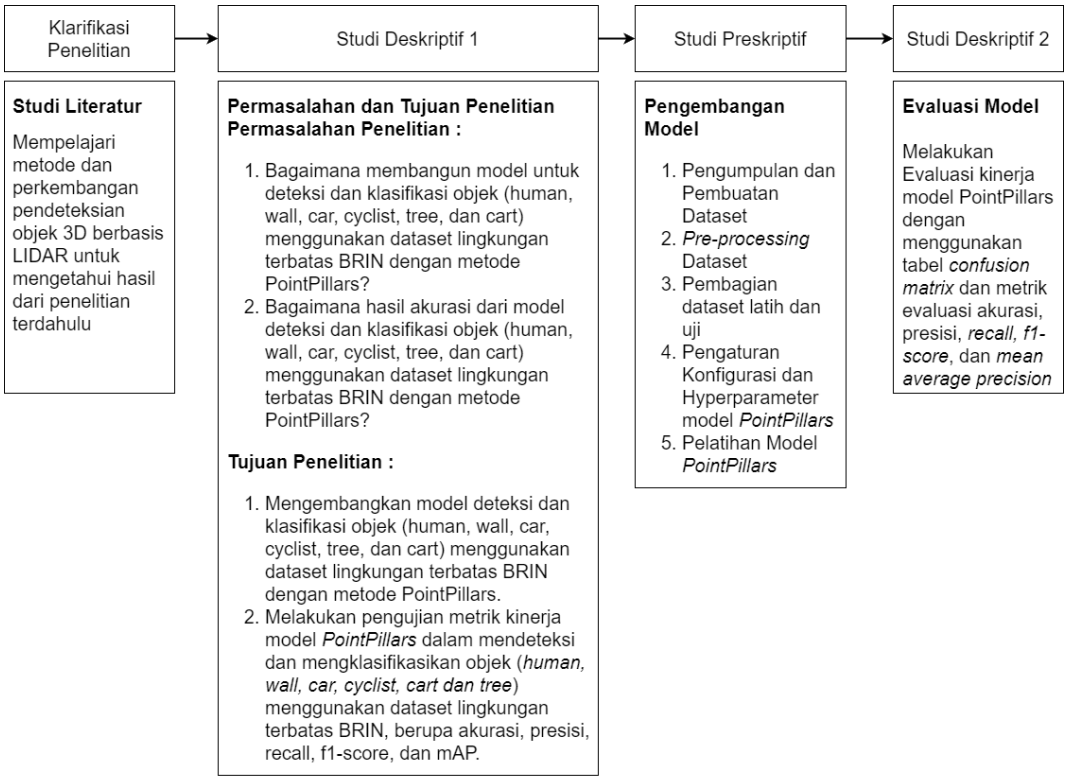
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini dirancang dengan menggunakan pendekatan *Design Research Methodology (DRM)* yang terdiri dari klarifikasi penelitian, studi deskriptif 1, studi prespektif, dan studi deskriptif 2 (Blessing & Chakrabarti, 2009). Dengan menggunakan DRM, maka skema dari penelitian ini dapat dirancang seperti pada gambar 3.1.



Gambar 3.1 Skema Penelitian

3.1.1 Klarifikasi Penelitian

Bagian ini, merupakan studi literatur yang terkait dengan deteksi objek 3D khususnya pada model jaringan saraf *PointPillars*, serta beberapa penelitian terdahulu yang mendukung pembentukan model jaringan saraf *PointPillars*. Hal-hal yang terkait dengan masukan dataset yang diperlukan maupun dengan konsep kerja model juga menjadi bahan yang perlu dikaji, termasuk juga bagaimana

mengatur parameter kerja model pada saat dilatih dengan dataset maupun saat menguji model yang telah terlatih. Studi literatur dilakukan dengan cara menulusuri artikel ilmiah internasional yang terkait melalui media internet, seperti pada website springer, IEEE, SINTA, dan Google Scholar.

3.1.2 Studi Deskriptif 1

Pada tahapan ini, masalah dan tujuan penelitian yang telah didapatkan melalui studi literatur dirumuskan. Dari mulai permasalahan bagaimana membangun model untuk deteksi dan klasifikasi objek (*human, wall, car, cyclist, dan tree*) menggunakan dataset lingkungan terbatas BRIN dengan metode *PointPillars*, hingga menghitung hasil akurasi dari model deteksi dan klasifikasi menggunakan dataset lingkungan terbatas BRIN dengan metode *PointPillars*. Permasalahan tersebut dibuat berdasarkan latar belakang penelitian.

Permasalahan yang dirumuskan tersebut selanjutnya mendorong penelitian untuk menuju tujuan dari penelitian yaitu mengembangkan model deteksi dan klasifikasi objek (*human, wall, car, cyclist, dan tree*) menggunakan dataset lingkungan terbatas BRIN dengan metode *PointPillars* dan melakukan pengujian akurasi model *PointPillars* dalam mendeteksi dan mengklasifikasikan objek (*human, wall, car, cyclist, dan tree*) menggunakan dataset lingkungan terbatas BRIN.

3.1.3 Studi Preskriptif

Pada bagian ini, tahapan pembuatan model jaringan saraf *PointPillars* dirumuskan. Tahap pertama adalah proses perolehan data *point cloud* LIDAR menggunakan sensor Velodyne VLP-16 dengan kemampuan mengukur sebanyak 600.000 poin data per detiknya dan memiliki 16 buah pantulan laser yang dapat melakukan bidang pandang horizontal 360° dan bidang pandang vertikal 30°. Sensor LIDAR ini, dipasang dengan beberapa sensor lainnya diatas sebuah kendaraan listrik otonom dan akan merekam setiap data objek yang terlintas oleh kendaraan otonom. Data yang diperoleh tersebut, selanjutnya diformulasikan kedalam bentuk frame-frame. Jadi setiap frame berisikan data *point cloud* dari objek yang terdeksi. Pada awalnya data tersebut akan disimpan dalam format .pcap

file. Didalam file tersebut tersimpan beberapa informasi seperti lokasi koordinat dari sebuah objek (x, y, dan z), dimensi dari objek (dx, dy, dan dz), informasi intensitas cahaya, informasi cahaya dari setiap pantulan lasernya dan informasi lainnya. Untuk dapat melihat file dengan format ini diperlukan sebuah aplikasi khusus bernama VeloView. Dengan aplikasi ini, file dengan format .pcap dapat dilihat dan diubah menjadi bentuk frame-frame *point cloud* yang berformat .bin. File dengan format ini merupakan bentuk standar dari data *point cloud* LIDAR yang digunakan oleh KITTI. Oleh karena itu, pengkonversian dataset dilakukan untuk mengubah data KITTI tersebut menjadi data yang dapat dimasukan kedalam jaringan saraf *PointPillars* yaitu format .npy atau format numpy. Format ini memiliki isian yang lebih sederhana dari format .pcap dan .bin sebelumnya, dimana datanya hanya berisi lokasi x, y, dan z dari titik *pointcloud* dan jumlah intensitas cahaya dari laser. Setelah pengkonversian dataset dilakukan, langkah selanjutnya adalah melakukan seleksi pada seluruh dataset yang akan dipakai untuk digunakan dalam pembagian data uji dan data latih. Hal ini dilakukan karena pada saat pengambilan dataset terdapat duplikasi frame-frame dataset yang isinya sama, sehingga hasil akhir dari jumlah keseluruhan dataset adalah 2423 *point cloud* dataset yang dibagi menjadi 2180 dataset latih dan 243 dataset uji dimana rasio dari pembagian datasetnya adalah 90% dataset latih dan 10% dataset uji. Rasio pembagian dataset ini diambil berdasarkan penelitian model *PointPillars* yang dilakukan oleh (Lang dkk., 2019) yang menggunakan 90% datasetnya untuk pelatihan model dan 10% datasetnya untuk pengujian model. Dengan menggunakan 90% data untuk pelatihan, model memiliki akses ke jumlah data yang lebih besar. Ini membantu model untuk belajar pola-pola yang lebih baik dan membuatnya lebih kuat serta akurat. Data yang lebih banyak membantu dalam menghindari overfitting, di mana model belajar terlalu spesifik pada data latih dan tidak dapat menggeneralisasi dengan baik pada data baru.

Selanjutnya, dataset tersebut akan melalui beberapa proses sebelum dilatih oleh model jaringan saraf *PointPillars* seperti proses anotasi dan proses ekstraksi informasi. Proses anotasi dilakukan dengan menggunakan tools bernama *supervisely* yang dapat diakses secara online tanpa perlu menggunakan GPU lokal untuk menganotasi datasetnya. Dataset dianotasi sesuai dengan kelas objek yang

telah ditentukan yaitu human, wall, car, cyclist, tree, dan cart. Hasil dari proses anotasi menggunakan *supervisely* ini adalah sebuah .json file yang berisi informasi tentang objek yang telah dianotasi. Untuk keperluan pelatihan model, format anotasi yang digunakan berbeda dengan hasil anotasi yang didapat melalui *supervisely*, sehingga proses konversi dari .json file menjadi format yang digunakan untuk pelatihan model yaitu .txt file harus dilakukan. Format dari txt file tersebut adalah lokasi koordinat (x, y, dan z) objek yang dianotasi, dimensi dari objek (dx, dy, dan dz), sudut arah objek, dan nama kelas dari objek. Setelah melalui tahap tersebut, dataset akan melalui proses ekstraksi informasi dimana semua data yang terdapat pada dataset dan file anotasi akan digabung dan disimpan didalam beberapa pickle file seperti *Ground Truth* Database yang berisi hasil ekstraksi objek *pointcloud* dari dataset berdasarkan anotasi yang telah dibuat, custom validation info yang berisi data *pointcloud* mana saja yang akan masuk kedalam data uji, custom dbinfos train yang berisi data-data tempat penyimpanan file ground truth dan file train, custom train info yang berisi data *pointcloud* mana saja yang akan masuk kedalam data latih.

Setelah dataset terformulasikan melalui dua tahap sebelumnya, model jaringan saraf *PointPillars* dikonfigurasikan agar dapat dilatih dengan dataset tersebut. Konfigurasi dari model jaringan saraf *PointPillars* ini menggunakan dua *backbone* 2D network yang berbeda yaitu *BaseBEVBackbone* dan BaseBEVResBackbone. Kedua *backbone* 2D tersebut memiliki perbedaan yaitu pada *BaseBEVBackbone* menggunakan blok konvolusional standar tanpa residual connection. Blok konvolusional ini terdiri dari serangkaian *layer* termasuk zero padding, konvolusi, normalisasi batch, dan aktivasi ReLU. Penumpukan blok-blok ini dilakukan beberapa kali untuk mengekstrak fitur dari data masukan. Selanjutnya, *backbone* 2D yang kedua yaitu BaseBEVResBackbone yang menggunakan blok residual, bukan blok konvolusional standar. Blok residual memperkenalkan koneksi pintasan yang memungkinkan gradien masuk langsung melalui *layer*, mengatasi masalah hilangnya gradien dan memungkinkan jaringan mempelajari fitur yang lebih kompleks. Selanjutnya, konfigurasi yang harus ditentukan adalah penentian jumlah *layer* dan voxel size yang akan digunakan pada kedua *backbone* 2D. Penentuan voxel size adalah menentukan besar jumlah

kumpulan titik-titik *point cloud* yang akan digunakan untuk *training*, sedangkan penentuan jumlah *layer* adalah menentukan jumlah dan *layer* apa saja yang akan digunakan pada model *PointPillars*. Penentuan kedua nilai ini sangat krusial dalam proses pelatihan karena akan berpengaruh terhadap performa deteksi dari model. Konfigurasi dari kedua nilai tersebut dapat dilihat pada tabel 3.1 berikut.

Tabel 3.1
Tabel konfigurasi jaringan saraf *PointPillars*

Konfigurasi Model	**Backbone**	**Layer Composition**	**Voxel Size**
1	BEV	3;5;5	16000
2	BEV	3;5;5	8000
3	BEV	3;5;5	32000
4	BEV	4;6;6	16000
5	BEVRes	2;3;3	16000
6	BEV	3;5;5	64000
7	BEVRes	2;3;3	8000
8	BEVRes	2;3;3	64000

Setelah proses pembuatan konfigurasi selesai maka hyperparameter dari proses pelatihan akan di ujicoba dan ditentukan. Selanjutnya akan dilakukan proses pelatihan model jaringan saraf *PointPillars* dengan menggunakan dataset dan konfigurasi model yang telah dijelaskan sebelumnya. Pada proses *training, epochs* atau satu siklus penuh di mana model melihat seluruh dataset pelatihan sekali yang digunakan pada setiap konfigurasi model adalah nilai *epochs* yang meninimalkan nilai *training* lossnya.

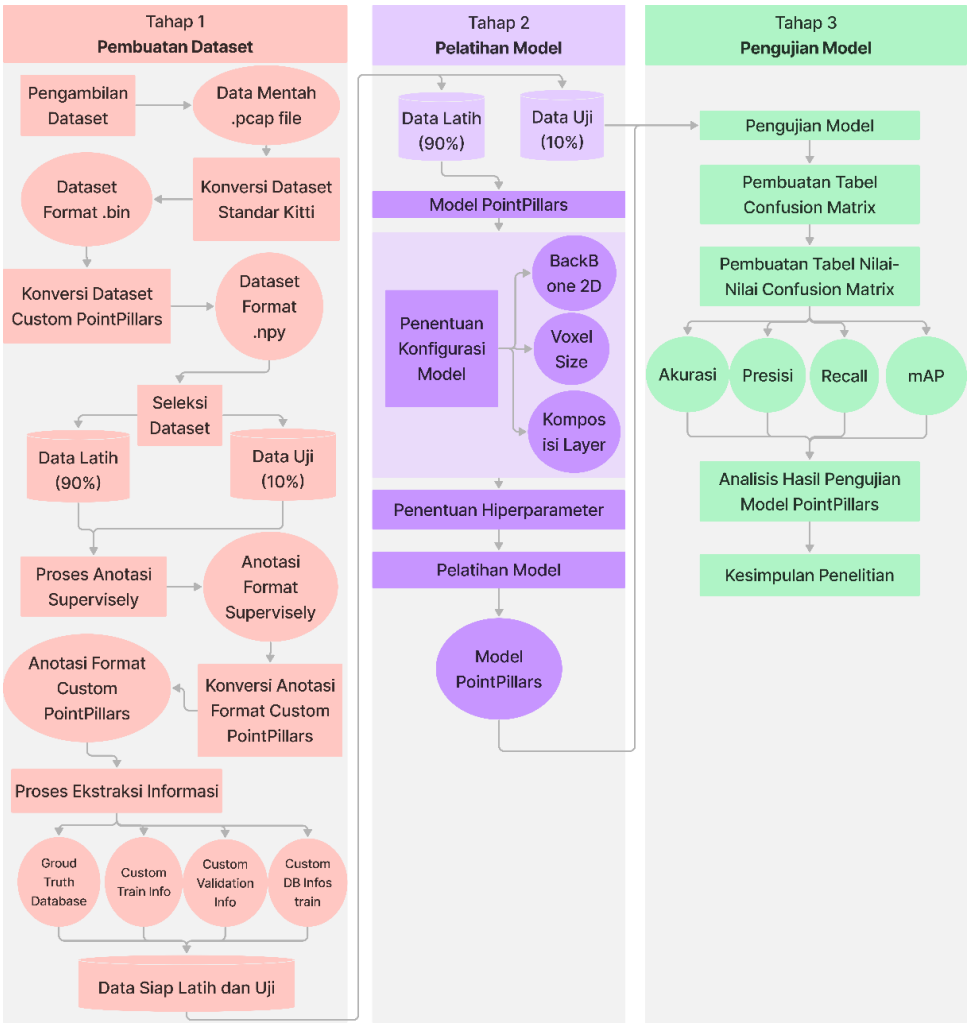
3.1.4 Studi Deskriptif 2

Pada bagian ini, Model yang telah terlatih selanjutnya diuji dengan menggunakan 10% dari dataset yang telah disiapkan sebelumnya. Tahap awal proses pengujian dilakukan dengan pembuatan tabel visualisasi *confusion matrix* yang membandingkan hasil aktual dan hasi prediksi dari model dengan menggunakan toolbox yang disediakan oleh OpenPCDet. Setelah itu, pembuatan

tabel dari nilai-nilai TP, TN, FP, dan FN dari *confusion matrix* dilakukan. Dari data tersebut diperoleh nilai akurasi, presisi, *recall*, *f1-score*, dan *average precision*. Setelah itu akan dilakukan analisis mendalam terkait nilai-nilai tersebut sehingga menghasilkan kesimpulan dari penelitian. Pada Lampiran 1, diperlihatkan contoh hasil aktual *pointcloud* (*Ground Truth*) dan hasil prediksi.

3.2 Prosedur Penelitian

Pada penelitian ini, prosedur penelitian yang diusulkan akan mencakup beberapa tahap proses untuk pembuatan model jaringan saraf *PointPillars* dan pengujian model jaringan saraf *PointPillars* menggunakan beberapa metrik yang telah dijelaskan. Prosedur penelitian yang diusulkan pada penelitian ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Prosedur penelitian yang diusulkan

Penjelasan dari tahap-tahap prosedur penelitian tersebut adalah sebagai berikut :

1. Pengambilan Dataset

   Pada tahap awal, pengambilan dataset dilakukan dengan menggunakan LIDAR Velodyne VLP-16 yang hasilnya adalah data mentah berupa file *point cloud* dengan format .pcap. File tersebut berisi tentang semua informasi yang telah ditangkap oleh LIDAR seperti kordinat xyz dari objek yang terdeteksi, intensitas cahaya, jarak pantulan objek dengan posisi LIDAR, *Azimuth* yaitu bidang pandang horizontal 360 derajat dari LIDAR, dan informasi lainnya mengenai keberadaan objek.

2. Pembuatan Dataset

   Pada tahap ini, file dataset LIDAR *point cloud* yang telah diambil akan dikonversi menjadi format standar KITTI .bin yaitu format data LIDAR yang berbentuk frame. Setelah itu, data tersebut dikonversi kembali dalam bentuk format yang dapat diproses oleh jaringan saraf *PointPillars* yaitu format .npy atau format *numpy*. Selanjutnya, dataset tersebut akan melalui proses seleksi dataset yaitu pembagian dataset keseluruhan menjadi data latih dan data uji dengan rasio 90% untuk data latih dan 10% untuk data uji. Kemudian, dataset tersebut akan melalui proses anotasi menggunakan tools *supervisely*. Hasil dari penganotasian tersebut akan disimpan dalam file .json dan akan dikonversi ke format yang sesuai dengan jaringan saraf *PointPillars* yaitu .txt. Setelah proses anotasi dan pemisahan dataset, proses selanjutnya adalah pengekstraksian informasi dari dataset tersebut ke dalam *ground truth database*, custom *validation info*, custom db info train, dan custom *train info* yang akan disimpan dalam file pickle.

3. Pembuatan Model

   Pada tahap ini, sebelum dataset dilatih kedalam jaringan saraf *PointPillars*, konfigurasi dari model seperti penentuan jumlah komposisi *layer*, penentuan *backbone* 2D, dan penentuan jumlah voxel dilakukan. Kemudian, pengaturan hyperparameter dilakukan dengan menetukan jumlah *epochs*, jumlah *batch_size*, optimizer yang dipakai, dan *learning rate* awal. Setelah proses-proses tersebut dilakukan, maka dataset sudah bisa dimasukan kedalam jaringan saraf

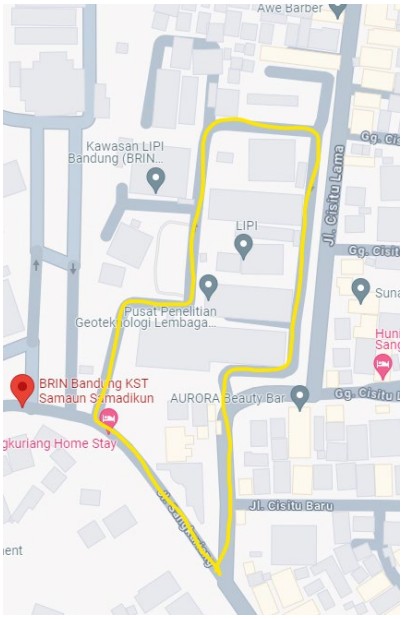
PointPillars dan pelatihan model jaringan saraf *PointPillars* dapat dimulai sesuai dengan konfigurasi yang telah dibuat.

4. Pengujian Model

   Pada saat pelatihan model telah selesai, model tersebut akan diuji dengan menggunakan 10% dataset uji dan toolbox dari OpenPCDet dimana nilai metrik yang akan dihasilkan adalah *Average Precision*, *confussion matrix*, akurasi, presisi, *recall*, dan *f1-score*. Kemudian, analisis akan dilakukan pada hasil nilai metrik tersebut sehingga kesimpulan dari penelitian dapat dicapai.

3.3 Sumber Dataset

Sumber dataset yang digunakan pada penelitian ini adalah dataset yang diambil langsung menggunakan LIDAR Velodyne VLP-16 dan dataset diambil di area jalan sekitar BRIN KST Samaun Samadikun dimana rute dari perjalanannya adalah seperti pada gambar 3.3. Dataset yang diperoleh berjumlah sekitar 2423 frame dalam bentuk *point cloud*. Jumlah ini adalah hasil tahap proses seleksi dari informasi yang mewakili 11.000 *point cloud* dataset dimana didalamnya tidak terdapat duplikasi dataset dan variasi dataset beragam. Dataset tersebut akan dibagi menjadi data latih dan data uji dengan rasio 90% dan 10% dimana data latih berjumlah 2180 dan data uji berjumlah 243. Dalam dataset tersebut terdapat enam buah class objek yaitu human, wall, car, cyclist, tree, dan cart.



Gambar 3.3 Peta Rute Lingkungan BRIN KST Samaun Samadikun Bandung

Proses seleksi dataset dilakukan dalam beberapa tahap untuk mendapatkan data yang relevan dan berkualitas bagi penelitian ini. Proses ini dimulai dengan dataset awal yang berjumlah 11.000 data. Tahap pertama melibatkan penghapusan data duplikat, yang mengurangi dataset menjadi 7.000 data, dengan tujuan mengeliminasi entri yang berulang dan memastikan setiap data unik. Pada tahap berikutnya, dilakukan penghapusan variasi data yang memiliki kesamaan untuk lebih menyaring dataset menjadi 2.423 data akhir. Tahap ini penting untuk memastikan bahwa data yang tersisa memiliki variasi yang signifikan dan relevan untuk analisis lebih lanjut, serta menghindari redundansi informasi yang dapat mempengaruhi akurasi hasil deteksi dan klasifikasi objek. Tahap pemrosesan data dan hasil akhir dapat dilihat pada tabel 3.2 berikut.

Tabel 3.2
Proses seleksi dataset

No	Pemrosesan Data	Jumlah Dataset
1	Dataset awal	11.000
2	Penghapusan duplikasi data	7000
3	Penghapusan variasi data yang memiliki kesamaan	2423

3.4 Instrumen Penelitian

Instrumen penelitian yang digunakan pada penelitian ini adalah *library* dan metrik-metrik yang digunakan dan terkait dengan pengembangan model jaringan saraf *PointPillars* untuk sistem pendeteksian objek berbasis LIDAR. Pada tabel 3.2 diperlihatkan *library* yang digunakan pada penelitian ini. Pada penelitian ini juga digunakan code editor visual studio code untuk menulis algoritma code dan mengembangkan model jaringan saraf *PointPillars*.

Tabel 3.3
Library yang digunakan pada penelitian

No	**Library**	**Deskripsi**
1	numpy	Modul Python yang menyediakan objek array multidimensi, berbagai objek turunan (seperti array dan matriks), dan berbagai macam rutin

		untuk operasi cepat pada array, termasuk matematika, logika, manipulasi bentuk, pengurutan, pemilihan, I/O, aljabar linier dasar, operasi statistik dasar, simulasi acak, dan banyak lagi.
2	pandas	Modul Python yang menyediakan struktur data dan fungsi untuk manipulasi data
3	OpenPCDet	Merupakan kumpulan *library* pendukung pemrosesan data *point cloud* LIDAR termasuk didalamnya model jaringan saraf *PointPillars*
4	pytorch	Merupakan framework pembelajaran mesin sumber terbuka yang dibangun berdasarkan bahasa pemrograman Python.
5	Open3D	Merupakan *library* untuk visualisasi data *point cloud* LIDAR
6	OpenCV	Merupakan *library* untuk pemrosesan gambar 2D
7	Matplotlib	Merupakan *library* untuk visualisasi grafik atau tabel
8	Tensorboard	Merupakan *library* untuk melakukan visualisasi pada saat proses *training* berlangsung
9	Python Package Manager (PIP)	Merupakan sistem pengelolaan paket atau package untuk menyimpan atau mengunduh *library-library* pendukung yang digunakan
10	Jupyter Notebook	Merupakan extension yang berasal dari visual studio code dan digunakan untuk menulis dan menjalankan algoritma code yang berekstensi

		.ipynb

Pada tahap pengujian, visualisasi untuk pengukuran performa klasifikasi dilakukan dengan membuat tabel *Confusion Matrix* yang dapat dilihat pada tabel 3.3. Untuk dapat menentukan nilai dari TP, FP, dan FN, nilai dari IoU perlu dihutung sebelumnya menggunakan persamaan IoU pada tabel 3.3. TP akan didapatkan saat IoUnya lebih dari 50% atau 0.5, FP akan didapatkann saat IoUnya kurang dari 0.5, dan FN akan didapatkan saat IoUnya lebih dari 0.5 namun objek yang dideteksi salah atau misklasifikasis. Sedangkan untuk nilai TN tidak terlalu berpengarung kepada deteksi karena selalu ada objek yang terprediksi di dalam *point cloud*.

Tabel 3.4
Confusion matrix

		Prediksi		
		Kelas 1	Kelas 2	Kelas 3
Aktual	Kelas 1	a	b	c
	Kelas 2	d	e	f
	Kelas 3	g	h	i

Langkah selanjutnya melibatkan perhitungan metrik untuk memperoleh nilai akurasi metrik sesuai dengan persamaan yang tercantum dalam Tabel 3.4. Metrik yang dihitung meliputi akurasi, presisi, recall, F1-score, Intersection over Union (IoU), dan Mean Average Precision (mAP). Setiap metrik ini memberikan gambaran yang berbeda mengenai kinerja sistem dalam mendeteksi dan mengklasifikasikan objek, sehingga penting untuk menghitung semuanya guna memperoleh evaluasi yang komprehensif.

Tabel 3.5
Persamaan metrik-metrik evaluasi

No	**Nama Metrik**	**Persamaan**
1	Akurasi	$\frac{TP + TN}{TP + TN + FP + FN}$
2	Presisi	$\frac{TP}{TP + FP}$

3	Recall	$\frac{TP}{TP + FN}$
4	F1-Score	$\frac{2 * precision * recall}{precision + recall}$
5	IoU	$IoU = \frac{Area\ Tumpang\ Tindih}{Area\ Union}$
6	*Mean Average Precision*	$mAP = \frac{\sum_{i=1}^{K} APi}{K}$

3.5 Alat dan Bahan

Pada penelitian ini digunakan alat dan bahan sesuai dengan kebutuhan penelitian. Alat dan bahan tersebut dikategorikan dalam bentuk perangkat keras dan perangkat lunak. Pada tabel 3.5 diperlihatkan susunan alat dan bahan yang digunakan.

Tabel 3.6
Alat dan bahan

Perangkat Keras	**Deskripsi**	**Perangkat Lunak**	**Deskripsi**
LIDAR Velodyne VLP-16	Digunakan untuk pengambilan data *point cloud* LIDAR	Linux Ubuntu 22.04	Sistem operasi komputer
Komputer dengan kemampuan : - Processor Intel i7 G11 - RAM 16GB - GPU NVIDIA	Digunakan untuk menjalankan aplikasi perangkat lunak yang dibutuhkan	Supervisely	Perangkat lunak untuk menganotasi data *point cloud* LIDAR

Perangkat Keras	Deskripsi	Perangkat Lunak	Deskripsi
RTX 3060 - Memory 2TB - Display 24 inch - Mouse - Keyboard			
Jaringan internet	Untuk mengakses sumber studi literatur	Web Browser	Untuk menelusuri studi literatur
-	-	Python	Bahasa pemrograman yang digunakan untuk penelitian
-	-	Anaconda	Untuk manajemen lingkungan kerja program python
-	-	CUDA 11.7 dan CUDNN 8.6	Komputasi GPU dan *library* yang membantu framework deep learning
-	-	VeloView	Aplikasi untuk melihat dan menkonversi data mentah LIDAR *point cloud*
-	-	Visual Studio Code	digunkan untuk menulis dan menjalankan code algoritma yang dibuat.