

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian terdahulu yang relevan dengan penelitian ini ialah :

Tabel 2.1 Penelitian Terdahulu

| No. Literatur | Penulis | Tahun | Judul |
|---------------|---|-------|---|
| Literatur 1 | Fahrudin Nur Iksan, Gunawan Tjahjadi | 2018 | Perancangan Stop Kontak Pengendali Energi Listrik Dengan Sistem Keamanan Hubung Singkat Dan Fitur Notifikasi Berbasis <i>Internet of Things</i> (IoT) |
| Literatur 2 | Ari Farhan Nurihsan, Irving V Paputungan | 2020 | Perancangan Stop Kontak Cerdas Berbasis IoT |
| Literatur 3 | Nur Rohman, Yuli Christyono, Sukiswo | 2020 | Perancangan Sistem Monitoring Tegangan, Daya dan Faktor Daya Pada Stop kontak Berbasis IoT |
| Literatur 4 | Utsman Al-Aydarus, Umar Ali Ahmad, Randy Erfa Saputra | 2022 | Perancangan Stop kontak Pintar berbasis IoT Menggunakan Real Time Clock |
| Literatur 5 | M. Abu Jihad Plaza R, Yulina, Sigit Gunanto | 2022 | Penerapan IoT Pada Stop kontak Lampu Berbasis Arduino |

2.1.2 Literatur 1

Pada penelitian yang dilakukan oleh Fahrudin Nur Iksan, Gunawan Tjahjadi (2018) Dalam penelitian ini, sistem yang diciptakan adalah sebuah stop kontak cerdas yang memiliki kemampuan untuk mengontrol energi listrik, menyediakan notifikasi, dan memiliki sistem keamanan hubung singkat yang dapat dikontrol melalui internet. Penelitian ini bertujuan untuk memudahkan masyarakat dalam mengendalikan dan memantau perangkat keras dari jarak jauh menggunakan teknologi *Internet of Things* (IoT). Alat yang digunakan dalam penelitian ini terdiri dari rangkaian sensor arus TA12-100, sensor suhu DS18B2, dan relay sebagai pemisah antara tegangan DC dan AC. Berdasarkan hasil pengujian, rata-rata perbedaan antara alat pembanding dengan power socket 1, power socket 2, dan power socket 3 adalah 0,004 A, 0,010 A, dan 0,002 A. Sedangkan rata-rata kesalahan antara alat pembanding dengan power socket 1, power socket 2, dan power socket 3 adalah 3,892%, 6,932%, dan 2,847%. Selain itu, data pengujian dengan rata-rata kesalahan 0,01 A juga menunjukkan hasil yang serupa untuk sensor TA12-100.

2.1.3 Literatur 2

Pada penelitian yang dilakukan oleh Ari Farhan Nurihsan, Irving V Paputungan (2020) Dalam penelitiannya membahas mengenai rancangan stop kontak cerdas berbasis *Internet of Things* (IoT) yang menggunakan Wemos DI RI, Relay, *Blynk*, dan Sensor PIR. Tujuan dari alat ini adalah untuk mengurangi penggunaan listrik yang berlebihan yang dapat menyebabkan korsleting listrik. Stop kontak cerdas berbasis IoT ini dapat menghidupkan perangkat elektronik yang terhubung ke dalam stop kontak hanya ketika ada orang, dan akan mematikannya saat tidak ada orang. Selain itu, stop kontak cerdas ini juga terhubung dengan aplikasi *Blynk* agar dapat dikontrol secara mudah dari jarak jauh. Berdasarkan hasil pengujian untuk periode harian, diperkirakan terdapat efisiensi daya sekitar 1,2 kWh. Efisiensi ini juga dapat diukur dalam nilai rupiah sebesar Rp. 1.761,38 untuk periode harian.

2.1.4 Literatur 3

Penelitian yang dilakukan oleh Nur Rohman, Yuli Christyanto, dan Sukiswo (2020) menggunakan ESP32 sebagai mikrokontroler, serta sensor ZMPT101B sebagai pembaca tegangan dan sensor ZMCT103C sebagai pembaca arus. Tujuan dari penelitian ini adalah untuk memaksimalkan penggunaan energi listrik dengan memanfaatkan teknologi yang semakin berkembang, salah satunya adalah *Internet of Things* (IoT). Dalam penelitian ini, IoT berperan dalam mengatur tegangan, daya, dan fake daya pada beban yang terhubung ke dalam stop kontak.

2.1.5 Literatur 4

Dalam penelitian yang dilakukan oleh Utsman Al Aydarus, Umar Ali Ahmad, dan Randy Erfa Saputra (2020), pengembangan stop kontak dilakukan dengan menggunakan aplikasi atau web pada *smartphone* yang berbasis *Internet of Things* (IoT). IoT adalah sebuah sistem yang berperan dalam menyimpan dan mentransfer data melalui jaringan nirkabel tanpa melibatkan interaksi manusia secara langsung. Sistem ini dapat dikendalikan hanya dengan menggunakan internet atau jaringan lokal seperti bluetooth. Komponen yang digunakan dalam penelitian ini adalah RTC (Real Time Clock) untuk mengatur pemutusan atau pemasangan arus listrik secara otomatis melalui *smartphone*. Mikrokontroler yang digunakan adalah Arduino Uno R3 dan *NodeMCU* Esp8266.

2.1.6 Literatur 5

Pada penelitian yang dilakukan oleh M. Abu Jihad Plaza R, Yulina, Sigit Gunanto (2022). Penelitian ini bertujuan untuk mempermudah penggunaan dan menghemat penggunaan listrik bagi masyarakat. Pendekatan yang digunakan dalam penelitian ini adalah dengan menggunakan mikrokontroler Arduino yang terhubung melalui komunikasi Bluetooth sehingga lampu dapat dinyalakan dan dimatikan melalui perintah yang dikirimkan melalui *smartphone* Android. Setelah dilakukan pengujian, hasilnya menunjukkan bahwa proses pengiriman dan penerimaan data perintah on-off lampu dapat berfungsi dengan baik pada jarak sekitar 10 meter. Dengan demikian, pengguna hanya dapat melakukan pengontrolan menggunakan *smartphone* dalam jarak kurang dari 10 meter.

2.1.7 Kelebihan Alat Stop Kontak Cerdas Berbasis IoT

Beberapa kelebihan alat stop kontak pintar berbasis *Internet of Things* (IoT) antara lain:

1. Kemudahan penggunaan: Dengan adanya fitur kontrol jarak jauh melalui aplikasi, pengguna dapat dengan mudah mengatur dan mengontrol perangkat listrik yang terhubung ke stop kontak pintar tanpa harus berada di dekatnya.
2. Efisiensi energi: Stop kontak pintar dapat membantu menghemat penggunaan energi listrik dengan cara mematikan perangkat listrik ketika tidak digunakan, atau mengatur jadwal waktu penggunaan perangkat. Hal ini dapat mengurangi biaya listrik dan berkontribusi pada pengurangan emisi gas rumah kaca.
3. Kemampuan monitoring: Stop kontak pintar dapat memberikan informasi tentang penggunaan listrik pada perangkat yang terhubung. Hal ini dapat membantu untuk mengidentifikasi perangkat yang boros energi dan mengoptimalkan penggunaan energi listrik.

2.2 *NodeMCU* ESP8266

NodeMCU adalah sebuah platform *open source* yang digunakan untuk pengembangan *Internet of Things* (IoT) dan kit prototyping. Platform ini menggunakan bahasa pemrograman eksternal yang dapat digunakan untuk membuat prototipe produk IoT atau menggunakan *Sketch* dengan Arduino IDE. *NodeMCU* didasarkan pada modul ESP8266 yang menyediakan fitur GPIO, PWM (Pulse Width Modulation), IIC, 1-Wire, dan ADC (Analog to Digital Converter) yang terintegrasi dalam satu board.

NodeMCU memiliki dimensi panjang 4,83 cm, lebar 2,54 cm, dan berat 7 gram. Board ini sudah dilengkapi dengan fitur *Wi-Fi* dan firmware yang bersifat *open source*. Dengan konsep *open source*, komunitas dapat bekerja sama untuk meningkatkan kinerja dan fungsionalitas *NodeMCU* ESP, memungkinkan pengguna memanfaatkannya sepenuhnya sesuai dengan proyek mereka. Ini berarti bahwa kode sumber atau program yang digunakan

dalam *NodeMCU* ESP dapat diakses dan diubah oleh siapa saja secara bebas, memungkinkan pengembang untuk menyesuaikan dan memperbaiki program sesuai kebutuhan mereka.

Spesifikasi *NodeMCU* mencakup:

1. Dibuat dengan menggunakan modul ESP8266 yang memiliki *Wi-Fi* terintegrasi dengan dukungan untuk standar IEE 802.11b/g/n. Dilengkapi dengan USB to TTL untuk koneksi dengan komputer.
2. Menggunakan dua kapasitor tantalum 100 mikro farad dan 10 mikro farad.
3. Terdapat regulator LDO 3V untuk mendukung kebutuhan daya.
4. Dilengkapi dengan LED biru sebagai indikator.
5. Menggunakan bridge USB to UART Cp2102 untuk komunikasi.
6. Terdapat tombol reset, port USB, dan tombol flash.
7. Memiliki 9 GPIO yang termasuk 3 pin PWM, 1 saluran ADC, serta pin RX dan TX.
8. Terdapat 3 pin *ground*.
9. S3 dan S2 dapat digunakan sebagai pin GPIO.
10. Mendukung jalur komunikasi SI MOSI (*Master Output Slave Input*) untuk mengirim data dari *master* ke *slave*.
11. Mendukung jalur komunikasi SO MISO (*Master Input Slave Output*) yang memungkinkan *slave* untuk mengirim data kepada *master*.
12. SK sebagai SCLK adalah sinyal clock yang digunakan untuk sinkronisasi antara *master* dan *slave*.
13. Terdapat pin Vin sebagai input tegangan.
14. Dibangun dengan mikrokontroler 32-bit.

6. GPIO12 berperan sebagai pin MISO (*Master In Slave Out*) dalam komunikasi serial SPI (*Serial Peripheral Interface*). Pin ini digunakan untuk mengirim data dari perangkat *slave* ke perangkat *master* dalam komunikasi SPI.
7. GPIO13 berperan sebagai pin SCLK (*Serial Clock*) dalam komunikasi serial SPI (*Serial Peripheral Interface*). Pin ini digunakan sebagai sinyal *clock* yang mengatur kecepatan *transfer* data antara perangkat *master* dan *slave* dalam komunikasi SPI.
8. VCC : VCC pada *NodeMCU* ESP8266 adalah pin pasokan daya atau tegangan yang digunakan untuk memberikan daya ke board. Fungsi VCC adalah untuk menghubungkan *NodeMCU* ESP8266 ke sumber daya eksternal, seperti baterai atau adaptor daya, untuk menjalankan board dan memasok listrik ke komponen lainnya yang terhubung ke *NodeMCU* ESP8266.
9. CS0 : Fungsi utama CS0 adalah untuk memungkinkan *NodeMCU* ESP8266 berkomunikasi dengan berbagai perangkat SPI eksternal yang terhubung, seperti sensor, memori, atau modul lainnya. Dengan menggunakan beberapa *chip select*, *NodeMCU* dapat berinteraksi dengan beberapa perangkat melalui SPI dengan menggunakan pin yang terpisah untuk masing-masing perangkat.
10. MISO : MISO adalah untuk mengirimkan data dari *slave* (misalnya sensor, EEPROM, atau LCD) ke *master* (misalnya *NodeMCU* ESP8266). Secara lebih spesifik, MISO digunakan sebagai jalur *output* serial yang mengirimkan data dari perangkat *slave* ke perangkat *master*. Ketika *master* mengirimkan perintah tertentu ke *slave*, *slave* akan mengirimkan respons atau data yang diminta melalui jalur MISO kepada *master*.
11. GPIO9 adalah salah satu pin yang digunakan sebagai GPIO (*General Purpose Input Output*).
12. GPIO10 juga dapat digunakan sebagai pin khusus untuk komunikasi SPI (*Serial Peripheral Interface*). Dalam setup SPI, pin GPIO10 digunakan sebagai MOSI (*Master Out Slave In*), yaitu jalur keluar dari *master* dan jalur masuk ke *slave* dalam komunikasi SPI.

13. MOSI : MOSI (*Master Out Slave In*) adalah salah satu pin yang digunakan dalam komunikasi SPI (*Serial Peripheral Interface*). Pin MOSI adalah jalur keluar dari *master* dan jalur masuk ke *slave* dalam komunikasi SPI.
14. SCLK : SCLK (*Serial Clock*) adalah salah satu pin yang digunakan dalam komunikasi SPI (*Serial Peripheral Interface*). Pin SCLK digunakan untuk mensinkronisasi transmisi data antara *master* dan *slave* dalam komunikasi SPI.
15. GND : GND (*Ground*) merupakan koneksi penting yang harus dilakukan untuk menghubungkan node adalah Mikrokontroler ke tanah atau *ground*. Pemisahan sirkuit yang efektif memerlukan titik referensi, dan ini disebut *ground* (tanah). GND pada *NodeMCU* ESP8266 berfungsi sebagai referensi nol atau koneksi tanah untuk semua sinyal di dalamnya.
16. IO15: GPIO15; MTDO; HSPICS; UART0_RTS : GPIO15 berperan dalam pemilihan mode *boot*. Dalam beberapa konfigurasi, GPIO15 dapat digunakan untuk memilih mode *boot* yang berbeda pada saat *NodeMCU* ESP8266 dinyalakan. Anda dapat mengatur GPIO15 dengan resistor *pull-up* atau *pull-down* agar pada saat *boot*, *NodeMCU* ESP8266 masuk ke mode yang diinginkan, seperti mode *Flashing Firmware* atau mode *Normal Execution*.
17. GPIO2 dapat digunakan untuk memilih mode *boot* yang berbeda pada saat *NodeMCU* ESP8266 dinyalakan. Anda dapat mengatur GPIO2 dengan resistor *pull-up* atau *pull-down* agar pada saat *boot*, *NodeMCU* ESP8266 masuk ke mode yang diinginkan, seperti mode *Flashing Firmware* atau mode *Normal Execution*.
18. RXD (*Receive Data*) digunakan sebagai pin penerima data serial. Fungsi pin ini adalah untuk menerima data serial yang dikirimkan oleh perangkat eksternal ke *NodeMCU*. Dalam komunikasi serial, RXD digunakan untuk menerima data dari perangkat eksternal, seperti sensor, perangkat GPS, atau perangkat berbasis mikrokontroler lainnya.
19. TXD (*Transmit Data*) digunakan sebagai pin pengirim data serial. Fungsi pin ini adalah untuk mengirimkan data serial dari *NodeMCU* ke perangkat eksternal. Dalam komunikasi serial, TXD digunakan untuk mengirimkan

data ke perangkat eksternal, seperti mikrokontroler lain, module Bluetooth, atau perangkat lain yang dapat menerima data serial.

2.3 Module RTC DS3231

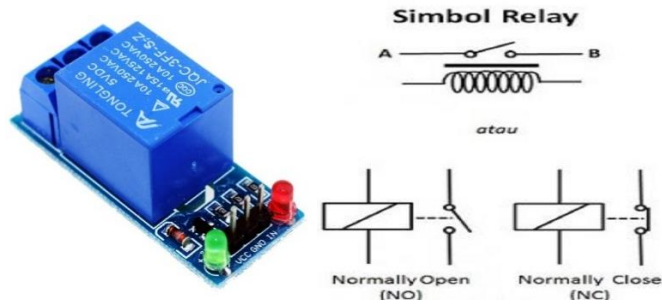
Modul RTC DS3231 adalah sebuah modul yang digunakan untuk mengukur waktu secara *real-time* (RTC) dan juga dilengkapi dengan pengukur suhu. Modul ini menyediakan semua fitur RTC dan suhu dalam satu paket. Contohnya dapat dilihat pada Gambar 2.2.



Gambar 2.2 Real Time Clock

2.4 Relay

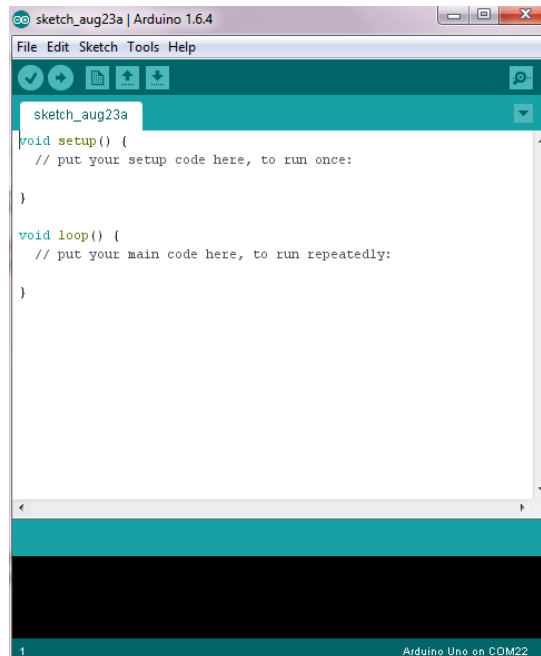
Relay adalah komponen elektronik yang berfungsi sebagai saklar atau *switch* listrik dan dioperasikan menggunakan listrik. Relay juga sering disebut sebagai komponen elektromekanik, karena terdiri dari dua bagian utama yaitu coil atau elektromagnet dan kontak saklar atau mekanik. Dalam komponen relay, prinsip elektromagnetik digunakan sebagai penggerak kontak saklar. Dengan menggunakan arus listrik kecil atau *low power*, relay dapat menghantarkan arus listrik dengan tegangan yang lebih tinggi. Pada **Gambar 2.3** terlihat gambar dan simbol komponen relay.



Gambar 2.3 Relay dan Simbol

2.5 Arduino IDE

Arduino IDE (*Integrate Development Environment*) adalah perangkat lunak yang digunakan untuk membuat, mengedit, memverifikasi, dan mengunggah kode program ke board Arduino. Arduino menggunakan bahasa pemrograman khusus. Arduino IDE memiliki berbagai fitur, seperti editor teks untuk membuat dan mengedit kode program, area pesan, *console* teks, dan *toolbar* serta tombol dengan fungsi umum. Program yang dibuat menggunakan Arduino IDE disebut dengan *Sketch*, dan ditulis dalam editor teks yang kemudian disimpan dengan ekstensi *.ino*. Gambar 2.4 menunjukkan tampilan Arduino IDE.



Gambar 2.4 Tampilan *Sketch* Arduino IDE

2.6 Stop Kontak

Soket merupakan salah satu bahan instalasi listrik yang berfungsi sebagai penghubung antara arus dengan peralatan listrik. Untuk menghubungkan suatu perangkat listrik ke stop kontak diperlukan kabel dan colokan atau colokan yang nantinya akan dicolokkan ke stop kontak. Soket digolongkan menjadi dua jenis berdasarkan bentuk dan fungsinya, yaitu :

- Stop kontak kecil adalah stop kontak yang mempunyai dua lubang (saluran), yang fungsinya mengalirkan listrik berdaya rendah ke peralatan listrik melalui colokan yang juga berukuran kecil.
- Stop kontak besar di dinding, juga stop kontak dengan dua saluran AC, dengan pelat logam di bagian atas dan bawah saluran AC berfungsi sebagai *ground*, sakelar jenis ini biasanya digunakan untuk daya yang lebih tinggi.



Gambar 2.5 Stop Kontak

2.7 Software

Driver IDE adalah komponen perangkat lunak yang memiliki beberapa perangkat lunak lain yang sangat berguna. *Integrated Development Environment (IDE)* adalah program khusus untuk komputer yang memungkinkan pembuatan desain atau sketsa program untuk papan Arduino. IDE Arduino terdiri dari beberapa komponen, seperti:

1. *Editor Program*: Ini adalah jendela yang digunakan untuk menulis dan mengedit program dalam bahasa *processing*.

2. *Compiler*: Berfungsi untuk mengkompilasi sketsa sebelum diunggah ke papan Arduino.
3. *Uploader*: Digunakan untuk mengunggah hasil kompilasi sketsa ke papan Arduino *target*. Jika ada kesalahan, pesan *error* akan ditampilkan di layar *log*.
4. *New Sketch*: Membuka jendela baru dan membuat sketsa baru.
5. *Open Sketch*: Membuka sketsa yang sudah dibuat sebelumnya. Sketsa yang dibuat dengan IDE Arduino disimpan di folder yang telah ditentukan sebelumnya
6. *Open Sketch*: Melakukan pembukaan *Sketch* yang telah dibuat sebelumnya. *Sketch* yang dibuat dengan IDE Arduino disimpan di dalam *folder* yang telah ditentukan sebelumnya.
7. *Monitor Serial*. Mengaktifkan antarmuka untuk komunikasi serial. Lebih lanjut akan dibahas pada bagian selanjutnya.
8. Status Aplikasi. Pesan-pesan yang dikeluarkan oleh aplikasi akan muncul di sini, seperti "*Compiling*" dan "*Done Uploading*" saat melakukan kompilasi dan mengunggah *Sketch* ke *board* Arduino.
9. *Log Konsol*. Pesan-pesan yang terkait dengan aplikasi dan pesan-pesan tentang *Sketch* akan muncul di sini. Misalnya, ketika aplikasi melakukan kompilasi atau ada kesalahan dalam *Sketch* yang dibuat, informasi kesalahan dan baris kode akan ditampilkan di sini.
10. Baris Aktif. Bagian ini akan menunjukkan posisi baris yang sedang aktif di dalam *Sketch*.
11. Informasi *Board* dan *Port*. Bagian ini akan memberikan informasi tentang port yang digunakan oleh board Arduino.

2.8 Hardware

Setiap jenis Arduino memiliki kelebihan dan kekurangan masing-masing. Arduino Uno adalah papan Arduino yang akan digunakan dalam penelitian ini. Arduino Uno dipilih karena sesuai dengan desain yang dibutuhkan dan memiliki fungsi tambahan yang relevan.