# MEKANISME PENYEDIAAN LAYANAN PADA PERANGKAT *SMART HOME* BERBASIS ESP32 BERDASARKAN DATASET *TIME SERIES* MENGGUNAKAN K-MEANS

## SKRIPSI

Untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik

Disusun oleh: Ghifari Adlil Baqi

NIM: 195150301111004



## PROGRAM STUDI TEKNIK KOMPUTER DEPARTEMEN TEKNIK INFORMATIKA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA

**MALANG 2023**

# PENGESAHAN

MEKANISME PENYEDIAAN LAYANAN PADA PERANGKAT SMART HOME BERBASIS ESP32 BERDASARKAN DATASET TIME SERIES MENGGUNAKAN K-MEANS

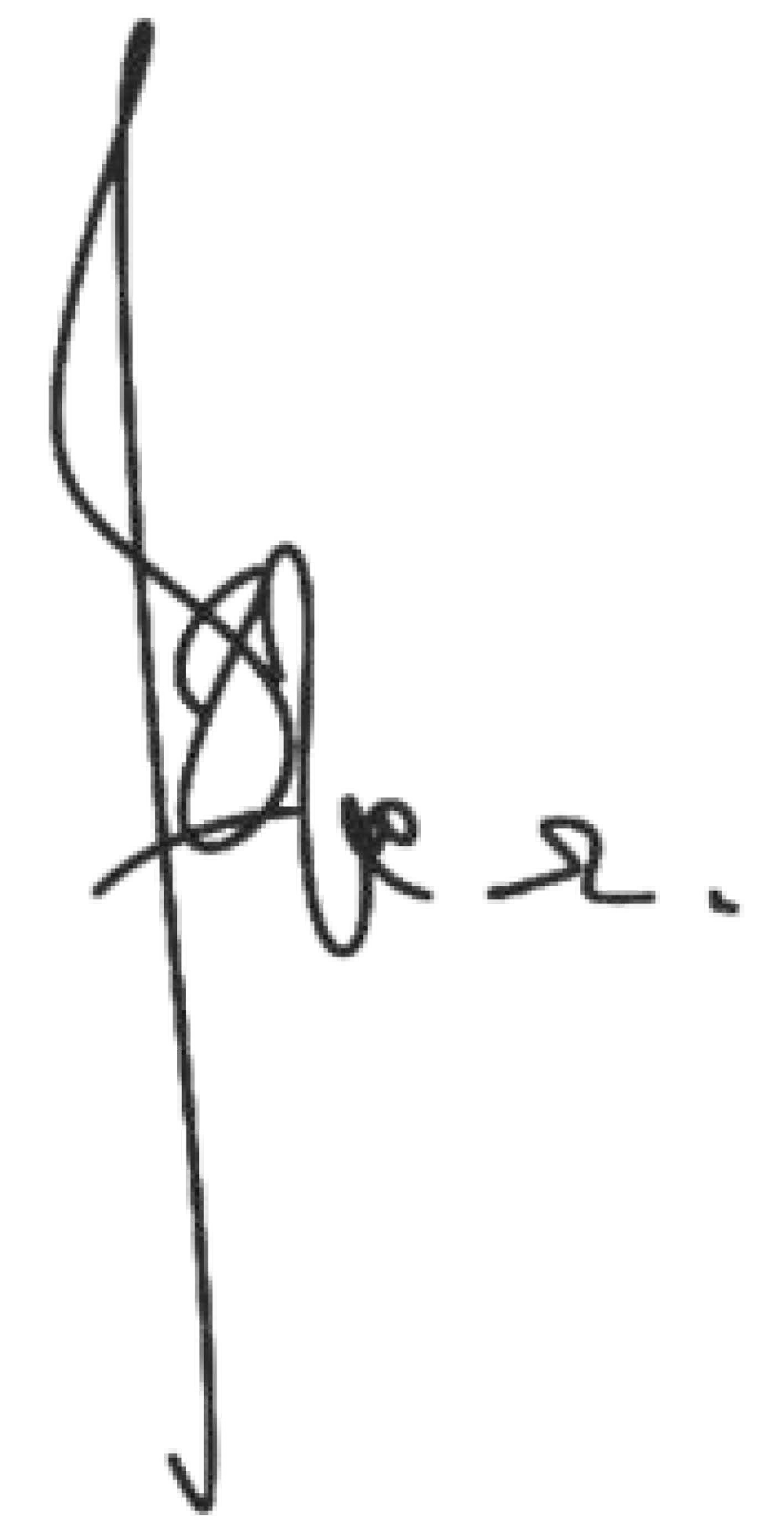
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik

Disusun Oleh :

Ghifari Adlil Baqi NIM: 195150301111004

Skripsi ini telah diuji dan dinyatakan lulus pada 24 Juli 2023

Telah diperiksa dan disetujui oleh: Dosen Pembimbing 1

Agung Setia Budi, S.T., M.T., M.Eng., Ph.D. NIP: 198704232022031003

Mengetahui

Ketua Departemen Teknik Informatika

Achmad Basuki, S.T., M.MG., Ph.D NIP: 197411182003121002

# PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur- unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan

Pasal 70).

Malang, 24 Juli 2023





Ghifari Adlil Baqi NIM: 195150301111004

# PRAKATA

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan berkat, rahmat, dan karunianya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“**Mekanisme Penyediaan Layanan pada Perangkat Smart Home Berbasis ESP32 Berdasarkan Dataset Time Series Menggunakan K-Means**”** dengan baik, lancar, dan tepat waktu sesuai dengan rencana penulis sebagai syarat dalam memperoleh gelar Sarjana Teknik pada program studi Teknik Komputer Fakultas Ilmu Komputer Universitas Brawijaya.

Pengerjaan penelitian dan penyusunan skripsi ini tidak lepas dari dukungan, bantuan, serta doa dari beberapa pihak. Dengan ini, penulis mengucapkan rasa hormat dan banyak terima kasih kepada:

1. Bapak Prof. Ir. Wayan Firdaus Mahmudy, S.Si., MT., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya,
2. Bapak Achmad Basuki, S.T, M.MG., Ph.D. selaku Ketua Departemen Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya,
3. Bapak Barlian Henryranu Prasetio, S.T., M.T., Ph.D. selaku Ketua Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya,
4. Bapak Agung Setia Budi, S.T., M.T., M.Eng., Ph.D. selaku dosen pembimbing pertama yang telah memberikan saran, masukan, serta dukungan kepada penulis selama proses penelitian dan penyusunan skripsi,
5. Seluruh jajaran dosen dan tenaga pendidik Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya,
6. Kedua orang tua penulis serta keluarga yang telah memberikan dukungan, memberikan semangat, serta selalu mendoakan penulis sehingga skripsi ini dapat selesai,
7. Seluruh teman-teman Teknik Komputer yang telah membantu serta mendukung penulis untuk menyelesaikan skripsi,
8. Semua pihak yang tidak dapat dituliskan satu persatu.

Dengan segala kerendahan hati, penulis menyadari bahwa laporan ini jauh dari kesempurnaan, penulis mengharapkan apabila terdapat kritik maupun saran yang bersifat membangun dari para dosen penguji dan pembaca, Penulis berharap laporan ini dapat memberikan manfaat bagi yang membutuhkan.

Malang, 24 Juli 2023

Penulis [ghifaribaqi@student.ub.ac.id](mailto:ghifaribaqi@student.ub.ac.id)

# ABSTRAK

##### Ghifari Adlil Baqi, Mekanisme Penyediaan Layanan pada Perangkat Smart Home Berbasis ESP32 Berdasarkan Dataset Time Series Menggunakan K-Means

**Pembimbing: Agung Setia Budi, S.T., M.T., M.Eng., Ph.D.**

Perkembangan teknologi Internet of Things (IoT) telah membawa inovasi dalam bentuk perangkat rumah pintar. Dalam konteks ini, penelitian ini bertujuan untuk mengusulkan mekanisme penyediaan layanan yang efisien pada perangkat Smart Home berbasis ESP32 menggunakan dataset time series dan algoritma K- Means, dengan fokus pada objek tunggal yaitu lampu. Sensor yang digunakan dalam sistem ini terdiri dari modul RTC dan modul relay

Pertama, dataset time series dikumpulkan dan berisi informasi mengenai penggunaan lampu dalam periode waktu tertentu. Kemudian, algoritma K-Means diterapkan untuk menganalisis pola penggunaan lampu berdasarkan dataset tersebut. K-Means adalah algoritma pembelajaran mesin yang digunakan untuk mengelompokkan data berdasarkan kesamaan mereka. Dalam penelitian ini, algoritma ini digunakan untuk mengelompokkan pola penggunaan lampu yang serupa agar dapat diidentifikasi pola umum. Setelah mengidentifikasi pola penggunaan lampu, mekanisme penyediaan layanan dikembangkan. Ketika pengguna mengirimkan permintaan melalui antarmuka pengguna, sistem akan menggunakan informasi dari pola penggunaan lampu yang telah diidentifikasi untuk memberikan layanan yang diminta. Sebagai contoh, jika pengguna menginginkan lampu menyala pada waktu tertentu, sistem akan merespons dengan mengontrol modul relay untuk menyalakan atau mematikan lampu sesuai dengan pola yang telah dipelajari dari dataset

Hasil penelitian menunjukkan bahwa mekanisme penyediaan layanan yang diusulkan pada perangkat Smart Home berbasis ESP32 mampu memberikan respons yang cepat dan akurat terhadap permintaan pengguna. Dengan menggunakan dataset time series dan algoritma K-Means, sistem dapat mempelajari pola penggunaan lampu yang ada dan membuat keputusan yang tepat dalam menyediakan layanan yang diinginkan.

Penelitian ini memiliki implikasi positif terhadap pengembangan sistem Smart Home yang lebih efisien dan responsif. Selain itu, mekanisme penyediaan layanan yang diusulkan juga dapat diterapkan pada objek lain dalam lingkungan Smart Home, memungkinkan pengguna untuk mengontrol dan mengelola berbagai aspek kehidupan sehari-hari dengan lebih mudah dan efisien

**Kata kunci: *Smart Home*, ESP32, *Internet of Things*, mekanisme penyediaan layanan, dataset time series, K-Means, lampu, RTC, Modul relay**

# ABSTRACT

##### Ghifari Adlil Baqi, Service Provision Mechanism on ESP32-Based Smart Home Devices Based on Time Series Datasets Using K-Means

**Supervisors: Agung Setia Budi, S.T., M.T., M.Eng., Ph.D.**

The advancement of Internet of Things (IoT) technology has brought about innovation in the form of smart home devices. In this context, this research aims to propose an efficient service provisioning mechanism for ESP32-based Smart Home devices, focusing on a single object, namely a lamp, utilizing time series datasets and the K-Means algorithm. The system incorporates sensors, including an RTC module and a relay module.

To begin with, time series datasets are collected, containing information about lamp usage during specific time periods. Subsequently, the K-Means algorithm is applied to analyze lamp usage patterns based on the collected datasets. K-Means is a machine learning algorithm employed to cluster data based on their similarities. In this study, it is utilized to group similar lamp usage patterns, facilitating the identification of common trends. Once the lamp usage patterns are identified, the service provisioning mechanism is developed. When users send requests through the user interface, the system leverages the information from the identified lamp usage patterns to provide the requested services. For instance, if a user wants the lamp to turn on at a particular time, the system responds by controlling the relay module to switch the lamp on or off, based on the learned patterns from the dataset.

The research results demonstrate that the proposed service provisioning mechanism for ESP32-based Smart Home devices is capable of delivering fast and accurate responses to user requests. By utilizing time series datasets and the K- Means algorithm, the system can learn existing lamp usage patterns and make appropriate decisions in providing the desired services.

This research holds positive implications for the development of more efficient and responsive Smart Home systems. Furthermore, the proposed service provisioning mechanism can also be applied to other objects within the Smart Home environment, empowering users to easily and efficiently control and manage various aspects of their daily lives.

**Keywords: Smart Home, ESP32, Internet of Things, service provision mechanism, Time Series, K-Means, lamp, RTC, Relay module**.

# DAFTAR ISI

[Mekanisme Penyediaan Layanan pada Perangkat *Smart Home* Berbasis](#_bookmark0)

[ESP32 Berdasarkan Dataset Time Series Menggunakan K-Means i](#_bookmark0)

[PENGESAHAN ii](#_bookmark1)

[PERNYATAAN ORISINALITAS iii](#_bookmark2)

[PRAKATA iv](#_bookmark3)

[ABSTRAK vi](#_bookmark4)

[ABSTRACT vii](#_bookmark5)

[DAFTAR ISI viii](#_bookmark6)

[DAFTAR TABEL xi](#_bookmark7)

[DAFTAR GAMBAR xii](#_bookmark8)

[BAB 1 PENDAHULUAN 1](#_bookmark9)

* 1. [Latar Belakang 1](#_bookmark10)
  2. [Rumusan Masalah 2](#_bookmark11)
  3. [Tujuan 2](#_bookmark12)
  4. [Manfaat 2](#_bookmark13)
  5. [Batasan Masalah 2](#_bookmark14)
  6. [Sistematika Pembahasan 2](#_bookmark15)

[BAB 2 LANDASAN KEPUSTAKAAN 4](#_bookmark16)

* 1. [Tinjauan Pustaka 4](#_bookmark17)
     1. [Thresholding based Smart Home Automation System](#_bookmark19)

[using K-means 5](#_bookmark19)

* + 1. [Smart Lamp Control Based on User Behavior For Two](#_bookmark20)

[Lamps Using K-Nearest Neighbour 5](#_bookmark20)

* + 1. [K-Means Clustering Algorithm for Smart Home](#_bookmark21) [Automation 6](#_bookmark21)
  1. [Dasar Teori 6](#_bookmark22)
     1. [Lampu 6](#_bookmark23)
     2. [EEPROM 6](#_bookmark24)
     3. [K-means 7](#_bookmark25)

[BAB 3 METODOLOGI PENELITIAN 9](#_bookmark27)

* 1. [Tipe Penelitian 9](#_bookmark28)
  2. [Metode Penelitian 9](#_bookmark29)
     1. [Studi Literatur 10](#_bookmark31)
     2. [Rekayasa Kebutuhan 10](#_bookmark32)
     3. [Perancangan dan Implementasi 11](#_bookmark33)
     4. [Pengambilan dan Pengolahan Data 11](#_bookmark34)
     5. [Pengujian Sistem 11](#_bookmark36)
     6. [Analisis Hasil Pengujian 12](#_bookmark38)
     7. [Kesimpulan dan Saran 12](#_bookmark39)
  3. [Lokasi Penelitian 12](#_bookmark40)
  4. [Peralatan Pendukung Yang Digunakan 12](#_bookmark41)

[BAB 4 REKAYASA KEBUTUHAN 13](#_bookmark42)

* 1. [Kajian Masalah 13](#_bookmark43)
  2. [Identifikasi Stakeholder 13](#_bookmark44)
  3. [Kebutuhan Fungsional 13](#_bookmark45)
  4. [Spesifikasi Sistem 14](#_bookmark46)
  5. [Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak 14](#_bookmark47)
     1. [Kebutuhan Perangkat Keras 14](#_bookmark48)
     2. [Kebutuhan Perangkat Lunak 17](#_bookmark55)

[BAB 5 PERANCANGAN DAN IMPLEMENTASI 18](#_bookmark56)

* 1. [Perancangan 18](#_bookmark57)
     1. [Perancangan Perangkat Keras 18](#_bookmark59)
     2. [Perancangan Pelangkat Lunak 22](#_bookmark64)
  2. [Implementasi Sistem 24](#_bookmark67)
     1. [Implementasi Perangkat Keras 25](#_bookmark68)
     2. [Implementasi Perangkat Lunak 26](#_bookmark73)

[BAB 6 PENGUJIAN DAN ANALISIS 41](#_bookmark74)

* 1. [Pengujian Nilai Centroid 41](#_bookmark75)
     1. [Prosedur Pengujian Nilai Centroid 41](#_bookmark76)
     2. [Hasil dan Analisis Pengujian Nilai Centroid 41](#_bookmark77)
  2. [Pengujian Keseluruhan Sistem 42](#_bookmark79)
     1. [Prosedur Pengujian Keseluruhan Sistem 42](#_bookmark80)
     2. [Hasil dan Analisi Pengujian Keseluruhan Sistem 43](#_bookmark81)

[BAB 7 Penutup 46](#_bookmark84)

* 1. [Kesimpulan 46](#_bookmark85)
  2. [Saran 46](#_bookmark86)

[DAFTAR REFERENSI 47](#_bookmark87)

[LAMPIRAN A KODE PROGRAM 49](#_bookmark88)

[A.1 Kode Program Keseluruhan Sistem 49](#_bookmark89)

[LAMPIRAN B DOKUMENTASI 55](#_bookmark90)

[B.1 Dokumentasi Pengunjian sistem 55](#_bookmark91)

# DAFTAR TABEL

[Tabel 2.1 Daftar Tinjauan Pustaka 4](#_bookmark18)

[Tabel 3.1 Pengambilan dan Pengolahan Data 11](#_bookmark35)

[Tabel 3.2 *Dataset Time Series* yang telah di kelompokan 11](#_bookmark37)

[Tabel 4.1 Spesifikasi ESP32 14](#_bookmark50)

[Tabel 4.2 Spesifikasi RTC DS1307 16](#_bookmark52)

[Tabel 4.3 Spesifikasi Relay Module 2 16](#_bookmark54)

[Tabel 5.1 Konfigurasi Pin 19](#_bookmark61)

[Tabel 6.1 Tabel Pengujian Nilai Centroid 41](#_bookmark78)

[Tabel 6.2 Tabel Hasil Pengujian Keseluruhan Sistem (Fungsional*)* 43](#_bookmark82)

[Tabel 6.3 Tabel Pengujian Akurasi Centroid K-Means 44](#_bookmark83)

# DAFTAR GAMBAR

[Gambar 2.1 *K-Means* 7](#_bookmark26)

[Gambar 3.1 Diagram Alir Penelitian 10](#_bookmark30)

[Gambar 4.1 ESP32 14](#_bookmark49)

[Gambar 4.2 Real Time Clock (RTC) 15](#_bookmark51)

[Gambar 4.3 Relay Module 16](#_bookmark53)

[Gambar 5.1 Skema Perancangan Sistem 18](#_bookmark58)

[Gambar 5.2 Skematik Perancangan Sistem 19](#_bookmark60)

[Gambar 5.3 Perancangan *Cover* Sistem 20](#_bookmark62)

[Gambar 5.4 Diagram Alir Penyimpanan EEPROM 21](#_bookmark63)

[Gambar 5.5 Diagram Alir Pembuatan Dataset 23](#_bookmark65)

[Gambar 5.6 Diagram Alir Keseluruhan Sistem 24](#_bookmark66)

[Gambar 5.7 Sistem Tampak Depan 25](#_bookmark69)

[Gambar 5.8 Sistem Tampak Belakang 25](#_bookmark70)

[Gambar 5.9 Sistem Tampak Samping 25](#_bookmark71)

[Gambar 5.10 Sistem Tampak Dalam 26](#_bookmark72)

# BAB 1 PENDAHULUAN

Pada bab 1 atau bab pendahuluan merupakan bab pertama dalam penelitian yang berisikan uraian mengenai latar belakang penelitian, rumusan masalah yang diangkat penulis, tujuan dan manfaat penelitian, serta Batasan masalah dari lingkup penelitian ini. Pada bab ini terdapat sistematika penulisan mengenai isi secara umum dari setiap bab yang terdapat dalam penelitian ini.

## Latar Belakang

Penggunaan teknologi Internet of Things (IoT) pada Smart Home semakin berkembang pesat. Salah satu teknologi IoT yang sering digunakan pada Smart Home adalah ESP32, sebuah sistem mikrokontroler yang dapat digunakan untuk mengendalikan perangkat rumah secara nirkabel. Namun, untuk memaksimalkan kinerja perangkat Smart Home, diperlukan mekanisme penyediaan layanan yang efektif. Menurut Kaur et al. (2017), Smart Home adalah sebuah konsep di mana berbagai perangkat rumah tangga terhubung satu sama lain melalui internet dan dapat dikontrol oleh pengguna melalui perangkat mobile seperti smartphone, tablet, atau laptop.

Dataset Time Series dapat digunakan untuk mengidentifikasi pola atau tren dari data yang telah dikumpulkan pada Smart Home. Dalam hal ini, dataset dapat diperoleh dari pengukuran atau pengamatan pada waktu yang berbeda-beda menggunakan sensor-sensor pada perangkat rumah yang terpasang pada Smart Home. Menurut Tripathi et al. (2017), data pada Smart Home dapat diperoleh melalui sensor-sensor yang terpasang pada perangkat rumah seperti sensor suhu, sensor cahaya, sensor kelembaban, dan sebagainya.

Namun, penelitian yang lebih spesifik dan detail mengenai mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 berdasarkan dataset time series menggunakan K-Means masih sangat terbatas. Oleh karena itu, penelitian ini diusulkan untuk mempelajari mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 berdasarkan dataset time series menggunakan K-Means. Beberapa faktor yang perlu dipertimbangkan dalam penelitian ini adalah pemilihan parameter centroid pada K-Means, metode pengambilan data pada dataset, serta pengukuran kinerja metode K-Means Dalam penelitian ini, akan dilakukan pengumpulan data sensor pada perangkat rumah di Smart Home untuk membentuk dataset time series. Kemudian, akan dilakukan pemrosesan data menggunakan metode K-Means untuk mengklasifikasikan data sensor dan memberikan respons yang sesuai untuk mengendalikan perangkat rumah pada Smart Home. Hasil dari penelitian ini diharapkan dapat membantu meningkatkan efektivitas dan efisiensi dalam penggunaan perangkat Smart Home berbasis ESP32.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, rumusan masalah dalam penelitian ini adalah sebagai berikut:

* + 1. Bagaimana akurasi perhitungan centroid *K-Means* di ESP32?
    2. Bagaimana kinerja secara keseluruhan sistem dalam hal implementasi mekanisme penyediaan layanan yang dibangun menggunakan algoritma *K- Means* pada dataset *time series*?

## Tujuan

Adapun beberapa tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah sebagai berikut:

* + 1. Mengetahui sejauh mana ESP32 dapat menghasilkan centroid yang representatif dari dataset time series yang diberikan.
    2. Mengetahui kinerja sistem terhadap implementasi mekanisme penyediaan layanan pada perangkat smart home berbasis ESP32 menggunakan algoritma K-Means pada dataset time series.

## Manfaat

Penyediaan Layanan ini diharapkan mampu meningkatkan pemanfaatan teknologi terbaru dalam kehidupan sehari-hari, yaitu *smart home*. Yang dimana dapat membantu memudahkan dan mempercepat berbagai kegiatan di rumah. Selain itu, penulis juga berharap agar masyarakat yang membaca skripsi ini menjadi meningkatkan pemahaman mengenai *smart home*.

## Batasan Masalah

Batasan masalah adalah batasan-batasan dari sistem dan penelitian yang dilakukan. Batasan masalah dalam penelitian ini adalah:

1. Implementasi sistem berupa prototipe
2. Subjek penelitian yang digunakan adalah lampu
3. Dataset yang digunakan dalam penelitian ini adalah dataset time series yang diperoleh dari kebiasaan pengguna diperangkat smart home berbasis ESP32.

## Sistematika Pembahasan

Sistematika pembahasan dibuat bertujuan untuk membantu memberikan pemahaman mengenai alur pembahasan dari penelitian ini, sebagai berikut:

##### BAB 1 Pendahuluan

Penelitian ini terdiri dari bagian-bagian bab yang disusun secara sistematis, yang akan memberikan gambaran dan penjelasan mengenai penelitian ini. Sistematika penulisan skripsi dibagi menjadi beberapa bab, yaitu:

BAB I PENDAHULUAN

Bab ini berisikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan skripsi.

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisikan kajian kepustakaan dan landasan teori dari penelitian terdahulu. Kajian dan landasan teori yang digunakan harus berkaitan penelitian saat ini dan mampu menguraikan teori dan metode yang dipakai.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan langkah-langkah dan metode yang akan digunakan dalam penelitian. Pada bab ini juga akan dijelaskan lebih terperinci mengenai metode yang akan digunakan oleh penulis seperti blok diagram dan flowchartnya.

BAB IV REKAYASA KEBUTUHAN

Bab ini berisikan pembahasan mengenai batasan sistem, kebutuhan sistem meliputi perangkat lunak dan perangkat keras, dan alat serta bahan yang akan digunakan.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini berisikan bagaimana proses perancangan dan pengimplementasian sistem dalam menjalankan algoritme *K-Means* untuk klasifikasi pada ESP32.

BAB VI PENGUJIAN DAN ANALISIS

Bab ini berisikan hasil pengujian sistem yang telah diimplementasikan. Pada bab ini juga akan menjawab semua pertanyaan pada rumusan masalah yang sebelumnya telah dipaparkan.

BAB VII PENUTUP

Bab ini berisikan kesimpulan dan saran yang didapat setelah penelitian dilakukan. Saran ini nantinya akan berguna untuk keberlanjutan penelitian yang akan datang.

# BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi kajian literatur dan teori guna menunjang ketersediaan informasi dalam penyusunan penelitian. Dalam tinjauan pustaka terdapat beberapa kajian yang telah dilakukan sebelumnya sebagai sumber referensi dalam penyusunan penelitian penulis. Kemudian teori dasar memuat istilah-istilah yang akan digunakan oleh penulis, seperti subjek penelitian, parameter, alat, dan metode.

## Tinjauan Pustaka

Berikut adalah isi dari kajian pustaka yang berisikan penelitian-penelitian sebelumnya, dapat dilihat pada Tabel.2.1.

**Tabel 2.1 Daftar Tinjauan Pustaka**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Penulis (Tahun),  Judul Penelitian | Persamaan Penelitian | Perbedaan | |
| Penelitian Sebelumnya | Penelitian Sekarang |
| 1 | **Aakash Katiyar (2023),**  **“*Thresholding based Smart Home Automation System using K-means*”** | Menggunakan metode K- Means untuk melakukan klasifikasi | Menggunakan ESP8266  dengan parameter riwayat penggunaan pendingin rungan untuk melakukan klasifikasi | Menggunakan parameter riwayat penggunaan peralatan rumah tangga dengan metode K-Means untuk melakukan klasifikasi |
| 2 | **Triono Nugroho (2019) “*Smart Lamp Control Based on User Behavior For Two Lamps Using K- Nearest Neighbour*”** | Menggunakan kebiasaan perilaku pengguna untuk kontrol lampu sebagai dataset | Pengambilan data set masih secara manual dengan di masukan langsung kedalam kode program | Pengambilan data set dapat melalui tombol agar lebih memudahkan pengguna |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | **Puji Catur**  **Siswipraptini (2020), “*K-Means Clustering Algorithm for Smart Home Automation*”** | menggunakan ESP32 dengan metode K- Means | menggunakan Arduino Uno dengan metode K- Means | Menggunakan ESP32 dengan parameter riwayat penggunaan peralatan rumah tangga dengan metode *K-Means* untuk melakukan klasifikasi |

* + 1. ***Thresholding based Smart Home Automation System using K-means***

Penelitian yang dilakukan oleh Aakash Katiyar (2023), dengan judul “*Thresholding based Smart Home Automation System using K-means*” merupakan penelitian yang telah membuka pikiran penulis dalam menemukan ide. Penelitian ini dilakukan dua minggu pembacaan, total observasi yang disusun adalah 93 baris. Pembacaan tersebut mencakup suhu ruang tamu (T1), suhu kamar tidur (T2), dan suhu dapur (T3). Hasil analisis menunjukkan bahwa rata-rata suhu masing-masing ruangan adalah 24,35°C, 22,06°C, dan 27,33°C dengan standar deviasi yang hampir sama. Pembacaan ini dilakukan dalam mode manual untuk memahami preferensi pengguna terkait suhu yang mereka sukai untuk menghidupkan alat listrik. Dengan menggunakan data manual tersebut, peneliti menjalankan program pembelajaran mesin K-Means untuk mencari titik temu nilai suhu, kelembapan, dan intensitas cahaya. Penelitian juga menggunakan teknik pemrosesan data seperti binning dan transformasi untuk memproses data dengan lebih baik dan mudah. Konsep kurva k-elbow digunakan untuk menentukan jumlah kluster yang paling optimal untuk suhu dan kelembapan setiap ruangan. Program K-Means juga digunakan untuk mengidentifikasi kemungkinan ambang batas. Meskipun jumlah data terbatas hanya 93 baris, penelitian tetap berhasil memperoleh berbagai ambang batas yang berbeda. Setelah mengidentifikasi klaster untuk suhu dan kelembapan, penelitian mengambil rata-rata dari semua titik klaster yang terkait untuk mendapatkan ambang batas yang sebenarnya.

#### Smart Lamp Control Based on User Behavior For Two Lamps Using K- Nearest Neighbour

Penelitian yang dilakukan oleh Triono Nugroho (2019) yang berjudul “*Smart Lamp Control Based on User Behavior For Two Lamps Using K-Nearest Neighbour*” merupakan penelitian yang telah membuka pikiran penulis dalam menemukan ide. Penelitian ini dilakukan mengklasifikasikan data kebiasaan pengguna menggunakan metode K-Nearest Neighbor. Dari hasil pengujian kinerja *system* dalam mengklasifikasikan data menghasilkan akurasi rata-rata 97,62% untuk lampu satu dan 98,36% untuk lampu kedua.

#### K-Means Clustering Algorithm for Smart Home Automation

Penelitian selanjutnya yaitu penelitian yang dilakukan oleh Puji Catur Siswipraptini (2020) dengan judul “*K-Means Clustering Algorithm for Smart Home Automation*”. Penelitian ini dilakukan untuk menganalisis pola konsumsi listrik pengguna aplikasi rumah pintar. Data yang digunakan berasal dari sensor yang terpasang pada smart prototipe rumah. Data dikelompokkan dan hasilnya digunakan untuk beri tahu pengguna apakah listrik harian rumah pintar konsumsi telah melebihi penggunaan normalnya. Notifikasi dan informasi real time tentang penggunaan listrik (dalam KWH, Kilo Watt Hour) disajikan dalam aplikasi. Penelitian ini adalah diharapkan dapat membantu pengguna untuk mengambil tindakan yang lebih baik tentang rumahnya penggunaan listrik. Metode K-Means digunakan untuk clustering karena mampu mengolah data dalam jumlah besar secara efisien dengan waktu komputasi yang singkat. Ada dua cluster yang dihasilkan, yaitu konsumsi energi tinggi dan konsumsi energi rendah. Itu nilai yang diperoleh dari validasi cluster menggunakan Davies Indeks Bouldin adalah 0,417103167 (tidak negatif dan mendekati 0), jadi hasil clustering dianggap optimal.

## Dasar Teori

Subbagian ini menjelaskan beberapa teori yang diperlukan untuk melakukan penelitian ini. Untuk mendukung pelaksanaan penelitian dan memastikan hasil yang diharapkan dapat dicapai secara optimal, pengetahuan teoritis sangat penting.

### Lampu

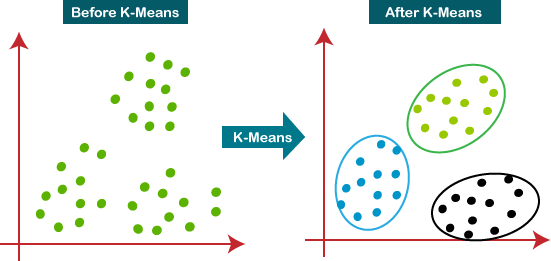
Lampu merupakan perangkat penerangan yang digunakan untuk menghasilkan cahaya buatan. Lampu sangat penting dalam kehidupan sehari-hari, karena memberikan pencahayaan yang diperlukan di berbagai ruangan dan area. Lampu dapat menggunakan berbagai sumber energi, seperti listrik, gas, atau panas, tergantung pada jenisnya. Beberapa jenis lampu yang umum digunakan adalah lampu pijar, lampu fluoresensi, dan lampu LED. Lampu juga dapat dikontrol menggunakan sistem pengendali, seperti sakelar atau dimmer, untuk mengatur kecerahan atau waktu hidup lampu. Dengan berbagai jenis dan kemajuan dalam teknologi pencahayaan, lampu telah berkembang menjadi lebih efisien, tahan lama, dan dapat disesuaikan dengan kebutuhan pencahayaan yang berbeda-beda.

#### EEPROM

EEPROM merupakan jenis ROM non-volatile yang memungkinkan setiap byte data dihapus dan diprogram ulang. Itulah mengapa chip EEPROM dikenal sebagai chip yang dapat dihapus byte. EEPROM biasanya digunakan untuk menyimpan sejumlah kecil data dalam komputasi dan perangkat elektronik lainnya.

#### K-means

K-Means Clustering adalah salah satu algoritma *Machine Learning* yang termasuk dalam kategori *Unsupervised Learning* yang sering digunakan. Dalam algoritma ini, kita perlu menentukan nilai k sebagai jumlah kluster yang akan dibentuk sebelum memulai proses K-Means *Clustering*. Sebelum menggunakan algoritma ini, kita harus menentukan nilai k terlebih dahulu, yang merepresentasikan jumlah centroid atau titik pusat dari setiap kluster yang dibutuhkan dalam dataset. (Dqlab, 2022). Dengan menggunakan K-Means *Clustering*, kita dapat membagi dataset yang tidak memiliki label menjadi k kluster yang berbeda, sehingga setiap kluster hanya memuat data yang memiliki karakteristik serupa. Algoritma ini bersifat iteratif, yang memungkinkan pembagian kluster yang lebih akurat dan efektif



**Gambar 2.1 *K-Means***

(**Sumber**: javatpoint.com, 2021)

###### Euclidean Distance

*Euclidean Distance* merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam euclidean space (meliputi bidang Euclidean dua dimensi, tiga dimensi atau bahkan lebih) (Nishom, 2019). rumus *Euclidean Distance* untuk mengukur tingkat kemiripan data menggunakan persamaan 2.1.

𝑑(𝑥, 𝑦) = |𝑥 − 𝑦| = √∑𝑁 (𝑥𝑖 − 𝑦𝑖)2 (2.1)

Keterangan:

*d* = jarak antara x dan y

*x* = data pusat klaster *y* = data pada atribut *i* = setiap data

*n* = jumlah data

*xi* = data pada pusat klaster ke i

*yi* = data pada setiap data ke i

*Euclidean Distance* mempunyai kelebihan dan kekurangan dibandingkan dengan metode perehitungan lain. Kelebihan dalam *Euclidean Distance* adalah sebagai berikut:

* + - * 1. Perhitungan jarak menggunakan *Euclidean Distance* lebih umum digunakan dalam metode *K- Means*.
        2. *Euclidean Distance* mempunyai hasil yang lebih optimal dibandingan dengan perhitungan yang lainnya.
        3. Referensi dari *Euclidean Distance* lebih banyak karena secara umum perhitungan *Euclidean Distance* digunakan dalam perhitungan jarak pada *K- Means Clustering*.

Kekurangan dari Euclidean Distance adalah sebagai berikut:

1. Perhitungan manual menggunakan Euclidean Distance cenderung lebih lama karena jumlah iterasi Euclidean Distance lebih banyak dibandingkan dengan Manhattan Distance (Fajriah, 2019)

# BAB 3 METODOLOGI PENELITIAN

Bab metodologi merupakan bab ketiga dalam penelitian ini yang memiliki beberapa sub bab yang akan dibahas. Pertama, bab ini akan menjelaskan jenis penelitian dari penelitian penulis. Selanjutnya terdapat metode penelitian yang akan penulis gunakan yang disajikan dalam bentuk flowchart penelitian. Kemudian juga mendiskusikan partisipan dalam penelitian ini. Keberadaan partisipan dalam penelitian ini sangat penting bagi penulis. Setelah itu, peneliti akan menginformasikan mengenai lokasi penelitian. Terakhir, penulis juga akan menyertakan peralatan yang mendukung penelitian.

## Tipe Penelitian

Penelitian ini menggunakan tipe penelitian Implementatif - Perancangan (Design). Tipe penelitian ini bertujuan untuk merancang dan mengimplementasikan suatu sistem atau mekanisme yang dapat memenuhi tujuan penelitian yang telah ditetapkan. Dalam penelitian ini, peneliti merancang dan mengimplementasikan mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 berdasarkan dataset time series menggunakan metode K-Means. Objek yang digunakan dalam penelitian ini adalah lampu dan RTC (Real-Time Clock) serta 2 relay module digunakan sebagai alat atau sensor tambahan.

## Metode Penelitian

Model metodologi penelitian yang digunakan dalam penelitian ini ditampilkan dalam bentuk diagram alir. Diagram alir ini akan menjelaskan alur penelitian yang digunakan dalam penelitian ini. Alur penelitian tersebut dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Diagram Alir Penelitian**

### Studi Literatur

Penelitian mengumpulkan informasi mengenai topik penelitian, teori-teori terkait, dan penelitian terdahulu yang relevan. Dengan melakukan studi literatur ini, penulis dapat merancang aspek-aspek yang telah diimplementasikan dari jurnal-jurnal yang berkaitan. Adapun beberapa jenis aspek, yaitu meliputi Konsep Smart Home dan perangkat ESP32, RTC (Real-Time Clock), Relay module, pengimplementasian metode K-Means kepada lampu sebagai objek dari dataset time series dalam Smart Home,

### Rekayasa Kebutuhan

Rekayasa kebutuhan dilakukan untuk menentukan kebutuhan dan tujuan dari mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 berdasarkan dataset time series menggunakan metode K-Means. Beberapa tahapan dalam rekayasa kebutuhan meliputi identifikasi kebutuhan pengguna untuk melakukan analisis agar memahami kebutuhan pengguna terkait mekanisme penyediaan layanan pada Smart Home khususnya terkait penggunaan lampu. Menyusun daftar kebutuhan bertujuan Menyusun kebutuhan yang harus dipenuhi oleh sistem, termasuk kemampuan analisis time series, pengelompokan data menggunakan K-Means, pengendalian lampu berdasarkan hasil clustering,

dan integrasi dengan RTC dan relay module.Menetapkan kriteria evaluasi bertujuan untuk kriteria evaluasi kinerja sistem, seperti akurasi clustering, efisiensi waktu, dan responsivitas layanan, yang akan digunakan dalam tahap evaluasi sistem. .

### Perancangan dan Implementasi

Selanjutnya tahap perancangan dan implementasi terdiri dari perancangan perangkat keras dan perangkat lunak. Pada tahap perancangan perangkat keras dibuat diagram skematik dari seluruh komponen perangkat keras yang digunakan pada sistem, seperti ESP32, RTC, 2 Modul Relay. Pada tahap ini juga dilakukan desain penutup pada sistem untuk mendukung mobilitas sistem, serta melindungi sistem dari kerusakan seperti korsleting akibat paparan udara atau zat lain.

Pada tahap perancangan perangkat lunak, dibuat kode program agar semua komponen perangkat keras yang digunakan dapat terintegrasi. Sedangkan pada tahap implementasi perangkat keras dilakukan pemasangan atau pengkabelan seluruh komponen perangkat keras yang akan digunakan pada sistem. Kemudian pada tahap implementasi perangkat lunak dilakukan dengan mengunggah kode program yang telah terintegrasi ke dalam sistem.

### Pengambilan dan Pengolahan Data

Pada bab ini, akan dijelaskan tentang pengambilan dan pengolahan data dalam mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32. Data yang digunakan adalah dataset time series yang mencatat penggunaan lampu. Selain itu, objek lain yang digunakan dalam penelitian ini adalah RTC (Real- Time Clock) dan 2 relay module. Berikut adalah Tabel 3.1 contoh dataset time series yang digunakan:

**Tabel 3.1 Pengambilan dan Pengolahan Data**

|  |  |  |
| --- | --- | --- |
| Cluster | Lampu | Status Lampu |
| 325 | 1 | Nyala |
| 380 | 0 | Mati |
| 1029 | 1 | Nyala |
| 1373 | 0 | Mati |

### Pengujian Sistem

Pada tahap pengujian dilakukan pengujian terhadap 4 Centroid*.* Tahap pengujian ini berfungsi untuk mengetahui kinerja baik dari masing-masing komponen ataupun keseluruhan sistem. Adapun penjabaran dari beberapa sampel *Dataset Time series* yang akan digunakan dalam pengujian akan disajikan dalam Tabel 3.2 dibawah ini.

**Tabel 3.2 *Dataset Time Series* yang telah di kelompokan**

|  |  |  |
| --- | --- | --- |
| Cluster | Lampu | Status Lampu |
| 310 | 1 | Nyala |
| 393 | 0 | Mati |
| 1028 | 1 | Nyala |
| 1360 | 0 | Mati |

### Analisis Hasil Pengujian

Dalam tahap analisis hasil pengujian dilakukan untuk merangkum hasil dari pengujian yang telah dilakukan sebelumnya. Dimana penulis menghubungkan hasil dengan suatu persamaan yang berlandaskan pada dasar teori.

### Kesimpulan dan Saran

Pada tahapan akhir dari penelitian ini, penulis akan menarik suatu kesimpulan yang akan menjawab rumusan masalah dari penelitian ini. Kesimpulan tersebut juga didapat berdasarkan hasil dari pengujian yang telah dilakukan. Selain itu, penulis juga memberikan saran yang harapannya agar penelitian ini dapat dikembangkan lagi di masa mendatang.

## Lokasi Penelitian

Lokasi penelitian bertempat di sekitaran tempat tinggal penulis. Hal ini agar lebih mudah dan efisien.

## Peralatan Pendukung Yang Digunakan

Pada penelitian ini, peralatan pendukung yang digunakan terbagi menjadi dua bagian yaitu perangkat keras dan perangkat lunak. Adapun perangkat keras dan perangkat lunak yang akan digunakan adalah sebagai berikut:

* Perangkat Keras
  + ESP32

- Lampu

* Real Time Clock DS1307
* 2 Relay Module

- Dua buah button

* Perangkat Lunak
  + Arduino IDE
* Pustaka RTClib
* Pustaka EEPROM.h

# BAB 4 REKAYASA KEBUTUHAN

Rekayasa kebutuhan merupakan bab cepat dalam penelitian ini yang berisi uraian terkait meninjau masalah secara umum, mengidentifikasi stakeholder, mengembangkan kebutuhan fungsional dan spesifikasi yang harus dimiliki oleh sistem, juga menganalisis kebutuhan perangkat keras dan perangkat lunak yang menyusun sistem.

## Kajian Masalah

Tujuan dari sistem ini adalah untuk mengoptimalkan penggunaan perangkat smart home dengan menggunakan algoritma K-Means untuk menganalisis data time series yang dihasilkan oleh ESP32. Salah satu tantangan terbesar dalam mengelola perangkat rumah pintar adalah menawarkan layanan efisien yang memenuhi kebutuhan pengguna. Dataset ini berfungsi sebagai dasar untuk mengidentifikasi model yang dapat digunakan untuk mengoptimalkan layanan perangkat smart home.

Algoritma K-Means dipilih sebagai metode analisis data dalam penelitian ini karena kemampuannya mengelompokkan data berdasarkan pola yang ada. Dalam konteks ini, K-Means digunakan untuk mengklasifikasikan dataset deret waktu ke dalam beberapa grup berdasarkan kesamaan pola penggunaan perangkat rumah pintar. Dengan demikian, mekanisme penyediaan layanan dapat diatur dan disesuaikan berdasarkan pola penggunaan yang teridentifikasi.

## Identifikasi Stakeholder

Penerapan algoritma K-Means pada perangkat Smart Home berbasis ESP32 akan memungkinkan pengguna untuk memperoleh layanan yang lebih personal dan efisien. Pengelompokan data time series akan membantu dalam mengidentifikasi perilaku pengguna dan preferensi mereka dalam penggunaan perangkat Smart Home. Dengan demikian, perangkat ESP32 dapat mengoptimalkan respons terhadap permintaan pengguna dan menyediakan layanan yang sesuai dengan kebutuhan individu.

## Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan layanan atau fungsi yang harus dipenuhi agar sistem yang dirancang dapat berjalan sesuai dengan tujuan pembuatannya. Adapun kebutuhan fungsional yang dibutuhkan pada sistem adalah sebagai berikut:

* + 1. Sistem dapat melakukan pengambilan data melalui ESP32 kedalam EEPROM.
    2. Sistem dapat melakukan *clustering* menggunakan metode K-Means.

## Spesifikasi Sistem

Sistem yang dijelaskan pada bab ini ialah prosedur penyediaan layanan pada perangkat Smart Home berbasis ESP32. Sistem ini menggunakan dataset time series buat melakukan analisis dan pengelompokan data menggunakan metode K- Means. Lampu digunakan sebagai objek pada sistem ini, sedangkan RTC (Real- Time Clock) serta 2 relay module digunakan menjadi alat atau sensor tambahan buat menaikkan fungsionalitas sistem.

## Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Pada kebutuhan non-fungsional akan dijelaskan tentang kebutuhan sistem dalam melaksanakan kinerjanya. Kebutuhan Non-fungsional dibagi menjadi dua, yaitu buat kebutuhan perangkat keras serta kebutuhan aplikasi.

### Kebutuhan Perangkat Keras

Kebutuhan perangkat keras sangat perlu disesuaikan antara setiap komponen yang akan digunakan dalam sistem. Hal ini bertujuan supaya mencocokkan masing-masing komponen sesuai spesifikasinya. Berikut merupakan kebutuhan perangkat keras yg akan dipergunakan oleh penulis

##### ESP 32



**Gambar 4.1 ESP32**

(**Sumber:** ecadio.com, 2021)

ESP32 merupakan sebuah mikrokontroler yang dikenalkan oleh *Espressif System* dan merupakan penerus dari mikrokontroler ESP8266. Salah satu kelebihan yang dimiliki oleh ESP32 adalah sudah terdapat Wifi dan Bluetooth di dalamnya, yang akan sangat mempermudah pembuatan sistem IoT yang memerlukan koneksi wireless. Fitur-fitur tersebut tidak ada di dalam ESP8266, sehingga ESP32 merupakan sebuah upgrade dari ESP8266.

##### Tabel 4.1 Spesifikasi ESP32

|  |  |
| --- | --- |
| Mikrokontroler | ESP32 |
| Prosesor Utama | Tensilica Xtensa 32-bit LX6 microprocessor |

|  |  |
| --- | --- |
| Inti | 2 atau 1 |
| Frekuensi | Hingga 240MHz |
| Performa | Hingga 600 DMIPS |
| Wi-Fi | 802.11 b/g/n/e/i (802.11n @ 2,4 GHz hingga 150 Mbit/detik) |
| Bluetooth | v4.2 BR/EDR dan *Bluetooth Low Energy* (BLE) |
| ROM | 448 KiB |
| SRAM | 520 KiB |
| RTC fast SRAM | 8 KiB |
| RTC slow SRAM | 8 KiB |
| eFuse | 1 Kibit |
| Embedded flash | * 0 MiB (ESP32-D0WDQ6, ESP32-   D0WD, and ESP32-S0WD chips)   * 2 MiB (ESP32-D2WD chip) * 4 MiB (ESP32-PICO-D4 SiP   module) |

(**Sumber:** raharja.ac.id, 2021)

##### RTC DS1307

**Gambar 4.2 Real Time Clock (RTC)**

(**Sumber:** lastminuteengineers.com, 2023)

RTC DS1307 adalah modul RTC yang dapat menampilkan info ketika menggunakan satuan detik, menit, jam, hari, tanggal, bulan, dan tahun. chip RTC ini dapat menampilkan waktu dalam format 12 jam maupun 24 jam. Penghitungan bulan dibatasi sampai 31 hari serta akan otomatis berpindah ke bulan selanjutnya

buat setiap bulan yang memiliki jumlah hari kurang dari 31 hari. Berikut ini adalah spesifikasi dari modul RTC DS1307.

##### Tabel 4.2 Spesifikasi RTC DS1307

|  |  |
| --- | --- |
| Seri | RTC DS1307 |
| Tegangan Operasi | 4,5 V – 5,5 V |
| Konsumsi Arus | < 1.5mA |
| Akurasi (0-40°C) | Tergantung pada kristal (± 20ppm type.) |
| Baterai | CR2032 |

(**Sumber:** lastminuteengineers.com, 2023)

##### Relay Module

**Gambar 4.3 Relay Module**

(**Sumber**: indomaker.com, 2022)

##### Tabel 4.3 Spesifikasi Relay Module 2

|  |  |
| --- | --- |
| Jumlah Relay | 2 |
| Arus Maksimum | 10A |
| Jenis Relay | Relai elektromagnetik |
| Tipe Kontak | NO/NC (Normal Open/Normal Close) |
| Input Kontrol | Sinyal Digital |
| Indikator LED | Ya |
| Proteksi | Dioda Perlindungan |
| Sumber Daya | 5V DC (dari Mikrokontroler, dll) |

(**Sumber**: aldyrazor.com, 2020)

Modul relay adalah suatu alat yang beroperasi berdasarkan prinsip elektromagnetik untuk menggerakkan kontaktor untuk memindahkan posisi ON ke OFF atau sebaliknya dengan memanfaatkan listrik. Penutupan dan pembukaan kontaktor terjadi akibat efek induksi magnet yang timbul dari kumparan induksi listrik. Perbedaan yang paling mendasar antara relay dan switch adalah saat berpindah dari posisi ON ke OFF. Relay melakukan perpindahan secara otomatis dengan arus listrik, sedangkan saklar dilakukan secara manual.

### Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak juga sangat dibutuhkan dalam pembuatan sistem. perangkat lunak dipergunakan untuk mengoperasikan komponen-komponen di sistem berupa kode program. Berikut kebutuhan perangkat lunak yg akan digunakan penulis.

###### Arduino IDE

Arduino IDE merupakan perangkat lunak yang dipergunakan untuk mengirimkan perintah atau coding dan akan diproses melalui perangkat keras Arduino Mega. Arduino IDE ini tidak hanya terbatas di board Arduino Mega saja. Melainkan juga dapat digunakan buat aneka macam jenis berasal perangkat keras Arduino lainnya, seperti Arduino Nano, Arduino Uno, ESP32 , serta lainnya.

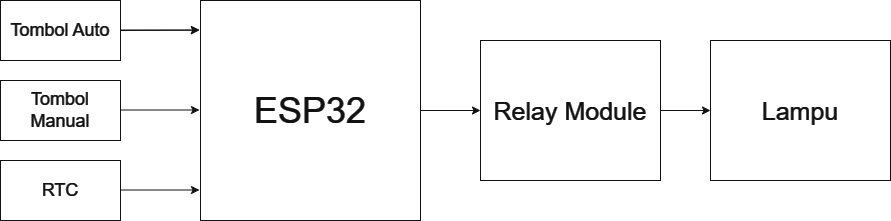
* + - 1. ***Library RTClib.h* (DS1307)**

Library RTC adalah suatu library yang tidak terdapat pada perangkat lunak Arduino IDE secara default. Untuk mengaksesnya dapat dilakukan dengan cara mengunduh file dari library tersebut di internet melalui web browser. RTC Library diperlukan agar penulis dapat menggunakan perintah atau fungsi untuk melakukan akses pada modul RTC DS1307. Baik perintah maupun fungsi yang sering digunakan misalnya seperti fungsi untuk menyimpan waktu pada komputer penulis ke dalam RTC.

# BAB 5 PERANCANGAN DAN IMPLEMENTASI

Proses perancangan sistem berdasarkan hasil rekayasa kebutuhan dijelaskan dalam bab ini. Keputusan yang dibuat berdasarkan pemahaman mendalam tentang kebutuhan pengguna, kendala teknis, dan persyaratan sistem adalah bagian dari perancangan. Rencana yang sistematis dan tercatat akan dibuat melalui proses perancangan yang baik. Penulis akan menggunakan rancangan ini sebagai panduan selama tahap implementasi. Rancangan yang dibuat meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Bab ini juga membahas tentang implementasi sistem yang telah dirancang dan diterapkan agar dapat memenuhi tujuan penelitian. Rencana konseptual diimplementasikan menjadi bentuk praktis atau riil yang dapat digunakan oleh pengguna.

## Perancangan

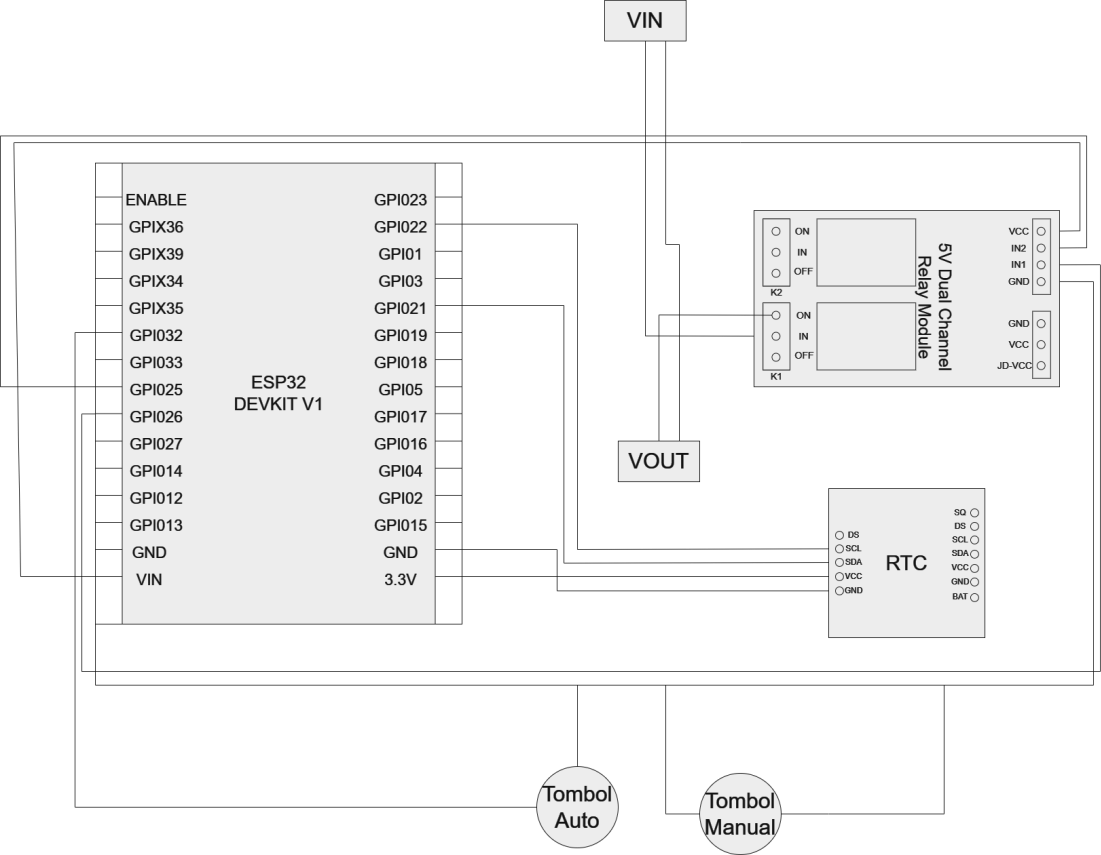
Subbab ini menjelaskan mengenai perancangan sistem yang dibagi menjadi dua tahap, yaitu perancangan perangkat keras (hardware) dan perangkat lunak (software). Gambar 5.1 menunjukkan skema perancangan dari sistem.

##### Gambar 5.1 Skema Perancangan Sistem

Berdasarkan blok diagram pada Gambar 5.1, mula-mula sistem akan mendapatkan nilai input dari data sensor yang dikumpulkan dari tombol manual, dan RTC akan memberikan informasi waktu kepada ESP32. Yang akan di berikan kepada lampu sebagai objek yang menjadi fokus pada mekasinisme penyediaan layanan. Selanjutnya, nilai input tadi akan diproses pada menggunakan algoritma *K-Means*. Indikator ini dihasilkan melalui pembacaan tombol manual dan digunakan sebagai data untuk langkah-langkah selanjutnya. Atribut yang dimanfaatkan dalam data adalah total daya yang terakumulasi selama tiga belas hari untuk setiap perangkat listrik. K-Means Clustering akan memproses data ini untuk membentuk empat klaster, yaitu klaster nyala dan mati yang berulang dalam dua siklus. Informasi mengenai klaster nyala dan mati pada lampu dapat berfungsi sebagai dasar pengetahuan yang berguna dalam menetapkan kebijakan terkait upaya penghematan energi listrik.

### Perancangan Perangkat Keras

Pada tahap ini, penulis membuat rancangan sistem agar saling terkoneksi antar komponen yang satu dengan lainnya. Adapun komponen-komponen yang dimaksud yaitu ESP32, 2 Relay Module, RTC. Untuk lebih jelasnya, rangkaian skematik antar komponen tersebut dapat dilihat pada Gambar 5.2 dibawah ini.



##### Gambar 5.2 Skematik Perancangan Sistem

Untuk melakukan perancangan diagram skematik diatas, penulis menyesuaikan dengan tabel konfigurasi pin yang dapat dilihat pada Tabel 5.1 dibawah ini.

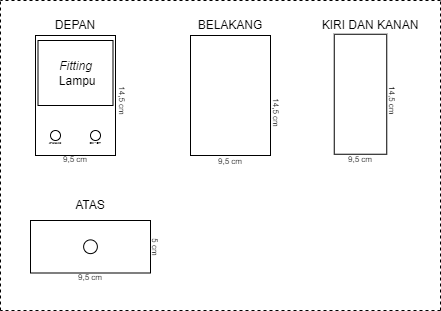
##### Tabel 5.1 Konfigurasi Pin

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin ESP32** | **Pin Komponen** | **Warna** | **Komponen** |
| VIN | VCC | Coklat | 2 Relay Module |
| GND | GND | Hitam |
| GPIO26 | IN1 | Merah |
| GPIO25 | IN2 | Oranye |
| 3.3V | VCC | Ungu | RTC |
| GND | GND | Putih |
| GPIO21 | SDA | Hijau |
| GPIO22 | SCL | Kuning |
| GPIO32 | GND | Putih | Button Node |
| VCC | VCC | Kuning |

|  |  |  |  |
| --- | --- | --- | --- |
| GND | GND | Hitam | Button *On - Off* |
| VCC | VCC | Kuning |

Tabel 5.1 berisikan daftar seluruh *wiring* untuk menjembatani dari setiap komponen-komponen dalam system dengan menggunakan kabel jumper. Kaberl jumper yang digunakan memeliki macam variasi berdasarkan kebutuhan dari jarak antara pin pada ESP32 dengan pin dari setiap komponen.

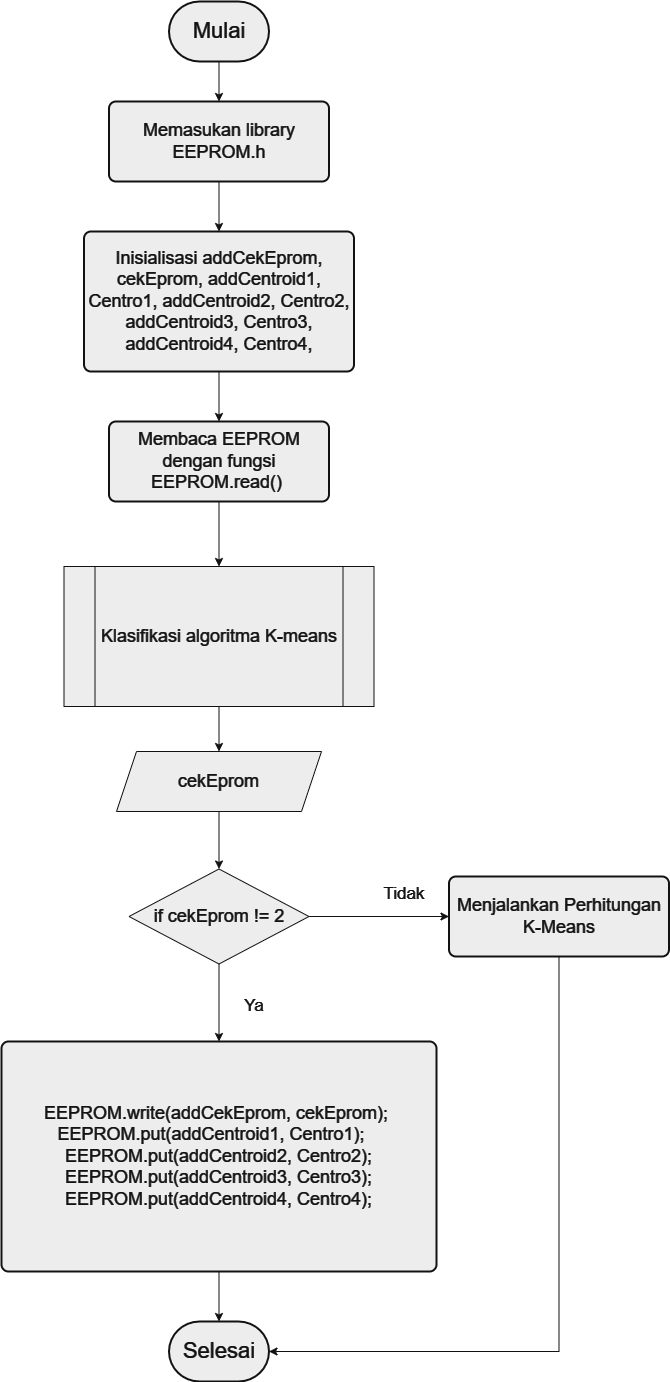
###### Perancangan Cover

Penulis juga melakukan perancangan *cover* atau *casing* yang akan digunakan untuk melindungi sistem. *Cover* atau *casing* juga berfungsi agar sistem dapat dengan mudah dibawa kemana-mana (portabel). Adapun rancangan dari *cover* atau *casing* dapat dilihat pada Gambar 5.3 dibawah ini.

##### Gambar 5.3 Perancangan *Cover* Sistem

###### Perancangan Penyimpanan EEPROM

Perancangan selanjutnya terdapat perancangan EEPROM pada ESP32 yang di tampilkan melalui Gambar 5.4



##### Gambar 5.4 Diagram Alir Penyimpanan EEPROM

Gambar 5.4 terdapat alur penyimpanan data pada EEPROM ESP32. Penulis menggunakan 4 buah alamat pada EEPROM untuk menyimpan data. Adapun Penjabaran dari penggunaan alamat EEPROM ini sebagai berikut:

* + - * 1. Alamat addCentroid1 pada EEPROM digunakan untuk menyimpan data Centroid 1.
        2. Alamat addCentroid2 pada EEPROM digunakan untuk menyimpan data Centroid 2.
        3. Alamat addCentroid3 pada EEPROM digunakan untuk menyimpan data Centroid 3.
        4. Alamat addCentroid4 pada EEPROM digunakan untuk menyimpan data Centroid 4.

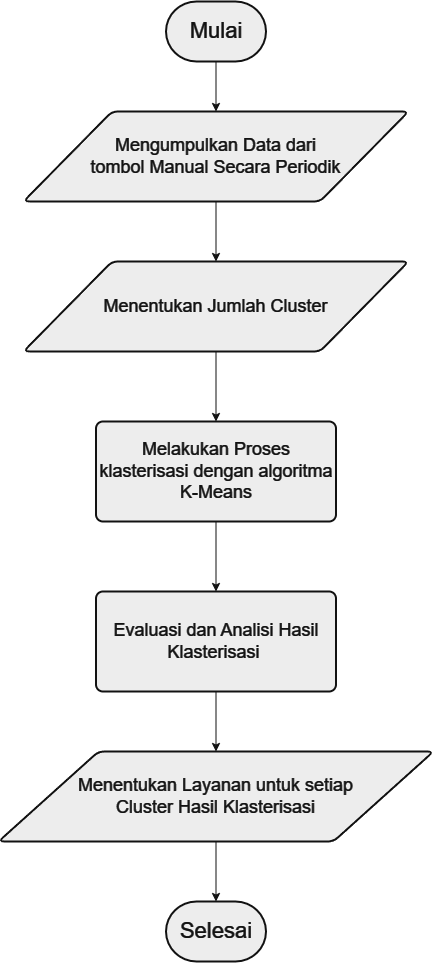
### Perancangan Pelangkat Lunak

Perangkat lunak merupakan sebuah komponen penting dalam sistem untuk menentukan dan mengontrol perangkat keras agar bekerja sesuai dengan tujuan dan dapat berkomunikasi antar komponen dalam sistem. Perancangan perangkat lunak akan membahas mengenai alur kerja pemrograman sistem hingga bagaimana komunikasi dalam sistem bekerja.

Pada perancangan perangkat lunak, dalam sistem ini secara garis besar dibagi menjadi tiga fase yaitu akuisisi data sensor, pengiriman data/penerimaan data dan pengolahan data. Ketiga fase tersebut diterapkan pada masing-masing node. Perancangan pada node transmitter diperlukan agar dapat mengakuisi data dari sensor dan mengirimkan datanya melalui modul MCP2515 dengan cara wired/kabel. Sedangkan perancangan perangkat lunak pada receiver diperlukan agar node receiver dapat menerima semua data dari node yang dikirim oleh transmitter secara wired dan mampu menampilkan data tersebut menggunakan serial monitor dan tertampil di LCD. Serta, dapat memberikan peringatan kepada user apabila terdapat kondisi bahaya.

###### Perancangan Diagram Alir Pembuatan Dataset

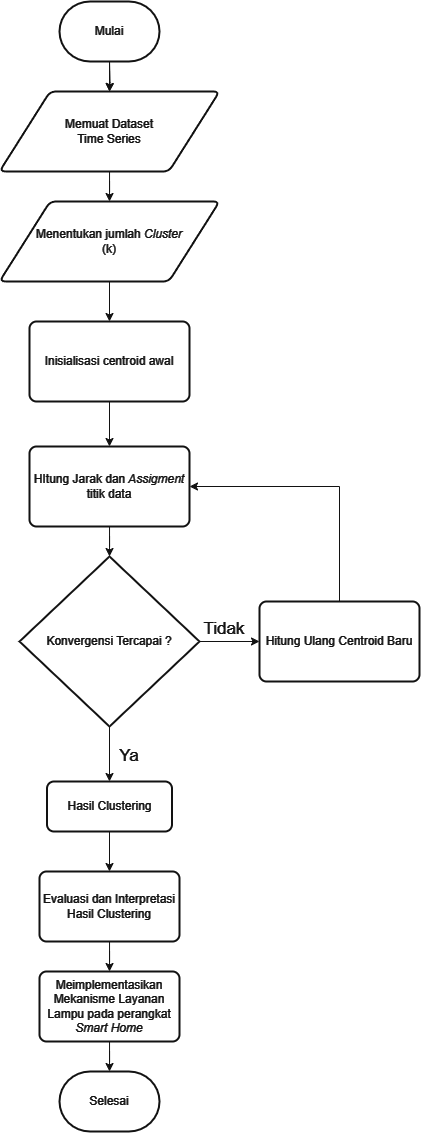
Perancangan diagram alir pembuatan dataset berguna sebab data yang penulis gunakan merupakan data primer. Perancangan diagram alir pembuatan dataset akan dijelaskan melalui Gambar 5.4.



**Gambar 5.5 Diagram Alir Pembuatan Dataset**

###### Perancangan Diagram Alir Keseluruhan Sistem

Perancangan terakhir yaitu perancangan diagram alir keseluruhan sistem. Perancangan ini dibuat agar penulis dapat menjelaskan lebih rinci terkait logika dan fitur-fitur yang dapat dilakukan oleh sistem. Perancangan diagram alir keseluruhan sistem dapat dilihat pada Gambar 5.5

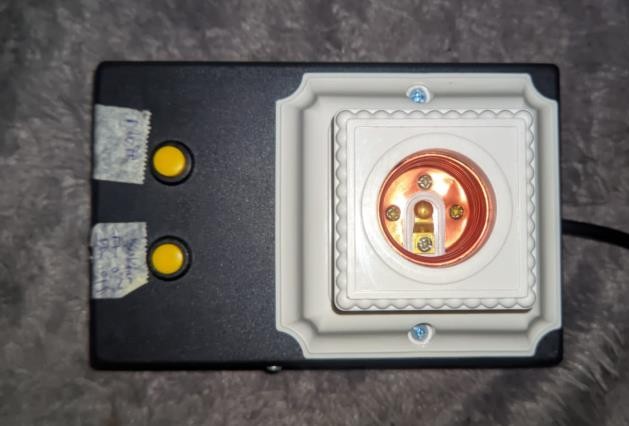


**Gambar 5.6 Diagram Alir Keseluruhan Sistem**

## Implementasi Sistem

Pada bagian implementasi, akan dijelaskan lebih mendalam mengenai proses implementasi perangkat keras dan perangkat lunak pada sistem yang sudah dirancang sebelumnya pada sub bab perancangan

### Implementasi Perangkat Keras

Pada implementasi perangkat keras, dilakukan pemasangan seluruh komponen sesuai dengan skema yang sudah tertera pada sub bab perancangan sebelumnya. Setelah itu dilakukan pembuatan *cover* atau *casing* yang bertujuan untuk melindungi komponen-komponen sistem. Adapun tampilan dari *cover* atau *casing* tersebut dapat dilihat pada gambar dibawah ini

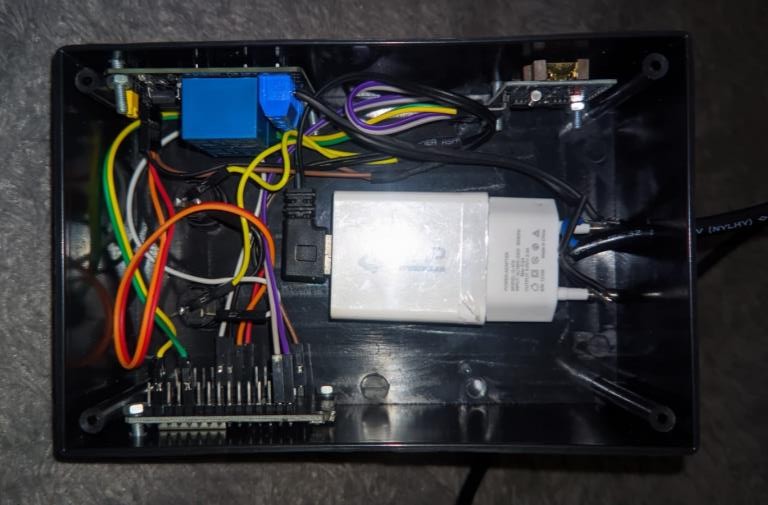
##### Gambar 5.7 Sistem Tampak Depan



**Gambar 5.8 Sistem Tampak Belakang**



##### Gambar 5.9 Sistem Tampak Samping



**Gambar 5.10 Sistem Tampak Dalam**

### Implementasi Perangkat Lunak

Pada bagian implementasi perangkat lunak merupakan kumpulan kode program yang terintegrasi sesuai dengan perancangan pada diagram alir perangkat lunak sebelumnya

##### Program Impor library yang dibutuhkan

|  |  |
| --- | --- |
| Program 5.2.2.1: Impor library yang dibutuhkan | |
| 1  2 | #include <RTClib.h>  #include <EEPROM.h> |

Penjelasan Kode Program:

* + - * + Kode baris 1 memuat library RTClib.h
        + Kode baris 2 memuat library EEPROM.h

##### Program menyimpan dan mengakses data di dalam EEPROM

|  |  |
| --- | --- |
| Program 5.2.2.2: Program menyimpan dan mengakses data di dalam EEPROM | |
| 4 | int addCekEprom = 0; |
| 5 | int addCentroid1 = 1; |
| 6 | int addCentroid2 = 20; |
| 7 | int addCentroid3 = 30; |
| 8 | int addCentroid4 = 40; |
| 9 | int adddatake = 47; |
| 10 | int idxDttrain = 0; |
| 11 |  |
| 12 | int Centro1, Centro2, Centro3, Centro4, cekEprom; |

Penjelasan Kode Program:

* + - * + Kode baris 4 memuat untuk menyimpan alamat (*address*) di dalam EEPROM tempat Anda ingin menyimpan nilai ‘cekEprom’.
        + Kode baris 6 – 8 memuat untuk menyimpan alamat-alamat di dalam EEPROM tempat Anda ingin menyimpan nilai-nilai centroid (pusat cluster) yang berkaitan dengan aplikasi atau algoritma.
        + Kode baris 10 memuat untuk ‘Centro1’, ‘Centro2’, ‘Centro3’, ‘Centro4’ Variabel-variabel ini digunakan untuk menyimpan nilai-nilai centroid yang dapat digunakan dalam perhitungan atau operasi lainnya. Sedangkan ‘cekEprom’, digunakan untuk menyimpan nilai yang berkaitan dengan pengecekan atau kondisi tertentu.

##### Program Untuk menginisialisasi variabel dengan nilai yang sesuai dan melakukan operasi

|  |  |
| --- | --- |
| Program 5.2.2.3: Program Untuk menginisialisasi variabel dengan nilai yang sesuai dan melakukan operasi | |
| 14 | int dttrainEprom[60] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
| 15 | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
| 16 | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
|  | 0, 0 }; |
| 17 |  |
| 18 | int dttrainyEprom[60] = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
| 19 | 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
| 20 | 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
|  | 1, 0 }; |
| 21 |  |
| 22 |  |
| 23 | int addDttrain[60] = { 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, |
| 24 | 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, |
| 25 | 170, 175, 180, 185, 190, 195, 200, 205, 210, 215, 220, 225, 230, |
| 26 | 235, 240, 245, 250, 255, 260, 265, 270, 275, 280, 285, 290, 295, |
| 27 | 300, 305, 310, 315, 320, 325, 330, 335, 340, 345 }; |
| 28 |  |
| 29 | int dttrain[52] = { 325, 420, 1029, 1373, 321, 423, 1020, 1380, |
| 30 | 300, 401, 1042, 1362, 324, 410, 1083, 1402, 299, 400, 1030, 1333, |
| 31 | 285, 360, 1019, 1398, 287, 382, 1023, 1327, 312, 378, 993, 1392, |
| 32 | 315, 393, 1019, 1320, 342, 392, 1054, 1381, 292, 400, 1006, 1262, |
| 33 | 300, 378, 1049, 1380, 330, 376, 997, 1379 }; |
| 34 | int dttrainY[52] = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, |
| 35 | 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, |
| 36 | 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0 }; |
| 37 | int centrox[4] = { 325, 380, 1029, 1373 }; |
| 38 | int centroy[4] = { 1, 0, 1, 0 }; |
| 39 | int nil, k = 10; |
| 40 | String hsaw = "111111111111111111111"; |
| 41 | String hsSbl, hsSkr; |

Penjelasan Kode Program:

* + - * + Kode baris 14 - 16 memuat untuk menyimpan data train yang akan disimpan di EEPROM.
        + Kode baris 17 - 20 memuat untuk menyimpan label atau kategori yang terkait dengan setiap data train dalam 'dttrainEprom'.
        + Kode baris 23 - 27 memuat array yang menyimpan alamat (address) di EEPROM untuk setiap data train yang disimpan.
        + Kode baris 29 - 33 memuat array yang digunakan untuk menyimpan data train dalam bentuk waktu dalam satuan menit.
        + Kode baris 34 - 36 memuat array yang digunakan untuk menyimpan label atau kategori yang terkait dengan setiap data train dalam 'dttrain'.
        + Kode baris 37 – 38 memuat untuk menyimpan nilai centroid pada sumbu x dan y dalam perhitungan K-Means.
        + Kode baris 39 memuat untuk menyimpan nilai waktu saat ini dalam bentuk menit.
        + Kode baris 40 memuat untuk pengecekan dan perbandingan.
        + Kode baris 41 memuat untuk menyimpan hasil perhitungan klasterisasi.

##### Program Untuk Mendefinisikan Pin

|  |  |
| --- | --- |
| Program 5.2.2.4: Program Untuk Mendefinisikan Pin | |
| 43 | unsigned long pmTombol, pmRtc, pmCek; |
| 44 | #define tModePin 32 |
| 45 | #define tSaklarPin 33 |
| 46 | #define lampuPin 25 |
| 47 | #define lampu2Pin 26 |
| 48 | bool stAuto, stLmp, Saklar, stLmpSbl, blmSimpanEprom, blmHitung; |
| 49 | int ctrck; |
| 50 |  |
| 51 | RTC\_DS1307 rtc; |
| 52 | int jam, menit; |

Penjelasan Kode Program:

* + - * + Kode baris 43 memuat untuk menyimpan waktu yang diukur dalam milisecond (ms) atau sebagai tanda waktu tertentu dalam program.
        + Kode baris 44 - 47 memuat untuk mendefinisikan konstanta.
        + Kode baris 48 memuat untuk menyimpan status atau kondisi tertentu dalam program.
        + Kode baris 33 memuat untuk mengakses atau mengontrol modul RTC\_DS1307.
        + Kode baris 34 memuat untuk menyimpan nilai jam dan menit dalam waktu

*real-time* yang diperoleh dari modul RTC.

##### Program Untuk memeriksa dan memanipulasi data yang disimpan di dalam EEPROM

|  |  |
| --- | --- |
| Program 5.2.2.5: Program untuk memeriksa dan memanipulasi data yang disimpan di dalam EEPROM | |
| 54 | void cekEEPROM() { |
| 55 | EEPROM.begin(512); |
| 56 | cekEprom = EEPROM.read(addCekEprom); |
| 57 | if (cekEprom != 2) { |
| 58 | hitungKmeans(); |
| 59 | cekEprom = 2; |
| 60 | EEPROM.write(addCekEprom, cekEprom); |
| 61 | EEPROM.write(adddatake, idxDttrain); |
| 62 | EEPROM.put(addCentroid1, Centro1); |
| 63 | EEPROM.put(addCentroid2, Centro2); |
| 64 | EEPROM.put(addCentroid3, Centro3); |
| 65 | EEPROM.put(addCentroid4, Centro4); |
| 66 |  |
| 67 | EEPROM.commit(); |

|  |  |
| --- | --- |
| 68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | }  EEPROM.get(addCentroid1, Centro1); EEPROM.get(addCentroid2, Centro2); EEPROM.get(addCentroid3, Centro3); EEPROM.get(addCentroid4, Centro4); idxDttrain = EEPROM.read(adddatake); delay(500);  for (int idxa = 0; idxa < 60; idxa++) { int al = addDttrain[idxa]; EEPROM.get(al, dttrainEprom[idxa]); al = 400 + idxa;  dttrainyEprom[idxa] = EEPROM.read(al); delay(50);  }  } |

Penjelasan Kode Program:

* + - * + Kode baris 55 memuat untuk menginisialisasi EEPROM dengan ukuran 512 byte.
        + Kode baris 56 memuat untuk membaca nilai yang tersimpan di alamat 'addCekEprom'.
        + Kode baris 57 - 68 memuat pengecekan jika nilai 'cekEprom' tidak sama dengan 2. Jika tidak sama, maka fungsi 'hitungKmeans()' dipanggil untuk melakukan perhitungan K-Means. Setelah itu, nilai 'cekEprom' diubah menjadi 2 dan disimpan kembali ke alamat 'addCekEprom' pada EEPROM. Selanjutnya, nilai 'idxDttrain' juga disimpan di alamat 'adddatak'e pada EEPROM. Seluruh nilai centroid (Centro1, Centro2, Centro3, Centro4) juga disimpan ke EEPROM menggunakan fungsi 'EEPROM.put()'. Akhirnya, perubahan-perubahan tersebut disimpan di EEPROM menggunakan fungsi 'EEPROM.commit()'.
        + Kode baris 69 - 73 memuat pembacaan nilai centroid dari EEPROM menggunakan fungsi 'EEPROM.get()', dan nilai-nilai tersebut disimpan di variabel Centro1, Centro2, Centro3, dan Centro4. Nilai 'idxDttrain' juga dibaca dari alamat 'adddatake' pada EEPROM.
        + Kode baris 76 - 81 memuat untuk membaca data train 'dttrainEprom' dan data target 'dttrainyEprom' dari alamat EEPROM yang sesuai dengan indeks idxa. Pembacaan dilakukan menggunakan fungsi 'EEPROM.get()' dan 'EEPROM.read()', dan nilai-nilai tersebut disimpan di array 'dttrainEprom' dan 'dttrainyEprom'.

##### Program membagi string berdasarkan indeks

|  |  |
| --- | --- |
| Program 5.2.2.6: Program membagi string berdasarkan indeks | |
| 87  88  89  90  91  92  93  94 | String seplit(String data, char separator, int index) { int found = 0;  int strIndex[] = { 0, -1 };  int maxIndex = data.length() - 1;  for (int i = 0; i <= maxIndex && found <= index; i++) { if (data.charAt(i) == separator || i == maxIndex) {  found++;  strIndex[0] = strIndex[1] + 1; |

|  |  |
| --- | --- |
| 95  96  97  98  99 | strIndex[1] = (i == maxIndex) ? i + 1 : i;  }  }  return found > index ? data.substring(strIndex[0], strIndex[1])  : "";  } |

Penjelasan Kode Program:

* + - * + Kode baris 87 memuat untuk deklarasi fungsi seplit() yang mengambil tiga parameter: data (string yang akan dipisahkan), separator (karakter pemisah), dan index (indeks potongan yang diinginkan).
        + Kode baris 88 - 90 memuat inisialisasi variabel-variabel yang digunakan dalam fungsi. 'found' akan menghitung jumlah pemisah yang ditemukan, 'strIndex' adalah array yang akan menyimpan indeks awal dan akhir setiap potongan yang dipisahkan, dan 'maxIndex' menyimpan indeks maksimum dalam string 'data'.
        + Kode baris 91 - 97 memuat Loop 'for' untuk iterasi melalui setiap karakter dalam string data, jika karakter saat ini sama dengan 'separator' atau sudah mencapai karakter terakhir ('i == maxIndex'), maka itu menandakan pemisah telah ditemukan. Variabel 'found' akan ditambah 1 untuk menghitung jumlah pemisah yang ditemukan dan indeks awal maupun akhir potongan yang dipisahkan disimpan dalam 'strIndex' menggunakan 'strIndex[0]' dan 'strIndex[1]' secara berurutan.
        + Kode baris 98 memuat untuk fungsi mengembalikan potongan yang dipisahkan dengan menggunakan 'data.substring(strIndex[0], strIndex[1])' jika jumlah pemisah yang ditemukan 'found' lebih besar dari index. Jika tidak, fungsi akan mengembalikan string kosong ("").

##### Program mengatur mode pin pada mikrokontroler

|  |  |
| --- | --- |
| Program 5.2.2.7: Program mengatur mode pin pada mikrokontroler | |
| 101  102  103  104  105  106 | void setpin() {  pinMode(tModePin, INPUT\_PULLUP); pinMode(tSaklarPin, INPUT\_PULLUP); pinMode(lampuPin, OUTPUT); pinMode(lampu2Pin, OUTPUT);  } |

Penjelasan Kode Program:

* + - * + Kode baris 101 memuat untuk mengatur mode pin pada mikrokontroler.
        + Kode baris 102 memuat untuk membaca input dan diaktifkan dengan resistor pull-up internal, sehingga keadaan awalnya adalah HIGH.
        + Kode baris 103 memuat untuk mengaktifkan resistor pull-up internal pada pin tersebut.
        + Kode baris 104 memuat untuk mengendalikan lampu yang terhubung ke pin tersebut dengan memberikan sinyal output.
        + Kode baris 105 memuat untuk pin output.

##### Program perhitungan untuk mengubah nilai waktu

|  |  |
| --- | --- |
| Program 5.2.2.8: Program perhitungan untuk mengubah nilai waktu | |
| 108  109  110  111  112  113  114  115  116  117  118 | void simpanDtTrain() {  Serial.println("Simpan data train ke eeprom"); int nilj = (jam \* 60) + menit; EEPROM.put(addDttrain[idxDttrain], nilj); EEPROM.write((400 + idxDttrain), stLmp); dttrainEprom[idxDttrain] = nilj; idxDttrain++;  if (idxDttrain > 59) idxDttrain = 0; EEPROM.write(adddatake, idxDttrain); EEPROM.commit();  } |

Penjelasan Kode Program:

* + - * + Kode baris 110 memuat perhitungan untuk mengubah nilai waktu (jam dan menit) menjadi nilai dalam satuan menit.
        + Kode baris 111 memuat fungsi 'EEPROM.put()', nilai nilj disimpan di alamat EEPROM yang ditentukan oleh 'addDttrain[idxDttrain]'. Ini berarti data train pada indeks 'idxDttrain' disimpan ke EEPROM.
        + Kode baris 112 memuat fungsi 'EEPROM.write()', nilai 'stLmp' (status lampu) disimpan di alamat EEPROM yang ditentukan oleh '(400 + idxDttrain)'. Ini berarti status lampu pada indeks 'idxDttrain' disimpan ke EEPROM.
        + Kode baris 113 - 114 memuat Nilai ‘idxDttrain’ ditingkatkan sebesar 1, dan jika nilai ‘idxDttrain’ melebihi 59, maka nilai ‘idxDttrain’ diatur kembali ke

0. Hal ini dilakukan untuk mengatur indeks penyimpanan data train di EEPROM.

* + - * + Kode baris 115 – 117 memuat nilai ‘idxDttrain’ disimpan di alamat EEPROM yang ditentukan oleh ‘adddatake’ menggunakan fungsi ‘EEPROM.write()’. Perubahan-perubahan tersebut kemudian disimpan di EEPROM menggunakan fungsi ‘EEPROM.commit()’.

##### Program membaca status atau nilai dari tombol yang terhubung ke pin

|  |  |
| --- | --- |
| Program 5.2.2.9: Program membaca status atau nilai dari tombol yang terhubung ke pin | |
| 120  121  122  123  124  125  126  127  128  129  130  131  132  133  134 | void bacaTombol() {  if (millis() - pmTombol > 100) { stAuto = digitalRead(tModePin); stLmp = digitalRead(tSaklarPin); if (!stAuto) {  if (stLmpSbl == stLmp) { Serial.print("sama"); if (blmSimpanEprom) {  ctrck++; Serial.println(ctrck); if (ctrck > 600) {  simpanDtTrain(); blmSimpanEprom = false;  }  } |

|  |  |
| --- | --- |
| 135  136  137  138  139  140  141  142  143  144 | } else {  blmSimpanEprom = true; stLmpSbl = stLmp; ctrck = 0;  }  }  pmTombol = millis();  }  } |

Penjelasan Kode Program:

* + - * + Kode baris 121 memuat pengecekan apakah sudah melewati interval waktu 100 milidetik sejak pemanggilan terakhir fungsi ini.
        + Kode baris 122 - 123 memuat status tombol pada pin tModePin dan pin tSaklarPin dibaca dan disimpan dalam variabel stAuto dan stLmp masing- masing.
        + Kode baris 124 - 134 memuat pengecekan apakah mode otomatis (stAuto bernilai false). Jika iya, maka lampu tidak dikendalikan oleh tombol dan tidak ada operasi tambahan yang dilakukan. Perintah-perintah setelahnya akan diabaikan. Jika mode tidak otomatis (stAuto bernilai true), maka dilakukan pengecekan status lampu saat ini dengan status lampu sebelumnya (stLmpSbl). Jika statusnya sama, maka berarti tidak ada perubahan pada lampu. Jika status lampu saat ini (stLmp) sama dengan status lampu sebelumnya (stLmpSbl) dan belum dilakukan penyimpanan data train ke EEPROM (blmSimpanEprom), maka dilakukan peningkatan nilai ctrck sebesar 1. Nilai ctrck digunakan untuk menghitung berapa lama lampu dalam keadaan sama (tidak berubah).Jika nilai ctrck melebihi 600 (artinya lampu dalam keadaan sama selama 60 detik), maka fungsi simpanDtTrain() dipanggil untuk menyimpan data train ke EEPROM. Setelah itu, variabel blmSimpanEprom diubah menjadi false agar tidak terjadi penyimpanan berulang
        + Kode baris 135 - 139 memuat status lampu saat ini (stLmp) berbeda dengan status lampu sebelumnya (stLmpSbl), maka dianggap terjadi perubahan pada lampu. Variabel blmSimpanEprom diubah menjadi true untuk menandakan bahwa perubahan terjadi dan data train perlu disimpan ke EEPROM. Status lampu sebelumnya (stLmpSbl) diperbarui dengan status lampu saat ini (stLmp), dan nilai ctrck diatur kembali menjadi 0.
        + Kode baris 82 memuat nilai pmTombol diupdate dengan nilai waktu saat ini menggunakan millis().
      1. **Program untuk menginisialisasi dan memeriksa modul RTC (*Real-Time Clock*)**

|  |  |
| --- | --- |
| Program 5.2.2.10: Program untuk menginisialisasi dan memeriksa modul RTC (*Real-Time Clock*) | |
| 146  147  148  149 | void RTCAwal() {  if (!rtc.begin()) { Serial.println("Couldn't find RTC"); Serial.flush(); |

|  |  |
| --- | --- |
| 150  151  152  153  154  155  156  157  158  159 | while (1) delay(10);  }  if (!rtc.isrunning()) {  Serial.println("RTC Abis OFF! setting waktu"); rtc.adjust(DateTime(F( DATE ), F( TIME )));  }  // rtc.adjust(DateTime(F( DATE ), F( TIME )));  // rtc.adjust(DateTime(2023, 7, 8, 22, 38, 0));  } |

Penjelasan Kode Program:

* + - * + Kode baris 146 memuat untuk menginisialisasi dan memeriksa modul RTC (*Real-Time Clock*).
        + Kode baris 147 – 150 memuat kondisi yang memeriksa apakah modul RTC dapat ditemukan atau diakses. Jika modul RTC tidak dapat ditemukan, maka program akan mencetak pesan "*Couldn't find RTC*" ke Serial Monitor dan menunda eksekusi dengan ‘delay(10)’.
        + Kode baris 153 memuat untuk memeriksa apakah modul RTC sedang berjalan atau tidak. Jika RTC tidak berjalan, berarti daya baterai atau sumber daya lainnya telah habis atau terputus.
        + Kode baris 154 memuat jika RTC tidak berjalan, maka pesan "RTC Abis OFF! setting waktu" akan dicetak ke Serial Monitor untuk memberi tahu bahwa RTC tidak berfungsi dan perlu diatur waktu.
        + Kode baris 155 – 157 memuat untuk mengatur waktu RTC dengan menggunakan waktu dari komputer saat ini.
        + Kode baris 158 memuat untuk mengatur waktu RTC secara manual.

##### Program untuk Membaca dan Memperbaharui Waktu RTC secara periodik

|  |  |
| --- | --- |
| Program 5.2.2.11: Program Membaca dan Memperbaharui Waktu RTC secara periodik | |
| 161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182 | void bacaRtc() {  if (millis() - pmRtc >= 1000) { nil = 0;  DateTime now = rtc.now(); jam = now.hour();  menit = now.minute(); Serial.print("Waktu : "); Serial.print(jam); Serial.print(":"); Serial.println(menit); nil += jam \* 60;  nil += menit; int sk;  int clus = 0;  if (nil < Centro1) { sk = centroy[3]; clus = 4;  } else if (nil < Centro2) { sk = centroy[0];  clus = 1;  } else if (nil < Centro3) { sk = centroy[1]; |

|  |  |
| --- | --- |
| 183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199 | clus = 2;  } else if (nil < Centro4) { sk = centroy[2];  clus = 3;  } else {  sk = centroy[3]; clus = 4;  }  if (sk == 0) Saklar = 1; else Saklar = 0;  if (stAuto) { Serial.print("Cluster : "); Serial.println(clus);  }  pmRtc = millis();  }  } |

Penjelasan Kode Program:

* + - * + Kode baris 161 memuat untuk membaca waktu dari modul RTC dan melakukan clusterisasi.
        + Kode baris 162 memuat kondisi yang memeriksa apakah telah berlalu 1 detik (1000 milisecond) sejak waktu terakhir membaca RTC 'pmRtc'.
        + Kode baris 163 memuat variabel nil diatur kembali menjadi 0 sebelum melakukan perhitungan waktu.
        + Kode baris 164 memuat waktu saat ini dari objek RTC dan menyimpannya dalam variabel 'now' dengan tipe data 'DateTime'.
        + Kode baris 165 - 166 memuat nilai jam dan menit dari waktu yang didapatkan dari RTC dan menyimpannya dalam variabel 'jam' dan 'menit' masing-masing.
        + Kode baris 167 - 170 memuat untuk mencetak waktu yang dibaca dari RTC ke Serial Monitor.
        + Kode baris 171 memuat konversi nilai jam menjadi menit dan menambahkannya ke variabel 'nil'. Ini dilakukan untuk melakukan perhitungan seluruh waktu dalam satuan menit.
        + Kode baris 172 memuat untuk menambahkan nilai menit ke variabel ‘nil’.
        + Kode baris 173 - 190 memuat untuk menentukan nilai 'sk' (nilai untuk pengendalian saklar) dan 'clus' (nilai untuk kluster) berdasarkan nilai nil yang telah dihitung sebelumnya dan nilai-nilai Centro yang telah dibaca dari EEPROM.
        + Kode baris 191 - 196 memuat pengecekan nilai sk. Jika nilainya 0, berarti lampu harus dinyalakan (Saklar diatur sebagai 1). Jika nilainya bukan 0, berarti lampu harus dimatikan (Saklar diatur sebagai 0). Jika mode otomatis (stAuto bernilai true), maka ditampilkan informasi mengenai cluster yang dipilih dengan mencetak nilai clus.
        + Kode baris 197 memuat perbarui dengan waktu saat ini 'millis()', sehingga waktu terakhir membaca RTC diupdate.

##### Program implementasi algoritma K-means untuk melakukan clustering

|  |  |
| --- | --- |
| Program 5.2.2.12: Program implementasi algoritma K-means untuk melakukan clustering | |
| 201 | void hitungKmeans() { |
| 202 | hsSbl = hsaw; |
| 203 | hsSkr = ""; |
| 204 | float cc1 = centrox[0]; |
| 205 | float cc2 = centrox[1]; |
| 206 | float cc3 = centrox[2]; |
| 207 | float cc4 = centrox[3]; |
| 208 |  |
| 209 | bool sama = false; |
| 210 | String hs = "0"; |
| 211 |  |
| 212 | int cndtrin = sizeof(dttrain) / sizeof(int); |
| 213 |  |
| 214 | while (!sama) { |
| 215 | Serial.print(cc1); |
| 216 | Serial.print("\t"); |
| 217 | Serial.print(cc2); |
| 218 | Serial.print("\t"); |
| 219 | Serial.print(cc3); |
| 220 | Serial.print("\t"); |
| 221 | Serial.println(cc4); |
| 222 | hsSkr = ""; |
| 223 | float tempc1 = 0; |
| 224 | float tempc2 = 0; |
| 225 | float tempc3 = 0; |
| 226 | float tempc4 = 0; |
| 227 |  |
| 228 | int ctr1 = 0; |
| 229 | int ctr2 = 0; |
| 230 | int ctr3 = 0; |
| 231 | int ctr4 = 0; |
| 232 | bool adanilai = true; |
| 233 | for (int aa = 0; aa < 111; aa++) { |
| 234 | float aa1, aa2, aa3, aa4; |
| 235 | if (aa < 52) { |
| 236 | aa1 = sqrt(sq(dttrain[aa] - cc1) + sq(dttrainY[aa] - |
|  | centroy[0])); |
| 237 | aa2 = sqrt(sq(dttrain[aa] - cc2) + sq(dttrainY[aa] - |
|  | centroy[1])); |
| 238 | aa3 = sqrt(sq(dttrain[aa] - cc3) + sq(dttrainY[aa] - |
|  | centroy[2])); |
| 239 | aa4 = sqrt(sq(dttrain[aa] - cc4) + sq(dttrainY[aa] -  centroy[3])); |
| 240 | } else { |
| 241 | int nildteprom = dttrainEprom[aa - 52]; |
| 242 | if (nildteprom > 0) { |
| 243 | aa1 = sqrt(sq(nildteprom - cc1) + sq(dttrainyEprom[aa - |
|  | 52] - centroy[0])); |
| 244 | aa2 = sqrt(sq(nildteprom - cc2) + sq(dttrainyEprom[aa - |
|  | 52] - centroy[1])); |
| 245 | aa3 = sqrt(sq(nildteprom - cc3) + sq(dttrainyEprom[aa - |
|  | 52] - centroy[2])); |
| 246 | aa4 = sqrt(sq(nildteprom - cc4) + sq(dttrainyEprom[aa - |
|  | 52] - centroy[3])); |
| 247 | } else { |
| 248 | adanilai = false; |
| 249 | } |
| 250 | } |

|  |  |
| --- | --- |
| 251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302 | if (adanilai) {  int pos = crpos(aa1, aa2, aa3, aa4);  hs = String(pos); switch (pos) {  case 0:  tempc1 += dttrain[aa]; ctr1++;  break; case 1:  tempc2 += dttrain[aa]; ctr2++;  break; case 2:  tempc3 += dttrain[aa]; ctr3++;  break; case 3:  tempc4 += dttrain[aa]; ctr4++;  break;  }  hsSkr += hs;  }  }  if (hsSbl == hsSkr) { sama = true; Centro1 = cc1; Centro2 = cc2; Centro3 = cc3; Centro4 = cc4;  } else {  hsSbl = hsSkr;  cc1 = tempc1 / ctr1; cc2 = tempc2 / ctr2; cc3 = tempc3 / ctr3; cc4 = tempc4 / ctr4;  hsSkr = "";  tempc1 = tempc2 = tempc3 = tempc4 = 0; ctr1 = ctr2 = ctr3 = ctr4 = 0;  }  }  Saklar = centroy[hs.toInt()]; Serial.print("hasilnya : "); Serial.println(hs);  } |

Penjelasan Kode Program:

* + - * + Kode baris 202 - 203 memuat string berisi angka 1 dengan panjang 21 dan 'hsSkr' diinisialisasi sebagai string kosong.
        + Kode baris 204 - 207 memuat inisialisasi dengan nilai centroid awal dari array centrox.
        + Kode baris 209 – 210 memuat variabel ‘sama’ diinisialisasi sebagai false. Variabel hs diinisialisasi dengan string "0".
        + Kode baris 212 memuat Variabel 'cndtrin' dihitung sebagai jumlah elemen dalam array 'dttrain'.
        + Kode baris 214 memuat perulangan while hingga kondisi sama menjadi true. Perulangan ini digunakan untuk melakukan iterasi pada algoritma K- Means hingga konvergensi.
        + Kode baris 215 – 221 memuat nilai centroid saat ini (cc1, cc2, cc3, dan cc4) dicetak melalui Serial Monitor.
        + Kode baris 222 – 226 memuat Variabel hsSkr diatur sebagai string kosong. Variabel tempc1, tempc2, tempc3, dan tempc4 diatur sebagai 0.
        + Kode baris 228 – 232 memuat variabel ctr1, ctr2, ctr3, dan ctr4 diatur sebagai 0. Variabel adanilai diatur sebagai true.
        + Kode baris 233 – 239 memuat perulangan for dari 0 hingga 110 untuk memproses setiap elemen data train (dttrain dan dttrainEprom).
        + Kode baris 241 - 246 memuat hasil perhitungan disimpan dalam variabel aa1, aa2, aa3, dan aa4.
        + Kode baris 248 memuat pengecekan kondisi adanilai digunakan untuk memastikan bahwa data train memiliki nilai yang valid. Jika nilai nildteprom lebih besar dari 0, maka dianggap sebagai data train valid. Jika tidak, variabel adanilai diatur sebagai false.
        + Kode baris 252 - 273 memuat jika data train valid, dilakukan pemilihan cluster berdasarkan jarak terdekat dengan menggunakan fungsi crpos(). Hasil pemilihan cluster disimpan dalam variabel pos. Variabel hs diatur sebagai string yang mengandung nilai pos. Menggunakan pernyataan switch-case, data train ditambahkan ke total centroid sesuai dengan cluster yang dipilih.
        + Kode baris 274 memuat variabel hsSkr ditambahkan dengan string hs yang berisi cluster hasil pemilihan pada setiap iterasi.
        + Kode baris 280 - 297 memuat pengecekan apakah hasil cluster pada iterasi saat ini sama dengan hasil cluster pada iterasi sebelumnya. Jika sama, maka iterasi dihentikan dan centroid akhir (Centro1, Centro2, Centro3, dan Centro4) diatur sesuai dengan nilai centroid terakhir. Jika tidak sama, iterasi dilanjutkan dengan memperbarui nilai centroid dengan rata-rata dari data train dalam masing-masing cluster.
        + Kode baris 299 - 301 memuat variabel Saklar diatur berdasarkan nilai centroy yang sesuai dengan cluster hasil pemilihan terakhir. Hasil cluster (hs) dicetak melalui Serial Monitor.

##### Program pengurutan dan pemilihan posisi terkait dengan perhitungan jarak Euclidean pada algoritma K-means

|  |  |
| --- | --- |
| Program 5.2.2.13: Program pengurutan dan pemilihan posisi terkait dengan perhitungan jarak Euclidean pada algoritma K-means | |
| 304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324 | int crpos(float a1, float a2, float a3, float a4) { float aarr[10] = { a1, a2, a3, a4 };  int arRet[10] = { 0, 1, 2, 3 }; int cnAr = 3;  for (int a = 0; a < cnAr; a++) { for (int b = 0; b < cnAr; b++) {  int c = b + 1;  float nilA = aarr[c]; float nilB = aarr[b]; int pos1 = arRet[c]; int pos2 = arRet[b]; if (nilA < nilB) {  aarr[c] = nilB; aarr[b] = nilA; arRet[c] = pos2; arRet[b] = pos1;  }  }  }  return arRet[0];  } |

Penjelasan Kode Program:

* + - * + Kode baris 304 memuat pengurutan dan pemilihan posisi terkait dengan perhitungan jarak Euclidean pada algoritma K-means.
        + Kode baris 305 - 306 memuat inisialisasi array 'aarr' dengan nilai-nilai 'a1', 'a2', 'a3', dan 'a4', serta inisialisasi array 'arRet' dengan posisi-posisi awal.
        + Kode baris 307 memuat inisialisasi dengan nilai 3, yang merupakan jumlah elemen dalam array 'aarr' dan 'arRet'.
        + Kode baris 308 - 319 memuat perulangan 'for' digunakan untuk melakukan pengurutan nilai dalam array 'aarr' dan penyesuaian posisi dalam array 'arRet'. Dalam perulangan tersebut, dilakukan perulangan bersarang untuk membandingkan setiap pasangan nilai dalam array aarr. Variabel 'nilA' dan 'nilB' digunakan untuk menyimpan nilai-nilai dalam array 'aarr' yang akan dibandingkan, sedangkan 'pos1' dan 'pos2' digunakan untuk menyimpan posisi yang akan disesuaikan dalam array 'arRet'. Jika 'nilA' lebih kecil dari 'nilB', maka nilai 'nilA' dan 'nilB' ditukar posisinya, serta posisi 'pos1' dan 'pos2' dalam array 'arRet' juga ditukar.
        + Kode baris 323 memuat fungsi pengembalikan nilai pertama dalam array 'arRet', yaitu 'arRet[0]', yang merupakan posisi dengan nilai terbesar dalam array 'aarr'.

##### Program memeriksa dan mengendalikan status lampu berdasarkan mode operasi yang dipilih *auto* atau *manual*

Program 5.2.2.14: Program memeriksa dan mengendalikan status lampu berdasarkan mode operasi yang dipilih *auto* atau *manual*

|  |  |
| --- | --- |
| 326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351 | void cekMode() {  if (millis() - pmCek > 1000) { if (stAuto) {  if (blmHitung) { hitungKmeans();  EEPROM.put(addCentroid1, Centro1); EEPROM.put(addCentroid2, Centro2); EEPROM.put(addCentroid3, Centro3); EEPROM.put(addCentroid4, Centro4); EEPROM.commit();  blmHitung = false;  }  Serial.print("\tMode AUTO");  digitalWrite(lampuPin, Saklar); digitalWrite(lampu2Pin, Saklar);  } else {  blmHitung = false;  if (ctrck > 600) ctrck = 0; Serial.println("\tMode MAnual"); digitalWrite(lampuPin, stLmp); digitalWrite(lampu2Pin, stLmp);  }  pmCek = millis();  }  } |

Penjelasan Kode Program:

* + - * + Kode baris 327 memuat untuk fungsi ini hanya dieksekusi setiap 1 detik.
        + Kode baris 260 memuat untuk memeriksa mode operasi dalam interval waktu tertentu.
        + Kode baris 329 - 341 memuat pengecekan kondisi blmHitung. Jika kondisi ini terpenuhi (bernilai true), maka dilakukan perhitungan K-Means clustering dengan memanggil fungsi hitungKmeans(). Hasil centroid yang diperoleh kemudian disimpan ke dalam EEPROM menggunakan pernyataan EEPROM.put(). Setelah perhitungan selesai, nilai blmHitung diatur sebagai false agar perhitungan tidak diulang pada pemanggilan berikutnya. Pada mode kerja otomatis, diaktifkan lampu dengan mengatur pin lampuPin dan lampu2Pin sesuai dengan nilai Saklar.'.
        + Kode baris 342 - 347 memuat Pada mode kerja manual (stAuto = false), nilai blmHitung diatur sebagai false dan nilai ctrck diatur sebagai 0. Pada mode kerja manual, diaktifkan lampu dengan mengatur pin lampuPin dan lampu2Pin sesuai dengan nilai stLmp.
        + Kode baris 349 memuat untuk menyimpan waktu terakhir pemanggilan fungsi.

##### Program pengaturan awal yang diperlukan sebelum program memasuki operasi utama

|  |  |
| --- | --- |
| Program 5.2.2.15: Program pengaturan awal yang diperlukan sebelum program memasuki operasi utama | |
| 354  355  356  357 | void setup() { Serial.begin(9600); cekEEPROM();  setpin(); |

|  |  |
| --- | --- |
| 358  359  360 | RTCAwal(); blmHitung = true;  } |

Penjelasan Kode Program:

* + - * + Kode baris 354 memuat pengaturan awal yang diperlukan sebelum program memasuki operasi utama.
        + Kode baris 355 memuat komunikasi antara mikrokontroler dan Serial Monitor untuk keperluan debugging dan pencetakan pesan.
        + Kode baris 356 memuat untuk melakukan pengecekan dan pengaturan awal nilai-nilai yang ada dalam EEPROM.
        + Kode baris 357 memuat melakukan pengaturan mode pin (*INPUT* atau *OUTPUT*) pada pin yang digunakan dalam program. Ini akan mengkonfigurasi pin 'tModePin', 'tSaklarPin', 'lampuPin', dan 'lampu2Pin'.
        + Kode baris 358 memuat inisialisasi modul RTC dan pengecekan status RTC.
        + Kode baris 359 memuat untuk menandakan bahwa perhitungan K-Means clustering belum dilakukan. Nilai ini akan digunakan pada fungsi cekMode() untuk memastikan bahwa perhitungan dilakukan pada awalnya.

##### Program untuk menjalankan program secara berulang membaca tombol, waktu RTC, dan mengendalikan mode operasi dan status lampu

|  |  |
| --- | --- |
| Program 5.2.2.16: Program untuk menjalankan program secara berulang membaca tombol, waktu RTC, dan mengendalikan mode operasi dan status lampu | |
| 363  364  365  366  367 | void loop() { bacaTombol(); bacaRtc(); cekMode();  } |

Penjelasan Kode Program:

* + - * + Kode baris 363 memuat untuk menjalankan logika program secara berulang untuk membaca tombol, membaca waktu RTC, dan memeriksa serta mengendalikan mode operasi dan status lampu.
        + Kode baris 364 memuat untuk membaca status tombol dan memperbarui variabel 'stAuto' dan 'stLmp' berdasarkan status tombol yang dibaca.
        + Kode baris 365 memuat untuk memperbarui variabel pmRtc untuk memantau waktu terakhir pembacaan RTC.
        + Kode baris 367 memuat fungsi yang akan memperbarui status lampu berdasarkan variabel 'stAuto', 'stLmp', dan 'Saklar'.

# BAB 6 PENGUJIAN DAN ANALISIS

Dalam bab pengujian dan analisis bertujuan untuk melihat hasil dan mengevaluasi kinerja. Baik komponen penyusun system ataupun sistem secara keseluruhan. Dalam bab ini meliputi hasil dan juga analisis hasil pengujian

## Pengujian Nilai Centroid

Pengujian ini dilakukan menggunakan *Euclidian Distance* untuk mengetahui nilai Centroid terbaik pada dataset yang telah dibuat. Besar nilai centroid sangat berpengaruh terhadap hasil perhitungan algoritma *K-Means*.

### Prosedur Pengujian Nilai Centroid

Pada pengujian ini, akan dilakukan penentuan dan evaluasi nilai-nilai centroid yang optimal untuk algoritma K-Means. Langkah-langkah pengujian yang dilakukan adalah sebagai berikut:

1. Memilih dataset time series yang mencakup data penggunaan lampu.
2. Menentukan jumlah kluster yang akan digunakan dalam analisis K-Means, sesuai dengan karakteristik dataset dan tujuan analisis.
3. Melakukan pra-pemrosesan data, seperti normalisasi data dan pengisian nilai yang hilang, jika diperlukan.
4. Inisialisasi nilai centroid awal untuk algoritma K-Means. Nilai-nilai ini dapat dipilih secara acak atau berdasarkan kebiasaan penggunaan
5. Menjalankan algoritma K-Means pada dataset dengan menggunakan nilai centroid awal yang telah ditentukan. Melakukan iterasi algoritma hingga mencapai konvergensi.
6. Analisis dan interpretasi hasil pengujian nilai centroid K-Means untuk memastikan bahwa nilai-nilai centroid tersebut mampu menghasilkan klustering yang sesuai dengan tujuan analisis.

### Hasil dan Analisis Pengujian Nilai Centroid

Hasil dan analisis dari pengujian nilai centroid tidak mengalami perubahan nilai centroid dari perhitungan normal dan perhitungan mikrokontroler. Hal ini ditunjukkan oleh tabel 6.1 mendapatkan nilai perhitungan yang serupa.

##### Tabel 6.1 Tabel Pengujian Nilai Centroid

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Perhitungan Normal | | | | | | | | Perhitungan Mikrokontroler | | | | | | | |
| C1 | | C2 | | C3 | | C4 | | C1 | | C2 | | C3 | | C4 | |
| X | Y | X | Y | X | Y | X | Y | X | Y | X | Y | X | Y | X | Y |
| 1 | 325 | 1 | 420 | 0 | 1029 | 1 | 1373 | 0 | 325 | 1 | 420 | 0 | 1029 | 1 | 1373 | 0 |
| 2 | 321 | 1 | 423 | 0 | 1020 | 1 | 1380 | 0 | 321 | 1 | 423 | 0 | 1020 | 1 | 1380 | 0 |
| 3 | 300 | 1 | 401 | 0 | 1042 | 1 | 1362 | 0 | 300 | 1 | 401 | 0 | 1042 | 1 | 1362 | 0 |
| 4 | 324 | 1 | 410 | 0 | 1083 | 1 | 1402 | 0 | 324 | 1 | 410 | 0 | 1083 | 1 | 1402 | 0 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 299 | 1 | 400 | 0 | 1030 | 1 | 1333 | 0 | 299 | 1 | 400 | 0 | 1030 | 1 | 1333 | 0 |
| 6 | 285 | 1 | 360 | 0 | 1019 | 1 | 1398 | 0 | 285 | 1 | 360 | 0 | 1019 | 1 | 1398 | 0 |
| 7 | 287 | 1 | 382 | 0 | 1023 | 1 | 1327 | 0 | 287 | 1 | 382 | 0 | 1023 | 1 | 1327 | 0 |
| 8 | 312 | 1 | 378 | 0 | 993 | 1 | 1392 | 0 | 312 | 1 | 378 | 0 | 993 | 1 | 1392 | 0 |
| 9 | 315 | 1 | 393 | 0 | 1019 | 1 | 1320 | 0 | 315 | 1 | 393 | 0 | 1019 | 1 | 1320 | 0 |
| 10 | 342 | 1 | 392 | 0 | 1054 | 1 | 1381 | 0 | 342 | 1 | 392 | 0 | 1054 | 1 | 1381 | 0 |
| 11 | 292 | 1 | 400 | 0 | 1006 | 1 | 1262 | 0 | 292 | 1 | 400 | 0 | 1006 | 1 | 1262 | 0 |
| 12 | 300 | 1 | 378 | 0 | 1049 | 1 | 1380 | 0 | 300 | 1 | 378 | 0 | 1049 | 1 | 1380 | 0 |
| 13 | 330 | 1 | 376 | 0 | 997 | 1 | 1379 | 0 | 330 | 1 | 376 | 0 | 997 | 1 | 1379 | 0 |

Tabel 6.1 menunjukkan hasil pengujian nilai centroid, di mana setiap data time series diberikan ke kluster yang memiliki pusat terdekat dengan data tersebut. Pusat kluster dihitung sebagai nilai rata-rata atau representasi dari data yang ada dalam kluster tersebut. Setelah seluruh data diberikan ke kluster, pusat kluster diupdate berdasarkan data yang ada dalam kluster tersebut. Proses penugasan data dan pembaruan pusat kluster diulang hingga kondisi berhenti terpenuhi, seperti mencapai jumlah iterasi yang telah ditentukan atau perubahan pusat kluster yang sangat kecil. Setelah algoritma K-Means konvergen, diperoleh hasil akhir dari K-Means Clustering pada data time series smarthome. Hasil tersebut berupa data time series yang telah dikelompokkan menjadi K kluster berdasarkan pusat kluster yang terbentuk. Dari tabel tersebut, dapat diamati bahwa nilai-nilai pusat kluster tidak mengalami perubahan dari perhitungan normal hingga perhitungan pada mikrokontroler.

## Pengujian Keseluruhan Sistem

Pada bagian ini, dilakukan pengujian keseluruhan sistem yang terdiri dari objek lampu, sensor RTC, dan modul relay pada perangkat Smart Home berbasis ESP32. Pengujian ini mencakup prosedur pengujian keseluruhan sistem dan pengujian black box menggunakan tabel.

### Prosedur Pengujian Keseluruhan Sistem

Prosedur pengujian keseluruhan sistem dilakukan dengan langkah-langkah berikut:

* + - 1. Pengaturan Waktu pada Sensor RTCMembuka software Arduino IDE
         1. Memastikan sensor RTC terhubung dengan perangkat ESP32.
         2. Mengatur waktu pada sensor RTC sesuai dengan waktu yang diinginkan.
      2. Pengendalian Lampu menggunakan Kluster
         1. Mengaktifkan sistem Smart Home berbasis ESP32.
         2. Memastikan lampu, sensor RTC, dan modul relay terhubung dan berfungsi dengan baik.
         3. Melakukan proses klustering menggunakan algoritma K-Means berdasarkan dataset time series yang telah disimpan dalam EEPROM.
         4. Memastikan lampu dikendalikan sesuai dengan kluster yang ditentukan
      3. Mode Operasi *Auto*.
         1. Dapat mengaktifkan tombol mode operasi *Auto* pada sistem Smart Home.
         2. Memverifikasi apakah lampu secara otomatis beradaptasi berdasarkan waktu yang terdeteksi oleh sensor RTC
      4. Mode Operasi *Manual*.
         1. Dapat mengaktifkan tombol mode operasi *Manual* pada sistem Smart Home
      5. Melakukan evaluasi hasil klustering yang dihasilkan oleh algoritma K- Means

### Hasil dan Analisi Pengujian Keseluruhan Sistem

Pada pengujian keseluruhan sistem ini, terdapat pengujian Black Box dan pengujian akurasi algoritma K-Means.

###### Pengujian Fungsional

Pengujian black box dilakukan dengan menggunakan tabel black box yang mencakup skenario pengujian dan hasil yang diharapkan. Tabel black box dirancang untuk menguji fungsionalitas sistem secara keseluruhan. Berikut adalah contoh Tabel 6.2:

**Tabel 6.2 Tabel Hasil Pengujian Keseluruhan Sistem (Fungsional*)***

|  |  |  |  |
| --- | --- | --- | --- |
| No | Skenario Pengujian | Hasil Yang Diharapkan | Hasil Pengujian |
| 1. | Pengaturan Waktu pada Sensor RTC | Sensor RTC berhasil diatur waktu | Sensor RTC waktu diatur dengan sukses |
| 2. | Pengendalian Lampu menggunakan Kluster | Lampu dikendalikan sesuai kluster | Lampu dikendalikan sesuai dengan kluster yang ditentukan |
| 3. | Mode Operasi AUTO | Lampu beradaptasi sesuai waktu | Lampu beradaptasi secara otomatis sesuai dengan waktu |

|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Mode Operasi MANUAL | Lampu dikendalikan secara manual | Lampu dikendalikan secara manual sesuai dengan kebutuhan pengguna |
| 5. | Evaluasi Kualitas Klustering | Klustering sesuai dengan tujuan | Evaluasi kualitas klustering memenuhi kriteria yang ditetapkan |

pada tabel 6.1 pengujian fungsional menunjukan dari tiap pengujian skenario terhadap Pengaturan waktu pada sensor RTC, pengendalian lampu menggunakan kluster, mode operasi *AUTO/MANUAL*, Evaluasi Kualitas Klustering mendapat hasil pengujian sesuai kriteria.

###### Pengujian Akurasi Centroid K-Means

Pada pengujian akurasi Centroid *K-Means* sangat diperlukan karena ini merupakan penentu tingakat kesuksesan system dalam menjalankan fungsi utamanya. Untuk mencari rata rata akurasi ini, dilakukan pengujian 3 sampel uji. Berikut adalah hasil pengujian akurasi centroid K-Means disajikan melalui Tabel 6.3.

##### Tabel 6.3 Tabel Pengujian Akurasi Centroid K-Means

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | C1 | | C2 | | C3 | | C4 | | Kondisi Sebenarnya | Status |
| X | Y | X | Y | X | Y | X | Y |
| 1 | 310 | 1 | 393 | 0 | 1028 | 1 | 1360 | 0 | Jam 05:10 | Nyala |
| Jam 06:33 | Mati |
| Jam 17:08 | Nyala |
| Jam 22:39 | Mati |
| 2 | 345 | 1 | 421 | 0 | 1010 | 1 | 1311 | 0 | Jam 05:45 | Nyala |
| Jam 07:01 | Mati |
| Jam 16:50 | Nyala |
| Jam 21:51 | Mati |
| 3 | 328 | 1 | 404 | 0 | 1000 | 1 | 1250 | 0 | Jam 05:50 | Nyala |
| Jam 07:44 | Mati |
| Jam 16:40 | Nyala |
| Jam 20:50 | Mati |

Berdasarkan hasil pengujian akurasi centroid *K-Means* pada tabel 6.3, dapat dilihat bahwa dengan 3 kali pengujian sistem mampu mendapatkan nilai serupa 100% dari perhitungan manual dan mikrokontroler. Dengan demikian, pengujian

ini memperkuat keandalan algoritma K-Means dalam mengklasifikasikan data dengan nilai centroid yang berbeda, serta menunjukkan bahwa mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 tetap dapat berfungsi dengan baik dan memberikan hasil yang konsisten meskipun terdapat variasi pada nilai centroid.

# BAB 7 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

## Kesimpulan

Dalam penelitian ini, penulis telah membangun mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 menggunakan algoritma K- Means pada dataset time series. Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja mekanisme tersebut dan mengukur ketepatan sistem secara keseluruhan dalam hal implementasi sebelum digunakan oleh pengguna. Berdasarkan hasil penelitian dan analisis yang telah dilakukan, kami menyimpulkan hal-hal berikut:

* + 1. Tingkat akurasi perhitungan dataset time series menggunakan algoritma K- Means pada mekanisme penyediaan layanan yang dibangun adalah 100%

. Algoritma K-Means mampu mengklasifikasikan pola data waktu dengan akurasi yang tinggi, sehingga sistem dapat memberikan pengaturan pencahayaan yang sesuai dengan kondisi yang diharapkan.

* + 1. Kinerja sistem secara keseluruhan dalam hal implementasi yang akan digunakan oleh pengguna cukup baik. Sistem mampu mengelompokkan data time series dengan efektif dan menghasilkan pengaturan pencahayaan yang adaptif. Implementasi mekanisme penyediaan layanan pada perangkat Smart Home berbasis ESP32 dengan menggunakan algoritma K-Means mampu memberikan pengalaman pengguna yang nyaman dan efisien.

## Saran

Berikut saran-saran oleh penulis yang berguna bagi penelitian mendatang.

Untuk mengoptimalkan hasil dari sistem ini sebagai berikut:

* + 1. Mengintegrasikan sistem layanan *Smart Home* dengan platform pengelolaan dan monitoring jarak jauh, sehingga pengguna dapat mengendalikan dan memantau perangkat *Smart Home* dari jarak jauh melalui aplikasi mobile atau web.

# DAFTAR REFERENSI

P. Catur Siswipraptini, R. et al. (2020). "K-Means Clustering Algorithm for Smart Home Automation,". 207-211, Diambil dari: <https://ieeexplore.ieee.org/abstract/document/9301506>

Sujitha, V. et al (2022). "IoT based Healthcare Monitoring and Tracking System for Soldiers using ESP32". 377-381, Diambil dari: <https://ieeexplore.ieee.org/abstract/document/9754076>

Ngatikoh, Hany (2022) SISTEM MONITORING DAN PENENTUAN KUALITAS AIR PADA KOLAM IKAN BERBASIS IOT DENGAN METODE K-MEANS

CLUSTERING. Skripsi thesis, Universitas Teknologi Digital Indonesia, Diambil dari: <https://eprints.utdi.ac.id/9657/>

Qodari, N. (2020). PERANCANGAN PROTOTIPE *REALTIME* AKUISISI DATA DAN KONTROL STOP KONTAK MENGGUNAKAN ESP 32 BERBASIS WEB. 2685-

0206, Diambil dari <https://ejournal3.undip.ac.id/index.php/transient/article/view/27921>

Venkatesh, K. and Naik, K. J. (2021) An IoT Framework for Groundnut Crop Yield Prediction Using K-Means Algorithm. 266-271, Diambil dari <https://ieeexplore.ieee.org/abstract/document/9655923>

Ahanger, T. A. (2019). IoT-Inspired Framework of Intruder Detection for Smart Home Security Systems. Electronics, 9(9), 1361. Diambil dari <https://www.mdpi.com/2079-9292/9/9/1361>

Katiyar, A. (2023). Thresholding based Smart Home Automation System using K- means. 978-1-6654-6263-1. Diambil dari https://ieeexplore.ieee.org/abstract/document/10048795

Nishom, M. (2019). Perbandingan Akurasi Euclidean Distance, Minkowski Distance dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi- Square. 20-24. Diambil dari https://[www.researchgate.net/publication/332758314\_Perbandingan\_Ak](http://www.researchgate.net/publication/332758314_Perbandingan_Ak) urasi\_Euclidean\_Distance\_Minkowski\_Distance\_dan\_Manhattan\_Distanc e\_pada\_Algoritma\_K-Means\_Clustering\_berbasis\_Chi-Square

Chansareewittaya, S. (2022). Development of Smart Light Detection and Monitoring System with Machine Learning. 12(1), 19-26. Diambil dari <https://ieeexplore.ieee.org/abstract/document/9720384>

Taiwo, O. (2021). *Internet of Things-Based Intelligent Smart Home Control System*, 2021, 9928254. Diambil dari <https://www.hindawi.com/journals/scn/2021/9928254/>

Kaur, et all. (2017). *Smart Homes: Sensible Living Using Internet of Things*. 10 (31),

1 – 8. Diambil dari [https://sciresol.s3.us-east-](https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2017/Issue-31/Article34.pdf) [2.amazonaws.com/IJST/Articles/2017/Issue-31/Article34.pdf](https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2017/Issue-31/Article34.pdf)

Pratama, A. A. (2023). Monitoring *drone* menggunakan ESP32. 3(2), 2023. Diambil dari <http://teknologiterkini.org/index.php/terkini/article/view/349>

Jethanandani, M. (2020). Multi-label classification based ensemble learning for human activity recognition in smart home. 11(10), 4511-4524. Diambil dari <https://www.sciencedirect.com/science/article/pii/S2542660520301554>

R. I. Fajriah, H. Sutisna and B. K. Simpony. (2019). Perbandingan Distance Space Manhattan Distance dengan Euclidean Distance pada K- Means Clustering dalam menentukan promosi. 37-49, 2019. Diambil dari <http://download.garuda.kemdikbud.go.id/article.php?article=1030004&v> al=10500&title=Perbandingan%20Distance%20Space%20Manhattan%20 Dengan%20Euclidean%20Pada%20K- Means%20Clustering%20Dalam%20Menentukan%20Promosi

DQlab. (2022). K-Means Clustering, Salah Satu Contoh Teknik Analisis Data Populer. Diambil dari [https://dqlab.id/k-means-clustering-salah-satu-](https://dqlab.id/k-means-clustering-salah-satu-contoh-teknik-analisis-data-populer) [contoh-teknik-analisis-data-populer](https://dqlab.id/k-means-clustering-salah-satu-contoh-teknik-analisis-data-populer)

Rahul Awati. (2022). EEPROM (electrically erasable programmable read-only memory). Diambil dari https://[www.techtarget.com/whatis/definition/EEPROM-electrically-](http://www.techtarget.com/whatis/definition/EEPROM-electrically-) erasable-programmable-read-only- memory#:~:text=EEPROM%20(electrically%20erasable%20programmable

%20read%2Donly%20memory)%20is%20a,that%20is%20higher%20than% 20normal.

Aldy Razor. (2020). Modul Relay Arduino: Pengertian, Gambar, Skema, dan Lainnya. Diambil dari [https://www.aldyrazor.com/2020/05/modul-relay-](https://www.aldyrazor.com/2020/05/modul-relay-arduino.html) [arduino.html](https://www.aldyrazor.com/2020/05/modul-relay-arduino.html)

Farhan. (2022). Cara Penggunaan Module Relay 2 Channel Arduino. Diambil dari <http://indomaker.com/product/blog/cara-penggunaan-module-relay-2-> channel-arduino/

Javapoint. (2021). K-Means Clustering Algorithm. Diambil dari https://[www.javatpoint.com/k-means-clustering-algorithm-in-machine-](http://www.javatpoint.com/k-means-clustering-algorithm-in-machine-) learning

Nurhilman, D (2021) ESP32. Diambil dari https://raharja.ac.id/2021/11/16/esp32- 2/? cf\_chl\_rt\_tk=5RGXUQsJhs40N.k8IXMTBV9bcWiBgoFn7zENp4aGDN0

-1677322055-0-gaNycGzNClA

# LAMPIRAN A KODE PROGRAM

## A.1 Kode Program Keseluruhan Sistem

|  |  |
| --- | --- |
| Program 5.2.2.8: Program mengontrol dan memperbarui pengaturan lampu berdasarkan mode perangkat | |
| 1 | #include <RTClib.h> |
| 2 | #include <EEPROM.h> |
| 4 | int addCekEprom = 0; |
| 5 | int addCentroid1 = 1; |
| 6 | int addCentroid2 = 20; |
| 7 | int addCentroid3 = 30; |
| 8 | int addCentroid4 = 40; |
| 9 | int adddatake = 47; |
| 10 | int idxDttrain = 0; |
| 11 |  |
| 12 | int Centro1, Centro2, Centro3, Centro4, cekEprom; |
| 13 |  |
| 14 | int dttrainEprom[60] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
| 15 | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
| 16 | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, |
|  | 0, 0 }; |
| 17 |  |
| 18 | int dttrainyEprom[60] = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
| 19 | 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
| 20 | 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, |
|  | 1, 0 }; |
| 21 |  |
| 22 |  |
| 23 | int addDttrain[60] = { 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, |
| 24 | 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165, |
| 25 | 170, 175, 180, 185, 190, 195, 200, 205, 210, 215, 220, 225, 230, |
| 26 | 235, 240, 245, 250, 255, 260, 265, 270, 275, 280, 285, 290, 295, |
| 27 | 300, 305, 310, 315, 320, 325, 330, 335, 340, 345 }; |
| 28 |  |
| 29 | int dttrain[52] = { 325, 420, 1029, 1373, 321, 423, 1020, 1380, |
| 30 | 300, 401, 1042, 1362, 324, 410, 1083, 1402, 299, 400, 1030, 1333, |
| 31 | 285, 360, 1019, 1398, 287, 382, 1023, 1327, 312, 378, 993, 1392, |
| 32 | 315, 393, 1019, 1320, 342, 392, 1054, 1381, 292, 400, 1006, 1262, |
| 33 | 300, 378, 1049, 1380, 330, 376, 997, 1379 }; |
| 34 | int dttrainY[52] = { 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, |
| 35 | 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, |
| 36 | 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0 }; |
| 37 | int centrox[4] = { 325, 380, 1029, 1373 }; |
| 38 | int centroy[4] = { 1, 0, 1, 0 }; |
| 39 | int nil, k = 10; |
| 40 | String hsaw = "111111111111111111111"; |
| 41 | String hsSbl, hsSkr; |
| 42 |  |
| 43 | unsigned long pmTombol, pmRtc, pmCek; |
| 44 | #define tModePin 32 |
| 45 | #define tSaklarPin 33 |
| 46 | #define lampuPin 25 |
| 47 | #define lampu2Pin 26 |
| 48 | bool stAuto, stLmp, Saklar, stLmpSbl, blmSimpanEprom, blmHitung; |
| 49 | int ctrck; |
| 50 |  |
| 51 | RTC\_DS1307 rtc; |
| 52 | int jam, menit; |
| 53 |  |
| 54 | void cekEEPROM() { |
| 55 | EEPROM.begin(512); |
| 56 | cekEprom = EEPROM.read(addCekEprom); |

|  |  |
| --- | --- |
| 57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120 | if (cekEprom != 2) { hitungKmeans(); cekEprom = 2;  EEPROM.write(addCekEprom, cekEprom); EEPROM.write(adddatake, idxDttrain); EEPROM.put(addCentroid1, Centro1); EEPROM.put(addCentroid2, Centro2); EEPROM.put(addCentroid3, Centro3); EEPROM.put(addCentroid4, Centro4);  EEPROM.commit();  }  EEPROM.get(addCentroid1, Centro1); EEPROM.get(addCentroid2, Centro2); EEPROM.get(addCentroid3, Centro3); EEPROM.get(addCentroid4, Centro4); idxDttrain = EEPROM.read(adddatake); delay(500);  for (int idxa = 0; idxa < 60; idxa++) { int al = addDttrain[idxa]; EEPROM.get(al, dttrainEprom[idxa]); al = 400 + idxa;  dttrainyEprom[idxa] = EEPROM.read(al); delay(50);  }  }  String seplit(String data, char separator, int index) { int found = 0;  int strIndex[] = { 0, -1 };  int maxIndex = data.length() - 1;  for (int i = 0; i <= maxIndex && found <= index; i++) { if (data.charAt(i) == separator || i == maxIndex) {  found++;  strIndex[0] = strIndex[1] + 1;  strIndex[1] = (i == maxIndex) ? i + 1 : i;  }  }  return found > index ? data.substring(strIndex[0], strIndex[1])  : "";  }  void setpin() {  pinMode(tModePin, INPUT\_PULLUP); pinMode(tSaklarPin, INPUT\_PULLUP); pinMode(lampuPin, OUTPUT); pinMode(lampu2Pin, OUTPUT);  }  void simpanDtTrain() {  Serial.println("Simpan data train ke eeprom"); int nilj = (jam \* 60) + menit; EEPROM.put(addDttrain[idxDttrain], nilj); EEPROM.write((400 + idxDttrain), stLmp); dttrainEprom[idxDttrain] = nilj; idxDttrain++;  if (idxDttrain > 59) idxDttrain = 0; EEPROM.write(adddatake, idxDttrain); EEPROM.commit();  }  void bacaTombol() { |

|  |  |
| --- | --- |
| 121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185 | if (millis() - pmTombol > 100) { stAuto = digitalRead(tModePin); stLmp = digitalRead(tSaklarPin); if (!stAuto) {  if (stLmpSbl == stLmp) { Serial.print("sama"); if (blmSimpanEprom) {  ctrck++; Serial.println(ctrck); if (ctrck > 600) {  simpanDtTrain(); blmSimpanEprom = false;  }  }  } else {  blmSimpanEprom = true; stLmpSbl = stLmp; ctrck = 0;  }  }  pmTombol = millis();  }  }  void RTCAwal() {  if (!rtc.begin()) { Serial.println("Couldn't find RTC"); Serial.flush();  while (1) delay(10);  }  if (!rtc.isrunning()) {  Serial.println("RTC Abis OFF! setting waktu"); rtc.adjust(DateTime(F( DATE ), F( TIME )));  }  // rtc.adjust(DateTime(F( DATE ), F( TIME )));  // rtc.adjust(DateTime(2023, 7, 8, 22, 38, 0));  }  void bacaRtc() {  if (millis() - pmRtc >= 1000) { nil = 0;  DateTime now = rtc.now(); jam = now.hour();  menit = now.minute(); Serial.print("Waktu : "); Serial.print(jam); Serial.print(":"); Serial.println(menit); nil += jam \* 60;  nil += menit; int sk;  int clus = 0;  if (nil < Centro1) { sk = centroy[3]; clus = 4;  } else if (nil < Centro2) { sk = centroy[0];  clus = 1;  } else if (nil < Centro3) { sk = centroy[1];  clus = 2;  } else if (nil < Centro4) { sk = centroy[2]; |

|  |  |
| --- | --- |
| 186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244 | clus = 3;  } else {  sk = centroy[3]; clus = 4;  }  if (sk == 0) Saklar = 1; else Saklar = 0;  if (stAuto) { Serial.print("Cluster : "); Serial.println(clus);  }  pmRtc = millis();  }  }  void hitungKmeans() { hsSbl = hsaw; hsSkr = "";  float cc1 = centrox[0]; float cc2 = centrox[1]; float cc3 = centrox[2]; float cc4 = centrox[3];  bool sama = false; String hs = "0";  int cndtrin = sizeof(dttrain) / sizeof(int); while (!sama) {  Serial.print(cc1); Serial.print("\t"); Serial.print(cc2); Serial.print("\t"); Serial.print(cc3); Serial.print("\t"); Serial.println(cc4); hsSkr = "";  float tempc1 = 0; float tempc2 = 0; float tempc3 = 0; float tempc4 = 0;  int ctr1 = 0; int ctr2 = 0; int ctr3 = 0; int ctr4 = 0;  bool adanilai = true;  for (int aa = 0; aa < 111; aa++) { float aa1, aa2, aa3, aa4;  if (aa < 52) {  aa1 = sqrt(sq(dttrain[aa] - cc1) + sq(dttrainY[aa] - centroy[0]));  aa2 = sqrt(sq(dttrain[aa] - cc2) + sq(dttrainY[aa] - centroy[1]));  aa3 = sqrt(sq(dttrain[aa] - cc3) + sq(dttrainY[aa] - centroy[2]));  aa4 = sqrt(sq(dttrain[aa] - cc4) + sq(dttrainY[aa] - centroy[3]));  } else {  int nildteprom = dttrainEprom[aa - 52]; if (nildteprom > 0) {  aa1 = sqrt(sq(nildteprom - cc1) + sq(dttrainyEprom[aa - 52] - centroy[0]));  aa2 = sqrt(sq(nildteprom - cc2) + sq(dttrainyEprom[aa - 52] - centroy[1])); |

|  |  |
| --- | --- |
| 245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307 | aa3 = sqrt(sq(nildteprom - cc3) + sq(dttrainyEprom[aa - 52] - centroy[2]));  aa4 = sqrt(sq(nildteprom - cc4) + sq(dttrainyEprom[aa - 52] - centroy[3]));  } else {  adanilai = false;  }  }  if (adanilai) {  int pos = crpos(aa1, aa2, aa3, aa4);  hs = String(pos); switch (pos) {  case 0:  tempc1 += dttrain[aa]; ctr1++;  break; case 1:  tempc2 += dttrain[aa]; ctr2++;  break; case 2:  tempc3 += dttrain[aa]; ctr3++;  break; case 3:  tempc4 += dttrain[aa]; ctr4++;  break;  }  hsSkr += hs;  }  }  if (hsSbl == hsSkr) { sama = true; Centro1 = cc1; Centro2 = cc2; Centro3 = cc3; Centro4 = cc4;  } else {  hsSbl = hsSkr;  cc1 = tempc1 / ctr1; cc2 = tempc2 / ctr2; cc3 = tempc3 / ctr3; cc4 = tempc4 / ctr4;  hsSkr = "";  tempc1 = tempc2 = tempc3 = tempc4 = 0; ctr1 = ctr2 = ctr3 = ctr4 = 0;  }  }  Saklar = centroy[hs.toInt()]; Serial.print("hasilnya : "); Serial.println(hs);  }  int crpos(float a1, float a2, float a3, float a4) { float aarr[10] = { a1, a2, a3, a4 };  int arRet[10] = { 0, 1, 2, 3 }; int cnAr = 3; |

|  |  |
| --- | --- |
| 308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367 | for (int a = 0; a < cnAr; a++) { for (int b = 0; b < cnAr; b++) {  int c = b + 1;  float nilA = aarr[c]; float nilB = aarr[b]; int pos1 = arRet[c]; int pos2 = arRet[b]; if (nilA < nilB) {  aarr[c] = nilB; aarr[b] = nilA; arRet[c] = pos2; arRet[b] = pos1;  }  }  }  return arRet[0];  }  void cekMode() {  if (millis() - pmCek > 1000) { if (stAuto) {  if (blmHitung) { hitungKmeans();  EEPROM.put(addCentroid1, Centro1); EEPROM.put(addCentroid2, Centro2); EEPROM.put(addCentroid3, Centro3); EEPROM.put(addCentroid4, Centro4); EEPROM.commit();  blmHitung = false;  }  Serial.print("\tMode AUTO");  digitalWrite(lampuPin, Saklar); digitalWrite(lampu2Pin, Saklar);  } else {  blmHitung = false;  if (ctrck > 600) ctrck = 0; Serial.println("\tMode MAnual"); digitalWrite(lampuPin, stLmp); digitalWrite(lampu2Pin, stLmp);  }  pmCek = millis();  }  }  void setup() { Serial.begin(9600); cekEEPROM();  setpin(); RTCAwal(); blmHitung = true;  }  void loop() { bacaTombol(); bacaRtc(); cekMode();  } |

# LAMPIRAN B DOKUMENTASI

## B.1 Dokumentasi Pengunjian sistem



