

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu sebagai acuan untuk penulis dalam pembuatan laporan akhir ini. Penelitian terdahulu sebagai referensi dan literatur dalam memperkaya bahan kajian pada penelitian penulis. Dalam penelitian penulis ini, tidak ditemukannya kesamaan judul dengan judul penelitian berikut. Berikut ini adalah beberapa jurnal yang terkait dengan penelitian yang dilakukan penulis.

Pada penelitian dengan judul **“Rancang Bangun Alat Pemutus Tegangan Pada Kwh Meter Pelanggan PLN”** oleh Akbar Abadi, Riza Widya, dan Julsam. Menjelaskan bahwa PT. PLN (Persero) merupakan salah satu perusahaan penjual jasa listrik di Indonesia. Besarnya daya yang digunakan oleh konsumen dihitung dengan menggunakan KWh meter. Pada konsumen yang menggunakan KWh meter pascabayar masih sering terjadi tunggakan tagihan listrik. Hal tersebut merupakan salah satu kerugian bagi perusahaan. Jika tunggakan tidak dibayar oleh konsumen, maka aliran listrik ke konsumen akan diputus. Untuk mempermudah pekerjaan, maka dibuatlah alat pemutus tegangan yang akan memutus aliran listrik ke rumah konsumen. Pemutusan ini cukup dilakukan dari satu tempat tanpa perlu mengunjungi rumah konsumen. Alat pemutus tegangan dibuat dengan menggunakan *Arduino Mega 2560, modul relay 5V 1 channel, relay omron MK2P* dan *modul SIM800L v2*. Perintah dari operator diterima oleh *modul SIM800L v2* yang kemudian diteruskan ke *Arduino Mega 2560* untuk mengkatifkan *relay*. Abadi, A, dkk (2021).

Pada penelitian dengan judul **“Rancang Bangun Pemutus Aliran Listrik Sistem Tegangan Rendah Berbasis *Internet of Things (Iot)* pada Daerah Banjir”** oleh Mihammad Hanif Zidan Banjir merupakan permasalahan tersendiri yang terjadi secara tiba-tiba tanpa mengenal waktu. Jika terjadi banjir maka listrik harus segera dipadamkan. Hal ini merupakan tindakan *preventif* dengan mempertimbangkan keselamatan pengguna tenaga listrik. Dibutuhkan suatu sistem untuk memberikan informasi tentang banjir dan juga sebagai pemutus arus yang dapat dipantau dimana saja dan kapan saja. Perancangan Pemutus Aliran Listrik

Tegangan Rendah Berbasis IoT pada Daerah Banjir merupakan perancangan alat sebagai pemutus jaringan tegangan rendah dan juga untuk dapat mengetahui ketinggian air melalui media IoT. Dengan adanya informasi ketinggian air tersebut, PLN sebagai penyedia jasa kelistrikan dapat dengan mudah memantau daerah tersebut terutama di daerah yang sering terjadi banjir, sehingga jika terjadi banjir di daerah tersebut PLN dapat segera memutus aliran listrik di daerah tersebut. Pengujian yang dilakukan pada ketiga sensor didapatkan alat dapat bekerja dengan baik dan akurat dalam membaca ketinggian air di daerah banjir, dengan kesalahan pembacaan sensor sebesar 1,24%. Kesimpulan yang diperoleh adalah Perancangan Pemutus Arus Listrik Tegangan Rendah Berbasis IoT di Daerah Banjir dapat bekerja dengan menggunakan dua metode yaitu sebagai pemutus otomatis dan juga sebagai pemantau ketinggian air sungai dan dapat digunakan sebagai pengaman dari bahaya sengatan listrik saat kondisi banjir. Zidan, M. H. (2022).

Pada penelitin dengan judul “**Pemutus Aliran Daya Listrik Melalui Fasilitas *Short Message Service (Sms)* Berbasis *Mikrokontroler***” oleh Besma Nugraha AP, Dikpride Despa, Noer Soedjarwanto. Menjelaskan tentang perkembangan teknologi berimbang pada tingginya kebutuhan energi listrik. Oleh sebab itu, perlu dirancang sebuah alat pemutus aliran daya yang dapat memutus aliran daya dari jarak jauh dan diharapkan dengan adanya alat ini energi listrik dapat dimanfaatkan secara efisien. Alat ini dirancang untuk dapat memutus, menghubungkan, serta mengirimkan status aktif atau tidak-aktifnya aliran daya listrik dengan beban hingga 2,2kW. Alat ini menggunakan bahasa C sebagai bahasa pemrograman pada *mikrokontroler Atmega8* sebagai pengendali utama, TRIAC dan kontaktor sebagai sakelar dan *handphone Siemens C55* sebagai penerima perintah. Dari hasil pengujian diperoleh: (1) sistem pemutus aliran daya listrik ini dapat menerima dan menerjemahkan perintah dalam bentuk SMS, (2) sistem pemutus aliran daya listrik ini dapat memutus dan menghubungkan aliran daya listrik (3) sistem pemutus aliran daya listrik ini dapat mengirimkan status aktif atau tidaknya aliran daya listrik. Besma Nugraha AP, dkk (2022).

Pada penelitin dengan judul “**Rancang Bangun Sistem *Monitoring Data Listrik pada Kamar Kos Berbasis (IoT)***” oleh Ivan Safril Hudan dan Tri Rijanto

menjelaskan tentang teknologi yang berkembang pesat di berbagai bidang keilmuan. Manusia terus berupaya mengembangkan dan meneliti teknologi-teknologi terbaru dalam rangka untuk mempermudah kehidupan manusia. Salah satunya yaitu pada bidang teknologi IoT (*Internet of Things*). Penggunaan peralatan listrik pada kamar kos, setiap kamar memiliki konsumsi daya listrik yang berbeda-beda. Dan ini sering terjadi sehingga dalam menggunakan peralatan listrik setiap kamar kos diperlukan alat untuk *memonitoring* penggunaan daya listrik, agar penggunaan daya listrik pada kamar kos ini sesuai dengan daya yang dibutuhkan. Oleh karena itu pula dirancang alat yang dapat mempermudah melakukan aktivitas memantau pemakaian daya listrik yang hasilnya dapat ditampilkan melalui LCD 16X2 dan dapat diinformasikan melalui *internet*. Tujuan penelitian ini adalah merancang bangun sistem *monitoring* daya listrik berbasis IoT untuk mempermudah memantau penggunaan daya listrik pada kamar kos berbasis IoT. Metode yang digunakan dalam pengumpulan data yaitu dengan metode kuantitatif. Dengan pengumpulan beberapa komponen yang dibutuhkan, yang dirancang pada penelitian ini seperti, sensor tegangan, sensor arus, *wemos D1 mini*, *relay 5V*, dan *arduino* Uno R3. Pada alat ini akan *memonitoring* daya berbasis IoT, dan dapat di *monitoring* melalui internet berupa tampilan grafik pada server *thingspeak.com*. Hasil penelitian menunjukkan bahwa telah didapatkan nilai error rata-rata pada pengujian sensor tegangan sebesar 0,02%, sensor arus memiliki nilai error sebesar 0,01 dan nilai pada daya sebesar 0,22. Sehingga dapat disimpulkan bahwa dengan memiliki selisih dan error cukup kecil, alat ini dikatakan cukup baik dan sistem dapat bekerja dengan baik. Hudan, I. S., dan Rijanto, T. (2019).

Pada penelitin dengan judul “**Rancang Bangun Sistem *Monitoring* Data Listrik pada Rumah Kos Menggunakan NodeMCU dan *Firebase* Berbasis *Android***” oleh Achmad Furqon, Agung budi Prasetijo dan Eko Didik Widiyanto. Menjelaskan tentang suatu sistem yang digunakan untuk memantau daya listrik yang dipakai oleh penyewa kos serta menghitung besarnya energi yang digunakan dan juga tarif yang dikenakan. Sistem dirancang menggunakan modul NodeMCU yang akan dipasangkan dengan sensor PZEM-004t dan *relay*. NodeMCU berfungsi untuk pengiriman data ke *database*. Sensor PZEM-004t digunakan untuk membaca

tegangan dan arus listrik yang mengalir sehingga bisa didapatkan nilai daya. *Relay* digunakan sebagai kontrol untuk memutus sambungan listrik jika diperlukan. Dan untuk mengetahui daya listrik tersebut data daya akan dikirimkan ke *Realtime Database Firebase* yang akan diakses oleh perangkat *Android* melalui *internet* sehingga sistem *monitoring* ini dapat dilakukan secara jarak jauh. Hasil pengukuran sensor menunjukkan akurasi kesalahan sebesar $\pm 1,8\%$ dengan pembandingan multimeter digital. Hasil pengujian aplikasi sistem menunjukkan aplikasi dapat memantau daya pada tiap kamar kos secara *realtime*. Furqon, A., dkk (2019).

Pada penelitian dengan judul “***Proposed Design and Modeling of Smart Energy Dashboard System by Implementing IoT (Internet of Things) Based on Mobile Devices***” oleh Syaiful Ahdan, Erliyan Redy Susanto, dan Nana Rachmana Syambas. Menjelaskan tentang listrik menjadi salah satu kebutuhan masyarakat yang paling penting, namun penggunaan listrik yang efisien seringkali sulit dilakukan ketika orang kurang memiliki kesadaran dan disiplin dalam menggunakan listrik. Salah satu masalah yang sering terjadi adalah ketika pemilik rumah tidak berada di lokasi dan ingin mengendalikan peralatan listrik. *Internet of Things* (IoT) adalah sebuah konsep yang bertujuan untuk memperluas konektivitas jaringan *internet* yang terhubung dengan jaringan *global*, IoT dapat digunakan untuk mengendalikan peralatan elektronik secara jarak jauh atau otomatis. Penelitian ini bertujuan untuk membangun alat pengendali jarak jauh yang memanfaatkan teknologi *internet* dengan melakukan proses pengendalian perangkat yang sudah ada pada jaringan *internet* melalui sistem pada perangkat seluler. Kontribusi dari penelitian ini adalah menciptakan desain dan model yang akan digunakan untuk membangun perangkat energi pintar yang diterapkan pada fitur pintar rumah dengan menggunakan perangkat sensor sehingga peralatan elektronik dapat dikendalikan sesuai dengan kondisi yang diinginkan, secara otomatis atau jarak jauh melalui aplikasi *android*. Papan *dashboard* energi pintar ini dirancang menggunakan *mikrokontroler* ESP8266 yang terhubung ke *cloud Firebase* yang akan mengirimkan sinyal digital ke *relay* sebagai pemutus aliran listrik. Berdasarkan hasil pengujian sistem, diperoleh tingkat ketergunaan sebesar 90,3%, fungsionalitas sebesar 85,8%, dan keandalan sebesar 90%. Uji efisiensi

aspek menghasilkan tingkat efisiensi CPU tertinggi sebesar 28% dan menurun secara stabil pada persentase 10%. Untuk penggunaan memori tertinggi sebesar 119,2MB dimulai dengan penggunaan memori sebesar 47,7MB dari alokasi total 143,1MB. Kesimpulan dari penelitian ini adalah proposal *Dashboard Energi Pintar* berbasis IoT dapat membantu dalam mengendalikan perangkat listrik secara *real-time*. Ahdan, S., dkk (2019).

Pada penelitian dengan judul “*D-Distance Technique to Determine Failure Probability of Power Circuit Breaker*” oleh Suphon Kumpalavalee, Thanapong Suwanasri, Cattareeya Suwanasri, dan Rattanakorn Phadungthin. Menjelaskan bahwa, diajukan sebuah faktor jarak D baru untuk menentukan probabilitas kegagalan dan memberi prioritas pada tindakan perawatan pemutus sirkuit daya di gardu induk tegangan tinggi. Faktor jarak D dihitung dengan menggunakan indeks kondisi dan indeks renovasi dari pemutus sirkuit tegangan tinggi (HVCB). Untuk memfasilitasi pengambilan keputusan yang efektif tentang perawatan dengan metode yang sederhana dan usaha komputasi yang lebih sedikit, model yang diusulkan menggabungkan metode penilaian bobot-skoring (WSM) dan proses hierarki analitis (AHP) dengan berbagai metode diagnostik untuk penilaian indeks kondisi serta persyaratan operasi HVCB untuk penilaian indeks renovasi. Banyak parameter penting dari pengujian pemutus sirkuit, seperti resistansi isolasi, resistansi kontak, waktu kontak, pengukuran gas SF₆, tingkat kebocoran gas, inspeksi visual, dll., telah dipertimbangkan untuk penentuan indeks kondisi. Selain itu, parameter-parameter penting lainnya, seperti usia pemutus sirkuit, usia pemutus dan mekanisme, jumlah gangguan arus, rasio arus beban aktual terhadap arus nominal, rasio arus hubung singkat aktual terhadap arus pemutusan nominal, kemampuan perawatan, ketersediaan suku cadang, tingkat keahlian perawatan, dll., juga dipertimbangkan untuk penentuan indeks renovasi. Untuk memvalidasi model yang diusulkan, data uji praktis dari dua puluh HVCB 115 kV di berbagai gardu induk dari sebuah perusahaan distribusi di Thailand digunakan dan diuji. Dengan menganalisis kondisi aktual dan persyaratan operasi dari pemutus sirkuit, keluaran berupa indeks kondisi dan indeks renovasi menggunakan metode yang diusulkan dibahas bersama ahli HVCB di perusahaan untuk menyesuaikan skor dan bobot dari

semua kriteria untuk mendapatkan model yang paling akurat dan dapat diandalkan. Hasilnya menunjukkan bahwa teknik jarak D yang diukur dari matriks risiko, yang didefinisikan sebagai probabilitas kegagalan, dapat digunakan untuk merangking jadwal perawatan dari tugas perawatan mendesak hingga normal. Selain itu, berbagai probabilitas kegagalan dalam matriks risiko pemutus sirkuit dapat digunakan untuk menentukan strategi perawatan yang tepat untuk pemutus sirkuit daya di setiap grup. Akhirnya, metode yang diusulkan dapat membantu para manajer perusahaan dan insinyur perawatan mengelola perencanaan perawatan secara efektif dan mudah untuk ribuan HVCB di jaringan, dan dapat diterapkan lebih lanjut dengan peralatan tegangan tinggi lainnya baik dalam sistem transmisi maupun distribusi untuk memfasilitasi kegiatan perawatan sesuai dengan biaya yang tersedia dan sumber daya manusia. Kumpalavalee, S., dkk (2023).

Pada penelitan dengan judul “*Feature Importance Analysis for Power Circuit Breaker Vibration-Based Condition Assessment*” oleh Kerim Obarcanin dan Bakir Lacevic. Menjelaskan tentang tentang parameter-parameter berpengaruh dalam penilaian kondisi pemutus sirkuit daya berdasarkan sidik jari getaran. Dengan menciptakan subset fitur berdasarkan domain komputasi yang berasal dari sidik jari getaran, fitur-fitur tersebut pertama-tama diurutkan berdasarkan empat algoritma perangkingan fitur. Untuk mengonfirmasi kontribusi fitur yang telah diurutkan terhadap kinerja klasifikasi, 11 *classifier machine learning* yang berbeda dilatih. Pelatihan *classifier* dilakukan pada set fitur lengkap, di mana kemudian *classifier* yang sama dilatih dengan subset fitur yang diurutkan berdasarkan algoritma perangkingan. Perangkingan dan kinerja *classifier* menghasilkan konsep kurtosis dalam domain waktu dan frekuensi sebagai fitur yang sangat menjanjikan untuk klasifikasi biner yang dapat menggambarkan kondisi mekanis pemutus sirkuit dengan kredibilitas. Obarcanin, K., dan Lacevic, B. (2023).

Pada penelitan dengan judul “*Effect of Contact Recovery Voltage of High Voltage Power Breakers with Different Types of Circuit Characteristics*” oleh Johanis Leuwol. Menjelaskan tentang gangguan yang sering terjadi pada saluran transmisi merupakan gangguan alam, seperti gangguan petir pada jaringan transmisi akibat jaringan transmisi melalui udara, panjang, tinggi dan tersebar di

berbagai daerah terbuka serta beroperasi dalam segala macam kondisi. Di antara pertimbangan yang diambil dalam desain transmisi saluran transmisi adalah letak kabel arde terhadap kabel fasa. Karena kabel *ground* saja, jadi persentase kecil pada kabel fasa. Dan hingga saat ini belum ada ulama yang menunjukkan kegunaan petir bagi kehidupan, belum mendapatkan cara untuk mencegah atau memanfaatkan energi yang ditimbulkan oleh petir tersebut. Meskipun demikian, ilmu pengetahuan manusia terus berkembang dengan kemajuan teknologi. Dengan satu alat pengaman, kabel udara (*Overhead Ground Wire*) untuk melindungi kabel fasa dari saluran transmisi. Leuwol, J. (2023).

Pada penelitan dengan judul “*Condition Assessment of Power Circuit Breakers Based on Machine Learning Algorithms*” oleh Kerim Obarcanin, Dzenita Skulj dan Bakir Lacevic. Menjelaskan tentang kondisi pemutus sirkuit daya. Yang pertama mencakup berbagai algoritma klasifikasi pembelajaran mesin di mana input untuk klasifikasi adalah kumpulan fitur yang dipilih secara manual. Yang kedua menggunakan klasifikasi pembelajaran mendalam berdasarkan jaringan saraf *convolutional*. Kedua pendekatan tersebut berkisar pada ide di balik kurtosis spektral, yang salah satunya mengeksplorasi representasi visualnya dalam bentuk kurtogram. Pendekatan pertama menggunakan kurva kurtosis spektral sebagai dasar ekstraksi fitur sedangkan pendekatan kedua menggunakan kurva kurtosis spektral sebagai masukan tunggal ke dalam jaringan syaraf tiruan. Validasi dilakukan pada sejumlah besar tanda getaran dan dibandingkan dengan algoritme canggih pesaing. Hasilnya menunjukkan fitur yang menjanjikan dari pendekatan yang diusulkan. Obarcanin, K., dkk (2023).

2.1 Rancang Bangun

Pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Zidan, M. H. (2022).

Jadi dapat disimpulkan bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian

pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

2.2 Internet of Things (IoT)

Menurut Casagras (*Coordination and support action for global RFID-related activities and standardisation*) pengertian *Internet of Things* (IoT) adalah sebuah infrastruktur jaringan *global*, yang dapat menghubungkan perangkat keras dan *virtual* melalui eksploitasi data *capture* serta kemampuan komunikasi. Dalam Infrastruktur terdiri dari jaringan yang sudah ada dan *internet* beserta pengembangan jaringannya. Sedangkan menurut SAP (*Systeme, Anwendungen and Produkte*). SAP, *Internet of Things* (IoT) merupakan sebuah benda-benda perangkat keras yang diintegrasikan ke dalam jaringan informasi secara berkesinambungan, serta di mana benda-benda fisik tersebut dapat berperan aktif dalam proses bisnis. Sedangkan ETT EPOSS mendefinisikan, IoT merupakan jaringan yang dibentuk oleh benda yang memiliki identitas.

Pada dunia maya dengan cara beroperasi di ruang itu dengan menggunakan kecerdasan perangkat untuk terhubung dan berkomunikasi dengan pengguna, konteks sosial dan lingkungan. *Internet of Things* merupakan sebuah konsep yang memiliki fungsi untuk memperluas konektivitas pada *Internet* yang tersambung secara terus – menerus. Konsep *Internet of Things* atau IoT awalnya dikeluarkan oleh Kevin Ashton pada tahun 1999 di salah satu presentasinya. Kini banyak perusahaan besar mulai mendalami *Internet of Things* sebut saja *Intel*, *Microsoft*, *Oracle*, dan banyak lagi lainnya. Bahkan banyak orang berpendapat IoT akan menjadi “*The Next Big Thing*” di dunia teknologi informasi, hal ini karena IoT menawarkan banyak potensi yang bisa dikembangkan kembali. Saat ini, perkembangan IoT ini sudah memasuki Teknologi *nirkabel*, *Internet*, dan *Microelectromechanical System*. Darmanto, T., dan Krisma, H. (2019).

2.2.1 Unsur dalam IoT

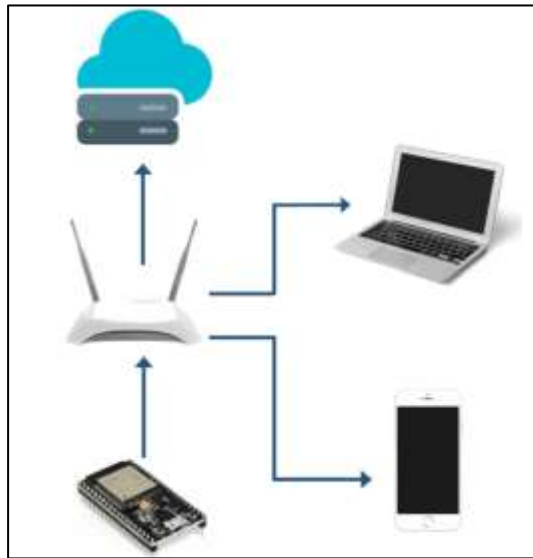
Adapun unsur yang harus ada di *Internet of Things* adalah sebagai berikut:

- a. Sensor, yakni perangkat yang bisa mendapatkan informasi terkait dari hal hal tertentu seperti sensor gerak, suhu, udara, panas, dan lainnya.
- b. Konektivitas, yakni yang berfungsi sebagai penghubung dan pertukaran informasi yang terjadi pada *Internet of Things* (IoT). Konektivitas ini biasanya yang dibutuhkan harus stabil namun tidak perlu dalam bentuk yang besar juga.
- c. Perangkat yang berukuran kecil, yakni yang bisa mendukung dan meningkatkan ketepatan, skalabilitas dan *fleksibel* dalam pengembangan IoT. Dan teknologi memang seperti itu makin kecil makin murah dan lebih kuat.

2.2.2 Manfaat Internet of Things

Adapun manfaat dari *Internet of Thing* adalah sebagai berikut:

- a. Monitoring Lingkungan
 Contohnya IoT bisa *memonitor* kondisi rumah selama pemilik keluar. Contoh lain adalah IoT digunakan untuk “melihat” kondisi air secara *real-time* di waduk, irigasi bagi para petani ataupun peternak untuk informasi debit air masih banyak atau tinggal sedikit.
- b. Pengelolaan Infrastruktur
 IoT bisa digunakan untuk hal pengelolaan infrastruktur. Contohnya MRT
- c. Untuk Sensor Peralatan
 Kebanyakan biaya konsumsi peralatan di pertambangan diukur berdasar kapasitas dan pengalaman saja. Tetapi, dengan IoT perusahaan tambang dapat mengukur peralatan mana yang BBM nya sudah mau habis, berapa stok BBM di site, peralatan mana yang oliya harus di ganti, dan lain sebagainya sehingga dapat terukur secara cepat dan tepat.
- d. Home Managemen
 Peralatan elektronik seperti kulkas atau AC bisa memberi tahu pengguna jika stok makanan habis, atau AC lupa dimatikan.



Gambar 2. 1 Skema pemanfaatan dari *Internet of Things*

2.2.2 Cara Kerja Internet of Things

Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah instruksi pemrograman yang dimana tiap-tiap perintah argumennya itu dapat menghasilkan sebuah interaksi antara sesama perangkat yang saling terhubung satu sama lainnya secara otomatis tanpa campur tangan manusia. Bahkan dalam jarak yang jauh sekalipun. *Internet* dapat menjadi penghubung diantara kedua interaksi perangkat tersebut. Sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung. Tantangan terbesar dalam dunia *Internet of Things* ialah menyusun jaringan komunikasinya sendiri, yang dimana jaringan tersebut sangatlah kompleks, dan memerlukan sistem keamanan yang ketat. Selain itu biaya operasional yang mahal sering menjadi penyebab kegagalan yang berujung pada gagalnya produksi.

2.3 ESP32

ESP32 adalah singkatan dari "*Espressif System Processor 32*," yang merupakan sebuah sistem pada *chip* (SoC) yang dikembangkan oleh perusahaan teknologi *Espressif Systems*. ESP32 adalah *mikrokontroler* yang populer dan kuat yang menawarkan konektivitas WiFi dan *Bluetooth*, serta memiliki dua inti CPU

(*dual-core*) yang bekerja secara independen untuk meningkatkan kinerja dan efisiensi. Illahi, R. (2021).

Mikrokontroler ESP32 menyediakan berbagai peripheral (*periferal*) yang dapat digunakan untuk menghubungkan dan mengontrol perangkat keras eksternal seperti sensor, layar, motor, dan banyak lagi. Berikut adalah diagram blok *chip* ESP32 ditunjukkan pada gambar 2.2. Illahi, R. (2021).



Gambar 2. 2 Diagram Blok *Chip* ESP32

A. Peripheral

Peripheral adalah bagian dari sebuah *mikrokontroler* yang berfungsi untuk menghubungkan dan mengontrol perangkat keras eksternal seperti sensor, modul komunikasi, motor, dan banyak lagi. Fungsi dari peripheral pada ESP32 adalah untuk memberikan kemampuan tambahan pada *mikrokontroler* tersebut sehingga dapat berinteraksi dengan dunia luar dengan lebih luas dan kompleks.

B. Bluetooth

Bluetooth pada ESP32 memungkinkan modul ini untuk berkomunikasi dengan perangkat lain melalui koneksi *nirkabel Bluetooth*. Fungsi *Bluetooth* pada ESP32 mencakup beberapa hal yang penting dalam pengembangan proyek IoT (*Internet of Things*) dan aplikasi yang memerlukan komunikasi antarperangkat.

C. WiFi

WiFi pada ESP32 adalah salah satu fitur utama yang membuatnya menjadi

platform yang kuat untuk proyek *Internet of Things* (IoT) dan komunikasi *nirkabel*.

D. ESP32 Core dan Memory

Pada ESP32, istilah ESP32 *Core* dan *Memory* mengacu pada konsep yang berbeda, tetapi keduanya memiliki peran yang krusial dalam mengoperasikan *mikrokontroler*. Berikut adalah penjelasannya:

- ESP32 *Core* merujuk pada dua inti CPU yang terdapat pada *mikrokontroler* ESP32. ESP32 memiliki prosesor *dual-core*, yang berarti terdapat dua unit pemrosesan utama (*core*) yang bekerja secara *independen* satu sama lain. *Core* ini disebut "*Core 0*" dan "*Core 1*". Pada umumnya, tugas yang lebih intensif komputasi (seperti pemrosesan data, algoritma kompleks) diberikan ke *Core 1*, sementara tugas yang lebih berkaitan dengan komunikasi dan pemantauan (seperti komunikasi WiFi atau *Bluetooth*, pengambilan data sensor) diberikan ke *Core 0*.
- *Memory* pada ESP32 memiliki dua jenis utama memori yang berperan penting dalam operasinya yaitu RAM (*Random Access Memory*) dan *Flash Memory*.

E. Radio

Radio mengacu pada fitur *nirkabel mikrokontroler* ini, yang mencakup WiFi (802.11b/g/n) dan *Bluetooth* (*Classic* dan *Bluetooth Low Energy*). Fungsi radio pada ESP32 adalah untuk memungkinkan komunikasi *nirkabel* dengan perangkat lain melalui koneksi WiFi atau *Bluetooth*.

F. ULP Coprocessor

ULP (*Ultra-Low Power*) *coprocessor* adalah bagian dari ESP32 yang bertanggung jawab untuk menjalankan tugas-tugas dengan konsumsi daya yang sangat rendah. ULP *coprocessor* adalah sebuah prosesor yang terintegrasi secara khusus pada ESP32 yang beroperasi secara *independen* dari unit CPU utama. Fungsinya adalah untuk memungkinkan ESP32 untuk tetap beroperasi dalam mode *ultra-low power* saat inti CPU utama (*dual-core*) tidak aktif atau saat ESP32 berada dalam mode *deep sleep*.

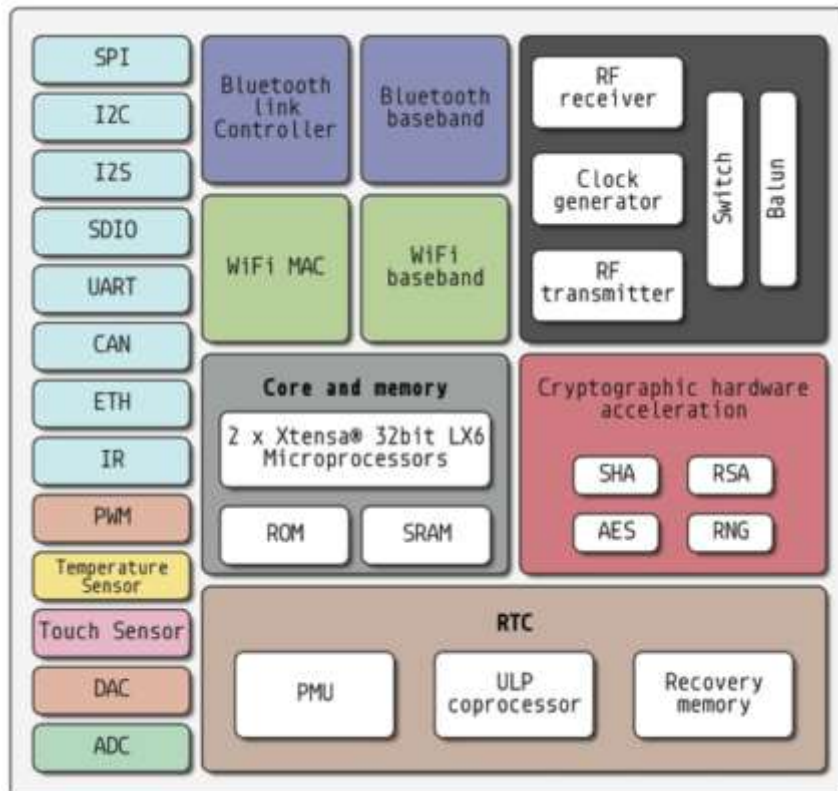
G. RTC dan RTC peripherals pada ESP32

- RTC (*Real-Time Clock*) pada ESP32 adalah komponen yang bertanggung

jawab untuk mengukur waktu secara akurat dan menyimpan informasi waktu dan tanggal.

- RTC *peripherals* pada ESP32 adalah modul tambahan yang terkait dengan RTC dan menyediakan beberapa fitur tambahan.

Pada gambar 2.3 dibawah ini adalah diagram arsitektur dari ESP32



Gambar 2. 3 Diagram Arsitektur pada ESP32

2.5 Relay

Modul *relay* adalah salah satu piranti yang beroperasi berdasarkan prinsip *elektromagnetik* untuk menggerakkan kontaktor guna memindahkan posisi *on* ke *off* atau sebaliknya dengan memanfaatkan tenaga listrik. Peristiwa tertutup dan terbukanya kontaktor ini terjadi akibat adanya efek induksi magnet yang timbul dari kumparan induksi listrik. Perbedaan yang paling mendasar antara *relay* dan sakelar adalah pada saat pemindahan dari posisi *on* ke *off*. *Relay* melakukan pemindahan-nya secara otomatis dengan arus listrik, sedangkan sakelar dilakukan dengan cara manual. Lukman, M. (2021).

2.5.1 Jenis-Jenis Relay

Macam macam *relay* dan fungsinya digolongkan menjadi dua macam:

1. Jenis *relay* berdasarkan *trigger* atau pemicunya

Ada dua jenis *relay* yang beredar di pasaran berdasarkan *trigger* atau pemicunya, yaitu:

- *Low Level Trigger*, adalah *relay* yang akan berfungsi (menyala) jika diberikan kondisi *LOW*
- *High Level Trigger*, adalah *relay* yang akan berfungsi (menyala) jika diberikan kondisi *HIGH*

2. Jenis *relay* berdasarkan jumlah *channel*-nya

- Modul *relay* 1 *channel*
- Modul *relay* 2 *channel*
- Jenis modul *relay* 4 *channel*
- Modul *relay* 8 *channel*
- Modul *relay* 16 *channel*
- Jenis modul *relay* 32 *channel*

2.5.2 Fungsi Relay

Fungsi modul *relay* adalah sebagai saklar elektrik. Dimana ia akan bekerja secara otomatis berdasarkan perintah logika yang diberikan. Kebanyakan, *relay* 5 volt DC digunakan untuk membuat *project* yang salah satu komponennya butuh tegangan tinggi atau yang sifatnya AC (*Alternating Current*). Sedangkan kegunaan *relay* secara lebih spesifik adalah sebagai berikut:

- a. Menjalankan fungsi logika dari *mikrokontroler Arduino*
- b. Sarana untuk mengendalikan tegangan tinggi hanya dengan menggunakan tegangan rendah
- c. Meminimalkan terjadinya penurunan tegangan
- d. Memungkinkan penggunaan fungsi penundaan waktu atau fungsi *time delay function*

- e. Melindungi komponen lainnya dari kelebihan tegangan penyebab korsleting.
- f. Menyederhanakan rangkaian agar lebih ringkas.

2.5.3 Cara Kerja Relay

Untuk dapat memahami prinsip kerja *relay*, berikut adalah lima fungsi komponen dari *relay*:

- a. Penyangga (*Armature*)

Penyangga atau sering disebut juga sebagai *armature* pada *relay* adalah komponen yang berfungsi sebagai penopang atau penahan bagi kontak-kontak dalam *relay* elektromagnetik. *Relay* adalah perangkat elektronik yang digunakan untuk mengendalikan aliran listrik dengan menggunakan prinsip *elektromagnetisme*. Dalam *relay elektromagnetik*, terdapat satu atau lebih lilitan kumparan yang terhubung dengan sumber daya listrik. Ketika arus mengalir melalui kumparan, medan magnet dihasilkan di sekitar kumparan. Medan magnet ini kemudian menarik penyangga (*armature*) yang terbuat dari bahan yang dapat terpengaruh oleh medan magnet, seperti logam *feromagnetik*. Penyangga (*armature*) biasanya terhubung dengan kontak-kontak dalam *relay*. Kontak-kontak ini dapat berupa kontak bantu (kontak *normally open* atau *normally closed*) atau kontak utama. Ketika penyangga ditarik oleh medan magnet, kontak-kontak dalam *relay* akan mengalami perubahan posisi, yaitu dari terbuka menjadi tertutup atau sebaliknya.

- b. Kumparan (*Coil*)

Kumparan (*coil*) pada *relay* adalah salah satu komponen utama yang menghasilkan medan magnet yang digunakan untuk mengendalikan kontak-kontak dalam *relay*. Kumparan terbuat dari kawat tembaga atau kawat lain yang bersifat konduktif dan dililitkan dalam bentuk kumparan di sekitar inti magnetik. Kumparan berfungsi sebagai penghasil medan magnet ketika arus listrik mengalir melaluinya. Ketika arus listrik mengalir melalui kumparan, medan magnet dihasilkan sepanjang kawat yang dililitkan. Medan magnet

ini kemudian mempengaruhi penyangga (*armature*) dalam *relay*, yang kemudian menggerakkan kontak-kontak *relay*.

c. Pegas (*Spring*)

Pegas (*spring*) pada *relay* adalah salah satu komponen yang digunakan untuk memberikan gaya pemulihan atau gaya penarikan pada penyangga (*armature*) setelah medan magnet berkurang atau hilang. Pegas ini berfungsi untuk mengembalikan penyangga ke posisi awal setelah terjadi perubahan posisi kontak-kontak dalam *relay*. Pegas pada *relay* biasanya terletak di antara penyangga (*armature*) dan kerangka atau struktur *relay*. Pegas ini dapat berbentuk pegas spiral (*coil spring*) atau pegas piringan (*disk spring*), tergantung pada desain dan kebutuhan *relay*.

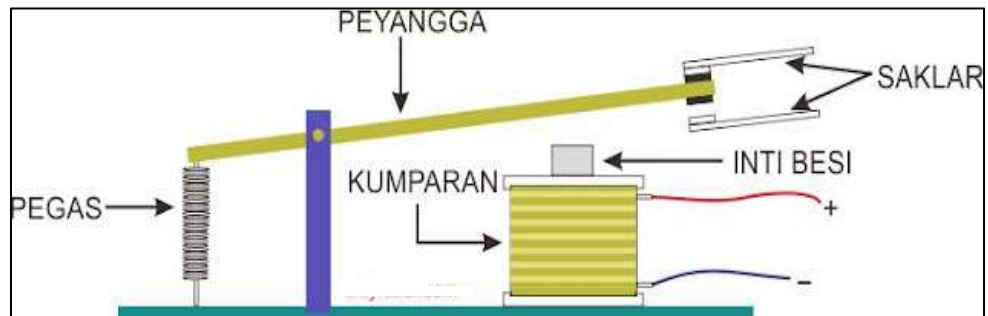
d. Saklar (*Switch Contact*)

Saklar (*switch contact*) pada *relay* adalah komponen yang bertanggung jawab untuk mengendalikan aliran listrik dalam suatu rangkaian. Saklar ini terdiri dari dua bagian utama: kontak bantu (*auxiliary contact*) dan kontak utama (*main contact*).

e. Inti Besi (*Iron Core*)

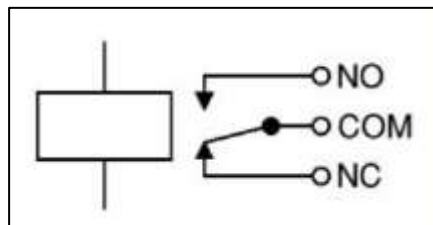
Inti besi (*iron core*) pada *relay* adalah komponen yang berfungsi sebagai peningkat medan magnet dalam *relay elektromagnetik*. Inti besi biasanya terbuat dari bahan *feromagnetik*, seperti besi atau baja yang memiliki sifat *feromagnetik* yang kuat. Inti besi pada *relay* berbentuk *silindris* atau berbentuk huruf E atau I tergantung pada desain *relay*. Inti besi ini dirancang untuk memaksimalkan konsentrasi medan magnet yang dihasilkan oleh kumparan *relay*.

Adapun untuk penempatan *pinout* modul *relay* dilihat pada gambar 2.3



Gambar 2. 4 Pinout Modul Relay

Berdasarkan gambar di atas, kita dapat memahami bahwa *relay* dapat bekerja karena adanya gaya *elektromagnetik*. Ini tercipta dari inti besi yang dililitkan kawat kumparan dan dialiri aliran listrik. Saat kumparan dialiri listrik, maka otomatis inti besi akan jadi magnet dan menarik penyangga sehingga kondisi yang awalnya tertutup jadi terbuka (*Open*). Sementara pada saat kumparan tak lagi dialiri listrik, maka pegas akan menarik ujung penyangga dan menyebabkan kondisi yang awalnya terbuka jadi tertutup (*Close*). Dapat dilihat pada gambar 2.5.



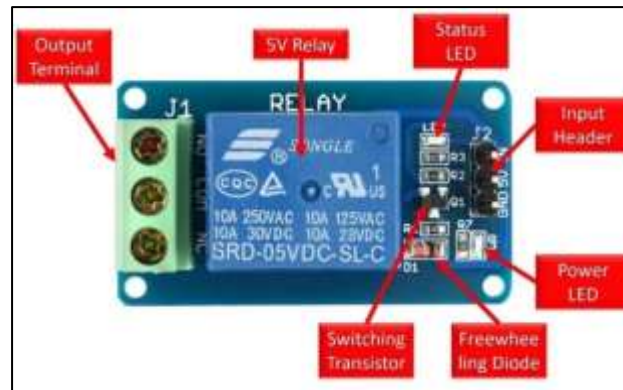
Gambar 2. 5 Tampilan skema rangkaian dari *Switch Contact*

Koneksi *switch relay* biasanya diberi label COM, NC dan NO:

- COM = *Common*, selalu terhubung ke ini, itu adalah bagian yang bergerak dari sakelar.
- NC = Biasanya Tertutup, COM terhubung ke ini ketika kumparan *relay* mati.
- NO = Biasanya Terbuka, COM terhubung ke ini ketika kumparan *relay* menyala.

2.5.4 Komponen Modul Relay

Gambar 2.6 berikut menggambarkan semua komponen dari modul *relay* satu saluran 5V.



Gambar 2. 6 *Komponen Modul Relay*

Tabel 2. 1 *Detail Komponen*

No	Komponen Relay Modul	Keterangan
1.	<i>5V Relay</i>	Pada inti modul ini terdapat sebuah <i>relay</i> 5V yang dilapisi dengan plastik berwarna biru. Arus dan tegangan operasi maksimum untuk beban AC dan DC juga tertera di atas penutup <i>relay</i> . SRD-05VDC-SL-C adalah nomor bagian dan menunjukkan tegangan operasinya. Disebut modul <i>relay</i> 5V karena <i>relay</i> ini beroperasi pada tegangan DC 5V. Dalam kata lain, sinyal aktif tinggi atau aktif rendah 5V mengaktifkan <i>relay</i> dengan mengenergikan koilnya. Seperti yang disebutkan sebelumnya, secara internal <i>relay</i> 5V terdiri dari terminal NC, NO, COM, dan <i>coil</i> .
2.	<i>Output Terminal</i>	Di sebelah kiri gambar terdapat <i>output terminal</i> yang digunakan untuk menghubungkan beban

		<p>DC/AC dan sumber daya masukan DC/AC. Kami akan membahas diagram pengawatan untuk menghubungkan beban dan sumber daya dengan terminal ini pada bagian berikutnya dari tutorial ini. Setiap terminal konektor <i>output</i> terhubung dengan pin NO, NC, dan COM dari <i>relay</i> 5V. Setiap titik pada modul ini memiliki sekrup yang memudahkan penghubungan kabel dengan modul <i>relay</i>. Modul <i>relay</i> satu saluran 5V ini mendukung arus <i>output</i> maksimum 10A dan tegangan kontak maksimum 250V AC dan 30V DC. Jika menggunakan tegangan AC yang tinggi dan beban arus tinggi dengan modul ini, gunakanlah kabel utama yang tebal.</p>
3.	Status LED	<p>LED status adalah LED SMD yang terhubung melalui <i>resistor</i> pembatas arus dan tersedia di sudut kanan atas modul. Ini menunjukkan status <i>relay</i>. Dengan kata lain, LED status menyala ketika <i>relay</i> aktif dan <i>coilnya</i> dienergikan melalui pin masukan sinyal. Arus DC melewati <i>coil relay</i>.</p>
4.	Power LED	<p>LED daya juga merupakan jenis LED SMD dan menunjukkan status sumber daya yang terhubung dengan modul <i>relay</i> satu saluran 5V. Jangan menghubungkan sumber daya lebih dari 5V ke pin Vcc dan GND modul ini. Jika tidak, tegangan yang lebih tinggi dapat merusak LED status dan LED daya.</p>

5.	<i>Freewheeling Diode</i>	<i>Freewheeling Diode</i> terhubung melintang pada koil untuk menghindari efek <i>back</i> EMF. Juga dikenal sebagai dioda <i>flyback</i> . <i>Coil</i> yang digunakan dalam <i>relay</i> ini adalah tipe induktif. Ketika arus melewati beban induktif, ia menghasilkan tegangan <i>back</i> EMF. <i>Back</i> EMF ini dapat merusak sirkuit. Oleh karena itu, dioda <i>freewheeling</i> digunakan untuk menghindari efek ini.
6.	<i>Input Connector</i>	Di sisi kanan modul <i>relay</i> terdapat konektor masukan. Digunakan untuk menyediakan sinyal masukan dan pasokan daya 5V. Selain itu, ini juga memberikan daya kepada LED status, LED daya, dan <i>coil relay</i> .
7.	<i>Switching Transistor</i>	Biasanya, kita memberikan sinyal masukan ke <i>relay</i> dari pin <i>input-output</i> tujuan umum <i>mikrokontroler</i> seperti <i>Arduino</i> , <i>TM4C123</i> , <i>ESP32</i> , dll. Namun, kemampuan arus sumber maksimum dari pin GPIO umumnya kurang dari 20mA. Oleh karena itu, transistor peralihan digunakan dalam modul <i>relay</i> ini untuk memperkuat arus hingga mencapai tingkat kebutuhan arus minimum pada <i>coil relay</i> . Dengan menggunakan transistor peralihan, kita dapat mengendalikan <i>relay</i> dari pin GPIO <i>mikrokontroler</i> .

2.6 MCB (Miniature Circuit Breaker)

MCB adalah pengaman rangkaian yang dilengkapi dengan pengaman *thermis (bimetal)* sebagai pengaman beban lebih dan juga dilengkapi *relay* elektromagnetik untuk pengaman hubung singkat. Keuntungan menggunakan MCB

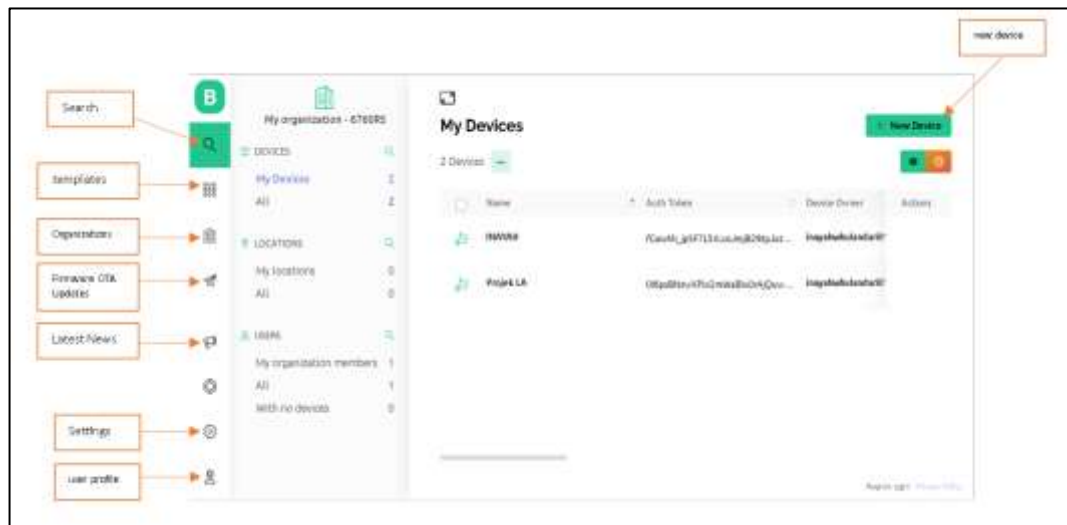
sebagai berikut: Dapat memutuskan rangkaian tiga fasa walaupun terjadi hubung singkat pada salah satu fasanya. Dapat digunakan kembali setelah rangkaian diperbaiki akibat hubung singkat atau beban lebih dan Mempunyai tanggapan yang baik apabila terjadi hubungsingkat atau beban lebih.

Pada MCB terdapat dua jenis pengaman yaitu secara *thermis* dan *elektromagnetis*, pengaman termis berfungsi untuk mengamankan arus beban lebih sedangkan pengaman *elektromagnetis* berfungsi untuk mengamankan jika terjadi hubung singkat. Pengaman *thermis* pada MCB memiliki prinsip yang sama dengan *thermal overload* yaitu menggunakan dua buah logam yang digabungkan (*bimetal*), pengamanan secara *thermis* memiliki kelambatan, ini bergantung pada besarnya arus yang harus diamankan, sedangkan pengaman elektromagnetik menggunakan sebuah kumparan yang dapat menarik sebuah angker dari besi lunak. Parawangsa, D. P. (2021).

2.7 Blynk

Blynk adalah *platform* untuk aplikasi OS *Mobile* (iOS dan *Android*) yang bertujuan untuk kendali *module Arduino, Raspberry Pi, ESP32, WEMOS D1, dan module* sejenisnya melalui *Internet*. Aplikasi ini merupakan wadah kreatifitas untuk membuat antarmuka grafis untuk proyek yang akan diimplementasikan hanya dengan metode *drag and drop widget*. *Blynk* tidak terikat pada papan atau *module* tertentu. Dari *platform* aplikasi inilah dapat mengontrol apapun dari jarak jauh, dimanapun kita berada dan waktu kapanpun. Dengan catatan terhubung dengan *internet* dengan koneksi yang stabil dan inilah yang dinamakan dengan sistem *Internet of Things (IoT)*. Camera, E. S. P. (2021).

Aplikasi *blynk* begitu populer untuk proyek-proyek berbasis IoT (*internet of things*). *Blynk* berbasis *Android* dan iOS, sangat *fleksibel* dalam penggunaannya. Aplikasi ini digunakan untuk mengendalikan *microcontroller*. Berikut adalah tampilan *dashboar Blynk* dapat dilihat pada gambar 2.6.



Gambar 2. 7 Tampilan dari *Dashboard Blynk*

Tabel 2. 2 Menu-menu yang ada pada *Dashboard Blynk*

No	Menu-menu yang ada pada <i>Dashboard Blynk</i>	
1.	<i>Search</i>	digunakan untuk mencari perangkat atau <i>node</i> yang terhubung dalam jaringan <i>Blynk</i>
2.	<i>Templates</i>	digunakan untuk membuat tampilan antarmuka yang sudah dirancang sebelumnya yang dapat digunakan kembali. <i>Template</i> adalah sekumpulan <i>widget</i> , konfigurasi, dan logika yang telah dipersiapkan sebelumnya untuk memudahkan pembuatan antarmuka pengguna.
3.	<i>Organizations</i>	<i>Blynk</i> adalah fitur yang memungkinkan pengguna untuk mengorganisasi dan mengelola proyek <i>Blynk</i> secara kolektif dalam satu entitas organisasi. Dengan <i>Organizations</i> , Anda dapat menghubungkan beberapa pengguna <i>Blynk</i> dan proyek-proyek mereka ke dalam satu grup yang terpusat.
4.	<i>Firmware OTA Updates</i>	<i>Firmware OTA (Over-the-Air) Updates</i> dalam

		<i>Blynk</i> adalah fungsi yang memungkinkan untuk memperbarui <i>firmware</i> perangkat keras secara <i>nirkabel</i> melalui jaringan <i>internet</i> . Dengan menggunakan <i>Firmware OTA Updates</i> , Anda dapat mengirimkan pembaruan <i>firmware</i> ke perangkat <i>Blynk</i> yang terhubung tanpa perlu memperbarui secara manual atau menghubungkan perangkat ke komputer.
5.	<i>Latest News</i>	" <i>Latest News</i> " (Berita Terbaru) mengacu pada fungsi atau fitur yang menyediakan pembaruan dan informasi terbaru tentang <i>platform Blynk</i> itu sendiri. Fitur ini biasanya dapat ditemukan di aplikasi <i>Blynk</i> atau di situs <i>web</i> resmi <i>Blynk</i> .
6.	<i>Settings</i>	<i>Settings</i> (Pengaturan) mengacu pada bagian atau menu yang memungkinkan pengguna untuk mengelola dan mengonfigurasi berbagai pengaturan yang terkait dengan akun <i>Blynk</i> dan proyek-proyek yang terhubung.
7.	<i>User Profile</i>	adalah bagian dari pengaturan akun yang memungkinkan pengguna untuk mengelola informasi pribadi dan <i>preferensi</i> yang terkait dengan akun <i>Blynk</i> mereka. Fungsi <i>User Profile</i> memberikan pengguna kontrol atas profil mereka dan memungkinkan mereka untuk mempersonalisasi pengalaman pengguna mereka di <i>platform Blynk</i> .
8.	<i>New Device</i>	dalam <i>Blynk</i> merujuk pada kemampuan untuk menambahkan perangkat baru ke akun <i>Blynk</i> . Fitur ini memungkinkan Anda untuk menghubungkan perangkat keras baru ke <i>Blynk</i>

		dan mengendalikannya melalui aplikasi <i>Blynk</i> .
--	--	--

2.8 Arduino IDE (*Integrated Development Environment*)

Arduino IDE adalah *software* yang digunakan untuk membuat *sketch* pemrograman atau dengan kata lain *arduino IDE* sebagai media untuk pemrograman pada *board* yang ingin diprogram. *Arduino IDE* ini berguna untuk mengedit, membuat, meng-*upload* ke *board* yang ditentukan, dan meng-*coding* program tertentu. *Arduino IDE* dibuat dari bahasa pemrograman *Java*, yang dilengkapi dengan *library C/C++(wiring)*, yang membuat operasi *input/output* lebih mudah.

Struktur pemrograman yang ada di *Arduino IDE* secara garis besar terbagi menjadi beberapa bagian seperti gambar di atas, yaitu:

1) Header

Header berisi *library* yang kita butuhkan. *Library-library* dasar yang ada dalam pemrograman bahasa C, sudah *terinput* otomatis dalam *Arduino IDE*. *Library* dasar seperti *stdio.h*, *stdlib.h*, *math.h*, kemudian *library mikrokontroler* yang digunakan dan masih banyak lagi. Sehingga, ketika memprogram di *Arduino IDE* dan perlu *library-library* dasar sudah tidak perlu dipanggil lagi di bagian *header*. *Library* yang sifatnya pengembangan, yang tidak termasuk *library* dasar maka harus dipanggil dibagian *header*. Misal *library* untuk LCD 16x2, HCSR04, dan masih banyak lagi. Cara penulisan *Header* dapat dilihat pada gambar 2.11.

```

1  #include <LiquidCrystal.h>
2  #include <HCSR04.h>

```

Gambar 2. 11 *Sktech Header*

2) Deklarasi Variabel

Deklarasi variabel yang dimaksud yaitu variabel global, yaitu variabel yang bisa digunakan diseluruh bagian program ini. Variabel terbagi menjadi dua, yaitu global dan lokal. Untuk deklarasi variabel global terletak dibagian ini,

tetapi untuk variabel lokal maka dideklarasikan di tiap awal fungsi/prosedur dimana variabel tersebut digunakan. Cara penulisan Deklarasi Variabel dapat dilihat pada gambar 2.12 sebagai berikut:

```
1 // Deklarasi Variabel Global
2 int bulat;
3 char s;
4 float pecahan;
```

Gambar 2. 12 *Sketch* Deklarasi Variabel Global

Cara penulisan deklarasi variabel lokal dapat dilihat pada gambar 2.13 sebagai berikut:

```
1 // Deklarasi Variabel Lokal
2 void nama_prosedur(){
3 int bulat;
4 char s;
5 float pecahan;
6
7 }
```

Gambar 2. 13 *Sketch* Deklarasi Variabel Lokal

3) Setup

Pada bagian ini, digunakan untuk mengkonfigurasi/mengatur *mikrokontroler* supaya sesuai kebutuhan pengguna. Pada dasarnya pin-pin yang ada pada *mikrokontroler* bisa digunakan sebagai masukan (*input*) atau keluaran (*output*), baik digital maupun analog. Maka, *mikrokontroler* harus diatur sebelum digunakan sesuai kebutuhan. Kegunaan lainnya yaitu untuk menjalankan program yang hanya sekali dijalankan seperti tampilan awal program, atau *init*. Cara menuliskan *Setup* dapat dilihat pada gambar 2.14 sebagai berikut:

```

2 void setup () {
3 // pinMode(pinDiatur, Mode)
4 pinMode(12, INPUT_PULLUP);
5 pinMode(13, OUTPUT);
6 }

```

Gambar 2. 14 *Sketch Setup*

4) Loop

Bagian *Loop* merupakan bagian yang isinya program utama yang akan dijalankan berulang-ulang. Program yang dijalankan sampai sumber tenaga (*power supply*) dicabut. Jika ingin membuat program menhidupkan LED yang dirangkai *active low*, maka untuk penulisan *Loop* dapat dilihat pada gambar 2.15 sebagai berikut:

```

2 void loop () {
3 // digitalWrite(pin, logika);
4 digitalWrite(ledPin, LOW);
5 }

```

Gambar 2. 15 *Sketch Loop*

2.7.2 Sintak dalam Penulisan Program

a. `//` (komentar 1 baris)

Digunakan untuk memberi komentar atau catatan pada kode-kode yang dibuat.

b. `/* */` (komentar 2 baris)

Untuk menuliskan catatan pada beberapa baris sebagai komentar.

c. `{ }` (kurung kurawal)

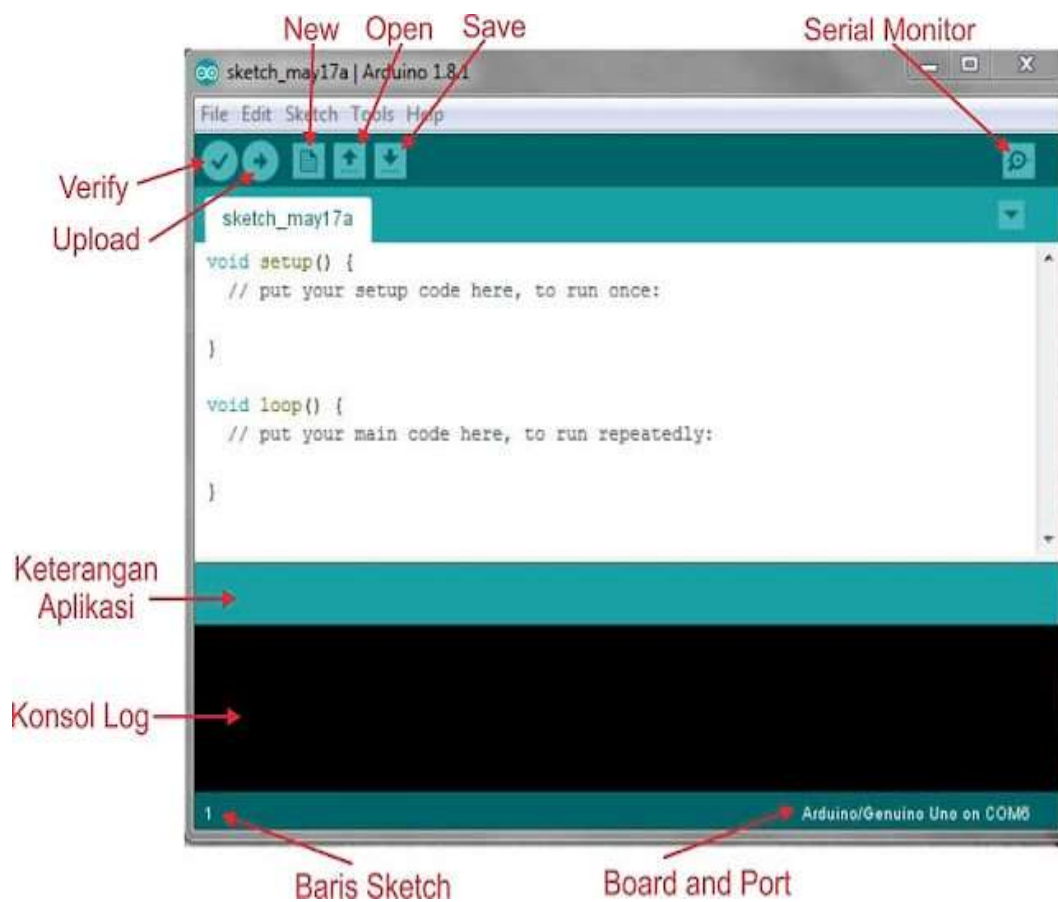
Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir serta digunakan juga pada fungsi dan pengulangan.

d. `;` (titik koma)

Setiap baris kode harus diakhiri dengan tanda ; (titik koma), jika ada titik koma yang hilang maka program tidak akan berjalan.

2.7.3 Pengenalan Tampilan Software Arduino IDE

Di *Arduino IDE*, program yang ditulis dinamakan *sketch*. Dimana *sketch* ini nantinya akan ditulis dalam suatu teks editor dan disimpan dengan *ekstensi .ino*. Layaknya teks editor pada umumnya. Teks editor di *Arduino IDE* juga memiliki fungsi dasar yang akan memudahkanmu dalam menuliskan program seperti *copy*, *.cut*, *paste*, *search*, dan *replacing*. Penjelasan *Menu Dasar Aplikasi Arduino IDE*. Gambar 2.16 berikut merupakan tampilan dari teks editor *Arduino IDE*.



Gambar 2. 16 Teks Editor *Arduino IDE*

Tabel 2. 3 Menu-menu yang ada pada *software Arduino IDE*

No	Menu-menu yang ada pada <i>software Arduino IDE</i>	
1.	<i>Verify</i>	Fungsi <i>verify</i> pada <i>Arduino</i> adalah memeriksa kode yang telah kamu buat. Apakah sudah sesuai kaidah pemrograman atau belum.
2.	<i>Upload</i>	Menu ini berfungsi untuk meng- <i>upload sketch</i> yang telah jadi ke <i>board Arduino</i> . Jika menggunakan menu ini maka otomatis <i>Arduino IDE</i> akan melakukan <i>verify</i> terlebih dahulu sebelum meng- <i>upload</i> .
3.	<i>New</i>	Berfungsi untuk membuat halaman <i>sketch</i> baru.
4.	<i>Open</i>	Fungsinya untuk membuka <i>sketch</i> yang pernah kamu buat dan simpan sebelumnya.
5.	<i>Save</i>	Berfungsi untuk menyimpan <i>sketch</i> yang telah kamu buat.
6.	<i>Serial Monitor</i>	Berfungsi untuk membuka <i>serial monitor</i> yang merupakan jendela untuk menampilkan data apa saja yang dikirimkan atau dipertukarkan antara <i>Arduino</i> dan <i>sketch</i> pada <i>port</i> serialnya. Biasanya <i>serial monitor</i> ini digunakan untuk melakukan <i>debugging</i> tanpa menggunakan LCD. Baik itu untuk menampilkan nilai proses, pembacaan, atau bahkan pesan <i>error</i> .
7.	Keterangan Aplikasi	Bagian ini berfungsi untuk menampilkan proses yang sedang berjalan di <i>Arduino IDE</i> dalam bentuk keterangan.
8.	<i>Konsol Log</i>	Adalah tempat munculnya informasi <i>error</i> pada saat <i>sketch</i> program di <i>verify</i> atau di <i>upload</i> . Tampilan <i>konsol log</i> berupa suatu <i>message box</i> berwarna hitam

		yang fungsinya untuk menampilkan status seperti pesan <i>error</i> , <i>compile</i> , serta <i>upload</i> program.
9.	<i>Baris Sketch</i>	Berisi keterangan tentang di baris ke berapa kursor diletakkan.
10.	<i>Board dan Port</i>	Berfungsi menampilkan informasi tentang <i>board</i> dan <i>port</i> apa yang sedang ter- <i>setting</i> di <i>Arduino IDE</i> .

2.8 Telegram

Telegram di dalam jurnal (Muhammad, R., dkk. 2023) adalah elegram adalah aplikasi pesan instan berbasis *cloud* yang fokus pada kecepatan dan keamanan. Telegram dirancang untuk memudahkan pengguna saling berkiriman pesan teks, suara, video, gambar, dan stiker dengan aman. Secara *default*, seluruh konten yang ditransfer akan dienkripsi berstandar internasional. Dengan demikian, pesan yang terkirim sepenuhnya aman dari pihak ketiga, bahkan dari telegram sekalipun. Bukan hanya teks, gambar, dan video. Telegram juga bisa jadi sarana Anda mengirimkan dokumen, musik, berkas zip, lokasi *real-time*, dan kontak yang tersimpan di perangkat ke orang lain. Jika, orang yang dituju juga mempunyai aplikasi dengan akun Telegram terdaftar di perangkatnya.

Bot telegram di dalam jurnal (Lenardo, G. C., dan Irawan, Y. 2020). Bot telegram *Application Programming Interface* (API) adalah sebuah teknologi open source yang disediakan oleh Telegram *Messenger* LLP untuk membangun aplikasi bot Telegram bagi para pengembang. Bot API ini merupakan *interface* berbasis HTTP untuk menghubungkan bot yang dikembangkan oleh para pengembang dengan sistem Telegram [3]. Kelebihan dari Telegram ini adalah adanya landasan untuk menggunakan *Application Programming Interface* (API) untuk masyarakat luas. Salah satu API yang disediakan adalah fitur bot. Bot Telegram adalah bot yang saat ini mulai populer dipergunakan.



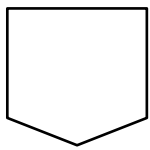
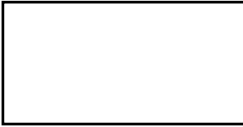
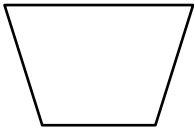
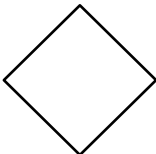


Gambar 2. 17 Logo Telegram


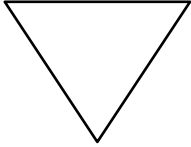


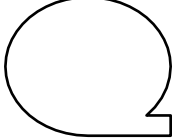

2.9 Flowchart


Flowchart atau sering disebut dengan diagram alir merupakan suatu jenis diagram yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada *programmer*. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. (Rosaly dan Prasetyo, 2019). Berikut tabel 2.5 adalah simbol-simbol yang digunakan dalam *flowchart* disertai dengan keterangan fungsinya:

Tabel 2. 4 *Flowchart* Simbol

NO	SIMBOL	KETERANGAN
1.		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalan arus suatu proses
2.		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama

3.		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4.		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) oleh <i>computer</i>
5.		Simbol <i>manual</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh <i>computer</i>
6.		Simbol <i>decision</i> , berfungsi untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak
7.		Simbol <i>terminal</i> , berfungsi untuk menyatakan permulaan atau akhir suatu program
8.		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal

9		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>
10.		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11.		Simbol <i>manual input</i> , berfungsi untuk memasukan data secara manual dengan menggunakan <i>online keybord</i>
12.		Simbol <i>input/ouput</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13.		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan input berasal dari pita <i>magnetis</i> atau <i>output</i> disimpan ke pita <i>magnetis</i>
14.		Simbol <i>disk storage</i> , berfungsi untuk menyatakan input berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>

15.		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui printer)
-----	---	---