

BAB II

TINJAUAN PUSTAKA

Pada bab II ini akan membahas tentang teori – teori penunjang yang digunakan dalam perancangan sistem kendali *Traffict Light* secara *realtime* berbasis IoT, dimana didalamnya memuat penjelasan mengenai apa itu teknologi berbasis IoT, traffic light on smartcity, teori deteksi gambar, teori pengolahan gambar oleh OpenCV, komponen yang digunakan serta perangkat lunak yang digunakan seperti php, python, dan Apache.

2.1 IoT (Internet of Things)

Internet of Thing (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan Internet. Penelitian pada IoT masih dalam tahap perkembangan. Oleh karena itu, tidak ada definisi dari Internet of Things. Berikut adalah beberapa definisi alternatif dikemukakan untuk memahami Internet of Things (IoT), antara lain :

- Casagras (Coordination and support action for global RFID-related activities and standardisation)

Mendefinisikan Internet of Things, sebagai sebuah infrastruktur jaringan global, yang menghubungkan benda-benda fisik dan virtual melalui eksploitasi data capture dan kemampuan komunikasi. Infrastruktur terdiri dari jaringan yang telah ada dan internet berikut pengembangan jaringannya. Semua ini akan menawarkan identifikasi objek, sensor dan kemampuan koneksi sebagai dasar untuk pengembangan layanan dan aplikasi ko-operatif yang independen. Ia juga ditandai dengan tingkat otonom data capture yang tinggi, event transfer, konektivitas jaringan dan interoperabilitas.

- SAP (Systeme, Anwendungen und Produkte)

Mendefinisikannya bahwa Dunia di mana benda-benda fisik diintegrasikan ke dalam jaringan informasi secara berkesinambungan, dan di mana benda-benda fisik tersebut berperan aktif dalam proses bisnis. Layanan yang tersedia berinteraksi dengan ‘objek pintar’ melalui Internet, mencari dan mengubah status mereka sesuai dengan setiap informasi yang dikaitkan, disamping memperhatikan masalah privasi dan keamanan.

- **CORDIS**

Rencana aksi untuk Uni Eropa untuk memperkenalkan pemerintahan berdasarkan Internet of Things.

- **ETP EPOSS**

Jaringan yang dibentuk oleh hal-hal atau benda yang memiliki identitas, pada dunia maya yang beroperasi di ruang itu dengan menggunakan kecerdasan antarmuka untuk terhubung dan berkomunikasi dengan pengguna, konteks sosial dan lingkungan.

2.2 Smart City

Kota cerdas merupakan sebuah visi pengembangan perkotaan untuk mengintegrasikan teknologi informasi dan komunikasi (TIK) dan teknologi *Internet of things* (IoT) dengan cara yang aman untuk mengelola aset kota. Aset ini meliputi sistem informasi instansi pemerintahan lokal, sekolah, perpustakaan, sistem transportasi, rumah sakit, pembangkit listrik, jaringan penyediaan air, pengelolaan limbah, penegakan hukum, dan pelayanan masyarakat lainnya. *Smart city* ditujukan dalam hal penggunaan informatika dan teknologi perkotaan untuk meningkatkan efisiensi pelayanan. TIK memungkinkan para pejabat kota berinteraksi langsung dengan masyarakat dan infrastruktur kota serta memantau apa yang terjadi di kota, bagaimana kota berkembang, dan bagaimana menciptakan kualitas hidup yang lebih baik. Melalui penggunaan sensor yang terintegrasi dengan *real-time monitoring system*, data yang dikumpulkan dari warga dan perangkat – kemudian diolah dan dianalisis. Informasi dan pengetahuan yang dikumpulkan adalah kunci untuk mengatasi inefisiensi.

2.3 Lampu Lalu Lintas

Simpang bersinyal merupakan bagian dari sistem kendali waktu tetap yang dirangkai atau sinyal aktuasi kendaraan terisolir, biasanya memerlukan metode dan perangkat lunak khusus dalam analisisnya. Pada umumnya sinyal lalu-lintas dipergunakan untuk satu atau lebih dari alasan berikut :

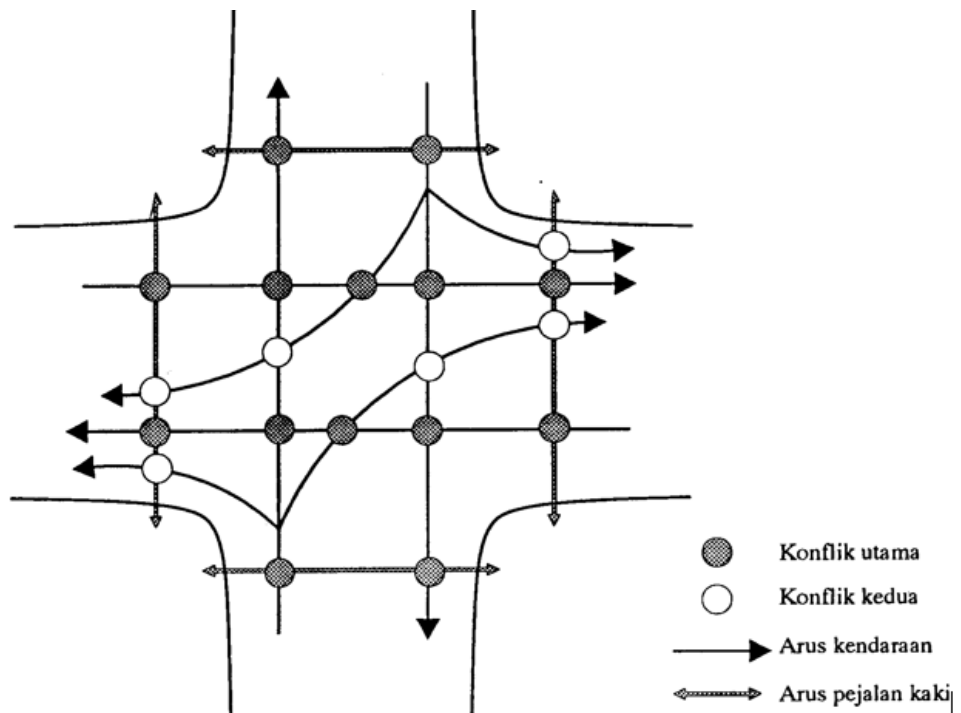
- Untuk menghindari kemacetan simpang akibat adanya konflik arus lalu-lintas, sehingga terjamin bahwa suatu kapasitas tertentu dapat dipertahankan, bahkan selama kondisi lalu-lintas jam puncak.

- Untuk memberi kesempatan kepada kendaraan dan/atau pejalan kaki dari jalan simpang (kecil) untuk/memotong jalan utama.
- Untuk mengurangi jumlah kecelakaan lalu-lintas akibat tabrakan antara kendaraan-kendaraan dari arah yang bertentangan.

2.3.1 Karakteristik Sinyal Lalu Lintas

Untuk sebagian besar fasilitas jalan, kapasitas dan perilaku lalu-lintas terutama adalah fungsi dari keadaan geometrik dan tuntutan lalu-lintas. Dengan menggunakan sinyal, kita dapat mendistribusikan kapasitas kepada berbagai pendekat melalui pengalokasian waktu hijau pada masing-masing pendekat. Maka dari itu untuk menghitung kapasitas dan perilaku lalu-lintas, pertama-tama perlu ditentukan fase dan waktu sinyal yang paling sesuai untuk kondisi yang ditinjau.

Penggunaan sinyal dengan lampu tiga-warna (hijau, kuning, merah) diterapkan untuk memisahkan lintasan dari gerakan-gerakan lalu-lintas yang saling bertentangan dalam dimensi waktu. Hal ini adalah keperluan yang mutlak bagi gerakan-gerakan lalu-lintas yang datang dari jalan-jalan yang saling berpotongan = konflik-konflik utama. Sinyal-sinyal dapat juga digunakan untuk memisahkan gerakan membelok dari lalu-lintas lurus melawan, atau untuk memisahkan gerakan lalu-lintas membelok dari pejalan-kaki yang menyeberang = konflik-konflik kedua.

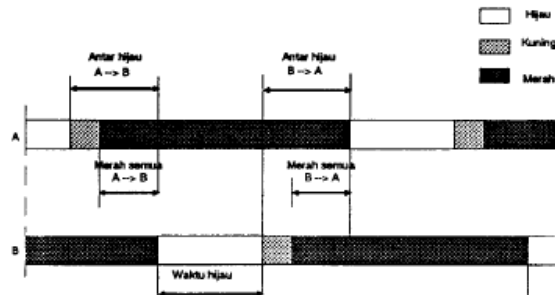
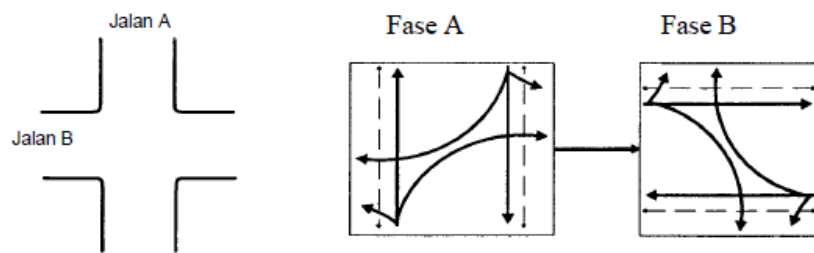


Gambar 2.1 Konflik-konflik utama dan kedua pada simpang bersinyal dengan empat lengan[1]

Gambar 2.1 memberikan penjelasan tentang urutan perubahan sinyal dengan sistem dua fase, termasuk definisi dari waktu siklus, waktu hijau dan periode antar hijau. Maksud dari periode antar hijau (IG = kuning + merah semua) di antara dua fase yang berurutan adalah untuk:

1. Memperingatkan lalu-lintas yang sedang bergerak bahwa fase sudah berakhir.
2. Menjamin agar kendaraan terakhir pada fase hijau yang baru saja diakhiri memperoleh waktu yang cukup untuk ke luar dari daerah konflik sebelum kendaraan pertama dari fase berikutnya memasuki daerah yang sama.

Fungsi yang pertama dipenuhi oleh waktu kuning, sedangkan yang kedua dipenuhi oleh waktu merah semua yang berguna sebagai waktu pengosongan antara dua fase.



Gambar 2.2 Urutan waktu pada pengaturan sinyal dengan dua-fase[1]

2.3.2 Perhitungan Siklus Lampu Lalu Lintas

Perhitungan dilakukan per satuan jam untuk satu atau lebih periode, misalnya didasarkan pada kondisi arus lalu-lintas rencana jam puncak pagi, siang dan sore. Arus lalu-lintas (Q) untuk setiap gerakan (belok-kiri QLT, lurus QST dan belokkanan QRT) dikonversi dari kendaraan per-jam menjadi satuan mobil penumpang (smp) per-jam dengan menggunakan ekivalen kendaraan penumpang (emp) untuk masing-masing pendekatan terlindungi dan terlawan.

Tabel 2.1 Pendekatan ekuivalen kendaraan penumpang (EMP)

Jenis Kendaraan	Emp Untuk Tipe Pendekatan	
	Terlindungi	Terlawan
Kendaraan Ringan (LV)	1,0	1,0
Kendaraan Berat (HV)	1,3	1,3
Sepeda Motor (MC)	0,2	0,4

Rumus Pendekatan Arus Lalu Lintas:

$$Q = Q_{LV} + Q_{HV} \times emp_{HV} + Q_{MC} \times emp_{MC} \dots\dots\dots(1)$$

Rumus Kapasitas Pendekatan Simpang Bersinyal:

$$C = S \times g/c \dots\dots\dots(2)$$

Definisi:

emp (Ekivalen Mobil Penumpang)	Faktor dari berbagai tipe kendaraan sehubungan dengan keperluan waktu hijau untuk keluar dari antrian apabila dibandingkan dengan sebuah kendaraan ringan (untuk mobil penumpang dan kendaraan ringan yang sasisnya sama, $emp=1,0$).
smp (Satuan Mobil Penumpang)	Satuan arus lalu-lintas dari berbagai tipe kendaraan.
Q (Arus Lalu Lintas)	Jumlah unsur lalu-lintas yang melalui titik tak terganggu di hulu, pendekat per satuan waktu (contoh: kebutuhan lalu-lintas kend/jam smp/jam).
C (Kapasitas)	Arus lalu-lintas maksimum yang dapat dipertahankan.
S (Arus Jenuh)	Besarnya keberangkatan antrian didalam suatu pendekat selama kondisi yang ditentukan (smp/jam hijau).
g (Waktu Hijau)	fuse untuk kendali lalu-lintas aktuasi kendaraan (detik).
c (Waktu Siklus)	Waktu untuk urutan lengkap dari indikasi sinyal (sbg. contoh, diantara dua saat permulaan hijau yang berurutan di dalam pendekat yang sama; detik).

2.4 Apache Web Server

Server HTTP Apache atau Server Web/WWW Apache adalah server web yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (GUI) yang memungkinkan penanganan server menjadi mudah. Apache HTTP server digunakan sebagai antarmuka pengguna berbasis grafik dengan program kendali utama. Beberapa modul bersama yang dipergunakan pada Tugas Akhir ini seperti modul PHP dan CGI (Common Gateway Interface).

2.5 PHP

PHP adalah bahasa server-side scripting yang dirancang untuk pengembangan web tetapi juga digunakan sebagai bahasa pemrograman tujuan umum. Yang Server-side scripting adalah skrip yang hanya dapat dieksekusi langsung pada server, dan skrip itu tidak dapat diunduh oleh klien/pengujung.

Kode PHP dapat dimasukkan ke dalam kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai sistem template web, sistem manajemen konten web, dan kerangka web. Kode PHP biasanya diolah oleh seorang juru bahasa PHP yang diimplementasikan sebagai modul di web server atau sebagai Common Gateway Interface (CGI) yang bisa dieksekusi.

Server web menggabungkan hasil kode PHP yang diinterpretasikan dan dieksekusi, yang mungkin merupakan jenis data, termasuk gambar, dengan halaman web yang dihasilkan. Kode PHP juga bisa dijalankan dengan command-line interface (CLI) dan bisa digunakan untuk mengimplementasikan aplikasi grafis mandiri.

2.6 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif.

Python dimaksudkan untuk menjadi bahasa yang mudah dibaca. Formatnya secara visual tidak berantakan, dan sering menggunakan kata kunci bahasa Inggris dimana bahasa lain menggunakan tanda baca. Tidak seperti banyak bahasa lainnya, tidak menggunakan kurung kurawal untuk membatasi blok, dan titik koma setelah pernyataan bersifat opsional. Ini memiliki lebih sedikit pengecualian sintaksis dan kasus khusus daripada C atau Pascal.

Sebuah studi empiris menemukan bahwa bahasa scripting, seperti Python, lebih produktif daripada bahasa konvensional, seperti C dan Java, untuk masalah pemrograman yang melibatkan manipulasi string dan pencarian di kamus, dan menentukan bahwa konsumsi memori sering kali "lebih baik daripada Java dan tidak jauh lebih buruk dari C atau C ++".

1.1

2.7 OpenCV

OpenCV (*Open Source Computer Vision*) adalah perpustakaan fungsi pemrograman yang ditujukan untuk penglihatan komputer secara real time. Awalnya dikembangkan oleh Intel, kemudian didukung oleh Willow Garage dan sekarang dikelola oleh Itseez. Perpustakaan ini cross-platform dan gratis untuk digunakan di bawah lisensi *open-source* BSD.

OpenCV ditulis dalam bahasa C ++ dan antarmuka utamanya ada di C ++, namun masih mempertahankan antarmuka C yang lebih tua namun luas. Ada bindings di Python, Java dan MATLAB / OCTAVE. API untuk antarmuka ini dapat ditemukan di dokumentasi online. Wrappers dalam bahasa lain seperti C #, Perl, Ch, Haskell dan Ruby telah dikembangkan untuk mendorong adopsi oleh khalayak yang lebih luas.

Fitur seperti Haar adalah fitur gambar digital yang digunakan dalam pengenalan objek. Mereka berutang nama mereka pada kesamaan intuitif mereka dengan wavelet Haar dan digunakan pada detektor wajah real-time pertama.

Secara historis, bekerja dengan hanya intensitas gambar (yaitu, nilai pixel RGB pada setiap piksel gambar) membuat perhitungan fitur perhitungan secara komputasi mahal. Publikasi oleh Papageorgiou et al. dibahas bekerja dengan seperangkat fitur alternatif berdasarkan wavelet Haar dan bukan intensitas gambar yang biasa. Viola dan Jones mengadaptasi gagasan menggunakan wavelet Haar dan mengembangkan fitur yang disebut Haar.

Fitur Haar-Like mempertimbangkan daerah persegi panjang yang berdekatan di lokasi tertentu di jendela deteksi, meringkas intensitas piksel di setiap wilayah dan menghitung perbedaan antara jumlah ini.

Perbedaan ini kemudian digunakan untuk mengkategorikan subbagian gambar. Sebagai contoh, katakanlah kita memiliki database gambar dengan wajah manusia. Ini adalah pengamatan umum bahwa di antara semua wajah daerah mata lebih gelap dari daerah

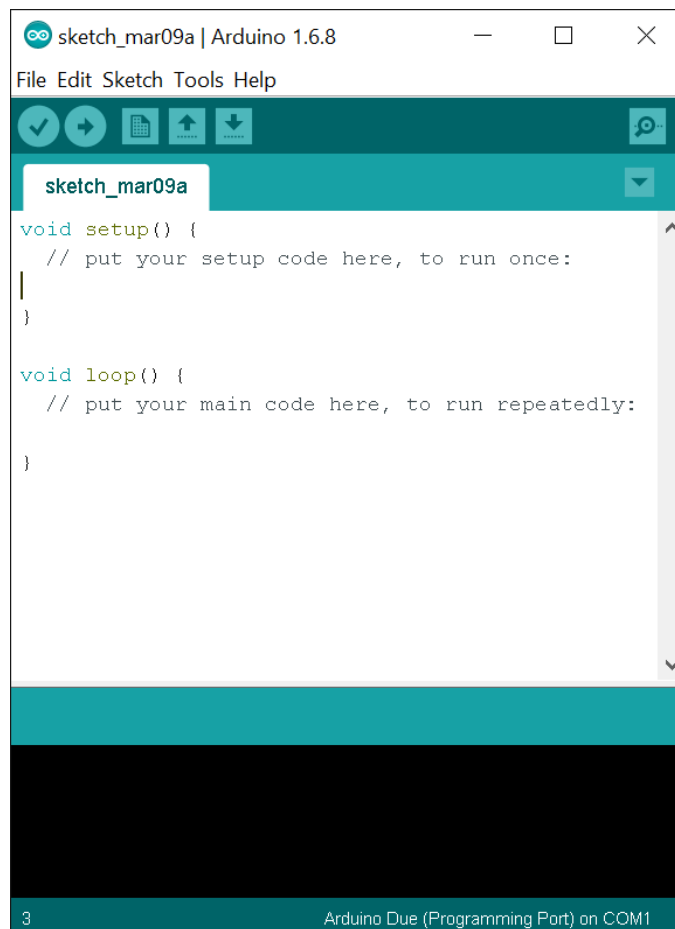
pipi. Oleh karena itu fitur Haar yang umum untuk deteksi wajah adalah sekumpulan dua persegi panjang yang berdekatan yang berada di atas mata dan daerah pipi. Posisi persegi panjang ini didefinisikan relatif terhadap jendela deteksi yang bertindak seperti kotak pembatas pada objek target (wajah dalam kasus ini) ^[8].

2.8 Arduino IDE

Arduino IDE (Integrated Development Environment) adalah software lingkungan pengembangan terpadu yang memungkinkan kita untuk menulis kode, membuat, mengupload dan mengatur berbagai setting pada hampir semua jenis mikrokontroler. Penampilan Arduino IDE seperti yang terlihat pada Gambar 2.3.

Hal ini dikarenakan Arduino IDE bersifat Open Source, yang memungkinkan semua pengembang perangkat lunak berkontribusi dalam bermacam-macam pustaka untuk berbagai mikrokontroler dan modul atau komponen lainnya.

Arduino IDE ditulis dalam bahasa Java, C dan C++. Program ataupun kode yang ditulis dalam Arduino IDE disebut “sketch”, memiliki struktur bahasa C/C++. Program tersebut memiliki 2 fungsi minimal, yaitu *setup()* dan *loop()*. Interface Arduino IDE dapat dilihat pada gambar 2.3.



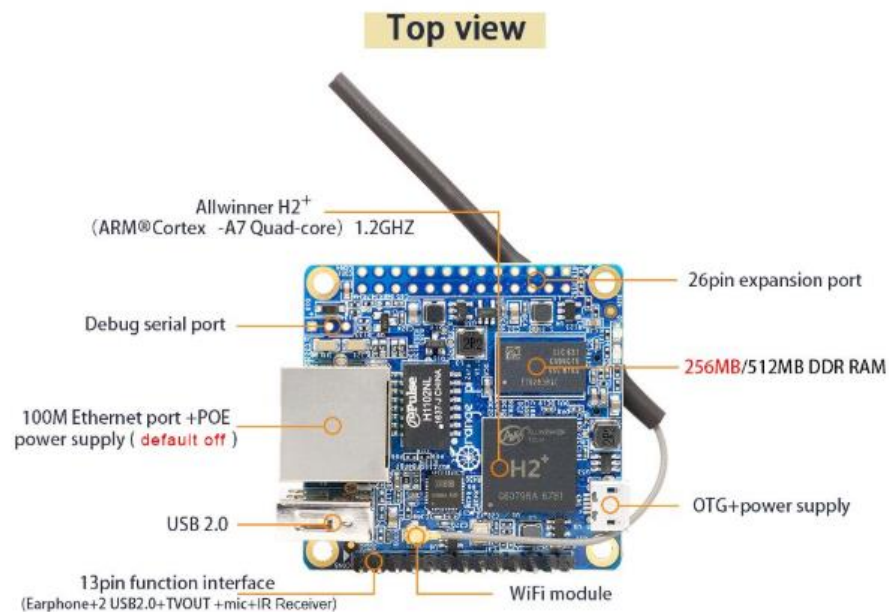
Gambar 2.3 Arduino IDE[10]

2.9 Orange Pi Zero

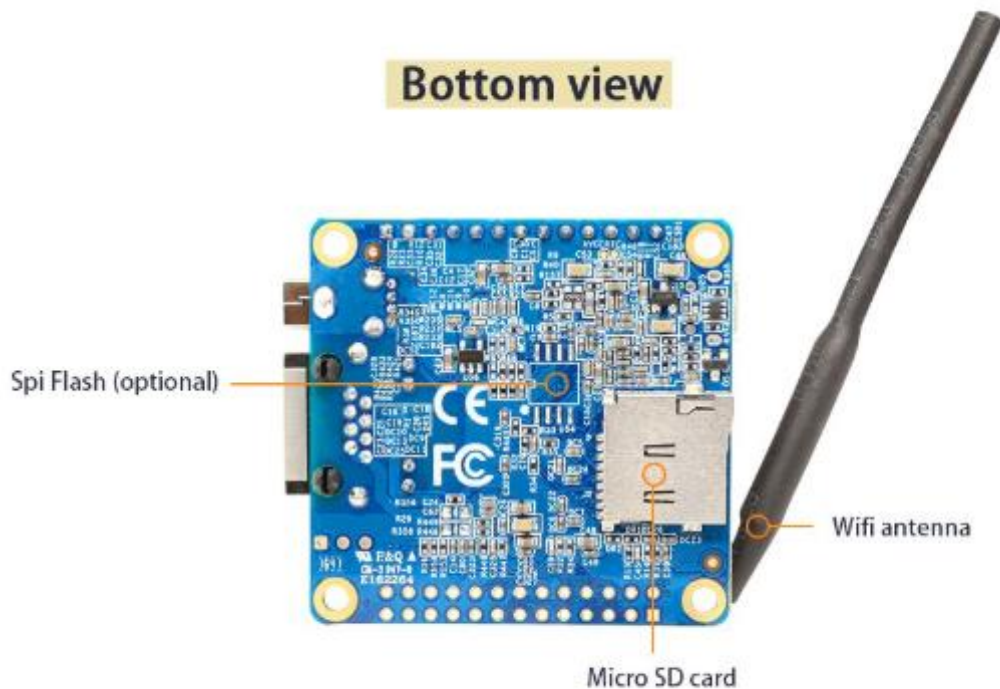
Orange Pi Zero adalah papan komputer tunggal yang bersifat opensource. Dapat menjalankan Sistem Operasi seperti Android 4.4, Ubuntu, Debian. Orange Pi Zero menggunakan Mikroprocessor jenis ARM Cortex-A7 AllWinner H2 SoC, dan memiliki 512MB DDR3 SDRAM. Spesifikasi Orange Pi Zero dapat dilihat pada tabel 2.2 dan bentuk perangkatnya seperti yang terlihat pada gambar 2.4 dan gambar 2.5

Tabel 2.2 Spesifikasi teknis Orange Pi Zero

Spesifikasi	
CPU	H2 Quad-core <u>Cortex-A7</u> H.265/HEVC 1080P.
GPU	·Mali400MP2 GPU @600MHz ·Supports OpenGL ES 2.0
Memory	256MB/512MB DDR3 SDRAM(Share with GPU)(256MB version is Standard version)
Onboard Storage	TF card (Max. 32GB)/ Spi Flash
Onboard Network	10/100M Ethernet RJ45 POE is default off.
Onboard WIFI	XR819, IEEE 802.11 b/g/n
Audio Input	MIC
Video Outputs	Supports external board via 13pins
Power Source	USB OTG, 5 VDC
USB 2.0 Ports	Only One USB 2.0 HOST, one USB 2.0 OTG
Buttons	Power Button
Low-level peripherals	26 Pins Header, compatible with Raspberry Pi B+ 13 Pins Header, with 2x USB, IR pin, AUDIO(MIC, AV)
LED	Power led & Status led
Supported OS	Android, Lubuntu, Debian, Raspbian



Gambar 2.4 Orange Pi Zero tampak atas [9]



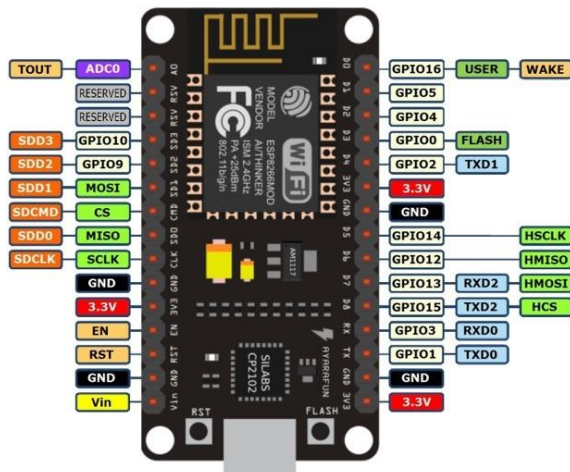
Gambar 2.5 Orange Pi Zero tampak bawah [9]

2.10 NodeMCU

NodeMCU adalah platform IoT open source. Ini mencakup *firmware* yang berjalan di ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang berbasis pada modul ESP-12. Spesifikasi dapat dilihat pada tabel 2.3 dan bentuk NodeMCU nya seperti yang terlihat pada gambar 2.6. Istilah "NodeMCU" secara default mengacu pada firmware daripada perangkat pengembangan. Firmware menggunakan bahasa scripting Lua. Hal ini didasarkan pada proyek eLua, dan dibangun di atas SDK Non-OS Espresso untuk ESP8266.

Tabel 2.3 Spesifikasi teknis NodeMCU 1.0 ESP-12E

Microcontroller	ESP8266
Operating Voltage	3.3V
Input Voltage (USB)	5V
Flash Memory	4 MB
SRAM	128 kB
Clock Speed	80 MHz

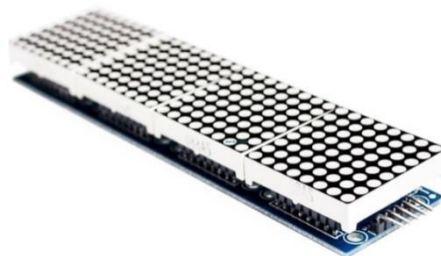


Gambar 2.6 NodeMCU 1.0 ESP8266-12E (AMICA)[10]

2.11 Dot Matrix Display

Dot-Matrix Display adalah perangkat layar yang digunakan untuk menampilkan informasi tentang mesin, jam, indikator keberangkatan kereta api dan banyak perangkat lain yang membutuhkan perangkat tampilan sederhana dengan resolusi terbatas. Bentuk display Dot-Matrix seperti terlihat pada gambar 2.7.

Layar terdiri dari dot matrix lampu atau indikator mekanis yang diatur dalam konfigurasi empat persegi panjang (bentuk lainnya juga mungkin, meski tidak umum) sehingga dengan menyalakan atau mematikan lampu, teks atau grafis yang dipilih dapat ditampilkan. Pengontrol dot matrix mengubah instruksi dari prosesor menjadi sinyal yang menghidupkan atau mematikan lampu dalam matriks sehingga tampilan yang dibutuhkan dihasilkan.



Gambar 2.7 Dot matrix display 32x8[11]

2.12 Logitech C270

Logitech C270 merupakan kamera jenis WebCam HD yang memiliki resolusi 3 Mega Pixel (1280 x 720 pixels), WebCam ini didesain untuk Video Call layar lebar (Wide

Screen) dengan resolusi 720 pixel. Kamera ini dilengkapi dengan koreksi level cahaya secara otomatis. Kamera ini dilengkapi juga dengan teknologi Logitech Fluid Crystal™ yang membuat kamera ini dapat menangkap video dan gambar dengan tajam dan ketajaman warna yang tinggi. WebCam tersebut dapat dilihat pada gambar 2.8.



Gambar 2.8 Logitech C270

Spesifikasi teknis:

- HD video calling (1280 x 720 pixels) with recommended system
- Video capture: Up to 1280 x 720 pixels
- Logitech Fluid Crystal™ Technology
- Photos: Up to 3.0 megapixels (software enhanced)
- Built-in mic with noise reduction
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors
- FoV: 60°

2.13 Konsep Pendeteksian Gambar Dengan Kamera

Sistem penghitungan kendaraan otomatis ini menggunakan data video yang diperoleh dari kamera lalu lintas. Metode blob detection digunakan untuk mengolah serangkaian frame yang diperoleh dari video untuk menghitung jumlah kendaraan pada setiap jalur. Pada setiap frame ditentukan region of interest (ROI) dengan melacak objek yang terdeteksi di dalam ROI tertentu dan kemudian melakukan perhitungan. Beberapa teori yang digunakan pada deteksi kendaraan adalah sebagai berikut.

A. Region of Interest (ROI)

ROI merupakan subcitra dari suatu region. Hal ini merefleksikan kenyataan bahwa suatu citra seringkali memiliki suatu koleksi objek yang dapat menjadi suatu region tersendiri. ROI memungkinkan dilakukannya pengkodean secara berbeda pada area tertentu

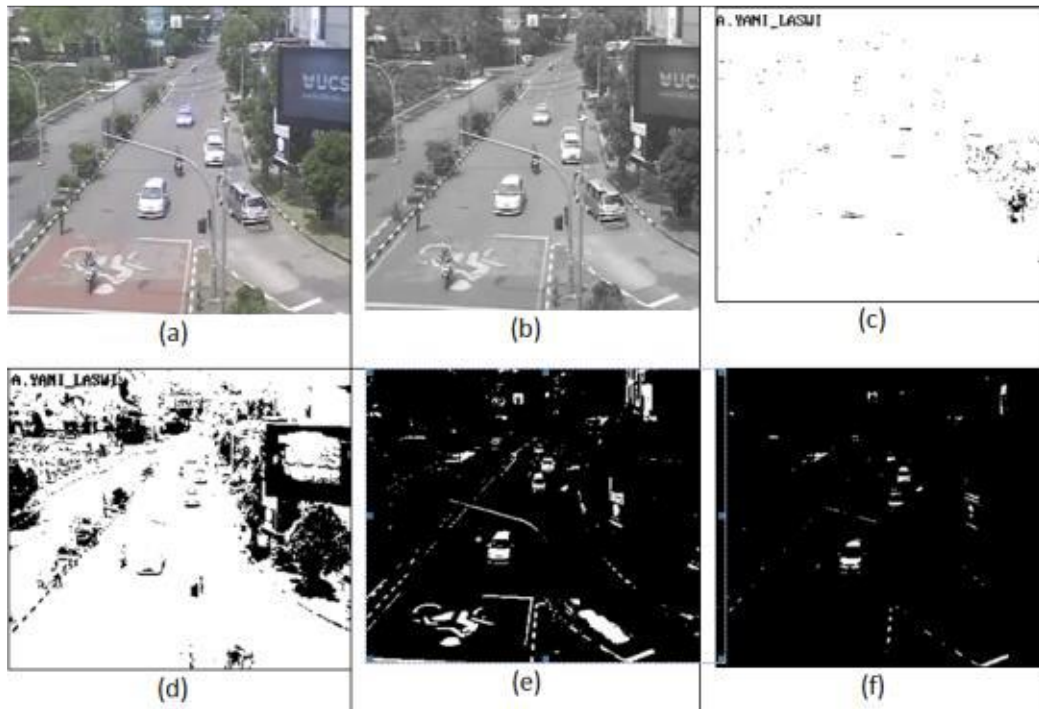
dari citra digital, sehingga mempunyai kualitas yang lebih baik dari area sekitarnya (background). Fitur ini menjadi sangat penting bila terdapat bagian tertentu dari citra digital yang dianggap lebih penting dari bagian lainnya. Metode seleksi ROI yang digunakan adalah dengan piksel karena model jalur yang digunakan lebih memungkinkan jika menggunakan metode piksel.

B. Thresholding

Thresholding digunakan untuk mengatur gray-level yang ada pada gambar. Misalnya pada sebuah gambar, $f(x,y)$ tersusun dari objek yang terang pada sebuah background yang gelap. Gray-level milik objek dan milik background terkumpul menjadi dua grup yang dominan. Salah satu cara untuk mengambil objek dari background-nya adalah dengan memilih sebuah nilai ambang (threshold) T yang memisahkan grup yang satu dengan grup yang lain. Maka, semua piksel yang memiliki nilai $> T$ disebut titik objek, sedangkan yang lain disebut titik background. Proses ini disebut thresholding. Sebuah gambar yang telah dikenai proses thresholding $g(x,y)$ dapat didefinisikan sebagai berikut [6].

$$g(x,y) = \begin{cases} 1 & \text{jika } f(x,y) > T \\ 0 & \text{jika } f(x,y) \leq T \end{cases} \dots\dots\dots(3)$$

Nilai T dapat ditentukan melalui perhitungan rata-rata dari keseluruhan nilai karna yang ada pada gambar. Pada perhitungan ini, nilai T yang diperoleh tetap disimpan dalam bilangan real. Nilai T yang diperoleh untuk gambar yang memiliki histogram yang telah mengalami proses penyamaan (equalization) adalah berkisar antara 127 dan 128. Nilai maksimum T adalah nilai tertinggi dari sistem warna yang digunakan dan nilai minimum T adalah nilai terendah dari sistem warna yang digunakan. Untuk 256-graylevel maka nilai tertinggi T adalah 255 dan nilai terendahnya adalah 0 [6]. Pengaruh nilai ambang ditunjukkan pada gambar 2.9.



Gambar 2.9 Pengaruh nilai ambang, (a) citra referensi, (b) citra gray-scale, (c) citra dengan TV=60. (d) citra dengan TV=100, (e) citra dengan TV=200 dan (f) citra dengan TV=250.[15]

C. Smoothing Filter

Smoothing, juga disebut blurring/pengaburan adalah operasi pengolahan citra sederhana dan sering digunakan. Ada banyak alasan dilakukannya smoothing. Pada dasarnya, smoothing digunakan untuk mengurangi derau. Untuk melakukan operasi smoothing, digunakan filter. Jenis filter yang paling umum adalah dari filter yang linear, dengan nilai keluaran piksel ($g(i, j)$) ditentukan sebagai jumlah bobot nilai masukan piksel yaitu, $f(i+k, j+l)$ [7].

$$g(i, j) = \sum_{k, l} f(i+k, j+l)h(k, l) \dots\dots\dots(4)$$

dengan $h(k, l)$ disebut kernel, yang tidak lebih dari koefisien filter. Ini membantu untuk memvisualisasikan filter sebagai jendela koefisien meluncur di gambar. Ada banyak jenis filter, di antaranya adalah Gaussian dan filter median [8], seperti yang ditunjukkan pada gambar 2.10.



Gambar 2.10 (a) Sebelum smoothing, (b) hasil smoothing[15]

D. Image Morphology

Image morphology merupakan teknik atau proses yang digunakan untuk mengolah citra berdasarkan prinsip morfologi matematika. Dalam pemrosesan citra, hasil yang diharapkan diperoleh berdasarkan bentuk atau struktur citra asal. Lebih lanjut dikatakan bahwa morfologi senantiasa berkaitan erat dengan proses ketetanggaan (*neighborhood*) yang terbentuk dari blok nilai biner, satu dan nol [9]. Lebih lanjut lagi, proses morfologi suatu citra merupakan kumpulan operasi nonlinear yang berkaitan dengan bentuk atau morfologi dalam suatu citra [10].

Hasil operasi morfologi dimanfaatkan untuk pengambilan keputusan dengan analisis lebih lanjut. Operasi ini antara lain meliputi pencarian batas/kontur, dilasi, erosi, penutupan (*closing*), pembukaan (*opening*), pengisian (*filling*), pelabelan, dan pengerangkaan (*skeletonization*) [11].

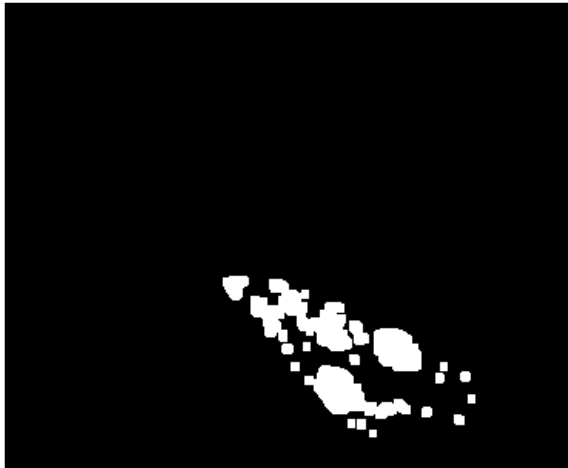
E. Dilasi

Dilasi adalah transformasi morfologi yang menggabungkan dua himpunan dengan menggunakan penjumlahan vektor elemen himpunan. Dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan, dengan contoh seperti pada gambar 2.11. Proses ini adalah kebalikan dari erosi, yaitu mengubah latar di sekeliling objek menjadi bagian dari objek tersebut [12]. Ada dua cara untuk melakukan operasi ini, yaitu sebagai berikut.

1. Dengan mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek.
2. Dengan mengubah semua titik di sekeliling titik batas menjadi titik objek.

Operasi morfologi dasar matematika dilasi dilakukan berdasarkan aljabar Minkowski, ditunjukkan oleh (3) [12].

$$D(A, B) = A + B = U_{\beta \in B} (A + \beta) \dots\dots\dots(5)$$



Gambar 2.11 Dilasi[15]

F. Erosi

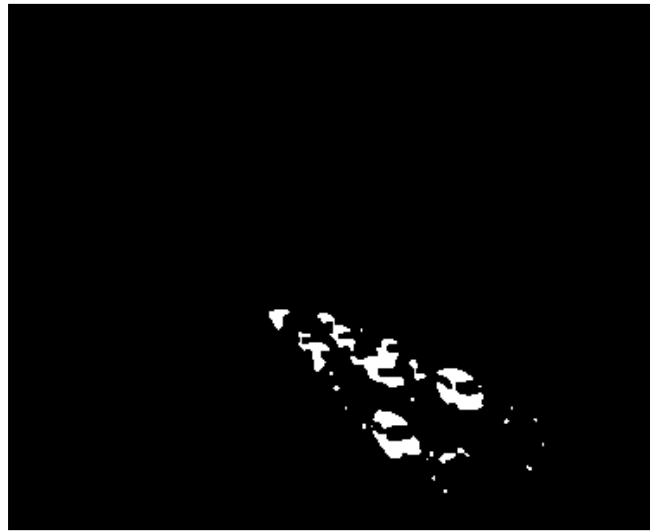
Erosi dapat diperoleh dengan melebarkan komplemen dari piksel hitam dan kemudian mengambil komplemen dari set point yang dihasilkan. Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan structuring element yang digunakan, dengan contoh yang ditunjukkan pada gambar 2.12. [12]. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Ada dua cara yang dapat dilakukan untuk melakukan proses erosi, yaitu sebagai berikut.

1. Dengan mengubah semua titik batas menjadi titik latar.
2. Dengan mengatur semua titik di sekeliling titik latar menjadi titik latar

Operasi morfologi dasar matematika erosi dilakukan berdasarkan aljabar Minkowski seperti pada (4) [12].

$$E(A,B) = A - (-B) = \bigcup_{\beta \in B} (A - \beta) \text{ dengan } -B = \{-B | \beta \in B\}. \dots\dots\dots(6)$$

Himpunan A atau B dapat dianggap sebagai "citra". A biasanya dianggap sebagai citra dan B disebut structuring element.

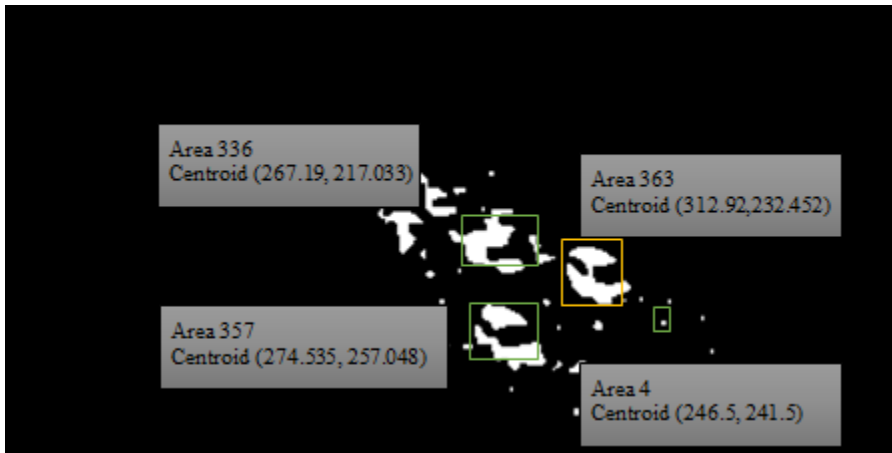


Gambar 2.12 Erosi[15]

G. Blob Detection

Blob detection adalah algoritma yang digunakan untuk menentukan suatu grup dari piksel saling berhubungan satu sama lain atau tidak. Metode ini sangat berguna untuk mengidentifikasi objek yang terpisah-pisah pada suatu citra atau menghitung jumlah dari suatu objek pada suatu citra. Pada metode blob detection, citra harus diproses dengan metode ambang terlebih dahulu, berdasarkan warna yang akan dideteksi. Setelah itu, citra dengan warna di atas nilai ambang dapat dikategorikan sesuai dengan aturan yang telah ditentukan terlebih dahulu. Misalnya untuk piksel yang memiliki nilai lebih kecil daripada nilai ambang akan dikelompokkan sebagai komponen dari objek yang dideteksi, sedangkan yang nilainya di atas ambang dikelompokkan sebagai informasi yang tidak penting. Sedangkan untuk mendapatkan titik berat massa adalah dengan merata-ratakan lokasi tiap piksel dengan warna tertentu.

Bagi banyak pengolahan citra, mendeteksi objek low-level dalam sebuah gambar merupakan hal yang sangat penting. Objek dalam bentuk dua atau tiga dimensi tersebut biasa disebut blob. Bentuk blob timbul dalam cara yang berbeda, bergantung pada ukuran, dan dapat dideteksi dengan menggunakan metode sederhana dalam sebuah representasi gambar [13]. Gambar 2.13 memperlihatkan *blob detection* dari video lalu lintas.



Gambar 2.13 Blob detection[15]