

BAB II

TINJAUAN PUSTAKA

Dalam bab tinjauan pustaka ini akan dijelaskan beberapa informasi terkait dengan istilah dan ilmu yang terkait.

II.1 Tinjauan Teori

II.1.1 LiDAR

LiDAR (*Light Detection and Ranging*) adalah teknologi yang menggunakan pulsa laser untuk penentuan posisi secara 3D yang hampir mendekati *realtime* [10]. Pada bidang transportasi, meskipun telah digunakan untuk membantu beberapa pekerjaan desain dan perencanaan, penerapannya sebagian besar berfokus pada kendaraan otonom, terlepas dari potensi besarnya dalam deteksi dan pelacakan semua pengguna jalan secara tepat jika diterapkan di lapangan [11].

II.1.2 Point Cloud

Point Cloud adalah sekumpulan titik yang didefinisikan dalam ruang metrik 3D. *Point Cloud* telah menjadi salah satu format data paling signifikan untuk representasi 3D dan semakin populer karena meningkatnya ketersediaan perangkat akuisisi, serta meningkatnya penerapan di berbagai bidang seperti robotika, mengemudi otonom, dan *augmented reality* dan *virtual reality* [12]. *Point Cloud Library* (PCL) adalah perpustakaan sumber terbuka yang menyediakan kumpulan alat dan fungsi untuk pemrosesan awan titik 3D dengan tujuan untuk mendukung aplikasi agar efisien, termasuk pemodelan 3D, pemetaan, dan analisis geometris. PCL juga mendukung berbagai jenis sensor termasuk LiDAR untuk kalibrasi sensor agar dapat memastikan akurasi dan konsistensi dari sensor LiDAR.

II.1.3 Komunikasi Data

Komunikasi data adalah pengiriman data dari satu komputer ke komputer lainnya atau antara komputer dan piranti lainnya. Data dalam bentuk digital dikirim melalui media komunikasi data. Diterapkan dalam berbagai bentuk aplikasi

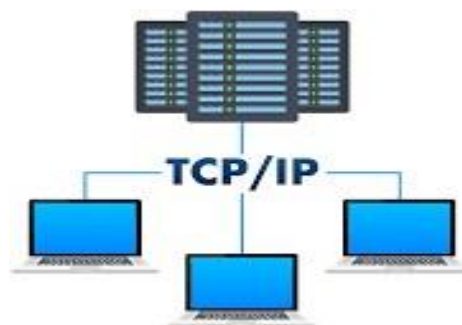
komunikasi data termasuk menghubungkan komputer atau perangkat lainnya yang dikenal dengan internet. [13].

II.1.4 Vehicle Type Recognition

Vehicle Type Recognition (VTR) adalah untuk mengkategorikan mobil ke dalam kategori yang lebih luas seperti van, mini-van, truk, sedan, bus, taksi, dan lainnya. Merek dan model karya VTR pastinya tidak diketahui. Penerapan seperti gerbang tol elektronik dan studi dan analisis lalu lintas dibantu oleh sistem VTR otomatis. Multiplikasi dalam konteks VTR mengacu pada masalah jenis kendaraan dengan berbagai bentuk dan ukuran [2].

II.1.5 TCP/IP

TCP/IP—singkatan dari Transmission Control Protocol/Internet Protocol—adalah salah satu perangkat lunak jaringan komputer yang termasuk dalam sistem. Digunakan untuk berkomunikasi data baik pada local area network (LAN) maupun internet, dan hampir semua perangkat keras dan sistem operasi menggunakan protokol ini. Dengan menggunakan protokol ini, sejumlah komputer dapat berinteraksi satu sama lain melalui protokol pengalaman sederhana yang dikenal sebagai alamat IP [14].



Gambar II.1 Network Video Recorder

II.2 Tinjauan Alat

II.2.1 Router

Router adalah salah satu perangkat keras jaringan komputer, yang berfungsi untuk membuat dan mengimplementasikan jaringan komputer yang luas. Dengan kehadiran router, jaringan dapat membagi dan memilih jalur mana yang harus dilewati oleh paket data, sehingga paket data dapat sampai ke penerima (komputer klien) [15].



Gambar II.2 Router

II.2.2 Network Video Recorder

Network Video Recorder (NVR) adalah perangkat elektronik yang digunakan untuk merekam, menyimpan, dan mengelola video dari kamera keamanan yang terhubung melalui jaringan. NVR berfungsi sebagai pusat pengendali untuk sistem pengawasan video berbasis IP [16] untuk merekam, menyimpan, dan mengelola data video yang diterima dari kamera keamanan yang terhubung melalui jaringan. NVR menggunakan protokol jaringan seperti TCP/IP untuk berkomunikasi dengan kamera, dan seringkali mendukung berbagai fitur termasuk manajemen penyimpanan, pemantauan jarak jauh, deteksi gerakan, dan analisis video.



Gambar II.3 Network Video Recorder

II.2.3 EGP2-X401 | M2 Serial RS-485 dan RS-232

EGP2-X401 | M2 serial RS-485 adalah modul komunikasi yang mendukung protokol serial RS-485 yang sering digunakan untuk mentransmisikan data serial antar perangkat elektronik dan juga sebagai modul komunikasi dengan perangkat mikrokontroler dengan serial RS-485 yang akan di konversi menjadi RS232 agar sinyal yang dikirimkan menjadi lebih sederhana.



Gambar II.4 EGP2-X401 | M2 serial RS-485

II.2.4 ATMEGA328

Mikrokontroler ATmega328 dikembangkan oleh Atmel, yang sekarang merupakan bagian dari Microchip Technology. Ini adalah salah satu keluarga mikrokontroler AVR (Advanced Virtual RISC) yang sangat disukai dan sering digunakan dalam berbagai proyek elektronik. Ini didasarkan pada arsitektur AVR, yang merupakan arsitektur mikrokontroler yang menggunakan set instruksi RISC (Reduced Instruction Set Computing), yang memungkinkan eksekusi instruksi yang lebih cepat dan efisien [17]. Untuk membuat program untuk ATmega328 dapat menggunakan berbagai lingkungan pengembangan, seperti Arduino IDE. Bahasa pemrograman yang paling umum adalah C atau C++. ATMEGA328 akan diprogram hanya untuk *reset* sistem apabila terjadi *error* pada sistem dimana dari lima sample data harus terdapat tiga sample data

error atau mati kepada *modbus protocol* dan akan mengirimkan sinyal yang diterima modbus dari hardware ke ATMEGA328 untuk melakukan reset sistem.



Gambar II.5 ATMEGA328

II.2.5 Wafer TGL U

Wafer TGL U merupakan *embedded board* pada industri 3.5” yang memiliki performa kinerja yang tinggi dan ideal digunakan untuk keperluan otomasi industry, analisis data, pengawasan video, dan aplikasi pencitraan [18]. Selain itu, Mini PC PCIE ini digunakan untuk menghubungkan perangkat keras termasuk *hardisk*, kartu grafis, kartu suara, serta penyimpanan seperti SSD. PCIE menawarkan kecepatan pengiriman data yang tinggi dan digunakan secara luar dalam dunia komputasi modern.



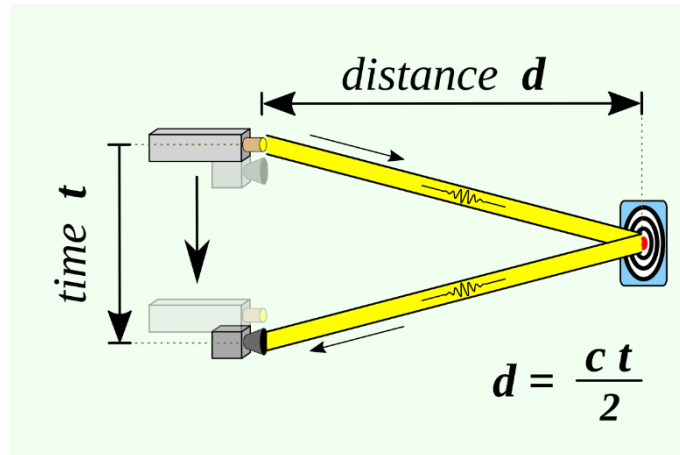
Gambar II.6 Mini PC PCIE Wafer TGL U [18]

II.2.6 LiDAR

LiDAR menghasilkan objek 3D dengan cara mengirimkan *pulse laser* yang dipantulkan kembali dari objek di sekitarnya dengan persamaan sebagai berikut:

$$[d = \frac{c.t}{2}] \dots \dots \dots [1]$$

Dengan prinsip kerja pendeteksian dengan menggunakan LiDAR sebagai berikut:



Gambar II.7 Prinsip Kerja LiDAR

Dimana bagian laser pada LiDAR akan menembakkan gelombang elektromagnetik berupa sinar laser untuk mengukur jarak ke objek dan akan diterima oleh objek yang nantinya akan dipantulkan oleh objek tersebut untuk diterima kembali oleh receiver, yang artinya objek berfungsi sebagai reflector atau pemantul yang nantinya data akan diterima oleh receiver atau dapat disebut juga *cycling* yang berarti sinar laser akan memancar apabila terdeteksi objek saja. Variable d merupakan jarak antara LiDAR dengan objek yang dipantulkan dalam satuan meter, variable c merupakan kecepatan cahaya dalam vakum, variable t merupakan waktu yang diperlukan cahaya laser dari LiDAR menuju objek dan dipantulkan kembali. Dengan mengukur waktu yang diperlukan untuk pantulan tersebut kembali ke sensor, LiDAR dapat menghitung jarak ke objek dengan sangat akurat. Lalu, apabila nilai d sudah diketahui, maka perhitungan untuk pendeteksian selanjutnya yaitu terdapat sudut dan koordinat dimulai dengan konversi sudut elevasi dan azimuth kedalam bentuk radian agar memiliki satuan yang konsisten dan juga memudahkan pemrograman dan pemrosesan data untuk memberikan hasil yang presisi dan memudahkan integrasi data dengan algoritma perangkat lunak, dengan persamaan sebagai berikut:

$$\left[\theta_{rad} = \frac{\theta_{deg} \times \pi}{180} \right] \dots \dots \dots [2]$$

Dimana variable θ_{deg} merupakan sudut elevasi dalam derajat, sedangkan variable π (phi) adalah konstanta matematika yang mendekati 3.14159. Sedangkan azimuth dengan persamaan sebagai berikut:

$$\left[\phi_{rad} = \frac{\phi_{deg} \times \pi}{180} \right] \dots \dots \dots [3]$$

Dimana variable ϕ_{deg} merupakan sudut azimuth atau sudut putar dalam derajat, dan variable π (phi) adalah konstanta matematika yang mendekati 3.14159. Lalu, dilanjutkan dengan menghitung nilai kosinus dan sinus untuk mendapatkan koordinat kartesian dengan persamaan sebagai berikut:

$$[\cos(\theta) = \cos(\theta_{rad})] \dots \dots \dots [4]$$

$$[\sin(\theta) = \sin(\theta_{rad})] \dots \dots \dots [5]$$

$$[\cos(\phi) = \cos(\phi_{rad})] \dots \dots \dots [6]$$

$$[\sin(\phi) = \sin(\phi_{rad})] \dots \dots \dots [7]$$

Dimana masing-masing sudut yang sudah dikonversi baik itu sudut elevasi ataupun azimuth dicari nilai kosinus dan juga sinus untuk mencari hasil kartesian yang nilai kosinus dan sin didapat dari hasil konversi *degree* ke *radiant*. Setelah nilai kosinus dan sin didapat, maka mencari nilai koordinat kartesian (x, y, dan z) dari objek, dengan persamaan sebagai berikut:

$$[x = d \cdot \cos(\theta) \cdot \cos(\phi)] \dots \dots \dots [8]$$

$$[y = d \cdot \cos(\theta) \cdot \sin(\phi)] \dots \dots \dots [9]$$

$$[z = d \cdot \sin(\theta)] \dots \dots \dots [10]$$

Variable d merupakan jarak antara LiDAR dengan objek dan nilai kosinus dan sinus dari hasil perhitungan sebelumnya. Setelah koordinat kartesian sudah didapatkan nilainya, maka langkah selanjutnya yaitu mencari kalkulasi sudut elevasi (θ), dengan persamaan sebagai berikut:

$$\left[\theta = \arcsin\left(\frac{z}{d}\right) \right] \dots \dots \dots [11]$$

Untuk mencari sudut elevasi digunakan arcsin yaitu nilai *inverse* dari sinus yang sudah diberikan atau didapatkan. Dimana z merupakan koordinat kartesian pada sumbu z dan d merupakan jarak antara LiDAR dan objek yang dideteksi. Dan kalkulasi sudut azimuth (ϕ) dengan persamaan sebagai berikut:

$$[\phi = \arctan 2(y, x)] \dots \dots \dots [12]$$

Variable y, x didapat dari koordinat kartesian yang sebelumnya sudah didapatkan nilainya, secara garis besar kalkulasi sudut elevasi dan kartesian ini hanya menguji konsistensi dari nilai sudut elevasi dan azimuth yang digunakan sebelumnya agar hasil perhitungan menjadi semakin akurat. Selain itu, dengan memindai laser secara

horizontal dan vertical, LiDAR dapat memberikan informasi tentang ukuran dan kecepatan kendaraan. Untuk persamaan kecepatan sebagai berikut:

$$[v = \frac{d}{t}] \dots\dots\dots[13]$$

Variable v merupakan *velocity* atau kecepatan kendaraan, variable d adalah jarak kendaraan terdeteksi oleh sensor hingga melewati sensor atau terakhir kali terbaca oleh sensor. Selanjutnya variable t adalah waktu yang ditempuh kendaraan sedari awal terdeteksi oleh sensor hingga melewati sensor sehingga kecepatan dapat didapat berdasarkan jarak dan waktu.

II.2.7 Point Cloud Library

Point Cloud Library merupakan pustaka yang digunakan untuk pemrosesan Point Cloud terlebih untuk komputer vision. PCL digunakan untuk *tracking object in real time* dan mendeteksi cluster dengan menggunakan *region growing segmentation* dan menghasilkan gambar 2D atau 3D.

II.2.7.1 Tracking Object In Real Time

PCL menyediakan fitur atau algoritma untuk membantu melacak sebuah objek atau *cluster*, salah satunya yaitu fitur atau algoritma *Tracking Object In Real Time* ini yang mana dapat digunakan untuk mempercepat dan meningkatkan akurasi *tracking*. Untuk melakukan pelacakan objek atau *cluster* PCL menggunakan *particle filter algorithm* dengan langkah-langkah seperti berikut:

Prediksi

Langkah prediksi ini dilakukan untuk menggerakkan setiap partikel yang sesuai dengan dinamika objek atau *cluster* setelah dilakukan inisiasi partikel atau inisiasi untuk menghasilkan partikel acak yang mewakili kecepatan objek dan perkiraan posisi, dengan permodelan matematika sebagai berikut:

$$X_{i,k} = f(X_{i,k-1}, u_k) + w_k$$

Dimana :

$X_{i,k}$ = posisi partikel i pada waktu k

F = fungsi dinamika

W_k = noise proses

Resampling

Proses ini merupakan proses untuk menghitung bobot untuk setiap partikel setelah proses prediksi atau berdasarkan kesesuaian ukuran untuk memilih partikel dengan bobot tinggi yang membentuk partikel baru, dengan persamaan sebagai berikut:

$$w_{i,k} = p(z_k | X_{i,k})$$

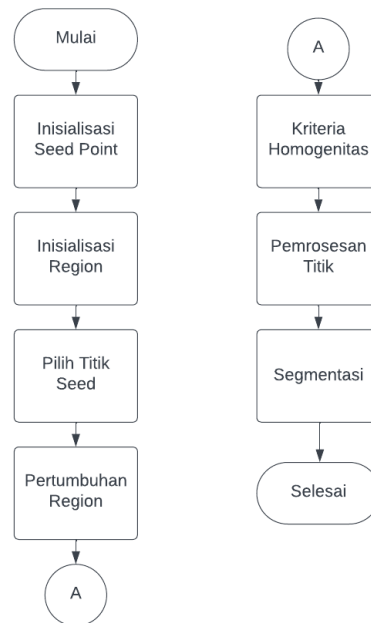
Dimana :

$W_{i,k}$ = bobot partikel i pada waktu k

$P(z_k | X_{i,k})$ = probabilitas pengukuran z_k pada posisi partikel $X_{i,k}$

II.2.7.2 Region Growing Segmentation

Merupakan metode untuk segmentasi data *point cloud* dan citra dengan cara mengelompokkan titik berdasarkan kriteria secara efektif. Alur proses algoritma *region growing segmentation* seperti pada diagram berikut:



Gambar II.8 Alur Proses Algoritma Region Growing Segmentation

Pada alur proses tahapan berdasarkan Gambar II.8, dimulai dengan proses inisialisasi *seed point* yang merupakan metode untuk memilih *seed* secara acak ataupun *region* tertentu dan untuk menentukan kriteria homogenitas atau kriteria yang sejenis untuk dilakukan segmentasi. Tahapan kedua yaitu inisialisasi region untuk menentukan *seed point* akan masuk kedalam region mana sesuai dengan kriteria yang didapatkan, kriteria homogenitas disini terkait *axle* kendaraan yang dikategorikan berdasarkan jumlah *axle* kendaraan tersebut. Lalu tahapan

selanjutnya yaitu pertumbuhan region atau pengolahan data pada region dengan menggunakan struktur data untuk menentukan apakah data memenuhi kriteria homogenitas, tahapan ini dibagi menjadi tiga tahapan yaitu *initial seed selection*, *neighbor search*, dan *region expansion*. Lalu untuk kriteria homogenitas didapatkan dengan beberapa metode, yaitu:

Distance-Based

$$\text{If } \|p_i - p_j\| \leq \epsilon$$

Dimana :

$\|p_i - p_j\|$ = jarak euclidean antara titik i dan j

ϵ = threshold jarak yang diizinkan

Normal-Based

$$\text{If } \text{angle}(n_i, n_j) \leq \theta$$

Dimana :

$\text{angle}(n_i, n_j)$ = sudut antara titik i dan j

θ = threshold sudut

Color-Based

$$\text{If } \|c_i, c_j\| \leq \delta$$

Dimana :

$\|c_i, c_j\|$ = jarak warna antaran titik i dan j

Δ = threshold warna

II.3 Studi Penelitian Terdahulu

Berdasarkan hasil studi yang dilakukan terhadap beberapa penelitian terdahulu, diperoleh berbagai informasi mengenai sistem yang telah dibuat. Berikut adalah hasil studi terdahulu pada Tabel II-1.

Tabel II-1 Penelitian terdahulu

No	Judul Penelitian	Nama Peneliti	Hasil Penelitian/Kesimpulan
1	<i>Vehicle detection based on point cloud intensity and distance clustering</i> [19].	Zhao Wang, Xing Wang, Bin Fang, Kun Yu, Jie Ma (tahun 2021).	➤ Dengan akurasi tinggi, metode kami dapat secara efisien menemukan dan mengklasifikasikan target dalam point cloud 3D yang tidak

			teratur berskala besar. Metode yang kami usulkan meningkatkan akurasi deteksi dalam kumpulan data Livox Mid-40 Lidar yang dipindai secara nyata sebesar 31% dibandingkan dengan metode pengelompokan Euclidean konvensional.
2	<i>Automatic Vehicle LicensePlate Recognition Using Optimal DeepLearning Model</i> [20].	Thavavel Vaiyapuri, Sachi Nandan Mohanty, M. Shivaram, Irina V. Pustokhina, Denis A. Pustokhin, and K. Shankar (tahun 2020).	➤ Hasil eksperimen memverifikasi bahwa metode SSA disajikan lebih baik dalam beberapa aspek dengan akurasi keseluruhan optimal sebesar 0,983%.
3	<i>Lidar Based Vehicle Classification</i> [21].	Benjamin Coifman, H. Lee (tahun 2015).	➤ Algoritma ini memeriksa setiap cluster kendaraan untuk mengetahui apakah ada bukti oklusi parsial dari kendaraan lain. Algoritma ini dengan tepat mengklasifikasikan lebih dari 99,5% kendaraan yang tersisa.
4	<i>Efficient Point Cloud Pre-processing using The Point Cloud Library</i> [22].	Marius Miknis, Ross Davies, Peter Plassmann & Andrew Ware (tahun 2015).	➤ Untuk mengoptimalkan komponen rantai pra-pemrosesan yang memiliki kinerja yang diukur paling lambat, teknik optimasi yang didasarkan pada pengukuran

			<p>siklus CPU diusulkan dan diterapkan. Hasilnya menunjukkan bahwa kinerja rantai pra-pemrosesan telah meningkat 69 kali lipat dengan modul yang dioptimalkan.</p>
5	<p><i>A System Architecture of a Fusion System for Multiple LiDARs Image Processing</i>[23].</p>	<p>Minwoo Jung, Dae Young Kim, Seokhoon Kim (Tahun 2022).</p>	<p>➤ Pendekatan dilakukan dengan multiprosesor yang efisien, dan untuk komunikasi internal dalam sistem fungsi dihubungkan di dalam satu server dengan metode soket. Dalam pemrosesan sekuensial, waktu respons meningkat dengan jumlah sensor LiDAR yang terhubung.</p>