembaban Gudang tembakau berbasis aplikasi android dapat diselesaikan. Oleh karena itu, penulis mengucapan terimakasih kepada:

1. Ibu Shita Fitria N, S.T., M.T. selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan kegiatan dan penyusunan laporan Praktik Kerja Lapangan ini;
2. Seluruh Staf Pengajar dan Karyawan Jurusan Teknik Elektro Politeknik Negeri Jakarta, khususnya Program Studi Telekomunikasi;
3. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
4. Sahabat yang telah banyak membantu penulis dalam menyelesaikan tugas akhir ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Tugas Akhir ini membawa manfaat untuk pengembangan ilmu.

Depok, Agustus 2023

Penulis

**RANCANG BANGUN SISTEM PEMANTAUAN SUHU DAN KELEMBABAN GUDANG TEMBAKAU BERBASIS APLIKASI ANDROID**

# ABSTRAK

*Sebelum tembakau diproses menjadi suatu produk diperlukan penyimpanan tembakau yang disebut juga gudang. Kualitas tembakau dapat terjaga selama kondisi suhu dan kelembaban ruang sesuai dengan kebutuhan tembakau. Maka diperlukan sistem yang dapat memantau suhu dan kelembaban Gudang tembakau. Kondisi suhu dalam Gudang akan diatur sebaik mungkin dalam range 27-32°C sedangkan kelembaban dalam range 50-70%RH. Pada tugas akhir kali ini dibuatnya sistem pemantauan suhu dan kelembaban gudang tembakau untuk mempermudah dalam menghidupkan kipas dan lampu dalam gudang. Sistem terintegrasi dengan mikrokontroler yang terhubung dengan Android Studio untuk mengetahui kondisi suhu dan kelembaban Gudang dan dapat menghidupkan atau mematikan lampu atau kipas berdasarkan suhu dan kelembaban gudang. Pengiriman data dari ESP32 yang terkoneksi internet ke Firebase yang selanjutnya akan diterima oleh aplikasi android.*

***Kata kunci:*** *Android, ESP, Firebase, Kelembaban, Suhu*

**DESIGN AND DEVELOPMENT OF TOBACCO WAREHOUSE TEMPERATURE AND HUMIDITY MONITORING SISTEM BASED ON ANDROID APPLICATION**

# ABSTRACT

*Before tobacco is processed into a product, it requires tobacco storage which is also called a warehouse. The quality of tobacco can be maintained as long as the temperature and humidity conditions of the room meet the needs of tobacco. So we need a sistem that can monitor the temperature and humidity of the tobacco warehouse. The temperature conditions in the Warehouse will be regulated as best as possible within the range of 27-32°C. In this final project, a temperature and humidity monitoring sistem for a tobacco warehouse is made. The sistem is integrated with a microcontroller that is connected to Android Studio to determine the temperature and humidity conditions in the Warehouse and the Firebase database to store data on the server. Sending data from the microcontroller to Firebase via ESP which will then be received by the Android application.*

***Keywords****: Android, ESP, Firebase, Humidity, Temperature*

# DAFTAR ISI

[HALAMAN PERNYATAAN ORISINALITAS iii](#_Toc143860930)

[LEMBAR PENGESAHAN TUGAS AKHIR iv](#_Toc143860931)

[KATA PENGANTAR v](#_Toc143860932)

[ABSTRAK vi](#_Toc143860933)

[ABSTRACT vii](#_Toc143860934)

[DAFTAR ISI viii](#_Toc143860935)

[DAFTAR GAMBAR xi](#_Toc143860936)

[DAFTAR TABEL xii](#_Toc143860937)

[DAFTAR LAMPIRAN xiv](#_Toc143860938)

[BAB I PENDAHULUAN 1](#_Toc143860939)

[1.1 Latar Belakang 1](#_Toc143860940)

[1.2 Perumusan Masalah 1](#_Toc143860941)

[1.3 Tujuan 2](#_Toc143860942)

[1.4 Luaran 2](#_Toc143860943)

[BAB II TINJAUAN PUSTAKA 3](#_Toc143860944)

[2.1 Gudang tembakau 3](#_Toc143860945)

[2.2 NodeMCU ESP32 3](#_Toc143860946)

[2.3 DHT11 4](#_Toc143860947)

[2.4 Relay modul 5](#_Toc143860948)

[2.5 Smartphone android 6](#_Toc143860949)

[2.6 Android Studio 6](#_Toc143860950)

[2.7 Firebase 7](#_Toc143860951)

[2.8 Power supply 8](#_Toc143860952)

[2.9 Parameter Kinerja LTE 9](#_Toc143860953)

[2.10 Quality of service (QoS) 10](#_Toc143860954)

[BAB III PERANCANGAN DAN REALISASI 13](#_Toc143860955)

[3.1 Rancangan alat 13](#_Toc143860956)

[3.1.1 Deskripsi alat 13](#_Toc143860957)

[3.1.2 Cara kerja alat 13](#_Toc143860958)

[3.1.3 Spesifikasi alat 17](#_Toc143860959)

[3.1.4 Diagram Blok 18](#_Toc143860960)

[3.1.5 Ilustrasi sistem 19](#_Toc143860961)

[3.2 Realisasi Alat 20](#_Toc143860962)

[3.2.1 Realisasi catu daya 20](#_Toc143860963)

[3.2.2 Realisasi Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau 21](#_Toc143860964)

[3.2.3 Pembuatan program ESP32 24](#_Toc143860965)

[3.2.4 Realisasi Real-Time Database Firebase 28](#_Toc143860966)

[3.2.5 Realisasi aplikasi android 32](#_Toc143860967)

[BAB IV PEMBAHASAN 50](#_Toc143860968)

[4.1 Pengujian Pengujian Catu daya 50](#_Toc143860969)

[4.1.1 Deskripsi Pengujian Catu daya 50](#_Toc143860970)

[4.1.2 Prosedur pengujian Catudaya 50](#_Toc143860971)

[4.1.3 Hasil Pengujian Catu Daya 51](#_Toc143860972)

[4.1.4 Analisa pengujian catu daya 51](#_Toc143860973)

[4.2 Tes Ping ESP32 51](#_Toc143860974)

[4.2.1 Deskripsi Tes Ping ESP32 51](#_Toc143860975)

[4.2.2 Prosedur Tes Ping ESP32 51](#_Toc143860976)

[4.2.3 Hasil Tes Ping ESP32 52](#_Toc143860977)

[4.2.4 Analisa Hasil Tes Ping ESP32 53](#_Toc143860978)

[4.3 Pengujian Kualitas DHT11 53](#_Toc143860979)

[4.3.1 Deskripsi Pengujian Kualitas DHT11 53](#_Toc143860980)

[4.3.2 Prosedur Pengujian Kualitas DHT11 53](#_Toc143860981)

[4.3.3 Hasil Pengujian Kualitas DHT11 53](#_Toc143860982)

[4.4 Pengujian Kualitas Relay 54](#_Toc143860983)

[4.4.1 Deskripsi Pengujian Kualitas Relay 54](#_Toc143860984)

[4.4.2 Prosedur Pengujian Kualitas Relay 54](#_Toc143860985)

[4.4.3 Hasil Pengujian Kualitas Relay 55](#_Toc143860986)

[4.5 Pengujian Sistem Keseluruhan 55](#_Toc143860987)

[4.5.1 Deskripsi Pengujian Sistem Keseluruhan 55](#_Toc143860988)

[4.5.2 Prosedur Pengujian Sistem Keseluruhan 55](#_Toc143860989)

[4.5.3 Hasil Pengujian Sistem Keseluruhan 56](#_Toc143860990)

[4.6 Pengujian Aplikasi Android 56](#_Toc143860991)

[4.6.1 Deskripsi Pengujian Aplikasi Android 56](#_Toc143860992)

[4.6.2 Prosedur Pengujian Aplikasi Android 56](#_Toc143860993)

[4.6.3 Hasil Pengujian Aplikasi Android 57](#_Toc143860994)

[4.7 Pengujian kecepatan internet, RSRQ dan RSRP 60](#_Toc143860995)

[4.7.1 Prosedur pengujian kecepatan internet, RSRP dan RSRQ 61](#_Toc143860996)

[4.7.2 Hasil pengujian kecepatan internet, RSRP dan RSRQ 61](#_Toc143860997)

[4.7.3 Analisa data hasil pengujian 62](#_Toc143860998)

[4.8 Pengujian QoS (*Quality of Service*) 63](#_Toc143860999)

[4.8.1 Prosedur pengujian QoS 63](#_Toc143861000)

[4.8.2 Data hasil pengujian QoS 64](#_Toc143861001)

[4.8.3 Analisa data pengujian QoS 65](#_Toc143861002)

[BAB V PENUTUP 66](#_Toc143861003)

[5.1 Simpulan 66](#_Toc143861004)

[5.2 Saran 66](#_Toc143861005)

[DAFTAR PUSTAKA 67](#_Toc143861006)

[DAFTAR RIWAYAT HIDUP 68](#_Toc143861007)

[LAMPIRAN 69](#_Toc143861008)

# DAFTAR GAMBAR

[Gambar 2.2 NodeMCU ESP 4](#_Toc143849610)

[Gambar 2.3 DHT11 5](#_Toc143849611)

[Gambar 2. 4 Modul relay 2-channel 5](#_Toc143849612)

[Gambar 2.5 *Smartphone* Android 6](#_Toc143849613)

[Gambar 2.6 Rangkaian power supply 9](#_Toc143849614)

[Gambar 3. 1 *Flowchart* sistem pemantauan berbasis aplikasi android 15](#_Toc143849590)

[Gambar 3. 2 *flowchart* sistem pemantauan kelembaban Gudang tembakau berbasis aplikasi android mode manual. 16](#_Toc143849591)

[Gambar 3. 3 Diagram Blok Sistem Pemantauan kondisi suhu dan kelembaban Gudang tembakau 19](#_Toc143849592)

[Gambar 3.4 Ilustrasi Sistem Pemantauan kondisi suhu dan kelembaban Gudang tembakau 19](#_Toc143849593)

[Gambar 3. 5 Desain skematik power supply 5V 20](#_Toc143849594)

[Gambar 3. 6 Rangkaian power supply 5V 21](#_Toc143849595)

[Gambar 3. 7 Skematik keseluruhan alat 22](#_Toc143849596)

[Gambar 3. 8 Skematik relay 2-channel 23](#_Toc143849597)

[Gambar 3. 9 Skematik DHT11 23](#_Toc143849598)

[Gambar 3. 10 Foto perangkat yang sudah terpasang 24](#_Toc143849599)

[Gambar 3. 11 *Flowchart* pembuatan realtime database Firebase 29](#_Toc143849600)

[Gambar 3. 12 Halaman Firebase Console 30](#_Toc143849601)

[Gambar 3. 13 Buat project firebase 30](#_Toc143849602)

[Gambar 3. 14 *rules realtime* database Firebase 31](#_Toc143849603)

[Gambar 3. 15 Tampilan real-time database firebase 31](#_Toc143849604)

[Gambar 3. 16 *flowchart* aplikasi android sistem pemantauan gudang 32](#_Toc143849605)

[Gambar 3. 17 Aplikasi terhubung ke Firebase 33](#_Toc143849606)

[Gambar 3. 18 tampilan *splashscreen* aplikasisistem pemantauan 35](#_Toc143849607)

[Gambar 3. 19 Tampilan *home page* aplikasi sistem pemantauan 37](#_Toc143849608)

[Gambar 3. 20 Tampilan *main activity* aplikasi sistem pemantauan 45](#_Toc143849609)

[Gambar 4. 1 Hasil Pengukuran Catu Daya 51](#_Toc143849577)

[Gambar 4. 2 Hasil Tes Ping ESP32 52](#_Toc143849578)

[Gambar 4. 3 Hasil Pengujian DHT11 54](#_Toc143849579)

[Gambar 4. 4 Hasil Tes Kualitas Relay 55](#_Toc143849580)

[Gambar 4. 5 Hasil Program Sistem Keseluruhan 56](#_Toc143849581)

[Gambar 4. 6 Hasil Pembuatan Firebase 57](#_Toc143849582)

[Gambar 4. 7 Halaman Aplikasi 58](#_Toc143849583)

[Gambar 4. 8 Mode Auto 59](#_Toc143849584)

[Gambar 4. 9 Mode Manual dalam Aplikasi 60](#_Toc143849585)

[Gambar 4. 10 Hasil pengujian Speedtest 61](#_Toc143849586)

[Gambar 4. 11 Hasil RSRP dan RSRQ menggunakan aplikasi Cell Tower 62](#_Toc143849587)

[Gambar 4. 12 Hasil pengukuran QoS 63](#_Toc143849588)

# DAFTAR TABEL

[Tabel 2. 1 *Range* Parameter RSRP 9](#_Toc143861019)

[Tabel 2. 3 Kategori *Throughtput* 11](#_Toc143861020)

[Tabel 2. 4 Kategori *Packet Loss* 11](#_Toc143861021)

[Tabel 2. 5 Kategori *Delay* 11](#_Toc143861022)

[Tabel 3. 1 Pin Sensor yang terhubung ke ESP32 21](#_Toc143861015)

[Tabel 4. 1 Hasil Pengujian DHT11 54](#_Toc143861016)

[Tabel 4. 1 Hasil pengujian Speedtest dan Cell Tower 62](#_Toc143861017)

[Tabel 4. 2 Hasil pengambilan data Qos dari Wireshark 64](#_Toc143861018)

# DAFTAR LAMPIRAN

Skematik Keseluruhan Alat…………………………………………………..L-1

Skematik Power Supply………………………………………………………L-2

Chasing Power Supply………………………………………………………..L-3

Desain Alat…………………………………………………………………...L-4

Sketch Program Alat………………………………………………………….L-5

Sketch aplikasi Android………………………………………………………L-6

Dokumentasi…...……………………………………………………………..L-7

# PENDAHULUAN

## Latar Belakang

Industri tembakau merupakan salah satu industri dengan pengaruh ekonomi yang besar di Indonesia. Namun dalam proses penyimpanannya harus memerhatikan kondisi suhu dan kelembaban Gudang penyimpanan tembakau guna tercapai tembakau yang berkualitas. Biasanya kondisi suhu pada penyimpanan tembakau berkisar dari 27-32oC dan kelembaban berkisar dari 50-70% RH.

Dalam proses pemantauan keadaan suhu dan kelembaban Gudang tembakau yang masih menggunakan termometer ruangan mengharuskan pekerjanya untuk mengecek setiap saat ke dalam Gudang. Namun tidak memungkinkan untuk pekerja dapat memantau gudang selama 24 jam. Maka dari itu supaya pekerja mudah dalam pemantauan dimana pun secara *real time*. Oleh karena itu pengusul membuat sistem pemantauan suhu dan kelembaban gudang tembakau menggunakan aplikasi android.

Maka penggunaan fungsi DHT11 dapat mengukur suhu dan kelembaban di dalam ruangan dengan bantuan NodeMCU ESP32 sebagai mikrokontrollernya. Adapun jika kondisi suhu dan kelembaban tidak sesuai dapat distabilkan oleh lampu ataupun kipas yang dapat menstabilkan kondisi Gudang dengan bantauan relay. Untuk lampu dan kipas dapat dinyalakan otomatis ataupun manual.

Jika ingin menyalakan lampu dan kipas secara otomatis cukup melalui aplikasi dan manual langsung pada tombol yang terdapat pada aplikasi android. Untuk itu maka pengusul membuat tugas akhir yang berjudul “Rancang Bangun Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau Berbasis Aplikasi Android”.

## Perumusan Masalah

Berdasarkan latar belakang diatas, maka permasalahan yang akan dibahas dalam tugas ini adalah sebagaimana berikut:

1. Bagaimana merancang *hardware* dan aplikasi alat Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau Berbasis Aplikasi Android ?
2. Bagaimana merealisasi alat sistem pemantauan suhu dan kelembaban gudang tembakau ?
3. Bagaimana menguji *hardware* dan aplikasi alat serta kecepatan dan kualitas layanan jaringan internet ?

## Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Melakukan perancangan alat sistem pemantauan suhu dan kelembaban Gudang tembakau berbasis aplikasi android.
2. Melakukan pembuatan dan realisasi alat sistem pemantauan suhu dan kelembaban Gudang tembakau berbasis aplikasi android.
3. Mampu menguji *hardware* dan aplikasi alat serta jaringan internet.

## Luaran

Luaran yang akan dicapai adalah:

1. Alat sistem pemantauan suhu dan kelembaban Gudang tembakau berbasis aplikasi android
2. Laporan Tugas Akhir prodi Teknik Telekomunikasi
3. Jurnal ilmiah kampus

# TINJAUAN PUSTAKA

## Gudang tembakau

Tembakau merupakan tanaman yang daunnya dimanfaatkan untuk bahan baku merokok. Daun tembakau terdapat kandungan nikotin yaitu zat adiktif dan juga sebagai bahan dasar sebagai bahan dasar untuk insektisida. Tembakau merupakan hasil tanaman *Nicotiana tobacum L*. dengan daun sebagai bagian yang dipanen.

Biasanya bahan baku tembakau disimpan didalam Gudang sebelum diolah menjadi rokok. Gudang tembakau memiliki kondisi suhu dan kelembaban yang harus dijaga supaya tembakau dapat terjaga kualitasnya. Suhu yang telah ditentukan rata-rata dalam *range* 27-32 oC dan untuk kelembaban Gudang sendiri dalam *range* 50-70%RH.

## NodeMCU ESP32

NodeMCU ESP32 berfungsi sebagai perangkat tambahan mikrokontroller seperti Arduino agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3,3V dengan memiliki tiga mode wifi yaitu station, access point dan *both* (keduanya). Modul ini dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung pada jenis ESP yang digunakan. Modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroller apapun karena sudah memiliki perlengkapan layaknya mikrokontroller. Menggunakan AT Command sebagai *firmware* *default* dan perangkat ini berbasis *open source*.



Gambar 2.1 NodeMCU ESP

Sumber: https://www.ardutech.com, 2022

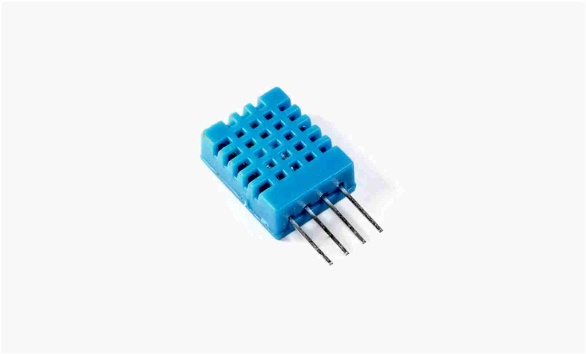
Spesifikasi nodeMCU ESP:

1. Jumlah pin : 32 meliputi pin tegangan dan GPIO.
2. 15 pin ADC (Analog to Digital Converter)
3. 3 UART Interface
4. 3 SPI Interface
5. 2 I2C Interface
6. 16 pin PWM (Pulse Width Modulation)
7. 2 pin DAC (Digital to Analog Converter

## DHT11

DHT11 adalah sebuah sensor yang mampu mengukur suhu dan kelembaban ruangan. DHT11 dapat deprogram dengan baik oleh Arduino ataupun ESP32.

1. Tegangan operasi 3.3-5V
2. Arus maks 2.5mA
3. *Range* pengukuran kelembaban 20-80% dengan akurasi 5%
4. *Range* pengukuran suhu 0°-50°C dengan akurasi 2°C
5. Kecepatan pengambilan *sample* tidak lebih dari 1Hz per detik

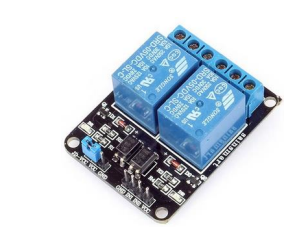


Gambar 2.2 DHT11

Sumber: www.mahirelektro.com, 2021

## Relay modul

Relay modul merupakan salah satu piranti yang beroperasi berdasarkan prinsip elektromagnetik unntuk menggerakkan kontaktor guna memindahkan posisi ON ke OFF atau sebaliknya memanfaatkan tenaga listrik. Berbeda dengan saklar atau *switch* biasa yang menggunakan cara manual. Fungsi *relay* sebagai saklar otomatis, dimana ia dapat bekerja otomatis berdasarkan perintah logika yang diberikan. Kebanyakan relay modul menggunakan tegangan 5VDC untuk bekerja. Jenis-jenis *relay* beranekaragam seperti *single*-*channel*, 2-*channel* dan banyak lainnya.



Gambar 2. 3 Modul relay 2-channel

Sumber: www.ditempel.com, 2021

Berdasarkan gambar diatas *relay module* memiliki pin output sebagai berikut:

1. COM (*Common*)

Pin yang wajib dihubungkan dari salah satu ujung kabel suatu komponen.

1. NO (*Normally Open)*

Pin yang menguhubungkan kabel yang satunya bila menginginkan kondisi awal yang *open* atau terbuka karena kondisi arus listrik terputus.

1. NC (*Normally Close*)

Pin yang menghubungkan kabel yang satunya bila menginginkan kondisi awal *close* atau tertutup karena kondisi arus listrik tersambung.

## Smartphone android

Pengertian Android adalah sistem operasi yang dirancang oleh *Google* dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau smartphone. Jadi, android digunakan dengan sentuhan, gesekan ataupun ketukan pada layar *gadget*.

Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi alias gratis.

Begitupun dengan para pembuat aplikasi, mereka bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan *Google*. Dengan seperti itu android memiliki jutaan support aplikasi gratis/berbayar yang dapat diunduh melalui *Google Play* (Prawiro, 2018). Gambar 2.2 merupakan bentuk fisik *Smartphone* Android.



Gambar 2.4 *Smartphone* Android

Sumber: android.com, 2023

## Android Studio

Android studio adalah *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA . Selain sebagai editor kode dan fitur developer IntelliJ yang andal (Catur Wibowo, Dimas, 2019). Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat 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. Terapkan perubahan untuk melakukan push pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
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*.

## Firebase

Firebase adalah suatu layanan dari *Google* untuk memberikan kemudahan bahkan mempermudah para developer aplikasi dalam mengembangkan aplikasinya. Firebase alias *Backend as a Service* (BaaS) merupakan solusi yang ditawarkan oleh *Google* untuk mempercepat pekerjaan developer (Intern, Dicoding. 2020). Dengan menggunakan Firebase*,* apps developer bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend.*

Singkat cerita mengenai sejarah dari Firebase didirikan pertama kali pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk Firebase yang pertama kali adalah *Realtime Database*. *Realtime* *Database* digunakan developer untuk menyimpan data dan s*ynchronize* ke banyak *user*. Kemudian ia berkembang sebagai layanan pengembang aplikasi. Pada bulan Oktober 2014, perusahaan tersebut diakuisisi oleh Google. Firebase *Realtime Database*

Firebase *Realtime Database* adalah *database* yang di-host melalui cloud. Data disimpan dan dieksekusi dalam bentuk JSON dan disinkronkan secara realtime ke setiap user yang terkoneksi. Hal ini berfungsi memudahkan kamu dalam mengelola suatu *database* dengan skala yang cukup besar. Ketika membuat aplikasi lintas-platform/multiplatform menggunakan SDK Android, iOS, dan juga JS (JavaScript), semua pengguna akan berbagi sebuah *instance Realtime Database* dan menerima *update* data secara serentak dan otomatis.

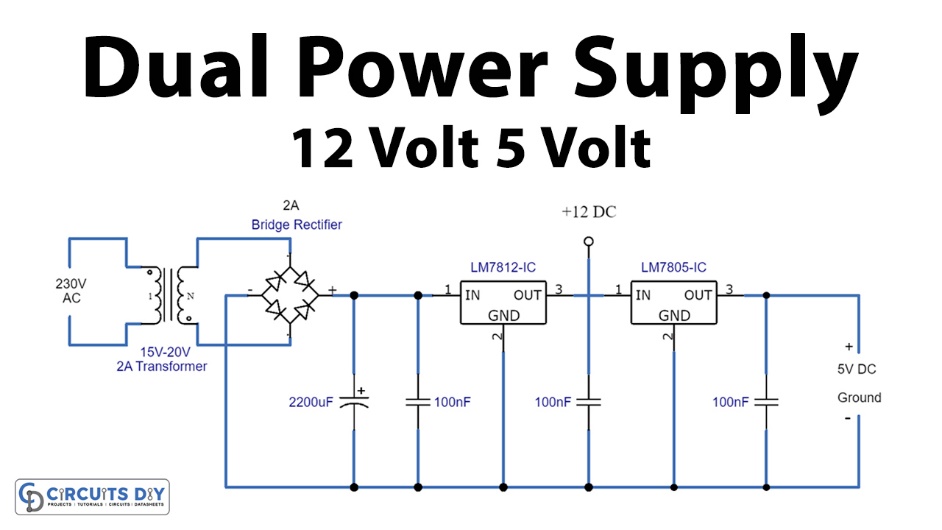
Kemampuan lain dari Firebase *Realtime Database* adalah tetap responsif bahkan saat *offline* karena SDK *Firebase Realtime Database* menyimpan data langsung ke *disk device* atau memori lokal. Setelah perangkat terhubung kembali dengan internet, perangkat pengguna (*user*) akan menerima setiap perubahan yang terjadi.

## Power supply

Power supply adalah suatu rangkaian elektronika yang berfungsi sebagai penyedia sumber energi listrik untuk perangkat elektronika, dalam hal ini energi listrik tegangan DC. Komponen yang terdapat pada rangkaian power supply adalah transformator, penyearah, filter, dan regulator. Transformator berfungsi untuk menurunkan tegangan listrik sehingga dapat dikelola oleh IC regulator. Diode bridge berfungsi sebagai penyearah tegangan AC menjadi tegangan DC.

Kemudian filter berfungsi menyaring tegangan *ripple* yang masih bocor sehingga arus yang dikeluarkan dari penyearah menjadi rata. Filter yang biasa digunakan ialah kapasitor ELCO. Regulator berperan untuk menstabilkan tegangan yang keluar dari filter tadi. Biasanya regulator yang dapat digunakan adalah IC regulator 78xx atau transistor.

Apabila menginginkan tegangan output power supply yang bervariasi seperti 5VDC, 9VDC, dan 12VDC dapat ditentukan dari output dengan saklar transformator. Dapat dilihat pada gambar 2.3 rangkaian power supply tersebut.



Gambar 2.5 Rangkaian power supply

Sumber: www.circuits-diy.com, 2022.

## Parameter Kinerja LTE

*Long Term Evolution* (LTE) merupakan suatu layanan dengan kemampuan tinggi, yang diperkenalkan oleh (3GPP) *The Third Generation Partnership Project*. LTE berfungsi sebagai suatu perangkat tambahan pada jaringan *Universal* *Mobile Telecommunications Sistem*. Disamping itu keunggulan secara umum *Long Term Evolution* dapat memberikan kecepatan akses maksimal hingga 50 Mbps saat melakukan *uplink*. Sedangkan saat melakukan *downlin*k kecepatan akses maksimal yang diberikan hingga 100 Mbps (Paramartha, 2019).

1. Parameter *Reference Signal Received Power* (RSRP)

RSRP merupakan parameter kuat signal dari jaringan LTE yang diterima oleh *user equipment*. Parameter ini berfungsi menentukan titik-titik saat terjadi *handove*r, dan mengetahui luas jangkauan dari sektor antena pada suatu eNodeB. *Range* parameter RSRP ditunjukkan pada Tabel 2.1.

Tabel 2. 1 *Range* Parameter RSRP

|  |  |  |
| --- | --- | --- |
| Warna | Nilai RSRP (dBm) | keterangan |
|  | < -60 | Luar biasa |
|  | -60 s/d -70 | Sangat baik |
|  | -70 s/d -80 | Baik |
|  | -80 s/d -90 | Normal |
|  | -90 s/d -110 | Buruk |
|  | -110 s/d -120 | Sangat Buruk |

Sumber: Paramartha, 2019

1. Parameter *Reference Signal Received Quality* (RSRQ)

RSRQ merupakan kualitas sinyal yang membantu parameter RSRP saat terjadi *handover*. Selain itu parameter RSRQ di definisikan sebagai rasio antara jumlah *resource block* terhadap rata-rata daya *linier* yang terima oleh *user* termasuk daya dari *serving cell*, *noise*, dan interferensi. *Range* parameter RSRP ditunjukkan pada gambar 2.2.



Gambar 2. 6 *Range* Parameter RSRP

Sumber: Paramartha, 2019

## Quality of service (QoS)

*Quality of service* (QoS) merupakan metode pengukuran tentang seberapa baik jaringan dan merupakan suatu usaha untuk mendefenisikan karakteristik dan sifat dari satu layanan. Parameter-parameter QoS adalah sebagai berikut (Muchlisin Riadi, 2019) :

1. *Throughput*

*Throughput* adalah kecepatan (*rate*) transfer data efektif yang diukur dalam bit *per* *second* (bps). Formula untuk menghitung *throughput* adalah sebagai berikut (ETSI, 1999) :

|  |  |
| --- | --- |
|  | (2.1) |

Untuk menentukan besar  *throughput* yang diukur dapat menyesuaikan data hasil pengukuran *throughput* yaitu perbandingan jumlah data yang dikirim dengan waktu pengiriman data dengan kategori pada Tabel 2.3. Dalam kategori tersebut indeks paling bagus diatas 2,1Mbps dan terburuk 0-338kbps.

Tabel 2. 3 Kategori *Throughtput*

|  |  |  |
| --- | --- | --- |
| Kategori *Throughput* | *Throughput* (kbps) | Indeks |
| *Bad*  *Poor*  *Fair*  *Good*  *Excelent* | 0 – 338  338 – 700  700 – 1.200  1.200 – 2,1 Mbps  > 2,1 Mbps | 0  1  2  3  4 |

Sumber: ETSI, 1999

1. *Packet Loss*

*Packet loss* merupakan parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan. Kategori *packet loss* ditampilkan pada Tabel 2.2.

Tabel 2. 4 Kategori *Packet Loss*

|  |  |  |
| --- | --- | --- |
| Kategori Degredasi | *Packet Loss* (%) | Indeks |
| Sangat Bagus | 0 | 4 |
| Bagus | 3 | 3 |
| Sedang | 15 | 2 |
| Jelek | 25 | 1 |

Sumber: ETSI, 1999

Untuk menentukan suatu jaringan sudah baik atau belum dapat menyesuaikan data *packet loss* yang diukur dengan kategori pada Tabel 2.2. Formula untuk menghitung *packet loss* adalah sebagai berikut (ETSI, 1999) :

|  |  |
| --- | --- |
|  | (2.2) |
|  |  |

1. *Delay (Latency)*

Merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, *congestion* atau waktu proses yang lama. Kategori *delay* ditampilkan pada Tabel 2.3.

Tabel 2. 5 Kategori *Delay*

|  |  |  |
| --- | --- | --- |
| Kategori Lantensi | Besar *Delay* (ms) | Indeks |
| Sangat Bagus | < 150 ms | 4 |
| Bagus | 150 ms s/d 320 ms | 3 |
| Sedang | 320 ms s/d 450 ms | 2 |
| Jelek | > 450 ms | 1 |

Sumber: ETSI, 1999

Untuk menentukan suatu jaringan sudah baik atau belum bisa dengan menyesuaikan data *delay* yang diukur dengan kategori yang ada pada Tabel 2.3. Formula untuk menghitung *delay* adalah sebagai berikut (ETSI, 1999) :

|  |  |
| --- | --- |
|  | (2.3) |

# PENUTUP

## Simpulan

Simpulan yang dapat dibuat dari hasil pembuatan tugas akhir “Rancang Bangun Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau Berbasis Aplikasi Android” adalah sebagai berikut:

1. Dalam perancangan catudaya dibuat dahulu skematiknya yang kemudian akan dibuat *layout* PCB dan *casing* catu daya itu sendiri. Kemudian dalam perancangan sistem alat dibuat dahulu skematik untuk hubungan pin perangkat yang digunakan dengan mikrokontroller. Terakhir perancangan aplikasi dibuat menggunakan aplikasi android.
2. Setelah perancangan, dilakukan pembuatan catudaya dengan output tegangan 4,95V. Program untuk sistem keseluruhan dibuat agar mode dalam keadaaan auto, suhu dan kelembaban sebagai *trigger* relay. Sedangkan mode manual menggunakan *button* aplikasi sebagai *trigger* relay. Kemudian pembuatan alat dilakukan dengan hubungkan pin antara *hardware* yang digunakan.
3. Hasil pembuatan alat adalah sistem bekerja sesuai program alat dan perangkat dapat bekerja dengan maksimal. Lalu aplikasi yang telah dibuat dapat menampilkan data dari mikrokontroller dan menghidupkan lampu serta kipas. Hasil saat keadaan auto suhu 33,8°C dan kelembaban 49%RH sehingga kipas hidup ditunjukkan pada status kipas yang ON. Hasil pada pengujian kecepatan internet menggunakan *provider* IM3 untuk *download* adalah 12,4 Mbps dan *upload* adalah 10,9 Mbps. Kemudian setelah pengujian kualitas layanan internet (QoS) didapatkan *delay* yang kecil yaitu *, throughput*serta *packet loss* .

## Saran

Saran yang dapat diberikan dari alat yang sudah dibuat adalah dalam penggunaan jenis tipe data untuk komunikasi data antara firebase dan aplikasi android harus menggunakan jenis yang sama, karena jika tidak dapat memungkinkan aplikasi akan *error* saat *running* aplikasi.

# DAFTAR PUSTAKA

Admin. (26 Mei 2021). “Cara Kerja Modul Relay Untuk Penggunaan Aplikasi Arduino”. https://www.ditempel.com/2021/05/cara-kerja-modul-relay-untuk-penggunaan.html [10 Mei 2023]

M. Habib Al Khairi. (17 April 2021). “Cara mengukur suhu dan kelembaban dengan DHT11 dan arduino”. *https://www.mahirelektro.com/2020/02/tutorial-menggunakan-sensor-DHT11-pada-Arduino.html/* [8 Mei 2023]

Catur Wibowo, Dimas. (2019).”Apa itu Android Studio dan Android SDK?”. *https://www.dicoding.com*/ [14 Mei 2023]

Intern, Dicoding. (2020).”Apa itu *Firebase*? Pengertian, Jenis-Jenis, dan Fungsi Kegunaannya”. *https://www.dicoding.com/* [14 Mei 2023]

# LAMPIRAN

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **01** | **SKEMATIK KESELURUHAN SISTEM** | | |
| ***PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA*** | | *Digambar* | *Elyas Agrianto* |
| *Diperiksa* | *Shita Fitria N, S.T., MT.* |
| *Tanggal* | *.......... Juli 2023* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **03** | **SKEMATIK POWER SUPPLY** | | |
| ***PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA*** | | *Digambar* | *Elyas Agrianto* |
| *Diperiksa* | *Shita Fitria N, S.T., MT.* |
| *Tanggal* | *.......... Juli 2023* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **02** | **TAMPILAN CHASING POWER SUPPLY** | | |
| ***PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA*** | | *Digambar* | *Elyas Agrianto* |
| *Diperiksa* | *Shita Fitria N, S.T., MT.* |
| *Tanggal* | *.......... Juli 2023* |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **04** | **DESAIN ALAT** | | |
| ***PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA*** | | *Digambar* | *Elyas Agrianto* |
| *Diperiksa* | *Shita Fitria N, S.T., MT.* |
| *Tanggal* | *.......... Juli 2023* |

Sketch Program ESP32

#include <WiFi.h>

#include <FirebaseESP32.h>

#include <DHT.h>

// Replace with your network credentials

const char\* WIFI\_SSID = "el diablo";

const char\* WIFI\_PASSWORD = "janganlupamandi";

// Replace with your Firebase project credentials

#define FIREBASE\_HOST "https://test-project-ta-56880-default-rtdb.firebaseio.com"

#define FIREBASE\_AUTH "sT11kfFrECRq4jbHauMJxWIgBNzXyowWP1hCWxNB"

#define DHT\_PIN1 13    // Pin untuk sensor DHT11 pertama

#define DHT\_PIN2 22    // Pin untuk sensor DHT11 kedua

#define RelayLampu 4   // Pin untuk relay channel 1

#define RelayKipas 15  // Pin untuk relay channel 2

DHT dht1(DHT\_PIN1, DHT11);

DHT dht2(DHT\_PIN2, DHT11);

FirebaseData fbdo;

FirebaseAuth auth;

FirebaseConfig config;

void setup() {

  Serial.begin(115200);

  dht1.begin();

  dht2.begin();

  pinMode(RelayLampu, OUTPUT);

  pinMode(RelayKipas, OUTPUT);

  digitalWrite(RelayLampu, HIGH);

  digitalWrite(RelayKipas, HIGH);

  WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD);

  Serial.print("Connecting to Wi-Fi...");

  while (WiFi.status() != WL\_CONNECTED) {

    Serial.print(".");

    delay(500);

  }

  Serial.println("Connected to Wi-Fi!");

  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);

}

void loop() {

  // Baca suhu dan kelembaban dari sensor DHT11 pertama dan kedua

  float suhu = dht1.readTemperature();

  float kelembaban = dht2.readHumidity();

  Serial.print("Temperature: ");

  Serial.print(suhu);

  Serial.print("\*C");

  Serial.print("Humidity: ");

  Serial.print(kelembaban);

  Serial.println("%RH");

  delay(1000);

  // Baca data trigger button dari Firebase

  Firebase.getString(fbdo, "Mode");

  String mode = fbdo.stringData();

  Firebase.getString(fbdo, "button\_lampu");

  String button\_lampu = fbdo.stringData();

  Firebase.getString(fbdo, "button\_kipas");

  String button\_kipas = fbdo.stringData();

  // Jika trigger button aktif, atur relay berdasarkan data dari sensor DHT11

   if (mode == "AUTO") {

    Serial.println("MODE:" + mode);

    if (suhu > 32) {

      digitalWrite(RelayKipas, LOW);

      Serial.println("KIPAS ON");

      if (kelembaban > 70) {

        digitalWrite(RelayLampu, LOW);

        Serial.println("LAMPU ON");

      } else {

        digitalWrite(RelayLampu, HIGH);

        Serial.println("LAMPU OFF");

      }

    } else if (suhu < 27) {

      digitalWrite(RelayLampu, LOW);

      Serial.println("LAMPU ON");

      if (kelembaban < 50) {

        digitalWrite(RelayKipas, LOW);

        Serial.println("KIPAS ON");

      } else {

        digitalWrite(RelayKipas, HIGH);

        Serial.println("KIPAS OFF");

      }

    } else {

      digitalWrite(RelayLampu, HIGH);

      digitalWrite(RelayKipas, HIGH);

    }

  } else if (mode == "MANUAL") {

    Serial.println("MODE:" + mode);

    if (button\_lampu == "ON") {

      Serial.println("lampu ON");

      digitalWrite(RelayLampu, LOW);

    } else {

      Serial.println("lampu OFF");

      digitalWrite(RelayLampu, HIGH);

    }

    if (button\_kipas == "ON") {

      Serial.println("kipas ON");

      digitalWrite(RelayKipas, LOW);

    } else {

      Serial.println("kipas OFF");

      digitalWrite(RelayKipas, HIGH);

    }

  } else {

    digitalWrite(RelayKipas, HIGH);

    digitalWrite(RelayLampu, HIGH);

    Serial.println("No Data Found");

  }

  int statuslampu = digitalRead(RelayLampu);

  int statuskipas = digitalRead(RelayKipas);

  statuslampu = !statuslampu;

  statuskipas = !statuskipas;

  Firebase.setFloat(fbdo, "suhu", suhu);

  Firebase.setFloat(fbdo, "kelembaban", kelembaban);

  Firebase.setInt(fbdo, "kipas", statuskipas);

  Firebase.setInt(fbdo, "lampu", statuslampu);

}

Activity\_splash\_screen.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_vertical"  
 android:background="@color/black"  
 tools:context=".SplashScreen"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau"  
 android:textSize="32dp"  
 android:textAlignment="center"  
 android:textColor="#FFA500"**/>  
 </**LinearLayout**>  
 </**RelativeLayout**>  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="220dp"  
 android:gravity="center\_vertical"**>  
  
 <**ImageView  
 android:layout\_width="200dp"  
 android:layout\_height="120dp"  
 android:src="@mipmap/ic\_homepage"** />  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="480dp"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="WELCOME"  
 android:textSize="40dp"  
 android:textColor="#FFA500"**/>  
 </**LinearLayout**>  
 </**RelativeLayout**>  
</**RelativeLayout**>

SplashScreen.java

**package** com.example.pemantauansuhukelembabangudang;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.os.Handler;  
**import** android.view.View;  
**import** android.view.Window;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.os.Bundle;  
  
**public class** SplashScreen **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splash\_screen***);  
 **int** Splash\_Time = 3200; *// waktu menampilkan splashscreen 3 detik* **new** Handler().postDelayed(**new** Runnable() {  
 @Override  
 **public void** run() {  
 Intent intent = **new** Intent(SplashScreen.**this**, HomeActivity.**class**);  
 SplashScreen.**this**.startActivity(intent);  
 SplashScreen.**this**.finish();  
 }  
 }, Splash\_Time);  
 }  
}

Activity\_home.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_vertical"  
 android:background="@color/black"  
 tools:context=".HomeActivity"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Sistem Pemantauan Suhu dan Kelembaban Gudang Tembakau"  
 android:textSize="32dp"  
 android:textAlignment="center"  
 android:textColor="#FFA500"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="220dp"  
 android:gravity="center\_vertical"**>  
  
 <**ImageView  
 android:layout\_width="200dp"  
 android:layout\_height="120dp"  
 android:src="@mipmap/ic\_homepage"** />  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="500dp"  
 android:gravity="center\_horizontal"  
 android:id="@+id/nextButton"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**Button  
 android:id="@+id/next"  
 android:layout\_width="150dp"  
 android:layout\_height="50dp"  
 android:background="#FFA500"  
 android:text="MASUK"  
 android:textSize="20dp"** />  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
</**RelativeLayout**>

HomeActivity.java

**package** com.example.pemantauansuhukelembabangudang;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.os.Bundle;  
**import** android.widget.Button;  
  
**public class** HomeActivity **extends** AppCompatActivity {  
  
 Button **next**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_home***);  
  
 **next** = (Button) findViewById(R.id.***next***);  
 **next**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent nextMain = **new** Intent(getApplicationContext(), MainActivity.**class**);  
 startActivity(nextMain);  
 }  
 });  
 }  
  
  
}

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center\_vertical"  
 android:background="#FFA500"  
 tools:context=".MainActivity"**>  
  
 <**RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**>  
  
 <**TextView  
 android:id="@+id/text\_view\_date"  
 android:layout\_width="250dp"  
 android:layout\_height="32dp"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 android:text="Date"  
 android:textSize="20dp"**/>  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/mode"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="70dp"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="70dp"  
 android:layout\_height="50dp"  
 android:text="Mode"  
 android:textColor="@color/black"  
 android:textStyle="bold"  
 android:gravity="center\_vertical"  
 android:textSize="20dp"**/>  
 <**Button  
 android:id="@+id/ManualMode"  
 android:layout\_width="90dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/black"  
 android:background="@color/white"  
 android:text="MANUAL"**/>  
  
 <**Button  
 android:id="@+id/AutoMode"  
 android:layout\_width="90dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/white"  
 android:background="@color/black"  
 android:text="AUTO"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
 <**RelativeLayout  
 android:id="@+id/temperature"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/mode"  
 android:layout\_marginTop="32dp"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="70dp"  
 android:src="@mipmap/ic\_temperature"**/>  
  
 <**TextView  
 android:id="@+id/TempValue"  
 android:layout\_width="70dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textStyle="bold"  
 android:textSize="20dp"  
 android:textColor="@color/black"  
 android:gravity="center\_vertical"  
 android:text="0"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/humidity"  
 android:layout\_marginTop="32dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/temperature"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="70dp"  
 android:src="@mipmap/ic\_humidity"**/>  
  
 <**TextView  
 android:id="@+id/HumValue"  
 android:layout\_width="90dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textStyle="bold"  
 android:textSize="20dp"  
 android:textColor="@color/black"  
 android:gravity="center\_vertical"  
 android:text="0"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/kipas"  
 android:layout\_marginTop="40dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/humidity"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="@string/Kipas"  
 android:textSize="20dp"**/>  
 <**Button  
 android:id="@+id/OnKipas"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/black"  
 android:background="@color/lime"  
 android:textStyle="bold"  
 android:text="ON"**/>  
 <**Button  
 android:id="@+id/OffKipas"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/black"  
 android:background="@color/red"  
 android:textStyle="bold"  
 android:text="OFF"**/>  
 </**LinearLayout**>  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/lampu"  
 android:layout\_marginTop="32dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/kipas"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="@string/Lampu"  
 android:textSize="20dp"**/>  
 <**Button  
 android:id="@+id/OnLampu"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/black"  
 android:background="@color/lime"  
 android:textStyle="bold"  
 android:text="ON"**/>  
 <**Button  
 android:id="@+id/OffLampu"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="10dp"  
 android:textColor="@color/black"  
 android:background="@color/red"  
 android:textStyle="bold"  
 android:text="OFF"**/>  
 </**LinearLayout**>  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/Status"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:layout\_below="@+id/lampu"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="@string/Kipas"  
 android:textSize="20dp"**/>  
  
 <**TextView  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="32dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="@string/Lampu"  
 android:textSize="20dp"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>  
  
 <**RelativeLayout  
 android:id="@+id/Status2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:layout\_below="@+id/Status"  
 android:gravity="center\_horizontal"**>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_vertical"**>  
  
 <**TextView  
 android:id="@+id/statusKipas"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="ON"  
 android:background="@color/purple\_200"  
 android:textSize="20dp"**/>  
  
 <**TextView  
 android:id="@+id/statusLampu"  
 android:layout\_width="80dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="32dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="ON"  
 android:background="@color/purple\_200"  
 android:textSize="20dp"**/>

<**TextView  
 android:id="@+id/statusMode"  
 android:layout\_width="125dp"  
 android:layout\_height="50dp"  
 android:layout\_marginLeft="32dp"  
 android:gravity="center"  
 android:textStyle="bold"  
 android:textColor="@color/black"  
 android:text="MANUAL"  
 android:background="@color/teal\_200"  
 android:textSize="20dp"**/>

</**LinearLayout**>  
  
 </**RelativeLayout**>  
</**RelativeLayout**>

MainActivity.java

**package** com.example.pemantauansuhukelembabangudang;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
  
**import** java.text.DateFormat;  
**import** java.text.SimpleDateFormat;  
**import** java.util.Calendar;  
**import** java.util.Locale;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 Button **lampuOn**;  
 Button **lampuOff**;  
 Button **kipasOn**;  
 Button **kipasOff**;  
 Button **modeAuto**;  
 Button **modeManual**;  
 TextView **nilaiTemp**;  
 TextView **nilaiHum**;  
 TextView **stLampu**;  
 TextView **stKipas**;  
 TextView **stMode**;  
  
 **private** TextView **dateTextView**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **stKipas** = (TextView) findViewById(R.id.***statusKipas***);  
 **stLampu** = (TextView) findViewById(R.id.***statusLampu***);  
 **stMode** = (TextView) findViewById(R.id.***statusMode***);  
 **nilaiTemp** = (TextView) findViewById(R.id.***TempValue***);  
 **nilaiHum** = (TextView) findViewById(R.id.***HumValue***);  
 **lampuOn** = (Button) findViewById(R.id.***OnLampu***);  
 **lampuOff** = (Button) findViewById(R.id.***OffLampu***);  
 **kipasOn** = (Button) findViewById(R.id.***OnKipas***);  
 **kipasOff** = (Button) findViewById(R.id.***OffKipas***);  
 **modeAuto** = (Button) findViewById(R.id.***AutoMode***);  
 **modeManual** = (Button) findViewById(R.id.***ManualMode***);  
  
 *//membaca waktu* Calendar calendar = Calendar.*getInstance*();  
 SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"EEEE, MMMM dd, yyyy - HH:mm:ss"**, Locale.*getDefault*());  
 String currentDate = DateFormat.*getDateInstance*(DateFormat.***FULL***).format(calendar.getTime());  
 TextView textViewDate = findViewById(R.id.***text\_view\_date***);  
 textViewDate.setText(currentDate);  
  
 *//membaca nilai suhu* FirebaseDatabase databasenilaiTemp = FirebaseDatabase.*getInstance*();  
 DatabaseReference nilaiTemperature = databasenilaiTemp.getReference(**"suhu"**);  
 nilaiTemperature.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **float** Suhu = snapshot.getValue(**float**.**class**);  
 **nilaiTemp**.setText(Suhu+**"°C"**);  
  
 }  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 *//membaca nilai kelembaban* FirebaseDatabase databasenilaiHum = FirebaseDatabase.*getInstance*();  
 DatabaseReference nilaiHumidity = databasenilaiHum.getReference(**"kelembaban"**);  
 nilaiHumidity.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **float** Kelembaban = snapshot.getValue(**float**.**class**);  
 **nilaiHum**.setText(Kelembaban+**"%RH"**);  
 }  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 *//Mengaktifkan mode manual* **modeManual**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseModeManual = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseModeManual.getReference(**"Mode"**);  
 myRef.setValue(**"MANUAL"**);  
 **stMode**.setText(**"MANUAL"**);  
 }  
 });  
 *//Mengaktifkan mode otomatis* **modeAuto**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseModeAuto = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseModeAuto.getReference(**"Mode"**);  
 myRef.setValue(**"AUTO"**);  
 **stMode**.setText(**"AUTO"**);  
 }  
 });  
 *//Menyalakan kipas* **kipasOn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseAktifkanKipas = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseAktifkanKipas.getReference(**"button\_kipas"**);  
 myRef.setValue(**"ON"**);  
 }  
 });  
 *//Mematikan kipas* **kipasOff**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseAktifkanKipas = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseAktifkanKipas.getReference(**"button\_kipas"**);  
 myRef.setValue(**"OFF"**);  
 }  
 });  
 *//Status kipas* FirebaseDatabase statusKipas = FirebaseDatabase.*getInstance*();  
 DatabaseReference statusRelayKipas = statusKipas.getReference(**"kipas"**);  
 statusRelayKipas.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **int** statusKipas = snapshot.getValue(Integer.**class**);  
 **if** (Integer.*valueOf*(statusKipas)==1) {  
 **stKipas**.setText(**"ON"**);  
 }  
 **else** {  
 **stKipas**.setText(**"OFF"**);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 *//Menyalakan lampu* **lampuOn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseAktifkanLampu = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseAktifkanLampu.getReference(**"button\_lampu"**);  
 myRef.setValue(**"ON"**);  
 }  
 });  
 *//Mematikan lampu* **lampuOff**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 FirebaseDatabase databaseAktifkanLampu = FirebaseDatabase.*getInstance*();  
 DatabaseReference myRef = databaseAktifkanLampu.getReference(**"button\_lampu"**);  
 myRef.setValue(**"OFF"**);  
 }  
 });  
 *//Status lampu* FirebaseDatabase statusLampu = FirebaseDatabase.*getInstance*();  
 DatabaseReference statusRelayLampu = statusLampu.getReference(**"lampu"**);  
 statusRelayLampu.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot snapshot) {  
 **int** statusLampu = snapshot.getValue(Integer.**class**);  
 **if** (Integer.*valueOf*(statusLampu)==1) {  
 **stLampu**.setText(**"ON"**);  
 }  
 **else** {  
 **stLampu**.setText(**"OFF"**);  
 }  
 }  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError error) {  
  
 }  
 });  
 }  
}

AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
   
 package="com.example.pemantauansuhukelembabangudang"**>  
 <**uses-permission android:name="android.permission.INTERNET"**/>  
 <**application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_icon"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.PemantauanSuhuKelembabanGudang"  
 tools:targetApi="31"**>  
 <**activity  
 android:name=".SplashScreen"  
 android:exported="true"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity  
 android:name=".HomeActivity"  
 android:exported="true"**>  
 </**activity**>  
 <**activity  
 android:name=".MainActivity"  
 android:exported="true"**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

Dokumentasi

