

BAB II

TINJAUAN PUSTAKA

2.1 Hasil Penelitian Terdahulu

Untuk mendalami masalah yang menjadi fokus penelitian ini, peneliti melakukan upaya untuk menelusuri berbagai literatur dan penelitian-penelitian terdahulu yang masih memiliki keterkaitan atau relevansi dengan isu yang sedang diteliti. Hal ini penting dilakukan sebagai bagian dari proses penelitian ilmiah, guna mendapatkan wawasan yang lebih luas dan mendalam tentang topik yang sedang dikaji. Selain itu, sangat penting untuk menjaga integritas dalam penelitian ilmiah, salah satunya dengan menghindari tindakan plagiarisme, yaitu menyalin hasil karya orang lain tanpa memberikan penghargaan yang semestinya.

Untuk memenuhi standar etika penelitian ilmiah yang berlaku, eksplorasi terhadap penelitian-penelitian terdahulu yang relevan menjadi langkah yang sangat diperlukan. Tujuan dari eksplorasi ini adalah untuk memperkuat fondasi penelitian yang sedang dilakukan, serta untuk menunjukkan posisi penelitian ini dalam konteks pengetahuan yang sudah ada. Lebih jauh lagi, penelitian terdahulu berfungsi sebagai teori pendukung yang dapat membantu peneliti dalam merumuskan konsep-konsep dan pendekatan berpikir yang digunakan dalam penelitian ini.

Setelah melakukan eksplorasi mendalam terhadap sejumlah penelitian yang sudah ada sebelumnya, peneliti menemukan beberapa kajian yang memiliki keterkaitan erat dengan penelitian ini. Meskipun terdapat persamaan dalam beberapa aspek, penting untuk dicatat bahwa penelitian ini tetap memiliki perbedaan yang signifikan dari penelitian-penelitian yang telah dilakukan sebelumnya. Berikut ini adalah beberapa penelitian terdahulu yang relevan dan menjadi referensi dalam studi ini:

1. *Eye Fatigue Detection in Vehicle Drivers Based on Facial Landmarks Features*. oleh Fia Dumi Hasanatul Fikriyah, Anan Nugroho , Alfa Faridh Suni (2021)

Penelitian ini bertujuan untuk mendeteksi kelelahan mata pada pengemudi kendaraan menggunakan kamera *smartphone*, khususnya *Droid Cam*, yang dapat memperoleh gambar wajah pengemudi dan mendeteksi mata melalui teknologi pengenalan wajah. Pengembangan aplikasi yang digunakan dalam penelitian ini memanfaatkan OpenCV, sebuah pustaka pemrograman komputer vision yang diterapkan dengan menggunakan bahasa pemrograman Python. Proses deteksi dimulai dengan identifikasi wajah pengemudi menggunakan algoritma landmark wajah 68 titik, yang kemudian digunakan untuk menemukan titik-titik sekitar area mata. Dari titik-titik tersebut, dilakukan perhitungan rasio aspek mata (*EAR*) untuk mendeteksi apakah mata pengemudi dalam kondisi terbuka atau tertutup.

Melalui analisis tersebut, penelitian ini dapat menunjukkan apakah pengemudi mengalami kelelahan berdasarkan pergerakan mata, khususnya frekuensi kedipan mata yang terjadi dalam satu menit. Kedipan mata yang jarang atau terlalu sedikit dalam periode waktu tertentu menjadi indikator pengemudi yang lelah atau mengantuk. Sebaliknya, kedipan mata yang lebih normal atau lebih sering dapat menunjukkan bahwa pengemudi dalam keadaan bugar. Dengan memonitor dan menganalisis jumlah kedipan mata ini, aplikasi dapat memberikan informasi yang penting kepada pengemudi tentang kondisinya, seperti memberi peringatan bahwa ia mungkin mengantuk dan perlu beristirahat sejenak untuk menjaga keselamatan dalam perjalanan.

2. Deteksi Mata Mengantuk pada Pengemudi Mobil Menggunakan Metode *Viola Jones*. Oleh Imanuddin, Rahmat Maulana, Munawir (2019)

Penelitian ini bertujuan untuk mengimplementasikan metode *Viola-Jones* dalam sistem deteksi mata kantuk yang sederhana dengan memanfaatkan berbagai library yang tersedia dalam MATLAB. Sistem ini dirancang untuk mendeteksi kondisi mata kantuk pada pengendara kendaraan bermotor. Setelah sistem selesai dibangun, dilakukan pengujian untuk mengukur karakteristik deteksi mata kantuk yang dapat dikenali oleh sistem

tersebut. Tujuan utama dari deteksi mata kantuk ini adalah untuk mengetahui apakah seorang pengemudi sedang mengantuk atau tidak saat berkendara. Sistem bekerja dengan cara menerima input berupa gambar deteksi mata yang diambil menggunakan kamera digital, kemudian gambar tersebut dimasukkan ke dalam program berbasis *Graphical User Interface* (GUI) MATLAB. Dalam GUI MATLAB ini, sistem akan memproses gambar untuk mendeteksi apakah mata pengemudi menunjukkan tanda-tanda kantuk atau tidak. Hasil analisis mata, yang berupa nilai biner yang mengindikasikan kondisi mengantuk atau tidak mengantuk, kemudian digunakan sebagai referensi untuk memproses lebih lanjut. Berdasarkan data yang dihasilkan, sistem akan memberikan output berupa suara peringatan secara otomatis, yang memberi tahu pengemudi apakah mereka sedang mengantuk, sehingga dapat meningkatkan kewaspadaan pengemudi dan mengurangi risiko kecelakaan.

3. Sistem Monitoring Mendeteksi Mata Lelah Pada Pengemudi Kendaraan Besar Berbasis Pengolahan Citra. Oleh Riza Samsinar , Haris Isyanto , Deni Almanda , Fachri Amrullah. (2024)

Penelitian ini berfokus pada pengolahan citra untuk mendeteksi tanda-tanda kelelahan pada pengemudi secara dini, khususnya bagi pengemudi truk logistik yang sering mengemudi pada malam hari. Mengingat risiko kecelakaan yang dapat disebabkan oleh kelelahan dan rasa kantuk, penting untuk memiliki sistem yang dapat mendeteksi secara otomatis apakah pengemudi sedang tertidur atau dalam keadaan sadar. Oleh karena itu, penelitian ini menggunakan metode *facial landmark* untuk mendeteksi tanda-tanda kantuk pada pengemudi.

4. *Computer vision-based approach to detect fatigue driving and face mask for edge computing device*. Oleh Ashiqur Rahman, Mamun Bin Harun Hriday, Riasat Khan (2022)

Penelitian ini menyajikan sistem tertanam yang lengkap untuk mendeteksi kelelahan saat mengemudi dengan menggunakan pendekatan yang menggabungkan pembelajaran mendalam, *computer vision*, dan

pemantauan detak jantung. Sistem ini dibangun menggunakan kit pengembang Nvidia Jetson Nano, Arduino Uno, dan modul detak jantung AD8232, yang bekerja bersama untuk memantau kondisi pengemudi secara real-time. Dengan menggunakan modul kamera yang terus memantau pengemudi, sistem ini mampu mengambil bingkai gambar yang kemudian dianalisis oleh sistem inti berbasis Nvidia Jetson Nano. Teknik pembelajaran mendalam dan *computer vision* digunakan untuk mengidentifikasi kondisi pengemudi dengan mengandalkan pendekatan lokalisasi mata dan mulut berdasarkan 68 *landmark* wajah yang berbeda. Untuk mengklasifikasikan status kelelahan, data ambang batas yang diperoleh secara eksperimen digunakan sebagai dasar penilaian.

5. Pendeteksi Kantuk Berbasis *Face Recognition* Dan Detak Jantung Pengemudi Roda Empat. Oleh Faiz Dhiaulhaq, Muhammad Farid Dhiyaulhaq, dan Muftito Muhammad. (2022)

Penelitian ini menggunakan metode *design thinking*, dirancang sebuah alat yang bertujuan untuk mendeteksi kondisi pengemudi melalui analisis wajah dan detak jantung. Alat ini bekerja menggunakan dua perangkat utama yang diprogram, yaitu ESP32 dan Raspberry Pi 3B. Sensor detak jantung yang diprogram melalui ESP32 akan dipasang pada ujung jari pengemudi, sementara pendeteksi wajah menggunakan kamera yang terhubung ke Raspberry Pi 3B, yang telah diprogram untuk menganalisis kondisi wajah pengemudi. Kamera ini nantinya akan diletakkan pada *dashboard* mobil untuk memantau pengemudi secara *real-time*.

6. Deteksi Kantuk pada Pengemudi melalui Jumlah Kedipan Mata Menggunakan *Facial Landmark* berbasis Intel NUC. Oleh Dewi Amalia, Fitri Utaminingrum. (2021)

Penelitian ini menggunakan metode *Facial Landmark* untuk mendeteksi area mata pada wajah sebagai bagian dari sistem deteksi kantuk. Dalam kondisi pencahayaan yang kurang atau tidak merata, penelitian ini mengandalkan fitur image thresholding, khususnya *adaptive thresholding*

gaussian, untuk meningkatkan akurasi deteksi. Pendeteksian area mata dan tingkat kantuk dilakukan menggunakan kamera yang ditempatkan pada jarak 30 cm, 40 cm, dan 50 cm dari pengemudi, dengan posisi kamera sejajar dengan bahu atau dada. Proses ini dilakukan dalam dua kondisi intensitas cahaya yang berbeda, yaitu 0-49 lux dan 50-400 lux.

7. *A Novel Multi-Level Driver Fatigue Detection System Using Deep Learning*. Oleh B. Sai, S.S.V.R. Kumar (2025)

Penelitian ini bertujuan untuk memperluas deteksi kantuk pengemudi di kendaraan menggunakan sinyal kamera pemantau pengemudi. Untuk tujuan ini, fitur yang terkait dengan perilaku kedipan mata pengemudi dan gerakan kepala diekstraksi dalam eksperimen simulator mengemudi. Berdasarkan kumpulan data besar tersebut, kami mengembangkan dan mengevaluasi metode pemilihan fitur berdasarkan algoritma *k-Nearest Neighbor* untuk klasifikasi kondisi pengemudi. Analisis akhir dari kumpulan fitur dengan kinerja terbaik menghasilkan wawasan berharga tentang pengaruh kantuk pada perilaku kedipan pengemudi dan gerakan kepala. Temuan ini akan membantu dalam pengembangan sistem pemantauan kantuk pengemudi yang kuat dan andal di masa mendatang untuk mencegah kecelakaan akibat kelelahan.

8. *Deteksi Mata dan Alis Menggunakan AdaBoost Classifier dan Haar Cascade*. Oleh Prabiantissa, C. N., & Muchamad, K. (2024)

Penelitian ini bertujuan untuk mengimplementasikan deteksi mata dan alis menggunakan metode *Haar Cascade*. Beberapa proses *Haar Cascade* yang dilakukan diantaranya *Preprocessing*, *Integral Image*, *Adaboost*, dan *Cascade*. Hasil penelitian menunjukkan metode *Haar Cascade* berhasil dalam mendeteksi objek mata dan alis dengan cukup baik dengan tingkat akurasi 95% pada data wajah bermasker, 90% pada wajah berkacamata, 87% pada wajah berkacamata dan miring, 87% pada wajah berkacamata, miring dan mata tertutup

9. *Detection of Driver Drowsiness Using Adaptive Eye Characteristic Ratio*. Oleh P.S. Lamba et al. (2025)

Penelitian ini menyajikan sistem deteksi waktu nyata yang dirancang untuk mendeteksi tidur pengemudi dan memberikan peringatan tepat waktu. Mekanisme tersebut menggunakan sistem pemantauan mata yang menggunakan titik fitur mata untuk mendeteksi apakah mata pengemudi lelah dan memberikan peringatan jika pengemudi mengantuk. Pekerjaan yang diusulkan mengenali wajah pengemudi dengan mengumpulkan penanda wajah dan memberikan peringatan kepada pengemudi untuk menghindari kecelakaan dengan melakukan ambang batas adaptif dan diagnosis kantuk berdasarkan Rasio Karakteristik Mata. Hasil empiris menunjukkan bahwa dengan menggunakan pengklasifikasi hutan acak, pengklasifikasi pohon keputusan, pengklasifikasi KNN, dan pengklasifikasi multinomial, model yang diusulkan mencapai akurasi masing-masing sebesar 90,48%, 89,44%, 88,63%, dan 84,32%.

10. *Driver Drowsiness Shield (DDSH): A Real-time Driver Drowsiness Detection System*. Oleh A. Bhanja et al. (2025)

Penelitian ini bertujuan untuk mengembangkan sistem deteksi rasa kantuk waktu nyata yang canggih menggunakan algoritma pembelajaran mendalam. Untuk tujuan ini, kami menggunakan kumpulan data citra mata dari *MRL Eye Dataset* dan melakukan rekayasa fitur dan praproses yang ekstensif untuk menyiapkan data untuk analisis. Sebuah algoritma telah diusulkan untuk mengklasifikasikan kondisi mata sebagai terbuka atau tertutup menggunakan *Transfer Learning* berdasarkan arsitektur *MobileNet*. Menggunakan kumpulan data yang seimbang, model dilatih untuk membedakan antara mata terbuka dan tertutup secara akurat. Hasil eksperimen menunjukkan keefektifan pendekatan kami, mencapai akurasi 90%, presisi 100%, ingatan 83,3%, dan skor F1 sebesar 90,9%. Untuk memvalidasi sistem lebih lanjut, kami mengintegrasikan model yang dilatih ke dalam aplikasi kamera waktu nyata yang memantau kondisi mata pengemudi. Aplikasi menganalisis aliran video waktu nyata, mendeteksi wajah dan mata, dan menggunakan contoh kelopak mata tertutup untuk

memprediksi tanda-tanda rasa kantuk. Efisiensi model dievaluasi menggunakan metrik seperti akurasi, presisi, daya ingat, dan skor F1

2.2 Tinjauan Pustaka

2.2.1 Kelelahan dan Kantuk

Kelelahan dapat dipandang sebagai suatu mekanisme perlindungan tubuh yang bertujuan untuk mencegah kerusakan lebih lanjut pada tubuh, sehingga proses pemulihan dapat berlangsung. Penting untuk membedakan antara kelelahan dan kejemuhan, meskipun keduanya saling terkait. Kejemuhan sendiri merujuk pada kondisi di mana lingkungan kerja tidak memberikan rangsangan yang cukup bagi tenaga kerja, yang bisa terjadi ketika pekerjaan menjadi monoton atau tidak menantang. Kejemuhan muncul ketika pekerjaan tidak lagi mampu menarik perhatian atau memotivasi individu, misalnya karena tugas yang dilakukan tidak memerlukan keterampilan khusus atau lingkungan kerja yang statis. Meskipun demikian, keadaan kejemuhan dapat segera dipulihkan jika pekerjaan tersebut kembali menawarkan rangsangan atau tantangan baru, seperti pekerjaan yang melibatkan interaksi lebih aktif atau yang tidak sepenuhnya dikendalikan oleh mesin (Haditia, 2012).

Berikut ini pengertian kelelahan kerja menurut beberapa ahli yaitu;

- 1) Notoatmodjo (2007), Menurut Notoatmodjo, kelelahan adalah suatu keadaan di mana tubuh merasa lemah dan tidak mampu lagi berfungsi dengan baik setelah menjalani aktivitas dalam jangka waktu yang panjang. Kelelahan ini bisa disebabkan oleh berbagai faktor, seperti fisik, emosional, atau mental. Di sisi lain, kelelahan juga dipengaruhi oleh faktor lingkungan dan gaya hidup, seperti pola makan yang buruk, kurang tidur, atau stres yang berkepanjangan.
- 2) Widyasari (2010), kelelahan kerja merujuk pada berbagai kondisi yang ditandai dengan penurunan efisiensi dan daya tahan dalam bekerja. Kelelahan ini dapat muncul dari berbagai sumber, seperti kelelahan visual yang disebabkan oleh penggunaan mata yang berlebihan, kelelahan fisik yang terkait dengan kerja tubuh yang intens, kelelahan

saraf yang mempengaruhi sistem saraf, serta kelelahan yang disebabkan oleh lingkungan kerja yang monoton dan tidak berubah. Selain itu, kelelahan kerja juga bisa disebabkan oleh faktor lingkungan yang terus-menerus dan bersifat kronis, yang berlangsung dalam jangka waktu lama dan menyebabkan penurunan performa secara bertahap. Semua faktor ini berkontribusi pada berkurangnya kapasitas individu dalam melakukan pekerjaan secara optimal.

Kantuk adalah suatu kondisi yang terjadi sebagai peralihan antara keadaan terjaga dan tidur. Rasa kantuk yang muncul pada malam hari setelah menjalani aktivitas seharian adalah hal yang normal, karena tubuh memang membutuhkan waktu untuk beristirahat. Namun, ketika rasa kantuk datang pada siang hari, terutama saat kita sedang aktif beraktivitas, hal ini menjadi masalah karena dapat mengganggu kelancaran kegiatan sehari-hari. Kantuk yang terjadi di luar waktu tidur yang seharusnya bisa menurunkan konsentrasi dan produktivitas. Ada beberapa faktor yang dapat menyebabkan seseorang sering merasa mengantuk, yang akan dijelaskan lebih lanjut dalam poin-poin berikut.

1) Gaya hidup

Gaya hidup tertentu dapat menjadi penyebab seseorang sering merasa mengantuk, seperti kebiasaan bekerja hingga larut malam yang mengganggu pola tidur, atau melakukan perjalanan jauh yang mengakibatkan *jetlag*. Kedua kondisi ini dapat mengganggu gaya hidup tubuh, sehingga menyebabkan rasa kantuk yang berlebihan, bahkan saat waktu yang seharusnya digunakan untuk beraktivitas.

2) Kesehatan mental

Sering mengantuk juga bisa dipengaruhi oleh kondisi kesehatan mental yang tidak baik, seperti depresi atau stres. Kedua gangguan mental ini dapat mengganggu kualitas tidur dan menyebabkan kelelahan yang berlebihan sepanjang hari. Depresi, misalnya, sering kali membuat seseorang merasa lelah dan tidak berenergi, meskipun

sudah tidur cukup lama. Sementara stres dapat menyebabkan kecemasan dan ketegangan yang menghambat tidur yang nyenyak, sehingga tubuh tidak mendapatkan istirahat yang cukup untuk memulihkan diri.

3) Obat-obatan

Beberapa jenis obat-obatan, seperti obat penenang dan obat tidur, dapat menyebabkan efek samping berupa rasa kantuk yang berlebihan. Penggunaan obat-obatan ini dapat mengganggu kewaspadaan dan mempengaruhi kemampuan seseorang untuk tetap terjaga dan fokus. Oleh karena itu, sangat tidak disarankan untuk mengonsumsi obat-obatan tersebut sebelum melakukan aktivitas yang memerlukan konsentrasi tinggi, seperti berkendara, karena dapat meningkatkan risiko kecelakaan akibat rasa kantuk yang mendalam dan berkurangnya reaksi tubuh.

4) Gangguan tidur

Gangguan tidur, seperti insomnia dan *sleep apnea*, dapat menjadi penyebab utama seseorang sering merasa mengantuk sepanjang hari. Insomnia, yang disebabkan oleh kualitas tidur yang buruk atau kesulitan untuk tidur, mengakibatkan tubuh tidak mendapatkan istirahat yang cukup, sehingga menyebabkan rasa kantuk yang berlebihan. Sementara itu, *sleep apnea* adalah gangguan tidur di mana penderita mengalami henti napas sementara saat tidur, sering disertai dengan mendengkur keras dan terbangun secara mendadak di tengah malam. Kondisi ini mengganggu siklus tidur yang normal dan menyebabkan tidur yang tidak nyenyak, sehingga tubuh tidak dapat pulih dengan baik dan menyebabkan kantuk berlebihan di siang hari.

2.2.2 Ketertiban dan Keselamatan Lalu Lintas

Safety riding merupakan suatu bentuk perilaku berkendara yang berfokus pada upaya menjaga keselamatan, baik bagi pengendara itu sendiri maupun bagi pengguna jalan lainnya. Konsep ini mencakup pemahaman menyeluruh mengenai pentingnya keselamatan di jalan raya serta langkah-

langkah pencegahan dan penanggulangan terhadap potensi kecelakaan lalu lintas. Dalam praktiknya, penerapan *safety riding* tidak hanya berkaitan dengan keterampilan teknis mengemudi, tetapi juga mencerminkan kesadaran hukum dan tanggung jawab sosial sebagai pengguna jalan. Aspek-aspek keselamatan dalam berkendara ini telah diatur secara hukum melalui Undang-Undang Republik Indonesia Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, serta diperkuat oleh Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 12 Tahun 2019 yang secara khusus mengatur tentang perlindungan keselamatan bagi pengguna sepeda motor, terutama yang digunakan untuk kepentingan masyarakat luas.

1. Tata cara berlalu lintas

Pengendara harus menaati aturan lalu lintas sesuai UU No. 22 Tahun 2009. Pasal 105 menyebutkan bahwa pengguna jalan wajib tertib dan mencegah hal yang membahayakan keselamatan atau merusak jalan. Dalam pasal 106 menjelaskan bahwa Pengemudi kendaraan bermotor wajib berkendara dengan wajar dan fokus, mengutamakan keselamatan pejalan kaki dan pesepeda, serta mematuhi aturan teknis kendaraan, rambu, marka, APILL, gerakan lalu lintas, aturan parkir, peringatan, kecepatan, dan cara menggandeng kendaraan. Pengendara dan penumpang motor juga wajib pakai helm berstandar SNI.

2. Memeriksa kondisi sepeda motor

Sebelum berkendara, pastikan roda, lampu, rem, dan mesin dalam kondisi baik. Cek tekanan angin ban, fungsi lampu dan rem, lalu panaskan mesin 2–3 menit agar pelumas merata.

3. Identifikasi kendaraan bermotor

Menurut UU No. 22 Tahun 2009 Pasal 68, setiap kendaraan bermotor wajib memiliki STNK dan TNKB. STNK berisi data kendaraan, pemilik, nomor registrasi, dan masa berlaku, sedangkan TNKB memuat kode wilayah, nomor registrasi, dan masa berlaku sesuai ketentuan.

4. Surat Izin Mengemudi (SIM)

Menurut UU No. 22 Tahun 2009, Pasal 77 menyebutkan bahwa pengendara wajib memiliki SIM sesuai jenis kendaraannya. SIM C digunakan untuk sepeda motor (Pasal 80). Untuk mendapatkannya, pemohon harus berusia minimal 17 tahun, memenuhi syarat administrasi, kesehatan, dan lulus ujian (Pasal 81).

5. Penggunaan helm

Pasal 106 ayat 8 UU No. 22 Tahun 2009 mewajibkan pengendara dan penumpang motor memakai helm berstandar SNI. WHO menyatakan helm bisa mengurangi risiko kematian hingga 40% saat kecelakaan. Helm berfungsi melindungi kepala (SNI 1811-2007).

2.2.3 Mikrokontroler

Mikrokontroler merupakan sebuah sirkuit terpadu (*Integrated Circuit* atau IC) yang dirancang dengan komponen-komponen elektronik dalam bentuk yang sangat kompak untuk mengatur dan mengendalikan berbagai operasi dalam sistem tertanam (*embedded system*). Secara umum, mikrokontroler terdiri dari beberapa elemen penting, seperti prosesor yang berfungsi sebagai otak pengendali, memori untuk menyimpan data atau program, serta bagian *input/output* (I/O) periferi yang memungkinkan komunikasi antara mikrokontroler dengan perangkat eksternal lainnya, semua ini digabungkan dalam satu chip tunggal. Dalam beberapa kasus, mikrokontroler juga sering disebut sebagai pengendali tertanam atau unit mikrokontroler, karena peranannya yang sangat penting dalam mengatur berbagai fungsi dalam sebuah sistem tertanam. Mikrokontroler ini dapat ditemukan dalam berbagai perangkat sehari-hari, mulai dari kendaraan bermotor, robot, mesin kantor, peralatan medis, pemancar radio bergerak, mesin penjual otomatis, hingga berbagai peralatan rumah tangga. Saat ini, mikrokontroler semakin berkembang, dengan banyak di antaranya yang sudah terintegrasi dengan komponen-komponen elektronik lainnya, sehingga memudahkan penggunaannya untuk berbagai keperluan dan aplikasi, salah

satunya adalah mikrokontroler berbasis Arduino yang populer di kalangan penggemar teknologi dan pendidikan.

Mikrokontroler bisa dianggap sebagai "otak" dalam sebuah sistem elektronik yang lebih kecil dan lebih terfokus. Meskipun prinsip kerjanya mirip dengan komputer, di mana mikrokontroler juga mampu menjalankan program, ruang lingkup aplikasi yang dijalankan lebih terbatas dan spesifik. Mikrokontroler umumnya digunakan dalam situasi di mana kontrol otomatis dibutuhkan, seperti untuk memproses data input dan mengendalikan *output* dengan tingkat kecepatan yang sangat tinggi. Keunggulan utama dari mikrokontroler adalah kemampuannya untuk melaksanakan tugas-tugas ini secara efisien dalam berbagai sistem yang membutuhkan pengolahan data secara cepat dan akurat, serta kontrol yang terprogram dengan tepat. Mikrokontroler sering menjadi pilihan utama dalam berbagai perangkat elektronik modern karena ukurannya yang kecil, konsumsi daya yang rendah, dan kemampuannya untuk beroperasi secara mandiri dalam menjalankan fungsi tertentu tanpa memerlukan perangkat keras tambahan yang besar atau kompleks.

Mikrokontroler memiliki prosesor yang berfungsi sebagai "otak" dalam sistemnya, dan prosesor ini hadir dalam berbagai ukuran bit. Secara umum, ukuran bit prosesor mikrokontroler bisa bervariasi, mulai dari yang sederhana dengan ukuran 4-bit, 8-bit, atau 16-bit, hingga yang lebih kompleks seperti mikrokontroler 32-bit dan 64-bit. Mikrokontroler yang lebih sederhana dengan prosesor 4-bit hingga 16-bit biasanya digunakan dalam aplikasi yang memerlukan pengolahan data terbatas, sementara prosesor 32-bit dan 64-bit digunakan dalam sistem yang lebih canggih, memungkinkan pemrosesan data yang lebih cepat dan kemampuan yang lebih besar. Selain prosesor, mikrokontroler juga dilengkapi dengan berbagai jenis memori, seperti *Random Access Memory* (RAM), memori flash, serta EPROM atau EEPROM, yang semuanya berfungsi untuk menyimpan data sementara, program, atau data yang perlu dipertahankan meskipun daya dimatikan. Mikrokontroler pada dasarnya dirancang untuk kemudahan penggunaan,

dengan memori *onboard* yang cukup dan pin untuk operasi *input/output* (I/O), memungkinkan mikrokontroler berinteraksi langsung dengan sensor dan komponen lainnya tanpa memerlukan tambahan perangkat komputasi yang kompleks.

Dari segi arsitektur, mikrokontroler dapat menggunakan dua jenis arsitektur utama, yaitu arsitektur Harvard dan arsitektur von Neumann, yang masing-masing memiliki karakteristik dalam cara data dipertukarkan antara prosesor dan memori. Pada arsitektur Harvard, bus data dan instruksi dipisah, sehingga memungkinkan transfer data dan instruksi secara bersamaan (simultan), yang dapat meningkatkan kecepatan proses. Sebaliknya, pada arsitektur von Neumann, baik data maupun instruksi berbagi satu bus yang sama, yang dapat membatasi kecepatan transfer karena instruksi dan data harus bergantian melalui bus tersebut. Kedua arsitektur ini memiliki kelebihan dan kekurangan masing-masing, tergantung pada jenis aplikasi yang diinginkan.

Selain itu, prosesor mikrokontroler juga dapat didasarkan pada dua pendekatan komputasi yang berbeda, yaitu *Complex Instruction Set Computing* (CISC) dan *Reduced Instruction Set Computing* (RISC). Prosesor CISC biasanya memiliki jumlah instruksi yang lebih banyak, sekitar 80 instruksi, dengan banyak mode pengalamatan yang lebih kompleks, yang membuatnya lebih fleksibel dan efisien dalam penggunaan memori. Namun, karena jumlah siklus *clock* yang lebih banyak untuk menjalankan instruksi, kinerja prosesor CISC bisa lebih rendah. Sebaliknya, prosesor RISC memiliki set instruksi yang lebih sederhana, biasanya hanya sekitar 30 instruksi, dan lebih mengutamakan perangkat lunak untuk mencapai efisiensi. Dengan instruksi yang lebih sederhana, prosesor RISC cenderung lebih cepat dalam menjalankan instruksi dan memberikan kinerja yang lebih baik. Namun, karena bergantung lebih pada perangkat lunak, desain perangkat lunak untuk RISC bisa menjadi lebih kompleks. Pemilihan antara CISC dan RISC pada mikrokontroler sangat bergantung pada aplikasi spesifik yang ingin dijalankan, serta kebutuhan kinerja dan efisiensi yang diinginkan.

Awalnya, mikrokontroler hanya dapat diprogram menggunakan bahasa *assembly*, yang mengharuskan pemrogram untuk menulis kode mesin yang sangat spesifik dan rendah tingkatannya. Namun, seiring berjalannya waktu, kemajuan dalam teknologi pemrograman memungkinkan penggunaan bahasa pemrograman yang lebih tinggi dan lebih mudah dipahami, seperti bahasa C dan Python, yang kini banyak digunakan untuk pemrograman mikrokontroler. Dengan bahasa pemrograman tersebut, proses pengembangan aplikasi untuk mikrokontroler menjadi lebih cepat dan lebih mudah. Beberapa contoh fisik mikrokontroler yang populer adalah Raspberry Pi dan Arduino, yang telah digunakan secara luas di berbagai proyek elektronik dan pendidikan. Selain itu, mikrokontroler kini juga memainkan peran penting dalam dunia *Internet of Things* (IoT), di mana mikrokontroler digunakan sebagai komponen utama dalam menghubungkan berbagai perangkat elektronik dengan internet, memungkinkan perangkat-perangkat tersebut untuk saling berkomunikasi dan bertukar data secara otomatis. Mikrokontroler dalam konteks IoT memungkinkan terciptanya berbagai aplikasi canggih, mulai dari perangkat rumah pintar hingga sistem otomatisasi industri.

Mikrokontroler terdiri dari beberapa komponen dasar yang mendukung kinerjanya. Berikut adalah komponen utama dalam mikrokontroler:

1. Unit Pemrosesan Sentral (CPU)

CPU (*Central Processing Unit*) adalah komponen utama yang bertanggung jawab untuk menjalankan instruksi program serta memproses data dalam sebuah sistem komputer atau mikrokontroler. Dalam konteks mikrokontroler, CPU berperan sebagai pusat kendali yang mengatur seluruh operasi sistem, mulai dari pengolahan data yang diterima dari sensor hingga pengendalian perangkat *output*. CPU mikrokontroler menerima instruksi program yang disimpan dalam memori dan mengeksekusinya untuk melakukan tugas-tugas tertentu. Misalnya, CPU akan memproses data dari sensor, seperti suhu atau

cahaya, dan kemudian mengirimkan perintah untuk mengendalikan perangkat *output*, seperti menyalakan LED, menggerakkan motor, atau mengaktifkan alarm.

Dengan kata lain, CPU dalam mikrokontroler bertindak sebagai otak yang mengatur dan mengoordinasikan seluruh proses dalam sistem tertanam tersebut. Setiap kali sistem menerima input dari dunia luar, seperti data dari sensor, CPU akan memproses informasi tersebut sesuai dengan program yang telah diinstal, dan kemudian menghasilkan *output* yang sesuai. Pengendalian ini bisa terjadi dalam waktu yang sangat cepat dan efisien, yang membuat mikrokontroler ideal untuk berbagai aplikasi yang memerlukan pengolahan data otomatis dan kontrol *real-time*. Sebagai pusat pengendali, CPU juga berperan dalam memastikan bahwa komunikasi antar komponen mikrokontroler, seperti memori, sensor, dan perangkat output, berjalan dengan lancar dan sesuai dengan instruksi yang telah ditentukan.

2. Memori

Memori pada mikrokontroler terbagi menjadi dua jenis utama, yaitu memori baca-tulis (RAM) dan memori baca-saja (ROM), masing-masing memiliki peran yang sangat penting dalam menjalankan sistem mikrokontroler. RAM (*Random Access Memory*) digunakan untuk menyimpan data sementara yang sedang diproses oleh CPU selama eksekusi program. Misalnya, ketika mikrokontroler menerima input dari sensor atau menjalankan perhitungan, data tersebut disimpan sementara di RAM untuk diproses dan digunakan dalam instruksi selanjutnya. Karena sifatnya yang dinamis dan cepat diakses, RAM memungkinkan CPU untuk bekerja dengan efisien, melakukan pembacaan dan penulisan data dalam waktu yang sangat singkat. Namun, karena RAM bersifat volatil, data yang ada di dalamnya akan hilang begitu daya dimatikan atau sistem dimulai ulang.

Di sisi lain, ROM (*Read-Only Memory*) menyimpan program atau kode yang dijalankan oleh mikrokontroler, dan sifatnya adalah *non-volatile*, artinya data yang tersimpan di dalamnya tetap ada meskipun mikrokontroler dimatikan. ROM berfungsi untuk menyimpan instruksi program atau firmware yang mengontrol operasi mikrokontroler secara permanen. Terdapat beberapa jenis ROM yang dapat digunakan dalam mikrokontroler, seperti EPROM (*Erasable Programmable Read-Only Memory*) atau EEPROM (*Electrically Erasable Programmable Read-Only Memory*), yang memungkinkan perubahan data atau kode meskipun hanya dalam keadaan tertentu, tetapi umumnya ROM bersifat tidak dapat diubah setelah diprogram. ROM memainkan peran yang sangat vital dalam memastikan mikrokontroler dapat memulai dan menjalankan program secara konsisten setiap kali sistem dihidupkan.

Secara keseluruhan, kombinasi RAM dan ROM dalam mikrokontroler memungkinkan sistem untuk menyimpan data yang sedang diproses sekaligus menjaga kestabilan dan konsistensi operasi dengan menyimpan program yang diperlukan untuk menjalankan berbagai tugas dan fungsi sistem. RAM memberikan fleksibilitas dalam pengolahan data sementara ROM menyediakan "petunjuk tetap" untuk melaksanakan instruksi yang telah diprogramkan.

3. Input/Output (I/O)

Mikrokontroler dilengkapi dengan *port input/output (I/O)* yang memungkinkan komunikasi antara mikrokontroler dan perangkat eksternal, seperti sensor, motor, layar, atau komponen lainnya. Port I/O ini berfungsi sebagai antarmuka yang memungkinkan mikrokontroler untuk berinteraksi dengan lingkungan luar dan menjalankan tugas-tugas yang terhubung dengan sistem yang lebih besar. Port I/O dapat dibagi menjadi dua kategori utama, yaitu *input* dan *output*, yang masing-masing memiliki fungsi dan peran yang spesifik dalam proses komunikasi data.

Port input pada mikrokontroler digunakan untuk menerima data dari perangkat eksternal, seperti sensor suhu, sensor cahaya, tombol, atau perangkat lainnya yang mengirimkan informasi ke mikrokontroler. *Input* ini bisa berupa *input* digital maupun analog. Pada *input* digital, mikrokontroler hanya dapat menerima dua nilai, yaitu tinggi (biasanya 1) atau rendah (0), seperti pada tombol tekan atau saklar. Sementara itu, *input* analog memungkinkan mikrokontroler untuk menerima sinyal kontinu, seperti data dari sensor suhu atau sensor cahaya, yang menghasilkan rentang nilai yang dapat diubah secara bertahap (misalnya, nilai antara 0 hingga 1023 dalam sistem 10-bit).

Sebaliknya, *port output* digunakan untuk mengirimkan sinyal atau data dari mikrokontroler ke perangkat eksternal untuk mengendalikan atau mengatur operasi perangkat tersebut. Output ini juga bisa berupa digital atau analog. Pada *output* digital, mikrokontroler mengirimkan sinyal yang hanya memiliki dua nilai (tinggi atau rendah) untuk mengendalikan perangkat seperti LED atau relay. Sedangkan pada *output* analog, mikrokontroler dapat mengeluarkan sinyal yang bersifat kontinu, seperti yang digunakan untuk mengontrol kecepatan motor atau kecerahan LED dengan menggunakan teknik PWM (*Pulse Width Modulation*), di mana mikrokontroler menghasilkan sinyal digital yang seolah-olah bersifat analog dengan mengatur rasio antara waktu sinyal aktif dan non-aktif.

Port I/O ini sangat penting dalam berbagai aplikasi mikrokontroler, karena memungkinkan mikrokontroler untuk berkomunikasi secara langsung dengan dunia fisik, baik untuk menerima informasi dari sensor-sensor yang ada di sekitarnya maupun untuk mengendalikan berbagai perangkat *output* yang merespons data atau perintah dari mikrokontroler.

4. *Timer dan Counter*

Mikrokontroler dilengkapi dengan *timer* dan *counter* yang memungkinkan pemrogram untuk mengukur waktu secara akurat atau menghitung kejadian-kejadian tertentu dalam sistem. Fitur ini sangat penting karena banyak aplikasi mikrokontroler yang memerlukan pengukuran waktu atau penghitung kejadian untuk menjalankan tugas-tugas tertentu dengan tepat. *Timer* pada mikrokontroler biasanya digunakan untuk mengatur interval waktu atau untuk menghasilkan jeda (*delay*) yang diperlukan dalam aplikasi, sedangkan *counter* digunakan untuk menghitung jumlah pulsa atau kejadian yang terjadi dalam waktu tertentu.

Timer bekerja dengan menghitung jumlah siklus *clock* atau frekuensi yang telah ditentukan, dan ketika jumlah tersebut tercapai, *timer* akan menghasilkan *interrupt* atau sinyal untuk memberi tahu mikrokontroler bahwa waktu yang diinginkan telah tercapai. Fungsi *timer* ini sangat berguna dalam berbagai aplikasi yang memerlukan pengaturan waktu yang presisi, seperti dalam pembuatan sistem pengatur waktu (*timer*) untuk menyalakan atau mematikan perangkat secara otomatis setelah periode tertentu, pengaturan interval waktu dalam aplikasi komunikasi, atau untuk menghasilkan sinyal gelombang dengan frekuensi tertentu. Sebagai contoh, *timer* dapat digunakan untuk mengontrol durasi LED yang menyala dalam aplikasi indikator atau untuk mengatur laju komunikasi data dalam protokol serial.

Di sisi lain, *counter* pada mikrokontroler digunakan untuk menghitung jumlah pulsa atau kejadian yang diterima oleh sistem. Misalnya, *counter* dapat digunakan untuk menghitung jumlah putaran motor, jumlah detik yang telah berlalu, atau jumlah pulsa yang diterima dari sensor eksternal seperti sensor kedekatan atau sensor optik. *Counter* bekerja dengan cara menghitung pulsa atau sinyal yang masuk pada input tertentu dan menyimpan hasilnya. Setelah mencapai jumlah yang ditentukan, *counter* dapat menghasilkan *interrupt* atau mengaktifkan aksi tertentu, seperti memulai proses baru atau

memberikan notifikasi. *Counter* sangat berguna dalam aplikasi seperti penghitungan kecepatan, pengukuran jarak dalam sistem pengukuran berbasis sensor, atau dalam aplikasi komunikasi data yang memerlukan penghitungan bit atau *byte* yang diterima.

Secara keseluruhan, *timer* dan *counter* dalam mikrokontroler memberikan kemampuan untuk mengontrol waktu dan menghitung kejadian secara presisi, yang penting untuk aplikasi yang memerlukan akurasi dan respon waktu nyata.

5. Periferal

Beberapa mikrokontroler dilengkapi dengan periferal tambahan yang sangat penting untuk memperluas kemampuannya dan memungkinkan penggunaannya dalam berbagai aplikasi yang lebih kompleks. Periferal tambahan ini antara lain adalah pengontrol komunikasi serial, yang mencakup berbagai protokol seperti UART (*Universal Asynchronous Receiver Transmitter*), SPI (*Serial Peripheral Interface*), dan I2C (*Inter-Integrated Circuit*). Protokol-protokol ini memungkinkan mikrokontroler untuk berkomunikasi dengan perangkat lain seperti sensor, modul komunikasi, dan perangkat eksternal lainnya dengan cara yang efisien dan terstandarisasi. Selain itu, mikrokontroler juga sering dilengkapi dengan pengontrol *analog-to-digital converter* (ADC), yang memungkinkan sinyal analog dari dunia luar diubah menjadi data digital yang dapat diproses oleh mikrokontroler. Begitu pula dengan *digital-to-analog converter* (DAC), yang memungkinkan mikrokontroler untuk mengubah data digital menjadi sinyal analog. Keberadaan periferal ini memperkaya fungsionalitas mikrokontroler, menjadikannya lebih fleksibel dan sesuai untuk berbagai jenis aplikasi, mulai dari sistem otomasi, kendali industri, hingga perangkat *embedded* yang membutuhkan komunikasi dan pemrosesan sinyal yang akurat dan efisien.

2.2.4 Raspberry Pi

Raspberry Pi, yang sering kali disingkat dengan nama Raspi, merupakan sebuah komputer berukuran kecil yang berupa papan tunggal atau dikenal dengan istilah *single-board computer* (SBC). Ukurannya yang sangat *compact*, yakni sebesar kartu kredit, membuatnya sangat praktis dan mudah untuk digunakan dalam berbagai aplikasi, baik itu untuk menjalankan program-program perkantoran, bermain permainan komputer, hingga memutar media, termasuk video dengan resolusi tinggi. Komputer mini ini dikembangkan oleh *Raspberry Pi Foundation*, sebuah organisasi nirlaba yang berkomitmen untuk memajukan teknologi komputer dengan harga yang terjangkau. Proyek ini didorong oleh sekelompok pengembang dan ahli komputer dari Universitas Cambridge di Inggris, yang berusaha menciptakan perangkat komputasi yang dapat diakses oleh banyak orang, baik untuk keperluan pendidikan maupun hiburan. Raspberry Pi pun sejak peluncurannya telah menjadi salah satu perangkat yang sangat populer, terutama di kalangan penggemar teknologi dan pendidikan, berkat fleksibilitas serta kemampuan untuk diprogram sesuai kebutuhan.

1. Raspberry Pi 4 Model B

Raspberry Pi 4 Model B adalah board terbaru yang dirilis oleh Raspberry Pi Foundation pada tahun 2019, dan merupakan peningkatan signifikan dibandingkan dengan versi sebelumnya, Raspberry Pi 3. Dengan berbagai perbaikan dalam hal performa dan spesifikasi, Raspberry Pi 4 Model B menawarkan kecepatan yang jauh lebih tinggi, menjadikannya lebih cocok untuk berbagai aplikasi yang membutuhkan daya komputasi lebih besar. Salah satu varian tertinggi yang dijual adalah model dengan kapasitas RAM 4GB, yang memungkinkan pengguna untuk menjalankan aplikasi yang lebih berat dan melakukan multitasking dengan lebih lancar. Peningkatan kapasitas RAM ini juga membuat Raspberry Pi 4 Model B lebih fleksibel, tidak hanya untuk proyek-proyek hobi, tetapi juga untuk penggunaan yang lebih

profesional, seperti pengembangan perangkat lunak, pengolahan data, dan aplikasi multimedia.



Gambar 2. 1 Raspberry Pi 4 Model B
(Sumber : <https://www.yangcanggih.com/>)

Berikut adalah spesifikasi raspberry pi 4 model b:

- a. Prosesor: *Quad-core* ARM Cortex-A72 1.5 GHz
- b. RAM: Pilihan 2GB, 4GB, atau 8GB LPDDR4-3200 SDRAM
- c. Port USB: 2x USB 3.0, 2x USB 2.0
- d. Konektivitas: Gigabit Ethernet, *Dual-band* Wi-Fi 802.11ac, *Bluetooth* 5.0
- e. HDMI: 2x micro-HDMI (maks 4K output)
- f. GPIO: 40 pin GPIO
- g. Daya: 5V 3A melalui USB-C
- h. *Storage*: microSD card slot (untuk OS dan penyimpanan)

2. **Raspberry Pi 3 Model B+**

Raspberry Pi 3 Model B+ adalah model terbaru dari keluarga Raspberry Pi 3 yang dirilis pada tahun 2018. Dibandingkan dengan model sebelumnya, Raspberry Pi 3 Model B+ menawarkan berbagai

peningkatan, termasuk prosesor *quad-core* 64-bit yang beroperasi pada kecepatan 1,4 GHz, memberikan performa yang lebih cepat dan responsif. Board ini juga dilengkapi dengan Wi-Fi *dual-band*, yang mendukung frekuensi 2.4 GHz dan 5 GHz, memastikan koneksi internet yang lebih stabil dan cepat. Selain itu, Raspberry Pi 3 Model B+ sudah mendukung *Bluetooth* 4.0, memungkinkan konektivitas nirkabel dengan perangkat *Bluetooth*. Koneksi Ethernet pada model ini juga ditingkatkan untuk kecepatan yang lebih tinggi, sementara kemampuan PoE (*Power over Ethernet*) memungkinkan board ini menerima daya langsung melalui kabel Ethernet, mengurangi kebutuhan akan adaptor daya terpisah. Semua fitur ini membuat Raspberry Pi 3 Model B+ menjadi pilihan yang lebih handal untuk berbagai aplikasi, baik di dunia pendidikan, hobi, maupun proyek industri.



Gambar 2. 2 Raspberry Pi 3 Model B

(Sumber : <https://www.thelab.id/>)

Berikut adalah spesifikasi Raspberry Pi 3 Model B+:

- a. Prosesor: *Quad-core* ARM Cortex-A53 1.4 GHz
- b. RAM: 1GB LPDDR2 SDRAM
- c. Port USB: 4x USB 2.0

- d. Konektivitas: Gigabit Ethernet (terbatas pada 300 Mbps), *Dual-band* Wi-Fi 802.11ac, *Bluetooth* 4.2
- e. HDMI: 1x HDMI (1080p *output*)
- f. GPIO: 40 pin GPIO
- g. Daya: 5V 2.5A melalui micro-USB
- h. Storage: microSD *card slot*

3. Raspberry Pi 2 Model B

Raspberry Pi 2 Model B adalah salah satu versi dari komputer papan tunggal (*single-board computer*) yang dikembangkan oleh Raspberry Pi Foundation. Raspberry Pi 2 Model B dirilis pada bulan Februari 2015, dan merupakan peningkatan besar dibandingkan dengan model sebelumnya, Raspberry Pi Model B+.



Gambar 2. 3 Raspberry Pi 2 Model B

(Sumber : <https://www.kubii.com/>)

Berikut adalah spesifikasi Raspberry Pi 2 Model B:

- a. Prosesor: *Quad-core* ARM Cortex-A7 900 MHz
- b. RAM: 1GB LPDDR2 SDRAM
- c. Port USB: 4x USB 2.0
- d. Konektivitas: 10/100 Ethernet
- e. HDMI: 1x HDMI (1080p *output*)
- f. GPIO: 40 pin GPIO
- g. Daya: 5V 2A melalui micro-USB
- h. *Storage*: microSD *card slot*

4. Raspberry Pi Zero / Zero W / Zero 2 W

Raspberry Pi Zero, Raspberry Pi Zero W, dan Raspberry Pi Zero 2 W adalah varian-varian dari Raspberry Pi yang dirancang dengan ukuran lebih kecil dan harga lebih terjangkau, namun tetap mempertahankan sebagian besar kemampuan dasar dari komputer papan tunggal (*single-board computer*) Raspberry Pi. Raspberry Pi Zero dirancang khusus untuk proyek-proyek yang membutuhkan perangkat kecil dan murah, seperti proyek IoT (*Internet of Things*), perangkat *embedded*, dan aplikasi di mana ukuran dan daya sangat penting.



Gambar 2. 4 Raspberry Pi Zero

(Sumber : <https://ecadio.com>)

Berikut adalah spesifikasi Raspberry Pi Zero / Zero W / Zero 2 W:

- a. Prosesor: Zero: ARM11 *single-core* 1 GHz, Zero W: ARM11 *single-core* 1 GHz + Wi-Fi/Bluetooth, Zero 2 W: *Quad-core* Cortex-A53 1 GHz (lebih cepat dari Zero dan Zero W)
- b. RAM: 512MB LPDDR2 SDRAM

- c. Port USB: 1x micro USB (untuk data), 1x micro USB (untuk daya)
- d. Konektivitas: Zero W dan Zero 2 W memiliki Wi-Fi 802.11n dan *Bluetooth* 4.2
- e. HDMI: 1x mini HDMI (1080p output)
- f. GPIO: 40 pin GPIO (lebih sedikit pada model Zero)
- g. Daya: 5V 1A (Zero dan Zero W)
- h. *Storage*: microSD card slot

5. **Raspberry Pi Pico (dan Pico W)**

Raspberry Pi Pico adalah sebuah *development* board kecil yang dirancang oleh Raspberry Pi, dengan ukuran yang sangat kompak namun memiliki kemampuan yang sangat mengesankan. *Board* ini ditenagai oleh prosesor *dual-core* Arm Cortex-M0+ yang memberikan performa efisien meskipun dengan ukuran yang mungil. Dilengkapi dengan 264KB RAM dan kapasitas penyimpanan *flash* hingga 2MB yang terpasang langsung pada *board*, Raspberry Pi Pico memiliki sumber daya yang cukup untuk menjalankan berbagai proyek dan aplikasi kecil. Raspberry Pi Pico juga dapat diprogram menggunakan bahasa pemrograman C atau MicroPython, yang memungkinkan para pengembang, hobiis, atau pelajar untuk membuat berbagai aplikasi dan eksperimen dengan cara yang mudah dan fleksibel. Kemampuannya untuk digunakan dalam berbagai proyek elektronik, seperti sensor, motor, dan perangkat lainnya, menjadikan Raspberry Pi Pico pilihan yang ideal untuk para pembuat (*maker*) atau mereka yang ingin mempelajari pemrograman dan pengembangan perangkat keras secara lebih praktis dan langsung.



Gambar 2. 5 Raspberry Pi Pico W

(Sumber : <https://sg.element14.com/>)

Berikut adalah spesifikasi Raspberry Pi Pico (dan Pico W):

- a. Prosesor: *Dual-core* ARM Cortex-M0+ 133 MHz (RP2040)
- b. RAM: 264KB SRAM
- c. GPIO: 26 pin GPIO
- d. Konektivitas: Pico W: Wi-Fi 802.11b/g/n (chip CYW43439)
- e. Daya: 1.8V to 3.6V, biasanya diberi daya melalui micro-USB
- f. *Storage: 2MB Flash onboard* (Pico)

2.2.5 Internet of Things (IoT)

Internet of Things (IoT) adalah sebuah konsep teknologi yang bertujuan untuk memperluas dan mengoptimalkan manfaat konektivitas internet yang terus-menerus, memungkinkan berbagai perangkat dan benda fisik untuk terhubung satu sama lain melalui jaringan internet. Dengan adanya IoT, mesin, peralatan, dan objek lainnya dilengkapi dengan sensor dan aktuator yang dapat mengumpulkan data, serta memantau dan mengelola kinerjanya

secara mandiri. Hal ini memungkinkan perangkat-perangkat tersebut untuk berinteraksi dan bahkan mengambil keputusan secara independen berdasarkan informasi baru yang mereka terima, tanpa intervensi manusia.

Pada dasarnya, IoT adalah gagasan tentang integrasi semua benda fisik di dunia nyata, yang dapat saling berkomunikasi dan bekerja bersama sebagai bagian dari sebuah sistem yang terhubung. Jaringan internet berfungsi sebagai penghubung utama antar perangkat tersebut. Misalnya, sistem CCTV yang terpasang di berbagai lokasi dapat saling terhubung melalui internet dan dikendalikan dari sebuah ruang kontrol yang mungkin berlokasi sangat jauh dari titik pengawasan, bahkan puluhan kilometer jaraknya. Contoh lainnya adalah rumah pintar, di mana berbagai perangkat rumah tangga dapat diatur dan dimonitor menggunakan aplikasi di *smartphone*, memanfaatkan koneksi internet sebagai media komunikasinya. Secara teknis, perangkat-perangkat IoT umumnya terdiri dari tiga komponen utama: sensor sebagai pengumpul data, sambungan internet sebagai media komunikasi, dan server yang berfungsi untuk menerima, menyimpan, serta menganalisis data yang dikirimkan oleh sensor.

Konsep ini pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999 dalam sebuah presentasi yang ia lakukan. Sejak saat itu, perkembangan IoT semakin pesat, dan kini banyak perusahaan besar yang terlibat dalam penelitian dan pengembangan IoT, seperti *Intel*, *Microsoft*, dan *Oracle*, yang melihat potensi besar dari teknologi ini. Banyak yang memprediksi bahwa IoT akan menjadi “*the next big thing*” dalam dunia teknologi informasi, karena kemampuannya untuk menghadirkan berbagai solusi inovatif di berbagai sektor kehidupan. Sebagai contoh, aplikasi IoT yang sederhana namun sangat bermanfaat adalah kulkas pintar yang dapat memberi tahu pemiliknya melalui SMS atau email mengenai persediaan makanan dan minuman yang sudah habis, sehingga memudahkan pemilik untuk mengetahui apa yang perlu dibeli kembali tanpa harus membuka kulkas.

Dengan demikian, *Internet of Things* tidak hanya berfungsi sebagai alat untuk menghubungkan perangkat, tetapi juga menciptakan ekosistem yang

lebih cerdas dan efisien, di mana berbagai perangkat dapat saling berinteraksi untuk mendukung kenyamanan, efisiensi, dan produktivitas kehidupan manusia. Teknologi ini terus berkembang dan diperkirakan akan semakin mendalam dampaknya di masa depan.

2.2.6 Face Recognition

Face recognition adalah sebuah teknologi canggih yang memungkinkan perangkat untuk mengenali wajah seseorang melalui citra digital yang diambil oleh kamera. Teknologi ini merupakan aplikasi biometrik yang menggunakan algoritma khusus untuk mengidentifikasi atau memverifikasi identitas seseorang berdasarkan fitur wajah mereka. Sistem biometrik ini bekerja dengan cara membandingkan dan menganalisis pola bentuk wajah yang dimiliki individu. Meskipun awalnya *face recognition* banyak digunakan dalam sistem keamanan, kini aplikasi teknologi ini sudah meluas ke berbagai bidang lainnya, seperti pemasaran, pengawasan, dan layanan publik.

Seiring dengan perkembangan teknologi, teknik *face recognition* kini telah mengalami kemajuan yang signifikan, termasuk kemampuan untuk mengidentifikasi wajah dalam bentuk tiga dimensi (3-D). Inovasi ini membantu mengatasi sejumlah masalah yang dihadapi oleh sistem *face recognition* sebelumnya, seperti ketidakakuratan saat identifikasi dilakukan dalam kondisi tertentu. Salah satu kelebihan utama dari *face recognition*, dibandingkan dengan teknik biometrik lainnya, adalah kemampuannya untuk beroperasi tanpa perlu kontak langsung dengan objek yang dikenali. Teknologi ini memungkinkan identifikasi wajah dilakukan dari jarak jauh, yang juga menjadikannya lebih sulit untuk dimanipulasi atau disalahgunakan. Fitur ini sangat mendukung penggunaan *face recognition* dalam sistem absensi atau pengelolaan waktu kerja, di mana kecepatan dan akurasi sangat diperlukan.

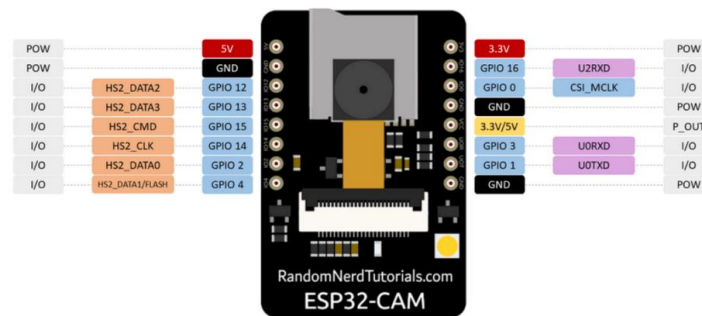
Namun, meskipun teknologi ini sangat bermanfaat, ada beberapa kelemahan yang perlu diperhatikan. Salah satunya adalah ketergantungan

pada kondisi pencahayaan yang memadai. Jika pencahayaan tidak cukup, sistem *face recognition* mungkin kesulitan dalam mengenali wajah dengan akurat, yang dapat menyebabkan kegagalan dalam proses identifikasi. Meskipun demikian, penelitian dan inovasi terus dilakukan untuk meningkatkan keandalan dan efektivitas teknologi ini di masa depan.

2.2.7 ESP32 Cam

ESP32 CAM adalah sebuah papan pengembangan yang menggunakan *chip* ESP32 dan dilengkapi dengan modul kamera, menjadikannya pilihan yang populer untuk berbagai aplikasi berbasis pengolahan gambar dan video. Papan ini menawarkan sejumlah fitur canggih, termasuk kemampuan konektivitas *Bluetooth* dan *WiFi*, serta dilengkapi dengan kamera dan *slot* kartu *microSD* untuk penyimpanan eksternal. Dibandingkan dengan produk ESP32 lainnya, ESP32 CAM memiliki jumlah pin I/O yang lebih sedikit, hanya menyediakan 10 pin GPIO yang dapat diakses secara langsung oleh pengguna. Pin-pin lainnya digunakan secara internal untuk mendukung fungsi kamera dan *slot* kartu *microSD*, yang merupakan bagian penting dari desain modularnya.

Modul ESP32 CAM tidak dilengkapi dengan port *microUSB*, yang berarti untuk menghubungkannya dengan komputer, dibutuhkan adaptor FTDI eksternal. Adaptor FTDI ini berfungsi sebagai antarmuka yang memungkinkan komunikasi antara ESP32 CAM dan komputer, memberikan akses untuk pemrograman dan transfer data. Salah satu fitur utama yang membedakan ESP32 CAM dari modul ESP32 lainnya adalah sensor kamera yang digunakan. Modul ini mengandalkan sensor kamera OV2640, yang dikenal karena kemampuannya dalam menangkap gambar dengan resolusi yang cukup tinggi. Sensor OV2640 ini tidak hanya dapat digunakan untuk aplikasi pengambilan gambar biasa, tetapi juga mendukung aplikasi yang lebih canggih, seperti deteksi objek dan pengenalan wajah, menjadikannya sangat berguna untuk proyek-proyek yang membutuhkan analisis visual atau interaksi berbasis penglihatan.



Gambar 2. 6 ESP32-Cam

(Sumber : <https://blog.indobot.co.id/>)

Spesifikasi ESP32 CAM antara lain:

- 802.11b/g/n Wi-Fi
- Bluetooth 4.2 with BLE
- Kamera 2MP
- UART, SPI, I2C and PWM interfaces
- Clock speed up to 160 MHz
- Computing power up to 600 DMIPS
- In-Built Micro SD Card slot
- Image transfer rate 15-60 frame per second
- 520 KB SRAM dan 4 MB PSRAM
- Upload gambar WiFi
- Built-in Flash LED

2.2.8 Open CV

OpenCV, atau yang dikenal dengan nama *Open Source Computer Vision Library*, adalah sebuah pustaka perangkat lunak sumber terbuka yang awalnya dikembangkan oleh perusahaan teknologi Intel. Tujuan utama dari pengembangan *OpenCV* adalah untuk menyederhanakan dan mempermudah proses pemrograman dalam bidang visi komputer, khususnya yang berkaitan dengan pengolahan citra digital. *OpenCV* memiliki beragam fitur canggih

yang mendukung berbagai aplikasi, seperti pengenalan wajah, pelacakan objek atau wajah, serta deteksi wajah secara *real-time*. Selain itu, pustaka ini juga dilengkapi dengan alat dan teknik canggih lainnya, termasuk *Kalman filtering* yang digunakan untuk memprediksi dan menghaluskan pergerakan objek, serta berbagai metode yang berbasis pada kecerdasan buatan (*Artificial Intelligence/AI*). Tak hanya itu, *OpenCV* juga menyediakan sejumlah algoritma yang sederhana dan mudah digunakan, yang memungkinkan pengembang untuk bekerja dengan API tingkat rendah dalam pengembangan aplikasi visi komputer. Dengan kemampuan tersebut, *OpenCV* menjadi salah satu pustaka paling populer di kalangan pengembang yang bekerja dengan citra digital dan visi komputer.

OpenCV adalah sebuah pustaka *open source* yang dirancang untuk pengolahan visi komputer, yang awalnya dikembangkan untuk bahasa pemrograman C/C++. Namun, seiring berjalannya waktu, pustaka ini telah diperluas dan kini mendukung berbagai bahasa pemrograman lainnya, seperti *Python*, *Java*, dan *Matlab*, memungkinkan lebih banyak pengembang dari berbagai disiplin ilmu untuk menggunakannya. *Library* ini dirancang untuk mempermudah implementasi algoritma visi komputer dan pengolahan citra digital dalam aplikasi-aplikasi praktis. *OpenCV* dapat diunduh secara gratis melalui berbagai situs, seperti <http://sourceforge.net/projects/opencvlibrary> atau <http://opencv.org/releases.html>. Pada saat ini, versi terbaru dari *OpenCV* adalah 3.3.0, yang dirilis pada tanggal 3 Agustus 2017. Versi terbaru ini membawa berbagai perbaikan dan fitur tambahan yang semakin memperkaya kemampuan pustaka ini dalam menangani tugas-tugas pengolahan citra dan visi komputer.

Intel pertama kali meluncurkan versi pertama dari *OpenCV* pada tahun 1999, dengan tujuan untuk menyediakan pustaka sumber terbuka yang memudahkan pengembangan aplikasi yang berkaitan dengan visi komputer dan pengolahan citra digital. Pada awalnya, *OpenCV* memerlukan ketergantungan pada pustaka lain, yaitu *Intel Image Processing Library*, untuk menjalankan fungsinya. Namun, seiring dengan perkembangan dan

pembaruan yang terus dilakukan, ketergantungan tersebut akhirnya dihapuskan, sehingga *OpenCV* dapat berdiri sendiri sebagai pustaka mandiri (*standalone library*) yang lebih fleksibel dan dapat digunakan secara luas. *OpenCV* kini mendukung berbagai platform, memungkinkan pengembang untuk menggunakan pustaka ini baik di sistem operasi *Windows* maupun *Linux*. Selain itu, seiring waktu, dukungan terhadap sistem operasi lainnya juga diperkenalkan, termasuk *MacOSX* dan *Android*, yang semakin memperluas jangkauan pengguna *OpenCV* di berbagai perangkat dan sistem operasi. Dengan kemampuan multiplatform ini, *OpenCV* semakin populer di kalangan pengembang yang bekerja dengan pengolahan citra dan visi komputer di berbagai jenis perangkat dan sistem operasi.

OpenCV menawarkan berbagai fitur yang sangat berguna bagi pengembang yang bekerja dengan pengolahan citra dan visi komputer. Berikut adalah beberapa fitur utama yang dapat dimanfaatkan dalam pustaka *OpenCV*:

1. Image dan Video I/O

Fitur ini memungkinkan pengguna untuk melakukan *input* dan *output* data gambar maupun video dengan mudah. Dengan antarmuka ini, kita dapat membaca gambar atau video dari berbagai sumber, baik itu dari *file* yang disimpan di perangkat ataupun dari umpan video secara langsung. Selain itu, *OpenCV* juga memungkinkan pengguna untuk membuat dan menyimpan *file* gambar atau video ke dalam berbagai format yang umum digunakan, menjadikannya alat yang sangat fleksibel untuk pengolahan data visual.

2. Computer Vision secara umum dan Pengolahan Citra Digital (untuk low dan mid-level API)

OpenCV menyediakan berbagai fungsi dan algoritma untuk pengolahan citra digital serta penerapan visi komputer secara umum. Pustaka ini mendukung berbagai teknik pengolahan citra seperti pemrosesan citra berbasis fitur, deteksi objek, dan pengenalan pola. *OpenCV* juga menyediakan API untuk pengolahan citra pada tingkat rendah (*low-level*) dan tingkat menengah (*mid-level*), yang

memungkinkan pengembang untuk lebih fleksibel dalam menyesuaikan aplikasi sesuai dengan kebutuhan spesifik mereka, baik itu untuk analisis citra dasar atau aplikasi yang lebih kompleks dan berbasis kecerdasan buatan.

Dengan antarmuka yang disediakan oleh *OpenCV*, pengguna dapat melakukan berbagai eksperimen dan uji coba menggunakan berbagai algoritma standar dalam visi komputer. *OpenCV* memungkinkan pengembang untuk mengimplementasikan dan menguji berbagai teknik pengolahan citra, yang mencakup:

a. *Modul computer vision high level*

Selain fitur-fitur dasar dan menengah, *OpenCV* juga menawarkan kemampuan "*high-level*" yang memungkinkan pengembang untuk melakukan tugas-tugas visi komputer yang lebih kompleks. Salah satu kemampuan utama adalah deteksi wajah, di mana *OpenCV* menyediakan algoritma canggih seperti *Haar Cascade Classifier* yang dapat mendeteksi wajah manusia dalam citra atau video secara otomatis dan efisien. Selain itu, *OpenCV* juga mendukung pengenalan wajah, yang memungkinkan aplikasi untuk mengenali identitas individu berdasarkan gambar wajah mereka. Teknik seperti *Eigenfaces*, *Fisherfaces*, dan LBPH (*Local Binary Patterns Histograms*) digunakan untuk pengenalan wajah, yang sangat berguna dalam sistem keamanan, verifikasi identitas, dan pelacakan individu. Kemampuan lainnya adalah *optical flow*, yang memungkinkan pelacakan pergerakan objek dalam citra atau video dengan menganalisis perubahan posisi piksel dari satu frame ke frame berikutnya. Dengan menggunakan algoritma *Lucas-Kanade Optical Flow*, *OpenCV* memungkinkan prediksi pergerakan objek dengan akurasi tinggi, yang banyak digunakan dalam aplikasi pelacakan objek, estimasi gerakan, dan analisis video. Fitur-fitur "*high-level*" ini menjadikan *OpenCV* lebih dari sekadar alat untuk

pengolahan citra dasar, tetapi juga memungkinkan pengembangan aplikasi canggih untuk deteksi dan pengenalan wajah serta analisis pergerakan dalam video secara *real-time*.

b. Metode untuk AI dan *machine learning*

Aplikasi visi komputer sering kali memerlukan teknik *machine learning* atau metode kecerdasan buatan (AI) lainnya untuk meningkatkan kemampuan dalam menganalisis dan memahami data visual. Beberapa metode *machine learning* ini dapat diakses melalui paket *OpenCV Machine Learning*, yang menyediakan berbagai algoritma untuk pembelajaran mesin dan pemrosesan data. Paket ini mencakup teknik-teknik seperti *k-Nearest Neighbors (k-NN)*, *Support Vector Machines (SVM)*, *Decision Trees*, dan *Random Forests*, yang digunakan untuk klasifikasi dan regresi data, serta pengenalan pola. Selain itu, *OpenCV* juga mendukung algoritma *Clustering*, *AdaBoost*, dan *Boosting*, yang berguna untuk meningkatkan akurasi model prediksi. Dengan menyediakan berbagai metode ini, *OpenCV* memungkinkan pengembang untuk mengintegrasikan teknik-teknik AI dan *machine learning* dalam aplikasi visi komputer mereka, seperti deteksi objek, pengenalan wajah, pelacakan, dan banyak lagi, yang menjadikan pustaka ini sangat fleksibel dan kuat untuk berbagai kebutuhan pemrosesan citra dan video.

c. Sampling gambar dan transformasi

Dalam *OpenCV*, terdapat berbagai antarmuka yang memungkinkan pengembang untuk melakukan operasi dasar pada citra dengan mudah, termasuk teknik seperti substraksi subregion dari gambar, *random sampling*, rotasi, dan lainnya. Substraksi subregion memungkinkan pengguna untuk mengekstrak bagian tertentu dari citra, yang sangat berguna ketika hanya sebagian gambar yang ingin dianalisis, seperti dalam

deteksi objek atau pemrosesan area spesifik. *OpenCV* juga menyediakan kemampuan untuk melakukan *random sampling* pada citra, yaitu mengambil sampel acak dari piksel dalam gambar. Teknik ini sering digunakan dalam eksperimen atau pengolahan citra berbasis statistik dan juga berguna dalam pelatihan model *machine learning* dengan subset data. Selain itu, *OpenCV* memungkinkan rotasi gambar, yang dapat digunakan untuk memutar citra dalam berbagai sudut, berguna untuk memperbaiki orientasi gambar atau meningkatkan keragaman data pelatihan dalam model AI. Dengan kemampuan-kemampuan ini, *OpenCV* menjadi pustaka yang sangat fleksibel, memudahkan pengembang dalam melakukan manipulasi citra dan menerapkan berbagai algoritma pengolahan gambar dengan efisien.

2.2.9 Haar Cascade Classifier

Haar Cascade merupakan sebuah algoritma dalam bidang pembelajaran mesin (*machine learning*) yang secara khusus dirancang untuk melakukan proses deteksi objek dalam sebuah gambar atau citra digital. Algoritma ini pertama kali diperkenalkan oleh dua peneliti bernama Paul Viola dan Michael Jones pada tahun 2001 melalui sebuah makalah yang menjadi tonggak penting dalam bidang visi komputer. Pendekatan ini tergolong sebagai teknik berbasis pembelajaran mesin karena proses deteksinya melibatkan pelatihan model menggunakan sejumlah besar data.

Dalam proses pelatihannya, *Haar Cascade* menggunakan dua jenis data utama, yaitu citra positif dan citra negatif. Citra positif adalah gambar-gambar yang mengandung objek tertentu yang ingin dideteksi—misalnya wajah, mata, atau bagian tubuh lainnya. Sebaliknya, citra negatif adalah gambar-gambar yang tidak mengandung objek target tersebut. Dengan mempelajari perbedaan antara kedua jenis citra tersebut, algoritma ini dapat mengenali dan membedakan objek pada gambar lain yang belum pernah dilihat sebelumnya.

Keunikan dari *Haar Cascade* terletak pada penggunaan struktur yang disebut *cascade function* yakni serangkaian klasifikator yang disusun secara bertahap atau berlapis, dimana setiap lapisan bertugas untuk menyaring bagian gambar yang tidak relevan. Pendekatan ini membuat proses deteksi menjadi lebih cepat dan efisien karena hanya bagian gambar yang lolos dari tahap awal yang akan dianalisis lebih lanjut.

Saat ini, algoritma *Haar Cascade* telah tersedia secara praktis dalam bentuk pustaka (*library*) yang disediakan oleh OpenCV, salah satu pustaka pemrosesan citra dan visi komputer paling populer. Pustaka ini menyertakan berbagai model *Haar Cascade* yang telah dilatih sebelumnya (*pre-trained models*), sehingga pengguna tidak perlu melatih model dari awal. Model-model tersebut telah dikelompokkan dalam berbagai kategori berdasarkan objek yang dideteksi, seperti deteksi wajah manusia, mata, mulut, kendaraan, dan lain sebagainya.

Secara teknis, *Haar Cascade* melakukan ekstraksi fitur dari sebuah citra dengan cara menerapkan filter sederhana yang menyerupai kernel pada teknik konvolusi dalam jaringan saraf tiruan. Filter ini mampu mengenali pola-pola tertentu dalam gambar yang merepresentasikan keberadaan objek yang dicari. Dengan demikian, *Haar Cascade* menjadi salah satu metode yang cukup andal dan cepat untuk keperluan deteksi objek dalam berbagai aplikasi komputer vision.

2.2.10 Arduino IDE

Arduino IDE dikembangkan berdasarkan perangkat lunak *Processing* yang kemudian dimodifikasi menjadi khusus untuk pemrograman dengan *Arduino*. IDE (*Integrated Development Environment*) ini adalah lingkungan terintegrasi yang digunakan untuk mengembangkan dan menulis kode program untuk mikrokontroler *Arduino*. Sebagai sebuah lingkungan, IDE ini memfasilitasi seluruh proses pemrograman, mulai dari menulis kode hingga mengunggahnya ke *board* *Arduino*, yang memungkinkan board untuk

menjalankan fungsi-fungsi yang diinginkan melalui sintaks pemrograman yang ditulis oleh pengguna.



Gambar 2. 7 Logo Arduino

(Sumber : <https://blog.indobot.co.id/>)

Arduino menggunakan bahasa pemrograman yang mirip dengan bahasa C, yang disebut sebagai *sketch*. Namun, bahasa pemrograman ini telah dimodifikasi agar lebih ramah untuk pemula, sehingga mereka dapat lebih mudah memahami dan menulis kode untuk Arduino, tanpa memerlukan pengetahuan yang mendalam tentang bahasa C yang lebih kompleks. Selain itu, sebelum *board* Arduino dipasarkan, IC mikrokontroler di dalamnya sudah dilengkapi dengan sebuah program *Bootloader*. Program ini berfungsi sebagai penghubung antara *compiler* Arduino dan mikrokontroler, sehingga memungkinkan proses pemrograman dan pengunggahan kode ke board Arduino dengan lebih mudah.

Arduino IDE dibangun menggunakan bahasa pemrograman Java, yang memungkinkan IDE untuk berjalan di berbagai platform, seperti Windows, macOS, dan Linux. Selain itu, Arduino IDE dilengkapi dengan *library C/C++* yang dikenal sebagai *Wiring*, yang sangat memudahkan operasi *input* dan *output*. *Library Wiring* ini menyederhanakan pemrograman *hardware*, memungkinkan pengembang untuk lebih mudah berinteraksi

dengan perangkat keras seperti sensor, motor, dan komponen lainnya. Dengan adanya berbagai fitur ini, Arduino IDE menjadi alat yang sangat berguna, tidak hanya untuk para pemula, tetapi juga bagi para profesional dalam mengembangkan berbagai proyek berbasis mikrokontroler.

Program yang ditulis menggunakan Arduino IDE disebut sebagai *sketch*. *Sketch* ini ditulis dalam editor teks yang terdapat di dalam Arduino IDE dan disimpan dalam format file dengan ekstensi *.ino*. Editor teks pada Arduino IDE dilengkapi dengan berbagai fitur yang memudahkan pengguna dalam menulis kode program, seperti *cutting/pasting* dan *searching/replacing*, yang memungkinkan pengeditan kode menjadi lebih efisien dan cepat.

Pada antarmuka Arduino IDE, terdapat *message box* berwarna hitam yang berfungsi untuk menampilkan berbagai status penting terkait proses pemrograman. Di dalam *message box* ini, pengguna dapat melihat pesan-pesan seperti *error*, status *compile* (kompilasi), dan status *upload* (pengunggahan program ke *board* Arduino). Ini memberikan *feedback* langsung kepada pengguna mengenai status dari program yang sedang dikompilasi atau diunggah, sehingga memudahkan dalam proses debugging dan pemrograman.

Di bagian bawah paling kanan dari *Arduino IDE*, terdapat informasi mengenai board yang terkonfigurasi dan COM *port* yang digunakan. Ini memberi tahu pengguna *board* Arduino mana yang sedang aktif dan *port* komunikasi yang digunakan untuk menghubungkan *board* dengan komputer. Fitur ini sangat penting untuk memastikan bahwa proses pengunggahan program dapat berjalan lancar, karena pengaturan yang salah pada *board* atau *port* bisa menyebabkan kesalahan saat mengunggah program ke *board* Arduino.

2.2.11 Thonny IDE

Thonny IDE untuk Python adalah sebuah aplikasi yang dirancang khusus untuk membantu pemula belajar dan mengajar pemrograman dengan

lebih mudah. Thonny menawarkan visualisasi yang jelas tentang bagaimana program yang sedang dibuat bekerja, sehingga bagian-bagian dari program tersebut menjadi lebih mudah dipahami dalam alur kerja yang diinginkan. Thonny Python IDE dilengkapi dengan Python 3 bawaan, memungkinkan pengguna untuk langsung memulai pemrograman tanpa perlu mengatur lingkungan Python secara terpisah. Antarmuka pengguna pada Thonny dibuat sederhana dan ramah pemula, sehingga sangat memudahkan pengguna yang baru mulai belajar pemrograman.

Meskipun ditujukan untuk pemula, Thonny juga dilengkapi dengan beberapa fitur canggih yang menjadikannya IDE yang sangat berguna untuk pengembangan Python yang lebih kompleks. Fitur-fitur tersebut termasuk penyorotan kesalahan sintaks, yang membantu pengguna untuk dengan cepat mengidentifikasi dan memperbaiki kesalahan dalam kode mereka, serta *debugger* yang memungkinkan pengguna untuk memeriksa dan memecahkan masalah pada kode secara lebih efektif. Selain itu, Thonny menyediakan penyelesaian kode otomatis yang membantu mempercepat penulisan kode dengan menyarankan opsi yang relevan, serta evaluasi langkah demi langkah, yang memungkinkan pengguna untuk melihat bagaimana kode mereka dieksekusi baris demi baris. Semua fitur ini menjadikan Thonny IDE tidak hanya cocok untuk pemula, tetapi juga alat yang baik untuk pengembangan aplikasi Python secara keseluruhan.

2.2.12 HTTP Server/ Web Server

Web server merupakan sebuah perangkat lunak yang berperan untuk menyediakan layanan data di dunia maya. Fungsi utama dari *web server* adalah untuk menerima permintaan yang dikirimkan oleh klien melalui protokol HTTP (*HyperText Transfer Protocol*) atau HTTPS (*HyperText Transfer Protocol Secure*) yang biasanya dilakukan melalui peramban web (*web browser*). Setelah menerima permintaan tersebut, *web server* kemudian mengolah dan mengirimkan kembali hasilnya dalam bentuk halaman web, yang umumnya disusun dalam format dokumen HTML (*HyperText Markup*

Language). *Web server* ini juga memiliki peran penting sebagai platform atau tempat bagi aplikasi web untuk dijalankan, sekaligus berfungsi sebagai penerima permintaan (*request*) dari klien yang menggunakan aplikasi atau layanan berbasis web. Sebagai alat yang menghubungkan antara pengguna dengan sumber daya yang ada di internet, *web server* memainkan peranan yang sangat penting dalam pengelolaan dan penyajian informasi berbasis web (Indra Warman & Zahni, 2013).

2.2.13 Buzzer

Menurut Rahadhian Angga Pratama (2012), buzzer merupakan sebuah komponen elektronik yang berfungsi untuk mengubah energi listrik menjadi energi suara dalam bentuk getaran. Prinsip kerja buzzer pada dasarnya serupa dengan cara kerja *loudspeaker*. Buzzer terdiri dari sebuah kumparan yang dipasang pada sebuah diafragma. Ketika kumparan tersebut dialiri arus listrik, kumparan tersebut akan berubah menjadi elektromagnet. Hal ini menyebabkan kumparan tersebut tertarik atau terdorong, tergantung pada arah arus listrik yang mengalir dan polaritas magnet yang ada. Karena kumparan dipasang pada diafragma, gerakan kumparan ini akan mempengaruhi diafragma untuk bergerak bolak-balik. Gerakan diafragma yang terus-menerus ini akan menyebabkan udara di sekitarnya bergetar, dan getaran udara tersebut akhirnya menghasilkan suara yang dapat didengar. Buzzer sering digunakan sebagai indikator atau pemberi tanda, misalnya untuk memberi tahu pengguna bahwa suatu proses telah selesai atau sebagai alarm untuk menandakan adanya kesalahan atau masalah pada suatu perangkat atau sistem.

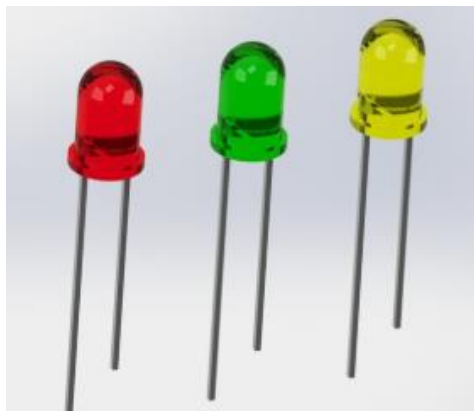


Gambar 2. 8 Buzzer

(Sumber : <https://www.edukasiaelektronika.com>)

2.2.14 LED

LED (*Light Emitting Diode*) adalah komponen elektronik yang memancarkan cahaya saat dialiri arus listrik. Seperti dioda lainnya, LED hanya memungkinkan arus mengalir satu arah, dengan anode dihubungkan ke tegangan positif dan katode ke tegangan negatif. LED merupakan lampu *solid-state*, yang tidak menggunakan bagian bergerak atau gas, dan termasuk dalam kategori optoelektronik, yang menggabungkan prinsip elektronik dan optik. Fungsi utama LED meliputi lampu indikator, transmisi sinyal cahaya, dan penghubung rangkaian elektronik dalam komunikasi optik. LED mengubah energi listrik menjadi cahaya, dengan warna yang bergantung pada bahan semikonduktor yang digunakan. Beberapa jenis LED, seperti LED inframerah, digunakan dalam aplikasi seperti *remote control* atau sensor.



Gambar 2. 9 LED

(Sumber : <https://muji.blog.unimma.ac.id/>)

2.2.15 Kabel Jumper

Kabel jumper adalah jenis kabel listrik yang dilengkapi dengan pin konektor pada kedua ujungnya, yang memungkinkan pengguna untuk menghubungkan dua komponen elektronik, seperti yang digunakan dalam proyek-proyek mikrokontroller, tanpa memerlukan proses penyolderan. Fungsi utama dari kabel jumper adalah sebagai penghantar atau konduktor listrik yang menghubungkan satu bagian rangkaian listrik ke bagian lainnya, memudahkan pembuatan sambungan sementara antar komponen.



Gambar 2. 10 Kabel Jumper

(Sumber : <https://www.arduinoindonesia.id>)

Kabel jumper umumnya digunakan dalam perakitan rangkaian pada *breadboard* atau alat *prototyping* lainnya, di mana fleksibilitas dan kemudahan dalam mengatur ulang sambungan rangkaian sangat dibutuhkan. Keunggulan penggunaan kabel jumper adalah kemudahannya dalam memasang dan melepas sambungan tanpa merusak komponen atau memerlukan keterampilan khusus seperti soldering. Pada ujung kabel jumper, terdapat dua jenis konektor, yaitu konektor jantan (*male connector*) dan konektor betina (*female connector*), yang masing-masing memiliki fungsi berbeda. Konektor jantan umumnya memiliki pin yang lebih kecil dan dapat dimasukkan ke dalam soket, sementara konektor betina memiliki lubang

untuk menerima pin konektor jantan. Dengan adanya berbagai jenis kabel jumper yang tersedia, pengguna dapat dengan mudah memilih kabel yang sesuai dengan kebutuhan dalam merakit dan menghubungkan komponen-komponen elektronik secara efisien dan praktis.