

**IMPLEMENTASI ALGORITMA K-MEANS PADA ARDUINO
PRO MINI UNTUK OTOMATISASI TERANG REDUP
CAHAYA LAMPU**

COVER
SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan mencapai gelar
Sarjana Komputer



Disusun oleh :

ANDRI BAGUS PRASTIYO

115060900111016

**UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
MALANG
2015**

LEMBAR PERSETUJUAN

**IMPLEMENTASI ALGORITMA K-MEANS PADA ARDUINO
PRO MINI UNTUK OTOMATISASI TERANG REDUP**

CAHAYA LAMPU

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun oleh:

ANDRI BAGUS PRASTIYO

NIM. 115060900111016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Agung Setia Budi, S.T., M.T., M.Eng.

NIK. 2013048704231001

Suprpto, S.T., M.T.

NIP. 19710727 199603 1 001

LEMBAR PENGESAHAN
IMPLEMENTASI ALGORITMA K-MEANS PADA ARDUINO
PRO MINI UNTUK OTOMATISASI TERANG REDUP
CAHAYA LAMPU

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun oleh :

ANDRI BAGUS PRASTIYO

NIM. 115060900111016

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 15 Juni 2015

Penguji I

Penguji II

Rekyan Regasari MP, S.T., M.T.

Barlian Henryranu Prasetio, S.T., M.T.

NIK. 770414 06 1 2 0253

NIK. 821024 06 1 1 0254

Penguji III

Drs. Marji, M.T.

NIP. 19670801 199203 1 001

Mengetahui

Ketua Program Studi Teknik Informatika

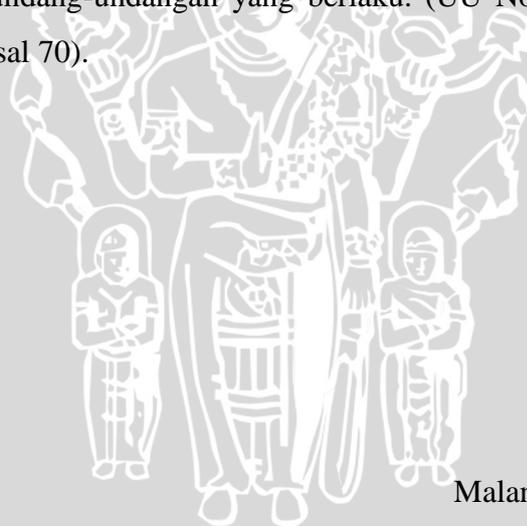
Drs. Marji, M.T.

NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

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, 15 Juni 2015

Mahasiswa,

Andri Bagus Prastiyo

NIM 115060900111016

KATA PENGANTAR

Puji syukur penulis panjatkan kahadirat Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Komputer pada Fakultas Ilmu Komputer. Judul skripsi yang disusun adalah: **“Implementasi Algoritma K-Means Pada Arduino Pro Mini Untuk Otomatisasi Terang Redup Cahaya Lampu ”**

Banyak kesulitan dan hambatan yang dialami oleh penulis dalam menyusun skripsi ini terutama dalam mendapatkan data dan mengolahnya, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagi pihak. Untuk itulah pada kesempatan ini penulis mengucapkan terima kasih kepada:

- 1) Bapak dan Ibu yang saya cintai, yang telah memberikan dukungan, semangat serta doa.
- 2) Bapak Drs. Marji, M.si selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
- 3) Bapak Adharul Mutaqqin, S.T., M.T. selaku Ketua Konsentrasi Teknik Komputer Universitas Brawijaya Malang.
- 4) Bapak Agung Setia Budi, S.T., M.T., M.Eng. selaku dosen pembimbing I yang telah memberikan pengarahan dalam pembuatan proposal skripsi ini.
- 5) Bapak Suprpto, S.T., M.T selaku dosen pembimbing II yang telah memberikan saran-saran yang sangat berguna dalam penyelesaian proposal skripsi ini.
- 6) Barlian Henryranu, S.T., M.T. selaku Kepala Laboratorium Sistem Komputer dan Robotika Fakultas Ilmu Komputer.
- 7) Aninditya Nugroho, S.T (Mas Didit) selaku laboran Kepala Laboratorium Sistem Komputer dan Robotika Fakultas Ilmu Komputer yang selalu membantu & menyediakan alat-alat skripsi untuk penulis.
- 8) Segenap dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segenap ilmu pengetahuan dan perhatian yang diberikan Segenap staff

dan pegawai Fakultas Ilmu Komputer Universitas Brawijaya atas segala bantuan yang bersifat administratif.

- 9) Seluruh teman-teman angkatan 2011 dan pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga proposal skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, 15 Juni 2015

Penulis



ABSTRAK

Andri Bagus Prastiyo.2015. Implementasi Algoritma K-Means Pada Arduino Pro Mini Untuk Otomatisasi Terang Redup Cahaya Lampu.

Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing: Agung Setia Budi, S.T., M.T., M.Eng. dan Suprpto, S.T., M.T.

Teknologi Smart Home sudah berkembang pesat saat ini. Dalam perkembangannya teknologi ini dituntut untuk menerapkan seluruh sistemnya dalam keadaan otomatis serta mampu berjalan secara dinamik, mengikuti keadaan atau aktifitas disekitarnya serta dapat menghemat energi listrik. Penelitian ini membahas bagaimana membuat sistem “ Implementasi Algoritma K-means Pada Arduino Pro Mini Untuk Otomatisasi Terang Redup Cahaya Lampu “. Sistem ini berjalan secara otomatis berdasarkan pembelajaran yang dilakukan atau histori penyalaan lampu sebelumnya. Proses pembelajaran ini menggunakan algoritma K-means sebagai metode otomatisnya. Pada penelitian ini algoritma K-means diimplementasikan pada sistem untuk mengelompokkan data berdasarkan kesamaan atau kemiripan sifat. Proses pengiriman data untuk menyalakan atau mematikan lampu dilakukan dengan *remote control*. *Remote control* ini mengirimkan terang redup cahaya melalui radio frekuensi (RF). Data yang dikirimkan *Remote control* ini oleh sistem akan disimpan di EEPROM sistem dan akan diambil pada saat perhitungan algoritma K-means dijalankan. Penelitian ini bertujuan untuk menghemat energi listrik berdasarkan kebiasaan manusia dalam mengatur terang redup cahaya lampu pada suatu ruangan. Berdasarkan hasil pengujian didapat persentase error untuk terang redup cahaya berdasarkan kebiasaan adalah 0% dan persentase error untuk waktu otomatisasi terang redup cahaya lampu adalah 0,055%.

Kata Kunci: K-means, energi listrik, pembelajaran

ABSTRACT

Andri Bagus Prastiyo.2015. K-Means Algorithm Implementation On Arduino Pro Mini For Lamp Brightness

Faculty of Computer Science,
University of Brawijaya, Malang.

Supervisor: Agung Setia Budi, S.T., M.T., M.Eng. and Suprpto, S.T., M.T.

Today Smart Home technology has been developed rapidly. The technology run automatically according to the situation around it in low energy mode. In this research show how to make a brightness automation of lamp system using K-means algorithm. The system use arduino pro-mini as microcontroller. The system record the usual habit of user when the lamp turned on. After recording process the system will work automatically according to the user habit. In this research K-means algorithm is used to classify the data according to its similarity. The system using remote control with radio frequency technology to transmit the input. The input will be saved into EEPROM and will be taken when K-means algorithm computation process run. This research was made to save energy according to human habit in a room. The result of this research is 0% error in usual habit recording and 0,055% recording in power automation.

Key World: K-means, electricity energy, learning.

DAFTAR ISI

COVER.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
ORISINALITAS SKRIPSI	iv
KATA PENGANTAR	v
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan masalah.....	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Sistematika Penulisan.....	4
BAB II KAJIAN PUSTAKA DAN DASAR TEORI	5
2.1. Kajian Pustaka	5
2.2. Dasar Teori.....	6
2.2.1. <i>Smart Home</i>	6
2.2.2. <i>Wireless Sensor Network (WSN)</i>	7
2.2.3. K-MEANS.....	9
2.2.4. Arduino Pro Mini.....	11
2.2.5. <i>Real Time Clock (RTC)</i>	12
2.2.6. NRF24L01	12
2.2.6.1. Spesifikasi pada modul NRF24L01	13
2.2.6.2. Fitur-fitur pada modul NRF24L01.....	14

2.2.7.	TRIAC.....	15
2.2.8.	<i>Optocoupler / Optoisolator</i> MOC3021.....	17
2.2.9.	OptoCoupler 4N25.....	18
BAB III METODOLOGI PENELITIAN.....		20
3.1.	Metodologi Penelitian.....	20
3.1.1.	Studi Literatur.....	21
3.1.2.	Analisis Kebutuhan.....	21
3.1.2.1.	Kebutuhan Perangkat Keras.....	21
3.1.2.2.	Kebutuhan Perangkat Lunak.....	21
3.1.3.	Penyusunan Dasar Teori.....	21
3.1.4.	Perancangan Sistem.....	22
3.1.5.	Implementasi.....	22
3.1.6.	Pengujian.....	22
3.1.7.	Analisis.....	23
3.1.8.	Kesimpulan.....	23
BAB IV PERANCANGAN DAN IMPLEMENTASI.....		24
4.1.	Perancangan.....	24
4.1.1.	Analisis Kebutuhan Perangkat Keras dan Lunak.....	24
4.1.2.	Diagram Blok Sistem.....	24
4.1.3.	Diagram Alir Sistem.....	25
4.1.4.	Perancangan Perangkat Keras.....	27
4.1.5.	Perancangan Arduino dan NRF24L01.....	27
4.1.6.	Perancangan Rangkaian <i>Dimmer</i>	27
4.1.7.	Perancangan Rangkaian RTC.....	28
4.1.8.	Perancangan Perangkat Lunak.....	29
4.1.9.	Perancangan Paket data.....	29
4.1.10.	Perancangan Algoritma K-means.....	30
4.1.11.	Perancangan Penyimpanan Data dan Pembacaan Data ke EEPROM.....	30
4.2.	Implementasi.....	33
4.2.1.	Implementasi Perangkat Keras.....	34

4.2.2.	Impementasi Perangkat Lunak	34
4.2.3.	Implementasi Algoritma K-means	34
4.2.4.	Implementasi Program <i>Dimmer</i>	37
4.2.5.	Implementasi Program RTC	38
4.2.6.	Implementasi Penulisan dan Pembacaan EEPROM Arduino.....	39
BAB V PENGUJIAN DAN ANALISIS		41
5.1.	Pengujian Penyimpanan dan Pembacaan Data ke eprom	41
5.2.	Pengujian Rangkaian <i>Dimmer</i>	48
5.3.	Pengujian Rangkaian RTC.....	50
5.4.	Pengujian Algoritma K-means.....	51
5.5.	Pengujian Komunikasi Data Dengan NRF24L01	55
5.6.	Pengujian Pengiriman waktu dan nilai terang redup cahaya melalui NRF24L01	57
5.7.	Pengujian Keseluruhan Sistem	60
5.7.1.	Skenario pngujian	60
5.7.2.	Analisis pengujian	60
5.8.	Pengujian Keseluruhan Sistem Dengan Data Acak.....	72
5.8.1.	Skenario pngujian	72
5.8.2.	Analisis pengujian	73
5.9.	Analisis Pengujian dari kedua pengujian keseluruhan sistem	76
BAB VI PENUTUP		80
6.1.	Kesimpulan	80
6.2.	Saran	81
DAFTAR PUSTAKA		82

DAFTAR GAMBAR

Gambar 2.1. WSN Kategori 1	8
Gambar 2.2. WSN Kategori 2	9
Gambar 2.3. Board Arduino Pro Mini	11
Gambar 2.4. Konfigurasi <i>Pin</i> DS1302	12
Gambar 2.5. NRF24L01	13
Gambar 2.6. Rangkaian ekuivalen TRIAC.	15
Gambar 2.7. Simbol TRIAC.	16
Gambar 2.8. Karakteristik TRIAC.	16
Gambar 2.9. Konfigurasi (kiri) dan Bentuk fisik (kanan) MOC3021	18
Gambar 2.10. Optocoupler tipe 4N25	19
Gambar 3.1. Flowchart Runtutan Pengerjaan Penelitian	20
Gambar 4.1. Diagram Blok Sistem	25
Gambar 4.2. Diagram Alir Sistem.....	26
Gambar 4.3. Rangkaian Arduino ke NRF24L01	27
Gambar 4.4. Rangkaian <i>Dimmer</i> lampu.....	28
Gambar 4.5. Rangkaian RTC	28
Gambar 4.6. Diagram alir proses perhitungan K-means.....	31
Gambar 4.7. Proses penyimpan data di EEPROM arduino	32
Gambar 4.8. Proses pembacaan data di EEPROM arduino	33
Gambar 4.9. Hasil implementasi perancangan perangkat keras	34
Gambar 4.10. <i>Source code</i> algoritma K-MEANS.....	36
Gambar 4.11. Proses algoritma K-means.....	37
Gambar 4.12. Potongan <i>source code</i> program <i>dimmer</i>	38
Gambar 4.13. Potongan <i>source code</i> program RTC	39
Gambar 4.14. Potongan <i>source code</i> program Pembacaan dan Penulisan data di EEPROM arduino.....	40
Gambar 5.1. Flowchart Penyimpanan EEPROM.....	42
Gambar 5.2. Flowchart Pembacaan EEPROM	45
Gambar 5.3. Data yang disimpan di EEPROM.....	48
Gambar 5.4. Proses Pengaturan <i>Dimmer</i>	49

Gambar 5.5. Terang redup Cahaya	50
Gambar 5.6. Waktu Sebelum daya Arduino dimatikan.	51
Gambar 5.7. Waktu setelah daya Arduino dimatikan.	51
Gambar 5.8. Diagram Alir Proses K-means.....	53
Gambar 5.9. Hasil Perhitungan Proses K-means diarduino Pro mini.....	55
Gambar 5.10. Diagram alir komunikasi.....	56
Gambar 5.11. Modul pada client.....	56
Gambar 5.12. Modul pada server.....	57
Gambar 5.13. Diagram blok <i>remote control</i> dan aktuator.....	58
Gambar 5.14. Pengiriman data dari modul di <i>remote control</i>	59
Gambar 5.15. Data yang diterima dimodul aktuator.....	59
Gambar 5.16. Data dan Hasil perhitungan hari pertama.....	62
Gambar 5.17. Data dan Hasil perhitungan hari kedua.	63
Gambar 5.18. Data dan Hasil perhitungan hari ketiga.....	64
Gambar 5.19. Data dan Hasil perhitungan hari keempat.....	65
Gambar 5.20. Hasil Pengujian perubahan intensitas cahaya lampu.....	72
Gambar 5.21. Hasil perhitungan sistem.....	74
Gambar 5.22. Diagram grafik hasil pengklasteran.....	75

DAFTAR TABEL

Tabel 2.1. Spesifikasi Arduino Pro Mini	11
Tabel 4.1. Rancangan Paket Data	29
Tabel 4.2. Keterangan Paket Data.....	29
Tabel 5.1. Penyimpanan di EEPROM Arduino	43
Tabel 5.2. Pembacaan di EEPROM Arduino.....	46
Tabel 5.3. Data masukkan Algoritma K-means	54
Tabel 5.4. Hasil <i>Centroid</i> perhitungan Manual.....	54
Tabel 5.5. Data masukkan.....	61
Tabel 5.6. Hasil centroid pada hari pertama	62
Tabel 5.7. Hasil centroid pada hari kedua.....	63
Tabel 5.8. Hasil centroid pada hari ketiga.....	64
Tabel 5.9. Hasil centroid pada hari keempat.....	65
Tabel 5.10. Input sistem	66
Tabel 5.11. Centroid Awal	66
Tabel 5.12. Proses Perhitungan pada iterasi pertama.....	67
Tabel 5.13. Centroid baru pada iterasi pertama	68
Tabel 5.14. Proses Perhitungan	69
Tabel 5.15. Centroid baru pada iterasi kedua.....	70
Tabel 5.16. Hasil <i>Centroid</i> perhitungan manual	71
Tabel 5.17. Data uji acak.....	73
Tabel 5.18. Hasil akhir perhitungan acak.....	75

BAB I

PENDAHULUAN

Bab ini berisi pembahasan tentang hal yang menjadi latar belakang topik, permasalahan yang akan dibahas dari topik, batasan dari analisis, tujuan dan manfaat dari topik yang akan dibahas, serta sistematika dalam penulisan skripsi.

1.1. Latar Belakang

Energi listrik merupakan salah satu kebutuhan pokok dalam kehidupan saat ini dimana hampir seluruh kegiatan menggunakan energi listrik, jumlah pembangkit energi listrik saat ini berbanding terbalik dengan konsumsi energi listrik yang semakin meningkat setiap harinya tanpa diimbangi dengan penambahan pembangkit energi listrik, selain itu penggunaan secara tidak bijak energi listrik sering kali terjadi dan mengakibatkan keborosan energi listrik, pemborosan yang sering terjadi adalah tingkah laku kebiasaan pengguna energi listrik, pemborosan itu kadangkala terjadi ketika menggunakan energi listrik bukan pada waktu yang tepat, Di Indonesia tarif dasar listrik yang terus meningkat memaksa berbagai pihak membuat sistem untuk penghematan konsumsi listrik dengan berbagai metode yang berbeda (Sherif, 2013). Seperti yang telah ketahui bahwa adanya krisis energi listrik di Indonesia mempengaruhi berbagai aspek strategis yang ada. Adanya fakta krisis energi listrik tersebut, mengakibatkan timbulnya upaya konkrit dari pemerintah untuk menemukan solusi dari permasalahan tersebut (Saputro, 2013), salah satu cara untuk mengatasi pemborosan dibidang energi listrik adalah dengan konsep teknologi *smart home*.

Teknologi *smart home* diharapkan selain cerdas dalam pengoperasiannya juga diharapkan hemat energi listrik, menurut penelitian yang berjudul "*Energi saving potential and strategies for electric lighting in future North European, low energi office buildings* " untuk penghematan energi untuk penerangan listrik digedung dapat dilakukan dengan perbaikan pada lampu (lampu hemat energi), *ballast* (Terang redup cahaya lampu) dan teknologi *luminer* (Dubois, 2011).

Otomatisasi dalam teknologi *smart home* sudah menjadi keharusan, bentuk otomatisasi suatu sistem dapat ditentukan dari data-data yang didapatkan dari masukan sebelumnya atau historisnya. Menurut penelitian yang berjudul “*Pattern Recognition Using Clustering Igorithm for Scenario Definition in Traffic Simulation-based Decision Support Sitem*s” dapat menentukan skenario atau prediksi waktu dan estimasi lalu lintas dari data historisnya, skenario atau prediksi yang akan datang menggunakan metode pengelompokan data atau klastering, klastering yang digunakan adalah K-means, dari hasil klastering menggunakan jarak *Euclidean* (Chen, 2014).

Teknologi *smart home* merupakan suatu sistem yang dapat digunakan untuk memonitor dan mengendalikan berbagai perangkat elektronik yang ada dirumah dari jarak jauh maupun secara otomatis. Salah satu sistem dalam *smart home* adalah pengendalian lampu secara otomatis untuk menghemat konsumsi energi listrik. Prinsip yang digunakan adalah *on/off* serta intensitas cahaya pada lampu yang dapat dikontrol melalui *remote control* secara *Wireless Sensor Network* (WSN) atau secara otomatis pada lampu, untuk mendapatkan otomatisasi tersebut dapat dilakukan dengan metode pembelajaran pada lampu. Metode ini dalam implementasinya memiliki suatu proses pembelajaran pada sistem untuk menentukan tindakan apa yang selanjutnya dilakukan oleh lampu (mati atau nyala dengan terang redup cahaya tertentu) berdasarkan data-data penggunaan lampu sebelumnya yang telah dipelajari. Proses pembelajaran lampu ini menggunakan metode algoritma K-means pada saat otomatisasi dilakukan, Algoritma K-means digunakan untuk memetakan pembelajaran dari masukan oleh *user* setiap hari, K-means yang digunakan dalam penelitian ini adalah K-means dua dimensi yaitu terdapat masukan waktu dan terang redup cahaya, proses kalkulasi pada algoritma K-means akan menghasilkan *centroid*, *centroid* inilah yang akan digunakan sebagai bentuk otomatisasi pada lampu.

1.2. Rumusan masalah

Berdasarkan uraian latar belakang, maka dapat dirumuskan permasalahan pada tugas akhir ini yaitu sebagai berikut:

- 1) Bagaimana membuat sistem otomatisasi mati-nyala lampu serta terang redup cahaya pada *smart home*?
- 2) Bagaimana mekanisme kerja atau algoritma pembelajaran untuk otomatisasi mati-nyala lampu serta terang redup cahaya pada *smart home*?

1.3. Batasan Masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian tugas akhir ini dibatasi dalam hal:

- 1) Lampu yang digunakan dalam penelitian ini berupa lampu pijar.
- 2) Modul arduino yang dipakai dalam penelitian ini adalah arduino pro mini 328/16Mhz.
- 3) Menggunakan modul NRF24L01 sebagai media komunikasi.
- 4) Jumlah klaster dalam penelitian ini dibatasi hanya 5 klaster, ini disebabkan karena jumlah memori dalam arduino pro mini hanya 1 kb, dengan memori tersebut hanya dapat menampung jumlah data sebanyak 20, dan apabila jumlah klaster lebih dari 5 atau lebih dari 5 data masukkan dalam 1 hari, maka semakin sedikit pula jumlah hari yang dijadikan data percobaan dalam penelitian ini.

1.4. Tujuan

Mengimplementasikan algoritma K-means pada salah satu sub *smart home* yaitu otomatisasi mati-nyala lampu.

1.5. Manfaat

Penelitian ini diharapkan dapat bermanfaat untuk berbagai pihak. Manfaat dari penelitian ini adalah sebagai berikut:

- 1) Manfaat Bagi Penulis
Sistem ini diharapkan dapat menghemat tenaga listrik dengan cara otomatisasi mati-nyalanya lampu.
- 2) Manfaat Bagi Pembaca
Mengetahui cara membuat sistem otomatisasi mati-nyala lampu.

1.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini adalah sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang permasalahan, identifikasi dan pembatasan masalah, rumusan masalah, tujuan dan manfaat, dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar teori

Menjelaskan teori dan referensi penelitian yang terkait dan memiliki tujuan dan perancangan yang hampir sama dengan penelitian ini.

BAB III Metode Penelitian

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, perancangan, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran dan Membahas analisis kebutuhan dan perancangan untuk pengujian perangkat keras (*hardware*).

BAB IV Perancangan dan Implementasi

Membahas tentang perancangan, analisis kebutuhan dan pengujian, serta implementasi dari algoritma K-means dan pada *hardware* sehingga dapat menunjukkan bahwa sistem dapat diimplementasikan dengan baik.

BAB V Pengujian dan Analisis

Memuat proses dan hasil pengujian terhadap metode yang telah direalisasikan dan melakukan analisis untuk mendapatkan kesimpulan

BAB VI Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian metode untuk pengembangan lebih lanjut.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan teori dan referensi penelitian yang terkait dan memiliki tujuan dan perancangan yang hampir sama dengan penelitian ini.

2.1. Kajian Pustaka

Teknologi *smart home* atau rumah *pintar* merupakan salah satu topik riset yang sedang populer didunia. *smart home system* adalah sebuah sistem berbantuan komputer yang akan memberikan segala kenyamanan, keselamatan, keamanan dan penghematan energi, yang berlangsung secara otomatis dan terprogram melalui komputer, pada gedung atau rumah tinggal.

Teknologi *smart home* diharapkan selain cerdas dalam pengoperasiannya juga diharapkan hemat energi listrik, menurut penelitian yang berjudul “*Energi saving potential and strategies for electric lighting in future North European, low energi office buildings* ” untuk penghematan energi untuk penerangan listrik digedung dapat dilakukan dengan perbaikan pada lampu (lampu hemat energi), *ballast* (Terang redup cahaya lampu) dan teknologi *luminer*. Dari hasil penelitian tersebut disimpulkan bahwa energi listrik dapat dihemat sampai dengan 40% dengan menggunakan pengaturan cahaya lampu atau *ballast* dan bisa menghemat hingga 80% penggunaan energi listrik dengan tambahan pola mematikan dan menyalakan lampu sesuai kebutuhan serta penggunaan teknologi *luminer*. Selain itu penghematan dapat dilakukan dengan berbagai macam cara diantaranya perbaikan dalam lampu, *ballast* dan teknologi *luminer*, penggunaan pencahayaan / *ambient*, peningkatan pemeliharaan dan faktor pemanfaatan atau kegunaan sehari hari, pengurangan dari mempertahankan tingkat pencahayaan dan jumlah switch yang tepat waktu serta penggunaan peredupan dan hunian sensor secara manual (Dubois, 2011).

Otomatisasi dalam teknologi *smart home* sudah menjadi keharusan, bentuk otomatisasi suatu sistem dapat ditentukan dari data-data yang didapatkan dari masukan sebelumnya atau historisnya. Menurut penelitian yang berjudul “*Pattern*

Recognition Using Clustering Algorithm for Scenario Definition in Traffic Simulation-based Decision Support Systems ” dapat menentukan skenario atau prediksi waktu dan estimasi lalu lintas dari data historisnya, skenario atau prediksi yang akan datang menggunakan metode pengelompokan data atau klastering, klastering yang digunakan adalah K-means, dari hasil klastering menggunakan jarak *Euclidean*. Dalam penelitian tersebut terdapat 100 data *training* berupa data kondisi cuaca, dari data tersebut diolah dengan metode K-means (*clustering non-hierarchical*) dan *hierarchical clustering*. Dari hasil perhitungan dari kedua metode tersebut didapatkan hasil bahwa metode K-means dengan metode *Euclidian distance* mampu memprediksi hasil waktu dan cuaca berdasarkan klasternya lebih baik dari pada metode *hierarchical clustering* (Chen, 2014).

Berdasarkan beberapa penelitian tersebut, penulis memperoleh ide untuk membuat sebuah sistem “**Implementasi Algoritma K-Means Pada Arduino Pro Mini Untuk Otomatisasi Terang Redup Cahaya Lampu**” yang dapat mengontrol mati dan nyala beserta terang redup cahaya lampu dengan metode pembelajaran sesuai dengan kebiasaan mematikan atau menyalakan lampu penggunaannya. Sistem ini dapat berjalan secara otomatis dengan memanfaatkan metode klastering dari data-data yang didapat dari kebiasaan pengguna menyalakan dan mematikan lampu pada kondisi sebelumnya.

2.2. Dasar Teori

Berdasarkan beberapa informasi dari beberapa kajian pustaka, maka dalam penulisan penelitian yang berjudul “Implementasi Algoritma K-Means Pada Arduino Pro Mini Untuk Otomatisasi Terang Redup Cahaya Lampu” terdapat beberapa dasar teori, antara lain:

2.2.1. Smart Home

Smart Home (Rumah Pintar) adalah sebuah teknologi terbaru yang mengimplementasikan sebuah sistem canggih disebuah rumah atau bangunan yang dapat berjalan otomatis dalam mengendalikan perangkat dan perlengkapan yang ada dirumah tersebut. Teknologi *smart home* diharapkan mampu terintegrasi

dengan *smart home* lain serta menciptakan kemudahan, kenyamanan, keamanan dan efisiensi bagi penggunaanya.

Teknologi *smart home* merupakan gabungan dari beberapa teknologi yang sudah ada dan saling terintegrasi satu dengan yang lainnya, antara lain teknologi *Wireless Sensor Network*(WSN), sensor canggih serta algoritma yang semakin cerdas. Sehingga teknologi *smart home* ini mampu dapat memonitor, mengontrol, memprediksi, mengantisipasi serta membuat keputusan dalam alur kerja beberapa sistem tersebut.

2.2.2. *Wireless Sensor Network* (WSN)

Wireless Sensor Network atau disingkat dengan WSN adalah suatu peralatan *sistem embedded* yang didalamnya terdapat satu atau lebih sensor dan dilengkapi dengan peralatan sistem komunikasi. Sensor disini digunakan untuk menangkap informasi sesuai dengan karakteristik informasi yang diinginkan.

Sebuah jaringan sensor merupakan sebuah insfraktur yang terdiri dari *sensing(masuring)*, komputasi dan elemen komunikasi yang memberikan administrator untuk melakukan instrumentsasi, mengamati, dan bereaksi terhadap fenomena dalam lingkungan tertentu. Administrator yang dimaksud adalah entitas sipil, komersial, ataupun industrial. Efisiensi daya di WSN umumnya dicapai dalam tiga cara yaitu:

- 1) Operasi *Low-duty-cycle*.
- 2) Proses lokal atau di sebuah jaringan untuk operasi mengurangi volume data karenanya transmisi waktu.
- 3) Jaringan multihop untuk mengurangi kebutuhan transmisi jarak jauh dan lebih hemat daya.

Pada saat ini ada berbagai jenis jaringan sensor nirkabel. *Wireless Sensor Network* saat ini dilengkapi dengan *transceiver radio* atau perangkat komunikasi nirkabel dan sumber energi yang biasanya berupa baterai. Ada empat komponen dasar dalam jaringan sensor:

- 1) Perakitan sensor secara distribusi atau lokal.
- 2) Jaringan interkoneksi.
- 3) Titik pusat dari informasi klustering.

- 4) Satu set sumber daya komputasi pada titik pusat (atau di luar) untuk menangani korelasi data.

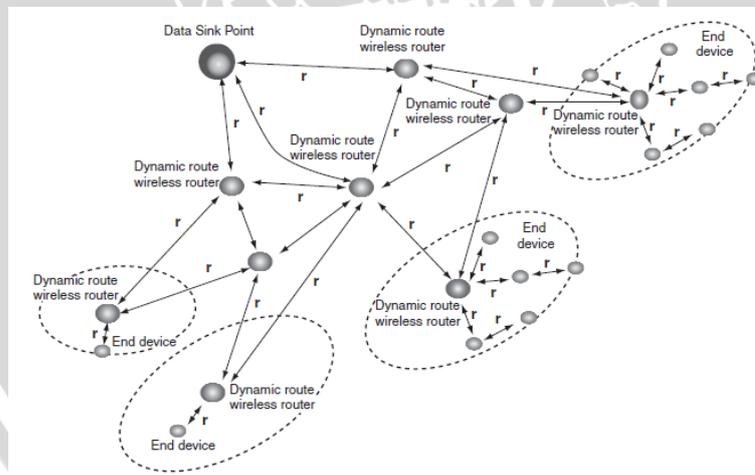
Kategori yang dipakai dalam *Wireless Sensor Network* (WSN) pada umumnya terbagi menjadi 2 kategori WSN, yaitu:

- 1) WSN Kategori 1 (C1WSNs)

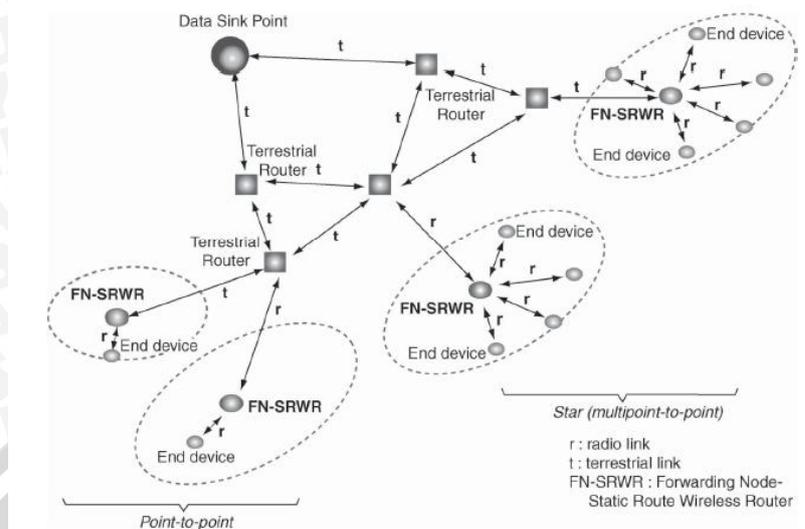
Pada kategori 1, jaringan menggunakan topologi mesh dengan jaringan radio multihop di kalangan atau diantara node pada WSN, dan memanfaatkan routing dinamis pada *wireless* dan kabel jaringan. C1WSNs merupakan jaringan dimana perangkat akhir (*end device*) yaitu sensor, diijinkan untuk lebih dari satu radio hop yang jauh dari *forwarding node*. Gambaran dari WSN kategori 1 dapat dilihat pada Gambar 2.1.

- 2) WSN Kategori 2 (C2WSNs)

Pada kategori 2, jaringan menggunakan topologi *star* (*point to point* atau *multipoint to point*) dengan jaringan radio *single-hop*, memanfaatkan *routing statis* melalui jaringan nirkabel. C2WSNs merupakan jaringan dimana perangkat akhir (*end device*) yaitu sensor, hanya terdapat satu *radio hop* yang jauh dari *forwarding node*. Gambaran dari WSN kategori 1 dapat dilihat pada Gambar 2.2.



Gambar 2.1. WSN Kategori 1
Sumber : (Sohraby, 2007)



Gambar 2.2. WSN Kategori 2

Sumber : (Sohraby, 2007)

2.2.3. K-MEANS

Klustering adalah proses membuat pengelompokan dari suatu data tertentu menjadi kelompok-kelompok (kluster) tertentu sesuai dengan persamaan berdasarkan matrik tertentu. Sebuah kluster adalah sekumpulan obyek yang yang bergabung sesuai dengan persamaan atau kedekatannya. Salah satu bentuk klustering adalah K-means.

K-means merupakan salah satu metode data *clustering non-hirarki* yang mengelompokan data dalam bentuk satu atau lebih kluster/kelompok. Data-data yang memiliki karakteristik yang sama dikelompokan dalam satu kluster/kelompok dan data yang memiliki karakteristik yang berbeda dikelompokan dengan kluster/kelompok yang lain sehingga data yang berada dalam satu kluster/kelompok memiliki tingkat variasi yang kecil (Y, 2007).

Menurut Santosa (Santosa, 2007) langkah-langkah melakukan klustering dengan metode K-Means adalah sebagai berikut:

- 1) Pilih jumlah kluster k.
- 2) Inisialisasi k pusat kluster ini bisa dilakukan dengan berbagai cara. Namun yang paling sering dilakukan adalah dengan cara acak. Pusat-pusat kluster diberiduberi nilai awal dengan angka-angka acak,

- 3) Alokasikan semua data/ objek ke kluster terdekat. Kedekatan dua objek ditentukan berdasarkan jarak kedua objek tersebut. Demikian juga kedekatan suatu data kekluster tertentu ditentukan jarak antara data dengan pusat kluster. Dalam tahap ini perlu dihitung jarak tiap data ke tiap pusat kluster. Jarak paling antara satu data dengan satu kluster tertentu akan menentukan suatu data masuk dalam kluster mana. Untuk menghitung jarak semua data ke setiap titik pusat kluster dapat menggunakan teori jarak *Euclidean* yang dirumuskan sebagai berikut:

$$4) D(i, j) = \sqrt{(X_{1i} - X_{1j})^2 + (X_{2i} - X_{2j})^2 + \dots + (X_{ki} - X_{kj})^2}$$

Dimana:

- - $D(i, j)$ adalah Jarak data ke i ke pusat kluster j
 - X_{kj} adalah Data ke i pada atribut data ke k
 - X_{ki} adalah Titik pusat ke j pada atribut ke k
- 5) Hitung kembali pusat kluster dengan keanggotaan kluster yang sekarang. Pusat kluster adalah rata-rata dari semua data/ objek dalam kluster tertentu. Sesuai dengan rumus berikut (Nur, 2012)

$$\bar{v}_{ij} = \frac{1}{N_i} \sum_{k=0}^{N_i} x_{kj}$$

Dimana : V_{ij} = Centroid / rata – rata kluster ke- i untuk variabel ke- j

N_i = Jumlah data yang menjadi anggota kluster ke- i

i, k = indek dari kluster

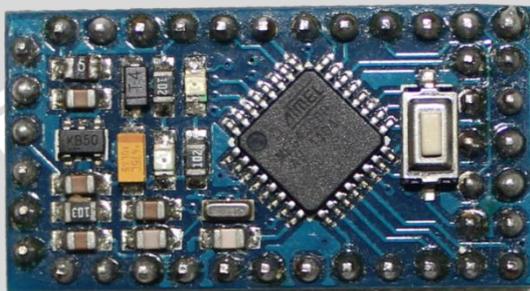
j = indek dari variabel

X_{ij} = nilai data ke- k yang ada dalam kluster tersebut untuk variabel ke- j

- 6) Tugaskan lagi setiap objek memakai pusat kluster yang baru. Jika pusat kluster tidak berubah lagi maka proses klustering selesai. Atau, kembali ke langkah nomor 3 sampai pusat kluster tidak berubah lagi.

2.2.4. Arduino Pro Mini

Arduino Pro Mini adalah board berbasis mikrokontroler ATmega328. Board ini memiliki 14 digital input / output pin (6 pin dapat digunakan sebagai output PWM), 8 pin analog, 16 Mhz osilator Kristal serta tombol reset. Untuk menulis program dari komputer ke arduino promini digunakan FTDI. Bentuk fisik arduino pro mini ditunjukkan dalam Gambar 2.3. Spesifikasi arduino promini ditunjukkan dalam Tabel 2.1.



Gambar 2.3. Board Arduino Pro Mini

Tabel 2.1. Spesifikasi Arduino Pro Mini

Microcontroller	Atmega328
Operating Voltage	5V (<i>depending on model</i>)
Input Voltage	5 - 12 V (5V model)
Digital I/O Pins	14 (<i>of which 6 provide PWM output</i>)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (<i>of which 2 KB used by bootloader</i>)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz (5V model)

Arduino pro mini memiliki fitur-fitur sebagai berikut:

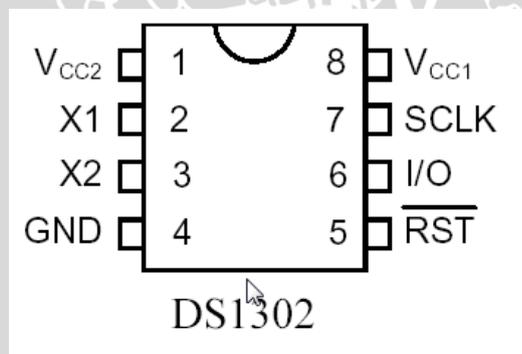
- 1) Serial: Pin yang digunakan untuk komunikasi Serial adalah Pin 0 sebagai TX dan pin 1 sebagai RX
- 2) I²C: pin yang digunakan untuk komunikasi I²C adalah pin SDA(A4) dan SDL (A5).

- 3) Pin Analog: Arduino Pro mini memiliki 6 pin analog yaitu A0, A1, A2, A3, A4, dan A5.
- 4) Pin Digital: ArduinoPro mini memiliki 14 pin digital yaitu pin 0 sampai pin 13.
- 5) SPI: Pin yang digunakan adalah 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).
- 6) PWM: Arduino pro mini sudah memiliki fitur PWM yaitu menggunakan pin 3, 5, 6, 9, 10, dan 11 dengan format 8 bit.
- 7) *Interrupts external*: pin yang digunakan adalah pin 2 dan pin 3.

2.2.5. Real Time Clock (RTC)

Real Time Clock (RTC) berfungsi sebagai pewaktu *digital*, pada RTC penelitian ini menggunakan IC DS1302 untuk menyimpan data jam, menit, detik, hari, bulan, dan tahun. RTC ini dilengkapi dengan baterai internal sebagai *backup* apabila catu daya dari mikrokontroler mati atau terputus.

DS1302 berkomunikasi dengan mikrokontroler melalui 3-wire (SCLK, I/O, CE). Agar dapat bekerja, IC DS1302 memerlukan *clock external* 32,768kHz. Konfigurasi *pin* dari IC DS1302 ditunjukkan pada Gambar 2.4.

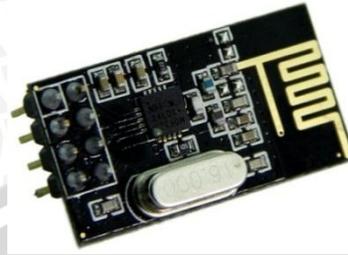


Gambar 2.4. Konfigurasi *Pin* DS1302
Sumber: (Maxim Integrated, 2008)

2.2.6. NRF24L01

NRF24L01 adalah sebuah modul komunikasi jarak jauh dengan *single-chip RF-transceiver* yang ditujukan untuk aplikasi pada gelombang 2.4 GHz ISM band. Modul NRF24L01 menggunakan antar muka SPI (*Serial Peripheral*

Interface) untuk komunikasi dengan tegangan kerja 3,3Volt DC. Bentuk fisik dari modul RF NRF24L01 ditunjukkan pada Gambar 2.5.



Gambar 2.5. NRF24L01
Sumber: (Nordic Semiconductor, 2008)

2.2.6.1. Spesifikasi pada modul NRF24L01

Modul NRF24L01 memiliki beberapa spesifikasi yang menunjang untuk penerapan sistem *smart home*, diantaranya sebagai berikut:

- 1) *Worldwide 2.4GHz ISM band operation.*
- 2) Jangkauan data di udara yaitu 250kbps, 1Mbps and 2Mbps.
- 3) *Ultra low power operation.*
- 4) 11.3Ma TX at 0dBm *Output Power.*
- 5) 13.5Ma RX at 2Mbps *air data rate.*
- 6) *Power turun pada 900Na .*
- 7) 26Ma in standby-I.
- 8) *On chip voltage regulator.*
- 9) 1.9 to 3.6V supply range.
- 10) *Enhanced ShockBurst™.*
- 11) *Automatic packet handling.*
- 12) *Auto packet transaction handling.*
- 13) 6 data pipe MultiCeiver™.
- 14) Drop-in compatibility with Nrf24L01, Nrf24E1 dan Nrf24E2.
- 15) Low cost BOM.
- 16) ±60ppm 16MHz *crystal.*
- 17) 5V *tolerant inputs.*
- 18) Compact 20-pin 4x4mm QFN *package.*

2.2.6.2. Fitur-fitur pada modul NRF24L01

Modul NRF24L01 memiliki beberapa fitur yang Fitur yang terdapat pada modul NRF24L01 antara lain:

1) Radio:

Worldwide 2.4GHz ISM band operation.

126 RF channels.

Common RX and TX interface.

GFSK modulasi.

250kbps, 1 and 2Mbps air data rate.

1MHz non-overlapping channel spacing at 1Mbps.

2MHz non-overlapping channel spacing at 2Mbps.

2) Transmitter:

Pemrograman daya output: 0, -6, -12 or -18dBm.

11.3Ma at 0dBm output power.

3) Receiver:

AGC yang cepat untuk meningkatkan jangkauan dinamis.

Penyaringan channel terintegrasi.

13.5Ma at 2Mbps.

82dBm sensitivity at 2Mbps, 85dBm sensitivity at 1Mbps.

94dBm sensitivity at 250kbps.

4) RF Synthesizer:

Fully integrated synthesizer.

Tidak ada loop filer external, VCO varactor diode atau resonator.

Accepts low cost ± 60 ppm 16MHz crystal.

5) *Enhanced ShockBurst™:*

1 to 32 bytes dynamic payload length.

Penanganan paket otomatis.

Auto packet transaction handling.

6 data pipe MultiCeiver™ untuk 1:6 jaringan star.

6) *Power Management:*

Integrated voltage regulator.

1.9 to 3.6V supply range.

Idle modes with fast start-up times for advanced power management.

26Ma Standby-I mode, 900Na power down mode.

Max 1.5ms start-up from power down mode.

Max 130us start-up from standby-I mode.

7) *Host Interface:*

4-pin SPI.

Maksimal 10Mbps.

32 bytes TX and RX FIFOs.

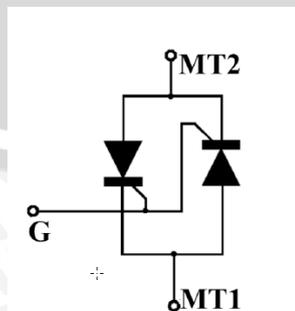
5V tolerant inputs.

Compact 20-pin 4x4mm QFN package.

2.2.7. TRIAC

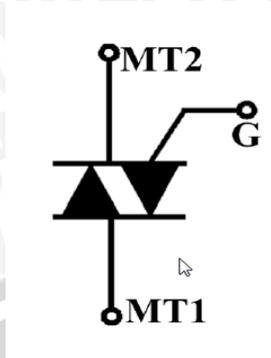
TRIAC merupakan singkatan dari *Triode Alternating Current*, yang artinya adalah saklar triode untuk arus bolak-balik. TRIAC adalah pengembangan dari pendahulunya yaitu DIAC dan SCR. Ketiganya merupakan sub-jenis dari *Thyristor*, piranti berbahan silikon yang umum digunakan sebagai saklar elektronik, disamping transistor dan FET. Perbedaan diantara ketiganya adalah dalam penggabungan unsur-unsur penyusunnya serta dalam segi arah penghantaran arus listrik yang melaluinya.

TRIAC sebenarnya adalah gabungan dua buah SCR (*Silicon Controlled Rectifier*) atau *Thyristor* yang dirancang anti paralel dengan 1 (satu) buah elektroda gerbang (*gate electrode*) yang menyatu. SCR merupakan piranti zat padat (*solid state*) yang berfungsi sebagai sakelar daya berkecepatan tinggi. Rangkaian ekuivalen dari TRIAC ditunjukkan pada Gambar 2.6. dan simbol dari komponen TRIAC ditunjukkan pada Gambar 2.7.



Gambar 2.6. Rangkaian ekuivalen TRIAC.

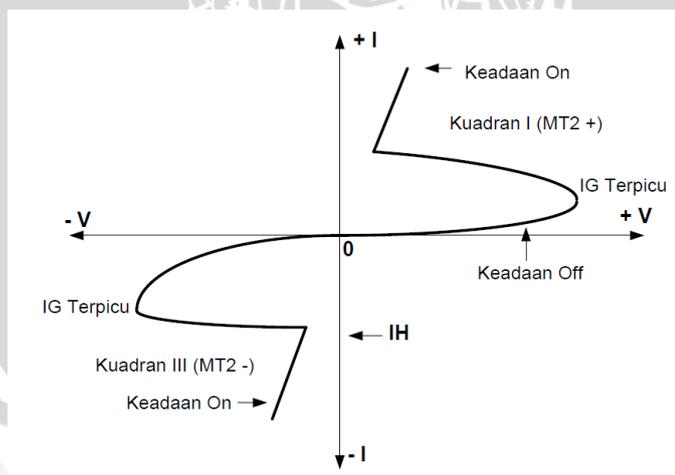
Sumber: (Semiconductors, Philips, 2001)



Gambar 2.7. Simbol TRIAC.
 Sumber: (Semiconductors, Philips, 2001)

TRIAC memiliki karakteristik *switching* seperti pada SCR, kecuali bahwa TRIAC dapat berkonduksi dalam berbagai arah. TRIAC dapat digunakan untuk mengontrol aliran arus dalam rangkaian AC. Elemen seperti penyearah dalam kedua arah menunjukkan kemungkinan dua aliran arus antara terminal utama MT1 dan MT2. Pengaturan dilakukan dengan menerapkan sinyal antara *gate* (gerbang) dan MT1.

Keadaan *On* Kuadran I (MT2 +) IG Terpucu Keadaan *Off* Keadaan *On* Kuadran III (MT2 -) IG Terpucu - V + V + I - I 0 IH (seperti karakteristik triac yang ditunjukkan pada Gambar 2.8.)



Gambar 2.8. Karakteristik TRIAC.
 Sumber: (Semiconductors, Philips, 2001)

Karena dapat bersifat konduktif dalam dua arah, biasanya TRIAC digunakan untuk mengendalikan fasa arus AC (contohnya kontroler tegangan AC). Selain itu, karena TRIAC merupakan *devais bidireksional*, terminalnya tidak

dapat ditentukan sebagai *anode* atau *katode*. Jika terminal MT2 positif terhadap terminal MT1, TRIAC dapat dimatikan dengan memberikan sinyal gerbang positif antara gerbang G dan MT1. Sebaliknya jika terminal MT2 negatif terhadap MT1 maka TRIAC akan dapat dihidupkan dengan memberikan sinyal pulsa negatif antara gerbang G dan terminal MT1. Tidak perlu untuk memiliki kedua sinyal gerbang positif dan negatif dan TRIAC akan dapat dihidupkan baik dengan sinyal positif atau negatif. Dalam prakteknya sensitifitas bervariasi antara satu kuadran dengan kuadran lain, dan TRIAC biasanya beroperasi di kuadran I (tegangan dan arus gerbang positif) atau kuadran III (tegangan dan arus gerbang negatif).

2.2.8. *Optocoupler* / *Optoisolator* MOC3021

Optocoupler digunakan untuk mengisolasi antara rangkaian AC dengan Rangkaian DC. Berdasarkan arti katanya *Opto* berarti *optic* dan *coupler* (kopling), berarti pemicu sehingga bias diartikan *optocoupler* bekerja berdasarkan picu cahaya *optic* dan digunakan sebagai saklar elektrik yang bekerja secara otomatis. *Optocoupler* terdiri dari berbagai jenis (baik merek, bentuk dan tipe). *Optocoupler* terdiri dari dua bagian yaitu:

- 1) Bagian *transmitter* yang dibangun dari sebuah LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.
- 2) Bagian *receiver* yang dibangun dengan dasar komponen *photodiode*. *Photodiode* merupakan suatu transistor yang peka terhadap tenaga cahaya. Suatu sumber cahaya menghasilkan energi panas, begitu pula dengan spectrum infra merah. Karena spectrum infra merah mempunyai efek panas yang lebih besar dari cahaya tampak, maka *photodiode* lebih peka untuk menangkap radiasi dari sinar infra merah.

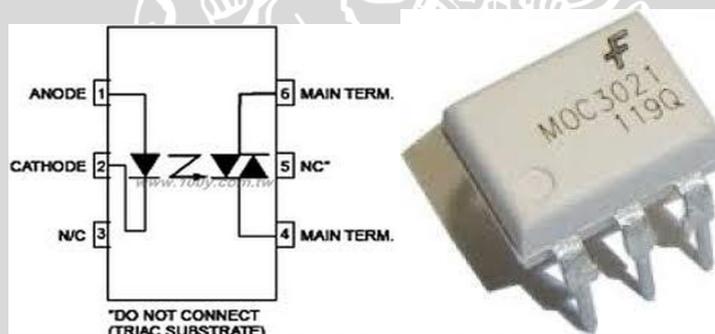
Oleh karena itu, *optocoupler* dapat dikatakan sebagai gabungan dari LED infra merah dengan *photodiode* yang terbungkus menjadi satu chip. LED infra merah ini merupakan komponen elektronika yang memancarkan cahaya infra

merah dengan konsumsi daya sangat kecil. Jika diberi bias maju, LED infra merah yang terdapat pada *optocoupler* akan mengeluarkan panjang gelombang sekitar 0,9 mikrometer. *Photodiode* merupakan komponen elektronika yang berfungsi sebagai detektor cahaya infra merah.

Prinsip kerja dari *optocoupler* adalah:

- 1) Jika antara *photodiode* dan LED terhalang maka *photodiode* tersebut akan *off* sehingga *output* dari kolektor akan berlogika *high*.
- 2) Sebaliknya jika antara *photodiode* dan LED tidak terhalang maka *photodiode* tersebut akan *on* sehingga keluarannya akan berlogika *low*.

IC MOC 3021 memiliki 6 kaki, kaki *anoda* (1), kaki *katoda* (2), kaki (4) dan (6) dihubungkan ke beban dan kaki 3 dan 5 tidak digunakan (*not connected*) Seperti gambar 2.9.. *Optocoupler* bisa juga dikatakan sebagai *driver TRIAC* karena secara tidak langsung, *optocoupler* yang memberikan pemicu pada kaki *gate TRIAC*.



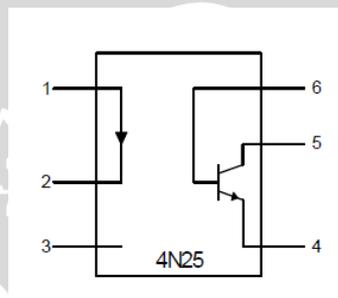
Gambar 2.9. Konfigurasi (kiri) dan Bentuk fisik (kanan) MOC3021
Sumber: (Instruments, Texas, 1996)

2.2.9. OptoCoupler 4N25

Optocoupler 4n25 adalah merupakan komponen elektronik *optoisolator* yang terdiri dari pemancar cahaya atau *emitter* yang mengkopel secara optik terhadap *photo detector* melalui media yang terisolasi. Pemancar cahaya dapat berupa penerang lampu ataupun LED. Media isolasi berupa udara, plastik, gelas atau fiber. Sedangkan *photo detector* dapat berupa photo konduktor, *photo dioda*, *photo transistor*, *photo SCR* atau rangkaian *photo dioda/amplifier*. Mengenai

pengontrolan pemancaran cahaya dan *photo detector* memungkinkan pemindahan informasi dari suatu rangkaian yang mengandung pemancar cahaya ke rangkaian yang mengandung *photo detector*. Informasi dilewatkan secara optik melintasi celah isolasi yang perpindahannya memiliki sistem satu arah sehingga *photo detector* tidak mempengaruhi rangkaian *input*. Isolasi optik mencegah adanya interaksi ataupun kerusakan rangkaian *input* yang disebabkan oleh perbedaan tegangan yang relatif tinggi terhadap rangkaian output.

Bentuk fisik dari kemasan optocoupler 4N25 terdiri dari 6 *pin* atau kemasan *dual-inline*. Konfigurasi ini *pin* 1 dan 2 umumnya dihubungkan ke pemancar cahaya, sedangkan *pin* 4,5 dan 6 dihubungkan ke *photo detector*. Konfigurasi dari optocoupler 4N25 seperti gambar 2.10.



Gambar 2.10. Optocoupler tipe 4N25
Sumber: (Semiconductors, Vishay, 2010)

Optocoupler dirancang untuk menggantikan fungsi saklar mekanis dan perubahan pulsa secara fungsional opto coupler sama dengan pasangan relay mekanis karena suatu isolasi tingkat tinggi diantara terminal *input* dan *output*.

Beberapa keunggulan *optocoupler* adalah:

- 1) Kecepatan operasi lebih cepat.
- 2) Ukuran kecil.
- 3) Tidak mudah dipengaruhi getaran dan guncangan.
- 4) Respon frekuensi.
- 5) Tidak ada *bounce*.
- 6) Kompatibel dengan banyak rangkaian logika dan mikroprosesor.

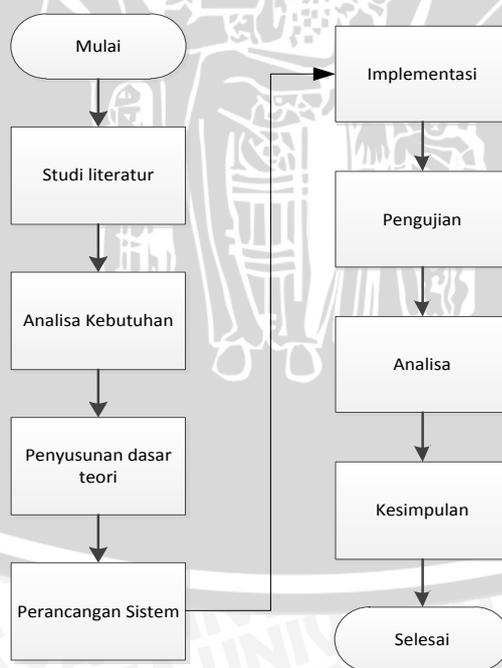
BAB III

METODOLOGI PENELITIAN

Bab ini membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, perancangan, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran dan Membahas analisis kebutuhan dan perancangan untuk pengujian perangkat keras (*hardware*).

3.1. Metodologi Penelitian

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam pengerjaan tugas akhir, yaitu studi literatur, penyusunan dasar teori, analisis dan perancangan perangkat keras, implementasi, analisis dan pengujian dari algoritma yang akan dibuat, hingga penulisan laporan. Kesimpulan dan saran disertakan sebagai catatan atas performa dari penelitian ini dan kemungkinan arah pengembangan penelitian selanjutnya. Berikut ini merupakan diagram alir runtutan pengerjaan penelitian ini:



Gambar 3.1. Flowchart Runtutan Pengerjaan Penelitian

3.1.1. Studi Literatur

Studi literatur mempelajari mengenai penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut diperoleh dari buku, jurnal, *e-book*, dan dokumentasi project.

3.1.2. Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan oleh sistem yang akan dibangun dan diuji. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat didalamnya. Berikut analisis kebutuhan dalam penelitian yang akan dilakukan.

3.1.2.1. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

- 1) Arduino Pro mini dan *Board* PCB digunakan sebagai kalkulasi algoritma.
- 2) Rangkaian *Dimmer* sebagai aktuator dari penelitian ini.
- 3) Rangkaian RTC sebagai *clock* waktu.
- 4) Lampu pijar.

3.1.2.2. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam penelitian ini adalah sebagai berikut:

- 1) Arduino IDE.
- 2) Microsoft Excel.

3.1.3. Penyusunan Dasar Teori

Penyusunan dasar teori dilakukan setelah mendapatkan referensi yang tepat untuk mendukung penulisan penelitian ini. Teori-teori pendukung tersebut meliputi:

- 1) *Smart Home*.
- 2) *Wireless Sensor Network* (WSN).
- 3) K-means klustering.

- 4) Arduino Pro mini.
- 5) NRF24L01.
- 6) Konsep *Optocoupler*.
- 7) Konsep TRIAC.

3.1.4. Perancangan Sistem

Pada perancangan sistem ini dibentuk dalam diagram blok yang diagram blok tersebut memiliki bagian-bagian dari beberapa sistem yang dibuat. Bagian-bagian dari diagram blok tersebut mempunyai tugas yang berbeda sehingga dari beberapa blok tersebut menjadi sebuah sistem yang utuh sesuai dengan tujuan penelitian ini. Diantara distem tersebut terdapat sistem pengaturan terang redup cahaya atau *dimmer*, *real time clock* (RTC), NRF24I01 dan tentunya Arduino pro mini.

3.1.5. Implementasi

Implementasi sistem ini dilakukan dengan mengacu pada perancangan sistem. Implementasi perangkat keras dan perangkat lunak Implementasi meliputi:

- 1) Perancangan Rangkaian *Dimmer*.
- 2) Perancangan Rangkaian RTC.
- 3) Pembuatan dan Perancangan algoritma K-means.
- 4) Perancangan paket data.
- 5) Perancangn algoritma penyimpanan dan pembacaan EEPROM arduino.

3.1.6. Pengujian

Pengujian pada sistem dilakukan dengan mengacu pada perancangan sistem, yang meliputi:

- 1) Pengujian penyimpan dan pembacaan data EEPROM arduino.
- 2) Pengujian komunikasi data dari arduino 1 dengan arduino 2 melalui NRF24L01.
- 3) Pengujian rangkaian *Dimmer*.
- 4) Pengujian rangkaian RTC.
- 5) Pengujian algoritma K-means.

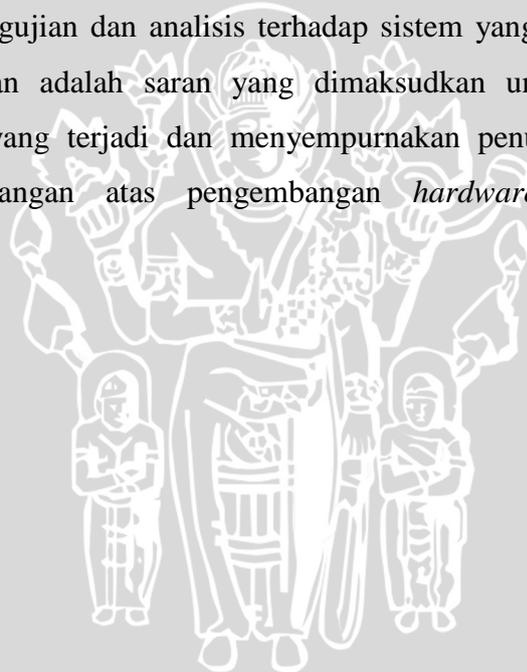
6) Pengujian Keseluruhan Sistem.

3.1.7. Analisis

Pengujian yang dilakukan, data dari masing-masing tahap pengujian dan analisis untuk mengetahui hasil yang akan dilakukan untuk menarik kesimpulan dari penelitian yang dilakukan. Hasil pengujian dianalisis untuk menentukan apakah sistem sudah berjalan baik sesuai dengan tujuan penulisan.

3.1.8. Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan atas pengembangan *hardware* dan algoritma selanjutnya.



BAB IV

PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas tentang perancangan, analisis kebutuhan dan pengujian, serta implementasi dari algoritma K-means dan pada *hardware* sehingga dapat menunjukkan bahwa sistem dapat diimplementasikan dengan baik.

4.1. Perancangan

Tahap perancangan berisi tentang analisis perangkat keras dan perangkat lunak yang digunakan untuk membuat sistem tersebut. Ada beberapa tahap di antaranya sebagai berikut:

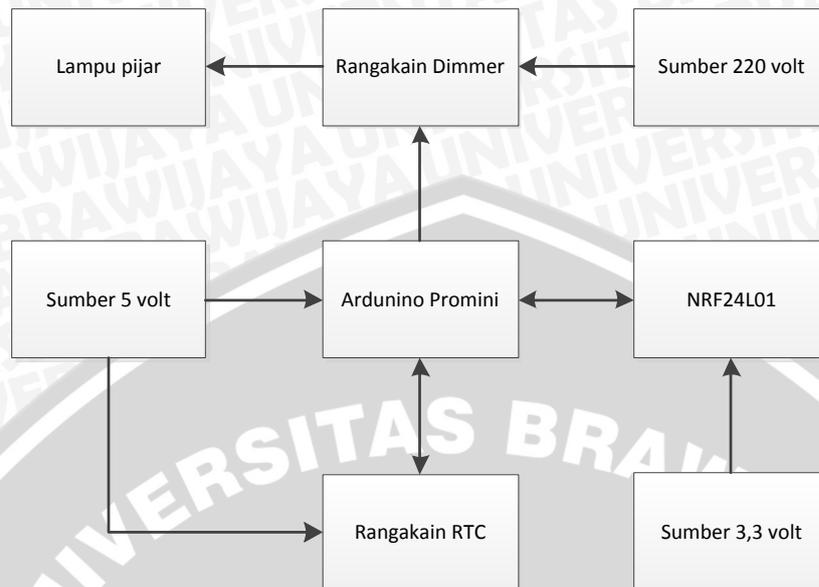
4.1.1. Analisis Kebutuhan Perangkat Keras dan Lunak

Analisis kebutuhan yang dibutuhkan seperti perangkat keras dan perangkat lunak yang akan digunakan untuk proses implementasi dari sistem yang ada. Di antaranya kebutuhan yang dibutuhkan yaitu:

- 1) Arduino dan NRF24L01.
- 2) Rangkaian *Dimmer*.
- 3) Rangkaian RTC.
- 4) Arduino IDE.
- 5) Microsoft Excel.

4.1.2. Diagram Blok Sistem

Dari analisis kebutuhan yang ada dapat kita buat diagram blok dari sistem tersebut yang terdiri dari bagian-bagiannya. Diagram blok untuk sistem ini yaitu seperti berikut:



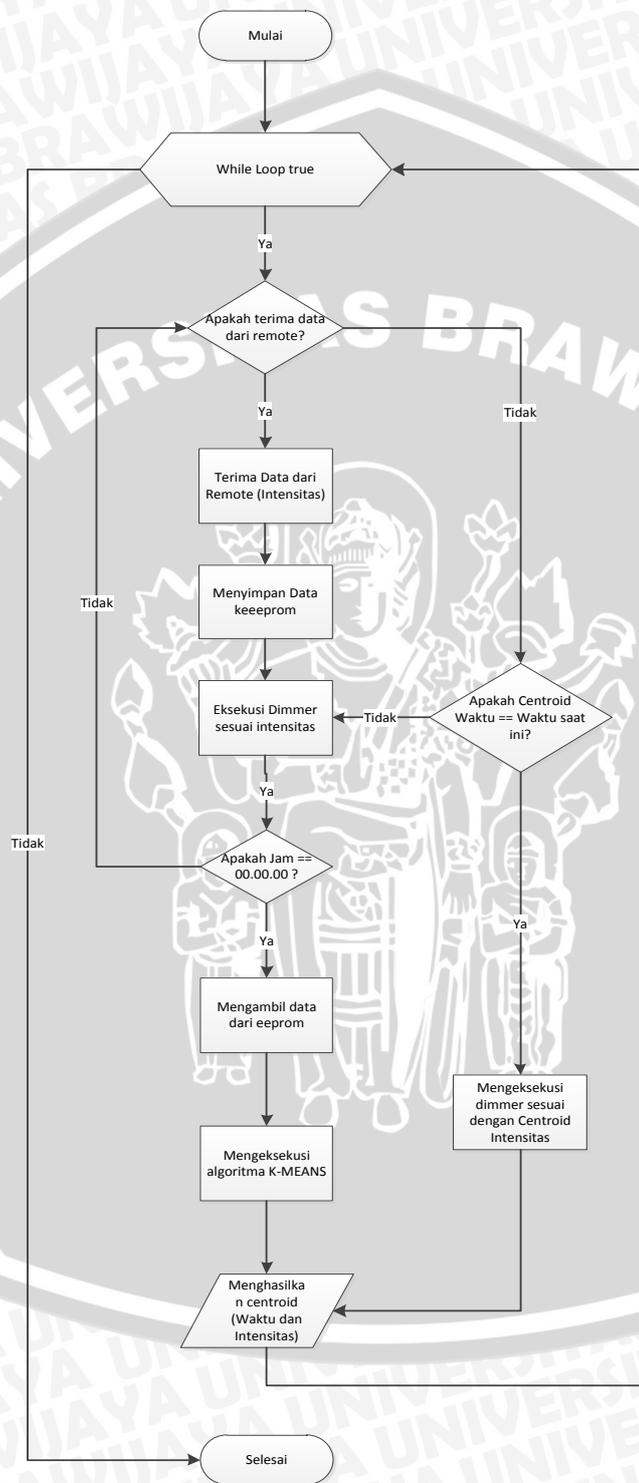
Gambar 4.1. Diagram Blok Sistem

Pada Gambar 4.1. diagram blok menggunakan *input* berasal dari data yang dikirimkan melalui modul NRF24L01 yang selanjutnya data waktu dan intensitas cahaya dituliskan di EEPROM arduino lalu diproses pada algoritma K-means dan pada proses otomatis dari data *centroid* dari perhitungan K-means dieksekusi kerangkaian *dimmer* dan setelah itu mengirimkan informasi ke *server* atau *base station*

4.1.3. Diagram Alir Sistem

Pada penitian ini alur proses berjalannya sistem ini berawal dari sistem ini menerima data dari *remote control* berupa nilai terang redup cahaya, selanjutnya nilai terang redup cahaya tersebut langsung dieksekusi keaktuator *dimmer* dan nilai dari terang redup cahaya dan waktu pengiriman data tersebut disimpan kedalam EEPROM sistem, dan proses tersebut berulang mengeksekusi nilai *dimmer* sesuai dengan masukkan dari *remote control* terakhir. Bergantinya nilai terang redup cahaya pada lampu akan berganti apabila ada waktu sekarang menunjukkan waktu yang sama dengan hasil perhitungan algoritma K-means dan perubahan nilai terang redup cahaya mengikuti nilai dari *centroid* yang dihasilkan.

Proses perhitungan algoritma K-means ini dilakukan 1 kali setiap hari yaitu pukul 00.00.00. Diagram alir proses pada penelitian ini ditunjukkan pada gambar 4.2.



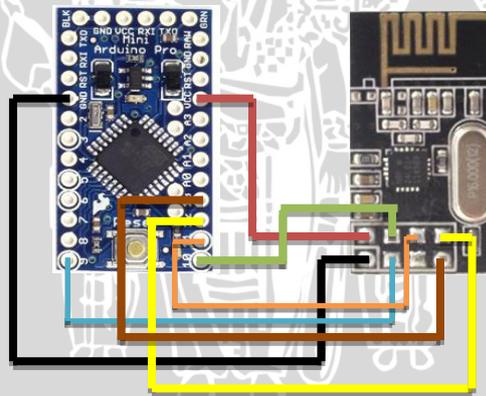
Gambar 4.2. Diagram Alir Sistem

4.1.4. Perancangan Perangkat Keras

Pada perancangan perangkat keras terdapat beberapa perancangan, yaitu antara lain: perancangan *hardware*. Perancangan *hardware* yang akan diimplementasikan berdasarkan analisis kebutuhan sistem. Diantaranya Arduino Pro Mini, NRF24L01, rangkaian *dimmer*, rangkaian RTC.

4.1.5. Perancangan Arduino dan NRF24L01

Rangkaian media komunikasi *radio frekuensi* (RF) NRF24L01 dan arduino digunakan sebagai media pengiriman data atau perintah dari pengguna untuk menyalakan atau mematikan lampu dari jarak jauh. Protokol yang digunakan arduino dan NRF24L01 untuk komunikasi adalah *Serial Peripheral interface* (SPI), maka *pin* yang dipakai adalah *pin* yang digunakan adalah 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Serta dua *pin* tambahan yaitu *pin* 9 untuk *Clock Enable* dan *pin* 10 untuk CSN. Skematik rangkaian arduino dan NRF24L01 ditunjukkan pada Gambar 4.3.

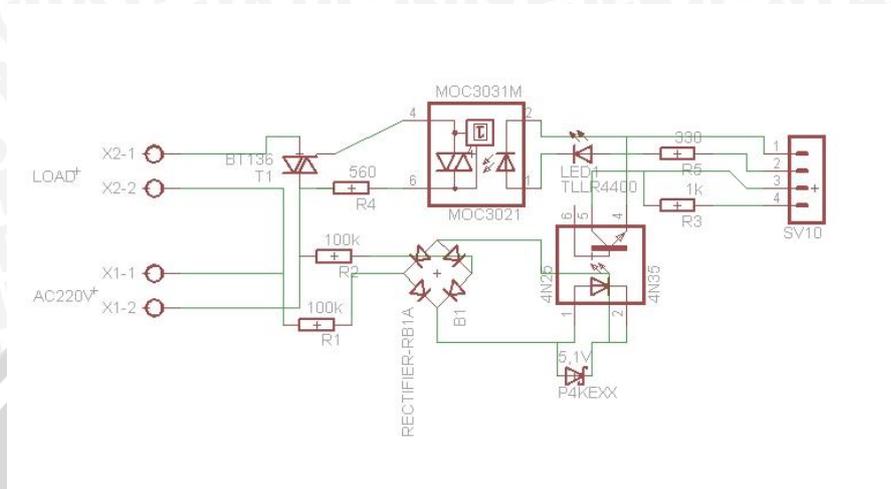


Gambar 4.3. Rangkaian Arduino ke NRF24L01

4.1.6. Perancangan Rangkaian Dimmer

Rangkaian *dimmer* adalah sebuah rangkaian untuk mengontrol terang redup cahaya lampu pijar, komunikasi rangkaian ini menggunakan *interrupt* serta *pin digital*. *Pin* interrupt pada arduino pro mini menggunakan *pin* 2 dan *pin* 3 untuk *pin digital* rangkaian ini. Cara kerja rangkaian ini menggunakan fungsi *delay* waktu dengan satuan *microsecond* untuk menyalakan dan mematikan *pin* 3 yang terhubung pada *optocoupler*. Rangkaian ini juga menggunakan TRIAC

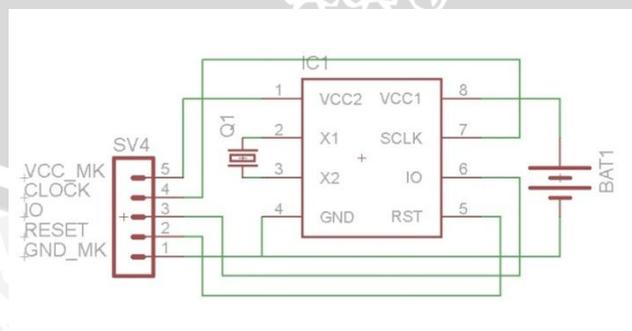
sebagai kontroler Lampu dengan tegangan 220v. Skematik rangkaian *dimmer* pengatur terang redup cahaya ditunjukkan pada Gambar 4.4.



Gambar 4.4. Rangkaian *Dimmer* lampu

4.1.7. Perancangan Rangkaian RTC

Rangkaian Real Time Clock (RTC) digunakan untuk menghitung dan menyimpan waktu mulai dari jam, menit, detik sesuai dengan *clock* waktu yang sebenarnya. Pada rangkaian ini menggunakan IC DS1302. Pada IC DS 1302 ini menggunakan 3 *pin* untuk berkomunikasi dengan arduino pro mini, *Pin* arduino yang dipakai pada rangkaian ini adalah *pin* 4 yang terhubung ke *pin* Chip Enable (CE) pada IC DS1302, *pin* 5 yang terhubung ke *pin* Input/Output (I/O) pada IC DS1302 dan *pin* 6 yang terhubung ke *pin* Serial Clock (SCLk) pada IC DS1302. Skematik rangkain RTC ditunjukkan pada Gambar 4.5.



Gambar 4.5. Rangkaian RTC

4.1.8. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak terdapat beberapa perancangan, yaitu antara lain: perancangan paket data, perancangan Algoritma K-means, perancangan penyimpanan dan pembacaan EEPROM arduino.

4.1.9. Perancangan Paket data

Perancangan paket data dapat dilihat pada Tabel 4.1. dan Tabel 4.2. perancangan paket data dimaksudkan untuk memberikan batasan data apa saja dikirim dari node dan diterima oleh *router* yang selanjutnya akan dipecah oleh aplikasi untuk memisah data tiap *byte* agar dapat dengan mudah disesuaikan dengan data base.

Tabel 4.1. Rancangan Paket Data

Panjang Data	1 <i>Byte</i>										
Nama Slot	<i>Header</i>	R/B	R/W	F_ID	To_ID	Jam	Menit	Detik	Nilai	GP	<i>Check sum</i>

Tabel 4.2. Keterangan Paket Data

No	Nama Slot	Deskripsi
1.	<i>Header</i>	Paket data ini digunakan untuk mendeteksi awal data
2.	R/B	Nilai pada paket ini mengganti status <i>node</i> apakah <i>ready</i> atau <i>busy</i> . Nilainya diisi 1 jika <i>busy</i> , diisi 0 jika <i>ready</i>
3.	R/W	Nilai pada paket ini diisi 1 untuk operasi <i>write</i> , dan diisi 0 untuk operasi <i>read</i>
4.	F_ID	Slot ini berisi informasi ID pengirim
5.	To_ID	Slot ini berisi informasi ID penerima
6.	Jam	Variabel ini khusus untuk node sensor yang menggunakan RTC untuk menyimpan waktu
7.	Menit	Variabel ini khusus untuk <i>node</i> sensor yang menggunakan

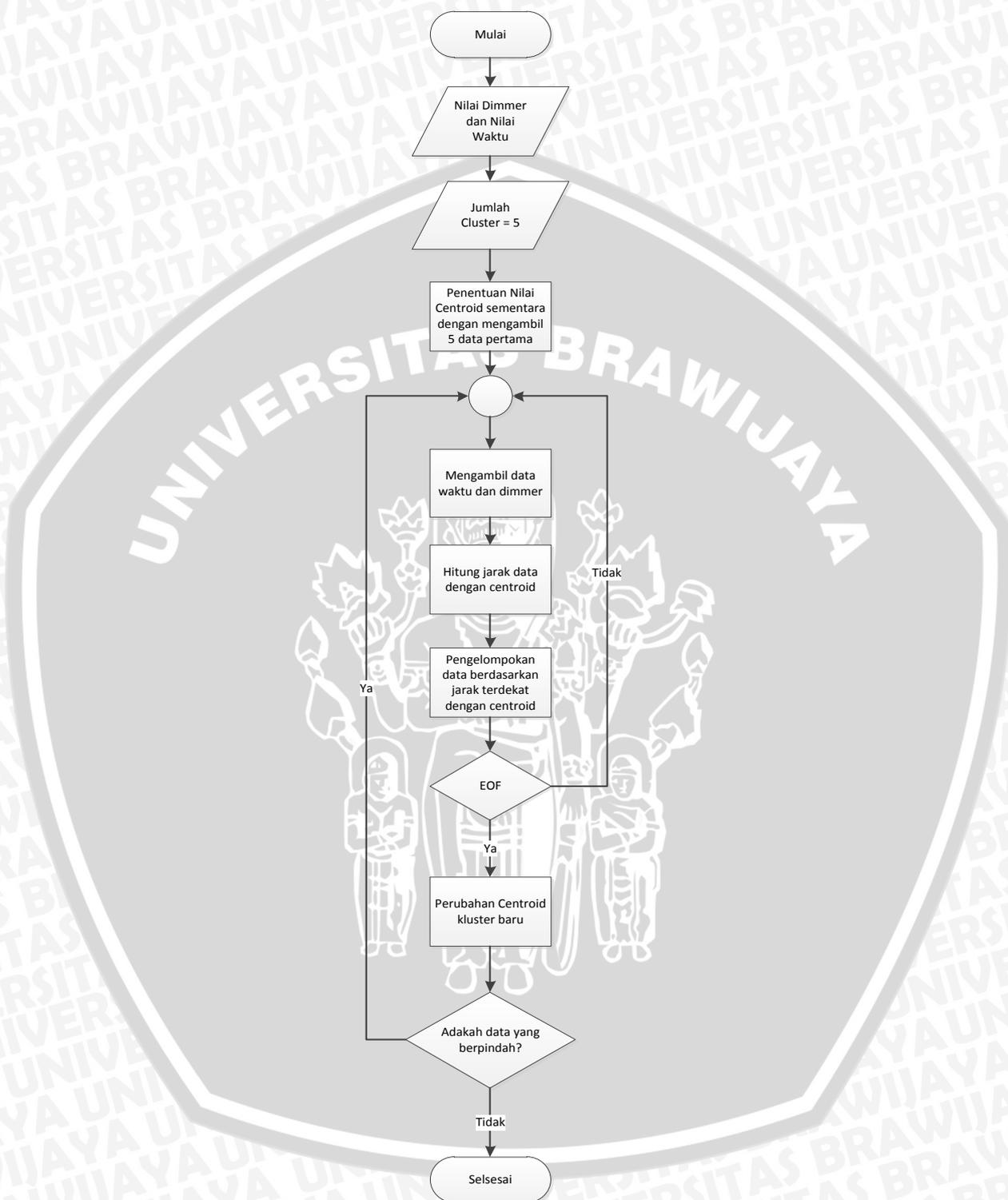
		RTC untuk menyimpan waktu
8.	Detik	Variabel ini khusus untuk <i>node</i> sensor yang menggunakan RTC untuk menyimpan waktu
9.	Nilai	Variabel ini berisi nilai yang digunakan untuk apa saja sesuai kebutuhan. Contohnya untuk menyimpan nilai terang redup cahaya
10.	GP	General propose Variabel ini adalah variabel untuk kebutuhan khusus jika dibutuhkan
11.	Checksum	Variabel ini digunakan untuk pengecekan kerusakan paket data.

4.1.10. Perancangan Algoritma K-means

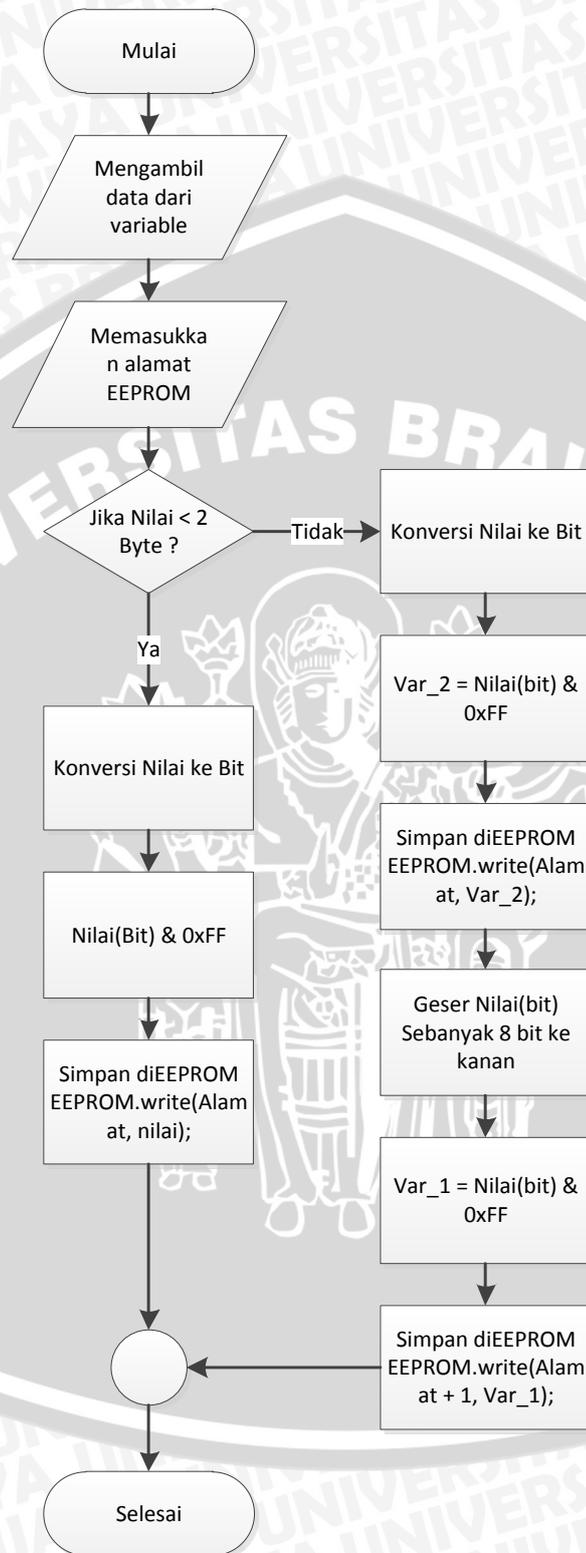
Perancangan Algoritma klasterisasi K-means digunakan untuk perhitungan klasterisasi waktu dan terang redup cahaya sesuai dengan kedekatan *centroid*, pada algoritma ini membutuhkan data waktu dan data terang redup cahaya yang dimasukan pengguna dan data yang pernah (*history*) dimasukan yang disimpan di EEPROM arduino. Berdasarkan data yang didapat, maka algoritma K-means ini akan memisahkan data berdasarkan kedekatan antar data. Diagram alir proses K-means ditunjukkan dalam Gambar 4.6.

4.1.11. Perancangan Penyimpanan Data dan Pembacaan Data ke EEPROM

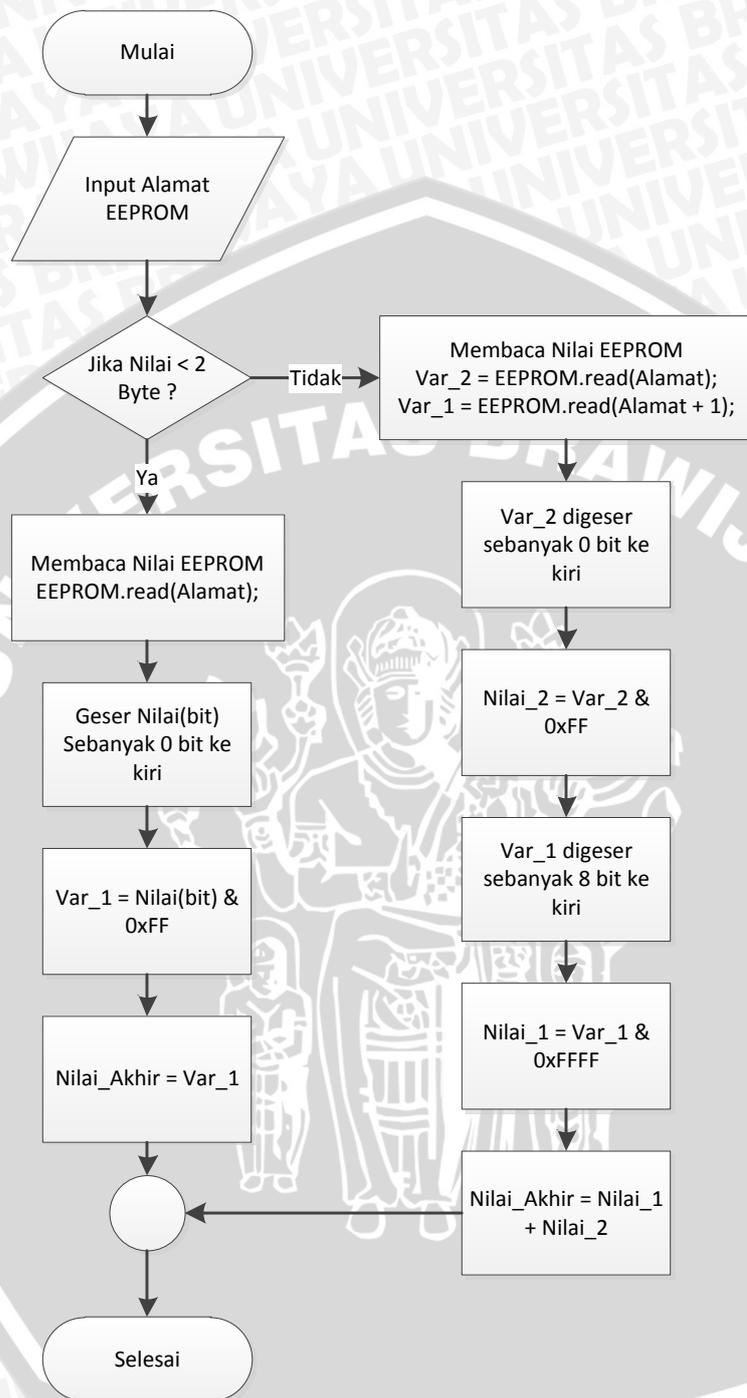
Perancangan proses penyimpanan data dan pembacaan data di EEPROM arduino pro mini sudah terdapat *library* penyimpanan atau pembacaan pada aplikasi arduino IDE, yaitu EEPROM.h. pada library tersebut proses penyimpanan dan pembacaan EEPROM hanya bisa menggunakan data bertipe *byte*, oleh karena itu data yang bertipe data *integer* dirubah dalam bentuk *byte*. Perintah pada library untuk menuliskan data pada arduino adalah “EEPROM.write(alamat_eeprom, Nilai_yang_disimpan)” dan untuk pembacaanya menggunakan perintah “EEPROM.read(alamat_eeprom)”. Diagram alir proses penyimpanan bisa dilihat pada Gambar 4.7. dan untuk pembacaan data ditunjukkan pada Gambar 4.8.



Gambar 4.6. Diagram alir proses perhitungan K-means



Gambar 4.7. Proses penyimpanan data di EEPROM arduino



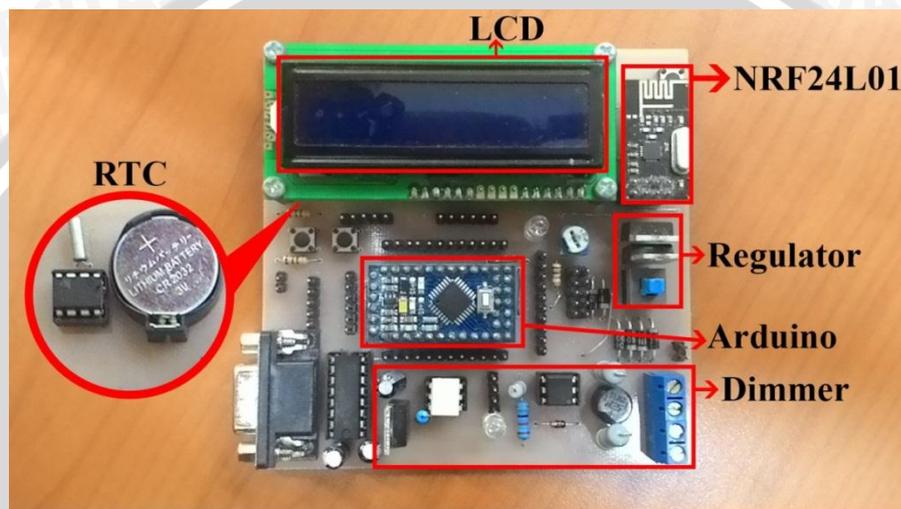
Gambar 4.8. Proses pembacaan data di EEPROM arduino

4.2. Implementasi

Sub Bab ini membahas mengenai tahapan implementasi berdasarkan perancangan yang telah dijelaskan pada subbab sebelumnya.

4.2.1. Implementasi Perangkat Keras

Implementasi perangkat keras pada penelitian ini meliputi implementasi rangkaian *dimmer*, implementasi rangkaian RTC, dan implementasi rangkaian arduino ke NRF24L01. Pada ketiga implementasi perangkat keras tersebut saling berhubungan, hasil dari implementasi perangkat keras pada penelitian ini ditunjukkan pada Gambar 4.9.



Gambar 4.9. Hasil implementasi perancangan perangkat keras

4.2.2. Implementasi Perangkat Lunak

Implementasi perangkat lunak dalam penelitian ini meliputi Implementasi algoritma K-means, program *dimmer*, program RTC, program penulisan dan pembacaan data ke EEPROM arduino.

4.2.3. Implementasi Algoritma K-means

Proses algoritma K-means adalah proses klusterisasi data yang dimasukan oleh pengguna dari modul *radio frekuensi*(RF). *Source code* algoritma K-means beserta hasil dari perhitungan klusterisasi algoritma K-means pada arduino pro mini ditunjukkan pada Gambar 4.10. dan hasilnya pada Gambar 4.11.

```
1. void K_MEANS() {
2.
3.     const int Jumlah_Klaster = 5;
4.     const int Jumlah_Data = 20;
5.
6.     int data_intensitas[Jumlah_Data] =
7.     {1020,1140,1260,1380,61,
8.                                     1022,1143,1264,1385,66,
9.                                     1027,1148,1269,1390,71,
10.                                    1032,1153,1274,1395,76
11.                                    };
12.     int data_waktu[Jumlah_Data] = {50,70,100,40,1,
13.                                   50,70,100,40,1,
14.                                   50,70,100,40,1,
15.                                   50,70,100,40,1
16.                                   };
17.     Data_Kmeans* k = new Data_Kmeans[Jumlah_Klaster];
18.     Data_Kmeans* Center_Centroid = new
19. Data_Kmeans[Jumlah_Klaster];
20.
21.     int count_Group[Jumlah_Klaster] ;
22.
23.     vector< Data_Kmeans > datasementara;
24.     vector< int > jarak[Jumlah_Klaster];
25.
26.     for (int i = 0; i < Jumlah_Data; i++) {
27.         Data_Kmeans tmp;
28.         tmp.x = (int)(rand()%100);
29.         tmp.y = (int)(rand()%100);
30.         cout << "data" << i << "(" << tmp.x << "," << tmp.y<<
31.         ")" << endl;
32.         datasementara.push_back(tmp); // meletakkan daata
33.         dibelakang
34.     }
35.
36.     for (int i = 0; i < Jumlah_Klaster; i++)
37.     {
38.         k[i] = datasementara[i];
39.         Center_Centroid[i].x = datasementara[i].x;
40.         Center_Centroid[i].y = datasementara[i].y;
41.         jarak[i].resize( Jumlah_Data );
42.         cout <<"Centroid" <<i<< "(" << Center_Centroid[i].x
43. <<"," << Center_Centroid[i].y << ")" << endl;
44.     }
45.
46.     bool loop = true;
47.     while (loop) {
48.         //Center_Centroid init
49.         for (int i = 0; i < Jumlah_Klaster; i++) {
50.             Center_Centroid[i].x = 0;
51.             Center_Centroid[i].y = 0;
52.             count_Group[i] = 0;
53.         }
54.         // jarak
55.         for (int i = 0; i < datasementara.size(); i++) {
56.             for (int j = 0; j < Jumlah_Klaster; j++) {
```

```

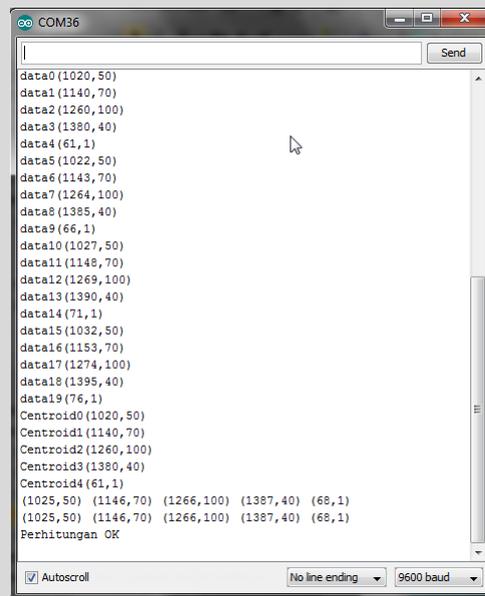
57.         int tmp_jarak = sqrt(pow( k[j].x -
58. datasementara[i].x, 2) + pow( k[j].y - datasementara[i].y,
59. 2) );
60.         jarak[j][i] = tmp_jarak;
61.     }
62. }
63. //Mencari centroid
64. for (int i = 0; i < datasementara.size(); i++) {
65.     int min = jarak[0][i];
66.     int min_j = 0;
67.
68.     for (int j = 1; j < Jumlah_Klaster; j++) {
69.         if (min > jarak[j][i]) {
70.             min = jarak[j][i];
71.             min_j = j;
72.         }
73.     }
74.     Center_Centroid[ min_j ].x += datasementara[i].x;
75.     Center_Centroid[ min_j ].y += datasementara[i].y;
76.     count_Group[min_j]++;
77. }
78. //menganti nilai K
79. int same_count = 0;
80. for (int i = 0; i < Jumlah_Klaster; i++) {
81.     if (count_Group[i] != 0) {
82.         if ((Center_Centroid[i].x / count_Group[i] ==
83. k[i].x
84.         && (Center_Centroid[i].y / count_Group[i] ==
85. k[i].y))
86.             same_count ++;
87.         k[i].x = Center_Centroid[i].x / count_Group[i];
88.         k[i].y = Center_Centroid[i].y / count_Group[i];
89.     }
90.     if (same_count == Jumlah_Klaster) {
91.         loop = false;
92.     }
93.     cout << "(" << k[i].x << setprecision(2) << "," <<
94. k[i].y << setprecision(2) << ") ";
95.     } cout << endl;
96. }
97.
98. for (int i = 0; i < datasementara.size(); i++) {
99.     int min = jarak[0][i];
100.    int min_j = 0;
101.
102.    for (int j = 1; j < Jumlah_Klaster; j++) {
103.        if (min > jarak[j][i]) {
104.            min = jarak[j][i];
105.            min_j = j;
106.        }
107.    }
108. }
109. }

```

Gambar 4.10. Source code algoritma K-MEANS

Penjelasan proses pada Gambar 4.10. yaitu:

- 1) Baris 2-16 proses inialisasi jumlah kluster dan jumlah data beserta data yang dipakai dalam implentasi ini.
- 2) Baris 24-35 proses penentuan *centroid* awal.
- 3) Baris 36-55 adalah proses pemasukkan data kedalam variable vector.
- 4) Baris 54-63 adalah perhitungan *Euclidian distance*.
- 5) Baris 64-72 adalah pencarian *centroid* baru.
- 6) Baris 78-89 adalah mengganti *centroid* lama dengan yang baru.
- 7) Baris 89-109 adalah proses interasi berhenti ketika *centroid* lama dan baru tidak berbeda.



```
data0(1020,50)
data1(1140,70)
data2(1260,100)
data3(1380,40)
data4(61,1)
data5(1022,50)
data6(1143,70)
data7(1264,100)
data8(1385,40)
data9(66,1)
data10(1027,50)
data11(1148,70)
data12(1269,100)
data13(1390,40)
data14(71,1)
data15(1032,50)
data16(1153,70)
data17(1274,100)
data18(1395,40)
data19(76,1)
Centroid0(1020,50)
Centroid1(1140,70)
Centroid2(1260,100)
Centroid3(1380,40)
Centroid4(61,1)
(1025,50) (1146,70) (1266,100) (1387,40) (68,1)
(1025,50) (1146,70) (1266,100) (1387,40) (68,1)
Perhitungan OK
```

Gambar 4.11. Proses algoritma K-means

4.2.4. Implementasi Program *Dimmer*

Proses *dimmer* digunakan untuk pengontrolan terang redup cahaya pada lampu pijar, pada proses tersebut menggunakan *interrupt* pada arduino pro mini untuk mengontrol terang redup cahaya. Potongan *source code* dari program *dimmer* seperti Gambar 4.12.

```
1. int AC_LOAD = 3;  
2. int dimming;  
3. void setup() {  
4.     pinMode(AC_LOAD, OUTPUT);  
5.     attachInterrupt(0, zero_crosss_int, RISING);  
6. }  
7. void zero_crosss_int()  
8.  
9. {  
10.     int dimtime = (75 * dimming); // For 60Hz =>65  
11.     delayMicroseconds(dimtime); // Off cycle  
12.     digitalWrite(AC_LOAD, HIGH); // triac firing  
13.     delayMicroseconds(10); // triac On propogation  
14.     delay //(for 60Hz use 8.33)  
15.     digitalWrite(AC_LOAD, LOW); // triac Off  
16. }  
17. void loop(){  
18.     dimming=75  
19. }
```

Gambar 4.12. Potongan *source code* program *dimmer*

Penjelasan proses pada Gambar 4.12. yaitu:

- 1) Baris 1 dan 2 proses inialisasi *pin* dan port.
- 2) Baris 5 dan 6 proses pengaturan dan inialisasi *interrupt*.
- 3) Baris 11-15 adalah proses *interrupt dimmer*.
- 4) Baris 18 adalah pengisian nilai *dimmer* dengan rentang 15 adalah nyala terang dan 120 adalah mati.

4.2.5. Implementasi Program RTC

Program RTC digunakan sebagai penunjuk dan penyimpan waktu dalam kondisi yang sebenarnya. Waktu pada penelitian ini menjadi salah satu masukkan dalam algoritma K-means. Pada implementasi program ini menggunakan library bawaan arduino yaitu DS1302.h berikut potongan *source code* yang ditunjukkan pada Gambar 4.13.

```

1. #include <DS1302.h>
2. const int kCePin = 4; // Chip Enable
3. const int kIoPin = 5; // Input/Output
4. const int kSclkPin = 6; // Serial Clock
5. DS1302 rtc(kCePin, kIoPin, kSclkPin);
6.
7. void setup(){
8.     rtc.writeProtect(false);
9.     rtc.halt(false);
10.     Time t(2015, 02, 20, 23, 49, 00, Time::kFriday);
11.     Serial.begin(9600);
12. }
13.
14. void loop(){
15.     Time t = rtc.time();
16.     char buf[50];
17.     snprintf(buf, sizeof(buf), "%02d:%02d:%02d", t.hr,
18.     t.min, t.sec);
19.     Serial.print(Jam);
20. }
21.

```

Gambar 4.13. Potongan *source code* program RTC

Penjelasan proses pada Gambar 4.13. yaitu:

- 1) Baris 1 adalah library yang digunakan.
- 2) Baris 2-5 adalah proses inisialisasi *pin* dan port.
- 3) Baris 8 dan 9 adalah proses menyalakan RTC.
- 4) Baris 10 dan 11 adalah proses pengaturan tahun, bulan, tanggal, jam, menit dan detik.
- 5) Baris 16 untuk memanggil fungsi waktu pada rangkaian RTC.
- 6) Baris 17-20 adalah proses penampilan waktu pada serial monitor arduino IDE.

4.2.6. Implementasi Penulisan dan Pembacaan EEPROM Arduino

Pada implementasi pembacaan dan penulisan data pada EEPROM arduino ini bertujuan untuk menyimpan data secara permanen, apabila *hardware* kehilangan tagangan sumber atau mati maka nilai yang tersimpan tidak akan hilang atau terhapus, pada implementasi penelitian ini, penyimpanan dan pembacaan dibagi menjadi 2 bagian, yaitu penyimpanan dan pembacaan data 1 *byte* dan penyimpanan dan pembacaan data 2 *byte*. Pada arduino terdapat library

EEPROM.h yang berguna untuk membaca dan penyimpanan data pada arduino, pada *library* tersebut, penulis membuat fungsi untuk mempermudah penyimpanan dan pembacaan data 1 *byte* dan penyimpanan dan pembacaan data 2 *byte*. Potongan fungsi tersebut ditunjukkan pada Gambar 4.14.

```

1. void EEPROM_WRITE_WAKTU(int Alamat_EEPROM, long Nilai)
2. {
3.     byte dua = (Nilai & 0xFF);
4.     byte satu = ((Nilai >> 8) & 0xFF);
5.     EEPROM.write(Alamat_EEPROM, dua);
6.     EEPROM.write(Alamat_EEPROM + 1, satu);
7. }
8. long EEPROM_READ_WAKTU(long Alamat_EEPROM)
9. {
10.    long dua = EEPROM.read(Alamat_EEPROM);
11.    long satu = EEPROM.read(Alamat_EEPROM + 1);
12.    return ((dua << 0) & 0xFF) + ((satu << 8) & 0xFFFF);
13. }
14. void EEPROM_WRITE_INTENSITAS(int Alamat_EEPROM, long
15. Nilai)
16. {
17.     byte satu = (Nilai & 0xFF);
18.     EEPROM.write(Alamat_EEPROM, satu);
19. }
20. long EEPROM_READ_INTENSITAS(long Alamat_EEPROM)
21. {
22.     long satu = EEPROM.read(Alamat_EEPROM);
23.     return ((satu << 0) & 0xFF);
24. }

```

Gambar 4.14. Potongan *source code* program Pembacaan dan Penulisan data di EEPROM arduino

Penjelasan proses pada Gambar 4.14. yaitu:

- 1) Baris 1- 7 adalah fungsi penyimpanan data 2 *byte*.
- 2) Baris 8- 12 adalah fungsi pembacaan data 2 *byte*.
- 3) Baris 13- 17 adalah fungsi penyimpanan data 1 *byte*.
- 4) Baris 18- 21 adalah fungsi pembacaan data 1 *byte*.

BAB V

PENGUJIAN DAN ANALISIS

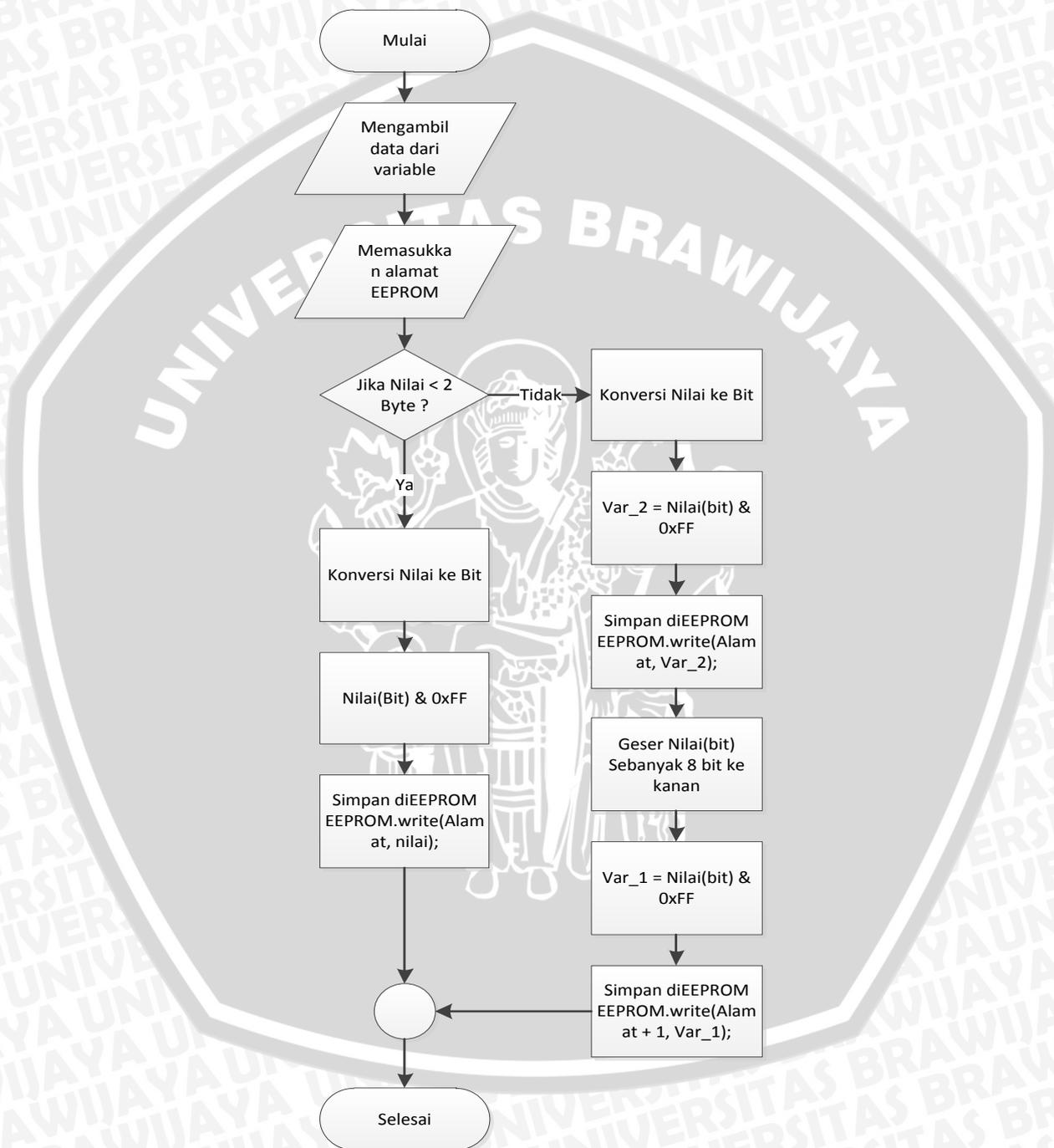
Bab ini membahas mengenai tahapan pengujian dan pembahasan sistem hasil implementasi dan pengujian dari penelitian ini. Proses pengujian dilakukan melalui dua macam pengujian yaitu pengujian perangkat keras dan pengujian perangkat lunak. Pengujian perangkat keras digunakan untuk menguji sistem dapat berjalan sesuai harapan serta konsumsi daya listrik dan Pengujian perangkat lunak digunakan untuk menguji algoritma K-means yang diterapkan pada perangkat keras.

Proses pembahasan bertujuan untuk mendapatkan kesimpulan dari hasil pengujian hasil implementasi dan pengujian K-means pada perangkat keras Arduino pro mini dalam mengatur kebiasaan serta kesesuaian terang redup cahaya dari cahaya lampu. Proses pembahasan mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Pembahasan dilakukan terhadap hasil pengujian di setiap tahap pengujian.

5.1. Pengujian Penyimpanan dan Pembacaan Data ke eprom

Pengujian penyimpanan dan pembacaan data ke EEPROM arduino dilakukan dengan menghubungkan arduino pro mini dengan kabel *USB TO SERIAL* melalui perantara Rangkaian MAX232. Pengujian ini bertujuan untuk menyimpan data yang menjadi masukan dari penelitian ini yaitu data waktu dan terang redup cahaya yang nanti akan menjadi masukan dari algoritma K-means. Pengujian ini data-data dari waktu serta terang redup cahaya disimpan terlebih dahulu dalam suatu variable tertentu selanjutnya dengan menekan tombol 1 untuk melakukan perintah penyimpanan dan tombol 2 untuk perintah pembacaan dan hasilnya ditampilkan di terminal arduino IDE seperti Gambar 5.1. Diagram alir penulisan atau penyimpanan data ke EEPROM yang ditanamkan pada arduino dapat dilihat dalam Gambar 5.1. dan data yang ditulis dapat dilihat dalam Tabel 5.1. Diagram alir untuk pembacaan data di EEPROM dapat dilihat dalam Gambar

5.2. dan hasil pembacaan dapat dilihat dalam Tabel 5.2. dan hasil dari pembacaan data dari EEPROM arduino sesuai dengan data yang dituliskan pada Tabel 5.1.

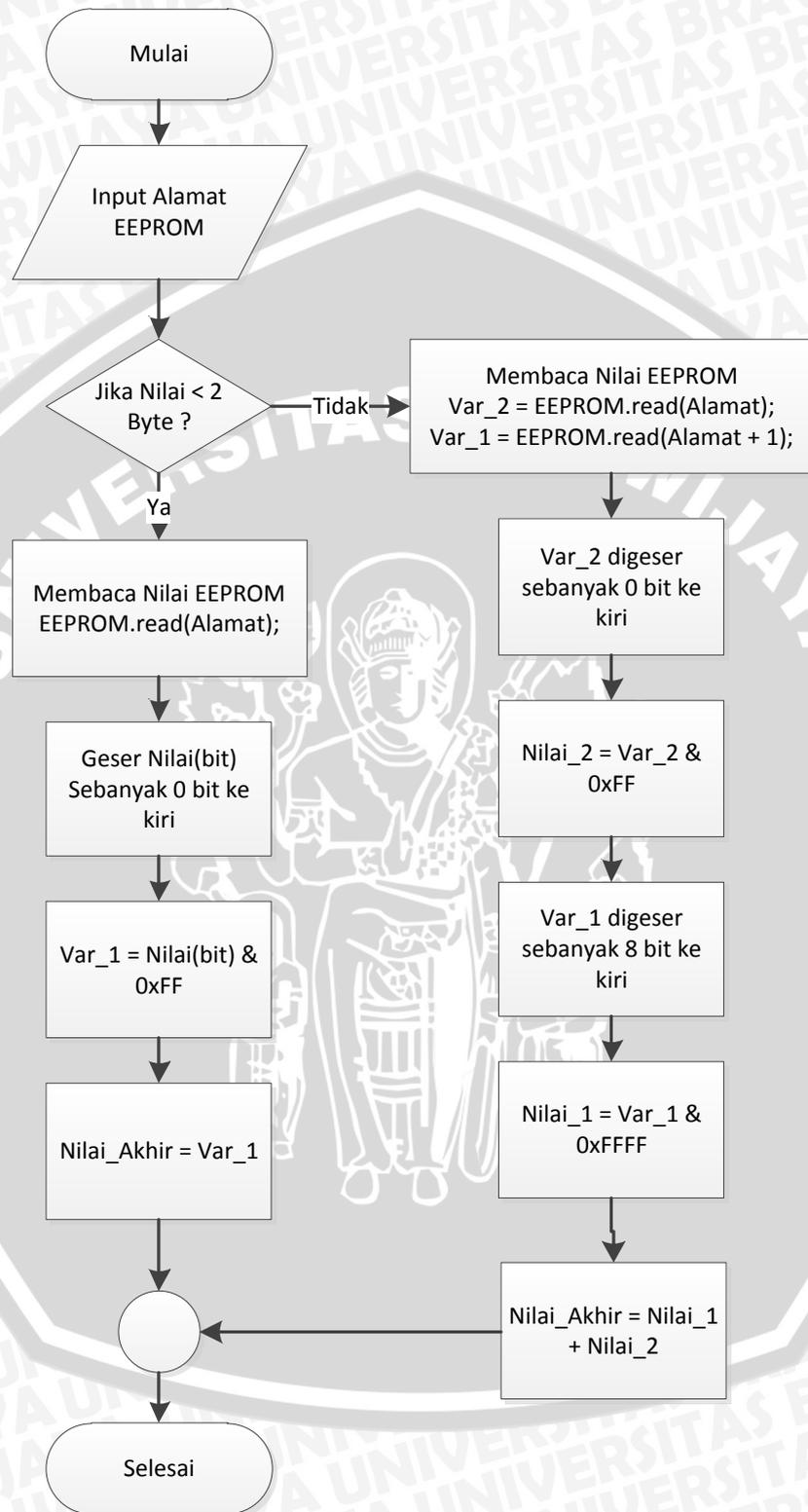


Gambar 5.1. Flowchart Penyimpanan EEPROM

Tabel 5.1. Penyimpanan di EEPROM Arduino

Alamat	Data yang ditulis di EEPROM	Keterangan
0	1020	Data Waktu
1		
2	50	Data Terang redup cahaya
3	1140	Data Waktu
4		
5	70	Data Terang redup cahaya
6	1260	Data Waktu
7		
8	100	Data Terang redup cahaya
9	1380	Data Waktu
10		
11	40	Data Terang redup cahaya
12	61	Data Waktu
13		
14	1	Data Terang redup cahaya
15	1020	Data Waktu
16		
17	50	Data Terang redup cahaya
18	1140	Data Waktu
19		
20	70	Data Terang redup cahaya
21	1260	Data Waktu
22		
23	100	Data Terang redup cahaya
24	1380	Data Waktu
25		
26	40	Data Terang redup cahaya
27	61	Data Waktu
28		
29	1	Data Terang redup cahaya

30	1020	Data Waktu
31		
32	50	Data Terang redup cahaya
33	1140	Data Waktu
34		
35	70	Data Terang redup cahaya
36	1260	Data Waktu
37		
38	100	Data Terang redup cahaya
39	1380	Data Waktu
40		
41	40	Data Terang redup cahaya
42	61	Data Waktu
43		
44	1	Data Terang redup cahaya
45	1020	Data Waktu
46		
47	50	Data Terang redup cahaya
48	1140	Data Waktu
49		
50	70	Data Terang redup cahaya
51	1260	Data Waktu
52		
53	100	Data Terang redup cahaya
54	1380	Data Waktu
55		
56	40	Data Terang redup cahaya
57	61	Data Waktu
58		
59	1	Data Terang redup cahaya

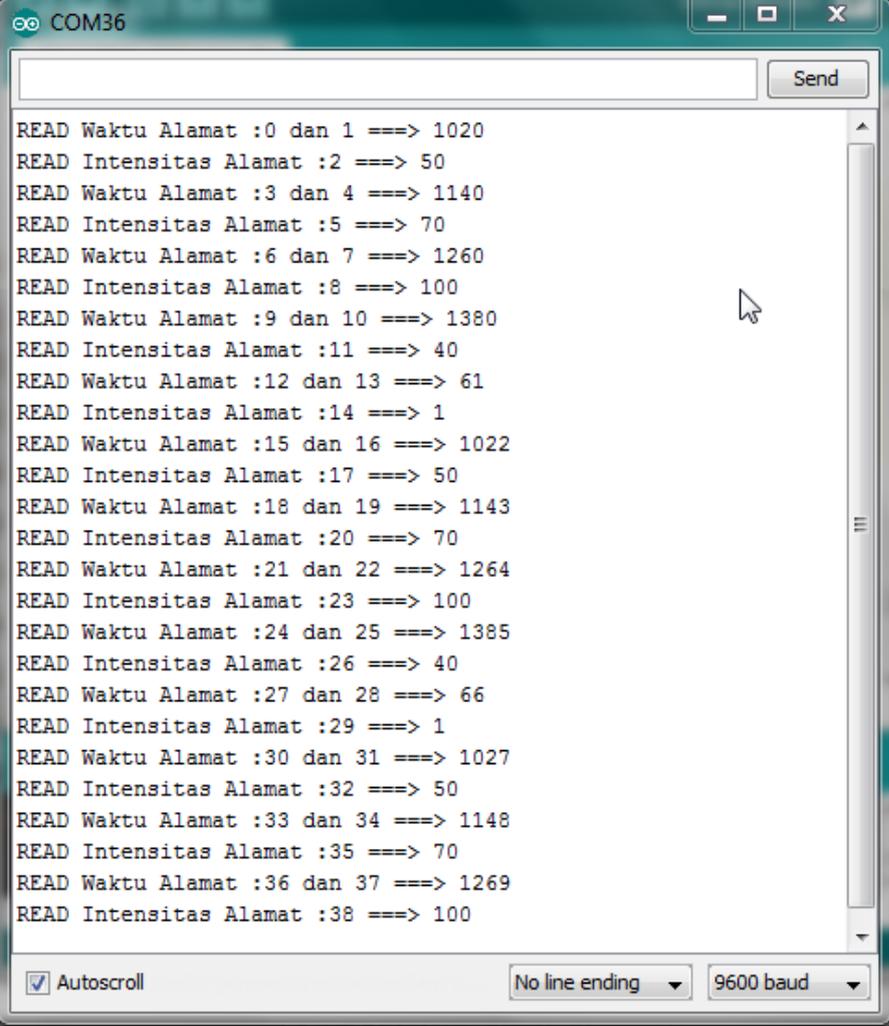


Gambar 5.2. Flowchart Pembacaan EEPROM

Tabel 5.2. Pembacaan di EEPROM Arduino

Alamat	Data yang dibaca di EEPROM	Keterangan
0	1020	Sesuai Data Waktu
1		
2	50	Sesuai Data Terang redup cahaya
3	1140	Sesuai Data Waktu
4		
5	70	Sesuai Data Terang redup cahaya
6	1260	Sesuai Data Waktu
7		
8	100	Sesuai Data Terang redup cahaya
9	1380	Sesuai Data Waktu
10		
11	40	Sesuai Data Terang redup cahaya
12	61	Sesuai Data Waktu
13		
14	1	Sesuai Data Terang redup cahaya
15	1020	Sesuai Data Waktu
16		
17	50	Sesuai Data Terang redup cahaya
18	1140	Sesuai Data Waktu
19		
20	70	Sesuai Data Terang redup cahaya
21	1260	Sesuai Data Waktu
22		
23	100	Sesuai Data Terang redup cahaya
24	1380	Sesuai Data Waktu
25		
26	40	Sesuai Data Terang redup cahaya
27	61	Sesuai Data Waktu
28		

29	1	Sesuai Data Terang redup cahaya
30	1020	Sesuai Data Waktu
31		
32	50	Sesuai Data Terang redup cahaya
33	1140	Sesuai Data Waktu
34		
35	70	Sesuai Data Terang redup cahaya
36	1260	Sesuai Data Waktu
37		
38	100	Sesuai Data Terang redup cahaya
39	1380	Sesuai Data Waktu
40		
41	40	Sesuai Data Terang redup cahaya
42	61	Sesuai Data Waktu
43		
44	1	Sesuai Data Terang redup cahaya
45	1020	Sesuai Data Waktu
46		
47	50	Sesuai Data Terang redup cahaya
48	1140	Sesuai Data Waktu
49		
50	70	Sesuai Data Terang redup cahaya
51	1260	Sesuai Data Waktu
52		
53	100	Sesuai Data Terang redup cahaya
54	1380	Sesuai Data Waktu
55		
56	40	Sesuai Data Terang redup cahaya
57	61	Sesuai Data Waktu
58		
59	1	Sesuai Data Terang redup cahaya

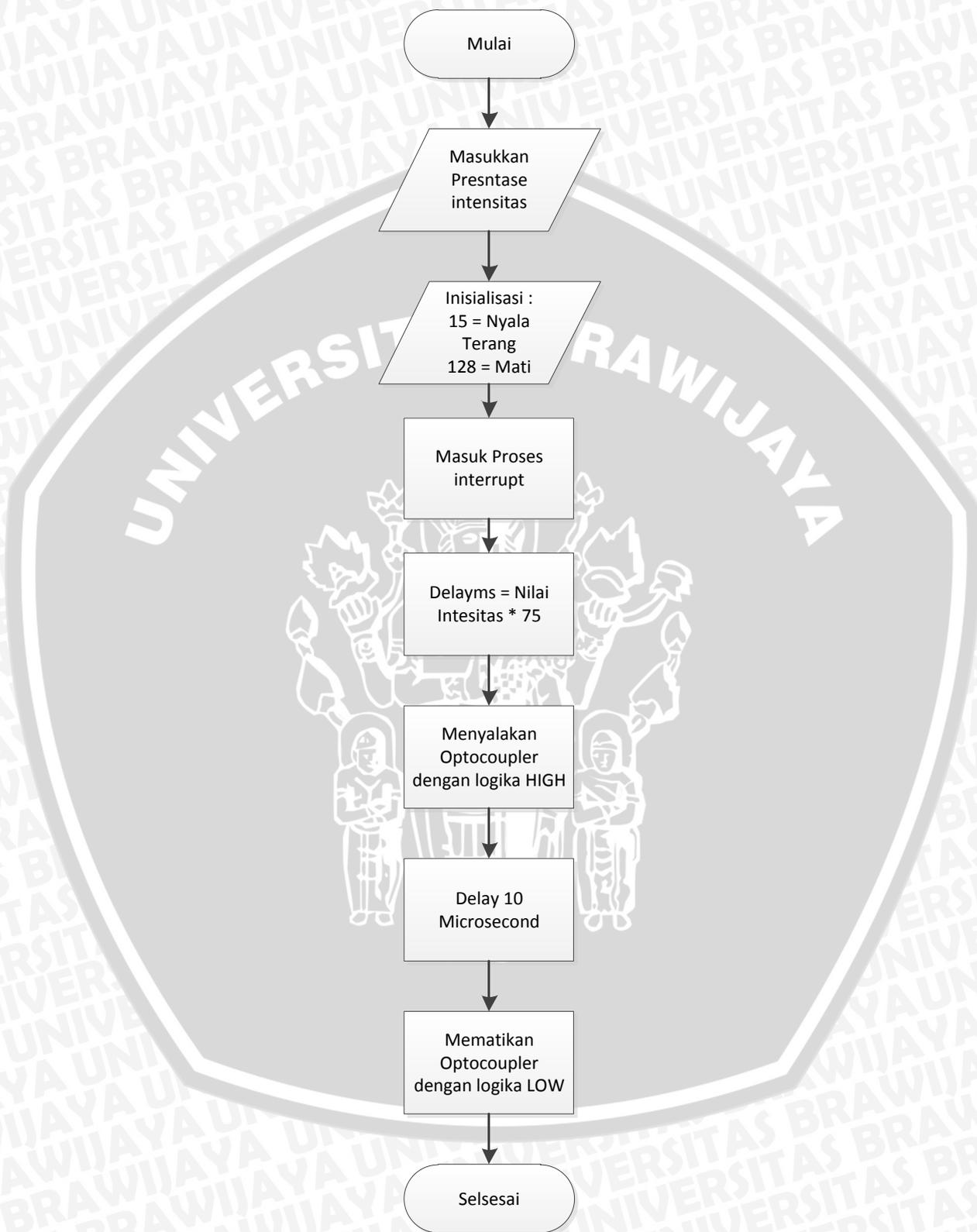


```
COM36
Send
READ Waktu Alamat :0 dan 1 ==> 1020
READ Intensitas Alamat :2 ==> 50
READ Waktu Alamat :3 dan 4 ==> 1140
READ Intensitas Alamat :5 ==> 70
READ Waktu Alamat :6 dan 7 ==> 1260
READ Intensitas Alamat :8 ==> 100
READ Waktu Alamat :9 dan 10 ==> 1380
READ Intensitas Alamat :11 ==> 40
READ Waktu Alamat :12 dan 13 ==> 61
READ Intensitas Alamat :14 ==> 1
READ Waktu Alamat :15 dan 16 ==> 1022
READ Intensitas Alamat :17 ==> 50
READ Waktu Alamat :18 dan 19 ==> 1143
READ Intensitas Alamat :20 ==> 70
READ Waktu Alamat :21 dan 22 ==> 1264
READ Intensitas Alamat :23 ==> 100
READ Waktu Alamat :24 dan 25 ==> 1385
READ Intensitas Alamat :26 ==> 40
READ Waktu Alamat :27 dan 28 ==> 66
READ Intensitas Alamat :29 ==> 1
READ Waktu Alamat :30 dan 31 ==> 1027
READ Intensitas Alamat :32 ==> 50
READ Waktu Alamat :33 dan 34 ==> 1148
READ Intensitas Alamat :35 ==> 70
READ Waktu Alamat :36 dan 37 ==> 1269
READ Intensitas Alamat :38 ==> 100
 Autoscroll
No line ending
9600 baud
```

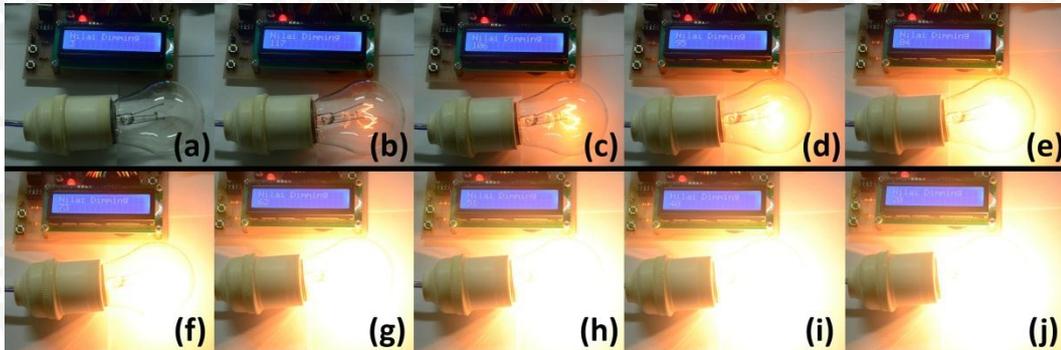
Gambar 5.3. Data yang disimpan di EEPROM

5.2. Pengujian Rangkaian *Dimmer*

Pengujian rangkaian *dimmer* ini dengan menghubungkan arduino pro mini *pin* 2 sebagai interrupt dan *pin* 3 sebagai *pin* logika dengan rangkaian *dimmer*. Pengujian ini bertujuan untuk mengatur terang redup cahaya lampu pijar. Pengujian ini menggunakan data-data presentase terang redup cahaya dari 0% sampai 100%, dimana 0% adalah keadaan lampu mati dan 100% adalah kondisi lampu menyala paling terang. Pengujian ini dilakukan dengan memasukkan data terang redup cahaya lewat terminal arduino pro mini. Diagram alur proses *dimmer* bisa dilihat pada Gambar 5.4. dan hasilnya dapat dilihat pada Gambar 5.5.



Gambar 5.4. Proses Pengaturan Dimmer



Gambar 5.5. Terang redup Cahaya

Penjelasan tentang Gambar 5.5. sebagai berikut:

- 1) Terang redup cahaya 0%. (Mati)
- 2) Terang redup cahaya 10%.
- 3) Terang redup cahaya 20%.
- 4) Terang redup cahaya 30%.
- 5) Terang redup cahaya 40%.
- 6) Terang redup cahaya 50%.
- 7) Terang redup cahaya 60%.
- 8) Terang redup cahaya 70%.
- 9) Terang redup cahaya 80%.
- 10) Terang redup cahaya 100%. (Nyala Maksimal).

5.3. Pengujian Rangkaian RTC

Pengujian rangkaian RTC (*Real Time Clock*) menggunakan IC DS1302 dengan menghubungkan *port Chip Enable* (CE) IC DS1302 dengan *pin 4* arduino pro mini, *port Input/Output* (I/O) IC DS1302 dengan *pin 5* arduino pro mini, *port Serial Clock* (Sclk) IC DS1302 dengan *pin 6* arduino pro mini. Pengujian ini bertujuan sebagai penunjuk waktu pada arduino pro mini, dimana bila arduino pro mini kehilangan sumber tegangan, maka data atau waktu tidak akan kembali ke kondisi awal melainkan sesuai dengan waktu yang sebenarnya. Hasil pengujian pada Gambar 5.6. dan Gambar 5.7. menunjukkan bahwa waktu tidak kembali ke kondisi awal (Tidak *reset*).



Gambar 5.6. Waktu Sebelum daya Arduino dimatikan.

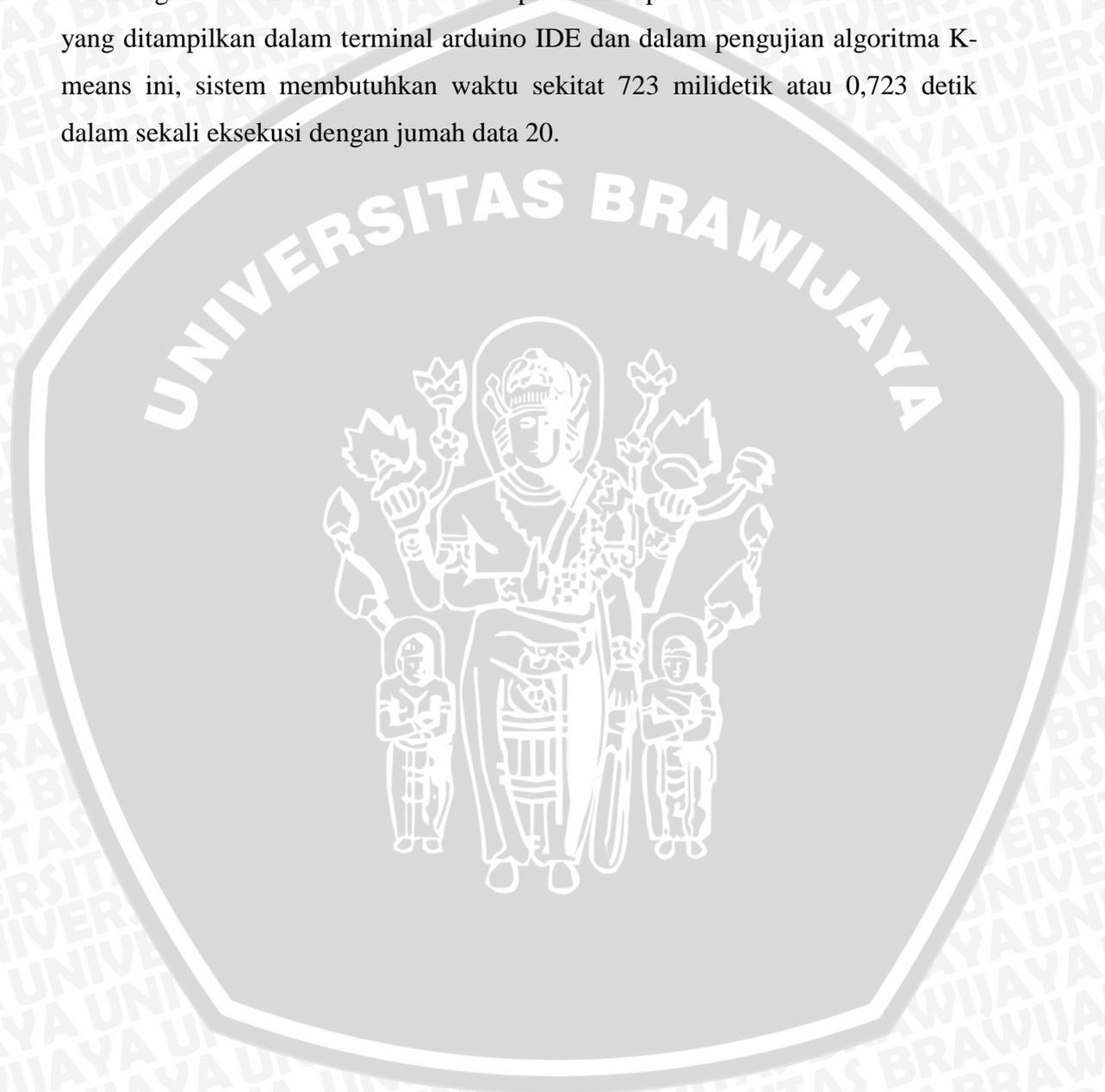


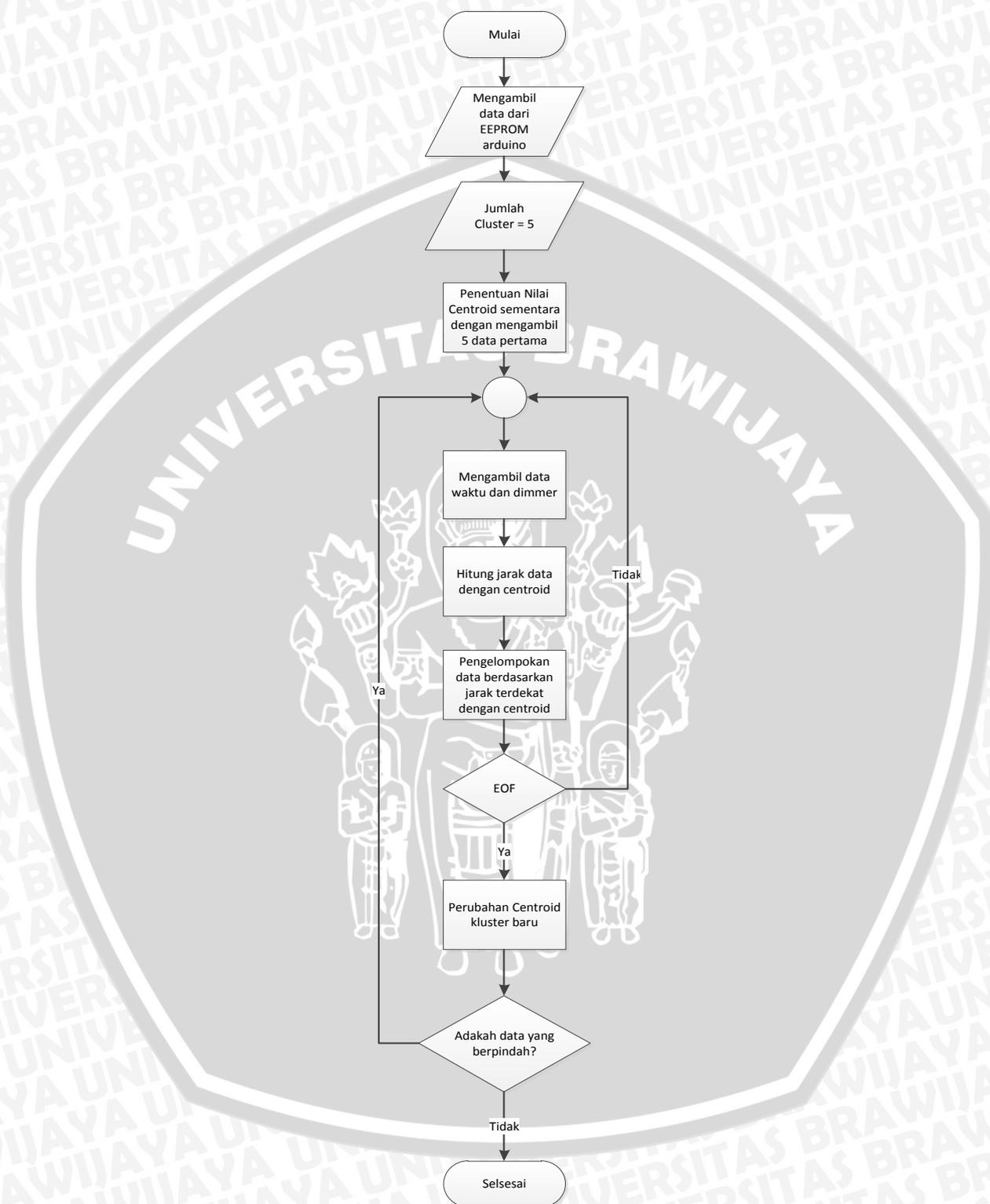
Gambar 5.7. Waktu setelah daya Arduino dimatikan.

5.4. Pengujian Algoritma K-means

Pengujian Algoritma K-means bertujuan untuk mengetahui hasil dari algoritma dimikrokontroler arduino pro mini sama atau tidak dengan perhitungan manual. Hasil dari pengujian ini digunakan sebagai keluaran yang akan dieksekusi oleh aktuator, dimana keluarannya yaitu data waktu dan data terang redup cahaya. Pengujian ini dilakukan dengan cara mengambil data dari EEPROM arduino sebagai masukkan data waktu dan data terang redup cahaya, selanjutnya diproses dengan menghitung *Euclidian Distance* atau jarak kedekatan antar data. Data yang mampu disimpan sesuai dengan flash memori arduino pro mini adalah sebanyak 20 data, dalam pengujian ini nilai klaster ditentukan. Yaitu 5 klaster. Diagram

alir proses dari algoritma K-means dapat dilihat dalam Gambar 5.7. dan data yang dipakai dalam pengujian ini bisa dilihat dalam Tabel 5.3. beserta *centroid* yang dihasilkan dari perhitungan manual dari 5 kluster pada Tabel 5.4. Pengujian Perhitungan dari mikrokontroler arduino pro mini dapat dilihat dalam Gambar 5.8. yang ditampilkan dalam terminal arduino IDE dan dalam pengujian algoritma K-means ini, sistem membutuhkan waktu sekitat 723 milidetik atau 0,723 detik dalam sekali eksekusi dengan jumlah data 20.





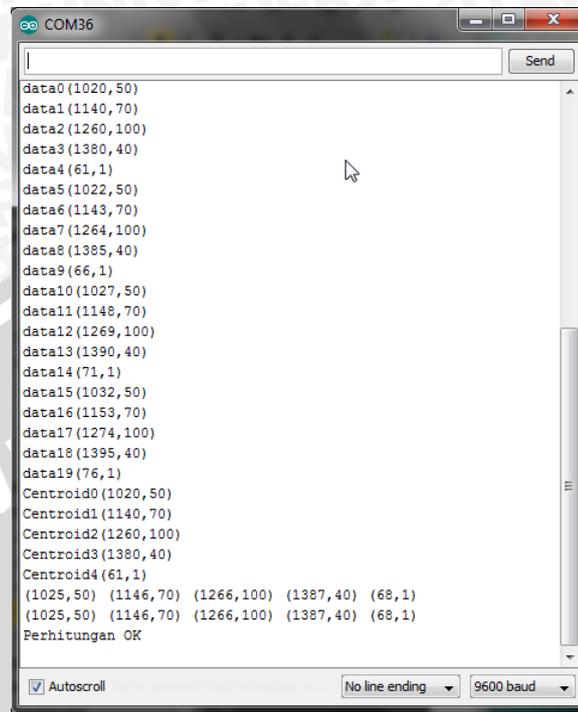
Gambar 5.8. Diagram Alir Proses K-means.

Tabel 5.3. Data masukkan Algoritma K-means

No	Data Waktu (Menit)	Data Terang redup cahaya
1	1020	50
2	1140	70
3	1260	100
4	1380	40
5	61	1
6	1022	50
7	1143	70
8	1264	100
9	1385	40
10	66	1
11	1027	50
12	1148	70
13	1269	100
14	1390	40
15	71	1
16	1032	50
17	1153	70
18	1274	100
19	1395	40
20	76	1

Tabel 5.4. Hasil *Centroid* perhitungan Manual

Klaster 1		Klaster 2		Klaster 3		Klaster 4		Klaster 5	
Wkt	Terang redup cahaya								
1020	50	1140	70	1260	100	1380	40	61	1
1025	50	1146	70	1267	100	1388	40	69	1
1025	50	1146	70	1267	100	1388	40	69	1

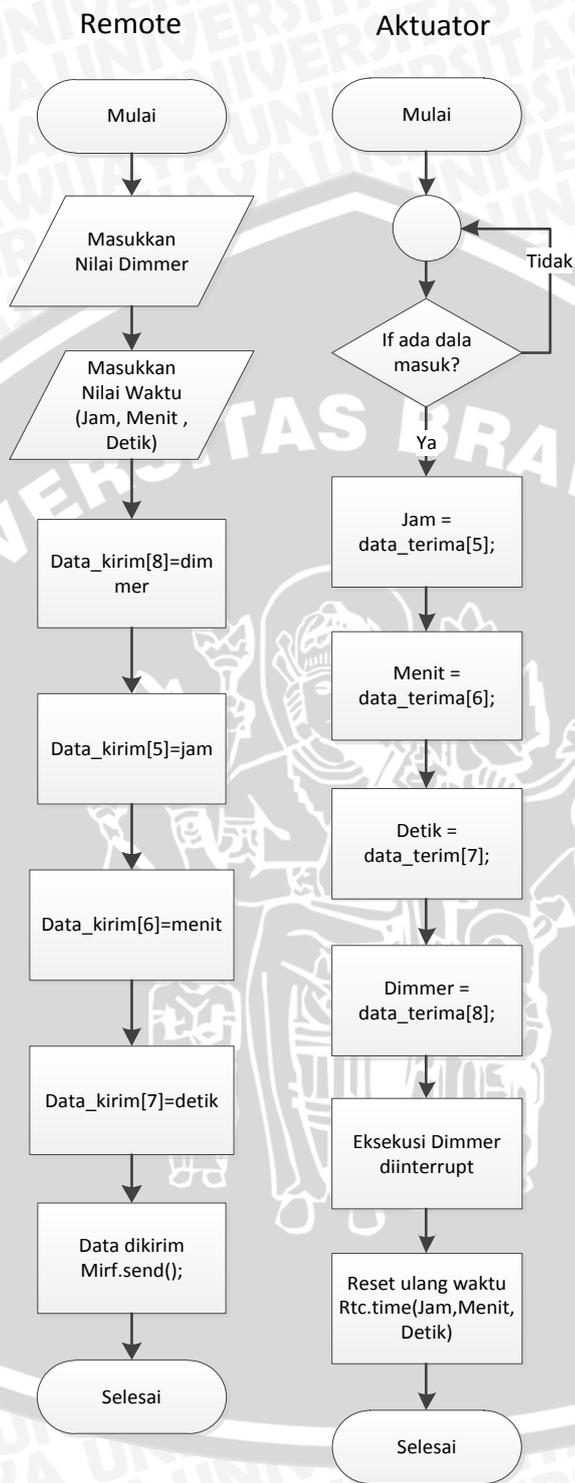


```
COM36
data0 (1020,50)
data1 (1140,70)
data2 (1260,100)
data3 (1380,40)
data4 (61,1)
data5 (1022,50)
data6 (1143,70)
data7 (1264,100)
data8 (1385,40)
data9 (66,1)
data10 (1027,50)
data11 (1148,70)
data12 (1269,100)
data13 (1390,40)
data14 (71,1)
data15 (1032,50)
data16 (1153,70)
data17 (1274,100)
data18 (1395,40)
data19 (76,1)
Centroid0 (1020,50)
Centroid1 (1140,70)
Centroid2 (1260,100)
Centroid3 (1380,40)
Centroid4 (61,1)
(1025,50) (1146,70) (1266,100) (1387,40) (68,1)
(1025,50) (1146,70) (1266,100) (1387,40) (68,1)
Perhitungan OK
Autoscroll No line ending 9600 baud
```

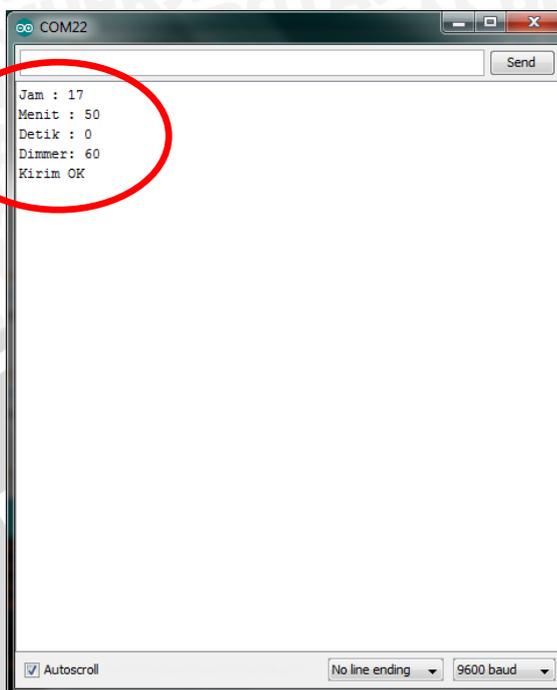
Gambar 5.9. Hasil Perhitungan Proses K-means diarduino Pro mini.

5.5. Pengujian Komunikasi Data Dengan NRF24L01

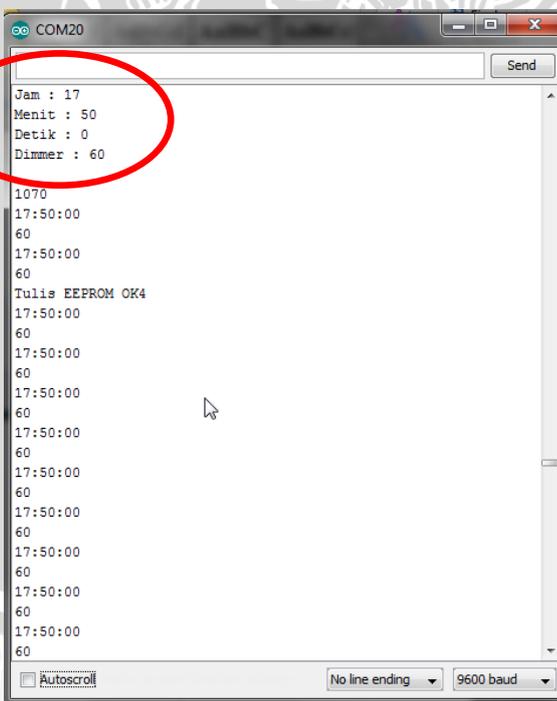
Pengujian Komunikasi pada Modul NRF24L01 menggunakan 2 buah arduino yang masing-masing terdapat modul NRF24L01. Pengujian ini bertujuan untuk mengetahui waktu pengiriman data antar modul. Pengujian ini dilakukan dengan cara modul 1 (Gambar 5.11. client) mengirimkan permintaan data ke modul 2 (Gambar 5.12. server), selanjutnya modul 2 mengirimkan data ke modul 1, dari pengiriman data dari kedua modul tersebut bisa diambil selisih waktunya, waktu inilah yang menjadi acuan seberapa lama modul ini berkomunikasi, untuk lebih jelasnya dapat dilihat pada Gambar 5.10. diagram alir dari proses tersebut dan hasil dari pengujian ini dapat dilihat pada Gambar 5.11. pada sisi client.



Gambar 5.13. Diagram blok remote control dan aktuator



Gambar 5.14. Pengiriman data dari modul di *remote control*.



Gambar 5.15. Data yang diterima dimodul aktuator.

5.7. Pengujian Keseluruhan Sistem

Pengujian ini dilakukan untuk menghitung akurasi dari algoritma K-means dalam sistem yang telah dibuat dan kemudian dilakukan analisis dari pengujian tersebut.

5.7.1. Skenario pengujian

Skenario pengujian seluruh sistem adalah dimana data terang redup cahaya dikirimkan dari modul *remote control* melalui modul NRF24L01 selanjutnya data tersebut diterima oleh modul aktuator dan data terang redup cahaya tersebut langsung dieksekusi kerangkaian *dimmer* untuk mengatur terang redup cahaya, setelah itu nilai dari terang redup cahaya dan waktu eksekusi lampu tersebut disimpan kedalam EEPROM sebagai data historis yang nantinya digunakan sebagai masukan dari algoritma K-means. Pada saat jam 00:00 data yang tersimpan dalam EEPROM akan diambil dan dilakukan perhitungan algoritma K-means dan dari hasil perhitungan tersebut didapatkan hasil dari *centroid* yang berupa data waktu serta terang redup cahaya. Data tersebut digunakan pada proses otomatisasi pada tahap selanjutnya atau pada hari berikutnya.

5.7.2. Analisis pengujian

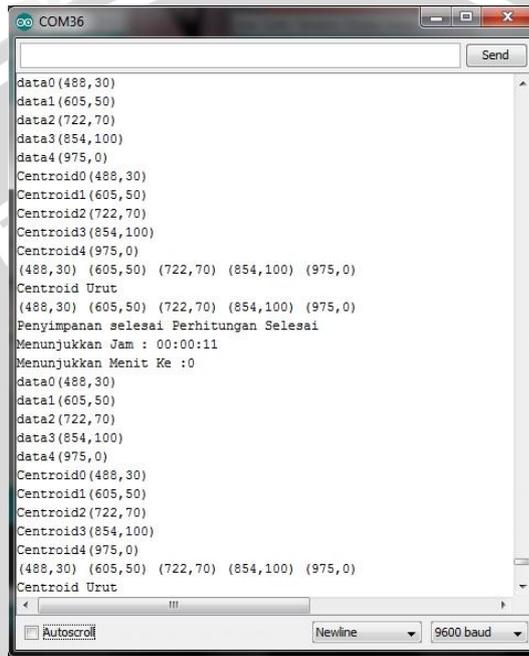
Pengujian ini dilakukan dalam waktu 4 hari dimana setiap hari memberi masukan berupa menyalakan (Terang redup cahaya) atau mematikan lampu melalui *remote control*, masukan dalam pengujian ini sesuai dengan table berikut:

Tabel 5.5. Data masukan

Hari Ke	Data					
	No	Waktu				Terang redup cahaya
		Jam	Menit	Detik	Total Linier (Menit)	
1	1	8	8	0	488	30
	2	10	5	0	605	50
	3	12	2	0	722	70
	4	14	14	0	854	100
	5	16	15	0	975	0
2	6	8	3	0	483	30
	7	10	8	0	608	50
	8	12	9	0	729	70
	9	14	13	0	853	100
	10	16	6	0	966	0
3	11	8	7	0	487	30
	12	10	9	0	609	50
	13	12	6	0	726	70
	14	14	1	0	841	100
	15	16	15	0	975	0
4	16	8	9	0	489	30
	17	10	5	0	605	50
	18	12	11	0	731	70
	19	14	10	0	850	100
	20	16	10	0	970	0

Proses perhitungan algoritma K-means dari table 5.5 data masukan diatas dilakukan tiap hari yaitu pukul 00.00 dari proses perhitungan tersebut langsung diterapkan kebentuk otomatisasi. Pada percobaan ini digunakan data sebanyak 20 data dikarenakan jumlah memori dalam arduino pro mini hanya sebesar 1 Kb dan

tipe data yang digunakan dalam masukkan tersebut adalah tipe data *integer* dan tipe data *long integer*. Jadi berdasarkan tipe data tersebut memori arduino pro mini yang sebesar 1 KB hanya dapat mengolah data sebanyak 20 data. Pada implementasinya hasil perhitungan dari algoritma K-means pada *hardware* dapat dilihat pada gambar 5.16 sampai gambar 5.19 berikut ini.

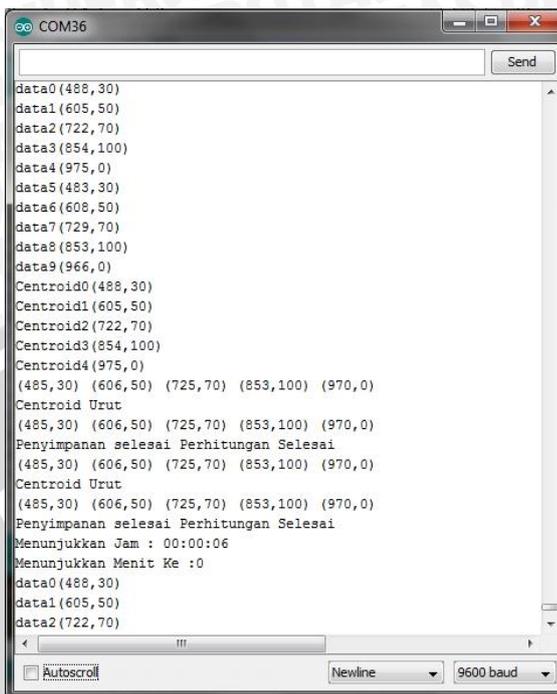


Gambar 5.16. Data dan Hasil perhitungan hari pertama.

Pada perhitungan sistem pada hari pertama sesuai pada gambar 5.16 dengan jumlah data sebanyak 5 data didapatkan perhitungan dari algoritma K-means pada Tabel 5.6. dibawah ini.

Tabel 5.6. Hasil centroid pada hari pertama

Data Centroid	Klaster 1 Baru		Klaster 2 Baru		Klaster 3 Baru		Klaster 4 Baru		Klaster 5 Baru	
	Wkt	Terang redup cahaya								
	488	30	605	50	722	70	854	100	975	0

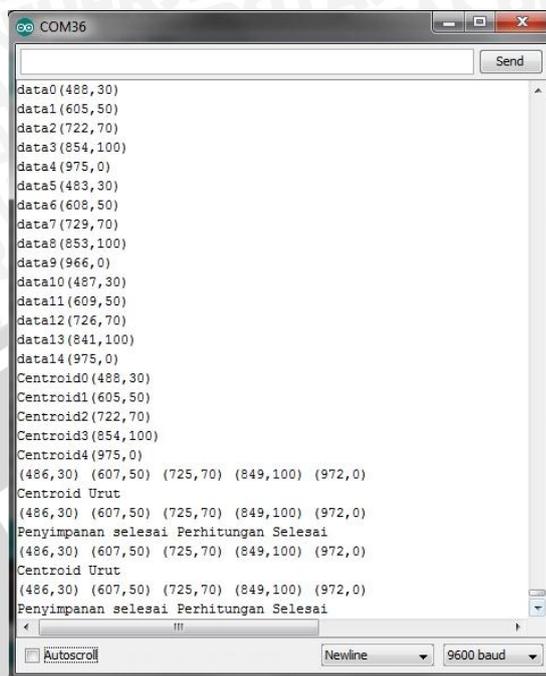


Gambar 5.17. Data dan Hasil perhitungan hari kedua.

Pada perhitungan sistem pada hari pertama sesuai pada gambar 5.17 dengan jumlah data sebanyak 10 data didapatkan perhitungan dari algoritma K-means pada Tabel 5.7. dibawah ini.

Tabel 5.7. Hasil centroid pada hari kedua

Data Centroid	Klaster 1 Baru		Klaster 2 Baru		Klaster 3 Baru		Klaster 4 Baru		Klaster 5 Baru	
	Wkt	Terang redup cahaya								
	485	30	606	50	725	70	853	100	970	0

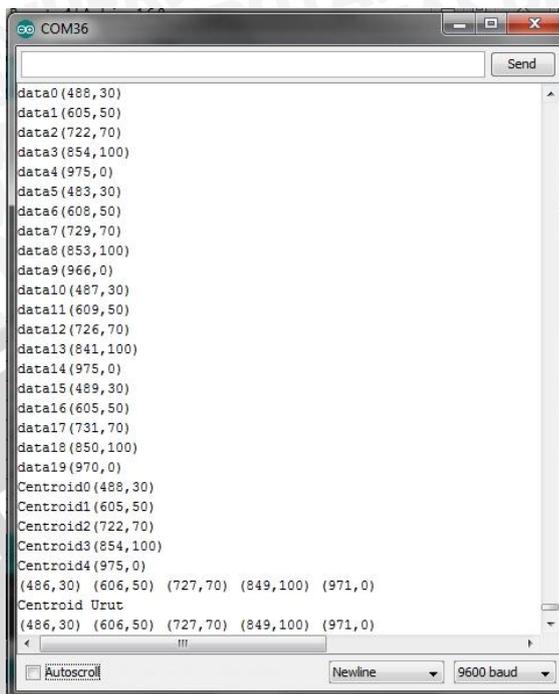


Gambar 5.18. Data dan Hasil perhitungan hari ketiga.

Pada perhitungan sistem pada hari pertama sesuai pada gambar 5.18 dengan jumlah data sebanyak 15 data didapatkan perhitungan dari algoritma K-means pada Tabel 5.8. dibawah ini.

Tabel 5.8. Hasil centroid pada hari ketiga

Data Centroid	Klaster 1 Baru		Klaster 2 Baru		Klaster 3 Baru		Klaster 4 Baru		Klaster 5 Baru	
	Wkt	Terang redup cahaya								
	486	30	607	50	725	70	849	100	972	0



Gambar 5.19. Data dan Hasil perhitungan hari keempat.

Pada perhitungan sistem pada hari pertama sesuai pada gambar 5.19 dengan jumlah data sebanyak 20 data didapatkan perhitungan dari algoritma K-means pada Tabel 5.9. dibawah ini.

Tabel 5.9. Hasil centroid pada hari keempat

Data Centroid	Klaster 1 Baru		Klaster 2 Baru		Klaster 3 Baru		Klaster 4 Baru		Klaster 5 Baru	
	Wkt	Terang redup cahaya								
	486	30	606	50	727	70	849	100	971	0

Proses perhitungan K-means manual dari data masukan selama 4 hari dilakukan menjadi beberapa tahap sebagai berikut:

- 1) Menentukan nilai masukan pada algoritma K-means sesuai dengan Tabel 5.10. sebagai berikut.

Tabel 5.10. Input sistem

Hari Ke	Data					
	No	Waktu				Intensitas
		Jam	Menit	Detik	Total Linier	
1	1	8	8	0	488	50
	2	10	5	1	605	70
	3	12	2	0	722	100
	4	14	14	0	854	40
	5	16	15	56	975	1
2	6	8	3	55	483	50
	7	10	8	54	608	70
	8	12	9	53	729	100
	9	14	13	52	853	40
	10	16	6	51	966	1
3	11	8	7	50	487	50
	12	10	9	49	609	70
	13	12	6	48	726	100
	14	14	1	47	841	40
	15	16	15	46	975	1
4	16	8	9	45	489	50
	17	10	5	44	605	70
	18	12	11	43	731	100
	19	14	10	42	850	40
	20	16	10	41	970	1

- 2) Menentukan centroid awal dengan mengambil 5 data pertama sesuai dengan Tabel 5.11. berikut.

Tabel 5.11. Centroid Awal

Data Centroid	Cluster 1 Baru		Cluster 2 Baru		Cluster 3 Baru		Cluster 4 Baru		Cluster 5 Baru	
	Waktu	Intensitas								
Iterasi / Random	488	50	605	70	722	100	854	40	975	1

3) Menghitung jarak antara data training dengan kelima centroid awal atau pada centroid pada iterasi sebelumnya dengan metode *Euclidean Distance*.

Iterasi 1 :

Tabel 5.12. Proses Perhitungan pada iterasi pertama

No	Jam	Intensitas	Cluster1				Cluster2					
			Centroid		OK	Baru		Centroid		OK	Baru	
			Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas
			481	50			601	70				
1	488	50	0		ok	488	50	119			0	0
2	605	70	119			0	0	0	ok		605	70
3	722	100	239			0	0	121			0	0
4	854	40	366			0	0	251			0	0
5	975	1	489			0	0	376			0	0
6	483	50	5		ok	483	50	124			0	0
7	608	70	122			0	0	3	ok		608	70
8	729	100	246			0	0	128			0	0
9	853	40	365			0	0	250			0	0
10	966	1	481			0	0	368			0	0
11	487	50	1		ok	487	50	120			0	0
12	609	70	123			0	0	4	ok		609	70
13	726	100	243			0	0	125			0	0
14	841	40	353			0	0	238			0	0
15	975	1	489			0	0	376			0	0
16	489	50	1		ok	489	50	118			0	0
17	605	70	119			0	0	0	ok		605	70
18	731	100	248			0	0	130			0	0
19	850	40	362			0	0	247			0	0
20	970	1	484			0	0	371			0	0

Cluster3				Cluster4				Cluster5						
Centroid		OK	Baru		Centroid		OK	Baru		Centroid		OK	Baru	
Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas
722	100				854	40				975	1			
239			0	0	366			0	0	489			0	0
121			0	0	251			0	0	376			0	0
0		ok	722	100	145			0	0	272			0	0
145			0	0	0		ok	854	40	127			0	0
272			0	0	127			0	0	0		ok	975	1
244			0	0	371			0	0	494			0	0
118			0	0	248			0	0	373			0	0
7		ok	729	100	139			0	0	265			0	0
144			0	0	1		ok	853	40	128			0	0
263			0	0	119			0	0	9		ok	966	1
240			0	0	367			0	0	490			0	0
117			0	0	247			0	0	372			0	0
4		ok	726	100	141			0	0	268			0	0
133			0	0	13		ok	841	40	140			0	0
272			0	0	127			0	0	0		ok	975	1
238			0	0	365			0	0	488			0	0
121			0	0	251			0	0	376			0	0
9		ok	731	100	137			0	0	263			0	0
141			0	0	4		ok	850	40	131			0	0
267			0	0	122			0	0	5		ok	970	1

4) Mencari centroid baru dengan metode rata-rata

Pada Tabel 5.12. diatas pada kolom member terdapat kata “OK” itu menunjukkan bahwa nilai dari data training tersebut lebih dekat pada centroid sesuai dengan label pada tabel tersebut. untuk mencari centroid baru data – data yang berada pada satu member dikelompokkan menjadi 1 dan dicari nilai rata-rata dari data tersebut sehingga dihasilkan centroid baru pada iterasi pertama seperti Tabel 5.13. berikut.

Tabel 5.13. Centroid baru pada iterasi pertama

Data Centroid	Cluster 1 Baru		Cluster 2 Baru		Cluster 3 Baru		Cluster 4 Baru		Cluster 5 Baru	
	Waktu	Intensitas								
	487	50	607	70	727	100	850	40	972	1

5) Mengulangi proses 4 – 5 sampai nilai centroid lama dan centroid baru tidak berubah.

Karena pada iterasi pertama nilai centroid awal dan centroid baru yang dihasilkan pada iterasi pertama tidak sama maka dilakukan proses 3 dan 4 untuk mencari centroid baru. Oleh karena itu diperlukan iterasi kedua pada Tabel 5.14.

Iterasi 2 :

Tabel 5.14. Proses Perhitungan

No	Jam	Intensitas	Cluster1				Cluster2					
			Centroid		OK	Baru		Centroid		OK	Baru	
			Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas
			486,75	50				606,75	70			
1	488	50	0		ok	488	50	119			0	0
2	605	70	119			0	0	0		ok	605	70
3	722	100	239			0	0	121			0	0
4	854	40	366			0	0	251			0	0
5	975	1	489			0	0	376			0	0
6	483	50	5		ok	483	50	124			0	0
7	608	70	122			0	0	3		ok	608	70
8	729	100	246			0	0	128			0	0
9	853	40	365			0	0	250			0	0
10	966	1	481			0	0	368			0	0
11	487	50	1		ok	487	50	120			0	0
12	609	70	123			0	0	4		ok	609	70
13	726	100	243			0	0	125			0	0
14	841	40	353			0	0	238			0	0
15	975	1	489			0	0	376			0	0
16	489	50	1		ok	489	50	118			0	0
17	605	70	119			0	0	0		ok	605	70
18	731	100	248			0	0	130			0	0
19	850	40	362			0	0	247			0	0
20	970	1	484			0	0	371			0	0

Cluster3				Cluster4				Cluster5						
Centroid		OK	Baru		Centroid		OK	Baru		Centroid		OK	Baru	
Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas	Jam	Intensitas		Jam	Intensitas
727	100				849,5	40				971,5	1			
244			0	0	362			0	0	486			0	0
126			0	0	246			0	0	373			0	0
5		ok	722	100	141			0	0	268			0	0
140			0	0	5		ok	854	40	124			0	0
267			0	0	131			0	0	4		ok	975	1
249			0	0	367			0	0	491			0	0
123			0	0	243			0	0	370			0	0
2		ok	729	100	135			0	0	262			0	0
140			0	0	4		ok	853	40	125			0	0
259			0	0	123			0	0	6		ok	966	1
245			0	0	363			0	0	487			0	0
122			0	0	242			0	0	369			0	0
1		ok	726	100	137			0	0	265			0	0
129			0	0	9		ok	841	40	136			0	0
267			0	0	131			0	0	4		ok	975	1
243			0	0	361			0	0	485			0	0
126			0	0	246			0	0	373			0	0
4		ok	731	100	133			0	0	260			0	0
137			0	0	1		ok	850	40	128			0	0
262			0	0	127			0	0	2		ok	970	1

Pada Tabel 5.14. diatas pada kolom member terdapat kata “OK” itu menunjukkan bahwa nilai dari data training tersebut lebih dekat pada centroid sesuai dengan label pada tabel tersebut. untuk mencari centroid baru data – data yang berada pada satu member dikelompokkan menjadi 1 dan dicari nilai rata-rata dari data tersebut sehingga dihasilkan centroid baru pada iterasi kedua seperti Tabel 5.15. berikut.

Tabel 5.15. Centroid baru pada iterasi kedua

Data Centroid	Cluster 1 Baru		Cluster 2 Baru		Cluster 3 Baru		Cluster 4 Baru		Cluster 5 Baru	
	Waktu	Intensitas								
	487	50	607	70	727	100	850	40	972	1

Pada Proses iterasi akan di ulang sampai nilai centroid lama dan nilai centroid baru tidak ada perubahan, pada perhitungan pengujian sistem ini, proses

perhitungan berhenti pada iterasi 2, dikarenakan pada centroid iterasi pertama dan centroid pada iterasi kedua sudah tidak ada perubahan nilainya.

Tabel 5.16. Hasil *Centroid* perhitungan manual

Data <i>Centroid</i>	Klaster 1 Baru		Klaster 2 Baru		Klaster 3 Baru		Klaster 4 Baru		Klaster 5 Baru	
	Wkt	Terang redup cahaya								
Iterasi	488	30	605	50	722	70	854	100	975	0
1	487	30	607	50	727	70	850	100	972	0
2	487	30	607	50	727	70	850	100	972	0
3	487	30	607	50	727	70	850	100	972	0
4	487	30	607	50	727	70	850	100	972	0
5	487	30	607	50	727	70	850	100	972	0
6	487	30	607	50	727	70	850	100	972	0
7	487	30	607	50	727	70	850	100	972	0
8	487	30	607	50	727	70	850	100	972	0
9	487	30	607	50	727	70	850	100	972	0
10	487	30	607	50	727	70	850	100	972	0

Berdasarkan hasil dari Tabel 5.16. perhitungan manual serta perhitungan yang dilakukan oleh *hardware* arduino didapatkan hasil nilai *centroid* yang sama pada setiap klasternya.

Pada implementasi otomatisasi lampu, didapatkan 5 hasil perubahan terang redup cahaya berdasarkan nilai terang redup cahaya dari perhitungan algoritma K-means, yaitu:

- 1) Terang redup cahaya 30 % pada menit ke 487 sampai 606.
- 2) Terang redup cahaya 50 % pada menit ke 607 sampai 726.
- 3) Terang redup cahaya 70 % pada menit ke 727 sampai 849.
- 4) Terang redup cahaya 100 % pada menit ke 850 sampai 971.
- 5) Terang redup cahaya 0 % pada menit ke 972 sampai 1439 dan menit ke 0 sampai ke 486.

Pada hasil percobaan secara manual dan perhitungan sistem, didapatkan perbedaan hasil centroid yang dihasilkan, hal ini dikarenakan pada sistem saya melakukan pembulatan data keatas, sedangkan pada perhitungan manual (dilakukan pada Microsoft Excel) melakukan pembulatan kebawah.

Hasil dari perubahan terang redup cahaya tersebut dapat dilihat dari gambar 5.20.



Gambar 5.20. Hasil Pengujian perubahan intensitas cahaya lampu

5.8. Pengujian Keseluruhan Sistem Dengan Data Acak

Pengujian ini dilakukan untuk menghitung akurasi dari algoritma K-means dalam sistem yang telah dibuat dan kemudian dilakukan analisis dari pengujian tersebut.

5.8.1. Skenario pengujian

Skenario pengujian seluruh sistem adalah dimana data acak terang redup cahaya dikirimkan dari modul *remote control* melalui modul NRF24L01 selanjutnya data tersebut diterima oleh modul aktuator dan data terang redup cahaya tersebut langsung dieksekusi kerangkaian *dimmer* untuk mengatur terang redup cahaya, setelah itu nilai dari terang redup cahaya dan waktu eksekusi lampu tersebut disimpan kedalam EEPROM sebagai data historis yang nantinya digunakan sebagai masukan dari algoritma K-means. Pada saat jam 00:00 data yang tersimpan dalam EEPROM akan diambil dan dilakukan perhitungan algoritma K-means dan dari hasil perhitungan tersebut didapatkan hasil dari *centroid* yang berupa data waktu serta terang redup cahaya. Data tersebut digunakan pada proses otomatisasi pada tahap selanjutnya atau pada hari berikutnya.

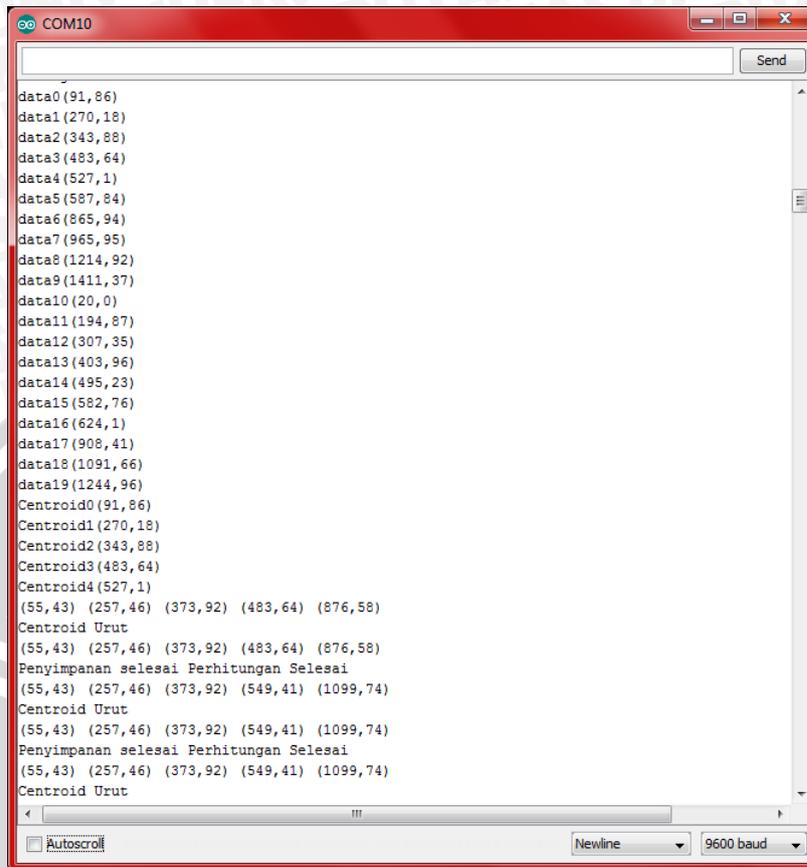
5.8.2. Analisis pengujian

Pengujian ini dilakukan dalam waktu 2 hari dimana setiap hari memberi masukan berupa menyalakan (Terang redup cahaya) atau mematikan lampu melalui *remote control*, masukkan dalam pengujian ini sesuai dengan Tabel 5.17. berikut:

Tabel 5.17. Data uji acak

Hari Ke	Data				
	No	Waktu			Intensitas
		Jam	Menit	Total Linier	
1	1	1	31	91	86
	2	4	30	270	18
	3	5	43	343	88
	4	8	3	483	64
	5	8	47	527	1
	6	9	47	587	84
	7	14	25	865	94
	8	16	5	965	95
	9	20	14	1214	92
	10	23	31	1411	37
2	11	0	20	20	0
	12	3	14	194	87
	13	5	7	307	35
	14	6	43	403	96
	15	8	15	495	23
	16	9	42	582	76
	17	10	24	624	1
	18	15	8	908	41
	19	18	11	1091	66
	20	20	44	1244	96

Proses perhitungan algoritma K-means dari Tabel 5.17. diatas, data masukan diatas dilakukan tiap hari yaitu pukul 00.00 dari proses perhitungan tersebut langsung diterapkan ke bentuk otomatisasi. Pada implementasinya hasil perhitungan dari algoritma K-means pada *hardware* dapat dilihat pada gambar 5.21 berikut ini.



```
COM10
data0 (91,86)
data1 (270,18)
data2 (343,88)
data3 (483,64)
data4 (527,1)
data5 (587,84)
data6 (865,94)
data7 (965,95)
data8 (1214,92)
data9 (1411,37)
data10 (20,0)
data11 (194,87)
data12 (307,35)
data13 (403,96)
data14 (495,23)
data15 (582,76)
data16 (624,1)
data17 (908,41)
data18 (1091,66)
data19 (1244,96)
Centroid0 (91,86)
Centroid1 (270,18)
Centroid2 (343,88)
Centroid3 (483,64)
Centroid4 (527,1)
(55,43) (257,46) (373,92) (483,64) (876,58)
Centroid Urut
(55,43) (257,46) (373,92) (483,64) (876,58)
Penyimpanan selesai Perhitungan Selesai
(55,43) (257,46) (373,92) (549,41) (1099,74)
Centroid Urut
(55,43) (257,46) (373,92) (549,41) (1099,74)
Penyimpanan selesai Perhitungan Selesai
(55,43) (257,46) (373,92) (549,41) (1099,74)
Centroid Urut
```

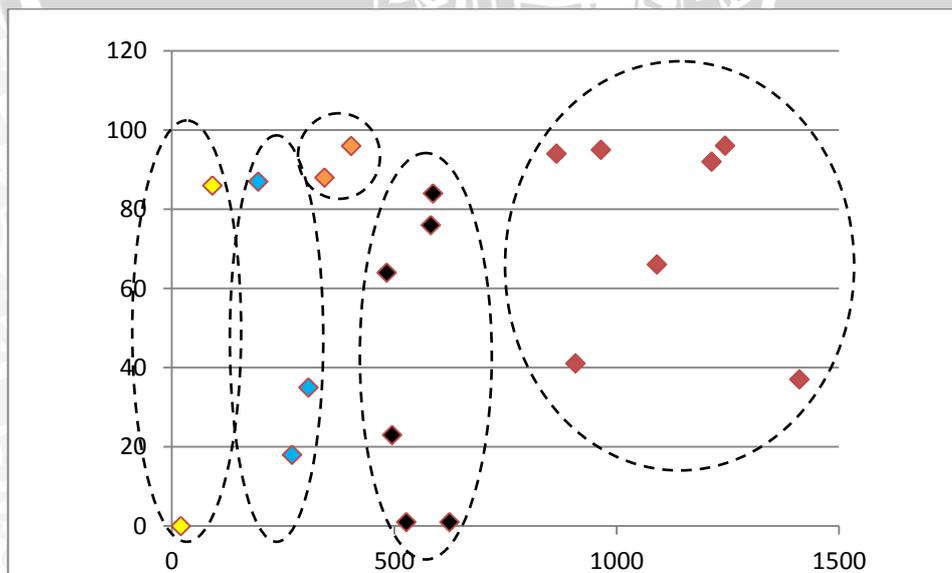
Gambar 5.21. Hasil perhitungan sistem

Proses perhitungan K-means manual dari data masukkan selama 2 hari dengan data acak ini dilakukan untuk mengetahui akurasi perhitungan manual dengan sistem didapatkan hasil pada Tabel 5.18. sebagai berikut.

Tabel 5.18. Hasil akhir perhitungan acak

Data Centroid	Cluster 1 Baru		Cluster 2 Baru		Cluster 3 Baru		Cluster 4 Baru		Cluster 5 Baru	
	Waktu	Intensitas								
Iterasi / Random	91	86	270	18	343	88	483	64	527	1
1	56	43	257	47	373	92	483	64	876	59
2	56	43	257	47	373	92	550	42	1100	74
3	56	43	257	47	373	92	550	42	1100	74
4	56	43	257	47	373	92	550	42	1100	74
5	56	43	257	47	373	92	550	42	1100	74
6	56	43	257	47	373	92	550	42	1100	74
7	56	43	257	47	373	92	550	42	1100	74
8	56	43	257	47	373	92	550	42	1100	74
9	56	43	257	47	373	92	550	42	1100	74
10	56	43	257	47	373	92	550	42	1100	74
11	56	43	257	47	373	92	550	42	1100	74
12	56	43	257	47	373	92	550	42	1100	74
13	56	43	257	47	373	92	550	42	1100	74
14	56	43	257	47	373	92	550	42	1100	74
15	56	43	257	47	373	92	550	42	1100	74

Berdasarkan hasil dari tabel diatas, maka data masukkan dapat dikelompokkan pada setiap kluster, data yang sudah dikelompokkan dapat dilihat pada Gambar 5.22. diagram kartesian berikut.



Gambar 5.22. Diagram grafik hasil pengklasteran

5.9. Analisis Pengujian dari kedua pengujian keseluruhan sistem

Pada hasil dari pengujian didapatkan presentase tingkat error pada perhitungan algoritma K-means dengan rumus perhitungan sebagai berikut:

$$\text{Hasil Error Per klaster} = \frac{|A|}{B} 100\%$$

Keterangan : A = Selisih perhitungan manual dengan perhitungan *hardware* pada setiap klaster

B = Jumlah maksimal (Centroid waktu nilainya 100 dan centroid terang redup 1439)

Untuk mengetahui persentase error secara keseluruhan dapat digunakan rumus sebagai berikut:

$$\text{Hasil Error Total} = \frac{X}{Z} 100\%$$

Keterangan : X = Jumlah total Hasil Error per klaster

B = Jumlah total klaster

- 1) Analisis error pada Pengujian keseluruhan pada sub bab 5.7.
 - a) Pada hasil *centroid* terang redup cahaya didapatkan tingkat error 0% dikarenakan hasil dari perhitungan *hardware* dan perhitungan manual didapatkan hasil yang sama. Hasil tersebut diperoleh dari perhitungan sebagai berikut sesuai dengan rumus diatas.

$$\text{Error Per klaster 1} = \frac{(30-30)}{100} \times 100\%$$

$$\text{Error Per klaster 1} = 0\%$$

$$\text{Error Per klaster 2} = \frac{(50-50)}{100} \times 100\%$$

$$\text{Error Per klaster 2} = 0\%$$

$$\text{Error Per klaster 3} = \frac{(70-70)}{100} \times 100\%$$

$$\text{Error Per klaster 3} = 0\%$$

$$\text{Error Per klaster 4} = \frac{(100-100)}{100} \times 100\%$$

$$\text{Error Per klaster 4} = 0\%$$

$$\text{Error Per klaster 5} = \frac{(0-0)}{100} \times 100\%$$

$$\text{Error Per klaster 5} = 0\%$$

Jadi total error secara keseluruhan pada centroid persentase cahaya lampu adalah sebagai berikut:

$$\text{Total Error} = \frac{0 + 0 + 0 + 0 + 0}{5}$$

$$\text{Total Error} = 0\%$$

b) Pada hasil *centroid* waktu didapatkan hasil error total sebesar 0.055%. Hasil itu didapat dengan perhitungan sebagai berikut.

$$\text{Error Per klaster 1} = \frac{(487-486)}{1439} \times 100\%$$

$$\text{Error Per klaster 1} = 0,069\%$$

$$\text{Error Per klaster 2} = \frac{(607-606)}{1439} \times 100\%$$

$$\text{Error Per klaster 2} = 0,069\%$$

$$\text{Error Per klaster 3} = \frac{(727-727)}{1439} \times 100\%$$

$$\text{Error Per klaster 3} = 0\%$$

$$\text{Error Per klaster 4} = \frac{(850-849)}{1439} \times 100\%$$

$$\text{Error Per klaster 4} = 0,069\%$$

$$\text{Error Per klaster 5} = \frac{(972-971)}{1439} \times 100\%$$

$$\text{Error Per klaster 5} = 0,069\%$$

Jadi total error secara keseluruhan pada centroid persentase cahaya lampu adalah sebagai berikut:

$$\text{Total Error} = \frac{0,069 + 0,069 + 0 + 0,069 + 0,069}{5}$$

$$\text{Total Error} = 0,055\%$$

2) Analisis error pada Pengujian keseluruhan pada sub bab 5.8.

- a) Pada hasil *centroid* terang redup cahaya didapatkan tingkat error 0% dikarenakan hasil dari perhitungan *hardware* dan perhitungan manual didapatkan hasil yang sama. Hasil tersebut diperoleh dari perhitungan sebagai berikut sesuai dengan rumus diatas.

$$\text{Error Per klaster 1} = \frac{(43 - 43)}{100} \times 100\%$$

$$\text{Error Per klaster 1} = 0\%$$

$$\text{Error Per klaster 2} = \frac{(47-47)}{100} \times 100\%$$

$$\text{Error Per klaster 2} = 0\%$$

$$\text{Error Per klaster 3} = \frac{(92-92)}{100} \times 100\%$$

$$\text{Error Per klaster 3} = 0\%$$

$$\text{Error Per klaster 4} = \frac{(42-42)}{100} \times 100\%$$

$$\text{Error Per klaster 4} = 0\%$$

$$\text{Error Per klaster 5} = \frac{(74-74)}{100} \times 100\%$$

$$\text{Error Per klaster 5} = 0\%$$

Jadi total error secara keseluruhan pada *centroid* persentase cahaya lampu adalah sebagai berikut:

$$\text{Total Error} = \frac{0 + 0 + 0 + 0 + 0}{5}$$

$$\text{Total Error} = 0\%$$

- b) Pada hasil *centroid* waktu didapatkan hasil error total sebesar 0.055 %.hasil itu didapat dengan perhitungan sebagai berikut.

$$\text{Error Per klaster 1} = \frac{(56-55)}{1439} \times 100\%$$

$$\text{Error Per klaster 1} = 0,069\%$$

$$\text{Error Per klaster 2} = \frac{(257-257)}{1439} \times 100\%$$

Error Per klaster 2=0%

$$\text{Error Per klaster 3} = \frac{(374-373)}{1439} \times 100\%$$

Error Per klaster 3=0,069%

$$\text{Error Per klaster 4} = \frac{(550-549)}{1439} \times 100\%$$

Error Per klaster 4=0,069%

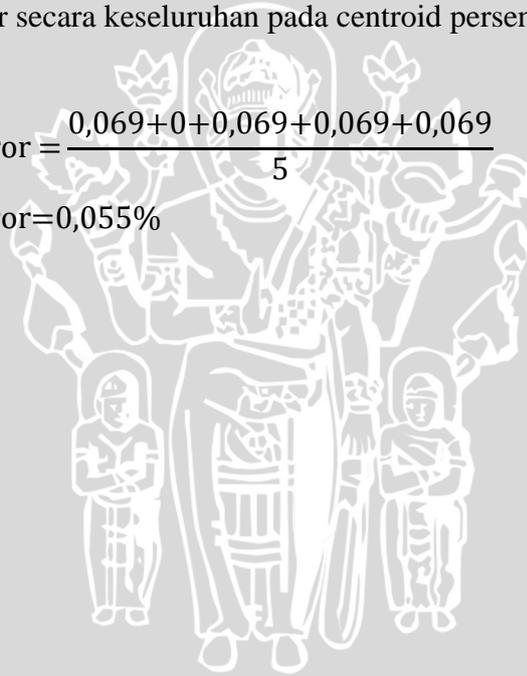
$$\text{Error Per klaster 5} = \frac{(1100-1099)}{1439} \times 100\%$$

Error Per klaster 5=0,069%

Jadi total error secara keseluruhan pada centroid persentase cahaya lampu adalah sebagai berikut:

$$\text{Total Error} = \frac{0,069+0+0,069+0,069+0,069}{5}$$

$$\text{Total Error}=0,055\%$$



BAB VI

PENUTUP

6.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut:

- 1) Untuk membuat sistem otomatisasi mati-nyala lampu serta terang redup cahaya pada *smart home* dapat dilakukan dengan cara menghubungkan arduino promini dengan rangkaian *dimmer* dan rangkaian RTC. Untuk proses otomatisasi, program algoritma K-means ditanamkan pada arduino promini yang bertindak sebagai kontrolernya.
- 2) Mekanisme pembelajaran pada penelitian ini yaitu sistem menerima semua masukan dari *user* (masukkan terang redup cahaya lampu) dan sistem langsung mengeksekusi perintah dari *user* tersebut. Setelah itu data yang menjadi masukan *user* tersebut disimpan ke dalam EEPROM sebagai riwayat aktivitas. Selanjutnya jika waktu menunjukkan 00.00.00 maka data yang disimpan di EEPROM diambil dan diolah menggunakan algoritma K-means. Dalam algoritma ini, data tersebut dikalkulasi dan dipisahkan menjadi beberapa klaster yang memiliki kriteria yang hampir sama. Klaster-klaster tersebut yang dijadikan acuan sebagai bentuk otomatisasi pada sistem ini.
- 3) Dari penelitian ini didapatkan hasil presentase error maksimal sebesar 0,055 %, hasil itu hanya untuk *centroid* waktu. Hal itu disebabkan karena format yang dipakai dalam perhitungan *hardware* berupa tipe data int, pemilihan tipe data tersebut untuk menghemat konsumsi memori *hardware*. Dan tipe data int melakukan pembulatan nilai kebawah. Untuk *presentase error centroid* terang redup cahaya sebesar 0 %.
- 4) Dari segi memori arduino pro mini 328/16Mhz, Memori yang dimiliki arduino pro mini adalah 1 KB dan dari memori tersebut hanya dapat mengolah data atau mengkalkulasi data sebanyak 20 data yang berupa nilai 1 *byte* yang digunakan untuk terang redup cahaya dan 2 *byte* yang digunakan untuk nilai waktu.

- 5) Berdasarkan perancangan dan implementasi serta pengujian, sistem ini dapat berjalan otomatis dalam mengatur terang redup cahaya sesuai dengan waktu yang ditentukan pada proses pembelajaran yang menggunakan algoritma K-means.

6.2. Saran

Saran yang dapat diberikan untuk pengembangan perangkat lunak ini antara lain:

- 1) Untuk pengembangan lebih lanjut, bisa menggunakan *hardware* yang memiliki SRAM yang lebih besar agar dapat melakukan kalkulasi data yang lebih banyak.
- 2) Untuk pengembangan bisa menggunakan lampu LED sebagai aktuator pengganti lampu pijar.
- 3) Untuk pengembangan lebih lanjut, dengan *hardware* yang mampu menerima jumlah data dan mengolah data yang lebih banyak serta memiliki SRAM yang lebih besar maka jumlah kluster juga dapat diperbanyak dan bersifat fleksibel sesuai dengan keadaan tiap harinya.

DAFTAR PUSTAKA

- Chen, Y. a. (2014). Pattern Recognition Using Clustering Algorithm for Scenario Definition in Traffic Simulation-based Decision Support Systems. *Jurnal IEEE, Qingdao, China.*
- Dubois, M.-C. a. (2011). Energi saving potential and strategies for electric lighting in future North European, low energi office buildings: A literature review. *Inst. of Architecture and Built Environment, Div. of Energi and Building Design, Lund University, P.O. Box 118, SE-221 00, Lund, Sweden.*
- Instruments, Texas. (1996). MOC3020 THRU MOC3023 OPTOCOUPPLERS/OPTOISOLATORS.
- Kling, W. L. (2013). Energi Efficiency in Smart Cities. *University of Technology, Eindhoven, the Netherlands, TU Dortmund University Dortmund, Germany, Eindhoven, Netherlands.*
- Maxim Integrated. (2008). DS1302 Trickle-Charge Timekeeping Chip.
- Nordic Semiconductor. (2008). nRF24L01+ Single Chip 2.4GHz Transceiver Preliminary Product Specification v1.0.
- Nur, W. (2012). CLUSTERING MENGGUNAKAN K-MEANS ALGORITHM (K-MEANS ALGORITHM CLUSTERING).
- O.J, O. (2010). Application of k-Means Clustering algorithm for prediction of Students' Academic Performance . *Department of Computer and Information Sciences, College of Science and Technology, Covenant University. Ora ,Nigeria.*
- Santosa, B. (2007). Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis. *Graha Ilmu, Yogyakarta.*
- Saputro, E. (2013). Rancang Bangun Pengendali Intensitas Cahaya Lampu Jarak Jauh Berbasis Modulasi Ask. *Teknik Elektro, Fakultas Teknik dan Ilmu Komputer, UNIKOM, Bandung.*
- Semiconductors, Philips. (2001). BT16 series.
- Semiconductors, Vishay. (2010). Optocoupler, Phototransistor Output with Base Connection.

- Sherif, M. (2013). An Intelligent Light Control Sistem for Power Saving. *Michigan: Jurnal IEEE, Wayne State University, Detroit, Michigan.*
- Sohraby, D. M. (2007). *Wireless Sensor network: teknologi, protocols, and applications.* New Jersey.
- Tan, Y. K. (2013). Smart Personal Sensor Network Control for Energi Saving in DC Grid Powered LED Lighting Sistem . *jurnal IEEE.*
- Y, A. (2007). K-means - Penerapan, Permasalahan dan Metode Terkait. *Jurnal Sistem dan Informatika Vol. 3.*
- Zatos, N. (2012). Case Study of a dimmable outdoor lighting sistem with intelligent management and remote control. *Department of Information & Communications Sitems Engineering, University of Aegean , Karlovasi , Greece.*

