

## **BAB II**

### **LANDASAN TEORI**

#### **2.1.Studi Literatur**

Studi literatur merupakan proses analisis teori-teori yang mendukung perancangan, pengujian, perakitan, analisis, dan pembahasan dalam penelitian. Sumber studi literatur dapat berasal dari berbagai referensi, seperti jurnal ilmiah, literatur, dan buku yang mengandung informasi yang relevan dengan penelitian mengenai perancangan sistem kontrol penerangan otomatis menggunakan mikrokontroler *ESP32* berbasis *internet of things*. Studi literatur ini mencakup beberapa pembahasan penting, antara lain:

1. Teori pengendalian sistem penerangan.
2. Teori *switching relay* untuk menciptakan kerja alat dengan kendali otomatis dalam proses pemutusan maupun penyambungan arus pada lampu.
3. Teori *internet of things* dalam proses di luar jangkauan area lokal.
4. Spesifikasi komponen maupun alat pembentuk sistem seperti mikrokontroler, nodeMCU, lampu, relay, sensor dan komponen-komponen lainnya.

Dalam penelitian ini, dilakukan pengkajian teoritis terkait dengan komponen-komponen yang membentuk sistem. Melalui pengkajian ini, tujuannya adalah untuk memahami spesifikasi dan cara kerja setiap komponen yang digunakan. Sumber-sumber yang digunakan dalam tahap studi literatur, termasuk jurnal, buku, dan literatur terkait, secara khusus terkait dengan topik penelitian yang dilakukan dalam tugas akhir ini. Studi literatur ini fokus pada pemahaman konsep-

konsep yang terkait dengan Internet of Things (IoT), Relay, Mikrokontroler ESP32, Firebase, dan penerangan otomatis.

Pada tahap studi literatur, dilakukan pengumpulan materi secara teoritis berdasarkan poin-poin yang telah dijelaskan sebelumnya. Materi ini menjadi bahan pengkajian dan dasar teori untuk penelitian ini. Pada bab ini, akan dilakukan pembahasan tentang alur kerja alat dan menjelaskannya secara teoritis, sesuai dengan implementasi nyata dari alat yang dibangun.

## **2.2.Penerangan Otomatis**

Menurut penelitian yang dilakukan oleh Mappalotteng & Syahrul (2015), sumber pencahayaan dapat diklasifikasikan menjadi dua jenis, yaitu pencahayaan alami yang diperoleh dari sinar matahari dan pencahayaan buatan yang menggunakan sumber energi seperti listrik, gas, dan minyak. Pada gedung, penerangan pada siang hari mengandalkan sinar matahari sebagai sumber utama cahaya. Namun, ada situasi tertentu ketika sinar matahari terhalang oleh awan atau pada malam hari, sehingga manusia perlu menggunakan penerangan buatan sebagai pengganti cahaya matahari.



Gambar 2. 1 Skalar Lampu

Gambar 2.1 menunjukkan skalar lampu pada bangunan. Fungsi saklar yaitu menjadi pemutus dan penghubung aliran listrik yang menuju beban.

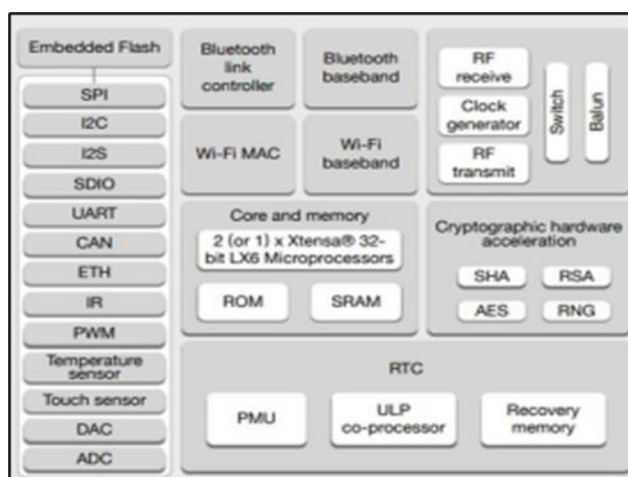
NO	KODE	WAKTU OPERASIONAL
1	Green	17.00 - 21.00 WIB / 18.00 - 21.00 WIB
2	Pink	17.00 - 06.00 WIB
3	Yellow	Selalu <b>ON</b> dan <b>OFF</b> bila penerangan area sekitar cukup terang
4	Red	<b>ON</b> bila diperlukan <b>OFF</b> bila tidak diperlukan

Gambar 2. 2 Waktu Operasional Penggunaan Lampu

Gambar 2.2 menunjukkan pengoperasian penggunaan lampu pada bangunan. Dalam melakukan pengelolaan gedung, pengoperasian ini digunakan untuk menyalakan dan mematikan lampu sesuai kebutuhan.

### 2.3. Mikrokontroler ESP32

Mikrokontroler ESP32 adalah sebuah sistem terpadu (*System on Chip*) yang dilengkapi dengan fitur *WiFi* 802.11 b/g/n, *Bluetooth* versi 4.2, dan berbagai periferal lainnya. *ESP32* memiliki keunggulan sebagai chip yang sangat komprehensif, karena dilengkapi dengan prosesor, penyimpanan, dan akses ke pin *GPIO* (*General Purpose Input Output*). *ESP32* dapat digunakan sebagai alternatif pengganti mikrokontroler *Arduino*, dan memiliki kemampuan untuk terhubung secara langsung ke jaringan *WiFi*. (Wagyana, 2019).



Gambar 2. 3 Arsitektur *ESP32* (Prafanto et al., 2021)

*ESP32* merupakan sebuah mikrokontroler yang dirancang dan dikembangkan oleh *Espressif Systems*. Mikrokontroler ini memiliki fitur yang mencakup kemampuan *WiFi* dan *Bluetooth*. Dengan adanya kemampuan *WiFi* dan *Bluetooth* ini, *ESP32* memungkinkan penggunaannya dalam mengontrol perangkat elektronik melalui jaringan nirkabel. Hal ini menjadikan *ESP32* sebagai salah satu pilihan yang sangat cocok untuk mengembangkan proyek-proyek *Internet of Things (IoT)* (Baharudin et al., 2022)

Dalam pengembangan aplikasi menggunakan *ESP32*, terdapat beberapa bahasa pemrograman yang dapat digunakan, antara lain C, C++, Python, dan JavaScript. Terdapat juga berbagai pustaka (library) dan kerangka kerja (*framework*) yang tersedia untuk mempermudah pengembangan aplikasi dengan *ESP32*, seperti Arduino IDE, ESP-IDF (*ESP32 IoT Development Framework*), dan *Micropython*. Dengan menggunakan pilihan bahasa pemrograman dan alat-alat pengembangan yang sesuai, pengembang dapat dengan mudah mengembangkan aplikasi yang berjalan pada platform *ESP32*.

Specifications	
Operating Voltage	2.2V to 3.6V
GPIO	36 ports
ADC	14 ports
DAC	2 ports
Flash memory	16 Mbyte
SRAM	250 Kbyte
Clock Speed	Up to 240 MHz
Wi-Fi	2.4 GHz
Sleep Current	2.5 $\mu$ A

Gambar 2. 4 Spesifikasi *ESP32* (Abed & Abed, 2019)

Ada dua versi board *ESP32* yang tersedia, yaitu versi dengan 30 pin GPIO dan versi dengan 36 pin GPIO. Keduanya memiliki fungsi yang sama, namun versi dengan 30 pin GPIO dipilih karena memiliki dua pin GND tambahan. Setiap pin pada board ini diberi label yang terletak di bagian atas, sehingga mudah dikenali dan diidentifikasi. *Board* ini juga dilengkapi dengan antarmuka USB to UART yang memudahkan dalam pemrograman menggunakan program pengembangan aplikasi seperti Arduino IDE. Sumber daya untuk board ini dapat disediakan melalui konektor *micro* USB (Wagyana, 2019).

## 2.4. Internet Of Things

Internet of Things (IoT) merupakan skenario dari suatu objek yang dapat melakukan suatu pengiriman data atau informasi melalui jaringan tanpa campur tangan manusia. *Internet of Things* sangat erat hubungannya dengan komunikasi mesin dengan mesin tanpa campur tangan manusia ataupun *smart computer* (Limantara et al., 2017).

Teknologi-teknologi dalam *Internet of Things (IoT)* terhubung dengan berbagai terminal pengumpulan data melalui jaringan internet maupun jaringan komunikasi lainnya. Informasi mengenai lingkungan disekitar objek diambil secara *real time*, kemudian diubah ke dalam format data yang sesuai untuk ditransmisikan melalui jaringan dan dikirim ke pusat data. Data tersebut kemudian di olah oleh pengolah cerdas dengan menggunakan komputasi awan dan teknologi komputasi cerdas lain yang dapat mengolah data dalam jumlah besar untuk mencapai tujuan IoT (Ernita Dewi Meutia, 2015).

Penggunaan *Internet of Things* bertujuan untuk membantu pengoperasian maupun pemantuan kerja alat secara di luar area lokal selama alat dan perangkat pemantau terhubung dengan koneksi dengan internet.

## 2.5. Arduino IDE

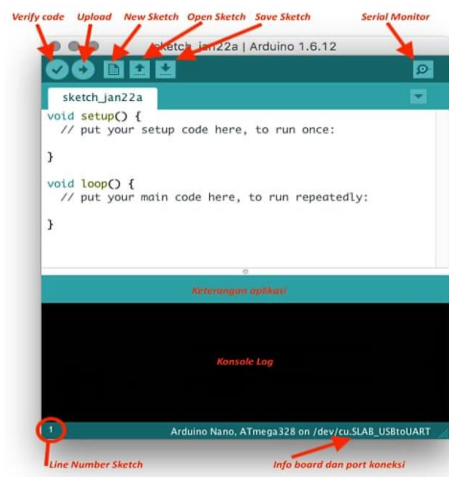
Arduino IDE adalah perangkat lunak yang menyediakan *Integrated Development Environment* (IDE) untuk membuat program pada *board* Arduino. Dalam konteks ini, Arduino IDE berfungsi sebagai perangkat lunak yang digunakan untuk membuat, mengedit, dan memvalidasi kode program. Selain

itu, Arduino IDE juga digunakan untuk meng-*upload* program ke *board* Arduino. Kode program yang digunakan pada Arduino dikenal sebagai "sketch" atau "*source code*" Arduino, dan biasanya memiliki ekstensi file `.ino`. (Endra et al., 2019).



Gambar 2. 5 Aplikasi Arduino IDE

(Sumber: <https://www.arduino.cc/en/software>)



Gambar 2. 6 Tampilan Layar Arduino IDE

(Sumber: <http://allgoblog.com/apa-itu-arduino-ide-dan-arduino-sketch/>)

Arduino IDE pada umumnya memiliki fitur untuk *cut* / *paste* dan *find* / *replace teks*, demikian juga pada arduino IDE. Pada bagian keterangan aplikasi memerlukan pesan balik saat menyimpan dan mengeksport serta sebagai tempat

menampilkan kssalahan. Konsol log menampilkan teks log dari aktifitas arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan port serial yang digunakan. Tombol toolbar terdapat ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat membuka, dan menyimpan sketch, dan membuka monitor serial.

- a. *Verivy* atau *Compile* merupakan sebuah tools untuk memverifikasi apakah code yang akan digunakan sudah benar sebelum di upload ke *board* Arduino.
- b. *Upload* merupakan sebuah perintah untuk mengirimkan kode program yang di buat pada Arduino IDE ke board Arduino.
- c. *New Sketch* perintah untuk membuka lembar baru untuk membuat Sketch pada Arduino IDE.
- d. *Open Sketch* membuka sketch yang pernah dibuat sebelumnya.
- e. *Save Sketch* menyimpan Sketch yang telah dibuat.
- f. *Serial Monitor* tampilan interface untuk komunikasi serial.
- g. Keterangan aplikasi merupakan sebuah pembacaan apakah sketch sudah Done Compiling atau Done Uploading.
- h. *Konsol Log* sebuah tampilan pesan-pesan ketika sedang proses compile atau Upload dan juga pesan-pesan tentang informasi error pada sketch.
- i. *Baris Sketch* menunjukkan posisi baris kursor yang sedang aktif pada *Sketch*.
- j. Informasi Board dan Port pada bagian ini berisi informasi *port* yang dipakai pada Arduino.



## 2.6. Firebase

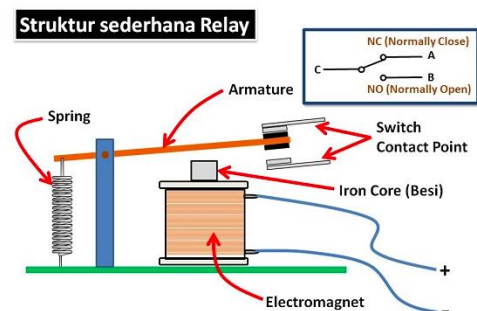
Firebase adalah layanan *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. Firebase dirancang untuk menyederhanakan pengembangan aplikasi mobile. Salah satu keunggulan Firebase adalah adanya fitur *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu, Firebase juga menyediakan fitur *Firebase Notification Console* untuk mendukung aplikasi yang membutuhkan pengiriman notifikasi push.

Firebase Database merupakan penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data ini antara lain String, Long, dan Boolean. Data pada Firebase Database disimpan sebagai objek JSON *tree*. Tidak seperti basis data SQL, tidak ada tabel dan baris pada basis data non-SQL. Ketika ada penambahan data, data tersebut akan menjadi *node* pada struktur JSON. *Node* merupakan simpul yang berisi data dan bisa memiliki cabang-cabang berupa *node* lainnya yang berisi data pula. Proses pengisian suatu data ke Firebase Database dikenal dengan istilah *push*.

Selain Firebase Database, Firebase menyediakan berbagai layanan tambahan yang juga digunakan dalam pengembangan aplikasi ini. Salah satu layanan tersebut adalah *Firebase Storage*. *Firebase Storage* berfungsi sebagai penyimpanan awan yang memungkinkan pengembang aplikasi untuk mengunggah dan mengunduh berkas. Dengan menggunakan *Firebase Storage*, pengguna dapat menyimpan dan mengakses berbagai jenis berkas seperti gambar, video, audio, dan lain sebagainya dalam pengembangan aplikasi (Sandy et al., 2017).

## 2.7. Relay

Relay adalah sebuah perangkat saklar yang dikontrol oleh arus listrik. Komponen ini terdiri dari sebuah kumparan dengan tegangan rendah yang dililitkan pada inti di dalamnya. Ketika arus mengalir melalui kumparan, armatur yang terbuat dari bahan besi akan tertarik ke inti. Armatur ini terhubung dengan tuas yang memiliki pegas. Saat armatur tertarik ke inti, posisi kontak dalam relay akan berubah dari kontak normal-tertutup menjadi kontak normal-terbuka. Dalam keadaan normal-tertutup, jalur listrik terhubung, sedangkan dalam keadaan normal-terbuka, jalur listrik terputus (Alexander & Turang, 2015).



Gambar 2. 7 Struktur Sederhana Relay

(Saleh & Haryanti, 2017)

Gambar 2.7 menunjukkan struktur sederhana dari relay. Ada beberapa fungsi relay yaitu menjadi selector (pemilih hubungan), eksekutor rangkaian delay (tunda) dan sebagai protector (pemutus arus pada kondisi tertentu). Di bawah ini merupakan sifat-sifat dari relay:

1. Impedansi kumparan, biasanya ditentukan oleh tebal kawat yang digunakan serta banyaknya lilitan. Nilai impedansi berkisar 1-50K $\Omega$  untuk memperoleh daya hantar yang baik.
2. Besar daya yang diperlukan untuk mengoperasikan relay sama dengan nilai tegangan dikalikan arus.
3. Banyaknya kontak-kontak jangkar dapat membuka dan menutup lebih dari satu kontak sekaligus. Hal tersebut tergantung pada kontak dan jenis relaynya. Jarak antara kontak-kontak menentukan besarnya tegangan maksimum yang diizinkan antara kontak tersebut.

Selain fungsi di atas, terdapat beberapa fungsi relay yang telah umum diaplikasikan ke dalam peralatan elektronik diantaranya adalah: (Saleh & Haryanti, 2017)

1. *Logic function*, menjalankan fungsi logika.
2. *Time delay function*, memberikan fungsi penundaan waktu.
3. Digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan signal tegangan rendah.

Kontak poin relay terdiri dari dua jenis yaitu:

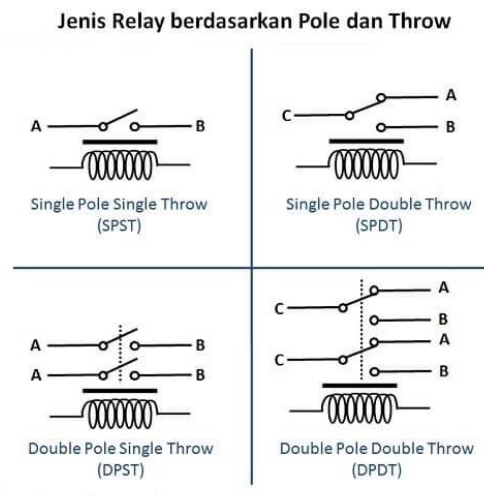
1. *Normally Close (NC)*, kondisi awal sebelum diaktifkan akan selalu berada di posisi *close* (tertutup).
2. *Normally Open (NO)*, kondisi awal sebelum diaktifkan akan selalu berada di posisi *open* (terbuka).

Karena relay merupakan salah satu jenis saklar, maka istilah *pole* dan *throw* yang dipakai dalam saklar juga berlaku pada relay. Di bawah ini adalah penjelasan singkat mengenai istilah tersebut:

1. *Pole*, banyaknya kontak yang dimiliki sebuah relay.
2. *Throw*, banyaknya kondisi yang dimiliki oleh sebuah kontak.

Berdasarkan penggolongan jumlah *pole* dan *throw* sebuah relay, maka relay dapat digolongkan menjadi:

1. *Single Pole Single Throw* (SPST), relay golongan ini memiliki 4 terminal dimana 2 terminal untuk saklar dan 2 terminal lainnya untuk *coil*.
2. *Single Pole Double Throw* (SPDT), relay golongan ini memiliki 5 terminal dengan 3 terminal untuk saklar dan 2 terminal lainnya untuk *coil*.
3. *Double Pole Single Throw* (DPST), relay golongan ini memiliki 6 terminal. Diantaranya adalah 4 terminal yang terdiri dari 2 pasang terminal saklar sedangkan 2 terminal lainnya untuk *coil*. Relay DPST dapat dijadikan 2 saklar yang dikendalikan 1 *coil*.



Gambar 2. 8 Jenis Relay Berdasarkan *Pole Dan Throw*

(Saleh & Haryanti, 2017)

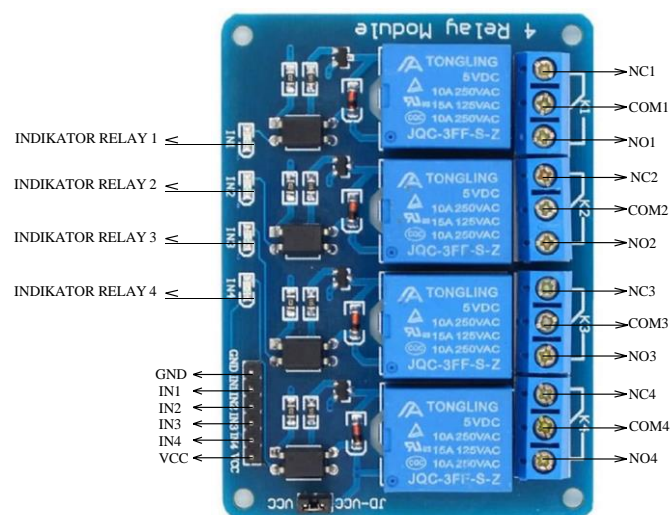
Gambar 2.8 menunjukkan beberapa jenis relay berdasarkan *pole* and *throw* yang terdiri dari *single pole single throw*, *single pole double throw*, *double pole single throw*, dan *double pole double throw*. Selain golongan relay di atas, terdapat juga relay dengan *pole* dan *throw* melebihi 2, diantaranya adalah *Triple Pole Double Throw* (3PDT) atau *Four Pole Double Throw* (4PDT).

Penggunaan relay ini dapat menyambungkan maupun memutuskan aliran listrik pada sebuah rangkaian tanpa perlu pengoperasian manual oleh manusia secara langsung. Pengaturan penyuplaian arus terhadap relay dapat menciptakan kendali sistem *switching* otomatis.

## 2.8.Modul Relay 4 Channel

Modul Relay 4 channel adalah sebuah perangkat yang memiliki empat relay di dalamnya. Modul ini memungkinkan pengguna untuk mengoperasikan relay secara individu atau secara bersamaan. Modul ini memiliki dua sisi yang berbeda,

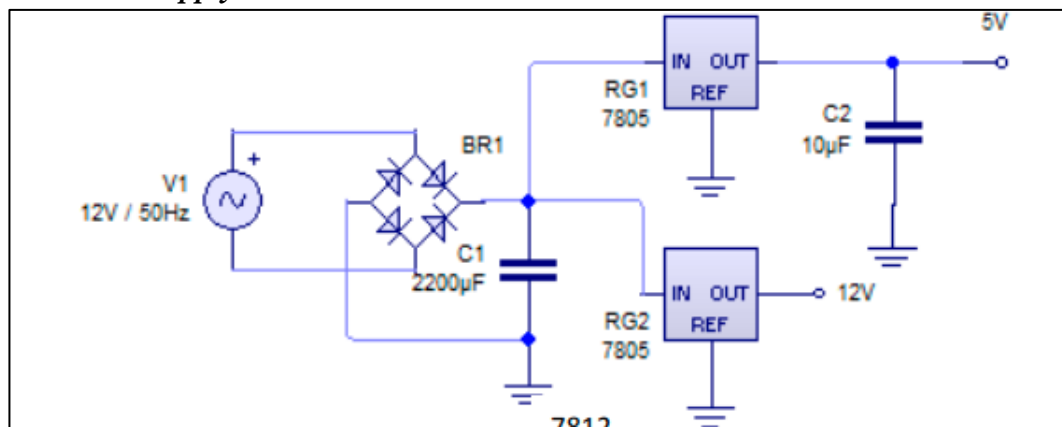
yaitu sisi trigger dan sisi switch. Pada sisi trigger, terdapat beberapa terminal dengan fungsi khusus. Terminal VCC atau *power supply* digunakan sebagai sumber daya dengan tegangan DC 5V. Terdapat juga terminal ground sebagai titik referensi nol. Selain itu, ada terminal IN1, IN2, IN3, dan IN4 yang digunakan sebagai input data untuk mengontrol masing-masing relay. Dengan kata lain, terminal IN berfungsi sebagai sinyal masukan yang mengendalikan sisi *switch* pada relay. Pengguna dapat mengirimkan sinyal logika (misalnya *HIGH* atau *LOW*) ke terminal input yang sesuai (IN1, IN2, IN3, atau IN4) untuk mengontrol relay yang diinginkan. Modul Relay 4 channel ini memungkinkan pengguna untuk mengendalikan hingga empat perangkat atau rangkaian secara independen dengan menggunakan sinyal masukan yang sesuai pada terminal input relay yang terkait. Dengan adanya modul ini, pengguna dapat dengan mudah melakukan otomatisasi atau pengendalian perangkat listrik dan elektronik melalui pengaturan sinyal pada modul relay ini.



Gambar 2. 9 Modul Relay 4 Channel

Gambar 2.9 menampilkan empat relay yang masing-masing memiliki keluaran NC (*Normally Close*), NO (*Normally Open*), dan COM (*Common*). Pada setiap relay, bagian Common terhubung dengan salah satu dari NC atau NO sesuai dengan konfigurasi yang diinginkan. Jika rangkaian dihubungkan antara NO dan COM, maka saat relay belum diaktifkan, sakelar pada relay akan berada dalam posisi terbuka, sehingga rangkaian akan terbuka. Namun, jika rangkaian dihubungkan antara NC dan COM, maka saat relay belum diaktifkan, sakelar pada relay akan berada dalam posisi tertutup, sehingga rangkaian akan tertutup.

### 2.9. Power Supply



Gambar 2. 10 Rangkaian *Power Supply* (Adewale et al., 2013)

*Power Supply* adalah sebuah perangkat yang digunakan untuk menyediakan daya pada satu atau lebih beban listrik. Untuk itu diperlukan catu daya yang mampu mengatur tegangan keluarannya sesuai dengan beban yang digunakan. Secara umum, catu daya DC berarti mengubah tegangan AC menjadi tegangan DC. (Putra et al., 2020).

Tegangan yang dihasilkan oleh transformator awalnya berbentuk gelombang bolak-balik (AC), yang perlu diubah menjadi tegangan searah (DC). Untuk mencapai hal ini, digunakan sebuah rangkaian penyearah yang menggunakan empat dioda yang biasa disebut dioda jembatan. Dioda jembatan dirancang khusus untuk mengubah kedua siklus gelombang AC menjadi arah yang sama. Setelah melalui dioda jembatan, gelombang yang semula dua arah berubah menjadi arah tunggal sebagai keluaran. Namun, tegangan keluaran ini masih memiliki riak, *noise*, atau amplitudo yang tidak merata. Hal ini terjadi karena dioda jembatan hanya menghilangkan komponen negatif dari gelombang AC dan masih terdapat fluktuasi pada tegangan positifnya. Siklus negatif gelombang awal diubah menjadi siklus positif tanpa mengubah bentuk gelombang, yang tetap memiliki lembah dan puncak. Untuk mengatasi ini, digunakan kapasitor dengan kapasitansi yang cukup besar untuk meratakan gelombang tersebut.

Ketika dioda mengalirkan arus, kapasitor akan mengisi dirinya sesuai dengan bentuk gelombang yang masuk. Setelah tegangan masukan mencapai nilai puncaknya, tegangan tersebut akan dipertahankan jika tidak ada beban yang mengambilnya. Namun, jika ada beban yang terhubung, tegangan pada kapasitor akan berkurang sesuai dengan besarnya beban tersebut. Hal ini terjadi karena kapasitor membutuhkan waktu untuk melepaskan muatan, sehingga amplitudo gelombang menjadi lebih merata. Setelah tegangan mencapai tingkat kestabilan yang cukup, tegangan tersebut dikendalikan oleh IC regulator. IC regulator adalah rangkaian elektronik yang bertugas menjaga agar tegangan keluaran tetap stabil ketika ada perubahan beban. Setelah melewati IC regulator, tegangan kemudian



mengalir melalui LED. Saat tegangan melalui LED, LED akan menyala. Pada rangkaian catu daya ini, LED digunakan sebagai indikator atau penanda.

## 2.10. Lampu LED



Gambar 2. 11 Lampu LED

Lampu LED atau *Light Emitting Diode* adalah jenis diode semikonduktor yang memiliki kemampuan untuk menghasilkan cahaya ketika diberi tegangan listrik. Keunggulan utama LED adalah efisiensinya yang tinggi dalam penggunaan energi, dimana mampu menghasilkan cahaya dengan tingkat kecerahan antara 60 hingga 100 Lumens per Watt. Teknologi LED terus berkembang seiring dengan kemajuan teknologi semikonduktor yang semakin meningkatkan kualitasnya (Mustaqim & Haddin, 2017).

Cahaya yang dihasilkan oleh LED adalah bentuk energi elektromagnetik yang terlihat oleh manusia. Cahaya tersebut terjadi karena kombinasi berbagai panjang gelombang yang berbeda dari energi yang dapat terlihat. Manusia memiliki kemampuan untuk melihat cahaya pada rentang panjang gelombang tertentu, yaitu antara radiasi ultraviolet dan inframerah. Proses terjadinya cahaya ini melibatkan pergerakan elektron di dalam atom.

Pada atom, elektron bergerak mengelilingi inti atom dalam orbit tertentu. Setiap orbit memiliki tingkat energi yang berbeda-beda. Ketika elektron berpindah dari orbit dengan tingkat energi yang lebih tinggi ke orbit dengan tingkat energi yang lebih rendah, ia melepaskan energi yang dimilikinya. Energi ini dilepaskan dalam bentuk foton, yang pada akhirnya menghasilkan cahaya. Besar energi yang dilepaskan oleh elektron mempengaruhi energi yang terkandung dalam foton yang dihasilkan. Semakin besar energi yang dilepaskan, semakin besar pula energi yang terkandung dalam foton tersebut (Suryana, 2013).

## 2.11. Sensor LDR

LDR (Light Dependent Resistor) adalah salah satu jenis resistor yang dikenal sebagai fotoresistor. Sifat unik dari LDR adalah bahwa nilai hambatannya akan berubah seiring dengan intensitas cahaya yang diterimanya dari lingkungan sekitar. Dalam kondisi gelap, LDR memiliki resistansi yang tinggi, sedangkan dalam kondisi terang, resistansinya menjadi rendah. Untuk menghitung tegangan keluaran pada LDR digunakan persamaan berikut:

$$V_o = \frac{LDR}{LDR+R_1} V_{CC}$$

Dimana:

$V_o$  Tegangan keluaran

LDR Resistansi LDR

$R_1$  Resistor

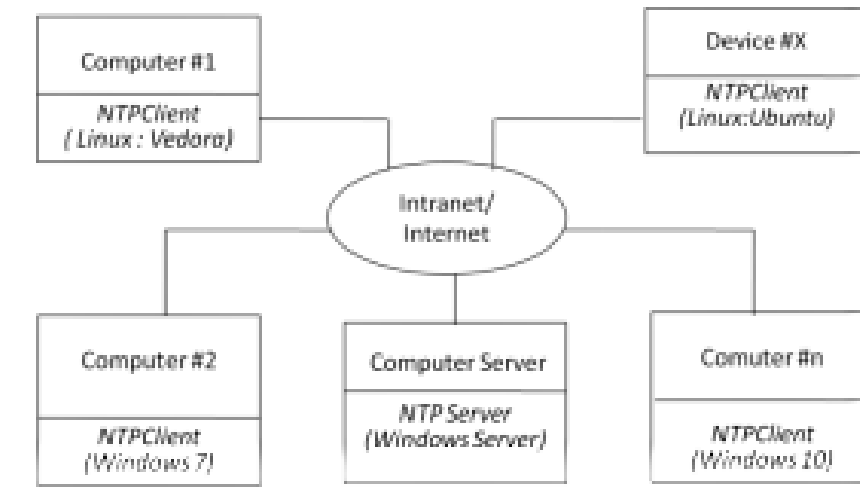
$V_{cc}$  Tegangan Masuk

Perubahan resistansi pada LDR disebabkan oleh efek fotokonduktif, di mana cahaya yang jatuh pada permukaan LDR menyebabkan perubahan pada struktur material semikonduktor di dalamnya. Ketika cahaya mengenai LDR, energi foton yang terkandung dalam cahaya tersebut mampu memutuskan ikatan-ikatan elektron dalam material semikonduktor, sehingga jumlah pembawa muatan (elektron atau lubang) akan meningkat. Peningkatan jumlah pembawa muatan dalam LDR menyebabkan resistansi LDR menurun, karena adanya lebih banyak jalur konduksi yang tersedia bagi arus listrik. Sebaliknya, dalam kondisi minim

cahaya, sedikitnya pembawa muatan mengakibatkan resistansi LDR meningkat, sehingga arus listrik yang mengalir melalui LDR menjadi terbatas. Untuk mengukur tegangan keluaran dari LDR, serta dapat menggunakan persamaan yang berkaitan dengan hukum Ohm. Persamaan tersebut menghubungkan resistansi LDR dengan arus listrik yang mengalir melalui LDR dan tegangan yang terbentuk di kedua ujungnya. Dengan mengetahui nilai resistansi LDR dan arus yang mengalirinya, kita dapat menghitung tegangan keluaran menggunakan persamaan tersebut. Dalam sistem kontrol penerangan otomatis, sensor cahaya berbasis LDR digunakan untuk mengukur intensitas cahaya di sekitar penerangan. Data intensitas cahaya yang diperoleh dari sensor LDR akan digunakan sebagai masukan untuk pengendalian penerangan sesuai dengan kondisi lingkungan yang diinginkan (Supatmi, 2010).

## **2.12. Network Time Protocol**

Sebagai gambaran bagaimana cara melakukan sinkronisasi waktu, pada bab ini dijelaskan sekilas tentang penggunaan NTP yaitu sebuah protokol telah digunakan oleh Microsoft dan Linux. NTP adalah singkatan dari Network Time Protocol, adalah sebuah protocol untuk meng-sinkronkan waktu sistem (clock) pada komputer terhadap sumber yang akurat, melalui jaringan intranet ataupun internet. NTP Server mentransmisikan paket informasi waktu kepada komputer client melalui NTP Client yang melakukan request. Ilustrasi penggunaan NTP ditunjukkan pada Gambar 2.1. NTP Server sangat bermanfaat sekali untuk mengelola jaringan yang sangat ketat sekali dalam urusan waktu.



Gambar 2. 12 Network Time Protocol

NTP menggunakan port komunikasi UDP dengan nomor 123. Protokol ini didesain untuk dapat bekerja dengan baik meskipun media komunikasinya bervariasi, mulai dari yang waktu latensinya tinggi hingga yang rendah, mulai dari media kabel sampai dengan media udara. Protokol ini memungkinkan perangkatperangkat komputer dalam jaringan tetap dapat melakukan sinkronisasi waktu dengan sangat tepat dalam berbagai media tersebut. Sebuah jaringan NTP biasanya mendapatkan perhitungannya dari sumber waktu yang terpercaya seperti misalnya radio clock atau atomic clock yang terhubung dengan sebuah time server. Kemudian jaringan NTP ini akan mendistribusikan perhitungan waktu akurat ini ke dalam jaringan lain. Sebuah NTP client akan melakukan sinkronisasi dengan NTP server dalam sebuah interval pooling yang biasanya berkisar antara 64 sampai 1024 detik. Namun, waktu sinkronisasi ini bisa berubah secara dramatis bergantung kepada kondisi dan keadaan jaringan yang akan digunakannya. Untuk mengaplikasikan NTP, perlu dilakukan setting atau

pengaturan baik pada sisi server maupun pada sisi client. Pengaturan yang perlu dilakukan pada sisi server :

- *Enable Windows Time Services*
- *Enable NTP Server*
- *Running NTP Server*

Sedangkan pengaturan pada sisi client adalah:

- *Enable Time Service*
- *Enable NTPClient*
- *Menentukan IP address NTP Server*
- *Running NTPClient*

Akurasi dari sinkronisasi waktu tergantung pada time server yang digunakan. Time Server adalah komputer server yang membaca waktu aktual dari jam referensi dan mendistribusikan informasi ini kepada kliennya menggunakan jaringan komputer. Server waktu dapat berupa server waktu jaringan lokal atau server waktu internet. Referensi waktu yang digunakan oleh Time Server dapat berupa server waktu lain di jaringan atau internet, jam radio yang terhubung, atau jam atom. Sumber waktu sebenarnya yang paling umum adalah GPS atau jam master GPS. Tugas dari Time Server adalah menyediakan waktu yang akurat. Server jaringan yang ada (misalnya server file) dapat menjadi server waktu dengan perangkat lunak tambahan (Iswanto, 2019).