

**RANCANG BANGUN STOP KONTAK LISTRIK PINTAR
BERBASIS IOT**

Design And Develop Iot Based Smart Electric Stop Contact

Skripsi

Untuk memenuhi sebagian persyaratan
Mencapai derajat sarjana S-1

ACC Revisi
11/10/2023
[Signature]

Diusulkan Oleh :
Ifan Aditya Ramadhan
18316102

Acc Revisi
12/10/2023
[Signature]
Yuri Darmanto



Acc Cetak 16/10/23
[Signature]

**PROGRAM STUDI S1 TEKNIK KOMPUTER
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS TEKNOKRAT INDONESIA
BANDAR LAMPUNG
2023**

LEMBAR PENGESAHAN

RANCANG BANGUN STOP KONTAK LISTRIK PINTAR BERBASIS IOT

Dipersiapkan dan diajukan oleh

Ifan Aditya Ramadhan

18316102

Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik Komputer

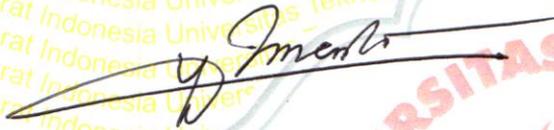
Telah dipertahankan di depan Dewan Penguji

Pada Tanggal, 12 September 2023

Penguji

Pembimbing,

Penguji,



Yuri Rahmanto, S.Kom., M.Kom.

NIK. 022 13 02 24



Dr. Ryan Randy Suryono, S.Kom., M.Kom.

NIK. 022 13 02 18

Skripsi ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana

Pada Tanggal, 16 Oktober 2023

Fakultas Teknik dan Ilmu Komputer

Dekan,

Program Studi S1 Teknik Komputer

Ketua,



Dr. H. Mahathir Muhammad, S.E., M.M.

NIK. 023 05 00 09



Styawati, S.T., M.Cs.

NIK. 022 09 10 04

LEMBAR PERSETUJUAN

RANCANG BANGUN STOP KONTAK LISTRIK PINTAR BERBASIS IOT

Dipersiapkan dan diajukan oleh

Ifan Aditya Ramadhan

18316102

Telah Disetujui
Tanggal 16 Oktober 2023

Diketahui :
Program Studi S1 Teknik Komputer
Ketua,

Pembimbing,



Styawati, S.T., M.Cs.
NIK. 022 09 10 04



Yuri Rahmanto, S.Kom., M.Kom.
NIK. 022 13 02 24



LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Ifan Aditya Ramadhan
NPM : 18316102
Program Studi : Teknik Komputer

Dengan ini menyatakan bahwa skripsi :

Judul : Rancang Bangun Stop Kontak Listrik Pintar Berbasis IoT

Pembimbing : Yuri Rahmanto, S.Kom., M.Kom.

Belum pernah diajukan untuk diuji sebagai persyaratan untuk memperoleh gelar akademik pada berbagai tingkatan di Universitas/Perguruan Tinggi manapun. Tidak ada bagian dalam skripsi ini yang pernah dipublikasikan oleh pihak lain, kecuali bagian yang digunakan sebagai referensi, berdasarkan kaidah penulisan ilmiah yang benar.

Apabila dikemudian hari ternyata skripsi ini yang saya tulis terbukti hasil saduran/plagiat, maka saya bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

Bandar Lampung, 16 Oktober 2023
Yang menyatakan,



Ifan Aditya Ramadhan

NPM. 18316102

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Fakultas teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia, saya yang bertanda tangan di bawah ini :

Nama : Ifan Aditya Ramadhan
NPM : 18316102
Program Studi : S1 Teknik Komputer

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia, **Hak Bebas Royaltas Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

“Rancang Bangun Stop Kontak Listrik Pintar Berbasis IOT”

Beserta perangkat yang ada (jika diperlukan). Dengan hak bebas royalti Noneksklusif ini Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan membulikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Bandar Lampung
Pada Tanggal : 16 Oktober 2023

Yang Menyatakan,



Ifan Aditya Ramadhan
NPM. 18316102

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat dan kasih sayang-Nya Kami dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mendapatkan Gelar Sarjana Teknik (S.T) pada Program Studi S1 Teknik Komputer Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia. Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Bapak Dr. H. M. Nasrullah Yusuf, SE., M.B.A., selaku Rektor Universitas Teknokrat Indonesia
2. Bapak Dr. H. Mahathir Muhammad, S.E., M.M. selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Teknokrat Indonesia.
3. Ibu Styawati, S.T., M.Cs. selaku Ketua Program Studi S1 Teknik Komputer.
4. Bapak Yuri Rahmanto, S.Kom., M.Kom. selaku pembimbing yang telah meluangkan waktu untuk membimbing skripsi ini.
5. Bapak Dr. Ryan Randy Suryono, S.Kom., M.Kom. selaku penguji yang telah meluangkan waktu untuk menguji demi kelancaran jalannya sidang skripsi ini.

Akhir kata, penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu dan semoga skripsi ini membawa manfaat.

Bandarlampung, 12 September 2023

Penulis,



Ifan Aditya Ramadhan

18316102

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	ii
LEMBAR PERSETUJUAN.....	iii
LEMBAR PERNYATAAN	iv
SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
KATA PENGANTAR.....	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	xi
ABSTRAK	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian.....	3
BAB II LANDASAN TEORI	5
2.1 Tinjauan Pustaka.....	5
2.1.1 Literatur 1	6
2.1.2 Literatur 2.....	6
2.1.3 Literatur 3.....	7
2.1.4 Literatur 4.....	7
2.1.5 Literatur 5.....	7
2.1.6 Literatur 6.....	7
2.1.7 Kelebihan Alat Stop Kontak Cerdas Berbasis IoT	8
2.2 <i>NodeMCU</i> ESP8266	8
2.3 Module RTC DS321	13
2.4 Relay	13
2.5 Arduino IDE.....	14
2.6 Stop Kontak.....	15

2.7	<i>Software</i>	15
2.8	<i>Hardware</i>	16
BAB III METODOLOGI PENELITIAN.....		17
3.1	Bahan Penelitian.....	17
3.2	Alat Penelitian.....	17
3.3	Kerangka Penelitian.....	18
3.4	Tahap Penelitian.....	19
3.5	Metode Pengumpulan Data.....	20
3.6	Rancangan Sistem.....	20
3.7	Perancangan Rangkaian.....	20
3.8	<i>Flowchart</i>	22
3.9	Rangkaian Skematik Alat.....	23
3.10	Desain Alat.....	23
3.11	Implementasi.....	24
3.12	Implementasi Perangkat Lunak.....	25
3.13	Implementasi Perangkat Keras.....	26
3.14	Pengelolaan Data.....	27
3.15	Analisis Hasil.....	28
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		29
4.1	Hasil Penelitian.....	29
4.2	Perancangan <i>Software</i> dan Mikrokontroler.....	29
	4.2.2 Program Arduino IDE.....	30
	4.2.3 Port yang Digunakan.....	31
	4.2.4 Program Alat.....	32
	4.2.5 Pengujian.....	36
	4.2.6 Cara Kerja Alat.....	38
4.3	Pembahasan Hasil Pengujian.....	39
BAB V KESIMPULAN.....		40
5.1	Kesimpulan.....	40
5.2	Saran.....	40
DAFTAR PUSTAKA.....		41
LAMPIRAN.....		43

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	5
Tabel 3.1 Bahan Penelitian.....	17
Tabel 4.1 Hasil Pengujian Perangkat terhadap <i>Wi-Fi</i>	36
Tabel 4.2 Percobaan Penjadwalan	37
Tabel 4.3 Pengujian Penjadwalan.....	38

DAFTAR GAMBAR

Gambar 2.1 Mapping <i>NodeMCU</i>	10
Gambar 2.2 <i>Real Time Clock</i>	13
Gambar 2.3 Relay dan Simbol.....	14
Gambar 2.4 Tampilan <i>Sketch</i> Arduino IDE.....	14
Gambar 2.5 Stop Kontak.....	15
Gambar 3.1 Metodologi Penelitian.....	18
Gambar 3.2 Tahap Penelitian.....	19
Gambar 3.3 <i>Flowchart</i>	22
Gambar 3.4 Skematik Alat	23
Gambar 3.5 Desain Alat Tampak Luar.....	23
Gambar 3.6 Desain Alat Tampak Dalam.....	24
Gambar 4.1 Bentuk Fisik Alat.....	29
Gambar 4.2 Program Alat.....	30
Gambar 4.3 Mikrokontroler <i>NodeMCU</i>	31
Gambar 4.4 Port yang digunakan	31
Gambar 4.5 Inisialisasi IoT <i>Cloud</i>	32
Gambar 4.6 <i>Set up</i> pin relay	33
Gambar 4.7 <i>Code</i> Program untuk <i>Wi-Fi</i> SSD	34
Gambar 4.8 <i>Update code</i> seluruh kondisi alat	34
Gambar 4.9 Tampilan Aplikasi.....	35

DAFTAR LAMPIRAN

Lampiran 1. Alat pengukur daya listrik yang digunakan	43
Lampiran 2. <i>Power Supply</i> yang digunakan.....	43
Lampiran 3. Rangkaian pada alat	44
Lampiran 4. Pengetesan alat pada pompa air	44
Lampiran 5. Pengetesan alat pada pompa air (lanjutan)	45
Lampiran 6. Penggunaan daya listrik pada alat.....	45
Lampiran 7. Penggunaan daya listrik pada alat (lanjutan).....	46
Lampiran 8. Penggunaan alat dengan Narasumber	46
Lampiran 9. Deskripsi pada Arduino IoT <i>Cloud</i>	47
Lampiran 10. Hasil Wawancara.....	48

ABSTRAK

Internet of Things (IoT) merupakan sebuah konsep teknologi yang bertujuan untuk menghubungkan objek-objek dalam rangka transfer data dan kendali melalui Internet. Contoh penerapan IoT adalah pengembangan soket listrik. Soket berbasis IoT disebut soket pintar karena soket ini dapat dikontrol dari jarak jauh melalui aplikasi Android atau kios sehingga nilai tegangan, arus, dan daya semu dapat diukur.

Penggunaan listrik yang berlebihan dapat mengakibatkan pemborosan listrik. Karena daya listrik terbatas, pemborosan listrik harus dicegah. Pencegahan kelistrikan dapat dilakukan dengan menggunakan energi listrik bila diperlukan. Cara ini merupakan solusi penghematan daya. Listrik rumah tangga mempunyai proporsi yang cukup tinggi yaitu mencapai 48,39%. Selain itu, masyarakat sering kali lupa menyalakan perangkat elektronik saat tidak digunakan lagi. Perangkat elektronik yang terhubung ke stop kontak tersebut dapat memboroskan daya karena daya terus mengalir meski tidak digunakan.

Pada penelitian ini pengguna dapat mengontrol penggunaan listrik dengan aplikasi *smartphone*. Untuk menyalakan lampu maka aplikasi pada *smartphone* akan mengirimkan perintah sesuai jadwal yang ditetapkan pengguna ke Relay dan Mikrokontroler sehingga lampu akan menyala. Setelah dilakukan pengujian jadwal dan perangkat *Wi-Fi*, lampu dapat menyala dengan jarak kurang lebih 20 meter. Dengan hasil tersebut maka perangkat *smartphone* dapat dijadikan kontrol stop kontak dengan jarak kurang dari 20 meter.

Kata Kunci : Stop kontak, *Internet of Things*, Listrik.

BAB I PENDAHULUAN

1.1 Latar Belakang

Teknologi yang sedang maju saat ini telah membawa peran penting dalam membantu dan memudahkan pekerjaan manusia. Namun, dampak negatif dari perkembangan ini adalah peningkatan penggunaan listrik. Isu pemborosan energi listrik semakin menjadi perhatian di lingkungan saat ini. Penyebab pemborosan ini adalah tingginya konsumsi listrik yang berlebihan. Menurut data dari Kementerian ESDM, sektor rumah tangga memiliki andil yang signifikan dalam penggunaan energi listrik, mencapai sekitar 48,38% (Santoso & Salim, 2019). Banyaknya peralatan elektronik rumah tangga yang menggunakan listrik menyebabkan penggunaan listrik semakin meningkat. Hal ini dapat mengakibatkan pemborosan dan peningkatan tagihan listrik karena keterbatasan pasokan listrik yang tersedia (Charun, 2017).

Stop kontak pintar berbasis *Internet of Things* (IoT) merupakan salah satu bentuk implementasi teknologi IoT di rumah tangga. Dalam perkembangannya, IoT menjadi semakin populer karena memberikan kemudahan dan kenyamanan bagi pengguna dalam mengontrol perangkat elektronik di rumah secara jarak jauh. Salah satu keunggulan dari stop kontak pintar berbasis IoT adalah kemampuannya untuk dioperasikan melalui aplikasi pada *smartphone* atau tablet¹. Dengan aplikasi ini, pengguna dapat mengontrol stop kontak pintar dari jarak jauh, bahkan ketika mereka tidak berada di rumah. Selain itu, stop kontak pintar juga dapat dikendalikan melalui suara menggunakan asisten virtual seperti *Google Assistant* atau *Amazon Alexa*.

Dalam pengembangan stop kontak pintar berbasis IoT, ada beberapa alasan mengapa teknologi ini menjadi semakin populer. Pertama, stop kontak pintar dapat membantu menghemat energi dengan mematikan perangkat listrik yang tidak digunakan. Kedua, stop kontak pintar dapat digunakan untuk mengontrol perangkat listrik dari jarak jauh, sehingga pengguna lebih mudah dalam mengatur perangkat listrik di rumah. Ketiga, stop kontak pintar dapat membantu pengguna untuk meningkatkan keamanan rumah dengan mengontrol

perangkat listrik dari jarak jauh. Dengan demikian, rancang bangun stop kontak pintar berbasis IoT menjadi solusi yang efektif dan efisien bagi pengguna dalam mengontrol perangkat listrik di rumah secara jarak jauh.

Untuk menghindari pemborosan listrik, diperlukan efisiensi dalam penggunaan energi listrik dengan hanya menggunakan listrik ketika diperlukan (Aripiyanto & Tukino, 2018). Pemborosan listrik sering terjadi secara tidak sengaja karena kelalaian masyarakat, seperti meninggalkan perangkat elektronik menyala saat tidak digunakan, seperti televisi, kipas, dan sebagainya. Perangkat elektronik yang tetap terhubung dengan stop kontak saat tidak digunakan dapat menimbulkan risiko korsleting listrik. Selain risiko korsleting listrik, dampak lainnya adalah kemungkinan kerusakan pada peralatan elektronik tersebut.

Perkembangan *Internet of Things* (IoT) di seluruh dunia, termasuk Indonesia, sangat pesat. IoT melibatkan penghubungan objek fisik seperti stop kontak dan televisi pintar dengan internet. IoT bekerja dengan menghubungkan perangkat elektronik ke aplikasi yang telah diprogram khusus. Melalui penggunaan sensor, RTC, *NodeMCU*, dan Arduino, informasi dapat diambil dan diubah menjadi tindakan sesuai dengan perintah dari pengguna atau pemilik⁴.

Untuk merancang dan membangun stop kontak cerdas berbasis IoT, dibutuhkan beberapa komponen utama seperti *NodeMCU* ESP8266, Relay, dan RTC DS3231. *NodeMCU* ESP8266 akan berperan sebagai otak yang mengatur dan mengendalikan seluruh fungsi dari stop kontak tersebut. Relay digunakan untuk mengalirkan dan mengendalikan arus listrik yang masuk ke stop kontak. Relay ini akan dihubungkan dengan perangkat elektronik yang akan diatur, seperti lampu atau alat lainnya yang terhubung ke stop kontak. RTC DS3231 akan bertugas sebagai pengukur waktu yang akan digunakan dalam mengatur jadwal penyaluran arus listrik. Dengan adanya RTC, stop kontak cerdas dapat diatur untuk menyala atau mati pada waktu yang sudah ditentukan sebelumnya.

1.2 Rumusan Masalah

Bedasarkan latar belakang pada penelitian ini dapat dirumuskan masalah yaitu :

1. Bagaimana cara mengembangkan stop kontak cerdas ke jaringan IoT?
2. Apa saja fitur dan fungsi yang dimiliki oleh stop kontak cerdas untuk memudahkan penggunaan perangkat listrik?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut :

1. Merancang dan membangun sistem stop kontak pintar berbasis *Internet of Things* yang dapat dikontrol melalui jaringan internet.
2. Mengembangkan sistem stop kontak pintar dengan memaksimalkan kerja Relay.
3. Memberikan alternatif solusi bagi pengguna agar dapat lebih efektif dan efisien dalam penghematan energi dan penggunaan listrik.

1.4 Batasan Masalah

Batasan masalah pada penelitian ini adalah :

1. Fokus pembahasan hanya pada rancang bangun dan implementasi stop kontak berbasis IoT dengan kontrol jarak jauh melalui aplikasi di *smartphone*.
2. Penelitian ini hanya membahas rancang bangun perangkat keras dari stop kontak serta integrasi antara *software* dan *hardware* yang dibutuhkan.
3. Penelitian ini tidak membahas mengenai penggunaan atau penerapan stop kontak berbasis IoT yang tepat dan optimal pada berbagai aplikasi praktis.

1.5 Manfaat Penelitian

Adapun manfaat penelitian yang akan didapat sebagai berikut :

1. Stop kontak berbasis *Internet of Things* dapat memungkinkan pengguna untuk memantau dan mengontrol penggunaan daya dari jarak jauh dan pengaturan otomatis, yang dapat mengurangi konsumsi listrik dan menghemat biaya listrik.

2. Dengan penyediaan kontrol jarak jauh, pengguna dapat mengontrol stop kontak tanpa harus terikat pada satu tempat atau rumah dan dapat melakukan kontrol semudah menggerakkan jari pada aplikasi yang diunduh.
3. Meningkatkan gaya hidup digital: Kehadiran *Internet of Things* dan aplikasi yang menyertainya dapat membantu pengguna untuk beradaptasi dengan kehidupan di era digital dengan mudah dan mengatasi perubahan teknologi.

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 memamatkannya 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

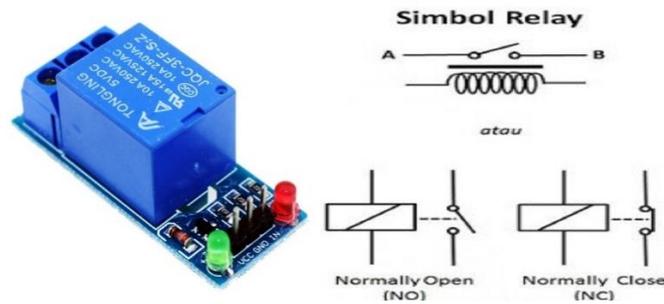
Modul RTC DS3231 adalah sebuah modul yang digunakan untuk mengukur waktu secara *real-time* (RTC) dan juga dilengkapi dengan pengukur suhu. Modul ini menyediakan semua fitur RTC dan suhu dalam satu paket. Contohnya dapat dilihat pada Gambar 2.2.



Gambar 2.2 Real Time Clock

2.4 Relay

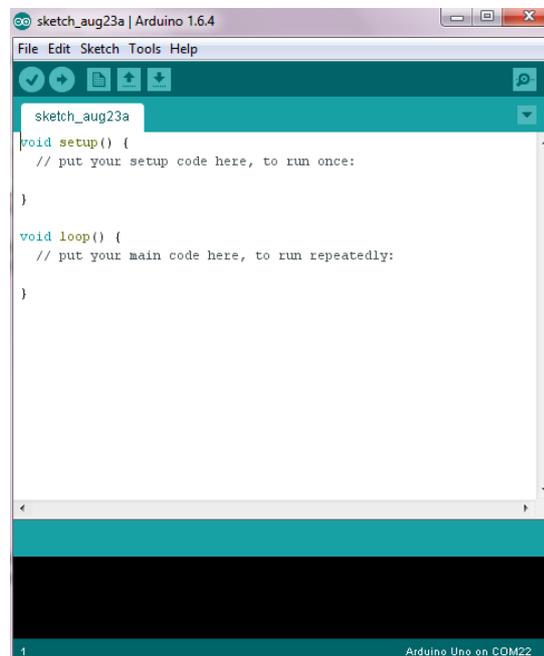
Relay adalah komponen elektronik yang berfungsi sebagai saklar atau *switch* listrik dan dioperasikan menggunakan listrik. Relay juga sering disebut sebagai komponen elektromekanik, karena terdiri dari dua bagian utama yaitu coil atau elektromagnet dan kontak saklar atau mekanik. Dalam komponen relay, 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.

BAB III METODOLOGI PENELITIAN

3.1 Bahan Penelitian

Sebelum melakukan pembuatan alat system, penulis perlu menyiapkan bahan-bahan yang akan digunakan dalam penelitian seperti dibawah ini :

Tabel 3.1 Bahan Penelitian

No.	Nama Bahan	Spesifikasi	Fungsi	Jumlah
1	<i>NodeMCU</i>	ESP8266	Membuat <i>prototype</i>	1
2	Relay		Penggerak kontak saklar	1
3	Stop Kontak		Muara penghubung arus listrik	1
4	Module RTC	DS321	Penghitung waktu digital	1
5	Kabel Jumper	<i>Male Female</i>	Penghubung seluruh komponen	20
6	<i>BreadBoard</i>		Sebagai papan sirkuit	1
7	<i>Power Supply</i>	5V	Mengatur daya tegangan <i>output</i> arus listrik	1

3.2 Alat Penelitian

Alat penelitian terdiri dari perangkat keras yang di persiapkan untuk membuat sistema perancangan yang akan penulis gunakan.

1. *NodeMCU*

Komponen ini sangatlah penting dan paling utama yang dibutuhkan dalam pembuatan alat ini karena komponen ini dibutuhkan sebagai alat untuk memasukkan program.

2. Laptop

Dibutuhkan untuk membuat program dan menulis laporan untuk pembuatan alat ini.

3. *Smartphone*

Digunakan untuk menguji alat melalui aplikasi Arduino IoT Cloud.

3.3 Kerangka Penelitian

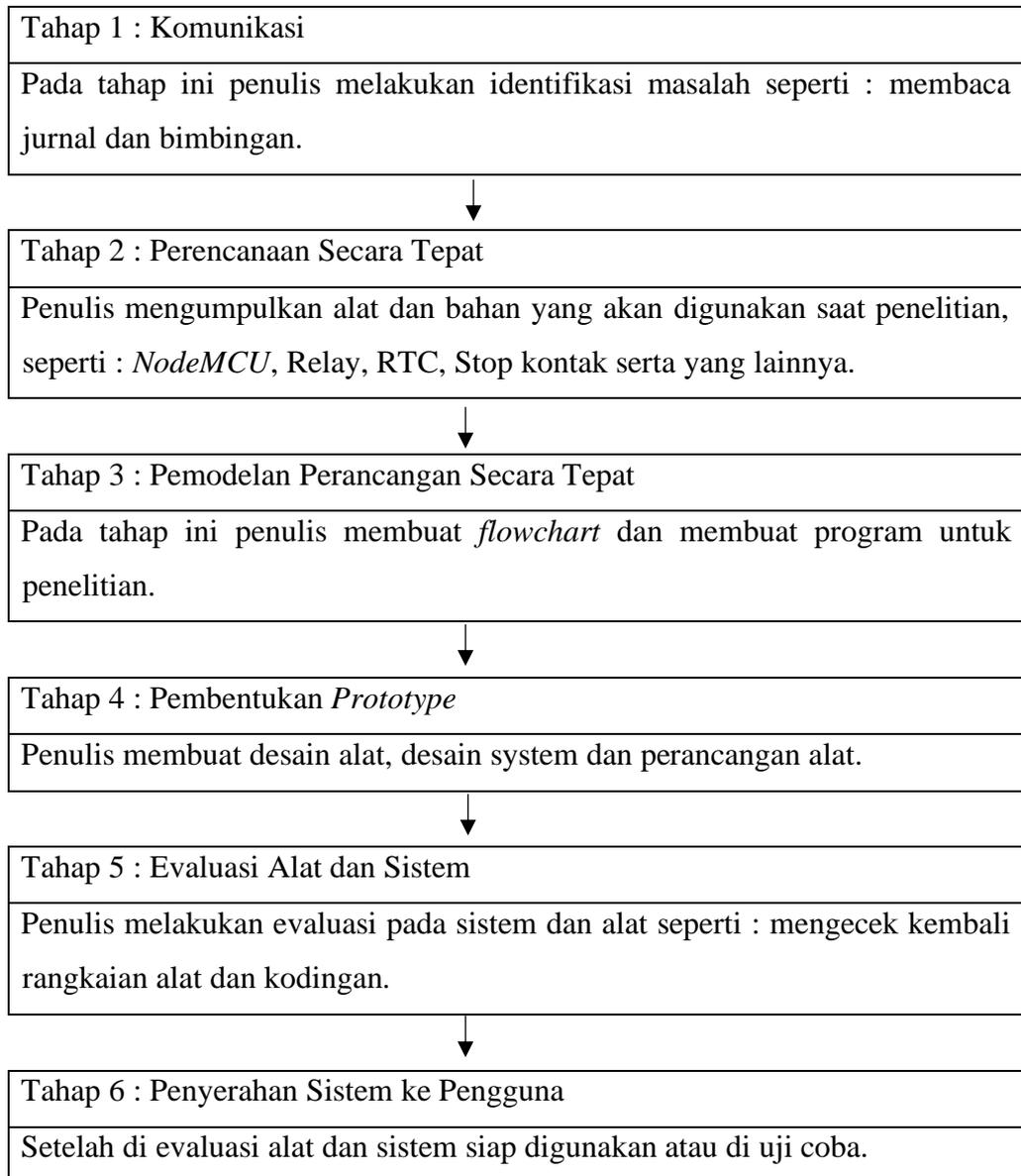
Dibawah ini merupakan kerangka penelitian penulis dalam melakukan penelitian:

1. Alasan Penelitian Dilakukan		
<ul style="list-style-type: none"> • Memberikan alternatif solusi bagi masyarakat agar dapat lebih efektif dan efisien dalam penghematan listrik dari jarak jauh dengan aplikasi <i>smarthphone</i>. 		
2. Proses Penelitian		
Kajian Literatur	Pengumpulan Data	Identifikasi Masalah
<ul style="list-style-type: none"> • Jurnal • Artikel Ilmiah 	<ul style="list-style-type: none"> • Wawancara 	<ul style="list-style-type: none"> • Penelitian Terdahulu • Perumusan Masalah
3. Untuk Memenuhi Kebutuhan		
<p>Penelitian ini dilakukan untuk membantu pengguna agar dapat mengontrol dan memonitor listrik yang digunakan pada perangkat yang terhubung dengan stop kontak tersebut melalui aplikasi <i>smartphone</i> dan perangkat lainnya yang terhubung dengan internet.</p>		
4. Hasil Penelitian		
Stop kontak Pintar berbasis IoT		

Gambar 3.1 Metodologi Penelitian

3.4 Tahap Penelitian

Dibawah ini merupakan tahap penelitian penulis dalam melakukan penelitian, dapat dilihat dibawah ini :



Gambar 3.2 Tahap Penelitian

3.5 Metode Pengumpulan Data

1. Studi Pustaka

Penulis mengumpulkan informasi melalui membaca jurnal yang berkaitan dengan masalah penelitian.

2. Metode Eksperimen

Penulis melakukan metode ini dengan cara merancang alat dan mengujinya secara langsung untuk mengecek dan mengetahui sistem kerja dari alat.

3.6 Rancangan Sistem

Dalam pembuatan alat sistem, perancangan sistem sangatlah penting karena merupakan langkah dasar sebelum mengimplementasikan alat. Sistem biasanya dirancang oleh seorang programmer atau engineer, yang akan menentukan keberhasilan alat yang akan dibuat. Jika seluruh tahapan desain dijalankan dengan baik dan kriteria yang telah ditentukan terpenuhi, mulai dari pembuatan diagram dan diagram alur hingga pemilihan komponen alat, hasilnya akan konsisten dengan deskripsi awal yang diperlukan.

3.7 Perancangan Rangkaian

Rancangan rangkaian *smart plug* berbasis *Internet of Things* (IoT) bertujuan agar pengguna dapat mengontrol smart plug dari jarak jauh menggunakan perangkat pintar seperti *smartphone* atau tablet. Rangkaian ini terdiri dari beberapa komponen utama antara lain :

1. Mikrokontroler : Mikrokontroler berfungsi sebagai pusat operasi rangkaian yang akan mengatur pengoperasian smart konektor berdasarkan instruksi yang diterima dan menggunakan jaringan IoT. Beberapa pilihan mikrokontroler yang bisa digunakan adalah Arduino atau Raspberry Pi.
2. Modul *Wi-Fi* terhubung ke Internet melalui modul *Wi-Fi*. Beberapa pilihan modul *Wi-Fi* yang dapat digunakan adalah ESP8266 atau ESP32. Modul *Wi-Fi* ini terhubung dengan mikrokontroler dan mengirimkan data serta perintah dari aplikasi IoT ke *smart socket*.
3. Relay mengontrol aliran listrik ke stop kontak. Saat relay diaktifkan, stop kontak menyala dan listrik mengalir. Sebaliknya, ketika relay dimatikan,

stop kontak juga ikut mati dan aliran arus terputus. Relay ini dihubungkan ke mikrokontroler melalui pin *output*.

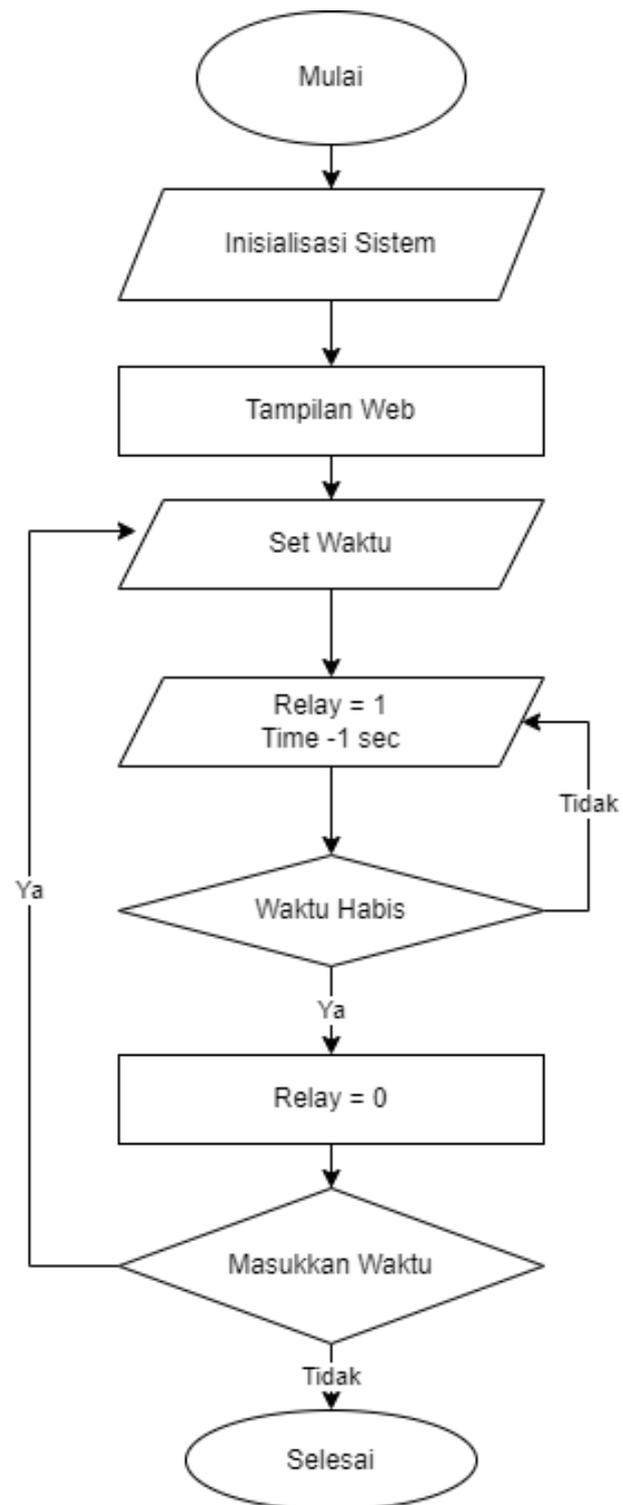
4. Aplikasi IoT akan menghubungkan stop kontak pintar dengan internet dan memungkinkan pengguna menggunakan perangkat pintar seperti ponsel cerdas atau tablet untuk mengontrol stop kontak pintar tersebut dari jarak jauh. Beberapa platform IoT yang bisa digunakan untuk mengembangkan aplikasi ini adalah *Blynk*, *ArduinoUno Thingspeak*, atau *IFTTT*. Aplikasi ini akan terhubung dengan modul *Wi-Fi* yang ada pada stop kontak pintar dan akan mengirimkan perintah serta menerima data dari stop kontak pintar tersebut.

Dalam merancang rangkaian stop kontak pintar berbasis IoT, beberapa hal yang perlu diperhatikan adalah keamanan jaringan, keandalan koneksi internet, dan kemampuan mikrokontroler untuk menangani permintaan dari aplikasi IoT. Selain itu, perlu juga memperhatikan sumber daya listrik yang digunakan, agar stop kontak pintar tidak mengalami *overheat* atau masalah listrik lainnya.

3.8 Flowchart

Pada tahap penelitian ini penulis membuat dan menjelaskan rangkaian cara kerja dari alat:

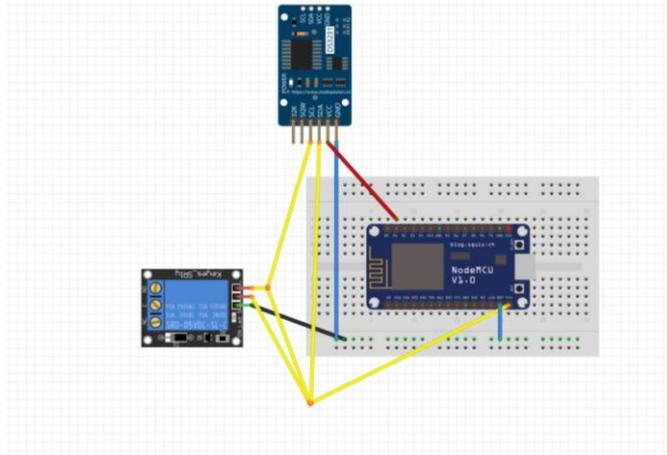
Pertama-tama alat hidup saat terdapat energi listrik, kemudian sistem akan diinisialisasi dan terhubung dengan internet. Pengguna akan memasukkan waktu yang diinginkan pada alat untuk bekerja melalui web. Waktu yang dimasukkan oleh pengguna pada web akan masuk ke alat (*NodeMCU*) dan akan diolah dengan rumus yang telah dimasukkan pada kode web. Jika waktu yang dimasukkan tidak sesuai atau belum tepat, alat akan melakukan perhitungan ulang. Jika waktu telah habis, maka alat atau relay akan bekerja dengan menghidupkan atau mematikan alat maupun listrik. Kemudian Pengguna dapat memasukkan lagi waktu untuk dimasukkan ke alat, jika tidak alat tidak akan berjalan. Lihat gambar rangkaian 3.3.



Gambar 3.3 *Flowchart*

3.9 Rangkaian Skematik Alat

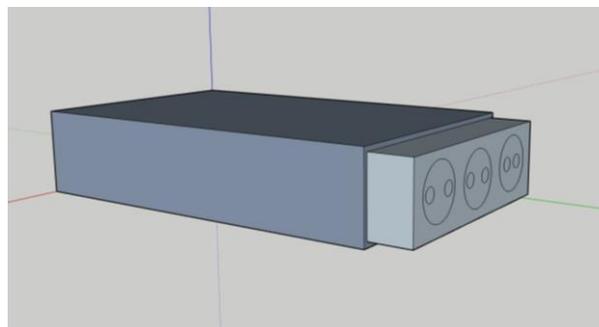
Rangkaian skematik alat di rancang menggunakan *software Fritzing* dalam bentuk gambaran keseluruhan untuk selanjutnya di implementasikan dalam bentuk nyata. Berikut rangkaian skematik dari keseluruhan alat yang akan digunakan seperti yang terlihat pada Gambar 3.4.



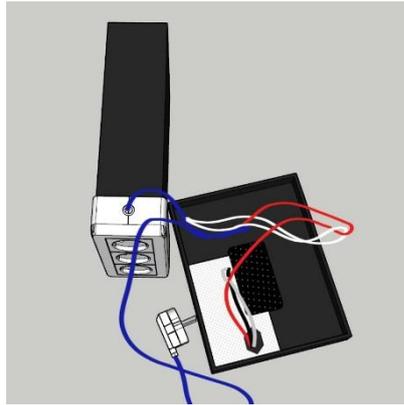
Gambar 3.4 Skematik Alat

3.10 Desain Alat

Desain alat dibuat untuk mendapat gambaran 3D dari alat yang akandirancang dengan tujuan sebagai panduan dalam pembuatan alat. Desain alat ini dibuat dengan *software SketchUp* dalam bentuk 3D modeling dengan rancangan sedemikian rupa untuk mendapat gambaran nyata. Dibawah ini adalah contoh desain alat yang akan penulis gunakan dalam penelitian. Seperti terlihat pada Gambar 3.5 dan 3.6.



Gambar 3.5 Desain Alat Tampak Luar



Gambar 3.6 Desain Alat Tampak Dalam

3.11 Implementasi

Implementasi perancangan rangkaian stop kontak pintar berbasis *Internet of Things* (IoT) melibatkan beberapa tahap, antara lain:

1. Perakitan *hardware*

Setelah merancang skema rangkaian, selanjutnya adalah merakit *hardware* sesuai dengan skema rangkaian yang sudah dibuat. Pada tahap ini, perlu diperhatikan koneksi antara setiap komponen dan pengaturan pin pada mikrokontroler. Pastikan juga bahwa pemasangan komponen dan kabel dilakukan dengan benar agar tidak terjadi kesalahan koneksi atau korsleting.

2. Pengaturan *software*

Setelah *hardware* dirakit, selanjutnya adalah memprogram mikrokontroler dengan menggunakan bahasa pemrograman yang mendukung platform IoT. Contohnya, dengan menggunakan platform *Blynk*, dapat dilakukan pemrograman dengan bahasa pemrograman C atau C++. Pada tahap ini, perlu memperhatikan koneksi antara mikrokontroler dan modul *Wi-Fi*, sehingga aplikasi IoT dapat berkomunikasi dengan stop kontak pintar.

3. Pengujian

Setelah *hardware* dan *software* sudah disiapkan, selanjutnya adalah melakukan pengujian untuk memastikan bahwa rangkaian berfungsi dengan baik. Pada tahap ini, perlu melakukan pengujian terhadap masing-masing komponen, seperti modul *Wi-Fi*, mikrokontroler, relay, dan sensor suhu, untuk memastikan semua komponen dapat berjalan dengan baik. Setelah itu, lakukan pengujian terhadap aplikasi IoT, untuk memastikan

aplikasi dapat berkomunikasi dengan stop kontak pintar dan mengontrol pengoperasiannya.

4. Integrasi

Setelah pengujian berhasil dilakukan, selanjutnya adalah mengintegrasikan rangkaian ke dalam sistem yang lebih besar. Misalnya, dapat dilakukan integrasi dengan sistem *smart home* yang sudah ada, atau dengan sistem monitoring energi yang dapat memantau penggunaan stop kontak pintar secara real-time.

Dalam mengimplementasikan perancangan rangkaian stop kontak pintar berbasis IoT, perlu memperhatikan faktor keamanan jaringan dan privasi pengguna, karena rangkaian ini terhubung dengan jaringan internet. Selain itu, perlu memastikan bahwa rangkaian beroperasi dengan baik dan tidak mengalami masalah yang dapat membahayakan pengguna atau sistem yang terhubung.

3.12 Implementasi Perangkat Lunak

Untuk mengimplementasikan perangkat lunak perancangan stop kontak pintar berbasis *Internet of Things*, berikut adalah langkah-langkah yang perlu dilakukan:

1. Identifikasi kebutuhan pengguna

Perancangan stop kontak pintar harus mengikuti kebutuhan pengguna. Pertimbangkan apa yang pengguna butuhkan dari stop kontak pintar. Apa yang diharapkan dari stop kontak pintar? Bagaimana cara memudahkan pengguna dalam mengoperasikan stop kontak pintar? Apakah ada fungsi tambahan yang diinginkan?

2. Merancang sistem *Internet of Things*

Rancang sistem *Internet of Things* (IoT) yang cocok untuk stop kontak pintar. Sistem ini harus bisa menghubungkan stop kontak pintar dengan internet. Pertimbangkan berbagai inovasi IoT yang bisa digunakan, misalnya platform *Cloud* yang bisa mencatat penggunaan energi, menyimpan profil pengguna, dan memonitor status stop kontak pintar.

3. Pilih *hardware* yang tepat

Pilih *hardware* (perangkat keras) yang cocok untuk stop kontak pintar. Pertimbangkan faktor seperti daya listrik, jangkauan *Wi-Fi*, dan kemampuan untuk terhubung dengan sistem IoT yang sudah dirancang sebelumnya.

4. Buat aplikasi untuk pengguna

Buat aplikasi yang mudah digunakan untuk mengontrol stop kontak pintar. Aplikasi ini harus sesuai dengan kebutuhan pengguna dan dapat memudahkan pengguna dalam mengaktifkan atau menonaktifkan stop kontak pintar.

5. Uji coba dan evaluasi

Setelah perancangan selesai, uji coba sistem dan aplikasi dengan lingkungan yang sesuai. Evaluasi sistem dan aplikasi untuk mengetahui apakah dapat memenuhi kebutuhan pengguna. Perbaiki jika ada kekurangan atau masukan dari pengguna.

6. Implementasi

Setelah semua perbaikan dilakukan, implementasikan perangkat lunak perancangan stop kontak pintar berbasis *Internet of Things* yang sudah dirancang dan diujicoba. Gunakan perangkat pada lingkungan nyata dan perbaiki jika ada masalah yang muncul selama penggunaan.

3.13 Implementasi Perangkat Keras

Implementasi perangkat keras untuk perancangan stop kontak pintar berbasis *Internet of Things* (IoT) dapat dilakukan dengan langkah-langkah berikut:

1. Memilih komponen

Memilih komponen seperti ESP8266, *solid state* relay dan beberapa resistor dan kapasitor.

2. Membuat *prototype*

Membuat prototipe dengan cara mengumpulkan semua komponen dan memperbaiki mereka pada PCB. Sambungkan perangkat keras dengan komputer dan Cobalah memprogram ESP8266 untuk menghubungkan ke *Wi-Fi*.

3. Kustomisasi perangkat keras

Setelah pembuatan *prototype*, pastikan perangkat keras dapat dikustomisasi sesuai kebutuhan. Misalnya, sensor arus dapat disesuaikan untuk mendeteksi apabila daya listrik melebihi batas yang diatur sebelumnya.

4. Uji coba

Uji coba stop kontak pintar dan pastikan bahwa mekanisme operasinya, seperti pemutus listrik dan pengontrol waktu, berjalan dengan baik.

5. Integrasi *software*

Akhirnya, hubungkan perangkat keras dengan *online dashboard* seperti Arduino Uno dan *Blynk*. Hal ini akan memungkinkan akses jarak jauh untuk perangkat stop kontak pintar dan juga memantau penggunaan energi.

Dengan melakukan implementasi perangkat keras pada perancangan stop kontak pintar, rumah atau tempat kerja dapat dikendalikan dengan mudah dan dapat diatur sesuai kebutuhan pengguna. Hal ini dapat menghemat energi dan meningkatkan efisiensi dalam penggunaan energi.

3.14 Pengelolaan Data

Dalam pengelolaan data perancangan stop kontak pintar berbasis *Internet of Things* (IoT), terdapat beberapa tahapan yang harus dilakukan. Berikut adalah tahapannya:

1. Identifikasi Kebutuhan

Langkah pertama adalah mengidentifikasi kebutuhan pengguna dalam menggunakan stop kontak pintar IoT. Dalam hal ini, perlu dipertimbangkan faktor-faktor seperti keamanan, kemudahan penggunaan, dan ketersediaan teknologi.

2. Rancangan Sistem

Setelah kebutuhan pengguna telah diidentifikasi, tahap selanjutnya adalah merancang sistem stop kontak pintar yang terhubung dengan IoT. Rancangan sistem ini meliputi perangkat keras dan perangkat lunak yang dibutuhkan untuk menghubungkan stop kontak pintar dengan jaringan

komunikasi. Rancangan ini juga perlu mempertimbangkan aspek keamanan dan privasi data.

3. Implementasi Sistem

Setelah rancangan sistem selesai, tahap selanjutnya adalah mengimplementasikan sistem stop kontak pintar IoT tersebut. Implementasi ini meliputi pemasangan perangkat keras dan instalasi perangkat lunak yang dibutuhkan.

4. Uji Coba Sistem

Setelah sistem stop kontak pintar IoT selesai diimplementasikan, tahap selanjutnya adalah melakukan uji coba sistem. Uji coba ini bertujuan untuk memastikan bahwa sistem berfungsi dengan baik dan dapat terhubung dengan jaringan komunikasi.

5. Monitoring dan Pemeliharaan Sistem

Setelah sistem stop kontak pintar IoT diuji coba dan berjalan dengan baik, tahap selanjutnya adalah memantau dan memelihara sistem secara berkala. Monitoring dan pemeliharaan ini bertujuan untuk memastikan sistem tetap berjalan dengan baik dan terhubung dengan jaringan komunikasi dengan stabil.

Dengan melakukan tahapan pengolahan data perancangan stop kontak pintar berbasis *Internet of Things* dengan baik, diharapkan dapat menghasilkan sistem stop kontak pintar yang aman, mudah digunakan, dan dapat terhubung dengan jaringan komunikasi dengan stabil.

3.15 Analisis Hasil

Analisis hasil ini dilakukan berdasarkan data yang telah dikumpulkan dari pengujian secara langsung yang kemudian di analisis untuk mendapatkan hasil kesimpulan.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Pada bab ini penulis akan menjelaskan tentang hasil uji coba alat yang telah dirancang beserta pembahasan untuk mengetahui hasil dari rancangan alat dan implementasi yang dilakukan apakah sudah sesuai dengan data yang dibutuhkan. Langkah awal yang dilakukan yaitu dengan melakukan pengujian dari beberapa komponen, agar jika terjadi kesalahan akan lebih mudah untuk mengetahuinya.

Berikut adalah hasil jadi alat stop kontak pintar berbasis IoT yang akan digunakan oleh peneliti dalam penelitian ini. Dimana stop kontak pintar tersebut sudah diberikan komponen yang dibutuhkan dalam penelitian



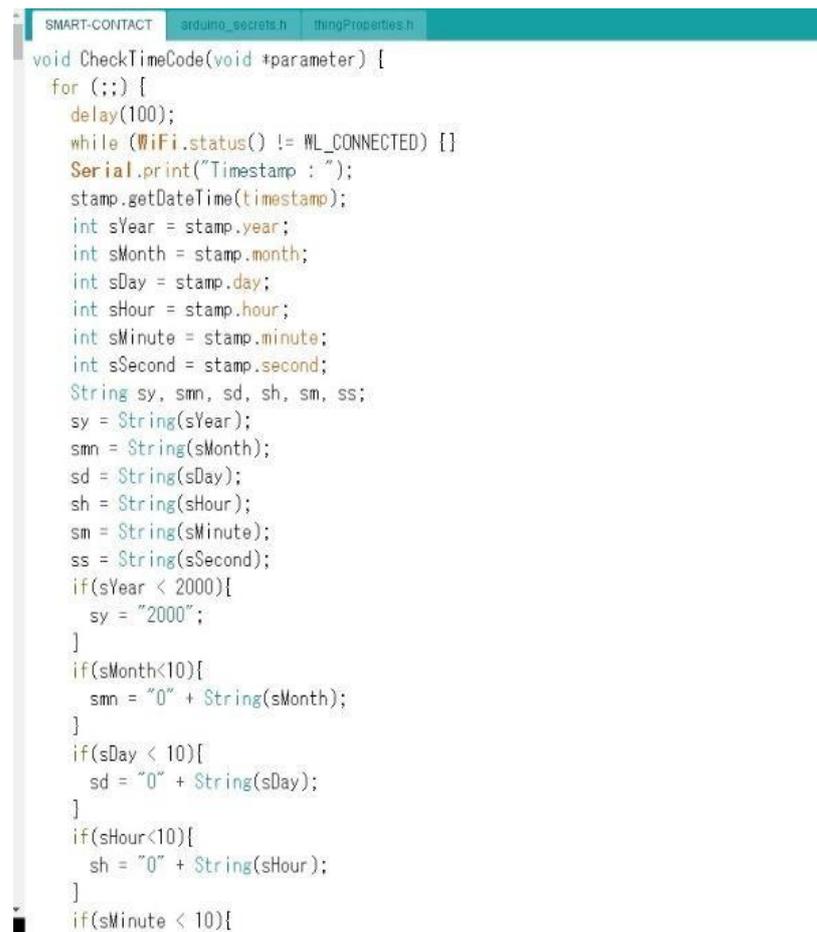
Gambar 4.1 Bentuk Fisik Alat

4.2 Perancangan *Software* dan Mikrokontroler

Perancangan *software* merupakan suatu prosedur pemberian perintah dan mutlak harus dilakukan, karena tanpa adanya prosedur ini maka alat yang akan dibuat hanyalah benda mati yang tidak dapat melakukan pekerjaan apa pun. Dalam hal ini perancangan *software* sangatlah erat hubungannya dengan sub bab *flowchart* karena pada dasarnya perintah yang akan dikerjakan harus mengikuti alur pada *flowchart* tersebut agar alat yang dibuat dapat bekerja sesuai dengan yang diinginkan. Jadi pada bab ini penulis akan melakukan perancangan sebuah sistem

pada mikrokontroler (NudeMCU) menggunakan arduino IDE (*Intergrated Development Enviroment*).

Berikut adalah *code* program untuk alat stop kontak pintar berbasis IoT,*code* ini adalah bagian dari desain dan implementasi perangkat lunak yang memungkinkan alat stop kontak dapat terhubung dan berkomunikasi dengan jaringan *Wi-Fi* yang nantinya akan tersambung dengan perangkat seperti *smartphone*.



```

SMART-CONTACT  arduino_secrets.h  thingProperties.h
void CheckTimeCode(void *parameter) {
  for (;;) {
    delay(100);
    while (WiFi.status() != WL_CONNECTED) {}
    Serial.print("Timestamp : ");
    stamp.getDateTime(timestamp);
    int sYear = stamp.year;
    int sMonth = stamp.month;
    int sDay = stamp.day;
    int sHour = stamp.hour;
    int sMinute = stamp.minute;
    int sSecond = stamp.second;
    String sy, smn, sd, sh, sm, ss;
    sy = String(sYear);
    smn = String(sMonth);
    sd = String(sDay);
    sh = String(sHour);
    sm = String(sMinute);
    ss = String(sSecond);
    if(sYear < 2000){
      sy = "2000";
    }
    if(sMonth<10){
      smn = "0" + String(sMonth);
    }
    if(sDay < 10){
      sd = "0" + String(sDay);
    }
    if(sHour<10){
      sh = "0" + String(sHour);
    }
    if(sMinute < 10){

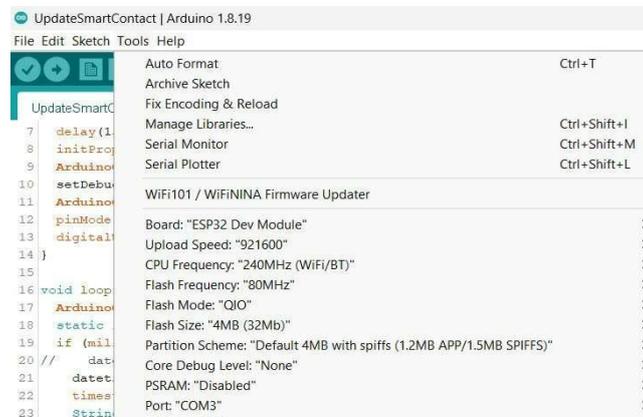
```

Gambar 4.2 Program Alat

4.2.2 Program Arduino IDE

Aplikasi yang digunakan penulis sebagai Mikrokontroler adalah *NodeMCU* ESP32. Aplikasi ini penulis gunakan untuk membuat program yang berisi perintah untuk menerima data yang dikirimkan oleh *smartphone* lalu mengeksekusi perintah tersebut baik secara mekanik, audio, atau pun dengan visualisasi. Hal ini yang dilakukan berupa inialisasi dan penulisan program.

Model mikrokontroler yang digunakan pada penelitian ini penulis menggunakan *NodeMCU* ESP32 karena program yang digunakan akan terhubung ke internet.

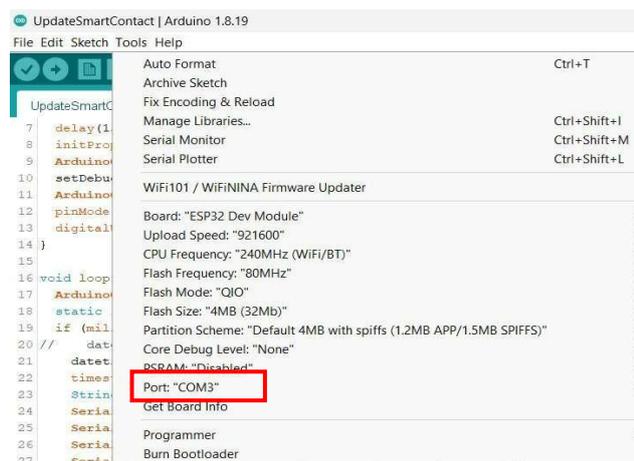


Gambar 4.3 Mikrokontroler *NodeMCU*

4.2.3 Port yang Digunakan

Code port Arduino ini berfungsi sebagai pengontrol utama untuk mengatur perilaku stop kontak. *Code* ini bertanggung jawab untuk menerjemahkan sinyal atau instruksi yang diterima dari aplikasi atau sistem lainnya ke dalam aksi konkret pada perangkat keras stop kontak.

Pada penelitian ini penulis menggunakan port COM3 sebagai port dalam programnya.



Gambar 4.4 Port yang digunakan

4.2.4 Program Alat

1. Inisialisasi IoT *Cloud*

Pada foto kode inisialisasi IoT *Cloud* untuk alat stop kontak pintar berbasis IoT, terdapat beberapa langkah kunci yang perlu dilakukan. Pertama, dilakukan impor library yang dibutuhkan untuk komunikasi dengan *Cloud* IoT dan pengendalian alat stop kontak pintar. Selanjutnya, harus diatur koneksi ke jaringan *Wi-Fi* yang digunakan. Setelah itu, perlu membuat objek klien IoT untuk mengatur dan mengelola komunikasi antara perangkat dan *Cloud* IoT. Langkah terakhir adalah menghubungkan alat stop kontak pintar ke *Cloud* IoT dengan menggunakan objek klien yang telah dibuat sebelumnya.

Inisialisasi ini merupakan langkah awal yang penting untuk memastikan alat stop kontak pintar dapat terhubung dan berkomunikasi dengan *Cloud* IoT. Dengan melakukan inisialisasi yang benar, alat stop kontak pintar bisa mengirim dan menerima data dari *Cloud* IoT, memungkinkan pengguna untuk mengendalikan alat tersebut melalui aplikasi atau platform yang terhubung dengan *Cloud* IoT. Dalam proses ini, penting untuk memeriksa dokumentasi dan petunjuk penggunaan yang disediakan oleh *Cloud* IoT yang digunakan, karena setiap platform bisa memiliki perbedaan dalam langkah inisialisasinya.

```

SMART-CONTACT$ arduino_secrets.h thingProperties.h

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  |
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();

  xTaskCreate(CheckTimeCode, "Check Time", 10000, NULL, 0, &CheckTime);
}

void loop() {
  ArduinoCloud.update();
}

```

Gambar 4.5 Inisialisasi IoT *Cloud*

2. Set Up Pin Relay

Set up pin relay di Arduino untuk stop kontak pintar berbasis IoT merupakan langkah yang penting dalam menghubungkan stop kontak dengan jaringan internet. Pada Arduino relay digunakan sebagai saklar yang mengatur aliran listrik pada stop kontak. Dalam konfigurasi ini menggunakan kode untuk memprogram Arduino agar dapat mengendalikan relay.

```
5 void setup() {
6   Serial.begin(9600);
7   delay(1500);
8   initProperties();
9   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
10  setDebugMessageLevel(2);
11  ArduinoCloud.printDebugInfo();
12  pinMode(pin_relay, OUTPUT);
13  digitalWrite(pin_relay, HIGH);-|
14 }
```

Gambar 4.6 *Set up* pin relay

3. Wi-Fi SSID

Code Wi-Fi SSID pada Arduino untuk stop kontak pintar berbasis IoT penting untuk menghubungkan Arduino dengan jaringan *Wi-Fi*. Seiring koneksi berhasil terhubung, Arduino dapat menggunakan perangkat IoT, seperti aplikasi ponsel pintar, untuk mengontrol stop kontak secara eksternal melalui jaringan *Wi-Fi*. Dengan demikian, ketika Arduino terhubung ke jaringan *Wi-Fi* dengan menggunakan kode SSID tersebut, kita dapat menerima perintah dari aplikasi ponsel pintar untuk mengaktifkan atau mematikan stop kontak. Hal ini memungkinkan pengguna untuk mengontrol stop kontak pintar berbasis IoT dari jarak jauh melalui koneksi *Wi-Fi*, memberikan kenyamanan dan fleksibilitas dalam mengontrol aliran listrik.



```

UpdateSmartContact - arduino_secrets.h | Arduino 1.8.19
File Edit Sketch Tools Help
UpdateSmartContact arduino_secrets.h thingProperties.h
1 #define SECRET_SSID "Detector-Vibration"
2 #define SECRET_OPTIONAL_PASS "@DV2023_"
3 #define SECRET_DEVICE_KEY "L5QTNH5DTIJRJX8SSLXB"

```

Gambar 4.7 Code Program untuk *Wi-Fi* SSD

4. Update Code Alat

Update code alat di Arduino untuk stop kontak pintar berbasis IoT diperlukan ketika kita ingin mengubah atau memperbarui fungsi atau perintah yang terkait dengan pengontrolan stop kontak. Dalam *code update*, kita perlu memperhatikan bagaimana Arduino berkomunikasi dengan aplikasi melalui jaringan *Wi-Fi*, serta bagaimana perintah yang diterima akan diinterpretasikan untuk mengontrol stop kontak.

```

16 void loop() {
17   ArduinoCloud.update();
18   static int ms = millis();
19   if (millis() - ms >= 1000) {
20     // datetime = tsToDatetime();
21     datetime = "Off dalam 7 Jam, Schedule : " + String(schedule.isActive()) + " Relay : "+String(relay);
22     timestamp = ArduinoCloud.getLocalTime();
23     String sch = (schedule.isActive() == HIGH) ? "Aktif" : "Tidak Aktif";
24     Serial.print("Datetime \t: "); Serial.println(datetime);
25     Serial.print("Schedule \t: "); Serial.println(sch);
26     Serial.print("Relay \t: "); Serial.println(relay);
27     Serial.print("Timestamp \t: "); Serial.println(timestamp);
28     if (schedule.isActive() || relay) {
29       digitalWrite(pin_relay, LOW);
30     } else {
31       digitalWrite(pin_relay, HIGH);
32     }
33     ms = millis();
34   }
35 }
36

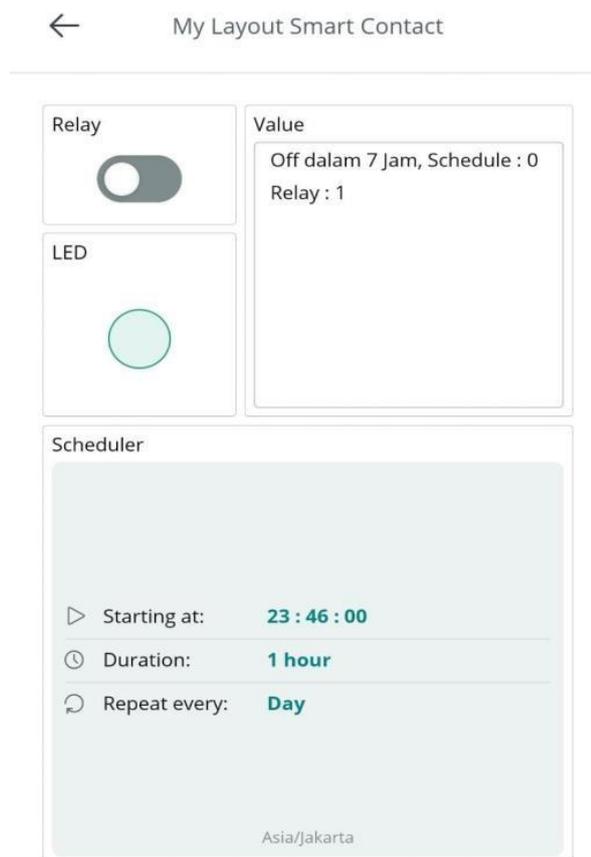
```

Gambar 4.8 Update code seluruh kondisi alat

5. Tampilan Aplikasi

Stop kontak pintar berbasis IoT dengan *NodeMCU*, relay dan *Wi-Fi* bekerja dengan cara menghubungkan *NodeMCU* ke jaringan *Wi-Fi* dan menggunakan relay sebagai saklar elektrik. *NodeMCU* berfungsi sebagai pengontrol utama yang menerima instruksi dari aplikasi melalui jaringan *Wi-Fi*, kemudian mengirimkan sinyal ke relay untuk mengatur aliran listrik pada stop kontak.

Pengguna dapat mengendalikan stop kontak pintar melalui aplikasi yang terhubung dengan *NodeMCU*, sehingga memungkinkan pengoperasian jarak jauh dan pengaturan waktu pengoperasian yang fleksibel. Gambar 4.9 dibawah menunjukkan hasil dari notifikasi web. Tingkat keberhasilan pengiriman data adalah 100%, hal ini ditandai dengan terkirimnya seluruh data yang dikirimkan oleh *NodeMCU* ESP8266.



Gambar 4.9 Tampilan Aplikasi

4.2.5 Pengujian

Pengujian alat adalah tahapan yang dilakukan secara langsung oleh penulis pada *prototype* stop kontak pintar, dilakukannya pengujian ini bertujuan untuk mengetahui tingkat keberhasilan alat/sistem dapat bekerja sesuai dengan harapan atau tidak.

1. Pengujian *Wi-Fi*

Pengujian dilakukan dengan menghubungkan perangkat stop kontak pintar IoT dengan *Wi-Fi* dan Google Firebase dan mengetahui pengaruh jarak perangkat ke hotspot *Wi-Fi* terhadap penerimaan dan pengiriman data.

Berikut ini merupakan tabel 1 menunjukkan hasil dari pengujian jarak *Wi-Fi*.

Tabel 4.1 Hasil Pengujian Perangkat terhadap *Wi-Fi*

No	Jarak Perangkat ke <i>Wi-Fi</i> (Meter)	Status	Keterangan
1	0-7	Terhubung	Penerimaan dan pengiriman data tanpa delay
2	7-12	Terhubung	Penerimaan dan pengiriman data terjadi delay 4 detik
3	12-16	Terhubung	Penerimaan dan pengiriman data terjadi delay 7 detik
4	16-20	Terhubung	Penerimaan dan pengiriman data terjadi delay 10
5	20	Tidak Terhubung	Penerimaan dan pengiriman data terhenti.

2. Relay

Tabel 4.2 Percobaan Penjadwalan

Percobaan	Penjadwalan	Status Hidup	Status Mati
1	17.00 - 17.01	Berhasil	Berhasil
2	17.04 - 17.05	Berhasil	Berhasil
3	17.07 - 17.08	Berhasil	Berhasil
4	17.11 - 17.12	Berhasil	Berhasil
5	17.16 - 17.19	Berhasil	Berhasil
6	17.21 - 17.22	Berhasil	Berhasil
7	17.30 - 17.34	Berhasil	Berhasil
8	17.36 - 17.37	Berhasil	Berhasil
9	17.45 - 17.50	Berhasil	Tidak Berhasil
10	18.02 - 18.09	Berhasil	Berhasil

Pada percobaan penjadwalan **Tabel 2**. Penulis melakukan percobaan dengan mengatur waktu untuk mati dan menyala dengan waktu yang sudah ditentukan, dari 10 kali percobaan sebanyak 9 kali alat berhasil bekerja dengan tingkat keberhasilan 95%.

3. Pengujian Jadwal

Pengujian jadwal pada aplikasi akan melibatkan pengujian fungsionalitas dan keadaan jadwal yang telah ditetapkan dalam aplikasi pengendali. Dalam pengujian ini penulis melakukan pengaturan jadwal dan pengoperasian stop kontak pintar lalu mengamati apakah stop kontak dapat berfungsi sesuai dengan waktu yang sudah ditentukan. Jika jadwal berhasil diatur maka stop kontak akan secara otomatis menyala atau mati sesuai jadwal yang ditetapkan. Pengujian ini juga memeriksa apakah relay pada aplikasi berfungsi dengan baik dalam pengendalian operasi stop kontak sehingga dapat terjaga kehandalan dan stabilitas operasionalnya.

Tabel 4.3 Pengujian Penjadwalan

No.	Waktu	Status
1	10.16 – 10.17	Berhasil
2	10.19 – 10.22	Berhasil
3	10.23 – 10.28	Berhasil
4	10.29 – 10.36	Berhasil
5	10.37 – 10.47	Berhasil
6	10.49 – 11.09	Berhasil
7	11.11 – 11.41	Berhasil
8	11.47 – 12.17	Berhasil
9	12.20 – 13.20	Berhasil
10	13.30 – 15.30	Berhasil

4.2.6 Cara Kerja Alat

Stop kontak pintar berbasis IoT dengan relay bekerja dengan menghubungkan peralatan yang terhubung ke sumber listrik melalui relay yang dikendalikan secara elektronik. Relay berfungsi sebagai saklar elektronik yang mengontrol aliran listrik ke peralatan. Proses kerjanya dapat diringkas sebagai berikut:

1. Pengendalian Melalui Aplikasi: Stop kontak pintar dapat dikendalikan melalui aplikasi yang terhubung ke jaringan *Wi-Fi* rumah Anda.
2. Pengiriman Perintah: Perintah untuk menghidupkan atau mematikan stop kontak pintar dikirim melalui aplikasi.
3. Penerimaan Sinyal oleh Relay: Relay yang terpasang di dalam stop kontak pintar menerima sinyal perintah melalui jaringan *Wi-Fi*.
4. Pengendalian Aliran Listrik: Relay mengontrol aliran listrik ke peralatan yang terhubung. Ketika relay menerima perintah untuk menghidupkan peralatan, itu membuka sirkuit listrik dan mengizinkan aliran listrik. Sebaliknya, ketika relay menerima perintah untuk mematikan peralatan, itu menutup sirkuit listrik dan menghentikan aliran listrik.
5. Pengaturan Jadwal dan Automatisasi: Stop kontak pintar dapat dikonfigurasi untuk menjalankan perintah otomatis berdasarkan jadwal

yang ditentukan. Ini memungkinkan pengaturan waktu kapan peralatan harus dihidupkan atau dimatikan secara otomatis.

4.3 Pembahasan Hasil Pengujian

Pada penelitian ini penulis menggunakan acuan jurnal (Ari Farhan Nurihsan, Irving V Papatungan., 2020). Penulis menggunakan acuan jurnal ini dikarenakan penelitian terdahulu belum ada yang menggunakan fitur pada website Arduino IoT *Cloud* sebagai inti penjadwalan alat.

Berikut beberapa aspek yang dapat dibahas dalam pengujian:

1. Fitur:

- a. Pengontrolan jarak jauh: Pengguna dapat mengendalikan stop kontak pintar melalui aplikasi *smartphone* atau perangkat lain yang terhubung ke internet.
- b. Penjadwalan: Pengguna dapat mengatur jadwal otomatis untuk menghidupkan dan mematikan stop kontak pada waktu yang ditentukan.

2. Kinerja:

- a. Stabilitas koneksi: Stop kontak pintar harus memiliki koneksi yang stabil dengan jaringan *Wi-Fi* atau jaringan lainnya agar dapat diakses dan dikontrol dengan lancar.
- b. Respon Alat: Stop kontak pintar merespons perintah pengguna dan mengaktifkan atau menonaktifkan perangkat yang terhubung dengan aliran listrik dan internet.
- c. Kestabilan relay: Relay dalam stop kontak pintar bekerja dengan baik dan dapat menangani beban daya yang ditujukan tanpa masalah seperti *overheat* atau kegagalan operasional.

BAB V

KESIMPULAN

5.1 Kesimpulan

Setelah dilakukan berbagai pengujian dan analisa pada alat atau sistem stop kontak pintar berbasis IoT baik itu perangkat keras maupun perangkat lunak. Maka dapat diambil kesimpulan sebagai berikut:

1. Stop kontak pintar berbasis IoT dapat dikembangkan dengan *NodeMCU*, relay, dan *Wi-Fi* dengan cara menghubungkan *NodeMCU* ke jaringan *Wi-Fi* dan menggunakan relay sebagai saklar elektrik. *NodeMCU* berfungsi sebagai pengontrol utama yang menerima instruksi dari aplikasi melalui jaringan *Wi-Fi*, kemudian mengirimkan sinyal ke relay untuk mengatur aliran listrik pada stop kontak. Pengguna dapat mengendalikan stop kontak pintar melalui aplikasi yang terhubung dengan *NodeMCU*, sehingga memungkinkan pengoperasian jarak jauh dan pengaturan waktu pengoperasian yang fleksibel.
2. Setelah dilakukannya pengujian alat, fitur yang dimiliki stop kontak pintar ini adalah pengguna dapat mengontrol aliran listrik stop kontak melalui website atau aplikasi *Arduino IoT Cloud*, baik untuk menghidupkan maupun mematikan perangkat yang terhubung dengan stop kontak tersebut. Hal ini berfungsi untuk memudahkan pengguna dalam mengatur perangkat elektronik tanpa harus berada di tempat yang sama dengan stop kontak.

5.2 Saran

Penelitian ini banyak memiliki kekurangan dan banyak hal yang harus dikaji kemudian dikembangkan saran yang dapat penulis berikan untuk penelitian selanjutnya yaitu:

1. Melakukan penelitian lebih dalam tentang potensi resiko keamanan yang berkaitan dengan stop kontak pintar berbasis IoT.
2. Menguji dan mempelajari untuk ditambahkan lagi fitur-fitur yang ada pada web sehingga dapat lebih efisien lagi.

DAFTAR PUSTAKA

- Akbar Gumilang YS, Hadi MS, Lestari D. Stop kontak Pintar Berbasis *Internet of Things* sebagai Solusi Manajemen Energi Listrik dengan Menggunakan Aplikasi Android. *JASIEK (Jurnal Apl Sains, Informasi, Elektron dan Komputer)*. 2022;4(2):55-66. doi:10.26905/jasiek.v4i2.8332
- Aydarus, U. Al, Ahmad, U. A., & Saputra, R. E. (2022). Perancangan Stop Kontak Pintar Berbasis *Internet of Things* Menggunakan Real Time Clock *Internet of Things* Based Electric Switch Design Using Real Time Clock. *E-Proceeding of Engineering*, 9(3), 1228.
- D. A. Putra dan R. Mukhaiyar, "Monitor Daya listrik secara Real Time", *J. Vocat. Tek. Elektron. Dan Inform.*, vol. 8, tidak. 2, 2020.
- Iksan, F. N., & Tjahjadi, G. (2018). Perancangan Stop Kontak Pengendali Energi Listrik Dengan Sistem Keamanan Hubung Singkat Dan Fitur Notifikasi Berbasis *Internet of Things* (IoT). *Jurnal Elektro*, 11(2), 83–92. <http://ejournal.atmajaya.ac.id/index.php/JTE/article/view/535>
- Khaledi, M. T. Al, Nasri, & Hanafi, (2022). Rancang Bangun Sistem Rumah Pintar Menggunakan Platform Google Firebase Berbasis IoT (*Internet of Things*). *Jurnal Tektro*, 6(2), 194-202.
- M. T. Al Khaledi, N. and H. , "No. 02," *Rancang Bangun Sistem Rumah Pintar Menggunakan Platform Google Firebase Berbasis Internet of Things*, vol. 06, p. 9, 2022.
- Nurihsan, A. F., & Papatungan, I. V. (2022). Perancangan Stop Kontak Cerdas Berbasis IoT. *Jurnal Universitas Islam Indonesia: Automata*, 3(2), 1–8. <https://journal.uii.ac.id/AUTOMATA/article/view/24226/14054>
- Prastiantari, A. (2017). SKOPIN (Stop Kontak Pintar) Pengendali Arus Listrik Menggunakan Timer pada Stop Kontak Berbasis Arduino. Skripsi. Universitas Negeri Jakarta.
- Qodari, N., Christyono, Y., & Sukiswo, S. (2020). Perancangan Prototipe Realtime

Akuisisi Data Dan Kontrol Stop Kontak Menggunakan Esp 32 Berbasis Web.
Transient: Jurnal Ilmiah Teknik Elektro, 9(3), 406–411.

<https://doi.org/10.14710/transient.v9i3.406-411>

Rohman, N., Christyono, Y., & Sukiswo, S. (2020). Perancangan Sistem Monitoring Tegangan, Daya Dan Faktor Daya Pada Stop Kontak Berbasis *Internet of Things*. *Transient: Jurnal Ilmiah Teknik Elektro*, 9(3), 399–405.

<https://doi.org/10.14710/transient.v9i3.399-405>

Sujadi, H., Wahyuni, T., & Hamadi, W. N. (2021). Implementasi Stop Kontak Pintar pada Lampu Taman dengan Menggunakan Mikrokontroler Wemos D1 R1 Berbasis IoT. *Jurnal J-Ensatec*, 8(1), 574-581.

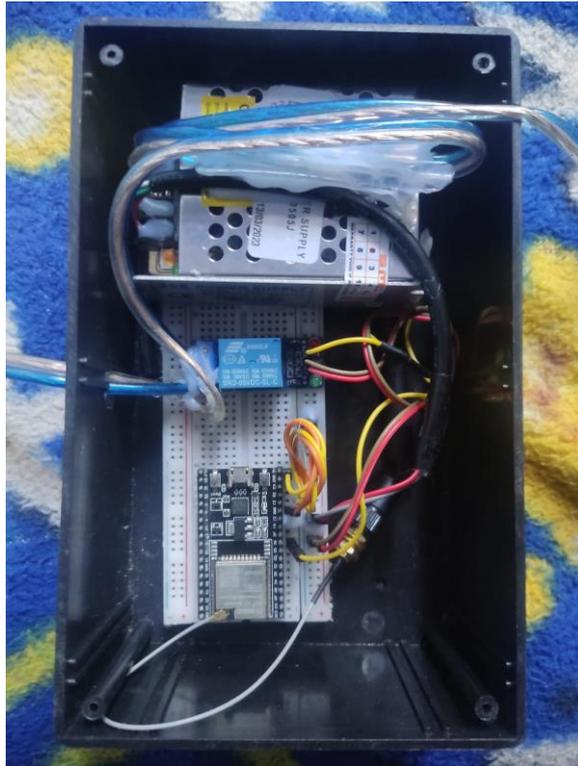
LAMPIRAN



Lampiran 1. Alat Pengukur Daya Listrik Yang Digunakan



Lampiran 2. Power Supply Yang Digunakan



Lampiran 3. Rangkaian Pada Alat



Lampiran 4. Pengetesan Alat Pada Pompa Air



Lampiran 5. Pengetesan Alat Pada Pompa Air (Lanjutan)



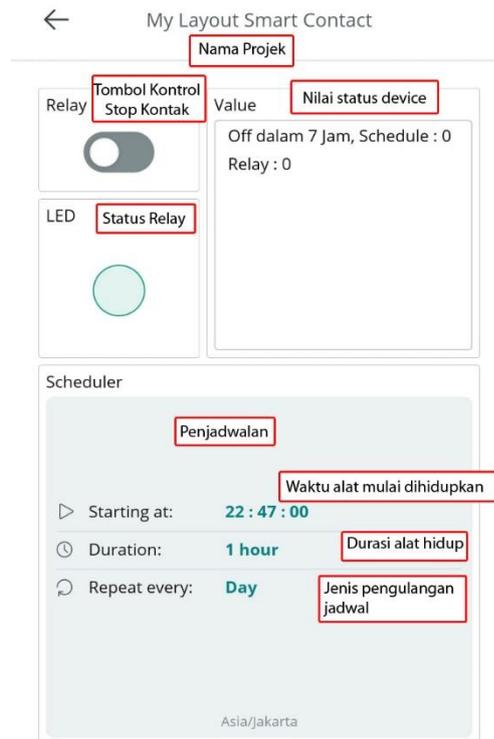
Lampiran 6. Penggunaan Daya Listrik Pada Alat



Lampiran 7. Penggunaan Daya Listrik Pada Alat (Lanjutan)



Lampiran 8. Penggunaan alat dengan Narasumber



Lampiran 9. Deskripsi pada aplikasi Arduino IoT *Cloud*

- Narasumber** : Farhan Amrillah Ahmad, S.Sos.
- Hari/Tanggal** : Sabtu, 7 Oktober 2023
- Waktu** : 17.00 WIB
- Tempat** : JL. Imam Bonjol Gg. Bungur, RT.001/ LK.II, Kel. Langkapura Baru, Kec. Langkapura
- Pertanyaan** : Menurut Anda bagaimanakah cara untuk menghemat listrik?
- Jawaban** : Mematikan barang elektronik jika tidak digunakan.
- Pertanyaan** : Apakah Anda tau barang elektronik yang memiliki tegangan listrik tinggi?
- Jawaban** : Yang saya ketahui ada mesin cuci, kulkas, air sanyo / pompa air sumur, *magicom* / *rice cooker*, dan setrika.
- Pertanyaan** : Menurut Anda di rumah ini barang elektronik mana yang sering anda hidup dan matikan?
- Jawaban** : Lampu dan air sanyo / pompa air sumur.
- Pertanyaan** : Berapa lama waktu yang diperlukan pompa air untuk mengisi air dari kosong sampai penuh?
- Jawaban** : Air akan penuh sekitar 30 - 60 menit.
- Pertanyaan** : Berapa kali pompa air dihidupkan dalam sehari untuk mengisi air?
- Jawaban** : Dalam sehari pompa air dihidupkan sebanyak 2 kali
- Pertanyaan** : Apakah Anda sering lupa mematikan pompa air ketika air sudah penuh?
- Jawaban** : Tidak karena sudah di cek terlebih dulu kalau hidupin jam berapa harus dimatikan jam berapa.
- Pertanyaan** : Apakah Anda tau daya listrik yang digunakan ketika pompa air dihidupkan?
- Jawaban** : Tidak
- Pertanyaan** : Apakah ada perubahan ketika dipasangkan alat pada pompa air dalam 1 hari percobaan ini?
- Jawaban** : Yang pasti sudah tidak perlu colok cabut pada colokan/steker, tidak perlu lihat-lihat jam selalu.
- Pertanyaan** : Apakah Anda terbantu dengan sistem pada alat stop kontak pintar?
- Jawaban** : Membantu dan efisien dalam hal waktu.

Lampiran 10. Hasil Wawancara