# PENGHEMATAN ENERGI LISTRIK DENGAN APLIKASI SENSOR PADA LAMPU

**KARYA ILMIAH**


Disusun oleh:

| | | |
|---|---|---|
| 1. 29782 | Angeline Naomi Tan | XII MIPA 3/02 |
| 2. 29924 | Genieve Hartono | XII MIPA 3/12 |
| 3. 30053 | Lucky Maximillian | XII MIPA 3/22 |
| 4. 30122 | Rafaela Princess A. S. | XII MIPA 3/28 |
| 5. 30187 | Vincent Wijaya Salim | XII MIPA 3/32 |
| 6. 30197 | Yorico Fernando | XII MIPA 3/36 |



**SMA KATOLIK ST. LOUIS 1**
**SURABAYA**
**2025**

# PENGHEMATAN ENERGI LISTRIK DENGAN APLIKASI SENSOR PADA LAMPU

## KARYA ILMIAH

Merupakan Ujian Keterampilan dan Syarat Kelulusan


Disusun oleh:

| | | |
|---|---|---|
| 1. 29782 | Angeline Naomi Tan | XII MIPA 3/02 |
| 2. 29924 | Genieve Hartono | XII MIPA 3/12 |
| 3. 30053 | Lucky Maximillian | XII MIPA 3/22 |
| 4. 30122 | Rafaela Princess A. S. | XII MIPA 3/28 |
| 5. 30187 | Vincent Wijaya Salim | XII MIPA 3/32 |
| 6. 30197 | Yorico Fernando | XII MIPA 3/36 |



**SMA KATOLIK ST. LOUIS 1**
**SURABAYA**
**2025**

# LEMBAR PENGESAHAN KARYA ILMIAH

Judul : Penghematan Energi Listrik dengan Aplikasi Sensor pada Lampu

| | | | |
|---|---|---|---|
| Penyusun | 1. 29782 | Angeline Naomi Tan | XII MIPA 3 / 02 |
| | 2. 29924 | Genieve Hartono | XII MIPA 3 / 12 |
| | 3. 30053 | Lucky Maximillian Winarjo | XII MIPA 3 / 22 |
| | 4. 30122 | Rafaela Princess Artha Sinathrika | XII MIPA 3 / 28 |
| | 5. 30187 | Vincent Wijaya Salim | XII MIPA 3 / 32 |
| | 6. 30193 | Yorico Fernando | XII MIPA 3 / 36 |

Pembimbing I : Irmina Indiyarti, S.Pd.

Pembimbing II : Fransiskus Asisi Subono, S.Si., M.Kes.

Tanggal Presentasi : Selasa, 4 Februari 2025

Disetujui oleh:

| Pembimbing 1 | Pembimbing 2 |
|---|---|
| Irmina Indiyarti, S.Pd. | Fransiskus Asisi Subono, S.Si., M.Kes. |

Mengetahui,

Kepala Sekolah

SMAK ST. LOUIS 1 SURABAYA

Dra. Sri Wahjoeni Hadi S.

# KATA PENGANTAR

Puji syukur kami haturkan kepada Tuhan yang Maha Esa karena atas berkat dan rahmat-Nya kami dapat melaksanakan penyusunan laporan pelaksanaan ujian praktik peminatan MIPA dengan judul "Penghematan Energi Listrik dengan Aplikasi Sensor pada Lampu". Adapun tujuan dari penelitian ini adalah sebagai salah satu syarat ujian keterampilan dan kelulusan.

Selama proses penyusunan proposal ini, kami mendapat banyak bantuan dan masukan dari berbagai pihak. Oleh karena itu, kami hendak menyampaikan terimakasih kepada:

1. Dra. Sri Wahjoeni Hadi S., selaku kepala SMA Katolik St. Louis 1 Surabaya,
2. Irmina Indiyarti, S.Pd., selaku wali kelas XII MIPA 3 dan guru mata pelajaran Fisika, sekaligus pembimbing I atas dukungan dan masukan yang diberikan dalam penyusunan proposal ini,
3. F. Asisi Subono, S.Si., M.Kes., selaku pembimbing II dan guru mata pelajaran Biologi,
4. Y. Hari Suyanto, S.Pd., M.Si., selaku guru mata pelajaran Matematika,
5. Dra. Maria Viciati, MM., selaku guru mata pelajaran Kimia,

Dengan demikian, laporan penelitian ini dibuat dengan seluruh kemampuan penulis agar dapat bermanfaat bagi pembaca. Penulis menyadari laporan penelitian ini jauh dari kata sempurna. Oleh karena itu, besar harapan penulis untuk mendapat saran dan kritik yang membangun untuk meningkatkan kualitas penelitian yang akan dilakukan.

Surabaya, 4 Februari 2025

Penyusun,

(Vincent Wijaya Salim)

Ketua Kelompok

# DAFTAR ISI

HALAMAN SAMPUL.....
HALAMAN JUDUL.....i
LEMBAR PENGESAHAN NASKAH PROPOSAL KARYA ILMIAH..... ii
KATA PENGANTAR..... iii
DAFTAR ISI..... iv
DAFTAR GAMBAR.....vi
DAFTAR TABEL..... vii
DAFTAR SINGKATAN..... vii
DAFTAR SIMBOL..... ix
BAB I PENDAHULUAN..... 1
1.1 Latar Belakang..... 1
1.2 Rumusan Masalah..... 2
1.3 Tujuan Penelitian.....2
1.4 Manfaat Penelitian.....3
BAB II TINJAUAN PUSTAKA..... 4
2.1 Lampu Pijar..... 4
2.2 Arduino.....5
2.3 RTC DS3231..... 7
2.4 Relay Module..... 8
2.5 AC Dimmer Module.....9
2.6 Wattmeter..... 10
2.7 Liquid Crystal Display (LCD).....11
2.8 Light Dependent Resistor (LDR)..... 12
BAB III METODE PENELITIAN.....13
3.1 Tempat dan Waktu Penelitian.....13
3.2 Alat dan Bahan Penelitian..... 13
3.2.1 Alat..... 13

3.2.2 Bahan....13
3.3 Tahapan Penelitian.... 14
3.3.1 Diagram Alur Penelitian....14
3.3.2 Skema Pembuatan Alat.... 14
3.3.3 Pengambilan Data Penelitian.... 24
3.3.4 Variabel Penelitian....25
3.4 Metode dan Analisis Data.... 25
3.4.1 Metode Penelitian....25
3.4.2 Teknik Analisis Data.... 26
BAB IV HASIL PENELITIAN DAN PEMBAHASAN....27
4.1 Hasil Penelitian.... 27
4.1.1 Perbandingan Konsumsi Energi pada Lampu Biasa, Lampu Sensor Cahaya, dan Lampu Sensor Waktu....27
4.2 Pembahasan....29
4.2.1 Cara Kerja Dua Sistem Lampu Sensor Hemat Energi.... 29
4.2.2 Desain Rangkaian Listrik yang Mendukung Kerja Sensor dan Lampu....33
BAB V KESIMPULAN DAN SARAN....35
5.1 Kesimpulan....35
5.2 Saran....36
DAFTAR PUSTAKA....37
LAMPIRAN....39

# DAFTAR GAMBAR

GAMBAR 2.1 LAMPU PIJAR..…….....…….……..…4
GAMBAR 2.2 ARDUINO BOARD (ESP32)..... 5
GAMBAR 2.3 REAL TIME CLOCK MODULE..... 7
GAMBAR 2.4 RELAY MODULE..... 8
GAMBAR 2.5 AC DIMMER MODULE..... 9
GAMBAR 2.6 WATTMETER.....10
GAMBAR 2.7 LIQUID CRYSTAL DISPLAY (LCD).....11
GAMBAR 2.8 LIGHT DEPENDENT RESISTOR.....12
GAMBAR 3.3.1 DIAGRAM ALUR PENELITIAN.....14
GAMBAR 3.3.2.1 PEMBUATAN PROTOTIPE.....14
GAMBAR 3.3.2.2 CODING SENSOR WAKTU……17
GAMBAR 3.3.2.3 CODING SETTING OLED LCD…….18
GAMBAR 3.3.2.4 CODING SCANNING OLED LCD…….19
GAMBAR 3.3.2.5 CODING FORMAT LCD……..21
GAMBAR 3.3.2.6 CODING SETTING RTC……..22
GAMBAR 3.3.2.7 CODING SENSOR CAHAYA……..23
GAMBAR 3.3.2.8 CODING TIMERONE SENSOR CAHAYA……23
GAMBAR 4.1.1.1 GRAFIK PENGGUNAAN LAMPU BIASA INDOOR…….. 28
GAMBAR 4.1.1.2 GRAFIK PENGGUNAAN LAMPU BIASA OUTDOOR……...29
GAMBAR 4.2.1.1 GRAFIK NILAI LDR DALAM 24 JAM……..30
GAMBAR 4.2.1.2 NILAI LDR PER 10 MENIT DALAM 24 JAM ……..31
GAMBAR 4.2.1.3 CODING PEMROGRAMAN LAMPU SENSOR LDR……...32
GAMBAR 4.2.1.4 CODING PEMROGRAMAN LAMPU SENSOR WAKTU……33
GAMBAR 4.2.2.1 DESAIN RANGKAIAN LISTRIK SENSOR CAHAYA……..……...33
GAMBAR 4.2.2.2 DESAIN RANGKAIAN LISTRIK SENSOR WAKTU……...34

GAMBAR 6.1 DESIGN MAKET……...39
GAMBAR 6.2 PEMBELIAN BAHAN-BAHAN PEMBUATAN MAKET……...39
GAMBAR 6.3 PROSES PEMBUATAN MAKET……..39
GAMBAR 6.4 HASIL JADI MAKET…….40

# DAFTAR TABEL

TABEL 2.2  TABEL SPESIFIKASI ARDUINO UNO R3.....6

TABEL 3.3.2 TABEL PENGAMBILAN DATA..... 24

TABEL 4.1.1.1 TABEL KONSUMSI ENERGI LAMPU DENGAN SISTEM SENSOR CAHAYA.....27

TABEL 4.1.1.2 TABEL KONSUMSI ENERGI LAMPU DENGAN SISTEM SENSOR WAKTU..... 27

TABEL 4.1.1.3 TABEL KONSUMSI ENERGI LAMPU BIASA INDOOR.....28

TABEL 4.1.1.4 TABEL KONSUMSI ENERGI LAMPU BIASA OUTDOOR.....29

# DAFTAR SINGKATAN

| Singkatan | Kepanjangan |
|---|---|
| AC | *Alternate Current* |
| DC | *Direct Current* |
| LCD | Liquid Crystal Display |
| RTC | Real Time Clock |
| PWM | Pulse Width Modulation |
| LDR | Light Dependent Resistor |
| PCB | Printed Circuit Board |
| TCXO | Temperature Compensated Crystal Oscillator |
| MW | Megawatt |
| MVA | Megavolt Ampere |
| RUPTL | Rencana Usaha Penyediaan Tenaga Listrik |
| PLN | Perusahaan Listrik Negara |
| USB | *Universal Serial Bus* |

# DAFTAR SIMBOL

| | |
|---|---|
| | *Start / Stop Point*<br>Menggambarkan permulaan atau juga akhir dari suatu proses diagram alir. |
| | *Arrow*<br>Menggambarkan arah dari proses diagram alir. |
| | *Process*<br>Menunjukkan proses yang terjadi pada diagram alir. |
| | *Decision*<br>Menggambarkan proses keputusan dalam proses diagram alir. |

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Kebutuhan penggunaan energi selalu berkembang sejalan dengan perkembangan manusia. Hal ini sejalan dengan naiknya tingkat teknologi manusia, dimana salah satu energi yang sering digunakan untuk kehidupan sehari-hari adalah energi listrik. Energi listrik merupakan salah satu komponen utama penggerak teknologi manusia saat ini, baik dalam bidang jasa, transportasi, komunikasi dan lain-lain. Hal ini memicu peningkatan kebutuhan energi listrik secara berlebihan, sehingga harus menggunakan metode yang berdampak buruk dalam menghasilkan energi listrik.

Berdasarkan Rencana Usaha Penyediaan Tenaga Listrik (RUPTL) PT. PLN 2013-2022, kebutuhan listrik Indonesia tumbuh rata-rata 8,4% per tahun. Setiap tahun diperlukan tambahan pasokan listrik sekitar 5.700 MW, dengan target 60 GW hingga 2022. Dibutuhkan juga pembangunan jaringan transmisi 58 ribu km dan gardu induk 134 ribu MVA, dengan total investasi Rp 88,4 triliun per tahun, sementara kemampuan PLN hanya Rp 60 triliun. Hal ini menunjukkan bahwa peningkatan kebutuhan listrik harus diimbangi dengan kebijakan penghematan dan sumber energi baru untuk menghindari kelangkaan energi di masa depan.

Usaha untuk melakukan penghematan energi listrik dalam jumlah besar terus diupayakan. Menurut data dari BPS dan IEA, salah satu sektor penggunaan energi listrik terbesar dalam kehidupan sehari-hari adalah pencahayaan, yang menyumbang sekitar 20% dari total konsumsi listrik global. Secara umum, penghematan energi dapat dilakukan dengan dua cara, yakni meningkatkan efisiensi teknologi yang digunakan maupun mengubah perilaku pengguna. Penelitian ini bertujuan untuk mengurangi konsumsi energi listrik di sektor pencahayaan, yang merupakan salah satu sektor dengan konsumsi energi terbesar dalam kehidupan sehari-hari. Pada bangunan, pencahayaan berkontribusi antara 20-60% dari total daya yang dikonsumsi, sehingga penghematan energi pada sektor ini dapat memberikan dampak signifikan terhadap pengurangan konsumsi energi secara keseluruhan.

Melalui penelitian ini, diharapkan terjadi pengembangan teknologi otomatis yang dapat mengatur kebiasaan perilaku pengguna guna meningkatkan efisiensi melalui pemanfaatan sensor elektronik.

Menurut penelitian Welman, sistem sensor dapat membantu manusia untuk mengontrol berbagai perangkat elektronik agar dapat berfungsi secara otomatis, salah satunya melalui sistem penerangan lampu elektronika. Dengan demikian, penggunaan sensor elektronik memungkinkan pengendalian otomatis pada sejumlah peralatan elektronik, terutama dalam sistem penerangan lampu, yang berpotensi meningkatkan efisiensi energi.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, dapat disimpulkan rumusan masalah sebagai berikut.

1. Bagaimana cara kerja sensor lampu elektronik agar dapat bekerja otomatis?
2. Bagaimana desain rangkaian listrik yang dapat mendukung kerja sensor dan lampu agar hemat energi dan tahan lama?
3. Bagaimana cara mengatur terang lampu sensor agar sesuai dengan waktu dan cahaya di sekitarnya?

## 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, dapat disimpulkan tujuan penelitian sebagai berikut.

1. Mendeskripsikan cara kerja sensor lampu elektronik agar dapat bekerja secara otomatis.
2. Mendeskripsikan desain rangkaian listrik yang dapat mendukung kerja sensor dan lampu yang hemat energi dan tahan lama.
3. Mendeskripsikan cara mengatur terang lampu sensor agar sesuai dengan waktu dan cahaya di sekitarnya.

## 1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian di atas, dapat disimpulkan manfaat penelitian sebagai berikut.

1. Siswa dapat memahami cara kerja sensor lampu elektronik otomatis.
2. Siswa memperoleh pengalaman dalam melakukan percobaan pembuatan prototipe proyek.
3. Siswa dapat mengetahui cara mengatur terang lampu otomatis agar sesuai dengan waktu dan cahaya di sekitarnya.

# BAB II
# TINJAUAN PUSTAKA

## 2.1 Lampu Pijar

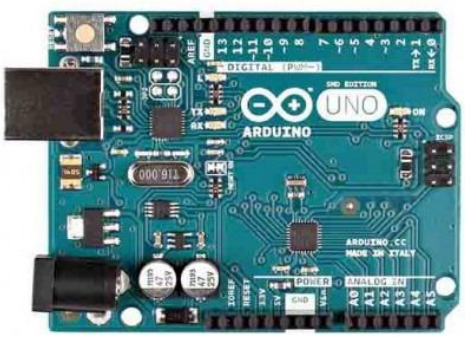
Lampu pijar adalah sumber cahaya buatan yang dihasilkan melalui aliran arus listrik lewat filamen yang memanas sehingga bisa menghasilkan cahaya. Kaca yang menyelubungi filamen panas tersebut menghalangi udara untuk langsung berhubungan dengan filamen, membuatnya tidak langsung rusak akibat teroksidasi.

*Gambar 2.1 Lampu Pijar*

Lampu pijar dipasarkan dalam berbagai bentuk dan tersedia dalam berbagai voltase mulai dari 1,25 volt hingga 300 volt. Energi listrik yang diperlukan lampu pijar untuk menghasilkan cahaya lebih besar dibandingkan lampu jenis lainnya seperti lampu pendar dan diode cahaya. Sekitar 90% daya yang digunakan oleh lampu pijar dilepaskan sebagai radiasi panas dan hanya 10% yang dipancarkan dalam cahaya. Oleh karena itu, beberapa penggunaan lampu pijar lebih memanfaatkan panas yang dihasilkan, contohnya sebagai pemanas kandang ayam dan pemanas inframerah di bidang industri.

## 2.2 Arduino

Arduino adalah platform elektronik *open-source* yang dirancang untuk mempermudah pembuatan proyek berbasis mikrokontroler. Terdiri dari perangkat keras berupa papan mikrokontroler dan perangkat lunak berupa lingkungan pemrograman, Arduino sangat populer untuk aplikasi IoT, robotika, dan otomasi. Platform ini memungkinkan pengguna dari berbagai tingkat keahlian, mulai dari pemula hingga profesional, untuk mengembangkan prototipe elektronik dengan cepat dan mudah.




*Gambar 2.2 Arduino Board (ESP32)*

Arduino memiliki beberapa komponen penting di dalamnya, seperti pin, mikrokontroler, dan konektor yang nanti akan dibahas lebih dalam selanjutnya. Selain itu, Arduino juga sudah menggunakan bahasa pemrograman *Arduino Language* yang sedikit mirip dengan bahasa pemrograman C++.

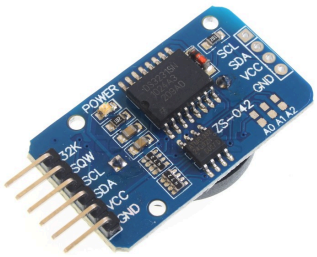
Dari segi material, perangkat keras Arduino dibuat dari PCB berbahan *fiberglass* dengan komponen elektronik seperti pin input/output, konektor USB, dan regulator tegangan. Ekosistem *open-source* Arduino juga mendukung library tambahan, menjadikannya alat yang fleksibel untuk berbagai proyek teknologi.

| Parameter | Spesifikasi |
|---|---|
| Mikrokontroler | ATmega328 |
| Tegangan Operasi | 5V |
| Tegangan input (disarankan) | 7-12V |
| Batas Tegangan input | 6-20V |
| *Pin digital I/O* | 14 (dimana 6 pin merupakan output PWM) |
| *Pin Analog Input* | 6 |
| Arus DC per *pin I/O* | 40 mA |
| Arus DC untuk 3.3V | 50 mA |
| *Flash memory* | 32 KB (ATmega), 0.5Kb digunakan sebagai *Bootloader* |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| *Clock* | 16 MHz |

*Tabel 2.2 Spesifikasi Arduino Uno R3*

## 2.3 RTC DS3231

RTC DS3231 adalah modul *Real-Time Clock* yang digunakan untuk mencatat waktu dengan presisi tinggi, bahkan saat perangkat utama dimatikan. Modul ini memiliki osilator kristal yang dikompensasi suhu (*Temperature-Compensated Crystal Oscillator/TCXO*), sehingga tetap akurat meskipun berada di lingkungan dengan perubahan suhu. DS3231 sering digunakan dalam aplikasi yang memerlukan pelacakan waktu seperti jam digital, sistem *data logging*, dan perangkat berbasis IoT.




*Gambar 2.3 Real Time Clock Module*

Modul ini memiliki spesifikasi unggulan seperti akurasi hingga ±2 ppm pada suhu 0°C hingga +40°C, serta kemampuan mencatat waktu dalam format detik, menit, jam, hari, bulan, hingga tahun dengan dukungan untuk tahun kabisat. DS3231 juga dilengkapi dengan memori RAM tambahan, pin alarm, serta kemampuan membaca suhu lingkungan. Keunggulannya dibanding RTC lain seperti DS1307 adalah stabilitas waktu yang lebih baik dan fitur kompensasi suhu, yang menjadikannya pilihan ideal untuk aplikasi yang memerlukan presisi tinggi.

Prinsip kerja DS3231 melibatkan penghitungan waktu menggunakan osilator internalnya yang dikendalikan oleh TCXO. Data waktu dapat diakses melalui antarmuka I2C yang mempermudah integrasi dengan mikrokontroler seperti Arduino atau Raspberry Pi. RTC DS3231 digunakan dalam berbagai aplikasi, seperti sistem pengingat waktu, sensor cuaca, pencatatan data ilmiah, hingga perangkat elektronik pintar, karena kehandalannya dalam menjaga waktu yang konsisten.

## 2.4 Relay Module

*Relay module* adalah komponen elektronik yang berfungsi untuk mengendalikan perangkat dengan arus tinggi menggunakan sinyal listrik dari mikrokontroler atau sistem kendali lainnya yang berarus rendah. Modul relay memungkinkan perangkat seperti lampu, motor, dan alat elektronik lain yang memerlukan daya tinggi untuk dihidupkan atau dimatikan tanpa kontak langsung dengan mikrokontroler. *Relay module* terdiri dari komponen utama seperti relay

mekanis, *driver transistor* atau *optocoupler*, dan pin input yang terhubung ke mikrokontroler.

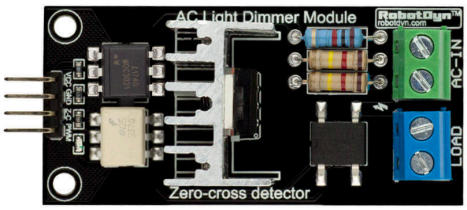
*Gambar 2.4 Relay Module*

*Relay module* umumnya memiliki spesifikasi seperti kapasitas tegangan dan arus tinggi yang dapat dikendalikan (misalnya, hingga 250V AC atau 30V DC), serta tegangan kerja rendah (biasanya 5V atau 12V). Beberapa relay module dilengkapi dengan fitur tambahan seperti LED indikator yang menunjukkan status relay, dan beberapa bahkan memiliki kemampuan untuk mengendalikan beberapa saluran (multi-channel) sekaligus. Keunggulan relay module dibandingkan komponen lain seperti transistor adalah kemampuannya untuk mengendalikan perangkat dengan beban listrik yang jauh lebih besar tanpa membebani mikrokontroler.

Prinsip kerja *relay module* melibatkan penggunaan sinyal listrik dari mikrokontroler untuk mengaktifkan elektromagnet di dalam relay, yang kemudian menghubungkan atau memutuskan jalur sirkuit pada perangkat yang dikendalikan. Ketika sinyal diterima, relay akan menarik saklar untuk menutup atau membuka rangkaian, memungkinkan atau menghentikan aliran listrik ke perangkat eksternal. *Relay module* banyak digunakan dalam berbagai aplikasi sensor, seperti dalam sistem otomasi rumah (pencahayaan otomatis), pengendalian motor, sistem pengatur suhu, dan pengawasan perangkat berbasis mikrokontroler yang memerlukan kontrol daya tinggi.

## 2.5 AC Dimmer Module

Modul dimmer AC adalah perangkat elektronik yang dirancang untuk mengontrol tegangan arus bolak-balik yang disuplai ke sumber cahaya buatan, seperti lampu pijar. Dengan modul ini, pengguna dapat mengatur intensitas cahaya, suhu, atau kecepatan dengan mengubah nilai tegangan RMS yang diterima oleh beban. Modul ini biasanya kompatibel dengan tegangan 3,3V hingga 5V, sehingga dapat diintegrasikan dengan mikrokontroler seperti Arduino




*Gambar 2.5 Dimmer Module*

Spesifikasi *dimmer module* dapat bervariasi tergantung pada tipe dan aplikasinya, namun umumnya mereka dapat mengendalikan beban dengan daya antara 100W hingga 1000W pada tegangan AC (biasanya 110V atau 220V). Beberapa dimmer module dilengkapi dengan fitur tambahan seperti kontrol potensiometer untuk pengaturan manual, antarmuka PWM (*Pulse Width Modulation*) untuk aplikasi lebih presisi, dan perlindungan terhadap lonjakan arus atau overheat. Keunggulan dimmer dibandingkan dengan saklar konvensional adalah kemampuannya untuk memberikan kontrol yang lebih halus dan lebih efisien pada penggunaan energi, serta meningkatkan umur perangkat yang dikendalikan.

Prinsip kerja *dimmer module* berfokus pada pengaturan siklus pemotongan gelombang AC, yaitu dengan memotong sebagian dari gelombang sinyal AC untuk mengurangi jumlah daya yang diteruskan ke beban. Modul ini biasanya bekerja dengan triac atau MOSFET untuk memodifikasi bentuk gelombang arus dan mengatur tingkat kecerahan lampu atau kecepatan motor. *Dimmer module* sering digunakan dalam aplikasi sensor seperti pengaturan pencahayaan otomatis berdasarkan waktu atau

intensitas cahaya lingkungan, serta kontrol kecepatan motor dalam aplikasi industri, alat pengatur suhu, atau sistem otomasi rumah pintar.

## 2.6 Wattmeter

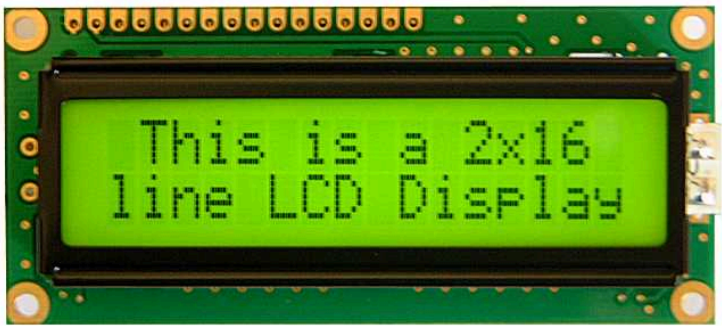
Wattmeter adalah alat untuk mengukur energi listrik dalam rangkaian watt untuk rangkaian sirkuit apapun. Terdapat 2 jenis wattmeter, digital dan analog. Wattmeter terdiri dari kumparan arus dan kumparan tegangan. Kumparan arus dihubungkan secara seri dengan beban dan berfungsi untuk mengukur arus. Kumparan tegangan dihubungkan secara paralel dengan sumber tegangan dan berfungsi untuk mengukur tegangan. Ketika arus mengalir melalui kumparan arus dan tegangan, medan magnet yang dihasilkan akan menyebabkan jarum penunjuk bergerak sesuai dengan daya yang dikonsumsi oleh beban.

*Gambar 2.6 Wattmeter*

Wattmeter digunakan untuk mengukur daya listrik dalam suatu rangkaian, menganalisis konsumsi daya pada perangkat listrik, membantu dalam perhitungan efisiensi energi listrik, dan digunakan untuk pengujian listrik lainnya.

## 2.7 Liquid Crystal Display (LCD)

*Liquid Crystal Display* adalah suatu jenis media tampilan yang menggunakan kristal cair (*liquid crystal)* untuk menghasilkan gambar yang terlihat. Teknologi ini banyak digunakan pada produk elektronik seperti layar laptop, jam digital, monitor komputer, dsb. LCD membutuhkan lampu *backlight* (cahaya latar belakang) sebagai cahaya pendukung karena LCD tidak bisa memancarkan cahaya.

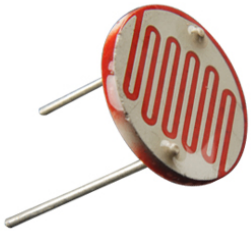



*Gambar 2.7 Liquid Crystal Display (LCD)*

LCD pada dasarnya terdiri dari 2 bagian utama, yaitu bagian *backlight* dan bagian *liquid crystal.* LCD hanya mentransmisikan cahaya yang melewatinya, sehingga membutuhkan cahaya *backlight* sebagai sumber cahayanya. Cahaya *backlight* umumnya berwarna putih yang akan memberikan pencahayaan pada *liquid crystal.* Kristal cair akan menyaring *backlight* yang diterima dan merefleksikannya sesuai dengan sudut yang diinginkan sehingga menghasilkan warna yang dibutuhkan.

## 2.8 Light Dependent Resistor (LDR)

Light Dependent Resistor (LDR) atau yang juga dikenal sebagai fotoresistor adalah jenis resistor yang nilai resistansinya tergantung pada intensitas cahaya yang mengenainya. Modul sensor cahaya bekerja menghasilkan output yang mendeteksi nilai intensitas cahaya.



*Gambar 2.8 Light Dependent Resistor (LDR)*

LDR mempunyai nilai resistansi yang terus berubah sesuai intensitas cahaya yang mengenai sensor. Semakin banyak cahaya yang mengenai sensor LDR, maka semakin turun nilai resistansinya, dan sebaliknya. Pada umumnya, sensor LDR mempunyai nilai resistansi sebesar 200 Ohm di tengah kegelapan dan akan turun menjadi 500 Ohm saat terkena banyak cahaya.

LDR memiliki peran sebagai sensor cahaya dalam rangkaian alat elektronik seperti saklar otomatis berdasarkan cahaya. Jadi, jika sensor terkena cahaya, maka arus listrik akan terhambat, namun jika berada dalam kondisi minim cahaya maka aliran listrik akan berjalan. LDR sering digunakan untuk lampu sensor kamar tidur, lampu penerangan jalan otomatis, dsb.

# BAB III
# METODE PENELITIAN

## 3.1 Tempat dan Waktu Penelitian

Penelitian ini akan dilaksanakan pada:

Tempat : SMA Katolik St. Louis 1 Surabaya

Waktu : Selasa, 3 Desember 2024 - Kamis, 30 Januari 2025

## 3.2 Alat dan Bahan Penelitian

### 3.2.1 Alat

Alat yang digunakan untuk pembuatan lampu sensor adalah:

1. Arduino uno R3
2. *Real Time Clock* (RTC) DS3231
3. *Light Dependent Resistor* (LDR)
4. *Liquid Crystal Display* (LCD) 2x16
5. Resistor
6. Watt meter
7. Modul relay
8. Modul dimmer AC
9. Beard board
10. Kabel jumper

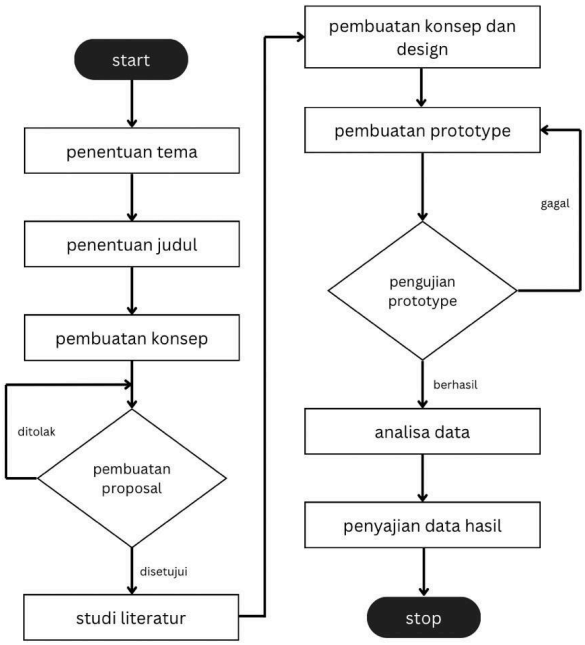
### 3.2.2 Bahan

Bahan yang digunakan untuk pembuatan lampu sensor adalah:

1. Lampu pijar
2. Baterai 9V
3. Akrilik
4. Duplex

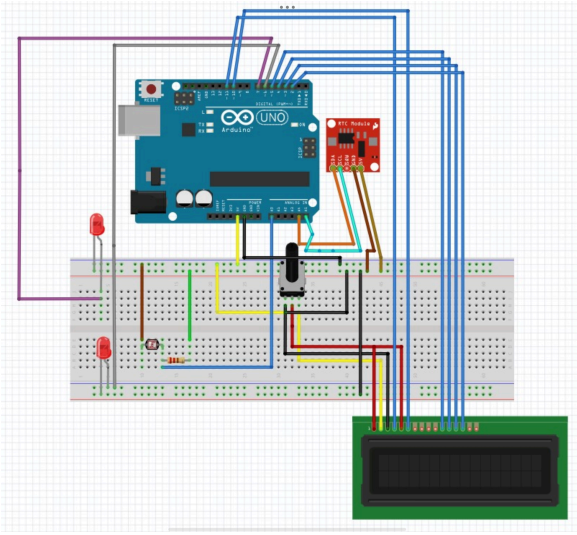
## 3.3 Tahapan Penelitian

### 3.3.1 Diagram Alur Penelitian




*Gambar 3.3.1 Diagram Alur Penelitian*

### 3.3.2 Skema Pembuatan Alat/Prototipe



*Gambar 3.3.2.1 Pembuatan Prototipe*

1. Persiapan Alat dan Bahan
2. Perencanaan Sistem
    a. Input: Sensor cahaya dan sensor waktu.
    b. Kontrol: *Microcontroller* untuk memproses input ke sensor.

c. Output: *Relay* untuk mengatur nyala/mati lampu

3. Merakit *Hardware*

a. Sensor Cahaya

i. Siapkan bahan bahan yang digunakan mulai dari lampu, steker, arduino, colokan, AC dimmer *light module*, kabel jumper dan LDR *module* (sensor cahaya ).

ii. Pasangkan sensor cahaya (*light dependent resistor*) ke *microcontroller* dengan kabel jumper *male to female* dimana VCC dihubungkan pada *port* 3,3V, *ground* dihubungkan pada *port ground*, *port A0* dihubungkan pada pin analog in A0 dan terakhir pin D0 dihubungkan pada pin digital PWM 4.

iii. Hubungkan *AC light dimmer module* dengan arduino, dimana pin PWM dihubungkan pada arduino pin digital PWM 33333 lalu pin ZC pada 66666 serta VCC pada pin 5V dan terakhir GND pada GND di arduino.

iv. Hubungkan AC dimmer *light module* pada steker dan lampu sebagai pembatas listrik.

b. Sensor Waktu

i. Siapkan bahan bahan yang digunakan dari lampu, steker, arduino, colokan, *relay module*, LCD, beardboard, RTC 3231, dan kabel jumper.

ii. Sambungkan RTC, LCD, dan Arduino pada breadboard dengan konfigurasi seri. Hubungkan pin SDA dan SCL dari LCD dan RTC secara seri, kemudian sambungkan ke beardboard. Selanjutnya, hubungkan pin VCC dan GND dari kedua modul tersebut secara seri pada beardboard. Pastikan pin VCC terhubung ke 5V dari Arduino, dan pin GND terhubung ke port GND pada beardboard untuk mendapatkan suplai daya yang stabil.

iii. Hubungkan antara *relay module* dan arduino pada pin in ke digital PWM 8 pada arduino.

iv. Hubungkan sumber listrik ke dalam rangkaian dengan menyambungkan dua kabel dari colokan listrik. Kabel pertama langsung dihubungkan ke lampu, sedangkan kabel kedua

disambungkan ke pin NO pada relay, lalu dihubungkan ke pin COM pada relay ke lampu.

4. Pemrograman *Microcontroller*
   a. Tujuan: Pengaturan sensor cahaya dan sensor waktu.
   b. Logika Pemrograman:
      i. Sensor Cahaya: Menyalakan lampu jika intensitas cahaya di bawah ambang batas.
      ii. Sensor Waktu: Untuk menyalakan dan mematikan lampu berdasarkan jadwal dari yang diprogram.
   c. *Coding* pemrograman:
      i. Sensor waktu

coding_sensor_waktu_1.ino

```
#include <Wire.h>
#include <RTClib.h>
#include <Adafruit_SSD1306.h>

// Inisialisasi objek RTC dan OLED
RTC_DS3231 rtc;
Adafruit_SSD1306 oled(128, 64, &Wire);

const int relay = 8;

void setup() {
  Serial.begin(115200);

  // Inisialisasi OLED
  if (!oled.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED tidak terdeteksi!");
    while (1); // Berhenti jika OLED tidak terdeteksi
  }
  oled.clearDisplay();
  oled.display();

  // Inisialisasi pin relay
  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH); // Relay mati di awal
```

coding_sensor_waktu_1.ino

```
82    // Log data ke Serial Monitor
83    Serial.print("Waktu: ");
84    Serial.print(jam);
85    Serial.print(":");
86    Serial.print(menit);
87    Serial.print(":");
88    Serial.println(detik);
89    Serial.print("Tanggal: ");
90    Serial.print(hari);
91    Serial.print("/");
92    Serial.print(bulan);
93    Serial.print("/");
94    Serial.println(tahun);
95
96    // Logika untuk mengontrol relay berdasarkan jam
97    if ((jam >= 12 && jam <= 24) || (jam >= 0 && jam <= 5)) {
98      digitalWrite(relay, LOW); // Nyalakan relay
99      Serial.println("Relay nyala");
100   } else {
101     digitalWrite(relay, HIGH); // Matikan relay
102     Serial.println("Relay mati");
103   }
104
105   delay(1000); // Tunggu 1 detik
106 }
```

coding_sensor_waktu_1.ino

```
75    oled.print("/");
76    oled.print(bulan);
77    oled.print("/");
78    oled.print(tahun);
79
80    oled.display();
81
82    // Log data ke Serial Monitor
83    Serial.print("Waktu: ");
84    Serial.print(jam);
85    Serial.print(":");
86    Serial.print(menit);
87    Serial.print(":");
88    Serial.println(detik);
89    Serial.print("Tanggal: ");
90    Serial.print(hari);
91    Serial.print("/");
92    Serial.print(bulan);
93    Serial.print("/");
94    Serial.println(tahun);
95
96    // Logika untuk mengontrol relay berdasarkan jam
97    if ((jam >= 12 && jam <= 24) || (jam >= 0 && jam <= 5)) {
98      digitalWrite(relay, LOW); // Nyalakan relay
99      Serial.println("Relay nyala");
100   } else {
```

coding_sensor_waktu_1.ino

```
// Mendapatkan tanggal dari RTC
int hari = now.day();  // Hari dalam bulan
int bulan = now.month();
int tahun = now.year();

// Menampilkan waktu di OLED
oled.drawRect(1, 5, 126, 55, WHITE);
oled.setTextSize(2);
oled.setTextColor(WHITE);
oled.setCursor(16, 20);
if (jam < 10) oled.print("0");
oled.print(jam);
oled.print(":");
if (menit < 10) oled.print("0");
oled.print(menit);
oled.print(":");
if (detik < 10) oled.print("0");
oled.print(detik);

// Menampilkan hari dan tanggal di OLED
oled.setTextSize(1);
oled.setCursor(16, 40);
oled.print(now.dayOfTheWeek()); // Menampilkan hari dalam bentuk string (Senin, Selasa, dll)
oled.setCursor(54, 40);
oled.print(hari);
oled.print("/");
```

coding_sensor_waktu_1.ino

```
// Periksa apakah RTC terhubung dengan baik
if (!rtc.begin()) {
  Serial.println("Tidak dapat menemukan RTC!");
  while (1); // Berhenti jika RTC tidak terdeteksi
}

// Memastikan waktu RTC sesuai dengan waktu saat ini
DateTime now = rtc.now();
if (now.year() < 2020) {  // Jika RTC belum diset dengan benar (misal tahun < 2020)
  Serial.println("RTC tidak terdeteksi dengan benar, mengatur waktu secara manual...");
  // Set waktu manual (contoh: 23 Desember 2024, 10:30:00)
  rtc.adjust(DateTime(2024, 12, 23, 10, 30, 0));
}
}

void loop() {
  oled.clearDisplay();

  // Mendapatkan waktu dari RTC
  DateTime now = rtc.now();
  int jam = now.hour();
  int menit = now.minute();
  int detik = now.second();

  // Mendapatkan tanggal dari RTC
  int hari = now.day();  // Hari dalam bulan
```

*Gambar 3.3.2.2 coding sensor waktu*

## ii. Setting OLED LCD

oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```
#include <DS3231.h>
#include <Adafruit_SSD1306.h>

DS3231 rtc(SDA, SCL);
Time waktu;
Adafruit_SSD1306 oled(128, 64, &Wire, 4);

int relay = 8;


void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  rtc.begin();

  oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  oled.clearDisplay();

  pinMode(relay, OUTPUT);
  digitalWrite(relay, HIGH);

  rtc.setTime(5, 59, 45);
```

Output

oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```
37
38   oled.setTextSize(1);
39   oled.setCursor(16, 40);
40   oled.print(rtc.getDOWStr()); //mengambil hari
41
42   oled.setCursor(54, 40);
43   oled.print(rtc.getDateStr()); //mengambil tanggal bulan tahun
44   oled.display();
45
46   waktu = rtc.getTime();
47
48   Serial.println(waktu.hour);
49   if (waktu.hour >= 17 && waktu.hour <= 24 || waktu.hour >= 0 && waktu.hour <= 10) {
50     digitalWrite(relay, LOW);
51     Serial.println("nyala");
52   } else {
53     digitalWrite(relay, HIGH);
54     Serial.println("mati");
55   }
56
57
58 }
59
```

*Gambar 3.3.2.3 coding setting OLED LCD*

iii. Scanning OLED LCD

i2c_scanner.ino

```
30 #include <Wire.h>
31
32
33 void setup()
34 {
35   Wire.begin();
36
37   Serial.begin(9600);
38   while (!Serial);             // Leonardo: wait for serial monitor
39   Serial.println("\nI2C Scanner");
40 }
41
42
43 void loop()
44 {
45   byte error, address;
46   int nDevices;
47
48   Serial.println("Scanning...");
49
50   nDevices = 0;
51   for(address = 1; address < 127; address++ )
52   {
```

i2c_scanner.ino

```
53     // The i2c_scanner uses the return value of
54     // the Write.endTransmisstion to see if
55     // a device did acknowledge to the address.
56     Wire.beginTransmission(address);
57     error = Wire.endTransmission();
58
59     if (error == 0)
60     {
61       Serial.print("I2C device found at address 0x");
62       if (address<16)
63         Serial.print("0");
64       Serial.print(address,HEX);
65       Serial.println("  !");
66
67       nDevices++;
68     }
69     else if (error==4)
70     {
71       Serial.print("Unknown error at address 0x");
72       if (address<16)
73         Serial.print("0");
74       Serial.println(address,HEX);
75     }
```

i2c_scanner.ino

```
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);           // wait 5 seconds for next scan
}
```

*Gambar 3.3.2.4 coding scanning OLED LCD*

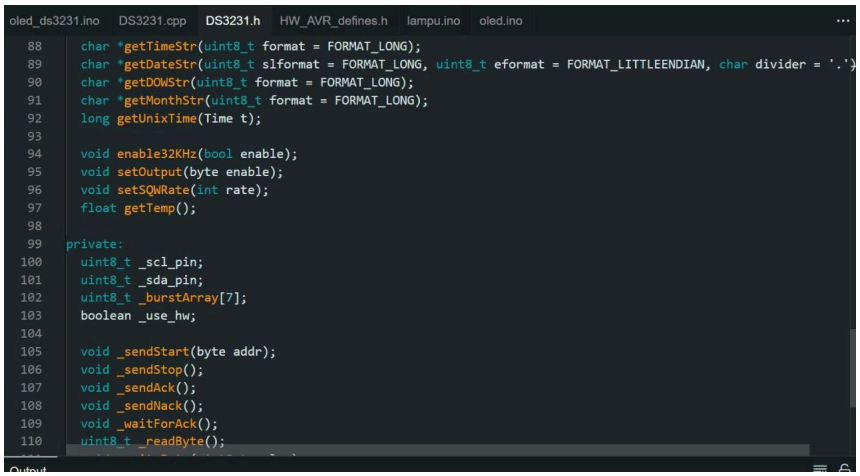
iv. Format LCD

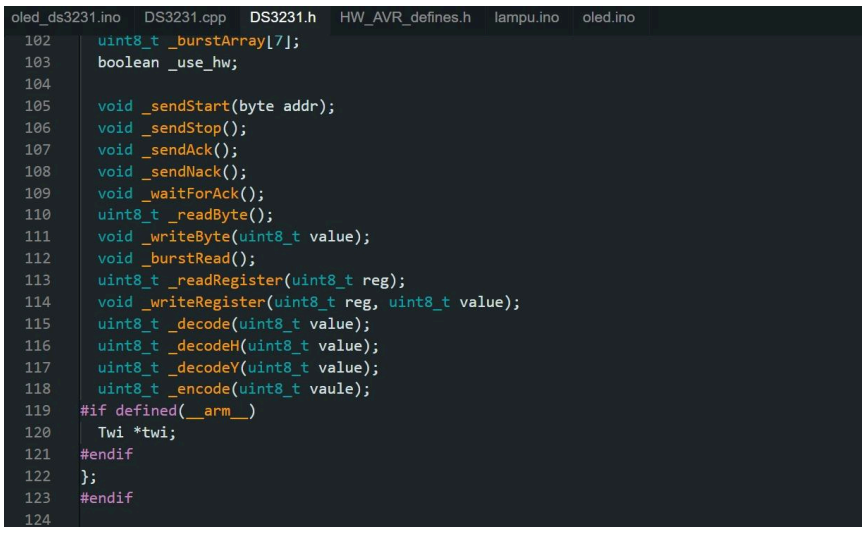
oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```
#ifndef DS3231_h
#define DS3231_h

#if defined(__AVR__)
#include "Arduino.h"
#include "Hardware/avr/HW_AVR_defines.h"
#elif defined(__PIC32MX__)
#include "WProgram.h"
#include "hardware/pic32/HW_PIC32_defines.h"
#elif defined(__arm__)
#include "Arduino.h"
#include "hardware/arm/HW_ARM_defines.h"
#endif

#define DS3231_ADDR_R 0xD1
#define DS3231_ADDR_W 0xD0
#define DS3231_ADDR 0x68

#define FORMAT_SHORT 1
#define FORMAT_LONG 2

#define FORMAT_LITTLEENDIAN 1
#define FORMAT_BIGENDIAN 2
#define FORMAT_MIDDLEENDIAN 3
```

oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```
#define FORMAT_LITTLEENDIAN 1
#define FORMAT_BIGENDIAN 2
#define FORMAT_MIDDLEENDIAN 3

#define SENIN 1
#define SELASA 2
#define RABU 3
#define KAMIS 4
#define JUMAT 5
#define SABTU 6
#define MINGGU 7

#define SQW_RATE_1 0
#define SQW_RATE_1K 1
#define SQW_RATE_4K 2
#define SQW_RATE_8K 3

#define OUTPUT_SQW 0
#define OUTPUT_INT 1

class Time
{
public:
```




*Gambar 3.3.2.5 coding format LCD*

v. Setting RTC

oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```ino
#include <DS3231.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_GFX.h> // Diperlukan oleh SSD1306

// Inisialisasi RTC DS3231
DS3231 rtc(SDA, SCL);

// Konfigurasi OLED
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 oled(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

void setup() {
  // Inisialisasi Serial Monitor
  Serial.begin(9600);

  // Inisialisasi RTC
  rtc.begin();

  // Atur waktu jika perlu
  // rtc.setDate(31, 12, 2020); // Format: Hari, Bulan, Tahun
  // rtc.setTime(23, 59, 45);  // Format: Jam, Menit, Detik
```

oled_ds3231.ino | DS3231.cpp | DS3231.h | HW_AVR_defines.h | lampu.ino | oled.ino

```ino
  // Inisialisasi OLED
  if (!oled.begin(SSD1306_I2C_ADDRESS, 0x3C)) { // Alamat I2C default: 0x3C
    Serial.println("Gagal menginisialisasi OLED");
    while (true);
  }
  oled.clearDisplay();
  oled.setTextSize(1);
  oled.setTextColor(SSD1306_WHITE);
}

void loop() {
  // Dapatkan waktu dan tanggal dari RTC
  String waktu = rtc.getTimeStr();  // Format: HH:MM:SS
  String tanggal = rtc.getDateStr(); // Format: DD/MM/YYYY

  // Tampilkan di OLED
  oled.clearDisplay();
  oled.setCursor(0, 0);
  oled.println("Waktu:");
  oled.println(waktu);
  oled.println("Tanggal:");
  oled.println(tanggal);
  oled.display();
```

*Gambar 3.3.2.6 coding setting RTC*

vi. Sensor cahaya

coding_sensor_cahaya_1.ino

```ino
#include <TimerOne.h>

volatile int i = 0;              // Variable to use as a counter volatile as it is in an interrupt
volatile boolean zero_cross = 0;  // Boolean to store a "switch" to tell us if we have crossed zero
int AC_pin = 6;                  // Output to Opto Triac (PWM AC connected to pin 6)
int LDR_pin = A0;                // AO from LDR connected to analog pin A0
int dim = 0;                     // Dimming level (0-128), start with off (0%)
int freqStep = 75;               // This is the delay-per-brightness step in microseconds.

void setup() {
  Serial.begin(9600);
  pinMode(AC_pin, OUTPUT);                          // Set the Triac pin as output
  attachInterrupt(digitalPinToInterrupt(3), zero_cross_detect, RISING); // Attach an Interrupt to Pin 3 (Zero Crossing Detection)
  Timer1.initialize(freqStep);                      // Initialize TimerOne library for the freq we need
  Timer1.attachInterrupt(dim_check, freqStep);
  Serial.println("Dimmer and LDR Monitoring Initialized");
}

void zero_cross_detect() {
  zero_cross = true;               // Set the boolean to true to tell our dimming function that a zero cross has occurred
  i = 0;
  digitalWrite(AC_pin, LOW);       // Turn off TRIAC (and AC)
}

// Turn on the TRIAC at the appropriate time
void dim_check() {
```

coding_sensor_cahaya_1.ino

```
  zero_cross = true;              // Set the boolean to true to tell our dimming function that a zero cross has occurred
  i = 0;
  digitalWrite(AC_pin, LOW);      // Turn off TRIAC (and AC)
}

// Turn on the TRIAC at the appropriate time
void dim_check() {
  if (zero_cross == true) {
    if (dim == 128) {
      digitalWrite(AC_pin, LOW); // Make sure TRIAC stays off if dim is 128
    } else if (i >= dim) {
      digitalWrite(AC_pin, HIGH);  // Turn on light
      i = 0;  // Reset time step counter
      zero_cross = false; // Reset zero cross detection
    }
    else {
      i++; // Increment time step counter
    }
  }
}
```

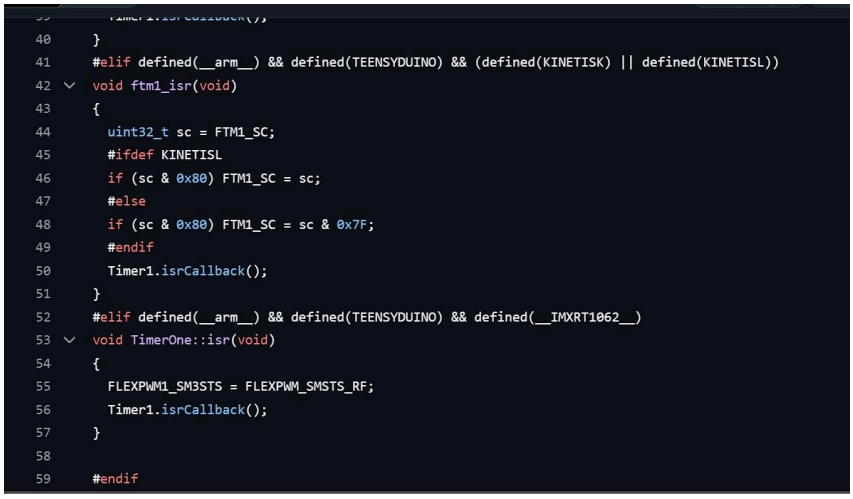
coding_sensor_cahaya_1.ino

```
void loop() {
  // Read LDR state
  int LDR_value = analogRead(LDR_pin);  // Read the analog value from LDR
  Serial.print("LDR Value: ");
  Serial.println(LDR_value);  // Print the LDR value

  // Adjust dim value based on LDR value
  if (LDR_value < 104) { // Gelap (terang rendah)
    Serial.println("Cahaya Terang");
    dim = 128;  // Lampu menyala penuh ketika cahaya gelap (100% brightness)
    Serial.println("Lampu: MATI (0%) ");
  }
  else if (LDR_value > 636) { // Terang (terang tinggi)
    Serial.println("Cahaya Gelap");
    dim = 0;  // Lampu dimmati ketika cahaya terang (off)
    Serial.println("Lampu: NYALA PENUH (100%)");
  }
  else { // Setengah terang (50% brightness)
    Serial.println("Cahaya Setengah Terang");
    dim = 64;  // Set dim to 50% brightness
    Serial.println("Lampu: NYALA 50%");
  }

  // Add a small delay to reduce Serial Monitor flooding
  delay(1000);
}
```

*Gambar 3.3.2.7 coding sensor cahaya*

vii. *Timerone* sensor cahaya

```
#include "TimerOne.h"

TimerOne Timer1;              // preinstantiate

#if !defined(ESP32)
unsigned short TimerOne::pwmPeriod = 0;
unsigned char TimerOne::clockSelectBits = 0;
void (*TimerOne::isrCallback)() = TimerOne::isrDefaultUnused;
void TimerOne::isrDefaultUnused() { /* noop */; }
#endif // not ESP32

// interrupt service routine that wraps a user defined function supplied by attachInterrupt
#if defined (__AVR_ATtiny85__)
ISR(TIMER1_COMPA_vect)
{
  Timer1.isrCallback();
}
#elif defined(__AVR__)
ISR(TIMER1_OVF_vect)
{
  Timer1.isrCallback();
```



```
}
#elif defined(__arm__) && defined(TEENSYDUINO) && (defined(KINETISK) || defined(KINETISL))
void ftm1_isr(void)
{
  uint32_t sc = FTM1_SC;
  #ifdef KINETISL
  if (sc & 0x80) FTM1_SC = sc;
  #else
  if (sc & 0x80) FTM1_SC = sc & 0x7F;
  #endif
  Timer1.isrCallback();
}
#elif defined(__arm__) && defined(TEENSYDUINO) && defined(__IMXRT1062__)
void TimerOne::isr(void)
{
  FLEXPWM1_SM3STS = FLEXPWM_SMSTS_RF;
  Timer1.isrCallback();
}

#endif
```

*Gambar 3.3.2.8 coding timerone  sensor cahaya*

5. Pengujian Program

   Menguji hasil pemrograman dengan hasil yang ditampilkan pada lampu.

6. Pembuatan Maket

   Membuat dan menguji hasil prototipe yang sudah dibuat ke dalam maket yang ada.

### 3.3.3 Pengambilan Data Penelitian

| SISTEM SENSOR CAHAYA | | | |
|---|---|---|---|
| Konsumsi Energi Per Jam (watt) | | | |
| Konsumsi per rentang waktu | | | |
| Jam | | | |
| Total | | | |

| SISTEM SENSOR WAKTU | |
|---|---|
| Konsumsi Energi per Jam (watt) | |
| Konsumsi per rentang waktu | |
| Jam | |

| | |
|---|---|
| Total | |

| LAMPU BIASA (INDOOR) | |
|---|---|
| Konsumsi Energi per Jam (watt) | |
| Konsumsi per rentang waktu | |
| Jam | |
| Total | |

| LAMPU BIASA (OUTDOOR) | |
|---|---|
| Konsumsi Energi per Jam (watt) | |
| Konsumsi per rentang waktu | |
| Jam | |
| Total | |

*Tabel 3.3.2 Pengambilan Data*

Rumus Standar Deviasi

$$\sigma = \sqrt{\frac{\Sigma(\chi i - \mu)^2}{N}}$$

### 3.3.4 Variabel Penelitian

Variabel penelitian adalah segala sesuatu yang berubah-ubah dan dapat diukur secara ilmiah. Terdapat tiga jenis variabel dalam penelitian ini, di antaranya sebagai berikut.

1. Variabel bebas : Pengaturan kontrol mati nyala dan tingkatan intensitas cahaya.
2. Variabel kontrol : Jenis lampu dimmable dan tegangan listrik.
3. Variabel terikat : Respon lampu serta tingkat kecerahan yang dihasilkan.

## 3.4 Metode dan Analisis Data

### 3.4.1 Metode Penelitian

Metode penelitian yang digunakan adalah sebagai berikut.

1. Studi pustaka

   Mengkaji literatur terkait teknologi sensor cahaya, pengatur waktu, dan efisiensi energi sebagai dasar pengembangan lampu hemat energi.

2. Pembuatan prototipe desain

   Merancang lampu hemat energi yang dilengkapi sensor cahaya untuk menyesuaikan pencahayaan dan sensor waktu untuk mengatur durasi operasi.

3. Pengumpulan data variabel terikat

   Mengumpulkan data kinerja lampu, seperti konsumsi daya, respons sensor, dan efisiensi pengaturan waktu.

4. Pengolahan data

   Menganalisis data untuk mengevaluasi hasil kerja lampu dalam menghemat energi.

5. Penyajian hasil analisis

   Menyajikan hasil analisis dalam penjelasan untuk menunjukkan keberhasilan desain lampu hemat energi.

### 3.4.2 Teknik Analisis Data

Metode analisis data yang diterapkan adalah metode secara kuantitatif untuk mengukur kinerja sistem aplikasi sensor lampu. Analisis akurasi sistem dilakukan dengan membandingkan output yang dihasilkan oleh program dengan nilai intensitas cahaya yang sebenarnya. Kecepatan respon sensor akan dianalisis berdasarkan waktu yang dibutuhkan sistem untuk mendeteksi perubahan intensitas cahaya dan mengaktifkan lampu. Stabilitas program akan diuji melalui simulasi berulang untuk mengamati frekuensi dan jenis kesalahan yang muncul selama pengoperasian.

# BAB IV
# HASIL PENELITIAN DAN PEMBAHASAN

## 4.1 Hasil Penelitian

### 4.1.1 Perbandingan Konsumsi Energi pada Lampu Biasa, Lampu Sensor Cahaya, dan Lampu Sensor Waktu

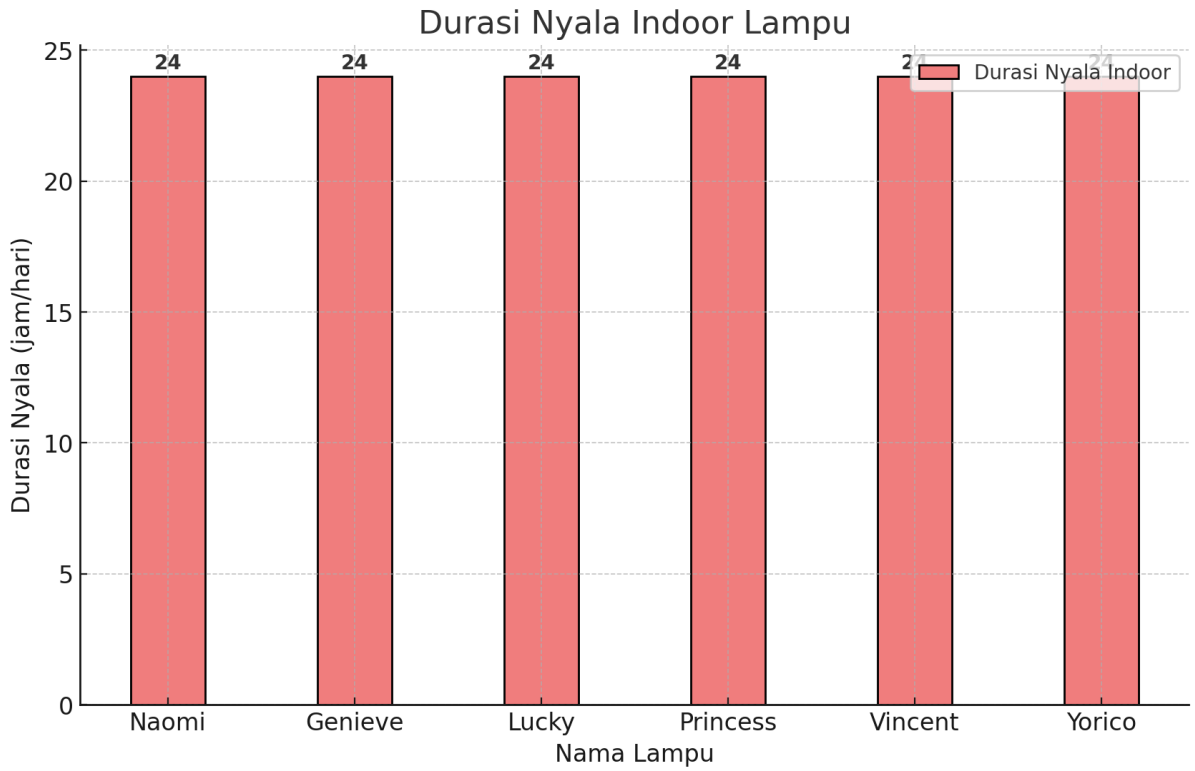
Berdasarkan data yang diperoleh dari penggunaan lampu anggota kelompok, dapat dilakukan perbandingan konsumsi energi yang digunakan oleh lampu konvensional, lampu pijar dengan sistem sensor cahaya, serta lampu pijar dengan sistem sensor waktu. Ketiga percobaan tersebut menggunakan lampu dengan daya 75 watt, yang memungkinkan analisis lebih lanjut terhadap efisiensi energi yang dihasilkan oleh masing-masing sistem.

| SISTEM SENSOR CAHAYA | | | |
|---|---|---|---|
| Konsumsi Energi Per Jam (watt) | 50 | 75 | 50 |
| Konsumsi per rentang waktu | 175 | 662.5 | 158 |
| Jam | 16.40 - 20.10 | 20.10 - 05.00 | 05.00 - 08.10 |
| Total | 996 watt | | |

*Tabel 4.1.1.1 Konsumsi Energi Lampu dengan Sistem Sensor Cahaya*

| SISTEM SENSOR WAKTU | |
|---|---|
| Konsumsi Energi per Jam (watt) | 75 |
| Konsumsi per rentang waktu | |
| Jam | 17.00 - 05.00 ( 12 jam ) |
| Total | 900 watt |

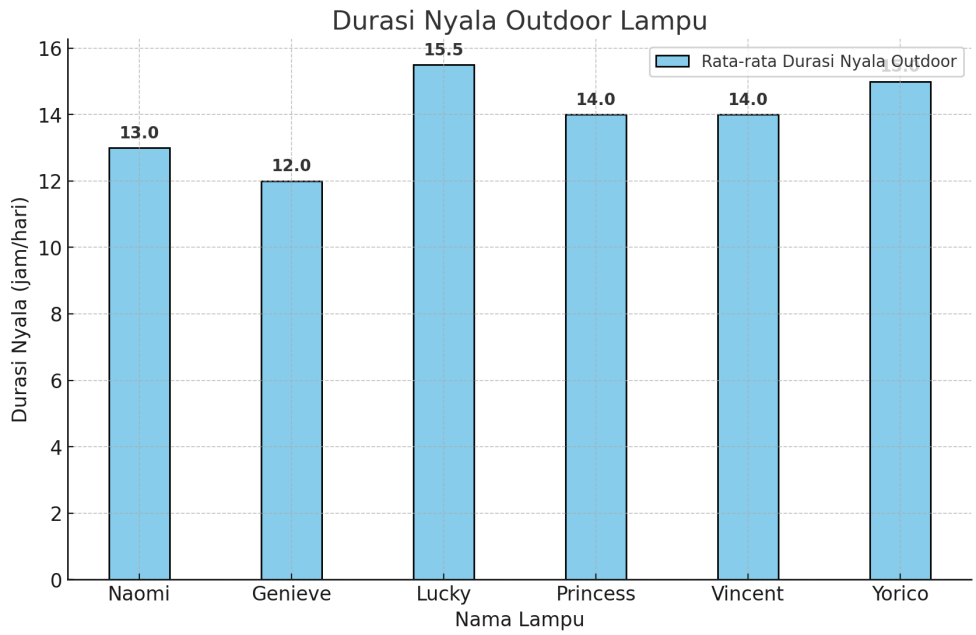
*Tabel 4.1.1.2 Konsumsi Energi Lampu dengan Sistem Sensor Waktu*




*Gambar 4.1.1.1 Grafik Penggunaan Lampu Biasa Indoor*

| LAMPU BIASA (INDOOR) | |
|---|---|
| Konsumsi Energi per Jam (watt) | 75 |
| Konsumsi per rentang waktu | - |
| Jam | 24 jam |
| Total | 1800 watt |

*Tabel 4.1.1.3 Konsumsi Energi Lampu Biasa Indoor*




*Gambar 4.1.1.2 Grafik Penggunaan Lampu Biasa Outdoor*

| LAMPU BIASA (OUTDOOR) | |
|---|---|
| Konsumsi Energi per Jam (watt) | 75 |
| Konsumsi per rentang waktu | - |
| Jam | 13.92 jam |
| Total | 1045 watt |

*Tabel 4.1.1.4 Konsumsi Energi Lampu Biasa Outdoor*

## 4.2 Pembahasan

### 4.2.1 Cara Kerja Dua Sistem Lampu Sensor Hemat Energi

Prototipe ini dirancang dengan memanfaatkan dua jenis lampu yang mengimplementasikan dua pendekatan berbeda dalam hal efisiensi energi. Kedua pendekatan tersebut menggunakan teknologi yang berbeda, yaitu sensor cahaya (LDR) dan sensor waktu.

1. Lampu dengan Sensor LDR

   Prinsip dasar dari sistem ini adalah mendeteksi intensitas cahaya lingkungan sekitar dan secara otomatis menyesuaikan tingkat pencahayaan lampu sesuai dengan kondisi tersebut. Ketika tingkat pencahayaan mencapai

ambang batas tertentu, sensor LDR akan mengirimkan sinyal ke mikrokontroler untuk mengaktifkan lampu pijar dengan intensitas yang telah disesuaikan. Penyesuaian kecerahan dilakukan melalui penggunaan modul dimmer yang memungkinkan pengaturan tingkat kecerahan sesuai kebutuhan.

Dalam proses perancangan sistem ini, digunakan pendekatan distribusi normal untuk menentukan ambang batas tingkat kecerahan yang tepat, yang ditentukan berdasarkan pengukuran data intensitas cahaya LDR yang diperoleh selama periode waktu tertentu.

*Gambar 4.2.1.1 Grafik Nilai LDR dalam 24 Jam*

Pemanfaatan lampu dengan sensor cahaya yang telah diuji selama periode 24 jam menghasilkan data yang menunjukkan perbedaan yang signifikan dalam penggunaan energi per jam. Data pengujian ini memberikan gambaran yang jelas mengenai variasi konsumsi energi sepanjang waktu, yang dipaparkan secara rinci secara berikut.

```
22 00 -> LDR Value: 675
22 10 -> LDR Value: 673
22 20 -> LDR Value: 674
22 30 -> LDR Value: 672
22 40 -> LDR Value: 671
22 50 -> LDR Value: 673

total = 4038

23 00 -> LDR Value: 672
23 10 -> LDR Value: 671
23 22 -> LDR Value: 672
23 30 -> LDR Value: 674
23 40 -> LDR Value: 673
23 50 -> LDR Value: 675

total = 4037

00 00 -> LDR Value: 673
00 10 -> LDR Value: 672
00 20 -> LDR Value: 675
00 30 -> LDR Value: 673
00 40 -> LDR Value: 671
00 50 -> LDR Value: 672

total = 4036

01 00 -> LDR Value: 671
01 10 -> LDR Value: 673
01 20 -> LDR Value: 672
01 30 -> LDR Value: 673
01 40 -> LDR Value: 674
01 50 -> LDR Value: 671

total = 4034

02 00 -> LDR Value: 671
02 10 -> LDR Value: 672
02 20 -> LDR Value: 671
02 30 -> LDR Value: 673
02 40 -> LDR Value: 672
02 50 -> LDR Value: 671

total = 4030
```

```
03 00 -> LDR Value: 671
03 10 -> LDR Value: 672
03 20 -> LDR Value: 673
03 30 -> LDR Value: 671
03 40 -> LDR Value: 673
03 50 -> LDR Value: 672

total = 4031

04 00 -> LDR Value: 670
04 10 -> LDR Value: 668
04 20 -> LDR Value: 665
04 30 -> LDR Value: 660
04 40 -> LDR Value: 650
04 50 -> LDR Value: 645

total = 3958

05 00 -> LDR Value: 632
05 10 -> LDR Value: 610
05 20 -> LDR Value: 596
05 30 -> LDR Value: 570
05 40 -> LDR Value: 553
05 50 -> LDR Value: 521

total = 3482

06 00 -> LDR Value: 493
06 10 -> LDR Value: 460
06 20 -> LDR Value: 434
06 30 -> LDR Value: 404
06 40 -> LDR Value: 371
06 50 -> LDR Value: 325

total = 2487


07 00 -> LDR Value: 283
07 10 -> LDR Value: 240
07 20 -> LDR Value: 223
07 30 -> LDR Value: 203
07 40 -> LDR Value: 182
07 50 -> LDR Value: 146

total = 1277
```

```
08 00 -> LDR Value: 120
08 10 -> LDR Value: 99
08 20 -> LDR Value: 89
08 30 -> LDR Value: 87
08 40 -> LDR Value: 85
08 50 -> LDR Value: 83

total = 563

09 00 -> LDR Value: 82
09 10 -> LDR Value: 81
09 20 -> LDR Value: 80
09 30 -> LDR Value: 80
09 40 -> LDR Value: 79
09 50 -> LDR Value: 79

total = 481

10 00 -> LDR Value: 78
10 10 -> LDR Value: 78
10 20 -> LDR Value: 77
10 30 -> LDR Value: 77
10 40 -> LDR Value: 76
10 50 -> LDR Value: 76

total = 462

11 00 -> LDR Value: 75
11 10 -> LDR Value: 75
11 20 -> LDR Value: 75
11 30 -> LDR Value: 74
11 40 -> LDR Value: 75
11 50 -> LDR Value: 74

total = 448


12 00 -> LDR Value: 73
12 10 -> LDR Value: 73
12 20 -> LDR Value: 73
12 30 -> LDR Value: 72
12 40 -> LDR Value: 73
12 50 -> LDR Value: 73

total = 437
```

```
13 00 -> LDR Value: 72
13 10 -> LDR Value: 73
13 20 -> LDR Value: 73
13 30 -> LDR Value: 73
13 40 -> LDR Value: 73
13 50 -> LDR Value: 72

total = 436

14 00 -> LDR Value: 74
14 10 -> LDR Value: 74
14 20 -> LDR Value: 75
14 30 -> LDR Value: 75
14 40 -> LDR Value: 75
14 50 -> LDR Value: 74

total = 447

15 00 -> LDR Value: 76
15 10 -> LDR Value: 78
15 20 -> LDR Value: 80
15 30 -> LDR Value: 78
15 40 -> LDR Value: 80
15 50 -> LDR Value: 82

total = 474

16 00 -> LDR Value: 85
16 10 -> LDR Value: 89
16 20 -> LDR Value: 95
16 30 -> LDR Value: 103
16 40 -> LDR Value: 112
16 50 -> LDR Value: 119

total = 603

17 00 -> LDR Value: 123
17 10 -> LDR Value: 142
17 20 -> LDR Value: 162
17 30 -> LDR Value: 186
17 40 -> LDR Value: 201
17 50 -> LDR Value: 226

total = 1040
```

```
18 00 -> LDR Value: 283
18 10 -> LDR Value: 317
18 20 -> LDR Value: 362
18 30 -> LDR Value: 387
18 40 -> LDR Value: 419
18 50 -> LDR Value: 452

total = 2220

19 00 -> LDR Value: 482
19 10 -> LDR Value: 523
19 20 -> LDR Value: 551
19 30 -> LDR Value: 583
19 40 -> LDR Value: 597
19 50 -> LDR Value: 608

total = 3344

20 00 -> LDR Value: 620
20 10 -> LDR Value: 635
20 20 -> LDR Value: 648
20 30 -> LDR Value: 659
20 40 -> LDR Value: 663
20 50 -> LDR Value: 667

total = 3892

21 00 -> LDR Value: 670
21 10 -> LDR Value: 670
21 20 -> LDR Value: 671
21 30 -> LDR Value: 672
21 40 -> LDR Value: 673
21 50 -> LDR Value: 673

total = 4029
```

*Gambar 4.2.1.2 Nilai LDR Per 10 Menit dalam 24 Jam*

Dari hasil data tersebut, dapat dihitung nilai tengah dan nilai ambang yang dibutuhkan untuk keperluan pembuatan program (mean, standar deviasi), yakni:

$$\bar{X} = (4038 + 4037 + .... + 4029) \ : 144 = 369.78 \approx 370$$

$$\text{Standar Deviasi } (\sigma) \ = \sqrt{\frac{\Sigma(\chi i - \mu)^2}{N}} =$$

$$\sqrt{\frac{\Sigma(303)^2+(303)^2+(302)^2+(302)^2+(301)^2+(301)^2+(290)^2+(211)^2+(45)^2+(158)^2+(276)^2+(290)^2+(293)^2+(295)^2+(297)^2+(298)^2+(296)^2+(291)^2+(270)^2+(196)^2+(0)^2(187)^2+(278)^2+(301)^2}{24}}$$

$$= \sqrt{\frac{70811}{24}} = 266$$

$\bar{X} + \sigma = 370 + 266 = 636$

$\bar{X} - \sigma = 370 - 266 = 104$

Pengaturan ini selanjutnya akan dimasukkan ke dalam pengodingan sebagai ambang batas maksimal maupun minimal seperti pada berikut ini.

```
void loop() {
  // Read LDR state
  int LDR_value = analogRead(LDR_pin);  // Read the analog value from LDR
  Serial.print("LDR Value: ");
  Serial.println(LDR_value);  // Print the LDR value

  // Adjust dim value based on LDR value
  if (LDR_value < 104 ) { // Gelap (terang rendah)
    Serial.println("Cahaya Terang");
    dim = 128;  // Lampu menyala penuh ketika cahaya gelap (100% brightness)
    Serial.println("Lampu: MATI (0%) ");
  }
  else if (LDR_value > 636 ) { // Terang (terang tinggi)
    Serial.println("Cahaya Gelap");
    dim = 0;  // Lampu dimmati ketika cahaya terang (off)
    Serial.println("Lampu: NYALA PENUH (100%)");
  }
  else { // Setengah terang (50% brightness)
    Serial.println("Cahaya Setengah Terang");
    dim = 64;  // Set dim to 50% brightness
    Serial.println("Lampu: NYALA 50%");
  }
```

*Gambar 4.2.1.3 Coding Pemrograman Lampu Sensor LDR*

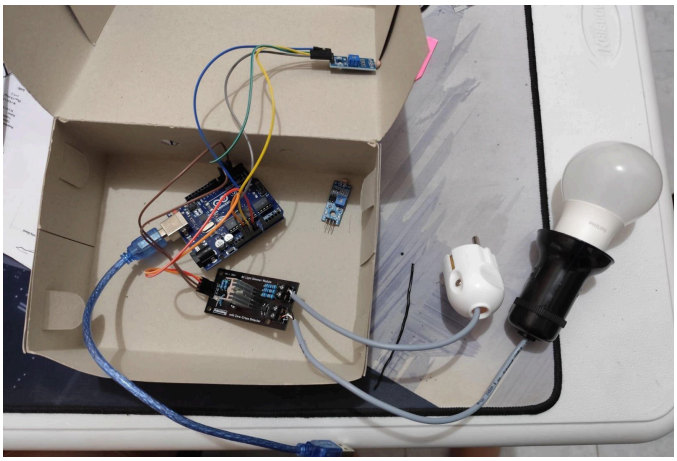
2. Lampu dengan Sensor Waktu

Prinsip utama dari sistem kerja lampu ini adalah pengaturan waktu yang dilakukan secara sistematis berdasarkan rentang waktu tertentu yang telah diprogram dalam sistem Real-Time Clock (RTC). Ketika waktu yang telah ditentukan tercapai, mikrokontroler akan secara otomatis mengaktifkan lampu sesuai dengan pengaturan yang telah dilakukan. Kode program yang dihasilkan untuk implementasi sistem ini disajikan sebagai berikut.

```
waktu = rtc.getTime();

Serial.println(waktu.hour);
if (waktu.hour >= 17 && waktu.hour <= 24 || waktu.hour >= 0 && waktu.hour <= 5) {
  digitalWrite(relay, LOW);
  Serial.println("nyala");
} else {
  digitalWrite(relay, HIGH);
  Serial.println("mati");
}
```

*Gambar 4.2.1.4 Coding Pemrograman Lampu Sensor Waktu*

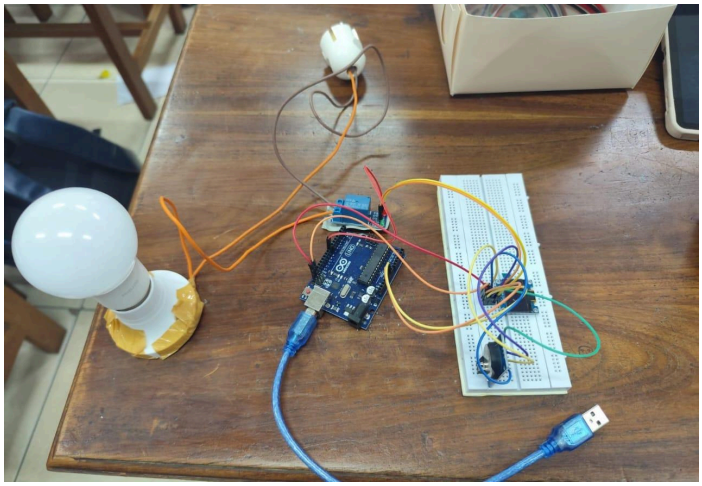
### 4.2.2 Desain Rangkaian Listrik yang Mendukung Kerja Sensor dan Lampu



*Gambar 4.2.2.1 Desain Rangkaian Listrik Sensor Cahaya*

Pada rangkaian listrik dengan sensor cahaya ini, sumber energi listrik input disambungkan langsung ke modul AC Triac Light Dimmer yang berfungsi untuk mengatur besaran arus listrik yang akan dialirkan ke lampu. Pengaturan besaran arus tersebut dilakukan berdasarkan intensitas cahaya yang terdeteksi oleh sensor cahaya yang telah terhubung dengan board Arduino.

Data hasil pembacaan sensor cahaya digunakan sebagai acuan atau ambang batas pengaturan intensitas pencahayaan. Sinyal dari Arduino kemudian dikirimkan ke modul Triac Light Dimmer untuk menyesuaikan besaran arus listrik yang dialirkan ke lampu. Melalui proses ini, pengujian konsumsi daya listrik (watt) pada lampu dapat dilakukan secara efektif untuk mengevaluasi efisiensi energi yang dicapai oleh sistem berbasis sensor cahaya ini.



*Gambar 4.2.2.2 Desain Rangkaian Listrik Sensor Waktu*

Pada rangkaian listrik dengan sensor waktu, terdapat dua sambungan sumber energi listrik input yang langsung dihubungkan ke relay module dan lampu. Rangkaian ini dirancang untuk mengatur penyambungan dan pemutusan aliran listrik yang diperlukan oleh lampu. Pengaturan waktu operasional dilakukan melalui pemrograman yang diintegrasikan dengan board Arduino, yang menjadi pusat kendali sistem.

Pada board Arduino, dipasang dan dihubungkan RTC (Real-Time Clock) module serta panel LCD. RTC module digunakan untuk memastikan Arduino dapat membaca dan mengikuti waktu yang berjalan secara real-time, sedangkan panel LCD berfungsi untuk menampilkan data waktu serta memverifikasi kesesuaian waktu yang berjalan dengan waktu aktual.

Berdasarkan data global, Indonesia yang terletak di wilayah khatulistiwa secara umum memerlukan pencahayaan selama sekitar 12 jam, yaitu dari pukul 17.00 hingga pukul 05.00. Hal ini dijadikan acuan dalam pengaturan waktu penyalaan dan pemadaman lampu secara otomatis. Sinyal yang dihasilkan oleh Arduino dikirimkan ke relay module untuk mengatur kapan aliran listrik dialirkan atau diputuskan ke lampu.

Melalui konfigurasi ini, dilakukan pengujian konsumsi daya listrik (watt) pada lampu untuk mengevaluasi efisiensi energi yang dihasilkan dari penerapan sistem berbasis sensor waktu.

# BAB V
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan lampu pijar yang dilengkapi dengan sensor cahaya dan sensor waktu terbukti lebih hemat energi dibandingkan dengan lampu pijar konvensional tanpa sensor.

Lampu pijar dengan sensor cahaya menunjukkan kinerja yang lebih efisien untuk penggunaan di luar ruangan. Hal ini disebabkan oleh kemampuannya dalam menyesuaikan tingkat pencahayaan berdasarkan intensitas cahaya lingkungan. Berdasarkan data yang diperoleh, lampu pijar dengan sensor cahaya hanya memerlukan konsumsi energi sebesar 996 watt untuk durasi penggunaan sekitar 14 jam per hari. Sebagai perbandingan, lampu pijar biasa yang digunakan di kondisi luar ruangan tanpa pengaturan otomatis membutuhkan energi sebesar 1.045 watt untuk durasi waktu yang sama. Efisiensi ini dapat terjadi karena lampu dengan sensor cahaya akan meredup secara otomatis ketika kondisi lingkungan sekitar terang, sehingga konsumsi energinya berkurang.

Sementara itu, lampu pijar dengan sensor waktu lebih sesuai digunakan di dalam ruangan. Sensor waktu memungkinkan pengguna untuk mengatur jadwal nyala lampu secara presisi sesuai kebutuhan, tanpa bergantung pada kondisi pencahayaan eksternal. Dari hasil penelitian, lampu pijar dengan sensor waktu yang dinyalakan selama 12 jam per hari hanya memerlukan konsumsi energi sebesar 900 watt. Sebaliknya, lampu pijar konvensional yang dinyalakan terus menerus selama 24 jam per hari memerlukan konsumsi energi yang jauh lebih besar, yaitu sebesar 1.800 watt.

Dengan demikian, pemanfaatan teknologi sensor, baik sensor cahaya maupun sensor waktu, dapat menjadi solusi yang efektif untuk meningkatkan efisiensi energi dalam penggunaan lampu pijar. Pemilihan jenis sensor hendaknya disesuaikan dengan kebutuhan serta lokasi penggunaan untuk mencapai hasil yang optimal.

## 5.2 Saran

Setelah melakukan berbagai rangkaian uji coba, kelompok kami dapat memberikan saran sebagai berikut:

1. Menerapkan penggunaan lampu sensor di berbagai lingkungan seperti di perkantoran, sekolah, dan berbagai fasilitas umum untuk mengoptimalkan penghematan energi dengan lebih luas.
2. Memilih sensor sesuai dengan kebutuhan seperti sensor gerak untuk area dengan aktivitas tinggi dan sensor cahaya untuk menyesuaikan intensitas nyala lampu, atau penggabungan kedua sensor tersebut untuk meningkatkan efisiensi sistem pencahayaan otomatis.
3. Melakukan penelitian lebih lanjut untuk menguji daya tahan sensor dalam jangka panjang selama minimal 1 bulan dalam berbagai kondisi lingkungan.

# DAFTAR PUSTAKA


Arduino Biz. (2022). *Penjelasan Tentang ESP32-CAM*. Diakses pada 10 November 2024, dari https://www.arduino.biz.id/2022/08/penjelasan-tentang-esp32-cam.html

Dicoding. *Apa Itu Arduino?*. Diakses pada 7 November 2024, dari https://www.dicoding.com/blog/apa-itu-arduino/

Dinas Perencanaan Sistem PT PLN (Persero). *Proyeksi Kebutuhan Listrik per Sektor per- Provinsi PLN dari Tahun 2003 s.d. Tahun 2013*. Jakarta 2004.

Kawan Lama. *Apa Itu Lux Meter?*. Diakses pada 7 November 2024, dari https://www.kawanlama.com/blog/ulasan/apa-itu-lux-meter

Lampbond. *Perbedaan LED Dimmable dan Non-Dimmable*. Diakses pada 27 November 2024, dari https://lampbond.co.id/blogs/news/perbedaan-led-dimmable-dan-non-dimmable#:~:text=Lampu%20LED%20dimmable%20memberikan%20fleksibilitas,tingkat%20kecerahan%20yang%20tetap%20diperlukan.

Liputan6. *Apa Itu Dimmer? Panduan Lengkap Memahami Teknologi Pengatur Cahaya*. Diakses pada 10 November 2024, dari https://www.liputan6.com/feeds/read/5801592/apa-itu-dimmer-panduan-lengkap-memahami-teknologi-pengatur-cahaya?page=2

Nyebar Ilmu. *Tutorial Arduino: Mengakses Module RTC DS3231*. Diakses pada 7 November 2024, dari https://www.nyebarilmu.com/tutorial-arduino-mengakses-module-rtc-ds3231/

Sagita, I. *Pengertian Modul Relay serta Perbedaannya dengan Relay*. Anak Teknik. Diakses pada 27 November 2024, dari https://www.anakteknik.co.id/ish_sagita/articles/pengertian-modul-relay-serta-perbedaannya-dengan-relay

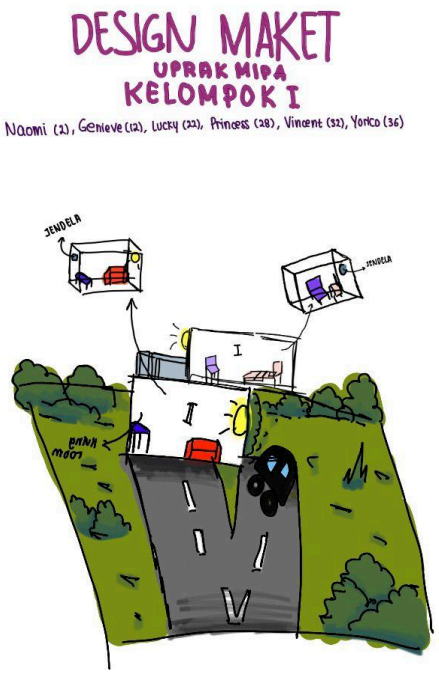
Supatmi, Sri. (2011). *Pengaruh Sensor LDR terhadap Pengontrolan Lampu*. Jurusan Teknik Komputer Universitas Indonesia.

Sutono. (2011). *Perancangan Sistem Aplikasi Otomatisasi Lampu Penerangan Menggunakan Sensor Gerak Dan Sensor Cahaya Berbasis Arduino Uno* (Atmega 328). Fakultas Program Studi Teknik Komputer Fakultas Teknik dan Ilmu Komputer Universitas Komputer Indonesia.

Teknik Elektronika. *Pengertian LCD (Liquid Crystal Display) dan Prinsip Kerja LCD.* Diakses pada 1 Desember 2024 dari https://teknikelektronika.com/pengertian-lcd-liquid-crystal-display-prinsip-kerja-lcd/

Welman. (2013). *Prototype Penerangan Rumah Otomatis Berbasis Mikrokontroler Atmega853*. Jurusan Tehnik Elektro UIN SUSKA Pekanbaru. 66 Indonesian Physical Review. 2(2): 57 - 67

# LAMPIRAN




*Gambar 6.1 Design Maket*

*Gambar 6.2 Pembelian Bahan-Bahan Pembuatan Maket*

*Gambar 6.3 Proses Pembuatan Maket*

*Gambar 6.4 Hasil Jadi Maket*