

BAB II

TINJAUAN PUSTAKA

2.1 Lampu Lalu Lintas



Gambar 2.1 Lampu Lalu Lintas

Sumber : (Zikra, 2022)

Lampu lalu lintas yaitu lampu yang berfungsi untuk mengontrol arus lalu lintas yang dipasang pada persimpangan jalan, *zebra cross*, dan tempat arus lalu lintas lainnya. Lampu lalu lintas dapat menunjukkan kapan kendaraan harus berhenti dan berjalan secara bergantian dari berbagai arah (Veda, 2021). Pengaturan lampu lalu lintas di persimpangan jalan bertujuan untuk mengatur pergerakan setiap kendaraan agar kendaraan dapat bergerak secara bergantian sehingga tidak saling mengganggu antar arus yang ada.

Lampu Lalu Lintas (*Traffic Light*) telah diadopsi di hampir seluruh kota di dunia dengan menggunakan warna yang diakui secara universal yaitu warna merah, kuning, dan hijau seperti ditunjukkan pada Gambar 2.1. Warna merah digunakan untuk menandakan berhenti, warna kuning untuk menandakan berhati-hati, dan warna hijau digunakan untuk memberi tahu bahwa kendaraan dapat berjalan (Hasfar & Adiwarsa, 2018).

2.1.1 Cara Kerja Lampu Lalu Lintas

Pada dasarnya lampu lalu lintas memiliki dua lampu indikator utama yaitu lampu merah yang berarti berhenti dan lampu hijau yang berarti dapat berjalan. Selain itu, lampu lalu lintas dilengkapi dengan satu lampu indikator tambahan yaitu lampu kuning yang berarti hati-hati. Disaat lampu merah menyala, seluruh kendaraan dari jalur tersebut tidak dapat berjalan. Sedangkan ketika lampu berubah menjadi lampu hijau, kendaraan dari jalur tersebut dapat mulai berjalan. Kemudian ketika lampu berubah menjadi kuning, kendaraan masih dapat berjalan tetapi harus berhati-hati karena lampu kuning hanya menyala beberapa detik saja sebelum berubah menjadi warna merah (Zikra, 2022). Namun, untuk lampu kuning terdapat dua kemungkinan, pertama lampu kuning yang menyala setelah lampu hijau padam yang berarti bahwa lampu merah akan segera menyala sehingga kendaraan harus bersiap untuk berhenti. Kedua yaitu lampu kuning yang menyala bersama dengan lampu merah yang berarti lampu hijau akan segera menyala sehingga kendaraan harus segera bersiap untuk bergerak (MENHUB RI, 2014).

2.1.2 Pengontrolan Lampu Lalu Lintas

Menurut PERMENHUB RI Nomor PM 49 Tahun 2014, pengaturan waktu siklus lampu lalu lintas atau Alat Pemberi Isyarat Lalu Lintas (APILL) terdiri dari:

1. Waktu Siklus Terkoordinasi

Waktu siklus terkoordinasi yaitu skema rencana siklus antar APILL yang diatur oleh sistem yang terpusat.

2. Waktu Siklus Tidak Terkoordinasi

Waktu siklus tidak terkoordinasi dibagi ke dalam 3 jenis dengan fungsi yang berbeda-beda. Ketiga jenis tersebut diantaranya:

1. Siklus Tetap

Waktu siklus tetap berupa rencana siklus yang tetap atau tidak dapat diubah-ubah dan paling sedikit memiliki 8 rencana siklus.

2. Siklus Semi Adaptif

Waktu siklus semi adaptif berupa rencana siklus yang tetap pada kaki simpang mayor dengan rencana siklus paling sedikit memiliki 8 rencana siklus serta rencana siklus yang bervariasi pada kaki simpang minor.

3. Siklus Adaptif

Waktu siklus adaptif berupa rencana siklus yang bervariasi pada kedua jenis kaki simpang yaitu kaki simpang mayor dan kaki simpang minor menurut situasi arus lalu lintas (MENHUB RI, 2014).

2.1.3 Sistem Lampu Lalu Lintas di Indonesia



Gambar 2.2 *Central Control System Room ATCS*

Sumber : (DISHUB Kota Bandung, 2023)

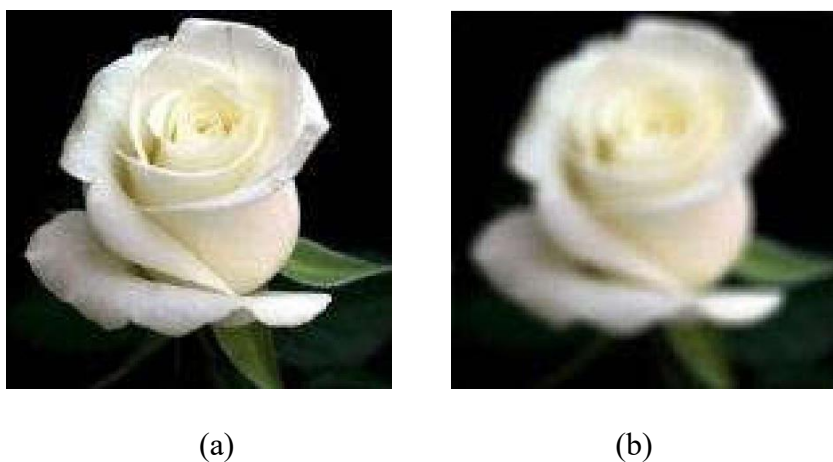
Saat ini, sistem lampu lalu lintas yang banyak diterapkan di Indonesia yaitu menggunakan *Area Traffic Control System* (ATCS). ATCS ini dibentuk di Indonesia pada tahun 1990-an dan diterapkan pertama kali di kota Malang yang kemudian diikuti oleh banyak kota di Indonesia (Silaban & Hazzah Nur, 2023). ATCS merupakan sistem pengendalian lalu lintas berbasis teknologi informasi pada suatu kawasan yang bertujuan untuk mengoptimalkan kinerja jaringan jalan melalui optimasi dan koordinasi pengatur lampu lalu lintas di setiap persimpangan (Kurniati, 2019). ATCS mengontrol lalu lintas dengan menyelaraskan waktu lampu merah pada jaringan jalan raya dari sebuah kota. Pengontrolan lalu lintas menggunakan sistem ini memerlukan parameter jumlah kendaraan dan waktu tempuh kendaraan (Silaban & Hazzah Nur, 2023).

ATCS merupakan sistem lampu lalu lintas terpusat di mana seluruh unsur APILL dikoneksikan dengan *channel* komunikasi antara kontroler di lapangan dengan komputer di *Central Control System Room* seperti terlihat pada Gambar 2.2. ATCS memiliki sistem operasi menggunakan server dan *workstation* yang berfungsi sebagai pusat operasional untuk memonitor dan mengendalikan kondisi lalu lintas dari semua persimpangan dalam satu area *wall map* yang berfungsi sebagai tempat untuk menyediakan informasi status dan kondisi dari *local controller*, *Video surveillance* (CCTV), dan detektor kendaraan.

ATCS digunakan untuk mengatur waktu sinyal di persimpangan secara responsif dan terkoordinasi dan dalam kondisi tertentu memberikan lampu hijau ketika terdapat kendaraan prioritas seperti ambulans, tamu kenegaraan, dan lainnya. Selain itu, ATCS juga dapat menyampaikan informasi kondisi lampu lalu lintas dan

alternatif lintasan, menyediakan rekaman data lalu lintas, kejadian kecelakaan dan kejadian-kejadian lainnya di persimpangan (Kurniati, 2019).

2.2 Image Processing



Gambar 2.3 (a) Citra Asli (b) Hasil *Image Processing*

Sumber : (Rafiq et al., 2019)

Image processing adalah proses pengolahan dan analisis citra/gambar yang banyak melibatkan persepsi visual (Rafiq et al., 2019). *Image processing* adalah suatu cara menjadikan suatu citra menjadi citra lain sesuai dengan yang kita inginkan (Sulistiyanti et al., 2016). *Image processing* berfungsi untuk menghasilkan kualitas citra yang lebih baik daripada citra sebelumnya yang mengalami gangguan dengan cara memanipulasi citra agar citra yang mengalami gangguan tersebut mudah diinterpretasi baik oleh manusia maupun mesin (Alrynto, 2018). Selain itu, *Image processing* juga berfungsi untuk memproses informasi pada citra/gambar untuk keperluan yang diinginkan biasanya untuk mengenali pola suatu objek (Sulistiyanti et al., 2016).

Terdapat 2 jenis metode yang digunakan dalam *image processing* yaitu pengolahan citra analog dan pengolahan citra digital. Pengolahan citra analog

digunakan seperti pada cetakan dan foto. Sedangkan pengolahan citra digital dapat membantu dalam manipulasi digital dari citra digital dengan menggunakan komputer seperti terlihat pada Gambar 2.3. Pada pengolahan citra digital, terdapat 3 fase umum yang harus dilalui oleh semua jenis data yaitu *preprocessing*, peningkatan dan tampilan, dan ekstraksi informasi. Selain itu, pengolahan citra digital berfokus pada 2 tugas utama yaitu peningkatan informasi bergambar untuk interpretasi manusia dan pemrosesan data citra untuk penyimpanan, pengiriman, dan representasi untuk persepsi mesin otonom (Rafiq et al., 2018).

2.2.1 Segmentasi Citra



Gambar 2.4 (a) Citra Asli (b) Hasil Segmentasi Citra

Sumber : (Rafiq et al., 2019)

Segmentasi citra yaitu proses membagi suatu citra digital menjadi beberapa bagian atau objek. Tujuan dari segmentasi citra yaitu untuk memisahkan objek atau daerah tertentu dalam citra seperti terlihat pada Gambar 2.4. Objek citra yang tersegmentasi disebut latar depan, sedangkan sisa citra disebut latar belakang (Dr. Pranowo, 2015). Segmentasi citra merupakan tahap awal yang dilakukan sebelum

tahapan analisis citra lanjutan dalam proses pengenalan citra dari suatu inputan tertentu (Rafiq et al., 2019).

Pendekatan metode segmentasi pada dasarnya dibagi menjadi 2 kelompok, yaitu:

1. Metode berdasarkan pencarian tepi atau batas (*edge-based*)

Metode ini mencari *edge* suatu citra untuk menentukan batas suatu objek pada citra. Batas yaitu lokasi di mana terjadi perubahan intensitas (Iriyanto & Zaini, 2014). Metode ini mirip dengan metode *edge detection*, hanya saja tepi segmentasinya berupa kurva yang berkesinambungan (Dr. Pranowo, 2015).

2. Metode berdasarkan pencarian daerah atau wilayah (*region-based*)

Dalam metode berdasarkan pencarian daerah atau wilayah, identifikasi dilakukan melalui area yang terdapat dalam objek tersebut (Iriyanto & Zaini, 2014). Metode ini menandai piksel-piksel yang memiliki warna, nilai, atau pola tekstur yang mirip sehingga membentuk suatu area (Dr. Pranowo, 2015).

2.2.1.1 Thresholding

Thresholding yaitu konversi dari citra *greyscale* ke citra biner dengan cara mengelompokkan nilai derajat keabuan setiap piksel ke dalam 2 kelas, yaitu hitam dan putih (M. A. A. Saputra et al., 2021). Hasil dari konversi citra tersebut yaitu dapat mengetahui daerah mana yang termasuk objek dan latar belakang dari citra secara jelas (Maria et al., 2018). *Thresholding* sangat berguna untuk membedakan *background* dari objek gambar yang diminati. Proses *thresholding* dilakukan pada citra digital dan menetapkan nilai *threshold* citra pada citra tersebut. Nilai *threshold*

diperoleh dari bagian histogram pada citra *grayscale*, selanjutnya citra tersebut diubah ke dalam bentuk citra biner (Wijayono et al., 2018).

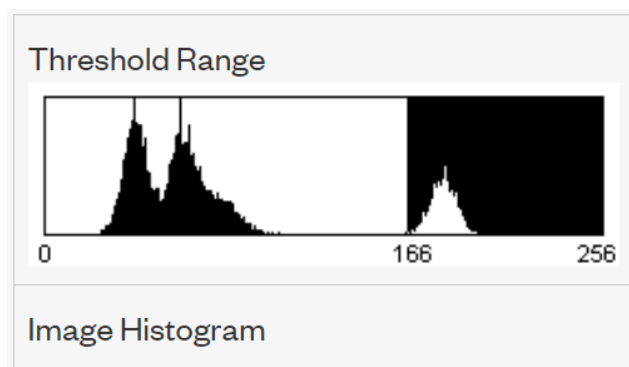
Thresholding menggunakan nilai piksel dalam gambar untuk mengelompokkan gambar menjadi dua wilayah yaitu wilayah partikel yang berisi objek yang diperiksa dan wilayah *background*. Rentang nilai piksel dapat ditentukan baik oleh pengguna atau secara otomatis sebagai nilai ambang batas. Nilai piksel apa pun di luar rentang menjadi 0 dan nilai piksel apa pun di dalam rentang menjadi 1, atau nilai berbeda yang ditentukan oleh pengguna. Hasil ambang batas yaitu dalam bentuk gambar biner (National Instruments, 2021).

Terdapat beberapa macam metode *Thresholding* diantaranya:

1. *Global Greyscale Thresholding*

Metode ini berfungsi paling baik pada gambar skala abu-abu dengan pencahayaan seragam. Konsep dasar dari metode ini menyatakan bahwa partikel dicirikan oleh rentang intensitas. Partikel ini terdiri dari piksel dengan nilai tingkat keabuan yang termasuk dalam interval ambang batas tertentu. Sedangkan semua piksel lainnya yang berada di luar interval ambang batas tersebut dianggap sebagai bagian dari *background*. Ambang batas menetapkan semua piksel yang termasuk dalam rentang nilai piksel yang disebut interval ambang batas menjadi 1 atau nilai yang ditentukan pengguna dan menetapkan semua piksel lain dalam gambar menjadi 0.

Gambar 2.5 menunjukkan histogram dari suatu gambar. Semua piksel dalam gambar yang nilainya berkisar antara 166 hingga 255 dianggap piksel partikel.



Gambar 2.5 Histogram Suatu Gambar

Sumber : (National Instruments, 2021)

Global greyscale thresholding memiliki dua teknik yang dapat digunakan yaitu teknik *manual threshold* dan teknik *automatic threshold*. Interval *threshold* dalam *manual threshold* memiliki dua parameter yang ditentukan pengguna yaitu *lower threshold* dan *upper threshold*. Semua piksel yang memiliki nilai tingkat keabuan sama dengan atau lebih besar dari ambang batas bawah dan sama dengan atau lebih kecil dari ambang batas atas dipilih sebagai piksel milik partikel dalam gambar. Berbeda dengan ambang batas manual, teknik *automatic threshold* tidak mengharuskan untuk menetapkan nilai *lower threshold* dan *upper threshold*. Teknik ini sangat cocok untuk kondisi di mana intensitas cahaya bervariasi dari satu gambar ke gambar lainnya.

2. *Global Color Thresholding*

Global color thresholding mengubah gambar berwarna menjadi gambar biner. Metode ini berfungsi paling baik pada gambar berwarna dengan pencahayaan seragam. Untuk melakukan ambang batas pada gambar berwarna, harus ditentukan interval ambang batas untuk masing-

masing dari ketiga komponen warnanya. Piksel pada gambar *output* diatur ke 1 apabila komponen warnanya berada dalam rentang yang ditentukan. Jika tidak, maka nilai pikselnya disetel ke 0. Terdapat 2 jenis gambar yaitu gambar RGB dan gambar HSL.

Pada gambar RGB, Untuk melakukan *thresholding* yaitu menentukan nilai merah, hijau, dan biru dari piksel yang membentuk objek yang ingin dianalisis setelah melakukan *thresholding*. Kemudian, menentukan nilai rentang *thresholding* untuk setiap bidang warna yang mencakup nilai warna yang diinginkan. Terakhir, memilih rentang yang benar untuk ketiga bidang warna untuk mengisolasi warna yang diinginkan.

Sedangkan untuk melakukan *thresholding* pada gambar HSL yaitu dengan menentukan nilai rona piksel yang ingin dianalisis setelah melakukan *thresholding*. Dalam beberapa aplikasi, terkadang perlu memilih warna dengan nilai rona yang sama, tetapi dengan nilai saturasi yang berbeda. Untuk bidang luminansi hanya mencakup informasi tentang tingkat intensitas dalam gambar, sehingga perlu mengatur rentang ambang batas luminansi untuk menyertakan semua nilai luminansinya, sehingga membuat proses *thresholding* tidak bergantung pada informasi intensitas.

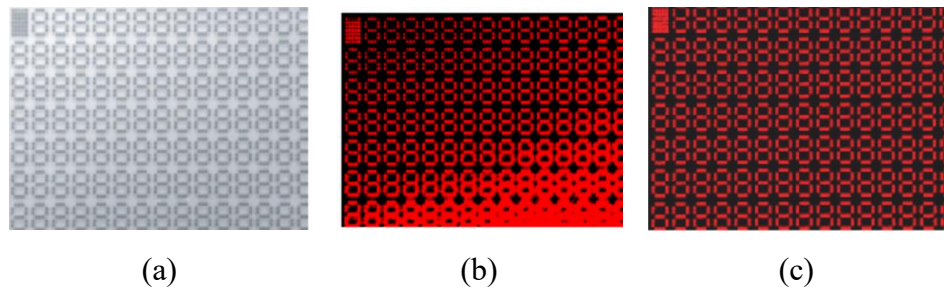
3. *Local Thresholding*

Local thresholding digunakan untuk mengisolasi objek menarik dari *background* pada gambar yang menunjukkan perubahan pencahayaan yang tidak seragam, misalnya akibat gradien pencahayaan atau bayangan yang kuat yang biasanya menyebabkan *global thresholding* menjadi tidak efektif.

Local thresholding memiliki kemiripan dengan *global greyscale thresholding* yang membuat gambar biner dengan mengelompokkan gambar skala abu-abu ke dalam wilayah partikel dan wilayah *background*. Namun, tidak seperti *global greyscale thresholding* yang mengkategorikan piksel sebagai bagian dari partikel atau *background* berdasarkan nilai *thresholding* tunggal yang diperoleh dari statistik intensitas keseluruhan gambar, *local thresholding* mengkategorikan piksel berdasarkan statistik intensitas *neighboring pixels*.

Local thresholding menghitung statistik intensitas piksel lokal seperti varians, rentang, parameter pemasangan permukaan, atau kombinasi logisnya untuk setiap piksel dalam gambar inspeksi. Hasil perhitungan ini adalah nilai *local thresholding* untuk piksel yang ditinjau. *Local thresholding* membandingkan nilai intensitas asli piksel yang dipertimbangkan dengan nilai *local thresholding* dan menentukan apakah piksel tersebut termasuk partikel atau *background*.

Gambar 2.6 menunjukkan pengaruh *global thresholding* dan *local thresholding* pada gambar dengan perubahan pencahayaan yang tidak seragam. Gambar 2.6 (a) menunjukkan gambar asli dari digit LCD. Gambar 2.6 (b) menunjukkan hasil *global thresholding* dengan banyak piksel nondigit di pojok kanan bawah dipilih secara keliru sebagai partikel. Gambar 2.6 (c) menunjukkan hasil *local thresholding* di mana hanya piksel milik digit LCD yang dipilih sebagai partikel.



Gambar 2.6 (a) Gambar Asli (b) *Global Thresholding* (c) *Local*

Thresholding

Sumber : (National Instruments, 2021)

Fungsi *local thresholding* pada umumnya memerlukan waktu komputasi yang besar. Selain itu, waktu yang diperlukan untuk menyelesaikan fungsi *local thresholding* sering kali bervariasi bergantung pada ukuran jendela. Kurangnya determinisme mencegah *local thresholding* digunakan dalam aplikasi *real time*. Fungsi *local thresholding* NI Vision menggunakan implementasi algoritma yang sepenuhnya dioptimalkan dan efisien sehingga kecepatan komputasinya tidak bergantung pada ukuran jendela. Hal ini secara signifikan mengurangi biaya komputasi dan memungkinkan penggunaan fungsi dalam aplikasi segmentasi waktu nyata (National Instruments, 2021).

2.3 Machine Vision

Menurut *Automated Imaging Association (AIA)*, *machine vision* mencakup semua aplikasi industri dan nonindustri di mana kombinasi *hardware* dan *software* memberikan panduan operasional kepada perangkat dalam menjalankan fungsinya berdasarkan pengambilan dan pemrosesan gambar. Sistem *machine vision* mengandalkan sensor digital yang dilindungi di dalam kamera industri dengan optik

khusus untuk memperoleh gambar, sehingga *hardware* dan *software* komputer dapat memproses, menganalisis, dan mengukur berbagai karakteristik untuk pengambilan keputusan. Sistem *machine vision* yang dibangun berdasarkan resolusi kamera dan optik yang tepat dapat dengan mudah memeriksa detail objek yang terlalu kecil untuk dilihat oleh mata manusia (Cognex, 2016).

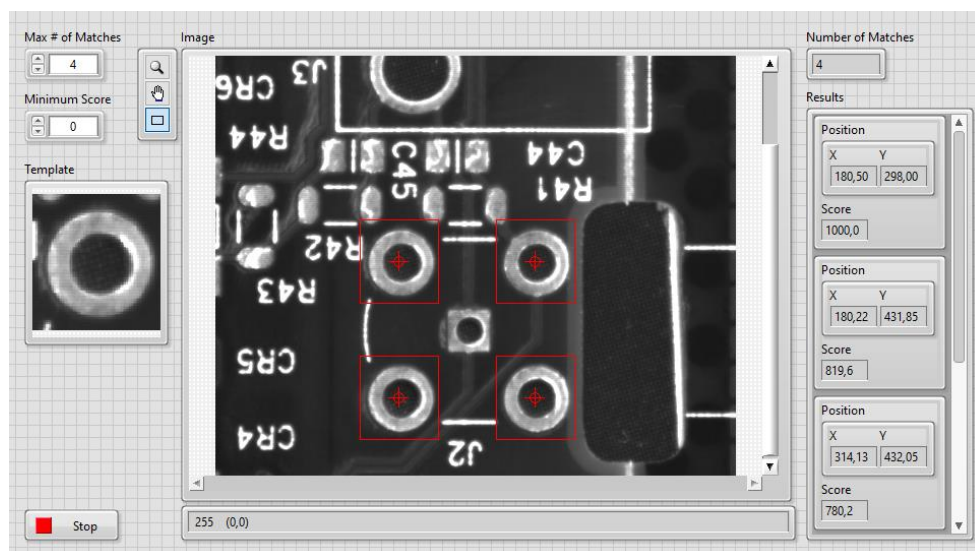
Tujuan dari *machine vision* adalah untuk menciptakan model dunia nyata dari gambar. Sistem *machine vision* memulihkan informasi berguna tentang suatu pemandangan dari proyeksi dua dimensinya karena gambar adalah proyeksi dua dimensi dari dunia yang tiga dimensi, sehingga informasinya tidak tersedia secara langsung dan harus diperoleh kembali (Jain et al., 1995).

2.3.1. *Pattern Matching*

Pattern matching merupakan metode yang digunakan untuk menemukan daerah pada citra *greyscale* yang cocok dengan pola citra referensi. Jika citra awal adalah citra berwarna, maka citra tersebut harus dikonversi ke citra *greyscale* terlebih dahulu agar dapat menggunakan *pattern matching*. *Pattern matching* menggunakan citra referensi atau templat untuk menemukan citra serupa dalam citra baru tanpa memandang lokasi, rotasi, atau skala templat (Kwon & Ready, 2015). *Pattern matching* dengan cepat menemukan wilayah citra *greyscale* yang cocok dengan citra referensi yang diketahui atau disebut juga sebagai model atau templat. Saat menggunakan *pattern matching*, templat yang dibuat harus mewakili objek yang akan dicari. Kemudian, *machine vision* akan mencari contoh templat di setiap citra yang diperoleh, lalu menghitung skor untuk setiap kecocokan. Skor ini

berkaitan dengan seberapa mirip templat tersebut dengan kecocokan yang ada (National Instruments, 2021).

Proses *pattern matching* terdiri dari 2 tahap yaitu tahap pembelajaran dan tahap pencocokan. Selama tahap pembelajaran, algoritma mengekstrak nilai abu-abu dan/atau informasi gradien tepi dari gambar templat. Algoritma mengatur dan menyimpan informasi dengan cara yang memfasilitasi pencarian lebih cepat pada gambar inspeksi. Informasi yang dipelajari selama tahap ini disimpan sebagai bagian dari gambar templat. Selama tahap pencocokan, algoritma *pattern matching* mengekstraksi nilai abu-abu dan/atau informasi gradien tepi dari gambar inspeksi (sesuai dengan informasi yang dipelajari dari templat). Kemudian, algoritma menemukan kecocokan dengan menemukan wilayah pada gambar inspeksi yang memiliki korelasi silang tertinggi (National Instruments, 2021). Contoh implementasi *pattern matching* ditunjukkan pada Gambar 2.7.



Gambar 2.7 *Pattern Matching*

2.3.2. iMAQ Count Objects 2

iMAQ Count Objects 2 digunakan untuk menemukan, menghitung, dan mengukur objek di area pencarian persegi panjang (Sarkar et al., 2014). iMAQ Count Objects 2 menggunakan ambang batas intensitas piksel untuk mengelompokkan objek dari latar belakangnya (National Instruments, 2021).

2.4 NI myRIO 1900



Gambar 2.8 NI myRIO 1900

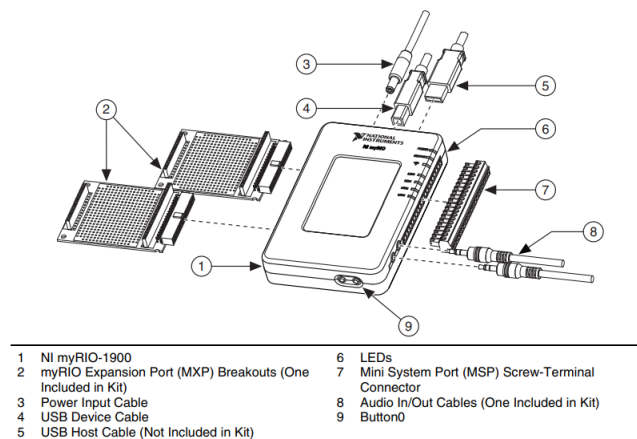
Sumber : (Maulana, 2018)

Gambar 2.8 menunjukkan bentuk fisik NI myRIO 1900. NI myRIO 1900 merupakan sebuah *hardware* yang dapat memanipulasi fungsi-fungsinya untuk membuat berbagai macam sistem. NI myRIO menggunakan *processor* ARM yang merupakan sebuah *processor Field Programmable Gate Arrays* (FPGAs) sehingga dengan memanfaatkan fitur-fitur yang tersedia, sistem yang dibuat dapat menjadi lebih kompleks (Maulana, 2018).

LabVIEW digunakan sebagai IDE pada myRIO 1900. LabVIEW digunakan untuk membuat aplikasi yang akan menjalankan fitur-fitur yang terdapat pada myRIO. Pada *board* myRIO terdapat beberapa instrumen umum yang dapat digunakan seperti WiFi, bluetooth, accelerometer, LED, *push button*, *analog input* dan *analog output*, serta RAM. NI myRIO juga dapat dikoneksikan ke sebuah IC

untuk membuat sistem yang lebih kompleks atau dikoneksikan ke papan lain yang memang diperlukan dalam pembuatan sistem (Maulana, 2018).

Salah satu jenis myRIO yang sering digunakan adalah myRIO-1900. National Instruments (NI) myRIO-1900 merupakan perangkat I/O (RIO) portabel yang dapat dikonfigurasi ulang yang dapat digunakan siswa untuk merancang sistem kontrol, robotika, dan mekatronik (National Instruments, 2018). NI myRIO-1900 terdiri dari modul nirkabel *built-in*, konektor USB, LED, jack audio, pin konektor input dan *output* dan akselerometer tiga sumbu. Untuk menyalakan perangkat myRIO-1900, diperlukan 6 hingga 16 VDC (Ng et al., 2018). Sedangkan untuk keluarannya, myRIO memiliki output daya 3,3V, 5V, dan 15/-15V yang dapat dikoneksikan ke perangkat dan aplikasi yang kita buat sendiri (Lindstål & Marklund, 2019). Untuk detail lengkap dari myRIO-1900 ditunjukkan pada Gambar 2.9 dan untuk blok diagramnya ditunjukkan pada Gambar 2.10.

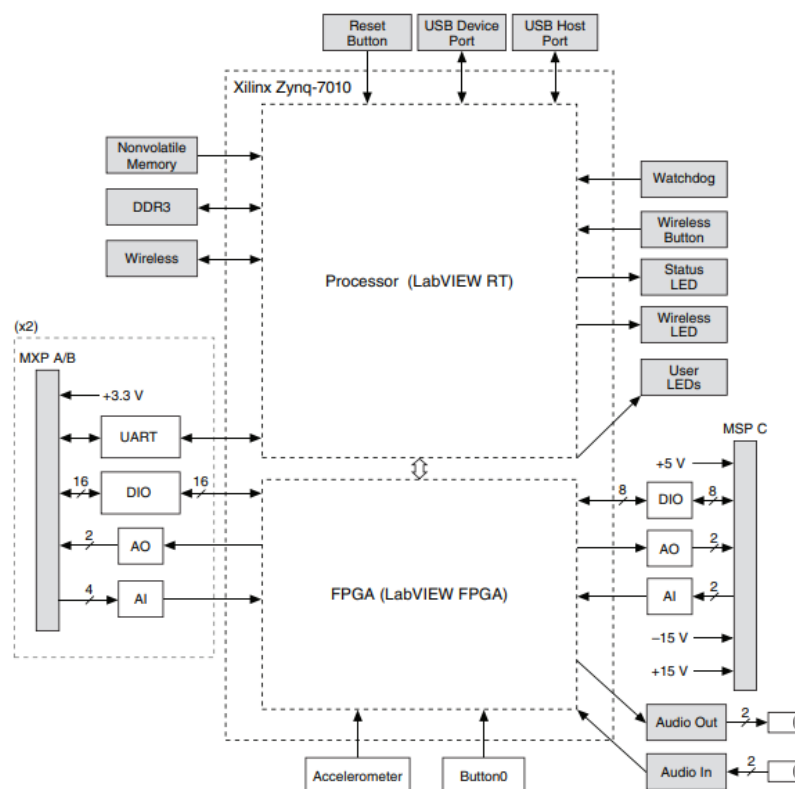


Gambar 2.9 Detail myRIO-1900

Sumber : (National Instruments, 2018)

Adapun penjelasan perangkat myRIO-1900 pada gambar 2.9 yaitu myRIO *Expansion Port* (MXP) merupakan pin pin berisi analog *output*, kemudian *Power*

Input Cable berfungsi sebagai input kabel power, *USB Device Cable* berfungsi sebagai *port* USB untuk mengkoneksikan myRIO dengan komputer, *USB Host Cable* sebagai *port* USB yang dapat digunakan sebagai memori maupun sebagai masukan sensor, kemudian LEDs berfungsi sebagai sinyal keluaran yang dapat di program, *Mini System Port* (MSP) berisi pin-pin analog dan digital, *Audio In/Out Cable* berfungsi sebagai *port* audio, dan Button0 yaitu input boolean yang dapat diberi nilai true atau false.



Gambar 2.10 Blok Diagram myRIO-1900

Sumber : (National Instruments, 2018)

Untuk spesifikasi lengkap dari myRIO-1900 yaitu sebagai berikut.

1. *Processor* : Xilinx Z-7010 667 MHz (ARM Cortex A9 x2 cores 28 nm process NEON SIMD, VFPv3 Vector Float).

2. Memori : *Nonvolatile* (NV), 256 MB, DDR3 512MB, 533 MHz, 16 bits.
3. FPGA : Xilinx Z-7010.
4. *Wireless*: IEEE 802.11 b,g,n ISM 2.4 GHz 20 MHz.
5. USB 2.0 *Hi-Speed*.
6. *Breakout Board Support*.
7. 2 *ports* of 16 *Digital I/O lines*.
8. Akselerometer 3 sumbu.
9. Konsumsi daya maksimum : 14 W.
10. Kondisi *idle*: 2.6 W.
11. LED's.

2.5 Web Camera Logitech C270



Gambar 2.11 *Web Camera* Logitech C270

Sumber : (Logitech, 2023)

Web Camera adalah *device* yang berupa sebuah kamera digital yang dapat dikoneksikan ke desktop melalui *port* USB ataupun *port* COM (Alrynto, 2018). Seperti kamera pada umumnya, sebuah *web camera* dapat mentransmisikan gambar-gambar secara langsung ke seluruh dunia dengan bantuan internet dari manapun kita berada. *Web Camera* bekerja dengan cara menangkap cahaya melalui lensa berukuran kecil di bagian depan dengan bantuan detektor cahaya mikroskopik

yang terpasang pada microchip di bagian penerima gambar dan pada umumnya berteknologi *Charge-Couple Device* (CCD) atau *CMOS image sensor* (M. A. A. Saputra et al., 2021).

Webcam Logitech C270 seperti ditunjukkan pada Gambar 2.11 yaitu salah satu jenis *Web Camera* yang dapat digunakan untuk akuisisi citra dalam bentuk video secara *real time*. Webcam ini memiliki konektor USB 2.0 *type A* sehingga dapat digunakan pada NI myRIO. Selain itu, webcam ini juga memiliki fitur *Automatic light Correction (RightLight 2)* yang dapat menyesuaikan dengan kondisi pencahayaan untuk menghasilkan gambar yang lebih cerah dan lebih kontras (Logitech, 2023). Spesifikasi lengkap dari webcam Logitech C270 ditunjukkan pada Tabel 2.1.

Tabel 2.1 Spesifikasi *Web Camera* Logitech C270

Spesifikasi	:	Keterangan
Kamera mega pixel	:	0,9
Resolusi maksimum	:	720p/30fps
Jenis fokus	:	<i>Fixed Focus</i>
Bidang pandang diagonal (dFoV)	:	55°
Jenis lensa	:	Plastik
Mikrofon internal	:	Mono
Jangkauan mikrofon	:	Maksimal 1 m
Panjang kabel	:	1,5 m
Tinggi	:	72,91 mm
Lebar	:	31,91 mm
Tebal	:	66,64 m
Berat	:	75 gr

Dari Tabel 2.1 dapat dilihat bahwa Webcam Logitech C270 memiliki dFoV sebesar 55° dengan rasio aspek *default* yaitu 4:3. Untuk mengetahui bidang pandang horizontal dan vertikal, maka dapat dihitung dari dFoV dan rasio aspek kamera. Perhitungan ini penting karena digunakan untuk menentukan seberapa luas area yang dapat ditangkap oleh webcam. Berdasarkan (Muslikhin et al., 2020) persamaan untuk menghitung *Horizontal Field of View* (hFoV) dan *Vertical Field of View* (vFoV) secara matematis ditulis sebagai berikut.

Persamaan *Horizontal Field of View* (hFoV):

$$\text{hFoV} = 2 \times \text{atan} \left(\tan \left(\frac{\text{dFoV}}{2} \right) \times \cos \left(\text{atan} \left(\frac{h}{w} \right) \right) \right) \quad (2.1)$$

Persamaan *Vertical Field of View* (vFoV):

$$\text{vFoV} = 2 \times \text{atan} \left(\tan \left(\frac{\text{dFoV}}{2} \right) \times \sin \left(\text{atan} \left(\frac{h}{w} \right) \right) \right) \quad (2.2)$$

2.6 LabVIEW



Gambar 2.12 LabVIEW

LabVIEW (*Laboratory Virtual Instrumentation Engineering Workbench*) seperti terlihat pada Gambar 2.12 merupakan *software* desktop yang digunakan untuk pemrosesan data dalam bidang akuisisi data serta merupakan kontrol instrumentasi dan otomasi industri. LabVIEW menggunakan bahasa pemrograman yang dapat menginterpretasikan data dengan sebuah grafis sebagai suatu fungsi yang disebut bahasa G. LabVIEW disebut juga sebagai *Virtual Instrument* (VI)

karena terdapat beberapa *display* dan operasi pada program LabVIEW yang menyerupai instrumen seperti multimeter dan osiloskop (Maulana, 2018).

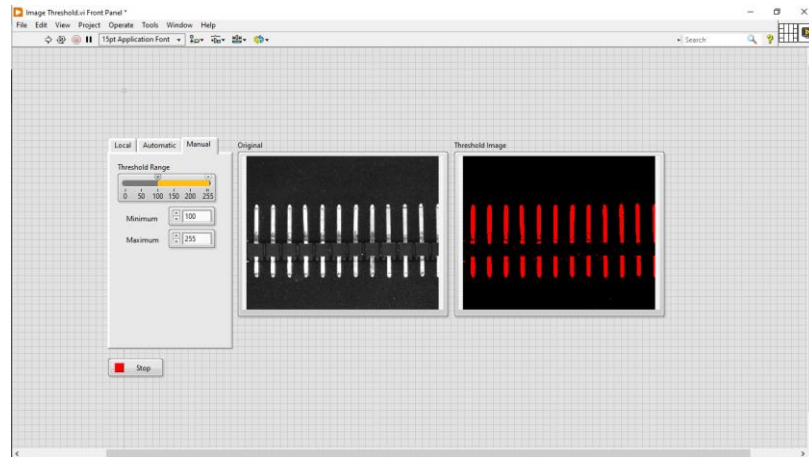
LabVIEW menggunakan bahasa pemrograman grafis yang berarti pada saat pembuatan program tidak menggunakan baris teks melainkan menggunakan ikon. Berbeda dengan bahasa pemrograman berbasis teks yang urutan eksekusinya ditentukan oleh instruksi, LabVIEW menggunakan *data flow programming* sehingga kode dan fungsi ditulis dalam diagram blok milik *Virtual Instrument (VI)* dan aliran datanya melewati node yang menentukan urutan eksekusi. Dengan kata lain, sebuah node akan mengeksekusi ketika sistem telah menerima semua inputannya dan ketika dieksekusi menghasilkan data output yang diteruskan ke node berikutnya dalam bentuk aliran data (Lindstål & Marklund, 2019).

Namun, LabVIEW lebih dari sekedar bahasa pemrograman. LabVIEW merupakan sistem pengembangan dan eksekusi program interaktif yang dirancang untuk orang-orang seperti ilmuwan dan insinyur yang memerlukan pemrograman dalam pekerjaannya. LabVIEW bekerja pada komputer yang menjalankan sistem operasi Windows, Mac OS X, atau Linux. Selain itu, LabVIEW juga dapat membuat program yang berjalan pada platform seperti Microsoft Windows CE, Microsoft Pocket PC, Palm OS, dan berbagai platform lainnya, termasuk FPGAs, *Digital Signal Processors (DSPs)*, dan mikroprosesor (Travis & Kring, 2007).

Terdapat 3 desain *interface* yang digunakan oleh LabVIEW diantaranya:

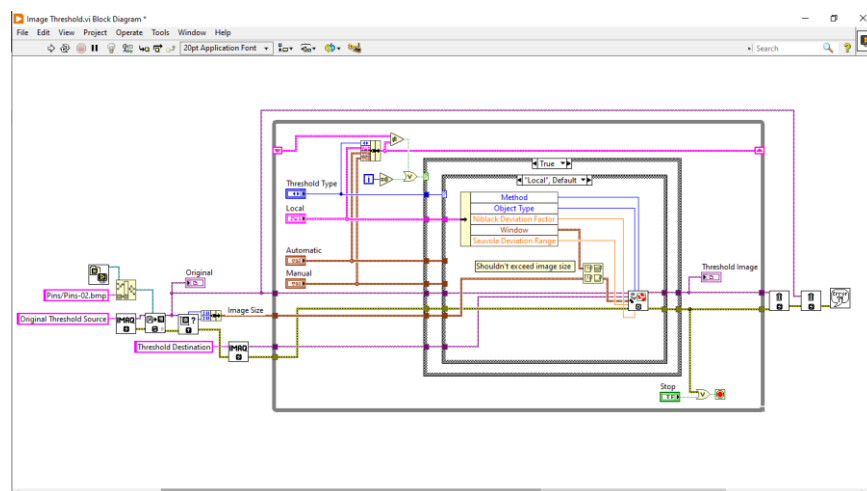
1. *Front panel* merupakan *user interface* interaktif dari VI, dinamai demikian karena *front panel* mensimulasikan instrumen fisik. *Front panel* dapat berisi knop, *push button*, grafik, dan banyak kontrol lainnya (yang merupakan

input pengguna) dan indikator (yang merupakan *output* program). Tampilan *Front Panel* dapat dilihat pada Gambar 2.13.



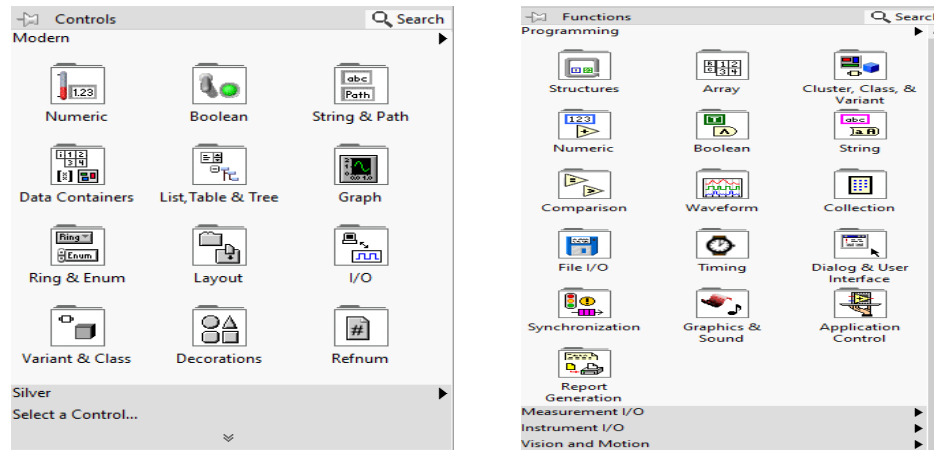
Gambar 2.13 *Front Panel* LabVIEW

2. *Block Diagram* adalah *source code* VI yang dibuat dalam bahasa pemrograman grafis LabVIEW, G. *Block diagram* merupakan desain *interface* yang berisi grafis fungsi-fungsi matematis yang digunakan dalam program LabVIEW. Grafis fungsi yang tersedia akan dihubungkan untuk membuat program yang kita inginkan. Tampilan *Block Diagram* dapat dilihat pada Gambar 2.14.



Gambar 2.14 *Block Diagram* LabVIEW

3. *Control and Functions Palettes* pada *functions* dan *controls* palettes merupakan grafis fungsi yang dimiliki oleh LabVIEW yang dapat digunakan pada *front panel* maupun pada *block diagram*. Tampilan *Control and Functions Palettes* dapat dilihat pada Gambar 2.15.



(a)

(b)

Gambar 2.15 (a) *Control Palettes* (b) *Functions Palettes*

2.7 PERMENHUB RI Nomor PM 49 Tahun 2014

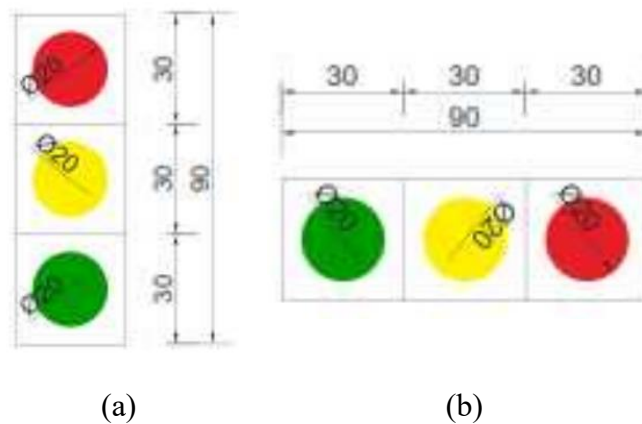
PERMENHUB RI Nomor PM 49 Tahun 2014 digunakan untuk menetapkan kinerja lampu lalu lintas di Indonesia. Menurut BAB I Pasal 2 tentang ketentuan umum, ruang lingkup pengaturan dalam peraturan ini meliputi jenis dan fungsi APILL, spesifikasi teknis APILL, penyelenggaraan APILL, dan pembuatan APILL.

2.7.1 Jenis dan Fungsi APILL

Terdapat tiga jenis APILL yaitu lampu 3 warna, 2 warna, dan lampu 1 warna. Ketiga jenis tersebut dapat berbentuk APILL otonom atau APILL terkoordinasi. APILL otonom dalam pengaturan waktu siklusnya hanya dapat dilakukan oleh APILL yang bersangkutan atau berdiri sendiri. Sedangkan APILL terkoordinasi

dalam pengaturan waktu siklusnya terkoordinasi dan berinteraksi dengan APILL yang dipasang pada lokasi lain.

Fungsi dari APILL adalah untuk mengatur lalu lintas orang dan/atau kendaraan di persimpangan atau pada ruas jalan. Untuk mengatur kendaraan saja, maka dapat menggunakan lampu 3 warna. Lampu 3 warna tersebut terdiri dari warna merah, kuning, dan hijau. APILL dengan lampu 3 warna dapat disusun secara vertikal berurutan dari atas ke bawah berupa lampu berwarna merah, kuning, dan hijau seperti pada Gambar 2.16 (a) atau horizontal berurutan dari sudut pandang pengguna jalan dari kanan ke kiri berupa lampu berwarna merah, kuning, dan hijau seperti pada Gambar 2.16 (b).



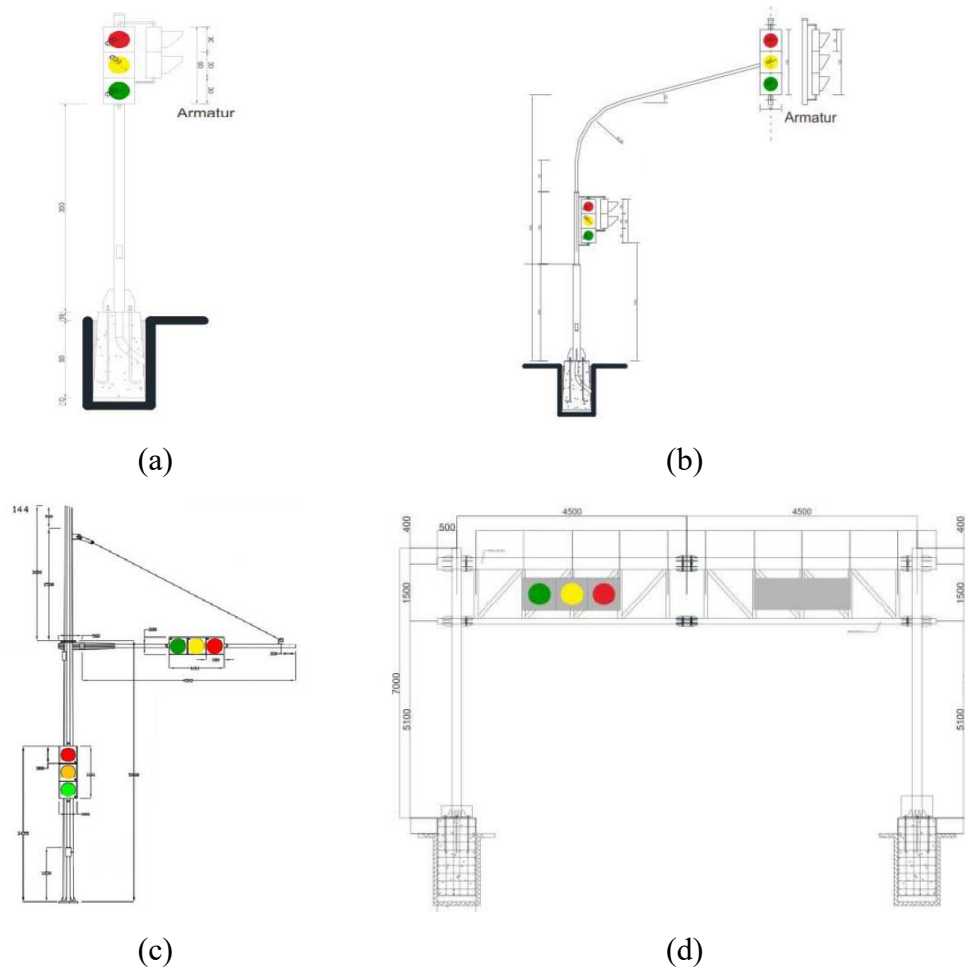
Gambar 2.16 Lampu Tiga Warna Tersusun (a) Vertikal (b) Horizontal

Sumber : (MENHUB RI, 2014)

2.7.2 Spesifikasi Teknis APILL

Terdapat beberapa komponen utama dalam APILL yaitu luminer, tiang penyangga, perangkat kontrol, bangunan konstruksi pondasi, dan kabel instalasi. Untuk tiang penyangga terdapat beberapa jenis yaitu tiang lurus seperti ditunjukkan pada Gambar 2.17 (a), tiang lengkung pada Gambar 2.17 (b), tiang siku pada

Gambar 2.17 (c), dan tiang gawang (*gantry*) pada Gambar 2.17 (d) dengan ketentuan 1 tiang penyangga hanya dapat dipasang paling banyak 3 buah armatur.



Gambar 2.17 APILL Tiang (a) Lurus (b) Lengkung (c) Siku (d) Gawang

Sumber : (MENHUB RI, 2014)

Selain beberapa komponen utama di atas, APILL juga dapat dipasang alat pendeteksi kendaraan, *Display Information System*, kamera, dan/atau peralatan teknologi informasi untuk kepentingan lalu lintas yang telah tersertifikasi sesuai dengan ketentuan perundang-undangan.

2.7.3 Penyelenggaraan APILL

Penyelenggaraan APILL dilakukan oleh DIRJEN untuk jalan nasional, gubernur untuk jalan provinsi, bupati untuk jalan kabupaten dan jalan desa, dan walikota untuk jalan kota.

Dalam hal pemasangannya, APILL dengan lampu 3 warna dapat dipasang pada persimpangan dan ruas jalan. Untuk APILL yang dipasang pada persimpangan, penempatan APILL setidaknya berjarak 60 sentimeter dari bagian terluar armatur ke tepi paling luar bahu jalan, tinggi penempatan armatur minimal 300 sentimeter atau minimal 500 sentimeter jika ditempatkan di atas ruang manfaat jalan yang diukur bagian tertinggi sampai dengan sisi armatur bagian bawah dengan posisi armatur diputar ke kanan atau ke kiri maksimal 5° menghadap permukaan jalan dari posisi tegak lurus sumbu jalan sesuai dengan arah lalu lintas.

2.7.4 Pembuatan APILL

Pembuatan APILL dilakukan oleh badan usaha yang telah memenuhi persyaratan dan telah dinilai oleh Direktur Jenderal. Persyaratan tersebut seperti bahan, peralatan produksi, perlengkapan, sampai sumber daya manusia yang kompeten di bidang perlengkapan jalan. Badan usaha yang telah memenuhi persyaratan tersebut didaftarkan di Direktorat Jenderal Perhubungan Darat sebagai badan usaha pembuat APILL.

2.8 *Confusion Matrix*

Confusion Matrix yaitu sebuah tabel yang memungkinkan visualisasi kinerja suatu algoritma. Setiap baris matriks mewakili kejadian kelas yang diprediksi, sementara setiap kolom menunjukkan kelas yang sebenarnya (Srinivas B. et al.,

2019). Tabel ini menunjukkan bagaimana prediksi model dibandingkan dengan hasil aktual. *Confusion matrix* terdiri dari empat elemen utama diantaranya:

1. *True Positive* (TP): Model memprediksi terdapat kendaraan dan memang benar terdapat kendaraan.
2. *True Negative* (TN): Model memprediksi tidak terdapat kendaraan dan memang benar tidak terdapat kendaraan.
3. *False Positive* (FP): Model memprediksi terdapat kendaraan, tetapi sebenarnya tidak terdapat kendaraan (kesalahan deteksi).
4. *False Negative* (FN): Model memprediksi tidak terdapat kendaraan, tetapi sebenarnya terdapat kendaraan (kesalahan melewatkan deteksi).

Kemudian model dievaluasi dengan menggunakan metrik evaluasi dari *Confusion Matrix* diantaranya:

1. *Accuracy*: Proporsi prediksi yang benar terhadap seluruh prediksi.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.3)$$

2. *Precision*: Dari semua yang diprediksi positif, berapa banyak yang benar-benar positif.

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

3. *Recall/Sensitivity*: Dari semua kasus yang sebenarnya positif, berapa banyak yang berhasil diprediksi dengan benar.

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

4. *F1-Score*: Gabungan dari *precision* dan *recall*, memberikan keseimbangan antara keduanya. Cocok untuk data yang tidak seimbang.

$$F - 1 \text{ Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (2.6)$$

2.9 Penelitian Terkait

Beberapa penelitian terkait sistem kontrol lampu lalu lintas yang telah dilakukan sebelumnya ditunjukkan pada Tabel 2.2.

Tabel 2.2 Penelitian Terkait

No.	Judul	Penulis, Tahun	Pembahasan
1.	Sistem Kontrol Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan <i>Image Processing</i>	(M. A. A. Saputra et al., 2021)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan teknik pengurangan citra yang dibuat dalam bentuk purwarupa dengan dua simpang. Perangkat yang digunakan yaitu dua buah Webcam Logitech C170, dan MATLAB. Hasil penelitian ini menunjukkan akurasi yang cukup baik dengan akurasi 75% untuk arah A dan 100% untuk arah B.
2.	Perancangan Sistem Kontrol Lampu Lalulintas Cerdas Dengan Menggunakan Mikrokontroler dan Kamera	(Pangemanan & Rondonuwu, 2019)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan metode <i>Pixel</i> RGB yang dibuat dalam bentuk purwarupa dengan 4 simpang. <i>Device</i> yang digunakan yaitu 1 buah kamera Pixy CMUCam 5, Arduino Mega 2560, dan PixyMon. Hasil penelitiannya

			menunjukkan bahwa kamera yang digunakan bekerja secara simultan dan sistem yang dibuat lebih <i>smart</i> karena hanya menggunakan 1 kamera untuk 4 simpang.
3.	<i>Smart traffic light control system using image processing</i>	(Meng et al., 2021)	Penelitian ini menggunakan <i>edge detection</i> dan <i>Image matching</i> . <i>Canny edge detection</i> digunakan dari 5 jenis teknik yang dibandingkan karena menjadi yang terbaik karena mampu mengekstrak tepi aktual dengan waktu rata-rata 0,453 detik. Berbentuk simulasi 4 simpang dengan menggunakan MATLAB. Namun, penelitian ini hanya sampai pada tahap menampilkan durasi waktu menyala lampu hijau (tidak terdapat proses kontrol lampu lalu lintas.)
4.	<i>Smart Traffic Light Scheduling in Smart City Using Image and Video Processing</i>	(Razavi et al., 2019)	Penelitian ini menggunakan teknik <i>Edge detection</i> , <i>image matching</i> , dan metode <i>video background removal</i> untuk 4 simpang dengan menggunakan Raspberry Pi, OpenCV, dan VNC Viewer. Hasil penelitian menunjukkan Model 2 dengan pendeteksian berdasarkan kepadatan kendaraan dan jumlah kendaraan memberikan hasil lebih baik dalam menentukan durasi

			waktu menyala lampu hijau. Namun, penelitian ini hanya sampai pada tahap menampilkan durasi waktu menyala lampu hijau (tidak terdapat proses kontrol lampu lalu lintas).
5.	<i>IoT based Smart Traffic density Control using Image Processing</i>	(Frank et al., 2019)	Penelitian ini menggunakan <i>image matching</i> dengan menggunakan <i>Webcam</i> USB, Raspberry Pi 3, aplikasi desktop, dan server M2X <i>IoT Cloud Platform</i> . Sistem ini memiliki dua mode pensinyalan yaitu mode pensinyalan otomatis dan mode pensinyalan manual. Sistem ini menggunakan Wi-Fi untuk mengirimkan data hasil pembacaan kamera dari Raspberry Pi ke M2X <i>IoT Cloud Platform</i> .
6.	Skenario Pengendali Lampu Lalu Lintas Berdasarkan Kepadatan Kendaraan Menggunakan Logika Fuzzy Dan Deteksi Tepi	(D. I. Saputra & Aunillah, 2019)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan <i>Edge Detection</i> tipe prewitt dan logika fuzzy yang dibuat dalam bentuk purwarupa 3 simpang dengan menggunakan MATLAB. Hasil penelitiannya menunjukkan bahwa kedua teknik yang digunakan dapat menyesuaikan lama lampu hijau sesuai kepadatan yang terjadi pada suatu simpang jalan.

7.	Sistem Pengefektifan Pemberian Waktu Lampu Lalu Lintas dengan Memanfaatkan CCTV ATCS (<i>Auto Traffic Control System</i>) dan Metode <i>Background Subtraction</i>	(Darma Sutisna et al., 2021)	Penelitian ini memanfaatkan ATCS lalu menggunakan teknik <i>Background Subtraction</i> untuk kontrol lampu lalu lintas 4 simpang dengan menggunakan MATLAB. Hasil penelitian ini menunjukkan bahwa sistem yang telah dibuat dapat digunakan sebagai alternatif dalam mengatasi kepadatan lalu lintas.
8.	<i>Prototype Smart Time Scheduler</i> Lampu Lalu Lintas Menggunakan Algoritma Haar Cascade	(Purwanda et al., 2023)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan algoritma deteksi objek yaitu Haar Cascade yang dibuat dalam bentuk purwarupa 3 simpang dengan menggunakan Raspberry Pi, Arduino Uno, 1 buah kamera, LCD 16 x 2, OpenCV, dan VNC Viewer. Penelitian ini hanya menguji 1 simpang dari 3 simpang yang ada. Hasil dari penelitian ini memiliki akurasi sebesar 84% dan error pengukuran sebesar 16%.
9.	<i>Integrated Smart Traffic Control System</i> Menuju Pekanbaru Sebagai <i>Smart City</i>	(Taufiq et al., 2020)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan algoritma <i>Quantized MobileNet-Single Shot Multibox</i> yang dibuat dalam bentuk purwarupa 1 simpang.

			Sistemnya diberi nama <i>Integrated Smart Traffic Control System</i> dengan dua fitur yaitu <i>Smart Traffic Control System</i> dan <i>Smart Map</i>. Komponen yang digunakan yaitu IP Camera Hikvision DS-2CD2021G1-I, Raspberry Pi, dan OpenCV dengan <i>framework</i> Tensorflow Lite. Hasil penelitiannya menghasilkan FPS sebesar 4,56 fps yang masih di bawah standar minimum <i>American Public Transportation Association</i> untuk kategori area <i>low traffic</i> yaitu 5 fps sehingga memberatkan sistem yang memiliki <i>resources</i> terbatas.
10.	<i>Density based Traffic Control System using Ni LabVIEW</i>	(Srinivas B. et al., 2019)	Penelitian ini melakukan kontrol lampu lalu lintas menggunakan segmentasi citra dan Algoritma KNN <i>Classification</i> dan teknik <i>Color Classification</i> menggunakan Ni <i>Vision Assistant</i> untuk mendeteksi ambulans. Sistemnya dibuat dalam bentuk purwarupa 4 simpang dengan menggunakan 1 buah kamera, myRIO, dan LabVIEW. Penelitian ini hanya melakukan pengujian pada 1 simpang saja. Hasil dari penelitian ini menunjukkan akurasi 86,66%, sensitivitas 0,866, spesifisitas 0,13, dan tingkat error 13,33%.

Tabel 2.2 menjelaskan tentang beberapa penelitian terkait yang digunakan sebagai referensi untuk melakukan kebaruan pada penelitian ini. Penelitian terkait tersebut memiliki kelebihan dan kekurangannya masing-masing. Oleh karena itu, penelitian ini melakukan integrasi terhadap beberapa penelitian terkait tersebut sehingga menghasilkan kebaruan penelitian yang diharapkan menghasilkan penelitian yang lebih baik daripada penelitian-penelitian sebelumnya.

Penelitian ini melakukan perancangan sistem kontrol lampu lalu lintas 4 simpang dengan teknik *Image Processing* dan *Machine Vision* menggunakan NI myRIO. Teknik *Image Processing* yang digunakan yaitu teknik Segmentasi Citra dengan metode *Thresholding* yang digunakan untuk memisahkan antara latar depan dan latar belakang. Kendaraan merupakan bagian dari latar depan yang akan dihitung menggunakan *Machine Vision* yaitu *iMAQ Count Objects 2*. *Machine Vision* lain yang digunakan yaitu *Pattern Matching* yang digunakan untuk mencari persentase kecocokan antara citra referensi berupa jalanan kosong dengan citra aktual. Tujuannya adalah untuk menentukan kepadatan kendaraan. Semakin tidak cocok citra referensi dengan citra aktual, maka semakin padat kendaraan di jalanan. Sebaliknya, semakin cocok citra referensi dengan citra aktual, maka semakin lengang kendaraan di jalanan.

Semua teknik tersebut diintegrasikan dengan harapan dapat memberikan keputusan pemberian durasi waktu menyala lampu hijau menjadi lebih akurat. Pemberian keputusan menggunakan kepadatan kendaraan dan jumlah kendaraan didasari oleh penelitian terkait poin 4 pada Tabel 2.2 yang dijadikan acuan untuk mengembangkan sistem ini. Penggunaan teknik segmentasi citra, *iMAQ Count*

Objects 2, dan *pattern matching* dipilih dengan mempertimbangkan efisiensi waktu pemrosesan dan juga agar tidak terlalu memberatkan myRIO sebagai pemroses citra video secara *real time*. NI myRIO digunakan karena kemampuannya yang dapat melakukan proses secara *real time* berkat *hardware* yang tertanam di dalamnya. *Output* dari sistem yang dibuat pada penelitian ini ditampilkan dalam bentuk *Graphical User Interface* (GUI) pada *software* LabVIEW.

Selain itu, simulasi dan purwarupa dengan 4 simpang dan melakukan kontrol lampu lalu lintas ke semua simpang juga merupakan kebaruan dari penelitian ini, karena pada penelitian terkait pada Tabel 2.2 terdapat kekurangan yaitu untuk penelitian dalam bentuk simulasi hanya melakukan pengujian deteksi kendaraan pada keempat simpang (tidak terdapat kontrol lampu lalu lintas). Selain itu, penelitian ini juga melakukan pengujian webcam di lapangan menggunakan sudut 45°, 60°, dan 90° pada kondisi pagi hari di mana orang-orang berangkat untuk bekerja (waktu sibuk) dengan menggunakan tiang dengan ketinggian yaitu 5 meter yang mirip dengan spesifikasi yang tercantum pada PERMENHUB RI Nomor PM 49 Tahun 2014.