

BAB II

LANDASAN TEORI

2.1 Buah Sawit dan Tingkat Kematangan

Buah sawit merupakan buah yang berasal dari tanaman *Elaeis guineensis Jacq.* yang menjadi sumber utama produksi minyak sawit mentah *Crude Palm Oil* (CPO) dan minyak inti sawit *Palm Kernel Oil* (PKO). Indonesia merupakan produsen minyak sawit terbesar di dunia sehingga kualitas buah yang dipanen sangat menentukan produktivitas dan mutu minyak yang dihasilkan. Buah sawit tersusun dalam tandan yang disebut *Fresh Fruit Bunch* (FFB), dimana setiap tandan terdiri atas ratusan hingga ribuan buah kecil (fruitlet). Kandungan minyak dalam buah sawit akan meningkat seiring bertambahnya tingkat kematangan, sehingga proses penentuan waktu panen menjadi salah satu faktor penting dalam meningkatkan rendemen minyak serta menjaga kualitas hasil pengolahan [4].

Secara fisiologis, proses pematangan buah sawit ditandai dengan perubahan warna kulit buah akibat degradasi klorofil dan peningkatan kandungan pigmen karotenoid. Pada fase awal, buah memiliki warna ungu kehitaman atau hitam, kemudian berubah menjadi merah keunguan, merah jingga, hingga jingga terang ketika mencapai kondisi matang. Perubahan warna tersebut merupakan indikator visual yang paling umum digunakan oleh petani maupun perusahaan perkebunan dalam menentukan tingkat kematangan buah. Selain perubahan warna, peningkatan kandungan minyak dan perubahan

komposisi kimia juga terjadi selama proses pematangan sehingga buah matang menghasilkan kualitas minyak yang lebih baik dibandingkan buah mentah [5].



Gambar 2.1 Tingkat Kematangan Buah Sawit

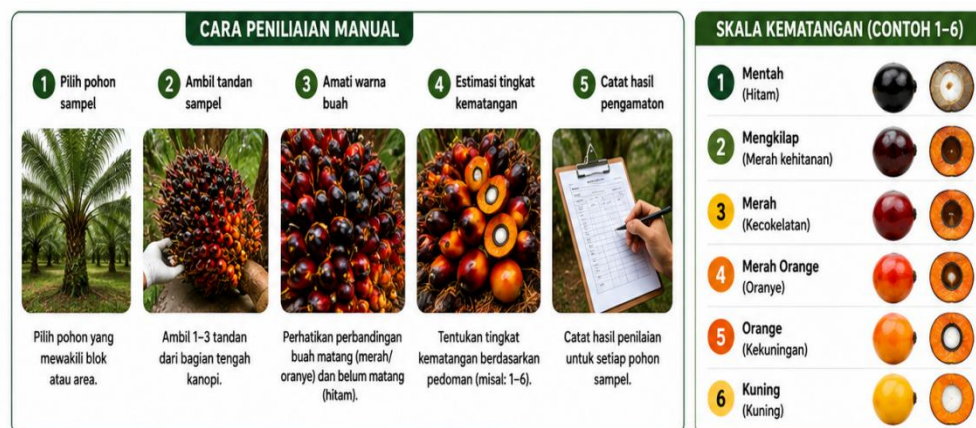
Tingkat kematangan buah sawit sangat mempengaruhi rendemen minyak *Oil Extraction Rate* (OER) dan kualitas *Crude Palm Oil* (CPO). Buah yang dipanen sebelum matang (*unripe*) memiliki kandungan minyak yang rendah karena proses akumulasi minyak belum berlangsung secara optimal. Sebaliknya, buah yang terlalu matang (*overripe*) cenderung mengalami peningkatan kadar asam lemak bebas *Free Fatty Acid* (FFA) akibat aktivitas enzim lipase setelah buah terlepas dari tandan. Kondisi tersebut menyebabkan penurunan mutu minyak serta meningkatkan biaya pengolahan di pabrik kelapa sawit. Oleh karena itu, pemanenan pada tingkat kematangan optimum menjadi langkah penting untuk memperoleh produktivitas dan kualitas minyak yang maksimal [6].

Dalam praktik di lapangan, tingkat kematangan buah sawit umumnya diklasifikasikan menjadi beberapa kategori, yaitu mentah (*unripe*), kurang matang (*underripe*), matang (*ripe*), dan lewat matang (*overripe*). Klasifikasi tersebut didasarkan pada kombinasi perubahan warna buah, jumlah brondolan (*loose fruit*) yang terlepas dari tandan, serta kondisi fisik tandan. Menurut pedoman yang banyak digunakan pada industri kelapa sawit, tandan matang umumnya ditandai dengan warna buah merah jingga dan mulai munculnya sejumlah brondolan di sekitar pangkal tandan. Pengelompokan tingkat kematangan ini menjadi dasar dalam pengembangan sistem otomatis berbasis sensor maupun pengolahan citra digital [7].

Seiring berkembangnya teknologi pertanian presisi (*precision agriculture*), proses identifikasi kematangan buah sawit mulai beralih dari metode konvensional menuju metode otomatis berbasis sensor dan kecerdasan buatan. Berbagai penelitian memanfaatkan sensor warna RGB, kamera digital, spektroskopi, sensor optik, hingga metode machine learning dan deep learning untuk meningkatkan akurasi klasifikasi tingkat kematangan. Pendekatan tersebut mampu mengurangi subjektivitas penilaian manusia, mempercepat proses identifikasi, serta mendukung implementasi sistem panen cerdas berbasis *Internet of Things* (IoT). Perkembangan ini menjadi landasan penting dalam penelitian mengenai rancang bangun alat ukur tingkat kematangan buah sawit menggunakan sensor warna berbasis IoT [8].

2.2 Metode Penilaian Kematangan Secara Manual

Metode penilaian kematangan buah sawit secara manual masih menjadi teknik yang paling banyak digunakan di perkebunan untuk menentukan waktu panen. Penilaian dilakukan melalui pengamatan langsung terhadap tandan buah segar *Fresh Fruit Bunch* (FFB), dengan memperhatikan perubahan warna kulit buah, jumlah brondolan yang telah terlepas, serta kondisi fisik tandan. Cara ini mudah diterapkan di lapangan karena tidak memerlukan peralatan khusus dan hanya mengandalkan pengalaman tenaga pemanen dalam mengenali tingkat kematangan buah [9].



Gambar 2.2 Penilaian Kematangan Secara Manual

Penentuan tingkat kematangan umumnya didasarkan pada perubahan warna buah dari hitam atau ungu gelap menjadi merah jingga hingga jingga terang. Selain itu, keberadaan brondolan yang jatuh secara alami juga menjadi indikator penting bahwa tandan telah siap dipanen sesuai standar operasional perkebunan. Kombinasi kedua indikator tersebut membantu pekerja menentukan waktu panen yang tepat [10].

Keunggulan metode manual terletak pada prosesnya yang sederhana, biaya penerapan yang rendah, dan kemudahan penggunaannya pada berbagai skala perkebunan. Namun, hasil penilaian sangat dipengaruhi oleh pengalaman pekerja sehingga berpotensi menimbulkan perbedaan penilaian. Faktor pencahayaan, kondisi cuaca, dan kelelahan saat bekerja juga dapat mengurangi ketepatan identifikasi, yang pada akhirnya berdampak pada rendemen minyak dan peningkatan kadar asam lemak bebas *Free Fatty Acid* (FFA) [11].

Selain itu, pemeriksaan secara manual memerlukan waktu lebih lama dan kurang efisien untuk perkebunan dengan jumlah tanaman yang besar. Oleh karena itu, berbagai penelitian mulai memanfaatkan sensor warna, pengolahan citra digital, machine learning, dan teknologi *Internet of Things* (IoT) untuk menghasilkan sistem identifikasi kematangan yang lebih cepat, objektif, dan konsisten. Teknologi tersebut diharapkan mampu meningkatkan akurasi penentuan waktu panen sekaligus mendukung produktivitas perkebunan kelapa sawit [12].

2.3 Konsep Sensor Warna untuk Deteksi Kematangan

Sensor warna merupakan perangkat elektronik yang berfungsi mengenali warna objek berdasarkan intensitas cahaya yang dipantulkan. Dalam bidang pertanian, sensor ini banyak digunakan untuk mengidentifikasi kualitas dan tingkat kematangan hasil panen karena mampu memberikan hasil yang lebih objektif dibandingkan pengamatan secara visual. Pada buah sawit, sensor warna dimanfaatkan untuk mendeteksi perubahan warna kulit buah yang

terjadi selama proses pematangan sehingga dapat dijadikan indikator dalam menentukan tingkat kematangan secara otomatis [13].

Prinsip kerja sensor warna dimulai dengan memancarkan cahaya ke permukaan objek, kemudian menangkap cahaya pantul untuk diubah menjadi data digital berupa komponen warna RGB (Red, Green, Blue). Data tersebut diproses oleh mikrokontroler untuk mengelompokkan tingkat kematangan buah. Pada beberapa penelitian, nilai RGB juga dikonversi ke ruang warna HSV (Hue, Saturation, Value) karena lebih stabil terhadap perubahan pencahayaan dan mampu meningkatkan akurasi klasifikasi [14].



Gambar 2.3 Deteksi Kematangan Buah Sawit

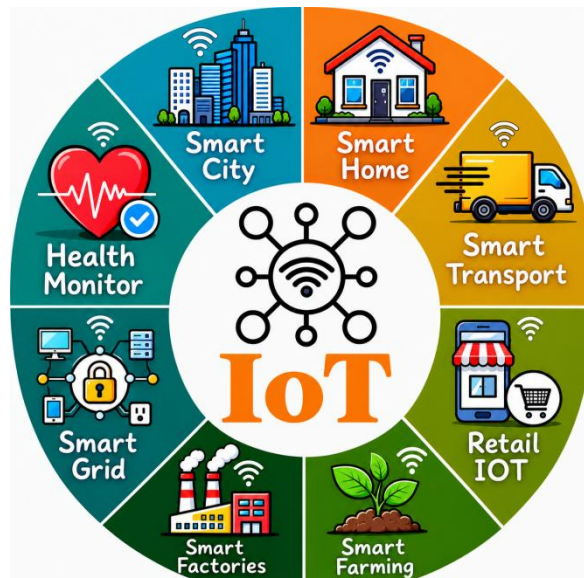
Salah satu sensor yang banyak digunakan adalah TCS3200, yang memiliki fotodiode dengan filter merah, hijau, biru, dan clear. Sensor ini mengubah intensitas warna yang diterima menjadi sinyal frekuensi yang kemudian diproses oleh mikrokontroler seperti ESP32 atau Arduino. Sensor TCS3200 dipilih karena harganya terjangkau, mudah diintegrasikan, serta memiliki sensitivitas yang baik dalam mendeteksi perubahan warna buah [15].

Perubahan tingkat kematangan buah sawit menyebabkan perubahan nilai RGB maupun HSV. Buah matang umumnya memiliki komponen merah yang lebih dominan dibandingkan buah mentah akibat meningkatnya kandungan pigmen karotenoid. Nilai tersebut digunakan sebagai dasar klasifikasi menggunakan metode threshold maupun algoritma seperti K-Nearest Neighbor (K-NN), K-Means, dan Support Vector Machine (SVM). Agar hasil pengukuran tetap akurat, penggunaan sensor perlu didukung dengan pencahayaan yang stabil, jarak pembacaan yang konsisten, serta proses kalibrasi. Integrasi sensor warna dengan teknologi Internet of Things (IoT) memungkinkan data dikirim dan dipantau secara real-time, sehingga proses penentuan waktu panen menjadi lebih cepat, akurat, dan efisien [16].

2.4 IoT (*Internet of Things*)

Internet of Things (IoT), juga dikenal sebagai *Internet of Everything*, *Cloud Computing*, atau *Industrial Internet*, adalah konsep teknologi modern yang merevolusi cara perangkat fisik berinteraksi. IoT menciptakan jaringan global yang menghubungkan berbagai perangkat, seperti sensor, mesin, dan alat elektronik, memungkinkan komunikasi, pertukaran data, serta otomatisasi antarperangkat. Teknologi ini mendukung terciptanya sistem yang lebih pintar, efisien, dan terintegrasi di berbagai aspek kehidupan [17].

Melalui IoT, perangkat-perangkat yang terhubung mampu mengumpulkan informasi dari lingkungannya, mengirimkan data ke pusat kontrol, dan menerima perintah untuk menjalankan fungsi tertentu. Contohnya, sensor suhu dapat mengirimkan data ke sistem pengatur suhu otomatis untuk menjaga kenyamanan ruangan. Proses ini dimungkinkan berkat koneksi internet, standar protokol komunikasi, serta teknologi cloud yang memproses data dengan cepat dan akurat, sehingga memudahkan pengelolaan perangkat secara efektif [18].



Gambar 2.4 *Internet of Things (IoT)*

IoT telah menjadi dasar dari berbagai inovasi modern, seperti rumah pintar, kota pintar, industri 4.0, dan transportasi pintar. Dalam dunia industri, IoT dikenal sebagai Industrial Internet, yang mendukung produktivitas melalui pemantauan waktu nyata dan otomatisasi sistem. Meskipun menawarkan banyak manfaat, IoT juga menghadapi tantangan, seperti keamanan data dan keselarasan antar perangkat. Namun, potensi besar yang dimilikinya

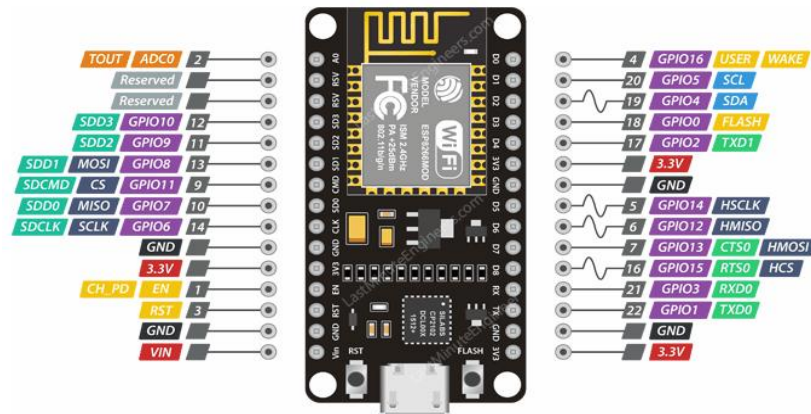
menjadikan IoT sebagai salah satu teknologi yang sangat menjanjikan untuk masa depan [19].

2.5 ESP32

ESP32 merupakan mikrokontroler yang dikembangkan oleh Espressif Systems dan banyak dimanfaatkan dalam pengembangan sistem Internet of Things (IoT) karena telah mengintegrasikan modul Wi-Fi dan Bluetooth pada satu perangkat. Integrasi tersebut memungkinkan komunikasi data dilakukan secara nirkabel tanpa memerlukan modul tambahan sehingga proses perancangan sistem menjadi lebih sederhana dan efisien. Selain itu, ESP32 memiliki prosesor dual-core 32-bit dengan performa yang baik serta mendukung berbagai antarmuka komunikasi, seperti UART, SPI, I2C, ADC, DAC, dan PWM, sehingga mudah dihubungkan dengan berbagai jenis sensor, termasuk sensor warna yang digunakan pada penelitian ini [20].

Pada sistem IoT, ESP32 berfungsi sebagai pengendali utama yang menerima data dari sensor, mengolah informasi berdasarkan program yang telah ditentukan, kemudian mengirimkan hasilnya ke server atau platform berbasis cloud melalui jaringan internet. Mekanisme tersebut memungkinkan data hasil pengukuran dapat dipantau secara real-time melalui aplikasi web maupun perangkat bergerak. Dalam penelitian ini, ESP32 digunakan untuk mengolah data dari sensor warna guna menentukan tingkat kematangan buah sawit, kemudian mengirimkan hasil identifikasi ke sistem monitoring sehingga

proses pengamatan menjadi lebih cepat, akurat, dan mengurangi kesalahan pencatatan secara manual [21].



Gambar 2.5 Microcontroller ESP32

Selain memiliki kemampuan komunikasi yang lengkap, ESP32 juga menawarkan konsumsi daya yang relatif rendah serta mendukung sistem operasi FreeRTOS yang memungkinkan beberapa proses dijalankan secara bersamaan (multitasking). ESP32 kompatibel dengan berbagai platform pengembangan, seperti Arduino IDE dan ESP-IDF, serta dapat diintegrasikan dengan layanan IoT, seperti MQTT, Firebase, dan Blynk. Kombinasi antara performa yang tinggi, kemudahan pengembangan, dan biaya implementasi yang relatif ekonomis menjadikan ESP32 sebagai salah satu mikrokontroler yang banyak digunakan dalam penelitian maupun pengembangan sistem monitoring berbasis Internet of Things [22].

2.6 Blynk

Blynk merupakan platform *Internet of Things* (IoT) yang digunakan untuk memantau dan mengendalikan perangkat elektronik melalui jaringan internet.

Platform ini menyediakan aplikasi seluler dan layanan cloud yang memungkinkan komunikasi data secara real-time antara pengguna dan perangkat IoT. Blynk mendukung berbagai mikrokontroler, seperti ESP8266, ESP32, Arduino, dan Raspberry Pi, sehingga banyak dimanfaatkan dalam pengembangan sistem monitoring dan kontrol berbasis IoT [23].

Seiring perkembangannya, Blynk Legacy telah diperbarui menjadi Blynk IoT (Blynk 2.0) yang menawarkan berbagai fitur tambahan, seperti manajemen perangkat, dashboard berbasis web, penyimpanan data pada cloud, notifikasi, dan visualisasi data sensor secara langsung. Mekanisme kerjanya dilakukan dengan mengirimkan data dari mikrokontroler ke server Blynk melalui internet, kemudian data tersebut ditampilkan pada aplikasi pengguna. Selain itu, pengguna juga dapat mengirimkan perintah dari aplikasi untuk mengendalikan perangkat yang terhubung [24].



Gambar 2.6 Platform Blynk

Arsitektur Blynk terdiri dari tiga komponen utama, yaitu perangkat keras (*hardware*), *server cloud*, dan aplikasi pengguna. Perangkat keras berfungsi untuk membaca data sensor dan mengontrol aktuator, sedangkan *server cloud* bertugas sebagai media pertukaran data antara perangkat dan aplikasi. Struktur

ini memungkinkan komunikasi dua arah yang cepat dan efisien dalam implementasi sistem IoT [25].

Berbagai penelitian menunjukkan bahwa Blynk memiliki keunggulan dalam kemudahan penggunaan dan fleksibilitas implementasi. Pada penelitian sistem kendali lampu berbasis NodeMCU ESP8266, Blynk mampu memberikan kontrol perangkat secara jarak jauh melalui smartphone dengan respons yang baik dan biaya implementasi yang relatif rendah. Selain itu, aplikasi ini juga banyak digunakan untuk monitoring data sensor secara real-time, seperti pemantauan konsumsi daya listrik menggunakan sensor PZEM-004T yang dapat menampilkan nilai tegangan, arus, dan daya melalui smartphone [26].

Pemanfaatan Blynk juga telah berkembang pada sektor pertanian dan peternakan. Sistem irigasi berbasis IoT yang terintegrasi dengan Blynk mampu melakukan pemantauan kondisi lahan dan pengendalian pompa air dari jarak jauh sehingga meningkatkan efisiensi penggunaan air. Pada bidang peternakan, Blynk digunakan untuk memonitor suhu dan kelembapan kandang ayam secara real-time, sehingga kondisi lingkungan dapat diawasi dengan lebih efektif. Dalam penelitian ini, Blynk digunakan sebagai media monitoring hasil pembacaan sensor warna TCS3200 untuk menentukan tingkat kematangan buah sawit. Data yang diperoleh dari sensor dikirim oleh NodeMCU ESP8266 ke server Blynk dan ditampilkan pada smartphone secara real-time. Dengan demikian, pengguna dapat memantau hasil identifikasi kematangan buah sawit dari mana saja selama terhubung ke internet.

Berdasarkan berbagai penelitian terdahulu, Blynk terbukti menjadi platform yang efektif dan mudah diimplementasikan untuk mendukung sistem monitoring berbasis IoT [27].

2.7 Sensor Warna TCS3200

Sensor warna TCS3200 merupakan sensor optik yang digunakan untuk mengenali warna objek berdasarkan intensitas cahaya yang dipantulkan. Sensor ini mengubah cahaya menjadi sinyal frekuensi yang dapat diproses oleh mikrokontroler seperti Arduino maupun ESP32. Karena memiliki harga yang terjangkau, mudah digunakan, dan cukup akurat, sensor TCS3200 banyak dimanfaatkan pada sistem otomasi, termasuk deteksi tingkat kematangan buah [28].



Gambar 2.7 TCS3200

Sensor TCS3200 terdiri atas 64 photodiode yang terbagi menjadi empat kelompok, yaitu filter merah, hijau, biru, dan tanpa filter (clear). Pemilihan filter dilakukan melalui pin kontrol S2 dan S3, sedangkan skala frekuensi diatur menggunakan pin S0 dan S1. Keluaran sensor berupa sinyal frekuensi

yang merepresentasikan intensitas warna objek. Prinsip kerja sensor dimulai dengan memancarkan cahaya ke permukaan objek, kemudian menangkap cahaya pantul menggunakan photodiode. Intensitas cahaya yang diterima diubah menjadi sinyal frekuensi dan diproses oleh mikrokontroler menjadi nilai RGB (Red, Green, Blue). Nilai tersebut selanjutnya digunakan sebagai dasar untuk mengidentifikasi warna atau tingkat kematangan objek [29].

Dalam deteksi kematangan buah, sensor TCS3200 membaca perubahan nilai RGB yang terjadi selama proses pematangan. Perbedaan nilai warna tersebut dimanfaatkan untuk mengelompokkan tingkat kematangan secara otomatis. Berbagai penelitian menunjukkan bahwa sensor ini mampu meningkatkan akurasi proses penyortiran buah dibandingkan metode visual. Kinerja sensor TCS3200 dipengaruhi oleh beberapa faktor, seperti intensitas cahaya, jarak sensor terhadap objek, dan posisi pembacaan. Oleh karena itu, diperlukan proses kalibrasi, pengaturan jarak yang tetap, serta penggunaan pencahayaan yang stabil agar hasil pengukuran lebih akurat dan konsisten [30].

Pada penelitian ini, sensor TCS3200 digunakan untuk memperoleh data warna buah sawit yang kemudian diproses oleh mikrokontroler ESP32. Hasil identifikasi tingkat kematangan selanjutnya dikirim melalui sistem Internet of Things (IoT) sehingga dapat dipantau secara real-time. Penggunaan sensor ini diharapkan mampu meningkatkan kecepatan dan objektivitas dalam menentukan tingkat kematangan buah sawit [31].

2.8 LCD 2004 (20×4)

LCD 2004 (20×4) merupakan modul *Liquid Crystal Display* (LCD) yang mampu menampilkan 20 karakter pada 4 baris, sehingga lebih banyak menampilkan informasi dibandingkan LCD 16×2. Modul ini banyak digunakan pada sistem berbasis mikrokontroler seperti Arduino dan ESP32 untuk menampilkan hasil pembacaan sensor, menu pengaturan, maupun status kerja alat secara real-time. Pada umumnya LCD 20×4 menggunakan antarmuka I2C, sehingga hanya memerlukan dua jalur komunikasi, yaitu SDA (Serial Data) dan SCL (Serial Clock), yang membuat proses pengkabelan menjadi lebih sederhana dan efisien [32].



Gambar 2.8 *Liquid Crystal Display* (LCD)

2.9 Buzzer Speaker

Buzzer speaker merupakan komponen elektronika yang berfungsi sebagai penghasil suara atau alarm pada suatu sistem. Komponen ini bekerja dengan mengubah sinyal listrik menjadi gelombang suara sebagai indikator kondisi tertentu, seperti peringatan, notifikasi, atau tanda bahwa suatu proses telah

selesai. Dalam sistem berbasis mikrokontroler seperti Arduino maupun ESP32, buzzer sering digunakan karena mudah diintegrasikan, memiliki konsumsi daya yang rendah, serta mampu memberikan umpan balik audio secara cepat kepada pengguna [33].



Gambar 2.9 Buzzer Speaker

Pada berbagai aplikasi *Internet of Things* (IoT) dan sistem otomasi, buzzer speaker dimanfaatkan sebagai media peringatan ketika sensor mendeteksi kondisi tertentu, misalnya suhu melebihi batas, ketinggian air meningkat, atau objek terdeteksi. Penggunaan buzzer membantu meningkatkan efektivitas sistem monitoring karena pengguna dapat segera mengetahui adanya perubahan kondisi tanpa harus selalu memperhatikan tampilan LCD atau aplikasi. Oleh karena itu, buzzer menjadi salah satu komponen output yang banyak digunakan pada penelitian dan pengembangan sistem elektronik berbasis mikrokontroler di Indonesia [34].

2.10 LED 5 mm

LED (*Light Emitting Diode*) 5 mm merupakan komponen elektronika yang berfungsi sebagai indikator maupun sumber cahaya pada rangkaian elektronik.

LED ini memiliki konsumsi daya rendah, umur pakai panjang, respon cepat, serta menghasilkan panas yang kecil sehingga banyak digunakan pada sistem berbasis Arduino, ESP32, IoT, dan perangkat otomasi. Dalam penggunaannya, LED 5 mm dipasang bersama resistor pembatas arus untuk mencegah kerusakan akibat arus berlebih. Dengan tegangan kerja sekitar 1,8–3,3 V dan arus 10–20 mA, LED 5 mm menjadi komponen yang sederhana, ekonomis, dan mudah diintegrasikan pada berbagai prototipe maupun alat elektronika [35].



Gambar 2.10 LED (*Light Emitting Diode*)

2.11 Kabel Jumper

Kabel jumper adalah kabel penghubung yang digunakan untuk menghubungkan antar komponen elektronika tanpa proses penyolderan. Kabel ini banyak dimanfaatkan pada perancangan prototipe berbasis Arduino maupun ESP32 karena memudahkan koneksi sensor, modul, dan aktuator pada breadboard. Selain itu, kabel jumper tersedia dalam tipe male-to-male, male-to-female, dan female-to-female sesuai kebutuhan rangkaian [36].

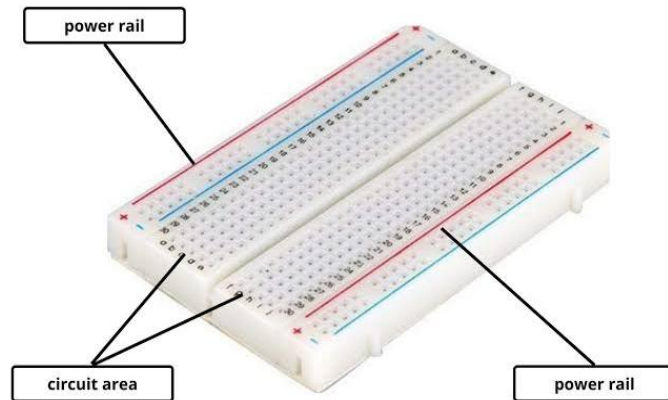


Gambar 2.11 Kabel Jumper

Kabel jumper berfungsi menjaga koneksi data dan catu daya antar komponen agar rangkaian bekerja dengan baik. Penggunaannya mempermudah proses perakitan, pengujian, dan mengurangi kesalahan pemasangan, sehingga menjadi salah satu komponen penting dalam proyek elektronika dan IoT [37].

2.12 Mini Breadboard

Mini breadboard merupakan papan perakitan rangkaian elektronik tanpa proses penyolderan (*solderless breadboard*) yang digunakan untuk merancang, menguji, dan memodifikasi rangkaian secara cepat. Komponen seperti resistor, LED, sensor, kabel jumper, serta mikrokontroler (Arduino atau ESP32) dapat dipasang dan dilepas dengan mudah sehingga sangat cocok untuk proses prototipe, praktikum, maupun pengembangan sistem IoT. Ukurannya yang ringkas membuat mini breadboard mudah dibawa dan banyak digunakan pada proyek elektronika skala kecil [38].



Gambar 2.12 Mini Breadboard

2.13 Kabel Micro USB

Kabel Micro USB digunakan untuk menghubungkan perangkat elektronik dengan sumber daya maupun komputer untuk pengisian daya dan transfer data. Pada sistem berbasis mikrokontroler seperti Arduino dan ESP32, kabel ini berfungsi sebagai media komunikasi serial sekaligus sumber catu daya saat proses pemrograman dan pengujian perangkat [39].



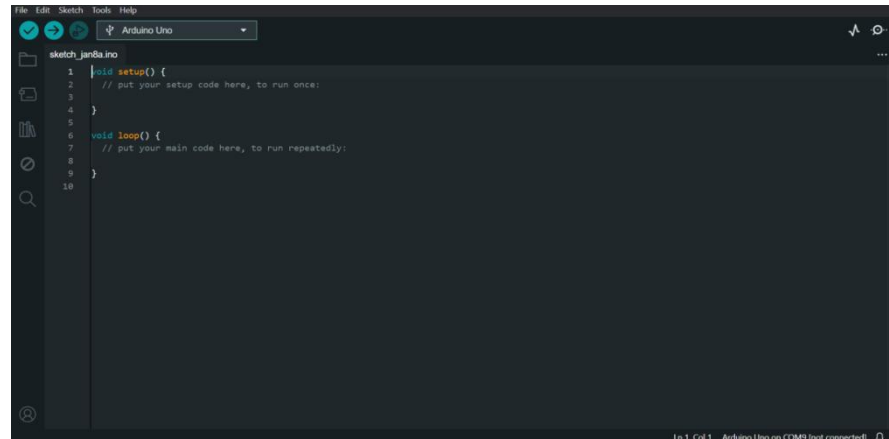
Gambar 2.13 Kabel Micro USB

2.14 Software Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah perangkat lunak sumber terbuka yang dirancang khusus untuk memprogram papan mikrokontroler Arduino dan berbagai perangkat berbasis *mikrokontroler* lainnya, seperti ESP8266, ESP32, dan papan lainnya yang mendukung Arduino. Arduino IDE memungkinkan pengguna untuk menulis, mengedit, dan mengunggah kode program ke perangkat keras dengan cara yang sederhana dan mudah dipahami. Salah satu alasan mengapa Arduino IDE begitu populer adalah antarmuka yang ramah pengguna, sehingga dapat digunakan oleh pemula sekalipun tanpa perlu pengalaman mendalam dalam pemrograman. Platform ini menggunakan bahasa pemrograman berbasis C/C++, yang memberi pengembang fleksibilitas dalam menulis kode untuk berbagai aplikasi dan proyek [40].

Arduino IDE hadir dengan berbagai fitur yang memudahkan pengguna dalam menulis dan mengelola kode program. Fitur utama yang disediakan adalah editor kode yang mendukung penyorotan sintaksis (*syntax highlighting*) untuk memudahkan pembacaan kode, pelengkapan otomatis (*auto-completion*), serta pengecekan kesalahan kompilasi yang membantu pengguna untuk menemukan dan memperbaiki kesalahan pada kode sebelum diunggah ke perangkat. Arduino IDE juga menyediakan alat untuk memprogram perangkat dengan mudah, melalui tombol "*Upload*" yang langsung mengunggah program ke papan mikrokontroler melalui kabel USB atau

koneksi nirkabel. Dengan antarmuka yang minimalis, pengguna tidak akan merasa kesulitan untuk memulai pemrograman [41].



Gambar 2.14 *Software Arduino IDE*

Salah satu kekuatan utama Arduino IDE adalah pustaka (*library*) yang luas dan terus berkembang, yang memungkinkan pengguna untuk menambahkan fungsi tambahan ke dalam proyek mereka. Pustaka ini mencakup berbagai jenis modul perangkat keras seperti sensor suhu, sensor gerak, motor, layar LCD, serta modul komunikasi seperti Wi-Fi dan Bluetooth. Dengan menggunakan pustaka ini, pengguna tidak perlu menulis kode dari awal untuk menangani perangkat keras tertentu, sehingga menghemat waktu dan mempermudah proses pengembangan. Selain itu, Arduino IDE juga menyediakan berbagai contoh program (*sketches*) yang dapat digunakan sebagai referensi atau titik awal dalam membuat aplikasi yang lebih kompleks [42].

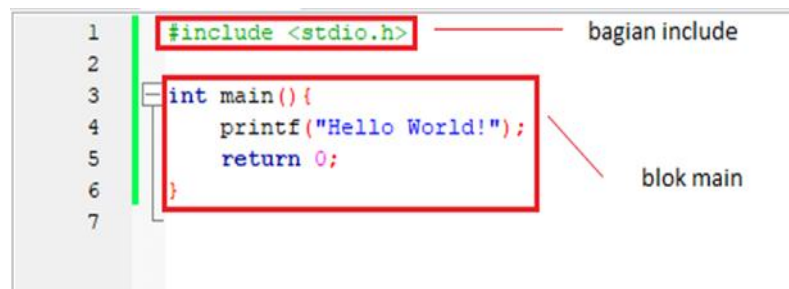
Arduino IDE juga mendukung berbagai perangkat keras lain yang kompatibel dengan Arduino, seperti ESP8266, ESP32, dan papan mikrokontroler lainnya. Untuk menggunakan perangkat keras selain Arduino,

pengguna cukup menginstal pustaka dan board manager yang sesuai. Misalnya, untuk memprogram ESP32, pengguna dapat mengunduh board manager khusus yang memungkinkan IDE untuk mengenali dan mengunggah kode ke papan ESP32. Hal ini memungkinkan pengembang untuk memanfaatkan potensi penuh dari berbagai *mikrokontroler*, seperti kemampuan Wi-Fi dan Bluetooth pada ESP32, atau komunikasi jarak jauh dengan ESP8266. Arduino IDE menjadi platform pilihan utama bagi banyak pengembang hobi dan profesional dalam membangun berbagai jenis proyek elektronik dan IoT, mulai dari robotika, alat pemantauan lingkungan, hingga perangkat otomasi rumah pintar. Kemudahan penggunaan, fleksibilitas, dan ekosistem yang luas menjadikan Arduino IDE alat yang sangat efisien untuk *prototyping* cepat dan pengembangan perangkat keras. Dengan terus berkembangnya teknologi dan dukungan dari komunitas yang kuat, Arduino IDE tetap menjadi pilihan utama dalam dunia pengembangan elektronik [43].

2.15 Bahasa Pemrograman

Bahasa pemrograman adalah sekumpulan aturan dan sintaksis yang digunakan untuk menulis instruksi yang dapat dipahami dan dijalankan oleh komputer atau perangkat keras lainnya. Dalam dunia pengembangan perangkat lunak dan perangkat keras, bahasa pemrograman menjadi alat utama yang memungkinkan manusia untuk berkomunikasi dengan mesin. Dengan menggunakan bahasa pemrograman, *programmer* dapat membuat aplikasi, sistem, dan perangkat yang dapat melaksanakan tugas tertentu secara otomatis.

Bahasa pemrograman dapat berupa bahasa tingkat tinggi, yang lebih mudah dipahami manusia, atau bahasa tingkat rendah, yang lebih mendekati bahasa mesin [44].



Gambar 2.15 Bahasa Pemrograman C

Bahasa pemrograman dibagi menjadi berbagai jenis, tergantung pada tujuan dan fungsinya. Beberapa contoh bahasa pemrograman tingkat tinggi yang sering digunakan antara lain Python, Java, C++, dan JavaScript. Bahasa-bahasa ini lebih mudah dipelajari karena sintaksisnya yang mirip dengan bahasa manusia dan dapat digunakan untuk berbagai jenis aplikasi, mulai dari aplikasi desktop, aplikasi web, hingga perangkat IoT. Selain itu, ada juga bahasa pemrograman yang lebih khusus, seperti bahasa pemrograman untuk mikrokontroler seperti C/C++ yang digunakan pada *platform* Arduino, atau Python untuk pengembangan AI dan analisis data. Bahasa pemrograman dapat digunakan untuk berbagai tujuan, mulai dari pengembangan perangkat lunak, aplikasi mobile, hingga pengendalian perangkat keras seperti robot atau sistem IoT. Beberapa bahasa pemrograman lebih cocok untuk aplikasi tertentu. Misalnya, Python sering digunakan untuk pengembangan aplikasi berbasis web dan analisis data, sedangkan C dan C++ banyak digunakan dalam pengembangan perangkat keras dan aplikasi yang membutuhkan pengolahan

cepat. Selain itu, ada pula bahasa pemrograman yang lebih terfokus pada pengembangan sistem operasi atau aplikasi yang sangat efisien dalam pengelolaan sumber daya komputer, seperti *Assembly* dan Rust [45].



```

1 #include <iostream>
2
3 int main()
4 {
5
6     int x = 10;
7     int nilai_awal_array[20];
8     const char *str = "Hello, World!";
9
10    return 0;
11 }

```

Gambar 2.16 Bahasa Pemrograman C++

Proses pemrograman melibatkan penulisan kode menggunakan bahasa pemrograman tertentu, yang kemudian diterjemahkan oleh *compiler* atau interpreter menjadi instruksi yang dapat dijalankan oleh komputer. Di setiap tingkat bahasa pemrograman, ada alat dan perangkat lunak yang membantu programmer menulis dan menguji kode mereka. Misalnya, *Integrated Development Environment* (IDE) seperti Arduino IDE atau Visual Studio Code menyediakan alat-alat yang mempermudah penulisan, *debugging*, dan pengujian kode. Meskipun ada banyak bahasa pemrograman yang tersedia, pilihan bahasa biasanya bergantung pada jenis aplikasi yang ingin dibuat, tingkat kesulitan, dan kebutuhan performa yang diinginkan [46].

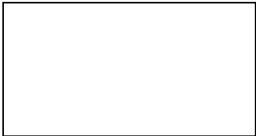
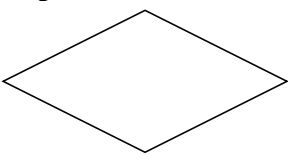
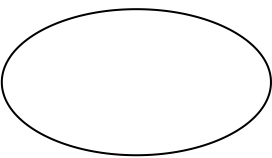
2.16 Flowchart

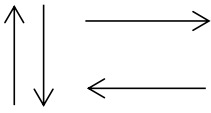
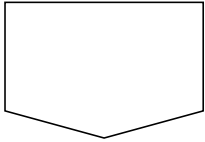
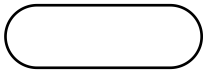
Flowchart adalah suatu bagan dengan simbol simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu

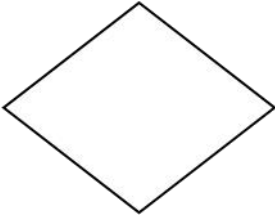
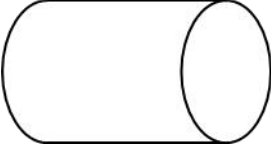
proses (instruksi) dengan proses lainnya dalam suatu program[47]. *Flowchart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah serta ada dua macam *flowchart* yang menggambarkan proses dengan computer yaitu :

- a. *System Flowchart* Bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.
- b. *Program Flowchart* Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program [48].

Table 2.1 Simbol-Simbol *Flowchart*

Simbol	Deskripsi
Entitas 	Suatu kumpulan objek atau sesuatu yang dapat dibedakan atau dapat didefinisikan secara unik
Relationship 	Hubungan yang terjadi antara satu entitas atau lebih
Atribut 	Karakteristik dari entitas atau <i>Relationship</i> yang menyediakan penjelasan detail entitas atau <i>relation</i>

Simbol	Deskripsi
Link 	Baris sebagai penghubung antara himpunan, relasi dan himpunan entitas dari atributnya
Flow Line 	Garis yang menghubungkan antar simbol-simbol lainnya pada <i>flowchart</i> dan menunjukkan arah alir <i>flowchart</i> tertentu
Off Page Connector 	Simbol untuk menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman/lembar yang berbeda
Terminal 	Menandakan awal atau akhir dan suatu <i>flowchart</i>
Input-Output 	Simbol untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
Process 	Simbol untuk proses perhitungan atau proses pengolahan data

Simbol	Deskripsi
Predefined Process (Sub Program) 	Permulaan sub program atau proses menjalankan sub program
Decision 	Perbandingan pernyataan, penyelesaian data yang memberikan pilihan untuk langkah selanjutnya
Disk Magnetik 	Data disimpan secara permanen di dalam disk magnetik, digunakan sebagai master file dan database