

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Perkembangan teknologi dan implementasi 3 dimensi (3D) terus ditingkatkan seiring dengan berkembangnya teknologi khususnya di bidang *computer vision*. Penelitian deteksi objek 3D mulai diterangkan khususnya implementasi pada *autonomous driving*. Sebagian besar upaya yang dilakukan pada pendeteksian objek 3 dimensi (3D) dalam *autonomous driving* ini mengandalkan sistem *Light Detection and Ranging* (LiDAR) [1]. Deteksi objek 3D berbasis LiDAR tidak dapat dihindari untuk *autonomous driving* [2]. Pada sebuah penelitian menyatakan bahwa dalam beberapa tahun terakhir, salah satu tantangan dalam deteksi objek 3D adalah mengurangi latensi *inference* (beban komputasi) dan memori tanpa mengorbankan akurasi deteksi berbasis LiDAR untuk *autonomous driving* [1]. Oleh karena itu, masalah tersebut menjadi perhatian bagi para peneliti untuk mengoptimalkan deteksi objek 3D dengan tetap mempertahankan tingkat akurasi.

Beberapa penelitian mampu menyelesaikan masalah diatas dengan menggunakan metode yang berbeda-beda. Beberapa penelitian deteksi objek 3D untuk *autonomous driving* telah dilakukan. Seperti *Frustum-based Networks* [3] telah menunjukkan performansi tinggi di deretan KITTI (*Karlsruhe Institute of Technology and Toyota Technological Institute*) *Benchmark*. Model ini menduduki posisi kedua untuk deteksi objek berupa mobil, pejalan kaki dan pengendara sepeda dengan *Bird-Eye-View* (BEV) menggunakan *Point-Net*. Model ini memiliki dua kelemahan yaitu pertama akurasi model bergantung pada gambar kamera dan *Convolutional Neural Network* (CNN) yang digunakan, sehingga tidak mungkin untuk menerapkan pendekatan hanya menggunakan data dari LiDAR saja serta model berjalan dengan kecepatan yang terlalu rendah pada sekitar 7 *frame per second* (fps) menggunakan GPU NVIDIA GTX 1080i sehingga sangat tidak efisien diterapkan secara *real-time*. Adapun penelitian yang hanya menggunakan data dari LiDAR saja yaitu dari Yin Zhou dan kawan-kawanya [4]. Model ini mendapatkan peringkat terbaik pada KITTI *Benchmark* untuk mendeteksi objek 3D dengan BEV

hanya menggunakan *input* berupa data dari LiDAR. Namun, meskipun akurasinya tinggi, model berakhir dalam waktu *inference* yang rendah yaitu 4 fps menggunakan GPU TitanX sehingga tidak efisien digunakan secara *real-time*. Selain itu, ada juga Chen X dan kawan-kawannya menghasilkan suatu model bernama *Multi-View 3D Object Detection Network* (MV3D) [5]. Model ini memproyeksikan LiDAR *point cloud* dalam *voxel* menggunakan RGB-Maps. Kelebihan dari model ini yaitu waktu komputasi yang cepat menggunakan NVIDIA GTX 1080i GPU, namun sayangnya terdapat kekurangan yaitu kebutuhan *input* sensor sekunder (kamera).

Berdasarkan permasalahan diatas, maka pada Tugas Akhir ini diusulkan untuk melakukan pengembangan pada metode yang sudah ada yaitu *YOLO-Complex* [2] dengan beberapa skema analisis yaitu analisis menggunakan pendekatan *upsample* pada bagian *backbone* CSPDarknet-53 untuk mengetahui metode ini dapat bekerja dengan baik secara *real-time*. Skema pengujian akan dilakukan setelah melakukan analisis. Dengan ini diharapkan Tugas Akhir ini dapat digunakan sebagai referensi pada penelitian selanjutnya.

## 1.2. Rumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah tersebut, maka rumusan masalah yang dapat diberikan pada Tugas Akhir ini adalah:

1. Bagaimana cara untuk meningkatkan akurasi pada model *Complex-YOLOv4*?
2. Bagaimana mencari pendekatan *upsample* terbaik untuk meningkatkan kinerja deteksi objek 3D berbasis LiDAR?
3. Bagaimana cara mengukur dan menganalisis parameter performansi untuk kasus objek deteksi *multi-kelas* 3D?

## 1.3. Tujuan dan Manfaat

Berdasarkan rumusan masalah di atas, manfaat dari penelitian Tugas Akhir ini yaitu mendapatkan nilai akurasi yang lebih tinggi untuk *Complex-YOLOv4*

sehingga *bounding box* 3D yang dihasilkan lebih presisi untuk digunakan secara *real-time*. Berikut merupakan tujuan dari Tugas Akhir ini:

1. Melakukan analisis pada bagian *upsample* dari *backbone Complex-YOLOv4* untuk menghasilkan efisiensi secara *real-time*.
2. Mendapatkan *upsample* dengan nilai mAP terbaik dan lebih tinggi dari hasil sebelumnya pada metode *Complex-YOLOv4* ini.
3. Meningkatkan performansi model dengan menggunakan *evaluation metrics* diantaranya *precision*, *recall*, *f1-score*, *average precision* dan *mean average precision* untuk meningkatkan akurasi namun tetap efisien dalam pemrosesannya secara *real-time*.

#### **1.4. Batasan Masalah**

Untuk membatasi cakupan pengerjaan dan memfokuskan area kerja, dalam penelitian Tugas Akhir ini dibentuk batasan masalah sebagai berikut:

1. Hanya berfokus pada analisis penggunaan *upsampling* terbaik untuk metode *Complex-YOLOv4*.
2. *Input* data hanya berupa data dari LiDAR yaitu *3D point cloud*.
3. Menggunakan KITTI *3D object detection dataset*.
4. Parameter analisis performansi yang dipakai menggunakan *evaluation metrics* diantaranya *precision*, *recall*, *average precision (AP)*, *f1-score*, dan *mean average precision (mAP)*.
5. Spesifikasi *tools* yang digunakan dalam penelitian ini diantaranya Ubuntu 20.04 LTS, GPU Single NVIDIA 1080Ti, *Pytorch* 1.5.0, dan *Python* 3.6.

#### **1.5. Metode Penelitian**

Metode penelitian yang digunakan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

##### **1. Studi Literatur**

Tahap ini berupa pengumpulan referensi berupa pencarian, pengumpulan, dan memahami jurnal, *paper*, artikel, buku, *website*, maupun referensi lain

yang berhubungan dengan objek deteksi 3D, YOLOv4, *Convolutional Neural Network* (CNN), *autonomous driving* dan LiDAR.

## 2. Persiapan Data

Tahapan ini dilakukan dengan menyiapkan beberapa data yang diperlukan diantaranya:

- (29 GB) *Input* data untuk metode *Complex-YOLOv4* berupa *velodyne point clouds*.
- (5 MB) *Input* label untuk metode *Complex-YOLOv4* berupa *training labels*.
- (16 MB) Matriks kalibrasi kamera dari *dataset* untuk visualisasi prediksi.
- (12 GB) Gambar *dataset* untuk visualisasi prediksi.

## 3. Perancangan Sistem

Sistem ini menggunakan bahasa pemrograman *Python* dengan *framework PyTorch* versi 1.5.0 dan sistem operasi Linux yaitu Ubuntu 20.04 LTS.

## 4. Proses Pengujian

Skema pengujian dilakukan dengan memodifikasi algoritma *upsampling* pada *backbone* CSPDarknet-53 dari *Complex-YOLOv4* untuk mengetahui sejauh mana efisiensi dari model ini dapat dikembangkan. Tahap ini dilakukan *improvement* jika performansi yang baik belum tercapai.

## 5. Analisis Performansi Hasil Pengujian

Tahap ini berupa analisis pada kinerja parameter performansi dari hasil proses pengujian berupa *precision*, *recall*, *average precision* (AP), *f1-score*, dan *mean average precision* (mAP).

## 6. Kesimpulan

Tahap ini dilakukan penyusunan laporan dari berbagai percobaan yang telah dilakukan dan didapatkan mode *upsampling* terbaik untuk metode *Complex-YOLOv4* dalam mendeteksi objek 3D dengan efisien secara *real-time*. Dimana pada tahap ini dilakukan penarikan kesimpulan dari berbagai skenario.

## 1.6. Sistematika Penulisan

Sistematika penulisan laporan Buku Tugas Akhir ini adalah sebagai berikut:

- **BAB II TINJAUAN PUSTAKA**  
Bab ini akan menjelaskan konsep dasar dan landasan teori yang berkaitan dengan sistem, metode dan algoritma yang digunakan dalam penelitian Tugas Akhir ini.
- **BAB III PERANCANGAN SISTEM**  
Bab ini menjelaskan tentang desain model sistem, analisis parameter kinerja dan spesifikasi perangkat yang digunakan.
- **BAB IV HASIL DAN ANALISIS**  
Bab ini berisi tentang analisis dari pengujian mode *upsample* pada bagian *backbone CSPDarknet-53* yang diimplementasikan menggunakan metode *Complex-YOLOv4* dengan menggunakan 3 mode *upsample* diantaranya *bicubic*, *nearest*, dan *bilinear*.
- **BAB V SIMPULAN DAN SARAN**  
Bab ini berisi mengenai kesimpulan dari seluruh hasil dan analisis yang didapatkan serta saran untuk pengembangan selanjutnya.