

BAB II

KERANGKA TEORITIS

A. Landasan Teori

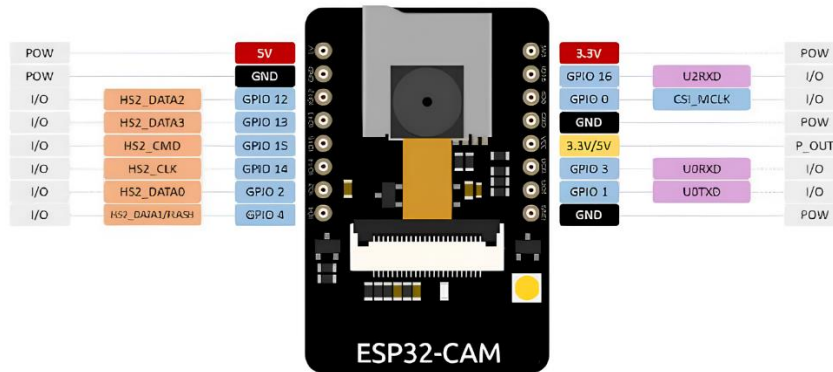
1. Internet of Things (IoT)

Di era teknologi yang semakin maju, Internet of Things (IoT) digunakan untuk mengintegrasikan dan mengoptimalkan berbagai perangkat dalam kehidupan sehari-hari. Menurut Yudhanto, Y., & Azis (2019, p. 17) mengungkapkan bahwa Internet of Things (IoT) adalah konsep dimana objek dapat mengirim data melalui jaringan Wi-Fi tanpa memerlukan interaksi manusia dan proses ini dijalankan secara otomatis oleh program IoT. Hal ini menjadikan IoT sebagai teknologi yang mampu mengintegrasikan berbagai perangkat untuk menciptakan sistem yang cerdas dan saling terhubung.

IoT mentransmisikan data menggunakan berbagai protokol komunikasi seperti MQTT dan teknologi jaringan seperti Wi-Fi dan Bluetooth, yang memungkinkan komunikasi baik dalam jarak pendek maupun jauh (Kurniati et al., 2024, p. 3). Gateway IoT berfungsi sebagai penghubung antara perangkat dan jaringan yang lebih luas, mengumpulkan serta memproses data sebelum mengirimkannya ke Cloud untuk analisis (Kurniati et al., 2024, p. 3). Cloud menyediakan sumber daya untuk penyimpanan dan analisis data dalam skala besar, dengan layanan seperti AWS IoT dan Google Cloud IoT, serta memastikan keamanan data melalui enkripsi dan autentikasi (Kurniati et al., 2024, p. 3).

2. ESP32-CAM

Menurut Siswoyo et al (2024, p. 1) *ESP32-CAM* adalah modul kamera kecil dan ekonomis dengan kemampuan untuk mengambil gambar, merekam video, serta mendukung streaming video. Modul ini menggunakan prosesor dual-core ESP32 berbasis Xtensa 32-bit LX6 yang mampu bekerja hingga 240 MHz, dilengkapi dengan 520 KB SRAM dan 4 MB PSRAM, serta kamera OV2640 dengan resolusi maksimal 1600x1200 piksel yang mendukung berbagai resolusi dari VGA hingga UXGA (Siswoyo et al., 2024, pp. 2–3). Selain itu, ESP32-CAM juga dilengkapi dengan slot kartu MicroSD yang mendukung kapasitas hingga 4 GB, memiliki 9 pin GPIO, dan desain kompak dengan ukuran 27 × 40,5 × 4,5 mm, sehingga menjadikannya solusi yang sesuai untuk sistem pemantauan berbasis pengolahan data visual (Siswoyo et al., 2024, pp. 2–3).



Gambar 2. 1 Pin Input Output ESP32-CAM

“Sumber: <https://makeradvisor.com> dalam (Siswoyo et al., 2024, p. 4)”

Keunggulan spesifikasi tersebut menjadikan ESP32-CAM sangat fleksibel untuk berbagai kebutuhan pemantauan visual berbasis IoT. Kombinasi kemampuan pemrosesan yang baik, konektivitas nirkabel, dan dukungan penyimpanan lokal melalui kartu MicroSD menjadikan modul ini sangat tepat untuk sistem yang memerlukan pengambilan dan pengiriman data secara cepat (Siswoyo et al., 2024, p. 5). Selain itu, kemudahan konfigurasi menggunakan Arduino IDE memungkinkan pengguna mengembangkan aplikasi berbasis ESP32-CAM dengan cepat dan praktis (Siswoyo et al., 2024, p. 6).

3. **Computer vision**

Menurut Marpaung et al (2022, p. 53) menyatakan bahwa *Computer Vision* sebagai bagian dari pembelajaran mendalam (*Deep learning*) dan kecerdasan buatan (*Artificial Intelligence*) yang memungkinkan komputer meniru kemampuan mata manusia dalam mengenali dan menafsirkan objek disekitarnya. Menurut Marpaung et al (2022, p. 55) cara kerja *Computer Vision* adalah mengekstraksi informasi secara otomatis dari gambar, yang meliputi pemahaman tentang model 3D, posisi kamera, serta kemampuan untuk mendeteksi dan mengenali objek, selain itu juga dapat mengelompokkan dan mencari konten gambar. Menurut Marpaung et al (2022, p. 54) penerapan *Computer Vision* dibagi menjadi tiga kategori sebagai berikut:

- (1) *low-level vision*, pada kategori awal ini berfokus pada pemahaman komputer terhadap gambar sebagai objek digital, dengan tugas seperti menghilangkan derau, memberi filter, dan klasifikasi tekstur permukaan;
- (2) *intermediate-level vision*, pada kategori ini mesin dapat mendeteksi garis, tepi, pola, dan melakukan pencocokan pola objek;

- (3) *3-d vision* dan gerakan, dimana komputer dapat mengenali objek bervolume dan objek bergerak, serta mengidentifikasi bentuk dan bayangan yang dihasilkan dari cahaya.

Kemampuan *Computer Vision* dalam mengenali objek sangat bergantung pada tahapan pengolahan citra yang menekankan pada identifikasi karakteristik visual seperti bentuk, warna, dan tekstur. Teknologi ini memanfaatkan metode segmentasi, pendeteksian tepi, serta ekstraksi fitur untuk memperoleh informasi penting dari gambar secara menyeluruh (Marpaung et al., 2022, p. 55). Penerapannya meluas ke berbagai sektor seperti sistem pengawasan digital, analisis kesehatan berbasis citra, sistem pengenalan wajah, dan pelacakan objek dalam video (Marpaung et al., 2022, pp. 61–63).

4. Pengertian Pengolahan Citra

Menurut Marpaung et al (2022, p. 3) pemrosesan dan pengolahan citra merupakan teknik yang digunakan untuk melakukan berbagai operasi pada gambar dengan tujuan meningkatkan kualitas gambar atau mengekstrak informasi yang relevan. Sebagaimana diungkapkan oleh Marpaung et al (2022, p. 3), proses ini termasuk dalam pemrosesan sinyal, di mana gambar berfungsi sebagai input, dan keluaran yang dihasilkan dapat berupa gambar baru atau fitur yang terkait. Maka dari itu, pengolahan citra memiliki peran yang signifikan dalam analisis dan manipulasi data visual.

Citra digital yang melalui sistem pengolahan citra umumnya melewati beberapa tahapan, seperti peningkatan kejernihan gambar, pengurangan gangguan visual (*noise*), dan penajaman tepi untuk mempermudah identifikasi fitur penting (Marpaung et al., 2022, p. 10). Peningkatan kualitas gambar dilakukan dengan memperjelas perbedaan kecerahan antara objek dan latar belakang, sedangkan gangguan visual dapat dikurangi melalui teknik perataan citra agar hasil tampilan menjadi lebih bersih (Marpaung et al., 2022, pp. 10–11). Deteksi tepi digunakan untuk memisahkan objek dari latar belakang dengan mengenali perubahan intensitas piksel yang tajam, dan tahap ini menjadi dasar penting dalam proses segmentasi gambar (Marpaung et al., 2022, p. 12).

5. Pengertian Deteksi dan Pengenalan objek

Deteksi dan pengenalan objek merupakan tugas penting dalam visi komputer yang digunakan untuk mengidentifikasi contoh objek visual dari kelas tertentu, seperti manusia, hewan, mobil, atau bangunan dalam gambar digital (Marpaung et al., 2022, p. 106). Tujuan dari deteksi objek adalah mengembangkan model komputasi yang memberikan informasi dasar yang diperlukan oleh aplikasi visi komputer (Marpaung et al., 2022, p. 106). Selain itu, pengenalan objek sangat

penting karena proses ini menjadi dasar bagi berbagai tugas visi komputer lainnya, seperti segmentasi instan, pengenalan teks dalam gambar, dan pelacakan objek (Marpaung et al., 2022, p. 106).

Proses deteksi dan pengenalan objek dalam gambar digital melibatkan beragam strategi yang dirancang untuk mengenali bentuk serta membedakan jenis objek berdasarkan karakteristik visualnya (Marpaung et al., 2022, p. 106). Beberapa pendekatan yang umum digunakan antara lain metode berbasis statistik, penggunaan jaringan saraf tiruan (*neural network*), algoritma *support vector machine* (SVM), serta teknik pencocokan *template*, yang seluruhnya bertujuan mengklasifikasikan objek dalam citra (Marpaung et al., 2022, p. 107). Teknik klasifikasi ini dapat diterapkan dengan cara pengawasan (*supervised*) atau tanpa pengawasan (*unsupervised*), bergantung pada ketersediaan data pelatihan dan tujuan analisis visual yang diinginkan (Marpaung et al., 2022, p. 107).

6. YOLO (*You Only Look Once*)

Menurut Batubara & Awangga (2020, p. 81) YOLO (*You Only Look Once*) adalah algoritma deteksi objek yang bekerja dengan membagi gambar menjadi kisi $S \times S$ dan mengambil kotak pembatas di setiap segmen. Menurut Batubara & Awangga (2020, p. 82) YOLO merupakan salah satu algoritma deteksi objek yang memiliki kecepatan tinggi mencapai 45 frame per detik, tetapi kurang optimal dalam mengenali benda kecil karena keterbatasan analisis citra. Menurut Batubara & Awangga (2020, pp. 82–83) adapun tahapan YOLO dalam deteksi objek adalah sebagai berikut:

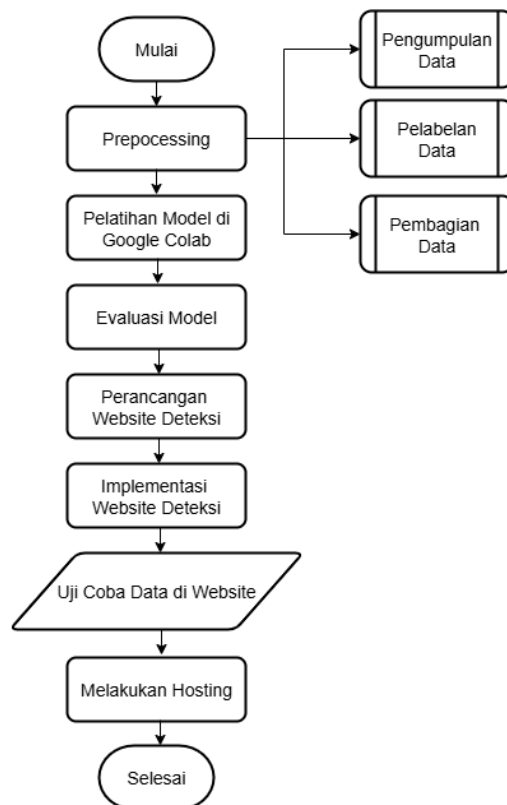
- (1) Gambar dibagi menjadi *grid* berukuran $S \times S$;
- (2) Setiap *grid box* membentuk sejumlah B “*bounding box*” dengan ukuran yang bervariasi, terletak di tengah *grid box*;
- (3) *Grid box* bertanggung jawab untuk mendeteksi objek yang terdapat dalam *bounding box*;
- (4) Setiap *grid box* hanya dapat mendeteksi satu objek saja;
- (5) *Non-Max Suppression* digunakan untuk memilih *bounding box* yang paling tepat saat terdapat banyak *bounding box* yang terdeteksi untuk objek yang sama.

YOLO merupakan metode deteksi objek berbasis *deep learning* yang bekerja secara menyeluruh dalam satu proses untuk menentukan letak dan jenis objek dalam gambar (Eko Supriyanto et al., 2025, p. 36). Algoritma ini memanfaatkan jaringan *deep convolutional neural network* yang membagi citra menjadi sejumlah *grid*, lalu setiap sel memprediksi keberadaan objek dengan

menetapkan *bounding box* dan kelas objek secara langsung (Eko Supriyanto et al., 2025, p. 37). Pengembangan model YOLO dari versi pertama hingga ketiga menunjukkan peningkatan kinerja dalam hal kecepatan, efisiensi pemrosesan, dan kemampuan untuk diterapkan pada perangkat dengan spesifikasi rendah seperti *edge device* (Eko Supriyanto et al., 2025, pp. 37–40).

Sebagai contoh penerapan, penelitian yang dilakukan oleh (Riza Luthfi Baihaqi, 2024) mengimplementasikan algoritma YOLOv5 untuk mendeteksi kualitas buah alpukat berbasis web. Sistem ini dirancang agar pengguna dapat mengunggah gambar buah alpukat, kemudian model YOLOv5 akan mendeteksi dan menentukan apakah buah tersebut termasuk kategori layak atau tidak layak. Dataset yang digunakan terdiri dari 1.000 gambar, yaitu 500 gambar alpukat layak dan 500 gambar tidak layak, yang dikumpulkan dari data primer (pemotretan langsung) dan data sekunder melalui Roboflow. Proses pelabelan dilakukan menggunakan *bounding box* sebelum model dilatih selama 50 *epoch* dengan resolusi citra 416×416 piksel. Langkah ini dilakukan untuk memastikan model dapat mengenali karakteristik visual dari dua kelas objek secara optimal.

Hasil pelatihan menunjukkan performa yang sangat baik dengan akurasi, *precision*, dan *recall* masing-masing sebesar 100%, serta *F1-score* mencapai 99,39% dengan waktu pelatihan sekitar 7 menit 38 detik. Uji coba di website dilakukan terhadap 20 citra, dan sebagian besar menunjukkan tingkat akurasi tinggi meskipun ditemukan beberapa kasus pendeteksian ganda pada satu objek. Penelitian ini membuktikan bahwa YOLOv5 efektif digunakan untuk klasifikasi kualitas buah secara *real-time*, serta memiliki keunggulan dalam hal kecepatan, efisiensi, dan kemudahan implementasi berbasis web. Berdasarkan tahapan proses yang dilakukan pada penelitian tersebut, alur kerja sistem deteksi kualitas buah alpukat menggunakan algoritma YOLOv5 digambarkan pada *Flowchart* Gambaran Umum yang ditampilkan pada Gambar 2. 2 berikut.



Gambar 2. 2 *Flowchart* Gambaran Umum Proses Deteksi Kualitas Buah Alpukat Menggunakan YOLOv5

“Sumber: (Riza Luthfi Baihaqi, 2024)”

Flowchart pada Gambar 2.2 menjelaskan alur proses pendeteksian kualitas buah alpukat menggunakan algoritma YOLOv5 berbasis web. Proses ini diawali dari tahap *preprocessing*, yang mencakup pengumpulan data, pelabelan data, dan pembagian data. Data dikumpulkan dari dua sumber, yaitu data primer hasil pemotretan langsung buah alpukat dan data sekunder yang diperoleh dari platform Roboflow. Seluruh citra yang telah dikumpulkan diunggah ke Roboflow untuk diproses menjadi dataset, kemudian dilakukan pelabelan dengan *bounding box* guna menandai area objek pada citra. Dataset yang telah dilabeli dibagi menjadi tiga bagian, yaitu 700 citra untuk pelatihan (*training*), 200 citra untuk validasi (*validation*), dan 100 citra untuk pengujian (*testing*).

Tahap berikutnya adalah pelatihan model menggunakan platform Google Colab. Repositori YOLOv5 dikloning dari GitHub dan dikonfigurasi sesuai dengan dataset yang telah dibuat. Proses pelatihan dilakukan dengan parameter ukuran citra 416×416 piksel, *batch size* sebesar 32, dan sebanyak 50 *epoch*. Dari hasil pelatihan diperoleh waktu pemrosesan selama 7 menit 38 detik, yang menunjukkan efisiensi tinggi pada proses pelatihan model. Setelah pelatihan

selesai, dilakukan evaluasi model untuk menilai performa deteksi. Evaluasi mencakup perhitungan akurasi, *precision*, *recall*, dan F1-score, yang menghasilkan nilai akurasi, *precision*, dan *recall* sebesar 100%, serta F1-score sebesar 99,39%. Analisis *confusion matrix* juga dilakukan untuk menilai kemampuan model dalam mengklasifikasikan citra secara tepat.

Setelah model selesai dilatih dan dievaluasi, tahap selanjutnya adalah perancangan dan implementasi website deteksi. Website dirancang agar pengguna dapat mengunggah gambar buah alpukat untuk dideteksi kualitasnya secara otomatis. Model YOLOv5 yang telah dilatih kemudian diintegrasikan ke dalam sistem web sehingga hasil deteksi dapat langsung ditampilkan dalam bentuk citra yang telah diberi *bounding box* beserta label kategori, yaitu layak atau tidak layak.

Tahap berikutnya adalah uji coba data pada website, yang dilakukan untuk mengukur keefektifan sistem dalam mengenali objek. Uji coba dilakukan terhadap sejumlah citra untuk menilai tingkat akurasi dan ketepatan deteksi. Hasil pengujian menunjukkan bahwa sistem mampu melakukan deteksi dengan akurasi tinggi, meskipun terdapat beberapa kasus pendeteksian ganda pada satu objek. Setelah proses uji coba selesai, tahap terakhir adalah hosting website agar sistem deteksi dapat diakses secara daring. Hosting dilakukan untuk memastikan website dapat digunakan kapan saja dan di mana saja melalui jaringan internet, serta memungkinkan proses klasifikasi dilakukan secara *real-time*. Secara keseluruhan, tahapan-tahapan pada *flowchart* ini menunjukkan bahwa algoritma YOLOv5 memiliki kemampuan yang sangat baik dalam mendeteksi dan mengklasifikasikan objek secara cepat dan akurat, serta mudah diimplementasikan pada sistem berbasis web.

7. Metode Penelitian dan Pengembangan (*Research and Development*)

Menurut Sugiyono (2019, pp. 752–754) mengatakan bahwa penelitian dan pengembangan (*Research and Development*) merupakan metode ilmiah yang bertujuan untuk menghasilkan dan memvalidasi produk tertentu melalui proses perencanaan, produksi, dan evaluasi. Menurut Sugiyono (2019, p. 754) metode ini melibatkan pendekatan sistematis yang berfokus pada pengembangan produk instruksional maupun non-instruksional agar efektif dan sesuai kebutuhan. Dengan memanfaatkan teknologi, penelitian ini diharapkan dapat menciptakan produk yang berkualitas, praktis, dan bermanfaat untuk meningkatkan produktivitas dan efektivitas.

Menurut Sugiyono (2019, p. 753), penelitian dan pengembangan tidak hanya digunakan untuk menciptakan produk baru, tetapi juga untuk menyempurnakan

produk yang sudah ada agar menjadi lebih efektif, efisien, dan sesuai kebutuhan. Menurut Borg dan Gall, penelitian dan pengembangan merupakan proses sistematis yang melibatkan perencanaan, perancangan, pembuatan, dan evaluasi guna menghasilkan produk yang valid dan bermanfaat bagi pengguna (dalam (Sugiyono, 2019, p. 753)). Metode ini dapat diterapkan dalam pengembangan produk berbasis teknologi, seperti perangkat lunak atau sistem informasi, untuk mendukung peningkatan kualitas layanan di berbagai bidang (Sugiyono, 2019, p. 754).

8. Metode Prototipe

Menurut Wijayanto et al (2024, p. 111) menyatakan bahwa metode prototyping adalah metode pengembangan yang melibatkan pembuatan model awal atau prototipe sistem untuk memahami kebutuhan pengguna sebelum sistem akhir dibuat. Menurut Wijayanto et al (2024, p. 111) prototipe ini berfungsi sebagai contoh fungsional yang memungkinkan pengguna memberikan masukan untuk perbaikan sistem. Metode ini membantu dalam memastikan bahwa produk akhir memenuhi harapan dan kebutuhan pengguna.

Menurut Mufarroha (2022, pp. 11–12), tahapan dalam metode prototipe meliputi:

- (a) Pengumpulan kebutuhan, yaitu proses identifikasi dan pendefinisian fungsi serta kebutuhan sistem;
- (b) Perancangan prototyping, yakni pembuatan rancangan awal yang menampilkan antarmuka input dan output;
- (c) Evaluasi prototyping, yaitu penilaian pengguna terhadap kesesuaian rancangan awal;
- (d) Pengkodean sistem, yaitu menerjemahkan desain ke dalam bentuk perangkat lunak;
- (e) Pengujian sistem, yakni memastikan bahwa fungsi sistem berjalan sesuai tujuan;
- (f) Evaluasi sistem, yaitu peninjauan ulang sistem secara menyeluruh oleh pengguna;
- (g) Penggunaan sistem, yaitu penerapan sistem yang telah disetujui untuk digunakan secara nyata.

Setiap tahapan dilakukan secara berulang agar pengembang dapat menyempurnakan sistem berdasarkan masukan pengguna hingga mencapai versi yang optimal (Mufarroha, 2022, p. 12). Model ini sesuai diterapkan pada proyek yang membutuhkan fleksibilitas tinggi dan melibatkan pengguna secara langsung dalam pengembangan (Mufarroha, 2022, p. 13).

9. **Black box**

Pengujian *black box* adalah metode untuk menguji perangkat lunak dengan cara mengamati hasil keluaran berdasarkan data uji yang diberikan, tanpa memeriksa struktur internal sistem (Cahyani et al., 2020, p. 162). Pengujian ini dianalogikan dengan melihat sebuah kotak hitam, dimana hanya tampilan luar yang terlihat tanpa mengetahui bagaimana mekanisme internalnya berfungsi (Cahyani et al., 2020, p. 162). Oleh karena itu, evaluasi dilakukan dengan fokus pada fungsi eksternal atau antarmuka pengguna (Cahyani et al., 2020, p. 162).

Menurut (Zatin Niqotaini, 2023) dalam Amalia Hanifa et al (2024, p. 150), pengujian *black box* merupakan metode evaluasi perangkat lunak yang menitikberatkan pada pengujian keluaran berdasarkan masukan tertentu tanpa perlu mengetahui struktur kode internal dari sistem. Menurut (Zatin Niqotaini, 2023) dalam Amalia Hanifa et al (2024, p. 151), keuntungan dari metode ini meliputi:

- (a) *independence*, karena tidak membutuhkan pemahaman terhadap kode program;
- (b) *simplicity*, karena mudah dilakukan tanpa keahlian teknis mendalam;
- (c) *user-oriented*, karena menguji sistem dari sudut pandang pengguna akhir;
- (d) *effectiveness*, karena mampu menemukan masalah integrasi antar komponen.

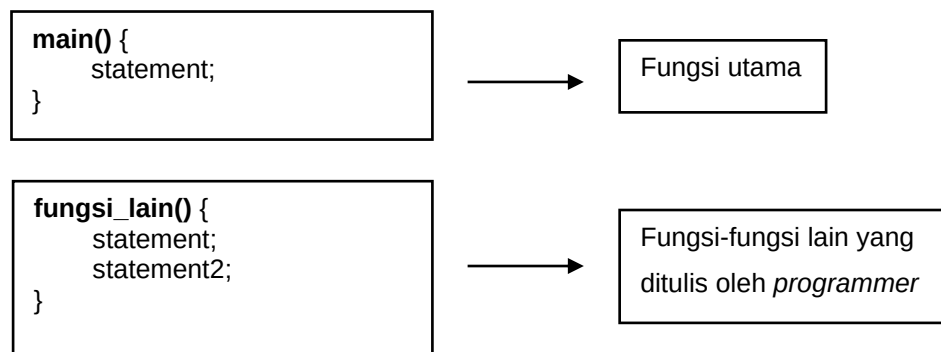
Kelemahan dari pengujian ini mencakup terbatasnya cakupan skenario yang diuji karena hanya berfokus pada fungsi eksternal, efisiensi yang rendah karena penguji tidak memahami struktur internal sistem, serta tingginya ketergantungan terhadap kelengkapan dan keakuratan dokumentasi (Amalia Hanifa et al., 2024, p. 152).

10. **C++**

Bahasa pemrograman C++ dikembangkan di AT&T Bell Laboratories pada awal 1980-an sebagai pengembangan dari bahasa C dengan penambahan konsep kelas (Esabella & Haq, 2021, p. 32). Pada periode 1983–1984, bahasa ini disempurnakan melalui penambahan fitur pembebanan operator dan fungsi, sehingga melahirkan nama C++ sebagai versi yang lebih maju dari C (Esabella & Haq, 2021, p. 32). Seiring perkembangannya, berbagai perusahaan seperti Borland, Topspeed, dan Zortech merilis compiler C++ yang dapat digunakan pada platform DOS maupun Windows (Esabella & Haq, 2021, p. 32).

Program C++ tersusun atas sejumlah fungsi, di mana minimal terdapat satu fungsi dalam setiap program (Esabella & Haq, 2021, p. 33). Setiap fungsi terdiri atas satu atau beberapa pernyataan (*statement*) yang akan diproses oleh

compiler C++ sebagai instruksi untuk menjalankan logika tertentu (Esabella & Haq, 2021, p. 33). Fungsi `main()` merupakan fungsi istimewa yang harus ada dalam program C++ karena berperan sebagai titik awal sekaligus titik akhir eksekusi, biasanya ditempatkan pada bagian atas kode program agar memudahkan dalam menemukan program utama (Esabella & Haq, 2021, p. 33).



“Sumber : (Esabella & Haq, 2021, p. 33)”

11. **Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) merupakan bahasa penulisan yang digunakan untuk mengatur tampilan dokumen markup, terutama HTML, dengan memisahkan konten dari desain sehingga perubahan gaya dapat dilakukan secara konsisten pada banyak halaman web hanya melalui satu file CSS (Mufarroha, 2022, pp. 41–42). CSS memungkinkan pengaturan elemen visual seperti warna, jenis dan ukuran huruf, tata letak, serta gaya lainnya dengan menggunakan *selector* berupa elemen, *id*, maupun *class* (Mufarroha, 2022, pp. 41–42). Seiring perkembangan teknologi, CSS tidak hanya terbatas pada HTML, tetapi juga dapat digunakan pada dokumen markup lain seperti XML, SVG, maupun untuk mendukung desain antarmuka aplikasi modern (Mufarroha, 2022, pp. 41–42).

CSS menggunakan sintaks sederhana berbasis bahasa Inggris dengan struktur utama berupa *selector* dan *declaration block* yang memudahkan penerapannya dalam mengatur tampilan elemen HTML (Mufarroha, 2022, pp. 50–51). Berdasarkan cara penggunaannya, CSS terbagi menjadi tiga jenis, yaitu *inline style sheet* yang diterapkan langsung pada elemen tertentu, *embedded style sheet* yang dituliskan di dalam tag `<head>` pada halaman HTML, serta *external style sheet* yang ditempatkan pada file terpisah sehingga dapat digunakan kembali pada banyak halaman sekaligus (Mufarroha, 2022, pp. 50–51). Dengan adanya variasi jenis CSS tersebut, pengembang web memiliki

fleksibilitas dalam memilih metode yang sesuai untuk menghasilkan tampilan yang konsisten dan efisien pada sebuah website.

12. Hypertext Markup Language (HTML)

HTML merupakan singkatan dari *Hypertext Markup Language* yang berfungsi untuk membuat dan menyusun elemen-elemen halaman web, seperti paragraf, heading, tautan (*link*), dan *blockquote* (Mufarroha, 2022, pp. 39–40). HTML termasuk bahasa markup, bukan bahasa pemrograman, yang menyusun halaman web hingga dapat ditampilkan melalui browser (Mufarroha, 2022, pp. 39–40). Dalam pembuatan website, terdapat dua komponen penting, yaitu WWW (*World Wide Web*) dan HTML, di mana WWW merupakan protokol standar internet, sedangkan HTML merupakan skrip standar yang dijalankan melalui WWW (Mufarroha, 2022, pp. 39–40). Proyek HTML umumnya dibuat menggunakan media *web editor* sederhana, seperti Notepad, Notepad++, Sublime Text, Atom atau yang lainnya (Mufarroha, 2022, pp. 39–40).

Setiap dokumen HTML yang ditampilkan melalui browser memiliki struktur dasar yang terdiri dari tag, elemen, dan atribut untuk mengatur isi informasi dan tata letak halaman (Mufarroha, 2022, pp. 39–40). Walaupun browser tetap dapat membaca dokumen tanpa struktur yang benar, hasil tampilan tidak akan sesuai standar, sehingga diperlukan penulisan struktur dasar HTML yang baik (Mufarroha, 2022, pp. 39–40). Struktur HTML umumnya terdiri atas tag `<html>`, `<head>`, `<title>`, dan `<body>`, di mana `<html>` berfungsi sebagai pembuka dokumen, `<head>` berisi informasi tambahan yang tidak ditampilkan secara langsung, `<title>` menampilkan judul halaman pada tab browser, dan `<body>` menampilkan seluruh konten utama halaman web (Mufarroha, 2022, pp. 39–40).

13. JavaScript

JavaScript merupakan bahasa pemrograman yang berperan penting dalam memberikan interaksi dan dinamika pada halaman web (Mukhlis et al., 2023, p. 71). Bahasa ini berjalan di sisi klien (*client-side*) melalui browser, sehingga memungkinkan perubahan konten halaman secara dinamis, merespons masukan pengguna, memvalidasi formulir, menambahkan animasi, serta mengakses dan memanipulasi data dari server (Mukhlis et al., 2023, pp. 71–72). Dengan fitur dan API yang dimiliki, JavaScript dapat memanipulasi elemen HTML dan CSS untuk menciptakan pengalaman web yang lebih interaktif dan responsif (Mukhlis et al., 2023, pp. 71–72).

Dalam pengembangan web, JavaScript biasanya digunakan bersama dengan HTML dan CSS, karena ketiganya saling melengkapi dalam memberikan struktur, tampilan, dan interaksi pada halaman web (Mukhlis et al., 2023, p. 72).

Pemahaman yang baik terhadap HTML, CSS, dan JavaScript menjadi hal penting agar aplikasi web dapat dibangun secara fungsional dan sesuai kebutuhan pengguna (Mukhlis et al., 2023, p. 72). Menurut Mukhlis et al (2023, p. 73) dalam penulisan kode JavaScript pada berkas HTML, terdapat beberapa cara umum yang dapat digunakan, yaitu;

- (1) *Inline scripting*, yaitu penulisan kode JavaScript langsung di dalam tag `<script>` pada berkas HTML;
- (2) *Internal scripting*, yaitu penulisan kode JavaScript di dalam tag `<script>` yang ditempatkan pada bagian `<head>` dalam berkas HTML;
- (3) *External scripting*, yaitu penulisan kode JavaScript di berkas terpisah dengan ekstensi `.js`, kemudian dihubungkan ke dalam HTML melalui tag `<script>`.

14. PHP

PHP merupakan salah satu bahasa pemrograman web *open source* yang populer dan pertama kali diperkenalkan pada Juni 1995 oleh Rasmus Lerdorf (Junus, 2021, p. 210). Pada awal pengembangannya, PHP masih bernama PHP/FI (*Personal Home Page/Form Interpreter*) yang ditulis menggunakan bahasa C pada tahun 1994. Bahasa ini awalnya digunakan sebagai alat sederhana untuk melacak kunjungan pengguna pada website pribadi Lerdorf (Junus, 2021, p. 210). Seiring berjalannya waktu, PHP dikembangkan menjadi lebih kompleks dengan kemampuan berinteraksi dengan basis data serta mendukung pembuatan halaman web yang dinamis (Junus, 2021, p. 210).

Selain itu, PHP juga mampu mengelola file pada server, mengumpulkan data melalui formulir, mengirim serta menerima *cookies*, mengontrol akses pengguna, hingga mendukung enkripsi data (Junus, 2021, p. 211). Tidak hanya menampilkan output dalam bentuk HTML, PHP juga dapat digunakan untuk menghasilkan file gambar, PDF, hingga multimedia seperti Flash. Menurut w3schools.com (n.d.) dalam Junus (2021, p. 211), penulisan sintaks PHP dapat dipadukan dengan HTML, CSS, JavaScript, maupun SQL. File PHP sendiri ditulis dengan ekstensi `.php` dan setiap baris pernyataan harus diakhiri dengan tanda titik koma (;) dalam blok `<?php ?>` (Junus, 2021, p. 211).

15. Python

Python merupakan bahasa pemrograman berorientasi objek yang diciptakan oleh Guido Van Rossum pada tahun 1991 dengan versi awal 0.9.0, yang sejak awal sudah mendukung konsep *exception*, *function*, serta berbagai tipe data dasar seperti *list*, *dict*, dan *string* (Abdiansah, n.d., p. 1). Perkembangannya berlanjut dengan rilis Python 1.0 pada tahun 1994 yang menambahkan fitur

pemrograman fungsional, Python 2.0 pada tahun 2000 dengan *list comprehensions* dan dukungan *Unicode*, hingga Python 3.0 pada tahun 2008 yang membawa perubahan besar dan tidak kompatibel dengan versi sebelumnya (Abdiansah, n.d., p. 1). Hingga kini, Python menjadi salah satu bahasa pemrograman paling populer karena kesederhanaan sintaksis, dukungan *multiplatform*, serta ketersediaan pustaka luas untuk berbagai kebutuhan, termasuk *machine learning* dan *deep learning* (Abdiansah, n.d., p. 1).

Keunggulan Python terletak pada fleksibilitas penggunaannya yang dapat dipakai untuk membangun aplikasi web, perangkat lunak desktop, basis data, analisis data, hingga komputasi kompleks (Abdiansah, n.d., pp. 2–3). Selain itu, Python mendukung antarmuka grafis (GUI) untuk aplikasi maupun game serta memungkinkan pengembangan perangkat lunak dengan pendekatan prosedural, berorientasi objek, maupun fungsional (Abdiansah, n.d., pp. 2–3). Dibandingkan bahasa pemrograman lain, Python memiliki sintaks sederhana mirip bahasa Inggris dengan indentasi sebagai penanda ruang lingkup kode dan baris baru untuk mengakhiri perintah, sehingga lebih mudah dipahami dan ditulis secara ringkas (Abdiansah, n.d., pp. 2–3).

16. MySQL

SQL merupakan bahasa standar dalam Relational Database Management System (RDBMS) yang bersifat non-prosedural dan berorientasi himpunan, serta dapat digunakan secara interaktif maupun terintegrasi dalam aplikasi (Pane, Zamzam, et al., 2020, p. 72). Standarisasi SQL dimulai tahun 1986 oleh ANSI, kemudian diperbaiki pada tahun 1989 dan 1992 (Pane, Zamzam, et al., 2020, p. 72). Beberapa DBMS yang menggunakan SQL antara lain Oracle, Sybase, Microsoft SQL Server, Access, dan Ingres, dengan dukungan perintah dasar seperti *SELECT*, *INSERT*, *UPDATE*, *DELETE*, *CREATE*, dan *DROP* (Pane, Zamzam, et al., 2020, p. 72).

MySQL adalah perangkat lunak open source dengan kinerja cepat, andal, serta mudah digunakan, yang bekerja menggunakan arsitektur client-server maupun embedded system (Pane, Zamzam, et al., 2020, p. 72). MySQL mampu mengelola basis data dari skala kecil hingga sangat besar, dan saat ini telah berkembang mencapai versi 5 (Pane, Zamzam, et al., 2020, p. 72). Selain itu, MySQL juga mendukung eksekusi perintah SQL untuk melakukan pengelolaan data dalam berbagai jenis basis data (Pane, Zamzam, et al., 2020, p. 72).

17. XAMPP

XAMPP merupakan perangkat lunak *open source* yang digunakan untuk pengembangan aplikasi berbasis web karena sudah menyediakan paket lengkap

seperti Apache, MySQL/MariaDB, PHP, phpMyAdmin, FileZilla, Tomcat, dan XAMPP Control Panel (Mahardika et al., 2025, p. 4). Perangkat ini berfungsi sebagai pengganti web hosting dengan menyimpan file website pada server lokal (*localhost*), sehingga dapat diakses melalui browser (Mahardika et al., 2025, p. 4). XAMPP pertama kali dikembangkan oleh tim *Apache Friends* dan dirilis secara gratis di bawah lisensi GNU (*General Public License*) (Mahardika et al., 2025, p. 4).

Nama XAMPP sendiri merupakan akronim dari *X* (*cross platform*) yang berarti dapat berjalan di berbagai sistem operasi seperti Windows, Mac, dan Linux; *A* (Apache) sebagai web server; *M* (MySQL/MariaDB) sebagai database server; *P* (PHP) sebagai bahasa pemrograman untuk membangun website dinamis; dan *P* (Perl) sebagai bahasa pemrograman fleksibel yang mendukung banyak kebutuhan (Mahardika et al., 2025, p. 5). Hingga saat ini, XAMPP terus dikembangkan dan tersedia secara gratis untuk berbagai platform, sehingga menjadi salah satu tools penting dalam pengembangan aplikasi web. Dengan dukungan PHP, MySQLi, dan MariaDB, XAMPP semakin memudahkan pengembang dalam membangun dan menguji aplikasi web secara lokal.

18. BPMN

Business Process Modeling Notation (BPMN) merupakan standar yang digunakan untuk memvisualisasikan proses bisnis melalui simbol-simbol grafis yang menggambarkan tahapan kegiatan dalam suatu organisasi (Siregar, 2025, pp. 105–106). BPMN menampilkan alur proses bisnis dalam bentuk diagram yang disusun berdasarkan konsep diagram alur, sehingga dapat memperlihatkan aktivitas, urutan kerja, serta kontrol alur yang terlibat di dalamnya (Siregar, 2025, pp. 105–106). Tujuan BPMN adalah menyediakan notasi yang mudah dipahami oleh pengguna bisnis maupun teknis serta mampu menggambarkan makna semantik dari proses bisnis yang kompleks (Pane, Lase, et al., 2020, p. 196). Berikut ini simbol notasi BPMN yang ditampilkan pada gambar 2.3.

No	Simbol	Keterangan
1		<i>Start Event</i>
2		<i>Intermediate event</i>
3		<i>End Event</i>
4		<i>Task/Activity</i>
5		<i>Gateway</i>
6		<i>Sequence Flow</i>
7		<i>Message Flow</i>
8		<i>Group</i>

Gambar 2. 3 Simbol-simbol BPMN

“Sumber: (Pane, Lase, et al., 2020, p. 197)”

19. Usecase Diagram

Use Case Diagram merupakan salah satu jenis diagram dalam UML yang digunakan untuk memodelkan kebutuhan fungsional suatu sistem serta menggambarkan hubungan antara pengguna dan sistem yang dirancang (Habibi & Aprilian, 2020, pp. 89–91). Diagram ini menunjukkan secara sederhana siapa yang berinteraksi dengan sistem dan fungsi apa saja yang dapat dijalankan, dengan penekanan pada apa yang dilakukan sistem, bukan bagaimana cara kerjanya (Habibi & Aprilian, 2020, pp. 89–91). *Use case diagram* berfungsi untuk menjelaskan interaksi antara aktor dan sistem guna memahami kebutuhan serta perilaku sistem secara keseluruhan (Habibi & Aprilian, 2020, pp. 89–91). Adapun simbol-simbol dari usecase diagram yang ditampilkan pada gambar 2.4.

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2. 4 Simbol-simbol *Use case Diagram*

“Sumber: (Iswara et al., n.d., pp. 26–27)”

20. Sequence Diagram

Sequence Diagram merupakan salah satu jenis diagram dalam UML yang digunakan untuk menggambarkan interaksi antarobjek serta komunikasi yang terjadi di antara objek-objek tersebut dalam suatu sistem (Habibi & Aprilian, 2020, p. 103). Diagram ini menampilkan urutan pesan yang dipertukarkan oleh objek-objek untuk menjalankan suatu tugas tertentu, di mana objek disusun secara horizontal dan waktu direpresentasikan secara vertikal dari atas ke bawah (Habibi & Aprilian, 2020, p. 103). Setiap objek atau aktor memiliki garis vertikal yang disebut *lifeline*, dan ketika objek melakukan suatu operasi, muncul kotak yang disebut *activation box* sebagai penanda aktivitas (Habibi & Aprilian, 2020,

p. 103). Berikut ini simbol-simbol *sequence diagram* yang ditunjukkan pada gambar 2.5.

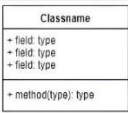
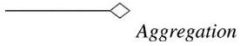
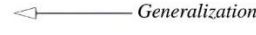
Simbol	Fungsi
	<i>Entity class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas - entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. <i>Control object</i> mengkoordinir pesan antara <i>boundary</i> dengan entitas.
	<i>Message</i> , simbol mengirim pesan antar class.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Gambar 2. 5 Simbol-simbol *Sequence Diagram*

“Sumber: (Experts & Community, 2019, p. 48)”

21. *Class Diagram*

Class Diagram merupakan salah satu jenis diagram dalam UML yang berfungsi untuk memvisualisasikan kelas beserta hubungan antar kelas dalam suatu sistem (Habibi & Aprilian, 2020, pp. 99–100). Diagram ini digambarkan dalam bentuk kotak yang terbagi menjadi tiga bagian, yaitu bagian atas berisi nama kelas, bagian tengah berisi atribut atau variabel yang dimiliki, dan bagian bawah berisi metode yang digunakan oleh kelas tersebut (Habibi & Aprilian, 2020, pp. 99–100). Berdasarkan konsep UML, *class diagram* bersifat statis karena menampilkan struktur sistem secara keseluruhan dengan memodelkan kelas, atribut, operasi, serta hubungan antarobjek seperti pewarisan, asosiasi, dan komposisi (Habibi & Aprilian, 2020, pp. 99–100). Berikut ini simbol-simbol *class diagram* yang ditunjukkan pada gambar 2.6.

Simbol	Keterangan
 <i>Assosiation</i>	Merupakan <i>class-class</i> yang saling terhubung satu sama lain secara konseptual.
 <i>Class</i>	Merupakan kumpulan obyek yang memiliki <i>attribute</i> dan <i>operation</i> yang sama. <i>Attribute</i> , merupakan <i>property</i> dari sebuah <i>class</i> . <i>Operation</i> merupakan sesuatu yang bisa dilakukan oleh sebuah <i>class</i> .
 <i>Aggregation</i>	Merupakan suatu garis yang menangani obyek- obyek dimana salah satunya adalah bagian dari yang lain.
 <i>Compostion</i>	Merupakan sebuah tipe agregasi yang kuat dimana bagian dari obyek tergantung pada keseluruhan obyek.
 <i>Dependency</i>	Merupakan sebuah garis yang berfungsi menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain.
 <i>Generalization</i>	Merupakan suatu garis yang berfungsi untuk mewariskan struktur data dan obyek induk kepada obyek anak yang dituju.

Gambar 2. 6 Simbol-simbol *Class Diagram*

“Sumber: (Zen & Saragih, 2024, p. 34)”

22. *Deployment Diagram*

Deployment Diagram merupakan salah satu jenis diagram dalam UML yang digunakan untuk memvisualisasikan aspek fisik dari sistem berorientasi objek dengan menampilkan hubungan antara komponen perangkat lunak dan perangkat keras tempat komponen tersebut dijalankan (Aprilian & Saputra, 2020, pp. 84–85). Diagram ini menggambarkan bagaimana artefak perangkat lunak didistribusikan ke berbagai node atau perangkat keras sehingga dapat memberikan gambaran arsitektur sistem secara statis pada saat run-time (Aprilian & Saputra, 2020, pp. 84–85). Selain itu, *deployment diagram* berfungsi untuk menunjukkan struktur sistem, hubungan antarperangkat keras, serta mendokumentasikan proses penyebaran komponen perangkat lunak (Aprilian & Saputra, 2020, pp. 84–85). Berikut ini simbol-simbol *deployment diagram*.

Gambar	Keterangan
	Pada deployment diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka
	Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
	Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara element-elemen hardware.

Gambar 2. 7 Simbol-simbol *Deployment Diagram*

Sumber: “(Sari & Indra, 2021, pp. 14–15)”

B. Tinjauan Studi

Penelitian ini merujuk pada penelitian-penelitian sebelumnya yang terkait dengan pemantauan menggunakan *ESP32-CAM*. Beberapa studi tersebut menjadi acuan penting dalam pengembangan sistem yang dibangun dalam penelitian ini. Adapun penelitian-penelitian yang relevan antara lain:

- (1) **Penelitian yang dilakukan oleh** (Nasution et al., 2022) **dengan judul “Perancangan Sistem Monitoring Stok Minuman di Vending Machine Berbasis IoT”**. Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem monitoring stok pada *vending machine* selama ini masih dilakukan secara konvensional, yang mengharuskan pemilik *vending machine* untuk memeriksa stok secara konvensional. Hal ini menyebabkan pemborosan waktu dan biaya, serta keterlambatan dalam pengisian ulang stok. Penelitian ini bertujuan merancang sistem berbasis Internet of Things (IoT) untuk memantau stok secara otomatis dengan menggunakan modul *ESP32-CAM* dan platform Blynk. Sistem ini juga dilengkapi dengan sensor fotodioda dan motor DC untuk mendeteksi dan mengatur stok minuman. Hasil pengujian menunjukkan bahwa sistem efektif dalam memberikan informasi secara *real-time*, meskipun terdapat kekurangan, seperti keterbatasan dalam mendeteksi hanya botol minuman dan data visual yang dihasilkan oleh *ESP32-CAM* hanya berupa gambar tanpa analisis lebih lanjut.
- (2) **Penelitian yang dilakukan oleh** (Silalahi et al., 2024) **dengan judul “Penerapan Metode Haar Cascade terhadap Objek Botol menggunakan ESP32-Cam”**. Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa belum banyak sistem deteksi objek yang efektif dalam mengidentifikasi botol dengan akurasi tinggi pada berbagai kondisi. Deteksi objek botol sering kali mengalami kendala, terutama ketika dihadapkan pada berbagai jenis bahan dan ukuran. Penelitian ini menunjukkan bahwa sistem yang menggunakan *ESP32-CAM* dan metode *Haar Cascade* mampu mendeteksi objek botol dengan akurasi rata-rata 75% untuk botol kaca dan 78% untuk botol plastik. Namun, akurasi ini menurun menjadi 44% saat objek dihalangi oleh kaca film, yang menunjukkan bahwa sistem masih rentan terhadap kondisi lingkungan yang tidak mendukung.
- (3) **Penelitian yang dilakukan oleh** (Ziliwu et al., 2024) **dengan judul “Penerapan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran”**. Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa pengklasifikasian gambar buah dan sayuran masih menghadapi tantangan dalam hal biaya dan efektivitas. Sistem pengklasifikasian yang ada sering kali

memerlukan biaya yang tinggi untuk pengadaan perangkat keras dan pengembangan model. Selama ini, banyak solusi yang tersedia tidak dapat memberikan akurasi yang memadai dengan biaya yang terjangkau. Diperlukan teknologi yang lebih efektif dan hemat biaya untuk membantu para petani dan pelaku industri pertanian dalam mengelola stok dan identifikasi produk. Dalam penelitian ini, ESP32-CAM digunakan sebagai perangkat keras yang terintegrasi dengan TinyML untuk melakukan klasifikasi gambar. Meskipun hasil akurasi mencapai 68,3% untuk F1 score, terdapat variasi nilai akurasi yang cukup signifikan antar jenis buah dan sayuran, yang dapat menyebabkan ketidakpastian dalam klasifikasi.

(4) **Penelitian yang dilakukan oleh (Rahman, 2024) dengan judul “Teknologi Internet of Things (IoT) Pada Penyimpanan Paket Online Terintegrasi Esp32-Cam Dan Bot Telegram Untuk Notifikasi Penerimaan Paket”.**

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa masih sedikit sistem penyimpanan paket yang mampu memberikan notifikasi secara *real-time* kepada pengguna saat menerima paket. Selama ini, penerimaan paket sering kali dilakukan secara konvensional, di mana kurir harus bertemu langsung dengan penerima. Hal ini berisiko, terutama jika paket yang diterima adalah barang berharga. Diperlukan sistem penyimpanan paket yang lebih modern, yang memungkinkan pengguna untuk memantau status paket dari jarak jauh. Dalam penelitian ini, digunakan modul ESP32-CAM yang dilengkapi dengan fitur kamera untuk memantau paket yang disimpan. Sistem ini juga terintegrasi dengan bot Telegram, yang berfungsi sebagai sarana notifikasi dan kontrol. Pengujian menunjukkan bahwa sistem ini berhasil memberikan notifikasi kepada pengguna dengan tingkat keberhasilan yang tinggi, meskipun terdapat beberapa kelemahan, seperti keterbatasan lokasi pengujian dan keamanan data.

(5) **Penelitian yang dilakukan oleh (Zazin et al., 2022) dengan judul “Implementasi IoT pada Sistem Surveillance Camera Via Telegram”.**

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem pengawasan menggunakan ESP32-Cam dan sensor PIR untuk mendeteksi gerakan dan mengirimkan notifikasi kepada pemilik rumah melalui aplikasi Telegram. Namun, terdapat kekurangan dalam sistem ini. Sistem ini lebih fokus pada aspek keamanan dari pencurian tanpa banyak membahas perlindungan data pengguna, yang merupakan hal penting dalam era digital saat ini. Pengujian sistem ini juga tidak dijelaskan secara rinci mengenai lokasi dan kondisi. Sistem ini tergantung pada sensor gerak untuk mendeteksi aktivitas,

yang mungkin tidak dapat menangkap semua pergerakan, terutama dalam situasi tertentu.

- (6) **Penelitian yang dilakukan oleh** (Selokaton et al., 2025) **dengan judul** **“Penerapan Internet of Things (IoT) Untuk Merancang Sistem Keamanan Pada Brankas Menggunakan ESP32-CAM dan Keypad”**. Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa belum banyak sistem pemantauan keamanan brankas yang mampu memberikan notifikasi secara *real-time* kepada pemilik. Sebagian besar sistem yang ada masih bergantung pada keamanan langsung, seperti penggunaan kunci fisik, yang tidak cukup efektif dalam menghadapi risiko pencurian. Sistem keamanan yang dirancang dalam penelitian ini menggunakan teknologi IoT, yang memungkinkan pemilik untuk menerima notifikasi melalui aplikasi Telegram. Meskipun demikian, tantangan yang dihadapi adalah kebutuhan akan koneksi Wi-Fi yang stabil, serta keterbatasan dalam pengenalan wajah yang dapat dipengaruhi oleh kondisi pencahayaan.
- (7) **Penelitian yang dilakukan oleh** (Nurhaliza et al., 2024) **dengan judul** **“Integration Of ESP32-CAM With And IoT Based English Indonesian Translation Application Using OCR Technology”**. Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa meskipun aplikasi ini berhasil mengintegrasikan teknologi OCR dengan ESP32-CAM untuk penerjemahan, terdapat kekurangan yaitu aplikasi ini masih memiliki keterbatasan dalam hal akurasi terjemahan, terutama ketika berhadapan dengan kalimat yang kompleks atau konteks yang ambigu. Meskipun OCR menunjukkan akurasi 100% dalam mendeteksi teks, kesalahan dalam terjemahan bahasa dapat terjadi jika konteks tidak dipahami dengan baik. Selain itu, sistem hanya dapat mendeteksi teks dan tidak mampu mengidentifikasi gambar atau simbol lain, yang dapat membatasi fungsionalitas pengguna.
- (8) **Penelitian yang dilakukan oleh** (Mahmoud et al., 2022) **dengan judul** **“Design of an IOT System based on Face Recognition Technology using ESP32-CAM”**. Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem pengenalan wajah yang diusulkan memiliki kekurangan. Meskipun sistem ini mampu melakukan pemantauan secara *real-time*, namun masih terbatas pada pengenalan wajah yang telah terdaftar dalam basis data. Hal ini berarti sistem tidak dapat mengenali wajah baru yang tidak ada dalam database, sehingga mengurangi fleksibilitasnya dalam situasi yang dinamis.
- (9) **Penelitian yang dilakukan oleh** (Afrena et al., 2024) **dengan judul** **“Poultry Farming Abnormality Detection using ESP32-CAM and OpenCV”**.

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan bahwa sistem yang diusulkan masih memiliki beberapa kekurangan. Pertama, sistem ini belum sepenuhnya dapat mendeteksi semua jenis abnormalitas perilaku ayam dengan akurasi yang diharapkan. Meskipun menggunakan teknologi seperti SSD MobileNet dan OpenCV, masih terdapat tantangan dalam mendeteksi perilaku tertentu.

- (10) **Penelitian ini dilakukan oleh** (Paryono et al., 2023) **dengan judul “*Detecting Vehicle Numbers Using Google Lens-Based ESP32CAM to Read Number Characters*”**. Berdasarkan penelitian yang telah dilakukan, dapat disimpulkan bahwa belum banyak sistem deteksi nomor kendaraan yang mampu memberikan hasil pembacaan secara akurat dan cepat. Meskipun penggunaan kamera sebagai sensor deteksi plat kendaraan telah diterapkan, masih terdapat keterbatasan dalam hal akurasi pembacaan, terutama pada plat yang memiliki warna cerah atau pencahayaan yang tinggi. Pada tabel 2.1 berikut ini menyajikan data penelitian sebelumnya yang sesuai dengan deskripsi yang telah diuraikan.

Tabel 2. 1 Penelitian Rujukan

No	Nama Peneliti	Judul Penelitian	Sumber	Kontribusi
1	(Nasution et al., 2022)	Perancangan Sistem Monitoring Stok Minuman Di <i>Vending Machine</i> Berbasis IoT	JURNAL SISTEM KOMPUTER TGD Volume 1, Nomor 2, 2022, Hal 57-64 https://ojs.trigunadharma.ac.id/index.php/jskom/article/view/5138	Kontribusi dalam penelitian ini adalah telah dihasilkan sistem pemantauan stok otomatis dengan mengintegrasikan ESP32-CAM dan platform Blynk, yang dapat menginformasikan ketersediaan barang melalui pemantauan visual.
2	(Silalahi et al., 2024)	Penerapan Metode <i>Haar Cascade</i> terhadap Objek Botol menggunakan <i>ESP32-Cam</i>	Jurnal Pendidikan dan Teknologi Indonesia (JPTI) Vol. 4, No. 6, Juni 2024, Hal. 269-276	Kontribusi dalam penelitian ini adalah telah dihasilkan aplikasi deteksi objek menggunakan

No	Nama Peneliti	Judul Penelitian	Sumber	Kontribusi
			https://www.jpti.journals.id/index.php/jpti/article/view/457	ESP32-CAM, yang dapat digunakan untuk meningkatkan akurasi dalam sistem pemantauan barang.
3	(Ziliwu et al., 2024)	Penerapan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran	Jurnal Ilmiah Teknik Informatika dan Sistem Informasi, Vol. 13, No. 1(2024) https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/article/view/1869	Kontribusi dalam penelitian ini adalah penerapan algoritma pembelajaran mesin untuk meningkatkan akurasi klasifikasi objek dan penerapan ESP32-CAM.
4	(Rahman, 2024)	Teknologi Internet of Things (IoT) Pada Penyimpanan Paket Online Terintegrasi Esp32-Cam Dan Bot Telegram Untuk Notifikasi Penerimaan Paket	Journal of Computing and Informatics Vol. 01, No. 02 (2024) https://ejournal.duniakampus.org/index.php/ijoki/article/view/94	Kontribusi dalam penelitian ini adalah telah dihasilkan pengembangan sistem pemantauan paket menggunakan ESP32-CAM dan integrasi fitur notifikasi.
5	(Zazin et al., 2022)	Implementasi IoT pada Sistem <i>Surveillance Camera</i> Via Telegram	Jurnal Teknolgi dan Manajemen, Vol. 3, No. 2(2022) https://ejurnal.itats.ac.id/jtm/article/view/3391	Kontribusi dalam penelitian ini adalah telah dihasilkan pengembangan sistem pengawasan dengan menggunakan ESP32-CAM.
6	(Selokaton et al., 2025)	Penerapan Internet of Things (IoT) Untuk Merancang Sistem	JATI (Jurnal Mahasiswa Teknik	Kontribusi dalam penelitian ini adalah telah dihasilkan

No	Nama Peneliti	Judul Penelitian	Sumber	Kontribusi
		Keamanan Pada Brankas Menggunakan ESP32-CAM dan Keypad	Informatika) Vol. 8 No. 5 (2024) https://www.ejournal.itn.ac.id/index.php/jati/article/view/10934	aplikasi teknologi IoT yang dapat memberikan notifikasi, yang relevan dalam pengembangan sistem pemantauan barang berbasis ESP32-CAM.
7	(Nurhaliza et al., 2024)	<i>Integration Of ESP32-CAM With And IoT Based English Indonesian Translation Application Using OCR Technology</i>	Jurnal Teknik Informatika (JUTIF) Vol. 5, No. 4 (2024) https://jutif.if.unsoed.ac.id/index.php/jurnal/article/view/1489	Kontribusi dalam penelitian ini adalah integrasi ESP32-CAM dengan aplikasi berbasis IoT untuk meningkatkan efektivitas dalam pemantauan barang secara otomatis.
8	(Saidi & Bouzazi, 2022)	<i>Design of an IOT System based on Face Recognition Technology using ESP32-CAM</i>	(IJCSNS) International Journal of Computer Science and Network Security, VOL. 20 No. 11 (2022) https://www.researchgate.net/publication/361877428_Design_of_an_IOT_System_based_on_Face_Recognition_Technology_using_ESP32-CAM	Kontribusi dalam penelitian ini adalah telah dihasilkan penerapan teknologi pengenalan objek berbasis ESP32-CAM.

No	Nama Peneliti	Judul Penelitian	Sumber	Kontribusi
9	(Afrena et al., 2024)	<i>Poultry Farming Abnormality Detection using ESP32-CAM and OpenCV</i>	Progress In Engineering Application And Technology Vol. 5 No. 2 (2024) https://publisher.uthm.edu.my/periodicals/index.php/peat/article/view/17174	Kontribusi dalam penelitian ini adalah penerapan ESP32-CAM untuk memantau objek.
10	(Paryono et al., 2023)	<i>Detecting Vehicle Numbers Using Google Lens-Based ESP32CAM to Read Number Characters</i>	Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer Vol. 22, No. 3 (2023) https://journal.universitiasbumigora.ac.id/index.php/matrik/article/view/2818	Kontribusi dalam penelitian ini adalah telah dihasilkan ilustrasi aplikasi deteksi berbasis gambar dengan ESP32-CAM yang relevan untuk sistem pemantauan barang.

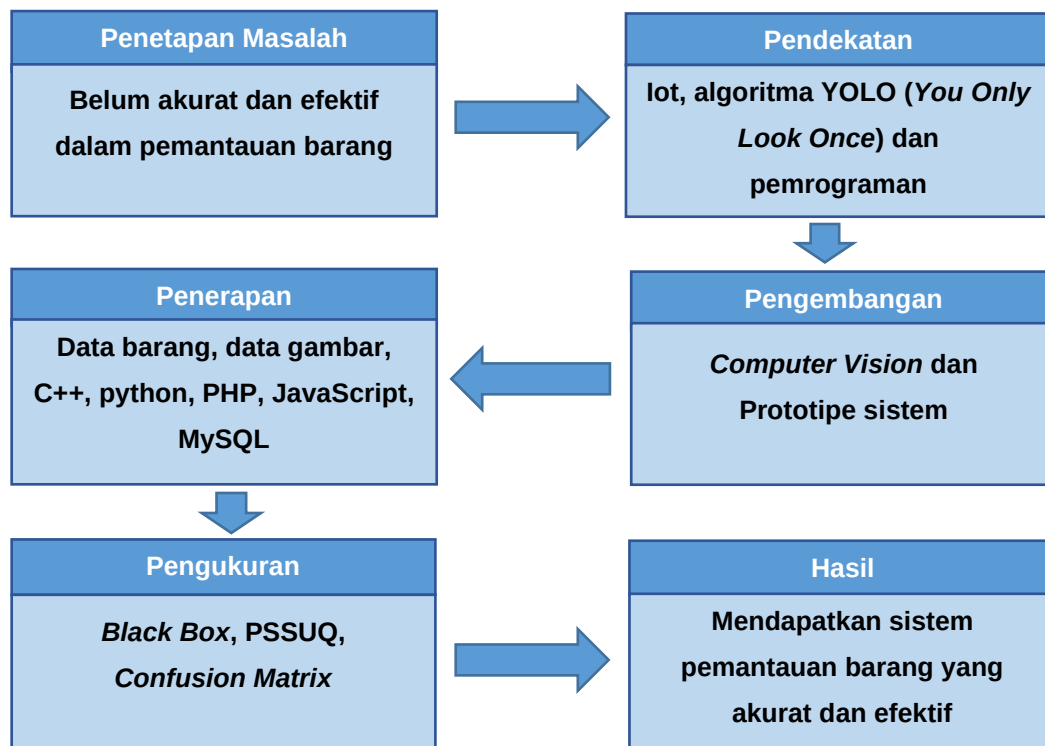
Berdasarkan tinjauan terhadap sepuluh penelitian sebelumnya, terdapat beberapa persamaan utama yang menjadi **kontribusi** dalam pengembangan sistem pemantauan barang di rak berbasis IoT menggunakan ESP32-CAM. Sebagian besar penelitian telah menerapkan ESP32-CAM sebagai perangkat utama dalam pemantauan objek, mendeteksi keberadaan barang, melakukan pengawasan keamanan, dan mengklasifikasikan objek secara otomatis. Selain itu, seluruh penelitian yang dirujuk telah mengintegrasikan teknologi Internet of Things (IoT) dengan berbagai metode, seperti penggunaan platform Blynk, Telegram, atau TinyML. Pendekatan *computer vision* serta algoritma deteksi, seperti *Haar Cascade* dan TinyML, juga banyak diterapkan dalam penelitian sebelumnya untuk meningkatkan akurasi identifikasi objek.

Meskipun memiliki beberapa persamaan dengan penelitian sebelumnya, penelitian ini memiliki **orisinalitas** yang membedakannya. Fokus utama dalam penelitian ini adalah mendeteksi keberadaan barang di rak untuk memastikan keteraturan serta mempercepat

proses pengisian ulang di lingkungan retail, yang belum banyak dibahas dalam penelitian lain. Penelitian ini menerapkan algoritma YOLO (You Only Look Once), yang lebih akurat dibandingkan dengan metode *Haar Cascade* atau metode deteksi konvensional yang digunakan dalam penelitian sebelumnya. Penelitian ini juga mengoptimalkan pengolahan data citra dari ESP32-CAM agar dapat diproses dengan lebih cepat dan efisien dalam penggunaan daya, sehingga memungkinkan implementasi yang lebih efektif di lingkungan ritel. Dengan demikian, penelitian ini tidak hanya merujuk pada penelitian sebelumnya, tetapi juga memberikan kontribusi baru dalam pengembangan sistem pemantauan barang di rak berbasis IoT yang lebih akurat, cepat, dan mudah diimplementasikan.

C. Kerangka Berfikir

Berikut ini adalah kerangka berfikir untuk memecahkan masalah dalam penelitian ini :



Gambar 2. 8 Kerangka Berfikir

- (1) Penetapan masalah dilakukan dengan mengidentifikasi bahwa proses pemantauan barang masih belum akurat dan belum efektif karena pengawasan secara konvensional memerlukan waktu yang lama dan rentan terhadap kesalahan manusia.
- (2) Pendekatan yang digunakan adalah dengan memanfaatkan teknologi Internat of Things (IoT), algoritma YOLO (You Only Look Once) dan pemrograman untuk

mendeteksi objek. Pendekatan ini bertujuan untuk meningkatkan kecepatan dan akurasi dalam pemantauan barang.

- (3) Penerapan sistem dilakukan dengan mengolah data barang dan data gambar menggunakan bahasa pemrograman seperti C++, Python, PHP, JavaScript, dan SQL agar informasi yang diperoleh dari hasil deteksi barang dapat tersimpan dan diproses dengan baik.
- (4) Pengembangan dilakukan dengan membangun implementasi awal sistem (*prototyping*) yang mengintegrasikan teknologi Internet of Things dan algoritma YOLO ke dalam sistem berbasis *Computer vision*. Tujuannya adalah membangun sistem deteksi objek menggunakan gambar dari kamera serta penumpukan data sensor yang mendukung pemantauan kondisi barang.
- (5) Pengukuran kinerja sistem bertujuan untuk mengevaluasi sistem yang telah dikembangkan. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk menguji fungsionalitas sistem, PSSUQ (*Post-Study System Usability Questionnaire*) untuk menilai aspek kegunaan dan kepuasan pengguna, serta *Confusion Matrix* untuk mengevaluasi akurasi deteksi objek.
- (6) Hasil yang diharapkan dari penelitian ini adalah menghasilkan sistem pemantauan barang yang lebih akurat dan efektif dalam mendeteksi ketersediaan barang di rak, sehingga dapat membantu pengelolaan barang secara optimal.

D. Hipotesis

Penelitian ini menggunakan teknologi IoT dengan modul *ESP32-CAM* dan algoritma pengolahan citra untuk mendeteksi kekosongan barang di rak. Berdasarkan penelitian sebelumnya, penerapan teknologi IoT dan algoritma pengolahan citra dalam pemantauan berbagai sistem telah menunjukkan tingkat keberhasilan rata-rata sebesar 90% dalam sepuluh penelitian terkait. Penggunaan teknologi IoT dan pengolahan citra dalam penelitian ini dirancang untuk mengatasi permasalahan pemantauan barang secara konvensional yang belum efektif. Berdasarkan pemahaman di atas, maka hipotesis penelitian ini adalah bahwa penerapan teknologi IoT dengan modul *ESP32-CAM* diduga akurat dan efektif untuk mendeteksi kekosongan barang di rak.