



TESIS - TF 142510

**RANCANG BANGUN SISTEM PRESEPSI BERBASIS  
SSD (*SINGLE SHOT MULTIBOX DETECTOR*)  
DENGAN SENSOR FUSION KAMERA MONO DAN  
ULTRASONIK**

**AKHMAD IBNU HIJA  
NRP. 02311950010005**

**DOSEN PEMBIMBING  
Dr. Katherin Indriawati, S.T., M.T**

**PROGRAM MAGISTER  
JURUSAN TEKNIK FISIKA  
FAKULTAS TEKNOLOGI INDUSTRY DAN REKAYASA  
SISTEM  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2022**

*Halaman ini sengaja dikosongkan*



TESIS - TF 142510

**DESIGN PERCEPTION SYSTEM BASED SSD (SINGLE  
SHOT DETECTOR) WITH FUSION SENSOR USING  
MONO CAMERA AND ULTRASONIC**

**AKHMAD IBNU HIJA  
NRP. 02311950010005**

**DOSEN PEMBIMBING  
Dr. Katherin Indriawati, S.T., M.T**

**MASTER PROGRAM  
ENGINEERING PHYSICS DEPARTMENT  
FACULTY OF INDUSTRIAL TECHNOLOGY AND  
SYSTEMS ENGINEERING  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2022**

*Halaman ini sengaja dikosongkan*

## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Teknik (MT)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**Akhmad Ibnu Hija**

**NRP. 02311950010005**

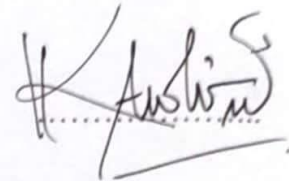
Tanggal Ujian: 8 Januari 2022

Periode Wisuda: September 2022

Disetujui oleh:

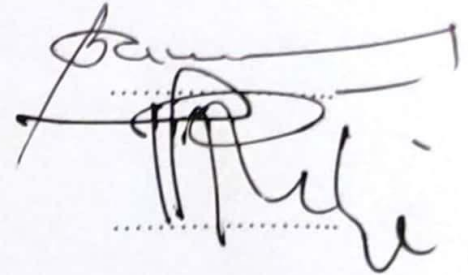
**Pembimbing:**

1. Dr. Katherin Indriawati, S.T., M.T.  
NIP. 19760523 200012 2 001



**Penguji:**

1. Dr. Bambang Lelono Widjiantoro S.T, M.T,  
NIP. 19690507 199512 1 001
2. Prof. Totok Ruki Biyanto S.T, M.T, Ph, D  
NIP. 19710702 199802 1 001



Kepala Departemen Teknik Fisika  
Fakultas Teknologi Industri dan Rekayasa Sistem



Dr. Suyanto, S.T., M.T.  
NIP. 19711113 199512 1 002

*Halaman ini sengaja dikosongkan*

## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan dibawah ini:

Nama : Akhmad Ibnu Hija  
NRP : 02311950010005  
Departemen / Prodi : Teknik Fisika / S2 Teknik Fisika  
Fakultas : Fakultas Teknologi Industri dan Rekayasa Sistem  
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tesis dengan judul “Rancang Bangun Sistem Presepsi Berbasis SSD (*Single Shot Multibox Detector*) Dengan Sensor Fusion Kamera Mono dan Ultrasonik” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila dikemudian hari terbukti terdapat plagiat pada Tesis ini, maka saya akan bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 18 Juli 2022

Yang membuat pernyataan,



Akhmad Ibnu Hija

NRP. 02311950010005

*Halaman ini sengaja dikosongkan*

# **RANCANG BANGUN SISTEM PERSEPSI BERBASIS SSD (*SINGLE SHOT MULTIBOX DETECTOR*) DENGAN *SENSOR FUSION* KAMERA MONO DAN ULTRASONIK**

Nama Mahasiswa : Akhmad Ibnu Hija  
NRP : 02311950010005  
Pembimbing : Dr. Katherin Indriawati, S.T., M.T.

## **ABSTRAK**

*Autonomous vehicle* telah berkembang pesat seiring dengan berkembangnya teknologi dibidang elektronika. Sistem persepsi sebagai system untuk mengenali lingkungan menjadi hal yang penting dalam *Autonomous Vehicle* sebelum melangkah ke sistem perencanaan dan kontrol. Banyaknya variable yang perlu diukur dalam sistem persepsi memerlukan gabungan beberapa sensor untuk mengenali lingkungan secara lebih baik yang disebut dengan *sensor fusion*. Penelitian ini berfokus pada *sensor fusion* yang diterapkan pada *single board computer* (SBC) NVIDIA Jetson Nano Board. *Sensor fusion* dilakukan menggunakan kamera mono Webcam Xiaovv 1080P dan sensor ultrasonik HC-SR04. Mono kamera mendeteksi objek menggunakan algoritma *Single Shot Detector* (SSD) untuk mendeteksi dua kelas dalam satu frame yaitu mobil dan lajur kiri/kanan. Lalu sensor ultrasonik digunakan untuk mendeteksi jarak antara mobil dengan *ego vehicle*. Kedua sensor tersebut kemudian digabungkan menggunakan estimasi pada Kalman Filter untuk mendapatkan hasil deteksi posisi objek, jarak terhadap *ego vehicle* dan posisi *ego vehicle* terhadap lajur jalan. Hasil penelitian menunjukkan bahwa sistem deteksi objek SSD pada mono kamera diterapkan pada beberapa skenario keadaan menunjukkan tidak adanya *false* dalam deteksi objek. Akurasi deteksi jarak *ego vehicle* dengan objek mengukur dengan baik. Penelitian ini masih perlu ditingkatkan lagi dikemudian hari terutama untuk sensor jarak yang lebih presisi serta kompleksitas dari lingkungan yang dideteksi dapat ditingkatkan agar mendekati pengukuran terhadap lingkungan pada kondisi sebenarnya.

**Kata kunci:** Kalman Filter, Mono Kamera, Sistem Persepsi, SSD, Ultrasonik

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Alhamdulillah segala puji dan syukur penulis panjatkan kepada Allah SWT atas limpahan rahmat, hidayat, serta petunjuk-Nya sehingga penulis dapat menyelesaikan tesis pada jenjang magister yang berjudul “*Rancang Bangun Sistem Persepsi Berbasis SSD (Single Shot Detector) Dengan Sensor Fusion Kamera Mono dan Ultrasonik*”. Selama proses pengerjaan dan penyusunan laporan tesis ini, telah banyak pihak yang terlibat dan membantu penulis, sehingga untuk kesempatan kali ini izinkan penulis mengucapkan terima kasih kepada:

1. Kedua orang tua tercinta, Bapak Dahlan Thohari dan Ibu Endang Sriwinarsih. Serta seluruh keluarga penulis, semuanya tanpa terkecuali, juga kepada Nabila Puspita Firdi yang telah mensupport penulis baik secara mental maupun finansial.
2. Ibu Dr. Katherin Indriawati, S.T., M.T., selaku dosen pembimbing yang telah membimbing dengan sabar kepada penulis dan memberikan banyak ilmu dalam membantu penulis menyelesaikan tesis ini.
3. Bapak Dr. Bambang Lelono Widjiantoro, S.T., M.T. dan Bapak Prof. Totok Ruki Biyanto S.T, M.T, Ph, D selaku dosen penguji yang telah memberikan juga memberika masukan yang membangun dan membantu penulis menyelesaikan tesis ini.
4. Teman-teman seperjuangan Laboratorium *Embedded Cyber-Physical System* (ECS) dan seperjuangan kuliah S2 angkatan 2019 yang telah saling memberikan dukungan, ilmu dan pengalaman.
5. Semua pihak yang telah memberikan dukungan dan memberikan doa yang tidak bisa penulis sebutkan satu per satu.

Penulis menyadari sepenuhnya bahwa laporan tesis ini masih jauh dari sempurna, sehingga diharapkan kritik dan saran yang membangun dari semua pihak demi penelitian yang lebih baik di masa mendatang. Harapan penulis, semoga laporan tesis ini bermanfaat bagi pembaca dan semua pihak.

Surabaya, Juni 2022

*Halaman ini sengaja dikosongkan*

## DAFTAR ISI

HALAMAN SAMBUNG .....	iii
LEMBAR PENGESAHAN TESIS .....	v
PERNYATAAN BEBAS PLAGIASI .....	vii
ABSTRAK .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2 Perumusan Masalah .....	3
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah .....	3
1.5 Manfaat Penelitian .....	4
BAB 2 DASAR TEORI .....	5
2.1 Autonomous Vehicle .....	5
2.2 Sistem Persepsi .....	6
2.3 Object Detection .....	9
2.4 Sensor Fusion .....	12
2.5 <i>Error</i> Pengukuran .....	16
BAB 3 METODE PENELITIAN .....	17
3.1 Studi Literatur .....	17
3.2 Penyusunan Prototipe .....	18
3.3 Pra-perancangan algoritma sensor fusion .....	19
3.4 Pengukuran Sensor ultrasonik .....	19
3.5 Object Detection .....	22
3.6 Perancangan <i>Sensor Fusion</i> .....	28
3.7 Pengujian Performa <i>Sensor Fusion</i> .....	29
3.8 Penyusunan Laporan .....	29
BAB 4 HASIL DAN PEMBAHASAN .....	31

4.1 Pengukuran jarak menggunakan sensor HC-SR04 .....	31
4.2 Object Detection .....	35
BAB 5 KESIMPULAN DAN SARAN .....	43
5.1 Kesimpulan.....	43
5.2 Saran.....	44
DAFTAR PUSTAKA .....	45
LAMPIRAN .....	49

## DAFTAR GAMBAR

Gambar 2. 1 Rentang Panjang gelombang sensor yang digunakan pada sistem persepsi (Rosique et al., 2019).....	6
Gambar 2. 2 Webcam Xiaovv 1080P.....	8
Gambar 2. 3 Proses kerja Sensor HC-SR04 (Morgan, 2014).....	8
Gambar 2. 4 Sudut dari sensor Ultrasonik (Morgan, 2014) .....	9
Gambar 2. 5 Sensor Ultrasonik HC-SR04.....	9
Gambar 2. 6 Gambar 2.6. Arsitektur <i>Single Shot Detector</i> (Liu et al., 2016) .....	11
Gambar 2. 7 Arsitektur <i>sensor fusion</i> berdasarkan kategori jenis input data .....	13
Gambar 2. 8. (a) <i>random</i> kecil, <i>systematic</i> kecil (b) <i>random</i> kecil, <i>systematic</i> besar (c) <i>random</i> besar, <i>systematic</i> kecil (d) <i>random</i> besar, <i>systematic</i> besar (Taylor, 1997) .....	16
Gambar 3. 1 Diagram Alir .....	17
Gambar 3. 2 Model Purwarupa.....	18
Gambar 3. 3 Indikasi terhubungnya setiap perangkat .....	19
Gambar 3. 4 Area aman yang disekitar <i>ego vehicle</i> .....	20
Gambar 3. 5 Diagram alir Kalman Filter.....	22
Gambar 3. 6 Antarmuka <i>platform</i> LabelImg .....	24
Gambar 3. 7 <i>Validation Loss</i> .....	26
Gambar 3. 8 Diagram Blok Sistem Sensor Fusion .....	29
Gambar 4. 1 Grafik <i>error</i> setiap pengukuran .....	33
Gambar 4. 2 Grafik plot presisi pengukuran tiap jarak .....	34
Gambar 4. 3 Sampel <i>ego-vehicle</i> dan hambatan di lajur kanan.....	35
Gambar 4. 4 FPS <i>ego-vehicle</i> dan hambatan di lajur kanan.....	36
Gambar 4. 5 Sampel <i>ego-vehicle</i> di lajur kanan hambatan di lajur kiri .....	37
Gambar 4. 6 FPS <i>ego-vehicle</i> di lajur kanan dan hambatan di lajur kiri.....	39
Gambar 4. 7 Skenario <i>ego-vehicle</i> di lajur kiri .....	39
Gambar 4. 8 Skenario mobil acak.....	40
Gambar 4. 9 FPS skenario pelatakan hambatan secara acak .....	42

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 2. 1 Spesifikasi Webcam Xiaovv 1080P.....	7
Tabel 2. 2 Spesifikasi Sensor Ultrasonik HC-SR04.....	9
Tabel 3. 1 Pembagian Objek Deteksi .....	23
Tabel 3. 2 Parameter <i>Training</i> SSD .....	24
Tabel 3. 3 Nilai Rata-rata Presisi Deteksi Objek .....	27
Tabel 4. 1 Data mentah pengukuran jarak sensor ultrasonik .....	31
Tabel 4. 2 Hasil pengukuran setelah kalibrasi .....	32
Tabel 4. 3 Nilai parameter <i>ego-vehicle</i> dan hambatan di lajur kanan .....	36
Tabel 4. 4 Nilai parameter <i>ego-vehicle</i> di lajur kanan dan hambatan di lajur kiri.....	38
Tabel 4. 5 Besarnya parameter skenario hambatan acak.....	41

*Halaman ini sengaja dikosongkan*

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi elektronik serta kecerdasan buatan berjalan sangat cepat. Salah satu hasil rekayasa di bidang teknologi transportasi adalah terciptanya *autonomous vehicle*. *Autonomous* sendiri berarti suatu sistem yang mampu untuk mengatur diri sendiri (Antsaklis, 1991). Sehingga *autonomous vehicle* merupakan suatu kendaraan dimana dalam sistem kendaraan tersebut mampu untuk mengoperasikan dirinya sendiri tanpa ada bantuan manusia. Menurut Diaz dkk pada tahun 2019 menyebutkan bahwa *autonomous vehicle* adalah suatu revolusi teknologi tidak hanya di bidang transportasi tapi juga mempengaruhi kehidupan sosial luas. Selain itu, diharapkan dengan adanya teknologi tersebut mampu meningkatkan keamanan dalam lalu lintas kendaraan juga mampu mengurangi kemacetan, sehingga penumpang mendapatkan kenyamanan yang lebih baik saat berkendara (Martínez-díaz et al., 2018).

*Autonomous vehicle* juga memiliki keuntungan selain ditinjau dari faktor keamanan, seperti untuk orang yang tidak mampu untuk mengemudi karena faktor usia ataupun karena disabilitas, maka dengan adanya *autonomous vehicle* orang-orang tersebut tetap mampu berkendara dengan aman secara mandiri. *Autonomous vehicle* juga mampu mengurangi dampak lingkungan akibat konsumsi energi, karena *autonomous vehicle* mampu melakukan penambahan serta pengurangan kecepatan sehingga mampu menghemat bahan bakar (Urmson & Whittaker, 2008).

*Autonomous vehicle* mampu mengurangi kecelakaan yang disebabkan dari faktor manusia baik pengemudi maupun dari pihak luar kendaraan, dengan adanya pengamatan, mendeteksi sekitar serta pengambilan keputusan secara cepat ketika terdapat potensi bahaya. Namun, hal ini tidak menjamin bahwa keberhasilan sistem *autonomous vehicle* untuk mencegah terjadinya kecelakaan setiap saat. Seperti yang diungkapkan oleh Karkoukus pada tahun 2020, bahwa kecelakaan kendaraan dengan teknologi mandiri tetap tidak terhindarkan bahkan hingga menimbulkan korban jiwa baik korban dari pengguna jalan lain seperti pejalan kaki ataupun

korban dari pihak penumpang *autonomous vehicle* (Karnouskos, 2020). *Autonomous vehicle* secara garis besar dibagi menjadi empat bagian utama. Sistem Persepsi, Sistem lokalisasi dan pemetaan, perencanaan jalur dan kontrol (Fayyad et al., 2020). Dikarenakan cakupan bidang yang luas pada *autonomous vehicle* maka pada penelitian ini berfokus pada sistem persepsi pada *autonomous vehicle*.

Sistem persepsi sendiri menurut Fayyad dkk. pada tahun 2020 merupakan sekumpulan sensor yang tertanam untuk mendeteksi kondisi sekitar, seperti batas jalan, kendaraan lain, pedestrian, rambu-rambu dan sebagainya. Hingga saat ini sistem persepsi masih mengalami banyak kendala ketika dihadapkan pada lingkungan yang sangat dinamis. Kondisi seperti berkabut, hujan deras, malam hari, bersalju dan sebagainya akan mempengaruhi performa dari sensor yang terpasang sebagai sistem (Yeong et al., 2021). Maka untuk mengurangi dampak tersebut, penggabungan 2 atau lebih sensor disebut sebagai sensor *fusion* digunakan sebagai sistem persepsi pada *autonomous vehicle*. Pada prinsipnya, sensor *fusion* bekerja dengan menggabungkan setidaknya dua data sensor untuk mendapatkan informasi dari masing-masing untuk mendapatkan kualitas pengukuran yang lebih baik. Hal tersebut memanfaatkan kelebihan dari masing-masing sensor untuk melengkapi kekurangan satu sama lain dengan pengukuran yang berbeda (Wu et al., 2019).

Penelitian tentang sensor fusion telah banyak dilakukan seperti yang dilakukan oleh (Du et al., 2017) menggunakan sensor LIDAR dan sensor kamera untuk mengenali objek kendaraan lain. Penelitian ini menggunakan algoritma PC-CNN (*Point Cloud-Convolutional Neural Network*) sebagai algoritma *deep learning* pada penggabungan kedua sensor tersebut. Gabungan kedua sensor tersebut mampu melakukan pemetaan jarak lingkungan terhadap *ego vehicle* (kendaraan itu sendiri) dengan pembagian jarak kurang dari 15 m, 15-30 meter, dan lebih dari 30 m. Namun, klasifikasi objek dan deteksi posisi *ego vehicle* terhadap jalan belum dimunculkan pada penelitian ini. Selain itu, deteksi visual menggunakan *two stage object detection* yang memiliki tingkat akurasi tinggi namun kemampuan pemrosesan yang lama (Soviany & Ionescu, 2018).

Pada penelitian ini menggunakan melakukan penggabungan sensor antara sensor ultrasonic HC-SR04 dan webcam Xiaovv 1080P sebagai kamera mono, yang tertanam pada purwarupa berupa *Jetracer AI Robot* sebagai *plant* lalu purwarupa

tersebut dilengkapi dengan *Single Board Computer* (SBC) NVIDIA Jetson Nano *board* sebagai CPU (*Central Processing Unit*). Sehingga pada pemrosesannya menggunakan *single-stage object detection* yang lebih cepat pemrosesan data serta lebih ringan. Algoritma *single-stage* yang digunakan oleh penulis adalah SSD (*Single Shot Detector*) yang dikembangkan oleh (Liu, et al., 2016), algoritma ini memiliki nilai *mean average precision* (mAP) lebih tinggi dibandingkan metode *two stage*. Walaupun pada algoritma ini memiliki kekurangan dalam mendeteksi benda yang kecil, namun kekurangan ini tidak mengganggu proses penelitian yang berlangsung.

## 1.2 Perumusan Masalah

Berdasarkan paparan latar belakang, maka permasalahan dalam penelitian ini adalah

- 1) Bagaimana merancang *Single Shot Detector* untuk mengenali dan mengklasifikasi objek?
- 2) Bagaimana merancang *Single Shot Detector* untuk menentukan posisi terhadap jalan?
- 3) Bagaimana merancang sistem sensor *fusion* menggunakan kamera mono dan sensor ultrasonik?

## 1.3 Tujuan Penelitian

Berdasarkan permasalahan tersebut, maka tujuan penelitian ini adalah merancang merancang sistem persepsi menggunakan *Single Shot Detector* untuk mengenali dan mengklasifikasi objek, mengenali posisi terhadap jalan, juga membangun sistem sensor *fusion* mono kamera dan sensor ultrasonic untuk mendapatkan hasil pengukutan yang lebih baik.

## 1.4 Batasan Masalah

Agar penelitian ini tidak terlalu melebar, tanpa mengurangi tujuan penelitian, maka batasan-batasan masalah yang digunakan yaitu sebagai berikut:

- 1) Kendaraan yang digunakan adalah prototipe *electric vehicle* modul *jetson AI Car*

- 2) Objek yang dikenali dan diklasifikasi dua jenis yaitu mobil dan lajur kiri/kanan
- 3) Kontur jalan yang digunakan rata dan lurus dengan perbandingan 1:10
- 4) Sensor yang digunakan adalah kamera mono Webcam Xiaovv 1080P serta sensor jarak Ultrasonik
- 5) Processor yang digunakan adalah SBC NVIDIA Jetson Nano *Board*.
- 6) *Ego Vehicle* dalam keadaan diam
- 7) Pengujian dilakukan dalam rentang 10 sampai 50 cm dari area aman didepan *ego vehicle*
- 8) *Sensor fusion* diterapkan dengan mengenali satu objek

### **1.5 Manfaat Penelitian**

Penelitian merancang sistem persepsi pada *autonomous vehicle* memiliki berbagai macam manfaat baik dari segi ilmu maupun dari penerapannya. Sistem persepsi yang semakin maju dan memiliki akurasi yang tepat akan menyokong teknologi *autonomous vehicle*. Selain itu, dari segi kebermanfaatan teknologi sistem persepsi tidak terbatas pada *autonomous vehicle* namun bisa diperluas penerapannya pada sistem yang berbasis otonom, seperti drone. Kebermanfaatan pengenalan terhadap lingkungan juga dapat diterapkan pada sistem keamanan lingkungan yang menggunakan CCTV sebagai basis keamanannya, hal ini memungkinkan untuk mengenali orang yang keluar masuk sehingga mencegah terjadinya kejahatan.

## **BAB 2**

### **DASAR TEORI**

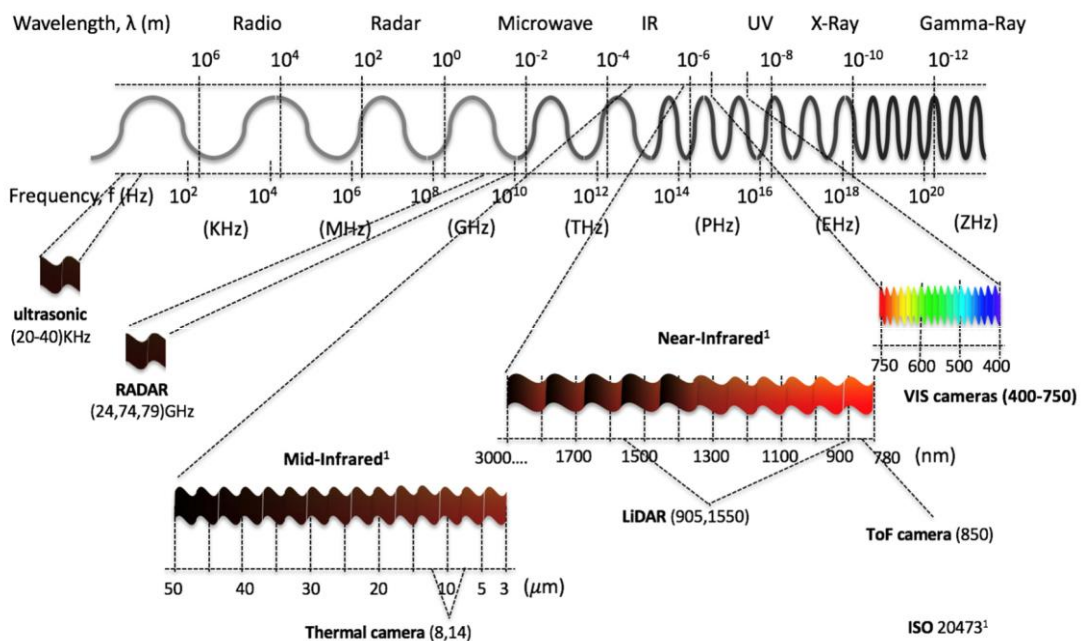
Pada bab ini akan dijelaskan mengenai teori yang digunakan sebagai penunjang dalam pelaksanaan penelitian. Paparan mengenai *Autonomous Vehicle*, Sistem Persepsi, *Object Detection*, *Sensor Fusion*, *Kalman Filter* serta komponen yang digunakan dalam penelitian ini.

#### **2.1 Autonomous Vehicle**

*Autonomous vehicle* merupakan suatu kendaraan tanpa pengemudi juga dikenal dengan istilah *self driving* (Jiang & Astolfi , 2018), ataupun disebut kendaraan pintar *Intelligent Vehicles* (Tagne, Talj, & Charara, 2016). Untuk menghasilkan kendaraan yang mampu untuk berkendara secara sendiri maka perlu mendefinisikan rute yang ingin ditempuh menggunakan Google Maps dari tempat tinggal menuju ke tempat tujuan. Selain itu, diperlukan juga sistem kontrol untuk menunjang keberhasilan kendaraan berjalan sesuai dengan lintasan yang diinginkan, secara garis besar, sistem kontrol terbagi menjadi dua bagian yaitu kontrol longitudinal dan kontrol lateral Kontrol longitudinal merupakan sistem kontrol yang berfokus pada kecepatan kendaraan (Liu, Hou , Tian , Deng , & Li, 2019). Kontrol lateral adalah sistem kontrol yang mengatur kemudi dari kendaraan agar kendaraan tetap berada pada jalur yang telah ditetapkan (Xia, Pu, Li, & Gao, 2016), maupun beroperasi di jalan seperti berpindah jalur ataupun menyalip kendaraan (Huang, Guo, Zhang, Wei, & Wu, 2018). Secara teori kedua sistem kontrol ini berjalan bersama-sama, terutama ketika kendaraan berjalan dengan kecepatan tinggi. Namun, secara umum pendekatan dalam pengembangannya dipisahkan. Hal ini disebabkan yang pertama karena pada kondisi kecepatan normal sistem kontrol kendaraan memiliki struktur yang sederhana, mudah untuk diatur, serta berbiaya rendah. Alasan kedua adalah, hampir semua kendaraan berjalan pada kondisi dibawah kecepatan normal dan bukan kecepatan yang sangat cepat (Liu, Hou , Tian , Deng , & Li, 2019). Oleh karena itu, pada penelitian kali ini akan berfokus pada sistem kontrol lateral pada *autonomous vehicle*.

## 2.2 Sistem Persepsi

Sistem persepsi pada *autonomous vehicle* adalah proses peninjauan terhadap lingkungan yang berada disekitar *autonomos vehicle*. Peninjauan ini melibatkan banyak sensor untuk mendapatkan hasil yang akurat. Proses peninjauan ini meliputi proses deteksi, pemahaman maupun mengintrepretasi lingkungan yang berada disekitar *autonomous vehicle*, baik terhadap hal-hal yang sifatnya statis maupun yang sifatnya dinamis (Fayyad et al., 2020). Hal yang bersifat statis seperti lajur jalan, garis tepi jalan, pepohonan, mobil parkir, rambu-rambu lalu lintas, dan sebagainya. Sedangkan hal-hal yang bersifat dinamis antara lain, mobil yang melaju, pedestrian, sepeda motor dan sebagainya. Lebih lanjut sistem persepsi ini selain melakukan peninjauan terhadap lingkungan, juga diharuskan mampu untuk melakukan peninjauan dari *ego vehicle* (kendaraan otonom itu sendiri) terhadap kendaraan lain seperti memprediksi kecepatan, arah, hingga posisinya.



Gambar 2. 1 Rentang Panjang gelombang sensor yang digunakan pada sistem persepsi (Rosique et al., 2019)

Sehingga sistep persepsi terhadap lingkungan menjadis salah satu inti teknologi dasi sistem *autonomous vehicle* (Duan et al., 2021). Maka dari itu, saat ini sistem pesepsi ditunjang dengan banyaknya sensor untuk meninjau banyaknya variable.

Banyak sensor yang terlibat dalam sistem persepsi diantaranya adalah camera, ultrasonic, RADAR (*Radio Detection and Ranging*), LiDAR (*Light Detection and Ranging*), IMU (*Inertial Measurement Unit*), GNSS (*Global Navigation Satellite Systems*) dan sebagainya (Rosique et al., 2019).

Panjang gelombang sensor yang digunakan untuk sistem persepsi dapat dilihat pada Gambar 2.1. Sistem persepsi terhadap lingkungan menjadi peran yang sangat penting, karena sistem ini akan memberikan informasi kepada *autonomous vehicle* tentang kondisi disekitarnya. Pada penelitian kali ini dua jenis sensor yang akan dibahas adalah sensor camera dan sensor ultrasonik yang pembahasannya sebagai berikut:

### 2.2.1 Sensor Camera

Pada Gambar 2.1 sensor kamera dapat diklasifikasikan sebagai VIS (*visible*) atau IR (*infrared*). kamera VIS memiliki rentang panjang gelombang antara 400 nm hingga 780 nm. Spektrum VIS dibagi menjadi tiga pita atau *channels* yaitu R, G dan B, yang mana masing-masing memiliki kode terpisah disetiap warna tertentu. Sensor kamera menjadi sensor yang hampir selalu digunakan pada setiap deteksi sistem persepsi pada *autonomous vehicle*, dikarenakan harga yang murah, mampu mendeteksi banyak informasi dan resolusi yang tinggi. Kamera VIS sangat dipengaruhi oleh kondisi cahaya yang bervariasi, seperti hujan, berkabut, hingga kondisi petang. Hal ini lah yang menjadi dasar sensor kamera seringkali dikombinasikan dengan sensor lain seperti RADAR, LiDAR dan Ultrasonik untuk meningkatkan hasil persepsi dari lingkungan (Rosique et al., 2019).

Penelitian kali ini menggunakan kamera Webcam Xiaovv 1080P sebagai sensor mono kamera. Kamera ini sebagaimana Gambar 2.2 memiliki spesifikasi sebagaimana dituliskan pada Tabel 2.1.

Tabel 2. 1 Spesifikasi Webcam Xiaovv 1080P

Spesifikasi	Nilai
Resolusi	1080P
<i>Wide-angle view</i>	150 Derajat
Ukuran	100 mm x 25 mm x 50 mm
Temperatur kerja	-10°C – 50°C
Daya masukan	5 V



Gambar 2. 2 Webcam Xiaovv 1080P

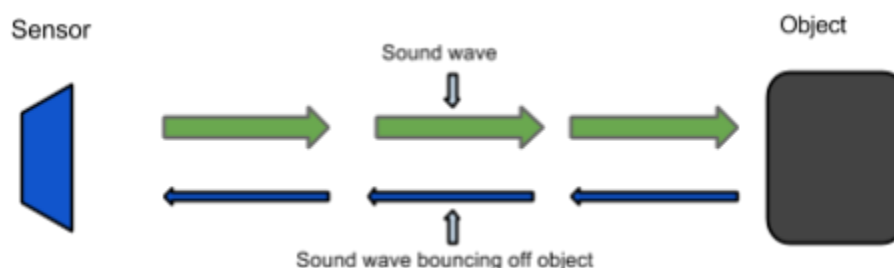
### 2.2.2 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja pada rentang Panjang gelombang ultrasonik yaitu, 20 kHz hingga 40 kHz. Prinsip kerja dari sensor ultrasonik yaitu mengukur waktu yang ditempuh dimulai ketika gelombang ultrasonik dipancarkan (*emitting*) sampai gelombang tersebut diterima kembali (*receiving*). Waktu ini disebut dengan *time of flight* (ToF). Proses perhitungan secara matematis dapat dituliskan berdasarkan persamaan berikut (Rosique et al., 2019).

$$d = \frac{c}{2} \times ToF \quad (2.1)$$

Dimana  $c$  adalah kecepatan gelombang tiap detik sedangkan  $d$  adalah jarak yang terukur.

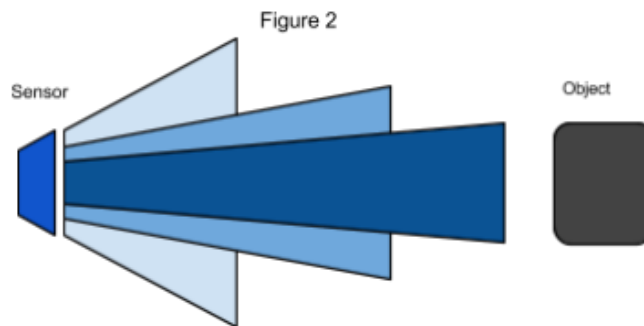
Sensor yang digunakan pada penelitian kali ini adalah sensor ultrasonik tipe HC-SR04. Sensor ini memiliki 2 pemancar (*Transmitter*), satu transmitter digunakan untuk memancarkan gelombang (*emitting*) sedangkan satunya digunakan menerima gelombang tersebut (*receiving*) (Al Tahtawi, 2018).



Gambar 2. 3 Proses kerja Sensor HC-SR04 (Morgan, 2014)

Gelombang yang dipancarkan terlihat pada Gambar 2.3 merupakan gelombang suara yang tergantung dari berbagai kondisi atmosfer udara saat itu, seperti temperatur, kelembapan dan tekanan (Morgan, 2014). Kemampuan sensor

ultrasonik dalam mendeteksi juga dipengaruhi oleh posisi benda terhadap sensor, hal ini dikarenakan sinyal yang dipancarkan berbentuk kerucut yang membentuk sudut. Jika objek berada diluar sudut tersebut maka deteksi jarak tidak dapat dilakukan, seperti yang terlihat pada Gambar 2.4.



Gambar 2. 4 Sudut dari sensor Ultrasonik (Morgan, 2014)

Berdasarkan buku manual (RTI, 2010) sensor ultrasonic memiliki spesifikasi yang tertera pada Tabel 2.2.

Tabel 2. 2 Spesifikasi Sensor Ultrasonik HC-SR04

Spesifikasi	Nilai
Masukan Daya	+5V DC
Arus Diam	< 2 mA
Arus Kerja	15 mA
Jarak Pengukuran	2 cm – 400 cm
Resolusi	0.3 cm
Sudut pengukuran	30 Derajat
Input lebar sinyal pulse	10 $\mu$ S
Dimensi	45 mm x 20 mm x 15 mm



Gambar 2. 5 Sensor Ultrasonik HC-SR04

### 2.3 Object Detection

Deteksi object (*object detection*) merupakan suatu proses pengenalan suatu objek pada sebuah bingkai (*frame*), baik bingkai tunggal atau gambar, maupun bingkai yang tersusun berurutan yang disebut sebagai video. Untuk

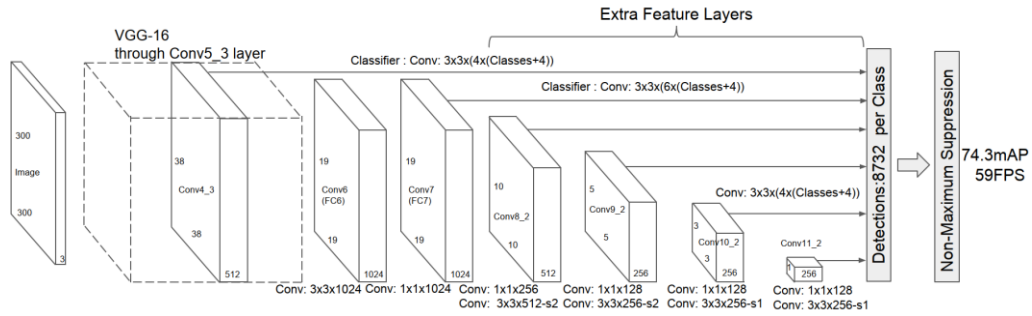
mendeteksi suatu objek berarti bahwa dalam bingkai secara jelas mengindikasikan adanya objek tersebut dan lebih lagi bahwa lokasi objek tersebut harus jelas tergambar dalam bingkai (Appel et al., 2014). Sehingga *object detection* juga dapat didefinisikan sebagai cara untuk melokasikan secara langsung objek yang ada pada kondisi sebenarnya didalam bingkai. Deteksi juga berkaitan erat dengan klasifikasi objek, hal ini dikarenakan proses deteksi melibatkan kehadiran dan posisi objek tertentu terhadap frame (Masita et al., 2020). Dalam *Object detection* saat ini, memprediksi lokasi dari objek tertentu pada bingkai menjadi salah satu topik yang paling penting pada *computer vision*. Menurut Soviany & Ionescu pada tahun 2018 terdapat dua jenis dari keterbaruan algoritma deteksi objek. Pertama disebut *two-stage detector*, algoritma pada *detector* ini seperti Faster R-CNN (*Region-based Convolutional Neural network*) atau *Mask R-CNN*. Untuk yang kedua disebut sebagai *single-stage detectors* seperti YOLO (*You Only Look Once*) dan SSD (*Single Shot Multibox Detector*), yang mendeteksi objek seperti regresi sederhana dengan mengambil masukan gambar dan mempelajari kemungkinan kelas yang muncul serta koordinat *bounding box* (Soviany & Ionescu, 2018). Pada penelitian kali ini menggunakan *single stage detector* dengan arsitektur SSD sebagai *detectornya*. Hal yang berkaitan dengan SSD akan dipelajari lebih lanjut dibawah ini.

### **2.2.3 Single Shot Multibox Detector (SSD)**

SSD adalah *one stage detector* yang dikarakteristikan oleh penggunaan dari *multiscale feature maps* dan *default boxes* (Liu et al., 2016). *Multiscales feature maps* penambahan fitur layer konvolusi pada akhir dari jaringan dasar yang terpotong. Layer ini akan menurunkan secara progresif ukuran dari gambar dan dapat memprediksi dari pendeteksian pada *multiple scales*. *default boxes* merupakan kumpulan dari *bounding box* yang memiliki aspek rasio dan skala per *feature maps* berbeda tiap *bounding box*. Penggunaan dari *default box* untuk menyederhanakan masalah deteksi untuk mengklasifikasi dan meregresi masalah dari *boxes*, dengan kata lain menghindari ekstraksi dari ROI (*Region of Interest*). Mempelajari SSD secara garis besar terdiri dari mempelajari desain arsitektur dan *training nya* (Yang et al., 2019).

### 2.2.3.1 Arsitektur

Arsitektur dari SSD terdiri dari dua bagian. Yang pertama adalah jaringan dasar yang digunakan sebagai fitur ekstraksi pre-eliminasi. VGG-16 (Simonyan & Zisserman, 2015) merupakan jaringan dasar yang digunakan pada arsitektur SSD seperti yang tertera pada Gambar 2.6 berikut ini.



Gambar 2. 6 Gambar 2.6. Arsitektur *Single Shot Detector* (Liu et al., 2016)

### 2.2.3.2 Training

Selama *training* SSD membutuhkan untuk menentukan yang mana *default boxes* yang bertanggung jawab terhadap pendeteksian *ground truth* dan melatih jaringan yang sesuai. Setiap *ground truth box* SSD menyeleksi dari *default boxes* yang bervariasi terhadap lokasi, aspek rasio dan skala. SSD *training objective* diturunkan dari *Multibox Objective* namun diperluas untuk mengatasi kategori objek yang banyak. Fungsi objektif kerugian secara keseluruhan adalah jumlah pembobotan dari kerugian *localization* (loc) dan kerugian *confident* (conf) (Liu et al., 2016) :

$$L_{loc}(x, c, l, g) = \frac{1}{N} \left( L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \quad (2.2)$$

Dimana  $N$  adalah jumlah *default boxes* yang cocok. Jika  $N = 0$ , maka kerugian akan jadi 0 juga. Maka diperoleh fungsi objektif kerugian terhadap *confident* dan *localization* berturut-turut adalah sebagai berikut (Yang et al., 2019):

$$L_{conf}(x, c) = -\sum_{i \in pos} x_{i,j}^1 \log(\hat{c}_i) - \sum_{i \in neg} \log(\hat{c}_i) \quad (2.3)$$

$$L_{loc}(x, l, g) = \sum_{i \in pos} \sum_{m \in \{cx, cy, w, h\}} x_{i,j}^k |l_i^m - g_j^m| \quad (2.4)$$

Dimana  $x$  adalah indicator untuk mencocokkan *default boxes* dengan *ground truth boxes*.  $l$  menunjukkan *predicted boxes* dan  $g$  menunjukkan *ground truth boxes*.  $cx$  dan  $cy$  adalah nilai koordinat titik tengah (*center*) dari boxes.  $w$  dan  $h$  masing

masing lebar dan tinggi dari *boxes*. Secara keseluruhan fungsi kerugian (*loss function*) terdiri kerugian *confident* dan kerugian *localization*.

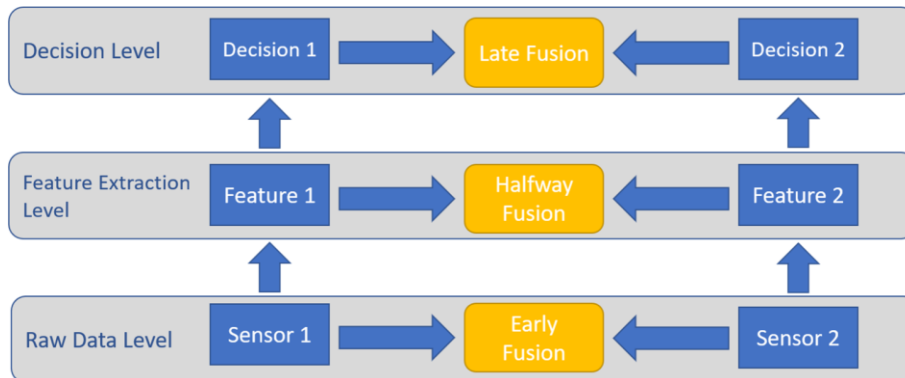
## 2.4 Sensor Fusion

*Sensor fusion* adalah sebuah proses dimana menggunakan informasi dari beberapa sensor untuk menghitung hasil estimasi keadaan dari suatu sistem dinamik. Menghasilkan estimasi dalam beberapa hal lebih baik hasilnya daripada ketika sensor digunakan sendiri-sendiri (Lundquist, 2011). *Sensor fusion* juga dapat didefinisikan sebagai proses mengelola dan menanganu data dan informasi yang berasal dari beberapa tipe sumber, dengan tujuan untuk meningkatkan beberapa kriteria spesifik dan data aspek untuk pengambilan keputusan. Pada prosesnya penggabungan (*fusion*) data terdiri dari mengkombinasikan keluaran dari masing-masing sensor atau keluaran dari algoritma spesifik seperti vector keadaan dan matriks ketidak pastian untuk menghasilkan capaian kombinasi baru yang telah diperkuat, diperluas. Diperkaya, lebih dapat diandalkan, lebih meyakinkan dan lebih kokoh daripada hasil yang dihasilkan oleh masing-masing sensor secara terpisah (Fayyad et al., 2020).

Beberapa skema kategori dari metode yang ada untuk *sensor fusion* diantaranya: pertama dikategorikan berdasarkan jenis input yang digunakan pada jaringan gabungan. Kategori pertama ini diawali oleh *data fusion (early fusion)* dimana gabungan sensor terletak pada level *raw data*. Kedua adalah *feature fusion (halfway fusion)*, dimana fitur pertama kali diekstrak dari data sensor kemudian fitur tersebut digabungkan ditengah – tengah jalan ketika melewati jaringan. Terakhir pada kategori ini adalah *decision fusion (late fusion)*, dimana beberapa pengklasifikasi digunakan untuk menghasilkan keputusan yang kemudian dikombinasikan sebagai bentuk keputusan akhir. Kategori ini dapat dilihat pada Gambar 5 (Fayyad et al., 2020).

Kategori kedua berdasarkan sumber penggabungan data, *sensor fusion* dikategorikan sebagai *multimodal fusion sensor* (Feng et al., 2021), dimana data gabungan diperoleh dari dua atau lebih jenis sensor yang berbeda. Menggabungkan LiDAR dan Kamera salah satu contohnya, dimana masing-masing sensor melengkapi fungsi satu sama lainnya dan memperoleh keluaran yang lebih baik. Kemudian dalam kategori ini juga ada yang disebut sebagai *multitemporal fusion*,

dimana data diperoleh dari sensor yang sama namun memiliki waktu data akuisisi yang berbeda. Dan yang ketiga adalah *multifocus fusion* (Malviya & Bhirud, 2009), dimana gambar diperoleh dari perbedaan nilai *focal length*. Terakhir adalah *multispectral fusion* (Guan et al., 2019) yang mana gambar ditangkap dari perbedaan panjang gelombang sensor seperti kamera RGB dan kamera thermal. Tipe ini sering dijumpai pada deteksi dan pengenalan objek pedestrian.



Gambar 2. 7 Arsitektur *sensor fusion* berdasarkan kategori jenis input data

Selanjutnya, kategori berdasarkan konfigurasi sensor, *sensor fusion* dapat dikategorikan menjadi konfigurasi saling melengkapi (*complementary*), yang mana dua independen sensor digunakan kemudian keluaran dari masing-masing sensor dikombinasikan untuk melengkapi satu sama lain. Berikutnya *sensor fusion* konfigurasi kompetitif (*competitive*) atau disebut juga konfigurasi berulang (*redundant*), dimana beberapa sensor digunakan untuk mengukur properti yang sama dan keluarannya digunakan untuk tujuan mengkoreksi. *Sensor fusion* berdasarkan konfigurasi terakhir adalah konfigurasi kerjasama (*corporative*) dimana dua atau lebih sensor digunakan untuk memperoleh keluaran yang tidak dapat diperoleh jika hanya mengandalkan sensor secara individual, seperti mengintegrasikan *stereo camera* untuk mendapatkan *3D depth image*.

Kategori terakhir ada *sensor fusion* adalah berdasarkan arsitektur gabungannya. Dibagi menjadi sentralisasi (*centralized*), desentralisasi (*decentralized*) dan gabungan (*hybrid*) (Gruyer et al., 2017). Pada arsitektur sentralisasi semua data dari sensor berbeda dikumpulkan pada *central processing unit*. Setelah itu semua data diperoleh, unit pusat akan menerima keluaran sebagai satu sumber dari informasi untuk menggabungkannya. Pada arsitektur

desentralisasi data diperoleh dari sensor yang diproses secara local, kemudian output yang diperoleh diteruskan kebagian pusat untuk digabungkan. Sedangkan arsitektur terkahir termasuk didalamnya proses secara local kemudian diteruskan pada prosessor pusat, dimana data tersebut diproses kembali dan digabungkan.

### 2.3 Filter Kalman

Filter Kalman pertama kali dikembangkan oleh Rudolf E. Kalman pada Tahun 1960 (Kalman, 1960) . Fikter kalman ini merupakan sebuah algoritma untuk estimasi dari *variable unobservable state* yang berdasarkan variabel *observable* yang terdapat beberapa kesalahan pengukuran (Urrea & Agramonte, 2021). Filter Kalman adalah algoritma yang membutuhkan dua jenis persamaan: yang pertama menghubungkan variabel keadaan (*state variable*) ke variabel yang diamati (*observable variable*)/variable utama. Sedangkan yang lain menentukan struktur sementara variabel keadaan. Filter Kalman menjadi salah satu algoritma yang populer untuk mengestimasi secara optimal keadaan tidak tentu atau proses dari sebuah sistem dinamik (Song et al., 2014). Keuntungan dari Filter Kalman sebagai estimators telah diketahui secara luas, yaitu Filter Kalman tidak membutuhkan seluruh data lampau untuk mengestimasi, namun hanya membutuhkan pengukuran saat ini dan kalkulasi data sebelumnya (Kim et al., 2022).

Estimasi dari variabel keadaan dipengaruhi berdasarkan dinamisasi dari variabelnya (dalam dimensi waktu), serta pada pengukuran dari *observable variables* diperoleh pada setiap waktu instan (transversal). Dengan kata lain, dinamisasi diringkas menjadi dua tahap. Pertama estimasi dari variabel keadaan menggunakan dinamisasi dari sistem itu sendiri (tahap prediksi). Kedua memberikan peningkatan dari estimasi pertama menggunakan informasi yang diberikan oleh *observable variables* (tahap koreksi). Karakteristik yang menarik dari metode ini adalah proses pengulangan, yang mana menyiratkan bahwa metode ini dapat digunakan secara *real time*. Setelah algoritma memprediksi keadaan baru pada moment  $t$ , hal ini menambahkan sebuah keadaan koreksi, dan keadaan terkoreksi baru berfungsi sebagai kondisi awal pada tahap berikutnya,  $t + 1$ . Pada kondisi ini, estimasi dari *variable keadaan* menggunakan semua informasi yang

tersedia sampai momen tersebut dan tidak hanya dari tahap sebelum estimasi (hal ini diketahui sebagai ekstraksi sinyal (*signal extraction*)).

Filter Kalman Linier merupakan salah satu dari variasi Filter Kalman yang dikenal juga sebagai variasi paling dasar dari Filter Kalman (Kim et al., 2022). Model *state-space* dari Filter Kalman Linear dapat ditunjukkan sebagai berikut:

$$x_k = Ax_{k-1} + w_k \quad (2.5)$$

Dimana  $x_k$  adalah *state vector* saat waktu  $k$ .  $A$  adalah matriks transisi keadaan (*state transition matrix*), dan  $w_k \sim \mathbb{N}(0, Q_k)$  adalah proses gangguan yang diasumsikan sebagai *Gaussian noise* dengan sebuah kovarian matriks yang diketahui  $Q_k$ .

Model pengukuran dari Filter Kalman Linier dapat dituliskan sebagai berikut:

$$z_k = Hx_k + v_k \quad (2.6)$$

dimana  $z_k$  adalah matriks vector pengukuran pada waktu  $k$ ,  $H$  adalah matriks pengukuran, dan  $v_k \sim \mathbb{N}(0, R_k)$  adalah gangguan pengukuran diasumsikan sebagai *Gaussian noise* dengan sebuah matriks kovarian yang diketahui  $R_k$ .

Filter Kalman Linier dapat dibagi menjadi dua tahapan utama, tahap prediksi dan tahap memperbaharui. Pertama, pada tahap prediksi, estimasi keadaan terprediksi (*the predicted state*) dan matriks kovarian kesalahan terprediksi (*the predicted error covariance matrix*) dapat dituliskan sebagai berikut:

$$\hat{x}_k | x_{k-1} = A\hat{x}_{k-1} \quad (2.7)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_k \quad (2.8)$$

dimana  $\hat{x}_k | x_{k-1}$  adalah estimasi keadaan terprediksi (*the predicted state estimation*) saat waktu  $k$  dengan pengukuran pada waktu  $k - 1$ , kemudian  $P_{k|k-1}$  adalah matriks kesalahan kovarian terprediksi.

Pada tahap memperbaharui (*update step*), Filter Kalman akan memperbaharui *Kalman gain*, vector keadaan dan kovariannya, sebagaimana dituliskan sebagai berikut:

$$y_k = z_k - H\hat{x}_{k|k-1} \quad (2.9)$$

$$S_k = HP_{k|k-1}H^T + R_k \quad (2.10)$$

$$K_k = P_{k|k-1}H_k^T S_k^{-1} \quad (2.11)$$

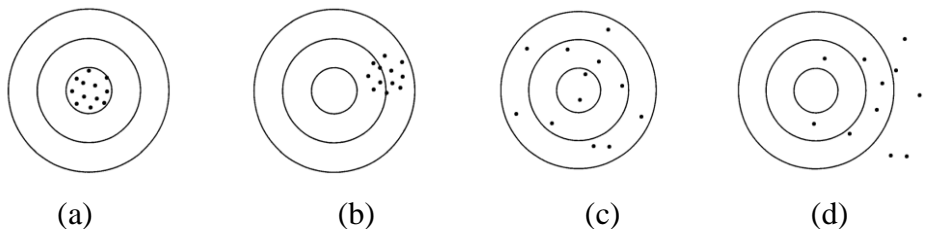
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \quad (2.12)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (2.13)$$

dimana  $y_k$  adalah residual pengukuran,  $S_k$  matriks kovarian perubahan,  $K_k$  adalah *Kalman gain*,  $\hat{x}_{k|k}$  estimasi keadaan yang telah terbaru, dan  $P_{k|k}$  adalah matrik kovarian kesalahan yang telah terbaru.

## 2.5 Error Pengukuran

*Error* pada sistem pengukuran juga dapat didefinisikan sebagai ketidakpastian yang tidak dapat dihindari yang terjadi pada semua sistem pengukuran (Taylor, 1997). Identifikasi *error* dilakukan sebagai dasar proses kalibrasi yang akan dilakukan. *Error* menurut (Taylor, 1997) terbagi menjadi dua, yaitu *error* yang didapatkan melalui proses berulang dinamakan dengan *random error*, sedangkan *error* yang terjadi disebabkan oleh sistem juga tidak dipengaruhi oleh proses pengukuran berulang disebut dengan *systematic error*. Berdasarkan hal ini, maka *error* yang menyebabkan ketidakakuratan dalam pengukuran disebut dengan *systematic error*. Sedangkan *error* yang menyebabkan proses pengukuran tidak presisi disebut dengan *random error* (Bevington et al., 1993). Gambar 2.8 menunjukkan pengukuran dengan kedua *error* tersebut.



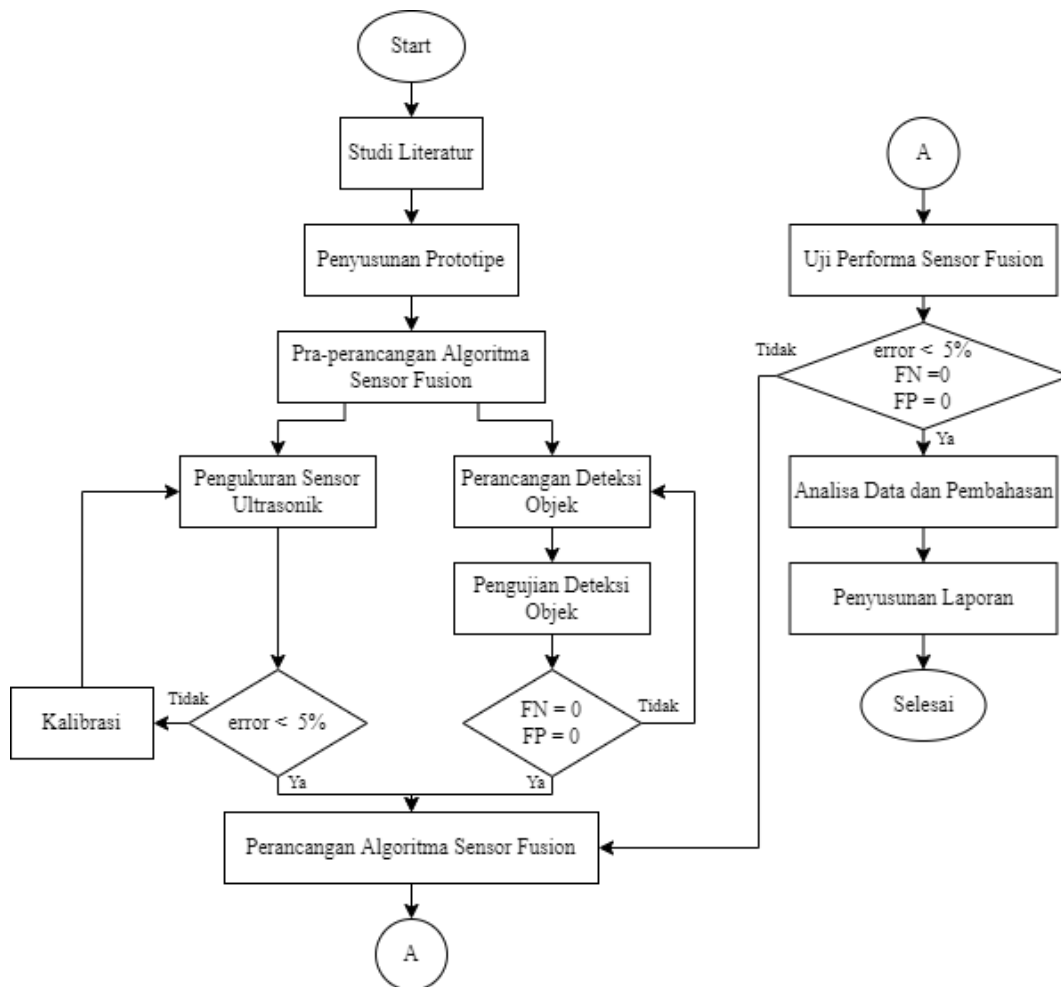
Gambar 2. 8. (a) *random* kecil, *systematic* kecil (b) *random* kecil, *systematic* besar (c) *random* besar, *systematic* kecil (d) *random* besar, *systematic* besar (Taylor, 1997)

*error* ini dapat dikurangi dengan berbagai cara dan metode sesuai dengan jenis dari *error* tersebut. *Random error* dapat dikurangi dengan melakukan pendekatan secara statistik, penyesuaian dalam komputasi, penambahan komponen pada alat, dsb. Sedangkan *systematic error* dapat dikurangi dengan memeriksa kondisi instrumen ukur, melakukan penyesuaian di laboratorium pengukuran terkait untuk menyesuaikan dengan standar yang benar disebut juga dengan mengkalibrasi instrumen tersebut.

## BAB 3

### METODE PENELITIAN

Pada bab ini berisi tahapan yang dilakukan dalam penelitian. Tahap penelitian meliputi studi literatur mengenai teori-teori yang dibutuhkan pada penelitian, penyusunan prototipe, tahap pra-perancangan, perancangan algoritma sensor *fusion*, pengujian dan penyusunan laporan.



Gambar 3. 1 Diagram Alir

#### 3.1 Studi Literatur

Studi literatur digunakan untuk membangun pemahaman awal tentang metode konsep, serta teori yang menunjang dalam pelaksanaan penelitian. *Autonomous vehicle*, *object detection* menggunakan *single shot detector* dalam mengenali dan mengklasifikasikan objek serta membangun sensor *fusion* untuk

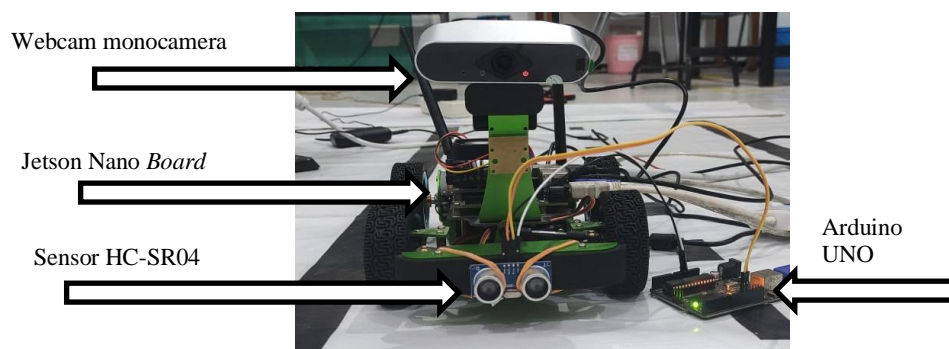
mendeteksi lingkungan lebih baik. Literatur yang dapat digunakan yaitu buku teks, jurnal, artikel ilmiah, prosiding seminar, laporan tesis, dan laporan disertasi yang berkaitan dengan penelitian yang sejenis.

### 3.2 Penyusunan Prototipe

Penelitian kali ini akan menggunakan prototipe JetRacer AI Kit dengan *single board computer* (SBC) Jetson Nano B01, Arduino UNO, Kamera webcam Xiaovv 1080p, serta HC-SR04 Ultrasonic Sensor.

Penyusunan purwarupa dilakukan dengan perakitan sasis JetRacer AI Kit sesuai dengan panduan perangkaian<sup>1</sup>. Berikutnya pemasangan perangkat keras lainnya seperti JetsonNano B01 lalu Arduino UNO sebagai modul untuk membaca sensor ultrasonic HC-SR04 secara *realtime* dan Webcam Xiaovv 1080p sebagai sensor mono kamera, detail *set up* prototipe dapat dilihat pada Gambar 3.2.

Sensor ultrasonik dihubungkan dengan Arduino UNO dengan cara menghubungkan *trigger pin* pada sensor ke PIN 11 dan *echo pin* ke PIN 12 menggunakan kabel *jumper*. Selain itu PIN VCC dan GND sensor HC-SR04 juga dihubungkan pada Arduino UNO yang memiliki sumber daya keluaran 5 V. pada Transfer data antara Arduino UNO dengan SBC Jetson Nano B01 dilakukan dengan menghubungkan USB (*Universal Serial Bus*) *Type B connector* ke input USB *type A* pada SBC Jetson Nano B01, selain itu koneksi ini juga digunakan sebagai sumber daya Arduino UNO. Webcam Xiaovv 1080p dipasangkan pada input USB *type A* SBC Jetson Nano B01, hal ini difungsikan sebagai transfer data dari sensor juga sebagai sumber daya pada sensor.

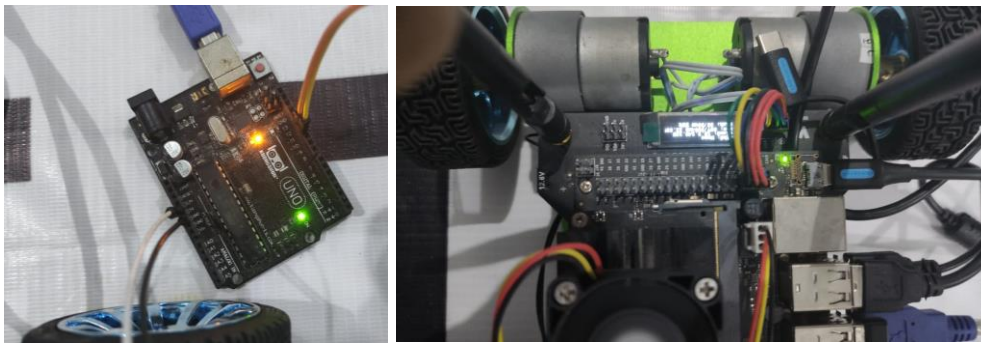


Gambar 3. 2 Model Purwarupa

<sup>1</sup> [https://www.waveshare.com/wiki/JetRacer\\_Assembly\\_Manual](https://www.waveshare.com/wiki/JetRacer_Assembly_Manual)

### 3.3 Pra-perancangan algoritma sensor fusion

Tahapan ini dilakukan proses pengecekan kondisi dari masing-masing sensor, ditandai dengan setiap komponen pada penelitian ini terhubung dan terindikasi telah tersambung dengan baik. Hal ini ditandai dengan setiap perangkat dapat menyala dan dideteksi oleh SBC Jetson Nano B01. Indikasi terhubungnya perangkat dapat dilihat pada lampu LED yang menyala kuning untuk Arduino UNO, kemudian lampu LED merah pada Webcam Xiaovv 1080p, seperti yang tertera pada Gambar 3.2 dan Gambar 3.3 .



Gambar 3. 3 Indikasi terhubungnya setiap perangkat

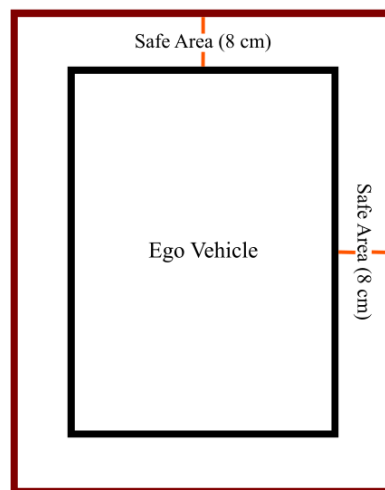
### 3.4 Pengukuran Sensor ultrasonik

Pengukuran jarak menggunakan sensor HC-SR04 dilakukan dengan menghubungkan sensor ke Arduino UNO. Sebagaimana (Morgan, 2014) menyebutkan bahwa *trigger* pada HC-SR04 berfungsi untuk memberikan transmitter tegangan *high* (5 V) selama 10  $\mu$ s, yang setidaknya akan menghasilkan 8 siklus gelombang dengan frekuensi 40 kHz. Kemudian *echo* berfungsi untuk menerima informasi dari *receiver* sensor, yaitu lama selang waktu antara gelombang yang ditransmisikan hingga diterima. Selang waktu ini menghasilkan keluaran *echo pin* dalam keadaan *high* (5 V). Proses konversi dari informasi waktu dengan jarak dilakukan menggunakan perhitungan matematis pada Persamaan (2.1). Secara spesifik konversi waktu menjadi jarak dalam cm dapat dituliskan sebagai berikut (RTI, 2010):

$$d = t/58 \quad (3.1)$$

dengan  $d$  adalah besar jarak dalam cm dan  $t$  adalah selang waktu dalam  $\mu$ s (mikrosekon).

Pada tahapan ini dilakukan proses penyesuaian, validasi dan kalibrasi sensor. Proses penyesuaian dilakukan dengan menambahkan jarak aman, yaitu 8 cm mengelilingi *ego vehicle* seperti pada Gambar 3.4, sehingga titik nol berada 8 cm dari depan *ego vehicle*. Hal ini dilakukan sebagai tindakan pencegahan agar jika terdapat kegagalan dalam sistem, *ego vehicle* masih memiliki jarak aman untuk bermanuver. Proses validasi meliputi pengukuran dalam beberapa skenario serta pengulangan dalam pengukuran untuk melihat efektivitas dari sensor HC-SR04 dalam deteksi jarak pada penelitian ini. Selisih nilai hasil pengukuran HC-SR04 dan nilai sebenarnya disebut juga nilai kesalahan atau *error*.



Gambar 3. 4 Area aman yang disekitar *ego vehicle*

Proses pengukuran berulang pada proses validasi kemudian dianalisa untuk menentukan proses kalibrasi yang akan dilakukan. Analisa ini terutama mengukur nilai kesalahan dari hasil pengukuran. Pada penelitian kali ini dilakukan maksimum selisih yang ditoleransi adalah 5% dari setiap skenario. Hal ini berdasarkan titik terjauh yang diukur menggunakan sensor HC-SR04 adalah 50 cm, sehingga maksimum yang diperbolehkan adalah 2,5 cm dengan mempertimbangkan kemampuan manuver dari *ego vehicle* yang kurang lebih mendapatkan jarak aman bermanuver menjadi 10 cm jika ditambahkan *safe area* pada Gambar 3.4.

Proses kalibrasi menggunakan pendekatan secara statistik untuk meningkatkan akurasi juga presisi dari pengukuran. Pendekatan yang digunakan

pada penelitian kali ini menggunakan metode *curve fitting*. *Curve Fitting* merupakan proses untuk membangun sebuah kurva atau fungsi matematika yang disesuaikan pada sebuah data titik dalam kurva (data diskrit) (Molugaram & Rao, 2017). Pada penelitian kali ini digunakan alat bantu yang terdapat pada LibreOffice Calc pada sistem operasi Linux Ubuntu 20.04 untuk memperoleh kurva yang sesuai dengan data diskrit hasil pengukuran dalam beberapa skenario dan menghasilkan persamaan matematika dari kurva tersebut.

Kurva yang dihasilkan pada bidang kartesius membandingkan antara nilai terukur dari sensor HC-SR04 yang diwakili sumbu x dan nilai yang diinginkan pada sumbu y. Berdasarkan pemrosesan pada LibreOffice Calc diperoleh Persamaan 3.2

$$y = 3,0183 \cdot 10^{-5}x^3 + 2,972 \cdot 10^{-3}x^2 + 0,954x + 0.216 \quad (3.2)$$

Dengan y adalah hasil pengukuran jarak setelah melalui proses kalibrasi dan x adalah data mentah yang dihasilkan sensor HC-SR04. Proses kalibrasi dilakukan didalam software pada Arduino UNO.

### 3.4.1 Kalman Filter pada Sensor Ultrasonik

*Kalman filter* pada sensor ultrasonik mengacu pada Persamaan (2.7) – Persamaan (2.13). Sensor ultrasonik merupakan sensor sistem linier dengan model orde satu sehingga parameter yang digunakan adalah sebagai berikut (Al Tahtawi, 2018).

$$A = [1] \quad (3.3)$$

$$H = [1] \quad (3.4)$$

Kemudian untuk parameter *noise covariant* ditentukan sebagai berikut.

$$Q = [1e - 5] \quad (3.5)$$

$$R = [2,92e - 3] \quad (3.6)$$

Persamaan (3.3) sampai Persamaan (3.6) dimasukkan kedalam pada Persamaan (2.7) – Persamaan (2.13) maka diperoleh persamaan *Kalman Filter* untuk sensor ultrasonik adalah sebagai berikut

$$\hat{x}_k | x_{k-1} = A \hat{x}_{k-1} | x_{k-1} + K_k (y_k - H \hat{x}_{k-1} | x_{k-1}) \quad (3.7)$$

$$P_{k|k-1} = A P_{k-1|k-1} A^T + Q_k$$

$$P_{k|k-1}$$

$$P_{k|k} = A P_{k-1|k-1} A^T + Q_k - K_k P_{k-1|k-1} H^T (H P_{k-1|k-1} H^T + R)^{-1} H P_{k-1|k-1} A^T + Q_k \quad (3.8)$$

$$y_k = z_k - \hat{x}_{k|k-1} \quad (3.9)$$

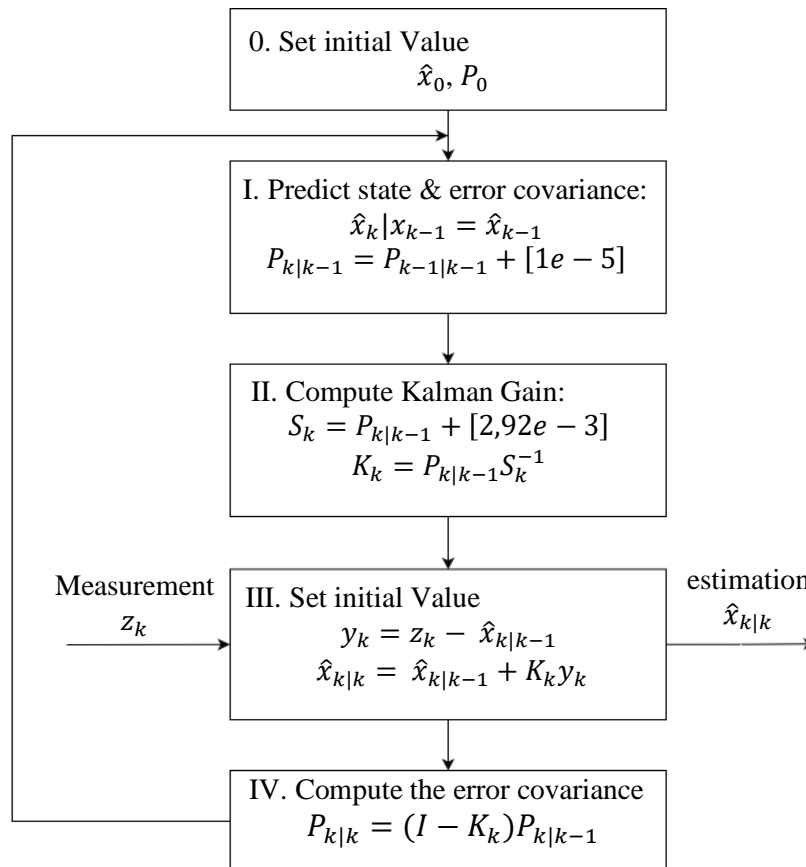
$$S_k = P_{k|k-1} + [2,92e - 3] \quad (3.10)$$

$$K_k = P_{k|k-1}S_k^{-1} \quad (3.11)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \quad (3.12)$$

$$P_{k|k} = (I - K_k)P_{k|k-1} \quad (3.13)$$

Dengan nilai  $z_k = x_k$ . Proses *Kalman filter* ini dapat dilihat pada Gambar 3.5, dimana input pengukuran  $z_k$  yang kemudian akan menghasilkan nilai estimasi  $\hat{x}_{k|k}$  sebagai nilai yang telah diproses dengan peningkatan nilai akurasi dan presisi dari sistem pengukuran jarak ultrasonic.



Gambar 3. 5 Diagram alir Kalman Filter

### 3.5 Object Detection

#### 3.5.1 Perancangan Object Classification

Pada tahap *object classification* ini digunakan SSD sebagai metodenya. Klasifikasi objek secara umum dibagi dua yaitu posisi terhadap jalan dan hambatan, sehingga pada *object detection* dibagi menjadi 4 kelas, yaitu lajur kiri, lajur kanan,

mobil dan “lainnya”. Tingkat keberhasilan dari pengenalan objek ini diukur banyaknya *false begative* dan *false positive* dalam mengklasifikasi objek (Guo et al., 2018). Semakin sedikit nilai *false positive* maka hasil pengenalan serta klasifikasi objek akan semakin baik begitu, kecuali untuk kategori “lainnya” jika semakin banyak *false negative* maka semakin baik karena semua objek dapat dikenali. Proses deteksi objek dapat dilihat pada proses sebagai berikut:

### 3.5.1.1 Pembuatan Data Set dari Objek

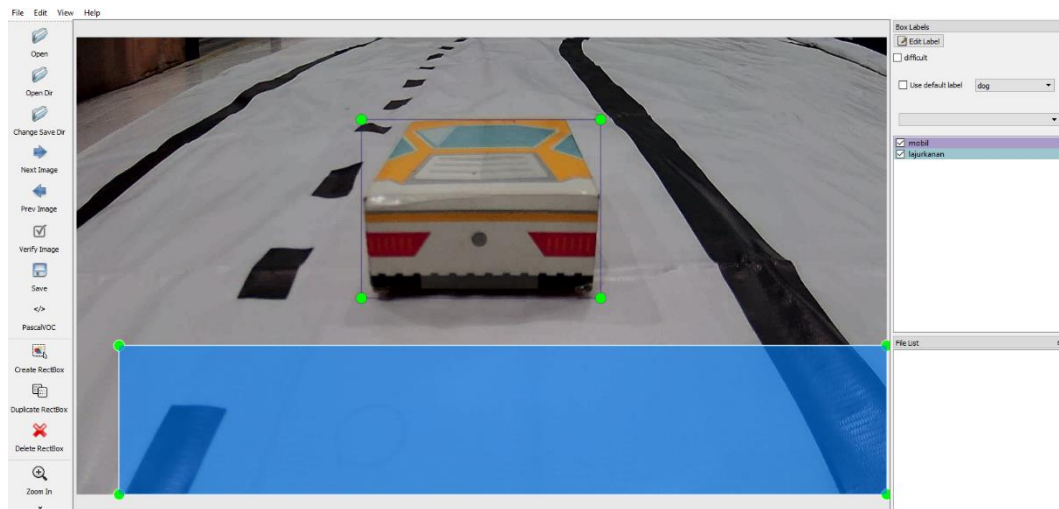
Pembuatan data set terdiri dari 100 gambar sebagai data training dan data testing pada arsitektur SSD. Jumlah ini disesuaikan dengan banyaknya jenis klasifikasi objek yang terdiri dari 4 klasifikasi objek yaitu, lajurkiri, lajurkanan, mobil dan “lainnya”. Pembagian gambar *training* sebesar 80% dengan data *testing* sebesar 20% seperti yang tertera pada Tabel 3.1.

Tabel 3. 1 Pembagian Objek Deteksi

Jenis Data	Jumlah
<i>Training Set</i>	80 Gambar
<i>Testing Set</i>	20 Gambar
Total Gambar	100 Gambar

### 3.5.1.2 Proses *labelling dataset*

*Labeling* (pelabelan) atau anotasi dilakukan untuk menghasilkan informasi berupa penamaan kelas dari objek target. Tahap ini dilakukan ketika data akuisisi berupa gambar banyak telah diambil. Proses pelabelan objek yang akan dideteksi dengan memberi blok pada seluruh gambar objek dari ujung hingga ujung. Proses pelabelan ini berlaku untuk seluruh gambar sesuai yang diinginkan untuk dideteksi. Proses *labeling* ini dilakukan dengan perangkat lunak *LabelImg* oleh Tzutalin. Setelah dilakukan proses *labeling*, hasil dari pelabelan disimpan ke dalam format *xml*. Dataset ini biasa juga disebut bertipe Pascal VOC. Gambar 3.6 menunjukkan antar muka dari *platform* *labelImg* oleh Tzutalin, juga sebagai contoh gambaran proses *labelling dataset* untuk penelitian ini.



Gambar 3. 6 Antarmuka platform LabelImg

### 3.5.1.3 Perancangan Arsitektur *Single Shot Multibox Detection*

Deteksi objek yang dilakukan pada penelitian kali ini menggunakan arsitektur SSD oleh *Liu et al.*, pada tahun 2016. Dimana arsitektur ini diterapkan menggunakan bahasa pemrograman Python 3.6 yang terinstall di SBC Jetson Nano beserta *platform library* Pytorch untuk deteksi objek dan *library* OpenCV untuk memvisualisasikan sehingga dapat diamati dan dianalisa oleh peneliti. Parameter *training* yang dilakukan terdapat pada Tabel 3.2

Tabel 3. 2 Parameter *Training* SSD

<b>Parameter</b>	<b>Keterangan</b>
<i>BackBone</i> SSD	MobileNetV2
<i>Optimizer</i>	<i>Stochastic Gradient Descent</i>
<i>Learning Rate</i>	0,01
<i>Epoch</i>	100
<i>Batch Size</i>	4
<i>Workers</i>	2

*Backbone* merupakan jaringan dasar dari arsitektur SSD seperti yang disebutkan pada Bab 2. Arsitektur SSD yang dikembangkan oleh (*Liu et al.*, 2016) menggunakan VGG-16 sebagai jaringan dasar. Pada penelitian kali ini menggunakan jaringan dasar berupa MobilNetV2 yang dikembangkan oleh (*Sandler et al.*, 2018), dimana jaringan ini lebih ringan, lebih cepat serta

menurunkan nilai *loss* pada proses *training* jika dibandingkan dengan jaringan VGG-16.

*Optimizer* yang digunakan adalah *Stochastic Gradient Descent* (SGD), dimana *optimizer* ini menurut (Gupta et al., 2022) mengatakan bahwa SGD sebagai *optimizer* jika diterapkan pada *image classification* menunjukkan performa yang lebih baik daripada *optimizer* ADAM yang sering digunakan proses *training neural network*. Dataset diberikan sebanyak  $n$ , maka nilai *objective function* pada SGD adalah sebagai berikut (Czum, 2020):

$$f(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) \quad (3.14)$$

Dengan  $f_i(x)$  adalah *loss function* ke- $i$  dimana  $x$  adalah vektor parameter atau bobot (*weight*). Sehingga fungsi gradient dari SGD dituliskan pada Persamaan (3.4)(Czum, 2020).

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x) \quad (3.15)$$

Setelah diketahui nilai gradient dari  $\nabla f(x)$  maka untuk memperbaharui bobot digunakan persamaan sebagai berikut:

$$x_i = x_i - \eta \nabla f_i \quad (3.16)$$

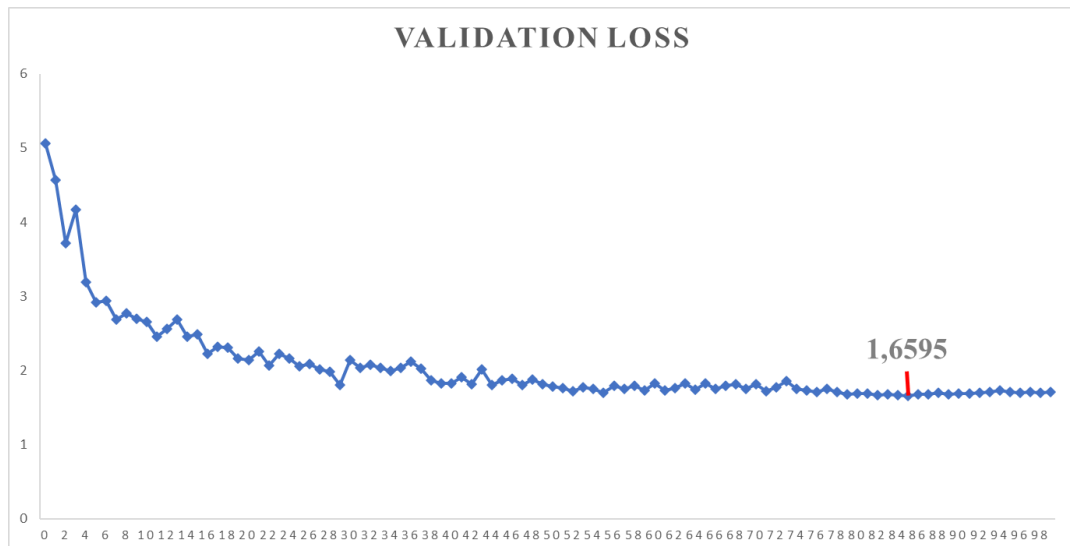
Dengan  $\eta$  merupakan *learning rate*.

Inisiasi *learning rate* penelitian kali ini tertera pada Tabel 3.2, kemudian nilai *learning rate* akan berubah secara dinamis hingga menemukan titik yang optimal dalam meminimalkan nilai Persamaan 3.14. *Batch size* berhubungan proses pengambilan gambar pada penyimpanan Jetson Nano, nilai 4 digunakan sebagai *default* program yang ditemukan, juga penelitian ini hanya menggunakan jumlah dataset yang sedikit sehingga nilai tersebut cukup, dikarenakan semakin besar *batch size* akan semakin meningkatkan akurasi, namun akan memberikan beban kerja yang lebih pada Jetson Nano. Parameter terakhir adalah *workers* merupakan parameter untuk membagi beban tiap iterasi yang berdampak pada waktu komputasi. Sehingga dengan nilai *batch size* adalah 4 dan data *training* terdapat 80 gambar mengakibatkan setiap *epoch* terdiri dari 2 iterasi yang setiap iterasi terdiri dari 10 gambar.

### 3.5.1.4 Training Data Set

*Training* dilakukan menggunakan data set 80 gambar dengan pembagian 4 kelas yaitu mobil, lajurkiri, lajurkanan dan “lainnya”. Proses *training* menggunakan parameter pada Tabel 3.2. Validasi model ditunjukkan dengan performa penurunan yang signifikan dan nilai *loss* yang kecil ditunjukkan pada Gambar 3.6, dimana pada validasi model ini pula penentuan besarnya *epoch* terjadi. *Epoch* ditentukan dengan melihat kestabilan hasil *validation loss* pada proses *training*. Pada penelitian ini, kestabilan nilai *validation loss* terjadi mulai dari *epoch* ke-75 yang angkanya stabil berkisar di angka 1,65 hingga 1,74. Berdasarkan hal tersebut maka besarnya *epoch* yang dipilih pada penelitian kali ini adalah 100 *epoch*.

Pada Gambar 3.7 ditunjukkan besarnya nilai *validation loss* yang terpilih adalah 1,6595 dengan skala 100 yang terjadi pada *epoch* ke-85. Setiap *epoch* akan menghasilkan model arsitektur. Sehingga pada proses *training* ini digunakan model pada *epoch* ke-85 yaitu “mb1-ssd-Epoch-85-Loss-1.6594752073287964.pth”. Model ini kemudian akan digunakan sebagai model pada proses *evaluation* dipenelitian ini.



Gambar 3. 7 Validation Loss

### 3.5.1.5 Evaluation Data Set

*Evaluation* dilakukan menggunakan 20 gambar yang terdiri dari 4 kelas sama dengan proses *training*. *Evaluation* menggunakan model *training* yang

terpilih dengan nilai *validation loss* yang paling kecil. Hasil *evaluation* rata-rata kepresisian deteksi kelas pada proses *evaluation* tertera pada Tabel 3.3 berikut ini

Tabel 3. 3 Nilai Rata-rata Presisi Deteksi Objek

<i>Class</i>	<i>Average Precision</i>
mobil	0,94
Lajurkiri	1,00
Lajurkanan	0,82
Rata-rata	0,92

Berdasarkan Tabel 3.3 nilai rata-rata keseluruhan deteksi dari model yang terpilih diatas 90%, sehingga model tersebut bisa digunakan untuk mendeteksi objek.

Pada *platform* Pytorch model yang terpilih kemudian dikonversi menjadi file .onnx untuk diterapkan pada proses deteksi *real time*. Model dengan format “mb1-ssd-Epoch-85-Loss-1.6594752073287964.pth” kemudian dikonversi menjadi “ssd-mobilenet.onnx”, untuk kemudian model ini akan menjadi model yang digunakan sebagai deteksi objek *real time* pada *platform* Pytorch.

### 3.5.1.6 Deteksi Objek

Proses deteksi objek menggunakan model “ssd-mobilenet.onnx” diberikan masukan gambar 720 x 320 pixel. Deteksi objek kali ini juga mendeteksi koordinat dari kelas mobil terhadap *ego-vehicle*. Pada deteksi objek titik (0,0) koordinat kartesian terletak pada pojok kiri atas, sehingga titik tengah *ego-vehicle* terletak pada koordinat (320,360) pixel. Posisi koordinat dari kelas mobil diperoleh dari titik tengah *bounding box* yang dihasilkan oleh SSD. Sehingga pada deteksi objek terdapat beberapa informasi yang dihasilkan yaitu deteksi hambatan, posisi “mobil” terhadap *ego-vehicle* dan posisi *ego-vehicle* terhadap lajur.

### 3.5.2 Kalman Filter Obstacle Position

Hambatan pada penelitian ini digunakan hambatan statis, serta hanya terdapat satu jenis hambatan. Posisi hambatan akan ditinjau pada bidang kartesian

terhadap *ego-vehicle*. Dikarenakan hambatan statis sehingga diasumsikan model sistem linier sehingga menggunakan *basic Kalman filter* untuk meningkatkan akurasi serta presisi dari posisi hambatan. Berdasarkan asumsi tersebut maka *state*  $X_k = [s_k \ x_k \ y_k \ u_k \ v_k]^T$  dengan nilai  $s_k$  adalah besarnya *confidence score*,  $x_k$  dan  $y_k$  adalah koordinat kartesian hambatan yang ditinjau dari titik tengah *bounding box* SSD,  $u_k$  dan  $v_k$  kecepatan arah horizontal dan vertical dari objek (Yang et al., 2019). Persamaan proses didefinisikan sebagai berikut

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & t & 0 \\ 0 & 0 & 1 & 0 & t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.18)$$

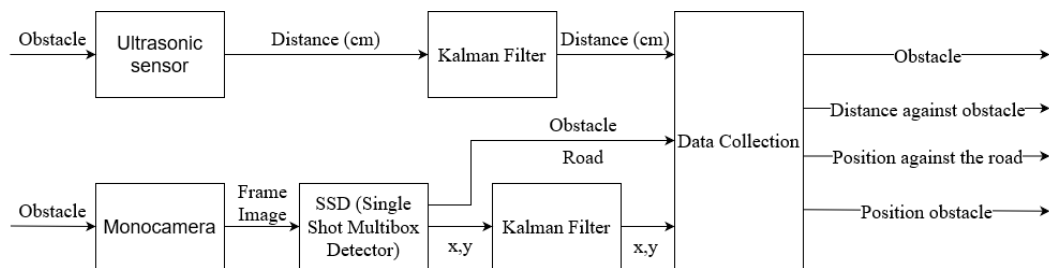
Sebagaimana alur yang didefinisikan pada Gambar 3.5, maka estimasi pada sistem objek deteksi dapat meningkatkan pengukuran hasil dari posisi hambatan pada bidang kartesian terhadap *ego-vehicle*

### 3.6 Perancangan *Sensor Fusion*

Bab 2 Subbab 2.4 menurut (Guan et al., 2019) *sensor fusion* dibagi menjadi beberapa kategori. Pada penelitian ini kategori *sensor fusion* berdasarkan sumber penggabungan data, yaitu *multimodal fusion sensor* dan berdasarkan konfigurasi sensor, *sensor fusion* saling melengkapi (*complementary*), yang mana dua independen sensor digunakan kemudian keluaran dari masing-masing sensor dikombinasikan untuk melengkapi satu sama lain.

Penggabungan sensor dilakukan dengan melakukan koleksi data dari masing-masing sensor yang telah melalui tahap pemrosesan. Sistem penggabungan sensor secara jelas dapat dilihat pada Gambar 3.7. Proses yang melalui filter Kalman yaitu data jarak pada sensor ultrasonik dan posisi *obstacle* terhadap sumbu (x, y) dari *ego-vehicle*. Keluaran dari sistem gabungan sensor ini terdiri dari 4 data. Data tentang keberadaan hambatan yang berada didepan *ego-vehicle*, data ini kemudian berhubungan dengan posisi hambatan tersebut terhadap bidang kartesian. Lalu, posisi *ego-vehicle* terhadap jalan, informasi ini ditujukan sebagai acuan arah proses manuver *ego-vehicle* pada sistem kontrol yang akan dikerjakan dikemudian hari.

Terakhir adalah informasi jarak antara *ego-vehicle* dengan hambatan data ini diperoleh dari sensor ultrasonik yang telah melalui tahap pemrosesan.



Gambar 3. 8 Diagram Blok Sistem Sensor Fusion

### 3.7 Pengujian Performa *Sensor Fusion*

Pengujian performa *sensor fusion* dilakukan dengan menganalisa data koleksi yang telah dikumpulkan. Analisa ini berupa adanya nilai selisih dari data sebenarnya terutama setelah melalui proses filter Kalman. Analisa juga menyangkut peningkatan nilai akurasi dan presisi yang dihasilkan setelah melalui pemrosesan di setiap sensor menggunakan filter Kalman. Peningkatan akurasi dan presisi akan meningkatkan proses keselamatan yang terjadi pada *autonomous vehicle*.

### 3.8 Penyusunan Laporan

Penyusunan laporan merupakan tahapan akhir dari penelitian ini. Laporan akhir disusun sesuai dengan format dari pasca sarjana Institut Teknologi Sepuluh Nopember. Laporan akhir merupakan bentuk pertanggungjawaban dari berbagai kegiatan yang telah dilakukan selama proses penelitian ini berlangsung.

*Halaman ini sengaja dikosongkan*

## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai data serta pembahasan yang telah diperoleh selama penelitian ini berlangsung.

#### 4.1 Pengukuran jarak menggunakan sensor HC-SR04

Pengukuran jarak dilakukan pada beberapa skenario, namun sebelumnya sebagai validasi pengukuran diperlukan peninjauan dari tingkat akurasi, tingkat presisi dan pengukuran secara dinamis yang dituliskan sebagai berikut.

##### 4.1.1 Peninjauan jarak sensor HC-SR04 dari tingkat akurasi

Pengukuran jarak sebagai validasi dengan pengambilan data sebanyak 100 kali disetiap pengukuran jarak, dengan rincian data mentah pengukuran jarak menggunakan sensor HC-SR04 pada Tabel 4.1. Tabel 4.1 merupakan hasil rata-rata pengambilan pengukuran.

Tabel 4. 1 Data mentah pengukuran jarak sensor ultrasonik

Kategori	Nilai									
Nilai Sebenarnya (cm)	5	10	15	20	25	30	35	40	45	50
Nilai Terukur (cm)	4,5	10,11	15,01	19,82	24,46	29,33	34,15	38,77	43,60	48,49
Selisih (cm)	0,5	0,11	0,01	0,18	0,54	0,67	0,85	1,23	1,4	1,5
Error (%)	10,06	1,154	0,9	0,88	2,15	2,24	2,43	3,07	3,12	3

Berdasarkan table tersebut dapat dilihat selisih pengukuran melebihi 1 cm pada jarak diatas 40 cm. selain itu, besarnya kesalahan ketika mengukur pada jarak dekat terutama pada jarak 5 cm akan berakibat fatal jika terjadi kecelakaan. Berdasarkan hal ini maka untuk meningkatkan nilai akurasi pada pengukuran jarak, dikalibrasi menggunakan Persamaan 3.2.

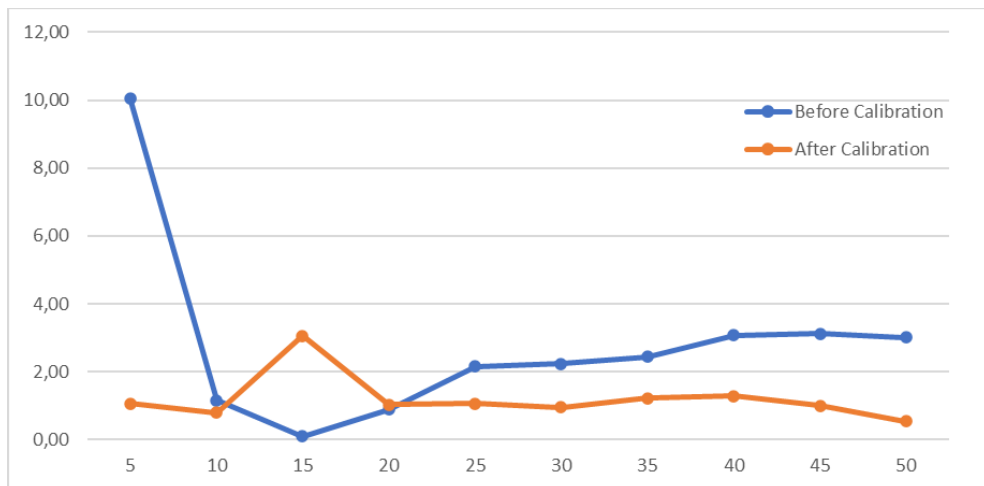
Proses pengambilan data setelah melalui proses kalibrasi dengan menggunakan persamaan 3.2 sama dengan proses validasi pengukuran pada data mentah sebelumnya. Prosesnya yaitu melakukan pengukuran disetiap jarak 5 hingga 50 cm dengan selisih pengukuran 5 cm, kemudian setiap pengukuran

dilakukan pengambilan data 100 kali, sehingga diperoleh 1000 data pengukuran jarak dari 10 skenario yang telah ditetapkan. Setelah itu, dirata-rata hasil pengukuran tiap jarak tersebut yang kemudian dibandingkan dengan nilai sebenarnya. Selisih kesalahan kemudian disajikan dalam bentuk ukuran cm dan besarnya presentasi. Sajian data dari proses kalibrasi dapat dilihat pada Tabel 4.2 berikut ini.

Tabel 4. 2 Hasil pengukuran setelah kalibrasi

Nilai Sebenarnya (cm)	Sebelum kalibrasi			Sesudah kalibrasi		
	Terukur (cm)	Selisih (cm)	Kesalahan (%)	Terukur (cm)	Selisih (%)	Kesalahan (%)
5	4,50	0,50	10,06	5,05	0,05	1,05
10	10,12	0,12	1,15	10,08	0,08	0,79
15	15,01	0,01	0,10	15,46	0,46	3,06
20	19,82	0,18	0,88	20,21	0,21	1,04
25	24,46	0,54	2,15	25,26	0,26	1,06
30	29,33	0,67	2,24	30,29	0,29	0,96
35	34,15	0,85	2,43	35,42	0,42	1,21
40	38,77	1,23	3,07	40,52	0,52	1,29
45	43,60	1,40	3,12	45,45	0,45	0,99
50	48,50	1,50	3,01	50,27	0,27	0,54

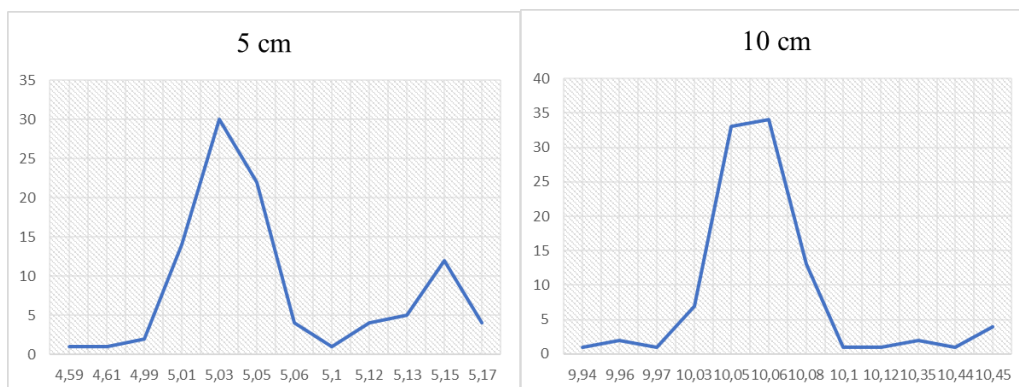
Tabel 4.2 menunjukkan perbandingan nilai antara sebelum dan sesudah dikalibrasi. Ditinjau dari selisih dalam besaran “cm” menunjukkan bahwa setelah dikalibrasi sensor pengukuran jarak tidak melebihi dari 1 cm. Besarnya batas selisih tidak melebihi 1 cm dikarenakan skala yang digunakan pada penelitian ini sebesar 1:20. Sehingga ketika melebihi 1 cm maka pada jarak sebenarnya sebesar 20 cm, yang pada kecepatan tinggi perbedaan ini sangat fatal dan dapat mengakibatkan telatnya respon *autonomous vehicle* untuk mencegah terjadinya kecelakaan. Oleh karena itu, batas/*threshold* 1 cm digunakan sebagai acuan pada pengukuran jarak ini. Tinjauan berdasarkan besarnya persentasi *error* disetiap jarak mengalami penurunan bahkan signifikan untuk jarak 40 cm hingga 50 cm. Namun, pada jarak 15 cm terjadi peningkatan persentasi kesalahan, yang signifikan pula yaitu meningkat 2,4%. Tetapi, jika *error* ini ditinjau dari selisih dalam “cm” nilai yang ditunjukkan sebesar 0,46 cm yang mana ini dibawah 1 cm, dan masih dalam batas aman pengukuran jarak pada *autonomous vehicle*. Besarnya nilai *error* dari setiap pengukuran jarak dapat dilihat pada Gambar 4.1.

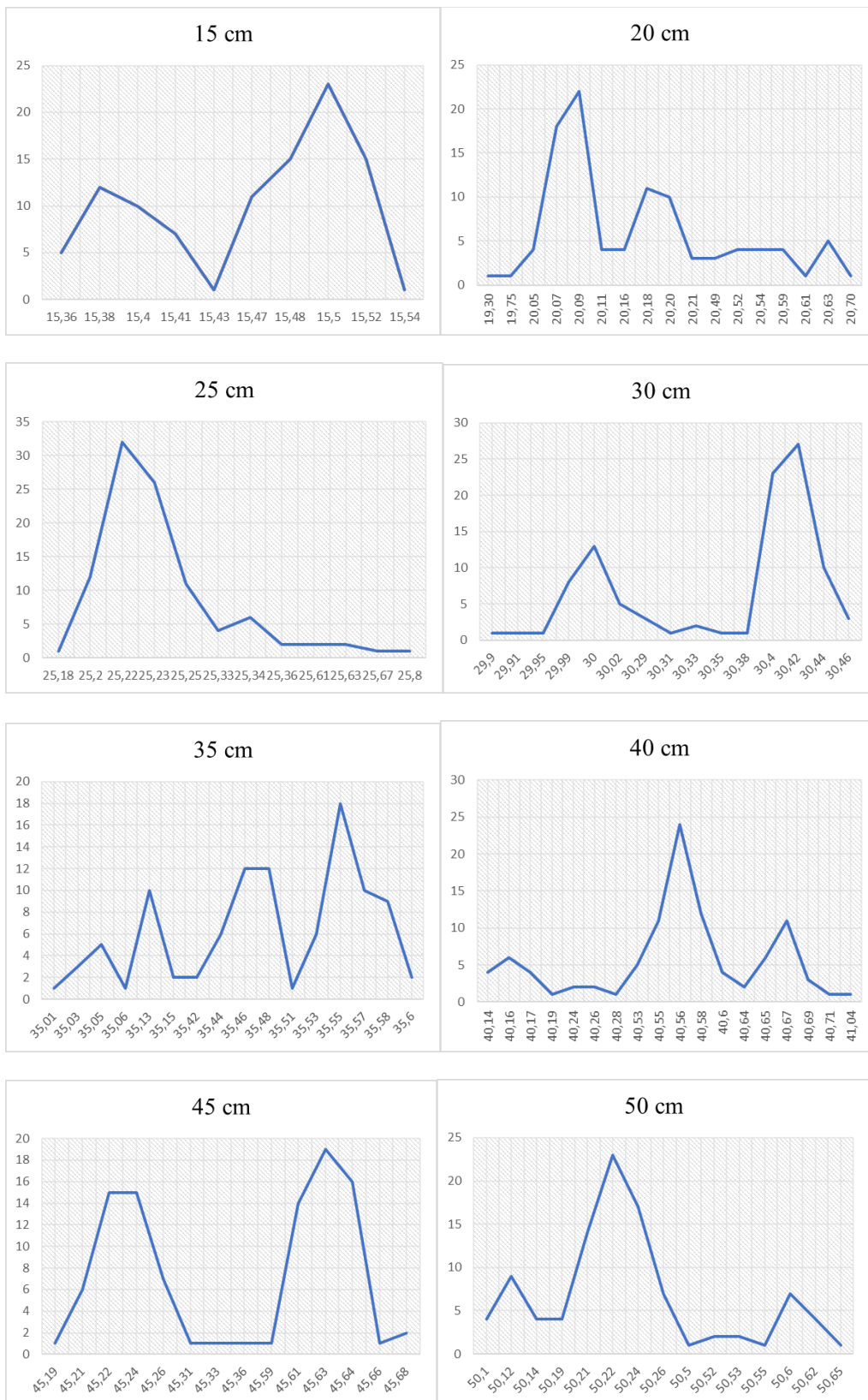


Gambar 4. 1 Grafik *error* setiap pengukuran

#### 4.1.2 Peninjauan jarak sensor HC-SR04 dari tingkat presisi

Setelah validasi ditinjau dari tingkat akurasi yang telah dibahas pada Subbab 4.1.1. Pada Subbab ini akan validasi dari sistem pengukuran ditinjau dari tingkat presisi dari pengukuran. Data yang digunakan merupakan data setelah dikalibrasi dengan setiap jarak pengukuran diambil 100 data seperti yang tertera pada Lampiran. Data kemudian akan dilihat berdasarkan frekuensi data yang dihasilkan yang kemudian akan diplot pada grafik. Gambar 4.2 menunjukkan bahwa kepresisian setiap pengukuran jarak, dimana hampir semua pengulangan nilai disetiap jarak pengukuran menunjukkan model gaussian. Untuk mereduksi hal ini maka digunakan *Kalman filter*. dengan model linier maka digunakan *Kalman filter basic*. Proses pereduksian menggunakan *Kalman filter* dengan proses seperti pada Gambar 3.5.





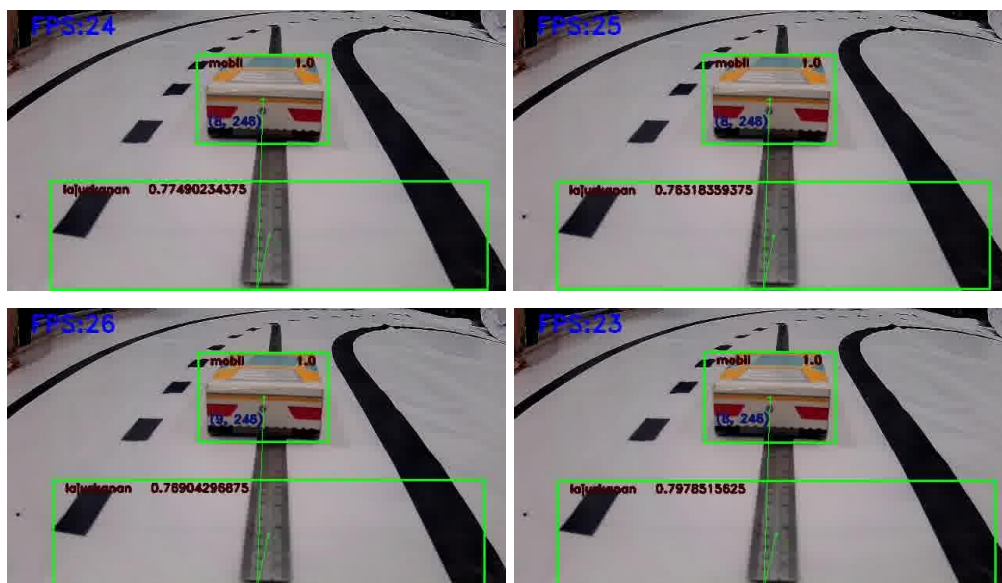
Gambar 4. 2 Grafik plot presisi pengukuran tiap jarak

## 4.2 Object Detection

Deteksi hambatan menggunakan arsitektur SSD dengan input kamera sebesar 640 x 320 pixel dibagi menjadi beberapa skenario, yang bertujuan untuk melihat performa dari deteksi hambatan. Skenario ini meliputi deteksi pada lajur kanan dengan kondisi mobil selajur. lalu lajur kanan dengan posisi hambatan mobil ada dilajur berbeda. hal ini dilakukan juga pada lajur kiri. Berikutnya juga deteksi hambatan mobil ketika mobil dalam kondisi yang acak. Berikut merupakan penjelasan dari proses deteksi objek

### 4.2.1 Lajur kanan

Deteksi hambatan ketika *ego vehicle* berada di lajur kanan. Pada proses ini akan diambil 20 *frame* untuk kemudian dianalisa dari nilai *confident score*, yang merupakan tingkat keyakinan suatu objek dikenali oleh. Berikutnya, posisi hambatan terhadap sumbu x dan sumbu y lalu *frame per second* (FPS) rata yang dihasilkan. Gambar 4.3 merupakan hasil sampel ketika hambatan berada di lajur kanan dengan hambatan berada di lajur yang sama.

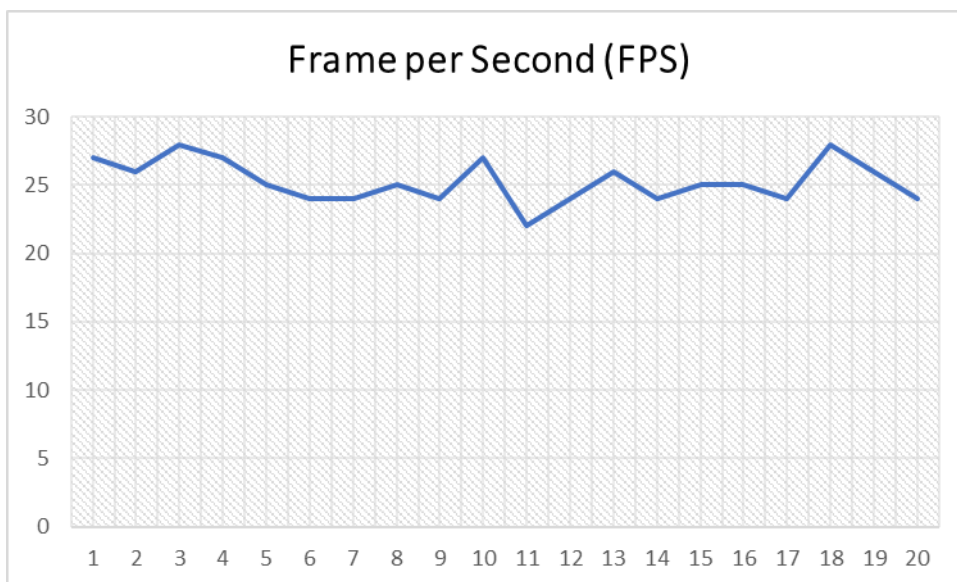


Gambar 4. 3 Sampel *ego-vehicle* dan hambatan di lajur kanan

Gambar 4.3 menunjukkan bahwa dalam skenario ini performa deteksi hambatan sangat bagus. Pada skenario ini juga tidak ada *false positive* maupun *false negative* yang muncul. Pada skenario ini diperoleh data rata-rata *confident score*, lajur kanan sebesar 77,7% dan untuk mobil memiliki nilai *confident score* sebesar 100%.

Tabel 4. 3 Nilai parameter *ego-vehicle* dan hambatan di lajur kanan

Frame ke-	Confident score		Sumbu x	Sumbu y
	lajur kanan (%)	Mobil (%)		
1	78,13	100	9	246
2	76,56	100	9	246
3	78,61	100	9	246
4	78,91	100	10	246
5	80,62	100	8	246
6	79,49	100	8	246
7	79,59	100	9	245
8	71,81	100	9	245
9	76,61	100	9	246
10	75,24	100	9	246
11	77,78	100	8	246
12	77,49	100	8	246
13	76,61	100	9	245
14	76,56	100	9	246
15	76,31	100	8	246
16	78,61	100	9	245
17	79,35	100	10	248
18	80,08	100	9	245
19	77,78	100	8	246
20	77,53	100	8	246
Rata-Rata	77,68	100	8,75	245,85

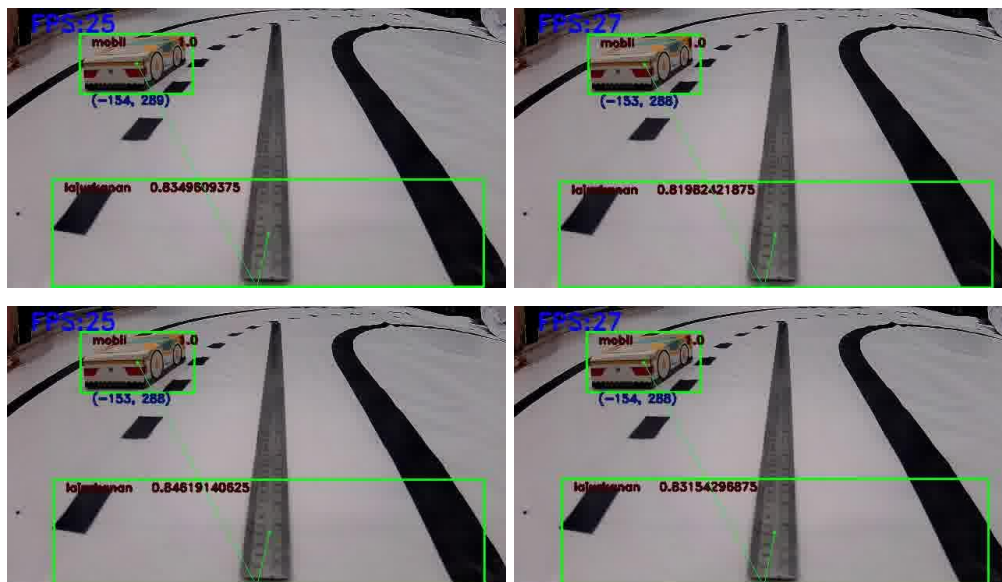


Gambar 4. 4 FPS *ego-vehicle* dan hambatan di lajur kanan

Nilai tersebut menunjukkan nilai kepastian baik terutama untuk kelas “mobil”. Dan untuk kelas “lajur kanan”, nilai tersebut dianggap cukup pada penelitian kali ini. Hal ini, dikarenakan juga lajur kanan hanya diperuntukkan untuk mobil mengetahui posisi terhadap lajur, dan hasil stabil tanpa adanya *false positive* maupun *negative*, mengindikasikan kestabilan pengenalan objek.

Berikutnya untuk posisi dari hambatan pada memiliki nilai rata-rata bidang kartesian yaitu (8,75; 245,85). Nilai positif pada sumbu x menunjukkan bahwa hambatan berada disisi kanan titik tengah dari *ego-vehicle*. Data berupa besaran nilai parameter pada skenario ini dapat dilihat pada Tabel 4.3. lalu dari rata-rata FPS yang diperoleh dari skenario ini ditunjukkan pada Gambar 4.4. Gambar tersebut menunjukkan bahwa nilai frame stabil dibesaran antara 22 hingga 28 FPS. Dan dari 20 frame yang diambil, diperoleh rata-rata FPS sebesar 25,25 FPS. Nilai ini menunjukkan angka yang bagus dalam deteksi objek/hambatan secara *real time*.

Skenario kedua ketika *ego-vehicle* berada dalam lajur kanan yaitu mengenali hambatan jika hambatan berada di lajur lainnya. Gambar 4.5 menunjukkan beberapa sampel hasil deteksi objek baik lajur maupun hambatan.



Gambar 4. 5 Sampel *ego-vehicle* di lajur kanan hambatan di lajur kiri

Pada Gambar 4.5 menunjukkan hasil yang sama dengan skenario sebelumnya dimana tidak muncul adanya *false positive* maupun *false negative* pada deteksi objek. Hal ini menandakan bahwa tidak adanya kesalahan dalam deteksi objek dan

kestabilan dalam deteksi objek memiliki performa yang bagus. Selanjutnya pada Tabel 4.4 menunjukkan nilai parameter pada skenario ini. Dimana hasil rata-rata *confident score* kelas “lajur kanan” mengalami peningkatan menjadi 82,26%, sedangkan pada kelas “mobil” nilainya tetap sama sebesar 100%. Kelas “mobil” memiliki nilai *confident score* tinggi padahal kondisi mobil berbeda dengan skenario sebelumnya dikarenakan banyaknya jenis kondisi mobil yang dijadikan *dataset* pada proses *training* sebelumnya.

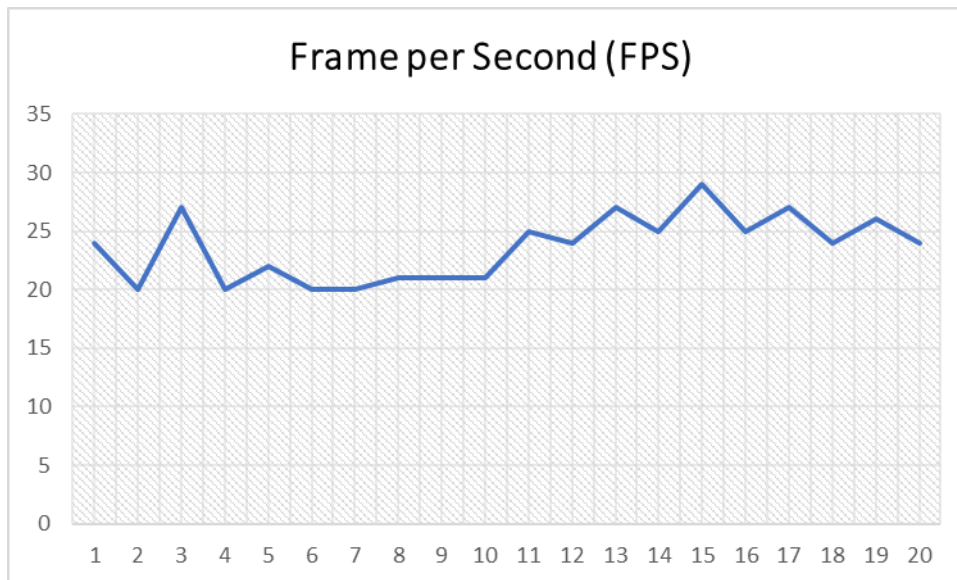
Tabel 4. 4 Nilai parameter *ego-vehicle* di lajur kanan dan hambatan di lajur kiri

Frame ke-	Confident score		Sumbu x	Sumbu y
	lajur kanan (%)	Mobil (%)		
1	82,86	100	-154	288
2	80,27	100	-153	288
3	83,40	100	-153	288
4	83,01	100	-153	288
5	82,37	100	-153	288
6	84,67	100	-153	288
7	81,83	100	-153	288
8	84,52	100	-153	288
9	79,44	100	-153	288
10	81,25	100	-153	288
11	81,64	100	-154	288
12	81,49	100	-153	288
13	83,30	100	-154	288
14	83,79	100	-153	288
15	84,18	100	-154	288
16	80,96	100	-154	288
17	80,81	100	-153	288
18	80,22	100	-153	288
19	81,83	100	-153	288
20	83,45	100	-152	288
Rata-Rata	82,26	100	-153,2	288

Table 4.4 juga menunjukkan perbedaan koordinat yang signifikan terutama pada sumbu x. Nilai negative pada sumbu x mengisyaratkan bahwa hambatan yang dideteksi berada pada sisi sebelah kiri dari *ego-vehicle*.

Parameter FPS ditunjukkan oleh Gambar 4.6, yang mana pada gambar tersebut nilai FPS dalam 20 *frame* sampel stabil disekitar angka 20-29 FPS. Lalu nilai rata-rata FPS yang diperoleh sebesar 23,6 FPS. FPS dengan rata-rata diatas 20

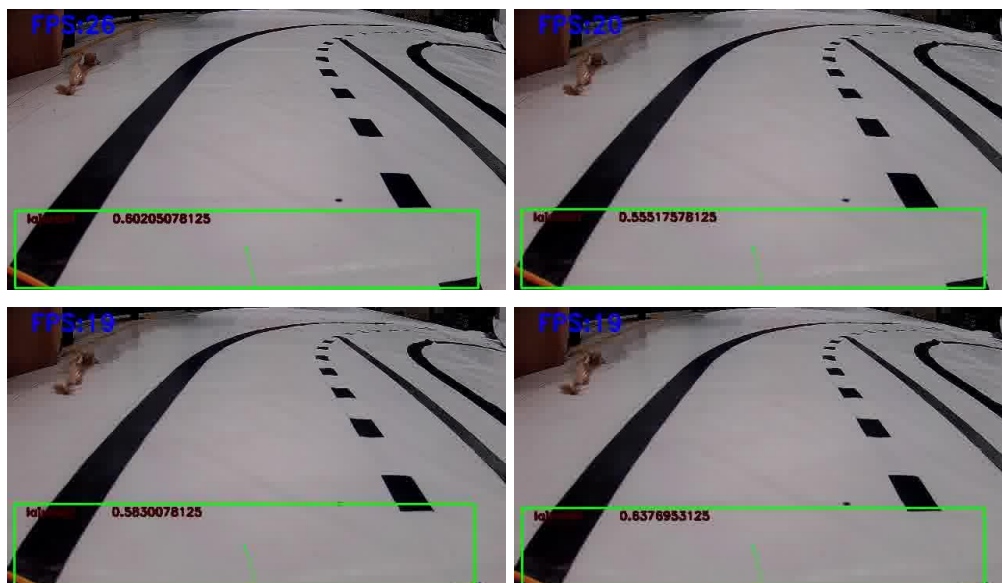
masih dalam performa yang baik untuk deteksi objek secara *real time*. Walaupun mengalami penurunan secara rata-rata FPS namun kestabilan algoritma SSD dalam skenario ini masih teruji secara baik.



Gambar 4. 6 FPS *ego-vehicle* di lajur kanan dan hambatan di lajur kiri

#### 4.2.2 Lajur Kiri

Pada penelitian kali ini skenario ketika mobil berada di lajur kiri juga diterapkan. Gambar 4.7 menunjukkan sampel skenario ini.



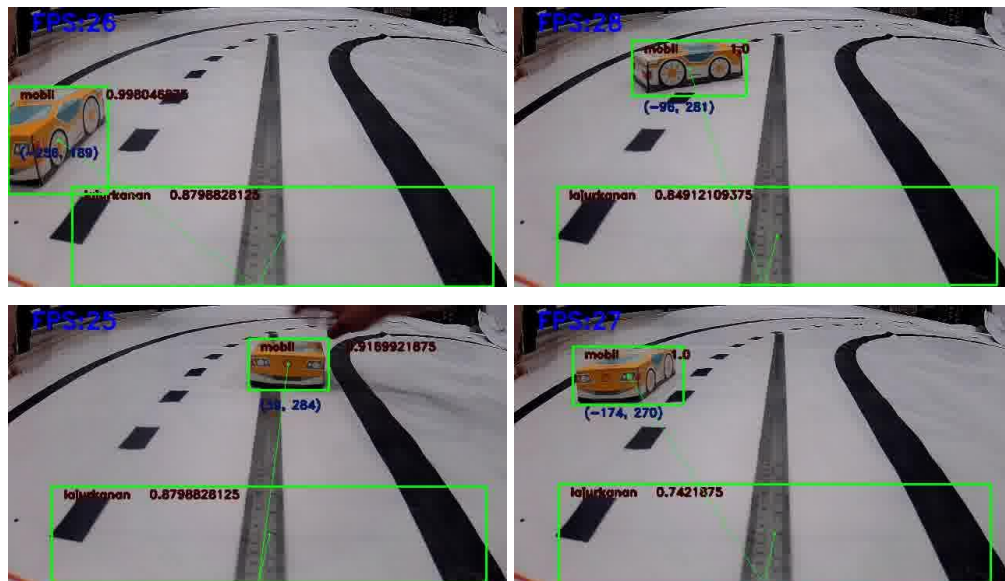
Gambar 4. 7 Skenario *ego-vehicle* di lajur kiri

Pada gambar tersebut tidak terdapat hambatan dalam *frame* dikarenakan ketika terdapat hambatan dalam *frame* maka lajur kiri akan berubah kelas menjadi lajur kanan. Hal ini merupakan *false positive* dalam penelitian ini. Selain itu beberapa kali deteksi objek kelas “lajur kiri” mengalami kesalahan deteksi dengan menganggap sebagai “lajur kanan”. Dan untuk nilai *confident score* tidak terlalu tinggi hanya berkisar sebesar 50-60%, serta FPS yang turun tidak mencapai 20 FPS.

Oleh karena itu pada skenario ini bisa dikatakan deteksi mengalami performa yang buruk. Sehingga memerlukan data *training* dengan variasi yang lebih banyak dan beberapa kondisi yang lebih beragam. Agar deteksi dalam skenario di lajur kiri mengalami performa yang baik.

#### 4.2.3 Hambatan Acak

Pada penelitian ini juga mengambil beberapa sampel deteksi hambatan yang diposisikan secara acak. Gambar 4.7 menunjukkan beberapa sampel kondisi mobil yang diterapkan secara acak pada lintasan.



Gambar 4. 8 Skenario mobil acak

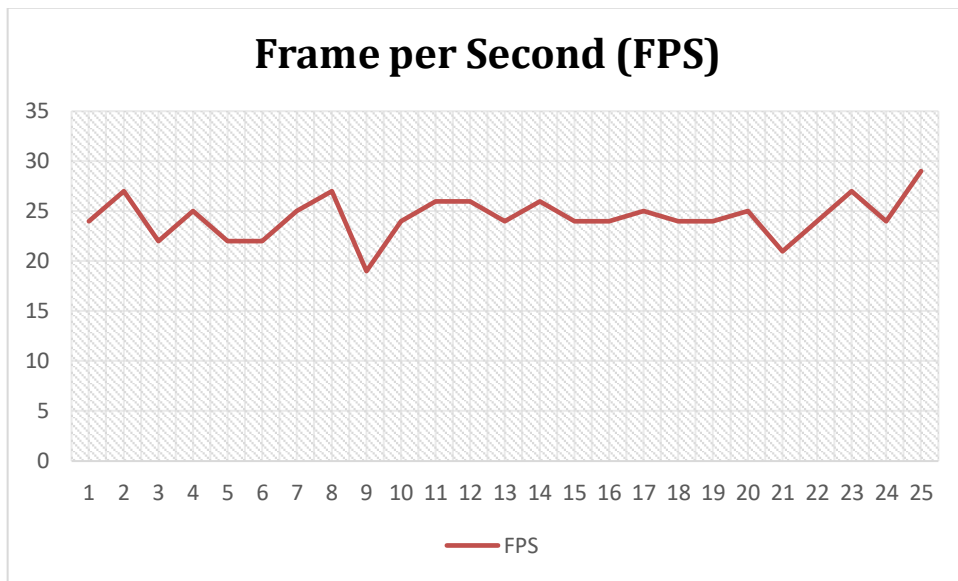
Pada Gambar tersebut menunjukkan bahwa, beberapa kondisi mobil seperti hanya terlihat Sebagian didalam *frame*. Lalu, menghadap berlawanan dengan *ego-vehicle*. Bergerak seakan-akan pindah lajur. Mampu dikenali dengan baik oleh arsitektur SSD, tanpa memunculkan *false negative* maupun *false positive*. Besarnya nilai *confident score* dari kelas “mobil” juga dapat diperhatikan pada Tabel 4.5 berikut.

Dimana nilai rata-rata *confident score* berada diangka 998,6 %. Terjadi penurunan nilai *confident score* ketika tangan peneliti berada didalam *frame* dan menyentuh hambatan ketika proses memindahkan hambatan ke skenario lainnya.

Tabel 4. 5 Besarnya parameter skenario hambatan acak

<i>Frame ke-</i>	FPS	<i>Confident score (%)</i>
1	24	100
2	27	100
3	22	100
4	25	100
5	22	100
6	22	94,43
7	25	91,89
8	27	85,88
9	19	93,70
10	24	100
11	26	100
12	26	100
13	24	100
14	26	100
15	24	100
16	24	99,9
17	25	99,9
18	24	99,9
19	24	99,8
20	25	99,83
21	21	99,9
22	24	100
23	27	100
24	24	100
25	29	100
Rata-rata	24,4	98,60

Besarnya FPS juga stabil diantara 19 hingga 29 FPS. Nilai ini diambil dari 25 *frame* sampel yang setiap skenario ada dalam sampel tersebut. Gambar 4.8 menunjukkan kestabilan FPS dengan nilai rata-rata 24,4 FPS. Oleh karena itu, berdasarkan hal ini, arsitektur SSD untuk deteksi hambatan kelas “mobil” terbukti menghasilkan performa yang baik, bahkan untuk kondisi-kondisi hambatan tidak terlihat sepenuhnya didalam *frame*.



Gambar 4. 9 FPS skenario pelatakan hambatan secara acak

## BAB 5

### KESIMPULAN DAN SARAN

Adapun kesimpulan dan saran yang dihasilkan oleh penelitian ini dituliskan sebagaimana berikut:

#### 5.1 Kesimpulan

Adapun kesimpulan yang diambil dari penelitian *sensor fusion* menggunakan mono kamera dan sensor jarak ultrasonik HC-SR04 adalah sebagai berikut:

- Sistem persepsi yang dirancang menunjukkan performa yang baik, dilihat dari sensor HC-SR04 yang menunjukkan nilai *error* maksimal pengukuran di jarak 15 cm yaitu 3,06% namun ketika ditinjau dalam ukuran “cm” nilai *error* 0,46 cm yang mana masih dibawah 1 cm sebagai batas maksimal toleransi kesalahan pengukuran jarak pada penelitian ini. Tingkat presisi menunjukkan berdistribusi normal, nilainya cukup baik namun untuk meningkatkannya dapat menggunakan *Kalman filter* yang dijadikan sebagai saran penelitian berikutnya
- Pada deteksi objek dan posisi hambatan menggunakan monokamera berbasis arsitektur SSD, menunjukkan performa yang baik ketika *ego-vehicle* berada di lajur kanan, yang menunjukkan tidak adanya *false positive* dan *false negative* serta *confident score* kelas “lajur kanan” dengan rata-rata 77,7 % dan 82,26%. Sedangkan deteksi hambatan dalam skenario lajur kanan dan posisi yang acak menunjukkan hasil yang maksimal ditandai dengan *confident score* 100% dan 98,6% ketika hambatan diposisikan dengan berbagai posisi menghadap kamera. Nilai FPS juga menunjukkan rata-rata diatas 20 untuk semua skenario sehingga arsitektur SSD mampu diterapkan secara *real time*. Namun pada skenario yang terjadi pada lajur kiri muncul *false positive* ketika ada ataupun tidak ada hambatan dalam *frame*. Hal ini

disebabkan karena kurang beragamnya data *training* pada proses training arsitektur SSD dilangkah awal.

## 5.2 Saran

Adapun saran yang diusulkan setelah penelitian kali ini, yang bertujuan untuk membangun penelitian yang lebih baik kedepannya adalah sebagai berikut:

- Deteksi objek secara *realtime* untuk sistem persepsi *autonomous vehicle*
- Meningkatkan nilai akurasi serta presisi dari sistem pengukuran menggunakan *Kalman filter*
- Data *training* dibuat lebih banyak lebih beragam serta dapat mencakup sebanyak mungkin kondisi yang akan ditempuh oleh *autonomous vehicle*

## DAFTAR PUSTAKA

- Al Tahtawi, A. R. (2018). Kalman Filter Algorithm Design for HC-SR04 Ultrasonic Sensor Data Acquisition System. *IJITEE (International Journal of Information Technology and Electrical Engineering)*, 2(1), 2–6. <https://doi.org/10.22146/ijitee.36646>
- Antsaklis, P. J. (University of N. D. ). (1991). An Introduction to Autonomous Control Systems.pdf. In *IEEE International Symposium on Intelligent Control* (p. 13).
- Appel, R., Belongie, S., Perona, P., & Doll, P. (2014). Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8), 1–14.
- Bevington, P. R., Robinson, D. K., Blair, J. M., Mallinckrodt, A. J., & McKay, S. (1993). Data Reduction and Error Analysis for the Physical Sciences. *Computers in Physics*, 7(4), 415. <https://doi.org/10.1063/1.4823194>
- Czum, J. M. (2020). Dive Into Deep Learning. *Journal of the American College of Radiology*, 17(5), 637–638. <https://doi.org/10.1016/j.jacr.2020.02.005>
- Du, X., Ang, M. H., & Rus, D. (2017). Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework. *IEEE International Conference on Intelligent Robots and Systems, 2017-Septe*, 749–754. <https://doi.org/10.1109/IROS.2017.8202234>
- Duan, X., Jiang, H., Tian, D., Zou, T., Zhou, J., & Cao, Y. (2021). V2I based environment perception for autonomous vehicles at intersections. *China Communications*, 18(7), 1–12. <https://doi.org/10.23919/JCC.2021.07.001>
- Fayyad, J., Jaradat, M. A., Gruyer, D., & Najjaran, H. (2020). Deep learning sensor fusion for autonomous vehicle perception and localization: A review. *Sensors (Switzerland)*, 20(15), 1–34. <https://doi.org/10.3390/s20154220>
- Feng, D., Haase-Schutz, C., Rosenbaum, L., Hertlein, H., Glaser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2021). Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341–1360. <https://doi.org/10.1109/TITS.2020.2972974>
- Gruyer, D., Magnier, V., Hamdi, K., Claussmann, L., Orfila, O., & Rakotonirainy, A. (2017). Perception, information processing and modeling: Critical stages for autonomous driving applications. *Annual Reviews in Control*, 44, 323–341. <https://doi.org/10.1016/j.arcontrol.2017.09.012>
- Guan, D., Cao, Y., Yang, J., Cao, Y., & Yang, M. Y. (2019). Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Information Fusion*, 50(November 2018), 148–157.

<https://doi.org/10.1016/j.inffus.2018.11.017>

- Guo, C., Kidono, K., Terashima, R., & Kojima, Y. (2018). Humanlike Behavior Generation in Urban Environment Based on Learning-Based Potentials with a Low-Cost Lane Graph. *IEEE Transactions on Intelligent Vehicles*, 3(1), 46–60. <https://doi.org/10.1109/TIV.2017.2788194>
- Gupta, A., Ramanath, R., Shi, J., & Keerthi, S. S. (2022). *Adam vs. SGD: Closing the generalization gap on image classification*. 1–7.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1), 35–45. <https://doi.org/10.1115/1.3662552>
- Karnouskos, S. (2020). Self-Driving Car Acceptance and the Role of Ethics. *IEEE Transactions on Engineering Management*, 67(2), 252–265. <https://doi.org/10.1109/TEM.2018.2877307>
- Kim, S., Petrunin, I., & Shin, H. S. (2022). A Review of Kalman Filter with Artificial Intelligence Techniques. *Integrated Communications, Navigation and Surveillance Conference, ICNS, 2022-April*, 1–12. <https://doi.org/10.1109/ICNS54818.2022.9771520>
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS, 21–37. [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- Lundquist, C. (2011). *Sensor Fusion for Automotive Applications* (Issue 1409).
- Malviya, A., & Bhirud, S. G. (2009). Wavelet based multi-focus image fusion. *Proceedings of International Conference on Methods and Models in Computer Science, ICM2CS09*. <https://doi.org/10.1109/icm2cs.2009.5397990>
- Martínez-díaz, M., Soriguera, F., & Pérez, I. (2018). *ReView by River Valley This Technologies IET Intelligent Transport Systems Autonomous driving : A bird's eye view IET Review Copy Only ReView by River Valley This Technologies IET Intelligent Transport Systems IET Review Copy Only*. 2–23.
- Masita, K. L., Hasan, A. N., & Shongwe, T. (2020). Deep learning in object detection: A review. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, IcABCD 2020 - Proceedings*. <https://doi.org/10.1109/icABCD49160.2020.9183866>
- Molugaram, K., & Rao, G. S. (2017). *Chapter 5 - Curve Fitting* (K. Molugaram & G. S. B. T.-S. T. for T. E. Rao (eds.); pp. 281–292). Butterworth-Heinemann.

<https://doi.org/https://doi.org/10.1016/B978-0-12-811555-8.00005-2>

- Morgan, E. J. (2014). HC SR04 Ultrasonic Ranging Sensor Module. *Evaluation Tecnical of Sensor*, Nov. 16 2014.  
[https://www.pcbway.com/blog/News/New\\_product\\_in\\_gift\\_shop\\_\\_HC\\_SR04\\_Ultrasonic\\_Ranging\\_Sensor\\_Module.html](https://www.pcbway.com/blog/News/New_product_in_gift_shop__HC_SR04_Ultrasonic_Ranging_Sensor_Module.html)
- Rosique, F., Navarro, P. J., Fernández, C., & Padilla, A. (2019). A systematic review of perception system and simulators for autonomous vehicles research. *Sensors (Switzerland)*, 19(3). <https://doi.org/10.3390/s19030648>
- RTI. (2010). *HC-SR04 User 's Manual*. 408, 1–38.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Song, E., Xu, J., & Zhu, Y. (2014). Optimal distributed kalman filtering fusion with singular covariances of filtering errors and measurement noises. *IEEE Transactions on Automatic Control*, 59(5), 1271–1282.  
<https://doi.org/10.1109/TAC.2014.2308451>
- Soviany, P., & Ionescu, R. T. (2018). Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *Proceedings - 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018*, 209–214.  
<https://doi.org/10.1109/SYNASC.2018.00041>
- Taylor, J. R. (1997). *An introduction to error analysis*. University Science Books.
- Urmson, C., & Whittaker, W. (2008). Self-driving cars and the Urban challenge. *IEEE Intelligent Systems*, 23(2), 66–68. <https://doi.org/10.1109/MIS.2008.34>
- Urrea, C., & Agramonte, R. (2021). Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation. *Journal of Sensors*, 2021(1). <https://doi.org/10.1155/2021/9674015>
- Wu, G., Zuo, Y., Wang, H., & Xu, X. (2019). Study on uncertainty reasoning of running reliability testing face to CNC machine. *The Journal of Engineering*, 2019(23), 8508–8512. <https://doi.org/10.1049/joe.2018.9043>
- Yang, F., Chen, H., Li, J., Li, F., Wang, L., & Yan, X. (2019). Single Shot Multibox Detector with Kalman Filter for Online Pedestrian Detection in Video. *IEEE Access*, 7, 15478–15488.  
<https://doi.org/10.1109/ACCESS.2019.2895376>

Yeong, D. J., Velasco-hernandez, G., Barry, J., & Walsh, J. (2021). Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors*, *21*(6), 1–37. <https://doi.org/10.3390/s21062140>

## LAMPIRAN

### A. Data pengukuran HC-SR04 setelah dikalibrasi

5 cm	10 cm	15 cm	20 cm	25 cm	30 cm	35 cm	40 cm	45 cm	50 cm
5,03	10,03	15,40	20,18	25,22	30,40	35,44	40,16	45,61	50,22
5,03	10,44	15,38	20,20	25,20	30,40	35,05	40,19	45,64	50,10
5,10	10,12	15,50	20,09	25,20	30,42	35,46	40,56	45,61	50,22
5,15	10,06	15,50	20,54	25,23	30,44	35,51	40,58	45,24	50,19
5,03	10,05	15,50	20,09	25,22	29,90	35,57	40,24	45,63	50,60
5,03	10,05	15,41	20,09	25,22	29,99	35,55	40,67	45,66	50,53
5,06	10,05	15,41	20,09	25,22	30,42	35,53	40,58	45,24	50,22
5,06	10,05	15,52	20,09	25,20	30,42	35,53	40,56	45,63	50,21
5,12	10,06	15,50	20,70	25,23	30,42	35,46	40,17	45,64	50,50
5,03	10,03	15,48	20,61	25,25	30,40	35,46	40,56	45,64	50,22
5,05	10,05	15,52	19,75	25,80	30,40	35,03	40,67	45,36	50,21
5,03	10,06	15,52	19,30	25,63	29,99	35,05	40,55	45,64	50,19
5,17	10,08	15,54	20,18	25,22	30,00	35,55	40,17	45,24	50,14
5,15	10,05	15,40	20,16	25,23	30,42	35,55	40,17	45,63	50,22
5,03	10,06	15,38	20,18	25,23	30,40	35,57	40,56	45,22	50,24
5,01	10,08	15,52	20,20	25,23	30,40	35,13	40,56	45,26	50,22
5,03	9,97	15,50	20,18	25,22	30,40	35,55	40,67	45,22	50,24
5,03	10,06	15,52	20,20	25,22	30,29	35,57	40,56	45,22	50,22
5,15	9,96	15,40	20,09	25,33	30,44	35,46	40,55	45,26	50,22
5,05	10,05	15,38	20,54	25,22	30,02	35,03	40,53	45,68	50,14
5,03	10,06	15,50	20,09	25,22	30,42	35,48	40,16	45,61	50,21
5,03	10,08	15,50	20,09	25,36	30,44	35,46	41,04	45,33	50,22
5,06	9,96	15,50	20,09	25,22	30,40	35,57	40,69	45,22	50,24
5,15	10,06	15,41	20,09	25,22	30,44	35,13	40,56	45,21	50,12
5,03	10,06	15,41	20,09	25,63	30,00	35,53	40,53	45,26	50,24
5,05	10,08	15,52	20,07	25,20	29,95	35,55	40,56	45,22	50,22
5,03	9,94	15,50	20,05	25,61	30,42	35,15	40,58	45,26	50,53
5,05	10,06	15,48	20,07	25,20	29,99	35,15	40,24	45,64	50,24
5,15	10,35	15,52	20,09	25,25	30,00	35,05	40,28	45,24	50,26
5,05	10,05	15,52	20,11	25,34	30,02	35,46	40,58	45,63	50,22
5,03	10,05	15,48	20,11	25,20	30,35	35,48	40,56	45,64	50,12
5,05	10,05	15,38	20,11	25,18	30,40	35,60	40,56	45,21	50,26
5,03	10,05	15,36	20,52	25,36	30,42	35,55	40,55	45,59	50,22
5,12	10,08	15,48	20,18	25,22	30,00	35,55	40,67	45,64	50,26
5,17	10,06	15,50	20,21	25,20	30,40	35,58	40,67	45,21	50,24
5,01	10,10	15,47	20,18	25,23	29,99	35,58	40,56	45,19	50,26
5,05	10,05	15,36	20,16	25,20	30,42	35,13	40,53	45,22	50,62
5,01	10,06	15,36	20,20	25,22	30,40	35,46	40,56	45,61	50,12
5,12	10,08	15,50	20,18	25,25	30,42	35,48	40,16	45,24	50,22
5,17	10,08	15,50	20,20	25,23	30,40	35,48	40,64	45,21	50,62
5,05	10,05	15,52	20,07	25,23	30,42	35,05	40,26	45,63	50,62
5,03	10,03	15,40	20,05	25,22	30,44	35,55	40,56	45,64	50,10
5,05	10,06	15,38	20,07	25,23	30,00	35,58	40,58	45,61	50,21

5,03	10,06	15,50	20,09	25,34	29,91	35,58	40,14	45,24	50,60
5,15	10,05	15,52	20,09	25,22	30,46	35,13	40,55	45,61	50,22
4,59	10,03	15,48	20,07	25,22	30,42	35,55	40,65	45,63	50,21
5,03	10,08	15,50	20,54	25,20	30,00	35,57	40,67	45,21	50,22
5,05	10,05	15,36	20,09	25,23	30,40	35,44	40,53	45,63	50,24
5,05	10,05	15,47	20,07	25,67	30,46	35,42	40,55	45,24	50,10
5,13	10,05	15,52	20,11	25,22	30,40	35,06	40,58	45,63	50,22
5,01	10,06	15,47	20,59	25,23	30,00	35,48	40,14	45,64	50,60
5,03	10,08	15,52	20,21	25,22	29,99	35,60	40,64	45,63	50,24
5,05	10,05	15,40	20,20	25,23	30,40	35,58	40,58	45,63	50,14
5,01	10,06	15,50	20,63	25,23	30,42	35,55	40,56	45,24	50,22
5,17	10,03	15,50	20,59	25,34	30,42	35,55	40,58	45,61	50,24
5,15	10,06	15,48	20,07	25,22	30,44	35,55	40,55	45,63	50,55
5,05	10,06	15,47	20,54	25,22	30,29	35,55	40,67	45,24	50,60
5,03	10,06	15,40	20,52	25,25	30,42	35,01	40,67	45,21	50,24
5,03	10,06	15,38	20,09	25,22	30,42	35,46	40,55	45,63	50,21
5,13	10,06	15,50	20,07	25,22	30,42	35,46	40,16	45,24	50,12
5,13	10,45	15,50	20,52	25,22	30,42	35,48	40,56	45,24	50,24
5,05	10,05	15,48	20,49	25,22	30,02	35,57	40,56	45,22	50,24
5,05	10,45	15,40	20,09	25,22	30,44	35,13	40,67	45,22	50,24
5,06	10,06	15,41	20,07	25,61	30,42	35,57	40,65	45,31	50,24
5,03	10,08	15,50	20,16	25,23	30,33	35,55	40,58	45,61	50,22
5,15	10,08	15,47	20,18	25,34	30,02	35,13	40,53	45,61	50,21
5,03	10,06	15,47	20,63	25,20	30,42	35,44	40,56	45,22	50,12
5,03	10,35	15,41	20,59	25,20	30,44	35,44	40,17	45,64	50,24
5,05	10,05	15,41	20,63	25,25	30,42	35,05	40,65	45,22	50,62
5,05	10,06	15,48	20,09	25,23	30,38	35,03	40,69	45,63	50,22
5,15	10,05	15,48	20,07	25,22	30,40	35,13	40,55	45,63	50,12
5,01	10,45	15,48	20,07	25,22	30,42	35,58	40,55	45,61	50,22
5,01	10,06	15,43	20,07	25,22	30,31	35,58	40,60	45,22	50,60
5,01	10,05	15,40	20,07	25,23	30,42	35,58	40,58	45,22	50,12
5,01	10,06	15,47	20,07	25,23	30,42	35,55	40,67	45,22	50,21
5,15	10,06	15,48	20,05	25,23	30,44	35,57	40,65	45,24	50,21
5,01	10,06	15,48	20,07	25,34	30,40	35,48	40,56	45,22	50,21
5,01	10,06	15,48	20,05	25,25	30,02	35,42	40,56	45,61	50,14
5,01	10,05	15,40	20,49	25,23	30,00	35,48	40,16	45,68	50,26
5,03	10,06	15,50	20,63	25,33	30,40	35,48	40,58	45,61	50,21
5,13	10,08	15,47	20,20	25,22	30,44	35,58	40,67	45,64	50,24
5,15	10,05	15,52	20,20	25,23	30,40	35,55	40,56	45,64	50,26
5,05	10,03	15,47	20,63	25,25	30,40	35,13	40,16	45,26	50,21
4,61	10,05	15,36	20,20	25,25	29,99	35,13	40,56	45,26	50,22
5,03	10,45	15,38	20,18	25,25	30,42	35,57	40,60	45,24	50,52
4,99	10,05	15,48	20,09	25,23	30,33	35,55	40,60	45,64	50,21
5,15	10,03	15,50	20,07	25,25	30,46	35,44	40,65	45,22	50,24
5,05	10,05	15,47	20,09	25,33	29,99	35,46	40,55	45,24	50,60
5,03	10,06	15,40	20,52	25,20	30,00	35,46	40,55	45,64	50,12
5,03	10,05	15,38	20,49	25,23	30,40	35,48	40,56	45,61	50,22
5,05	10,06	15,48	20,07	25,33	30,42	35,53	40,14	45,63	50,22
5,13	10,05	15,47	20,07	25,25	30,40	35,53	40,65	45,63	50,52

5,05	10,05	15,50	20,09	25,23	30,00	35,53	40,71	45,63	50,21
5,05	10,06	15,38	20,09	25,23	30,40	35,57	40,56	45,63	50,19
4,99	10,05	15,38	20,16	25,22	30,00	35,55	40,14	45,26	50,26
5,01	10,05	15,50	20,18	25,23	30,00	35,13	40,58	45,64	50,12
5,12	10,06	15,52	20,18	25,22	29,99	35,48	40,60	45,64	50,19
5,03	10,06	15,52	20,21	25,23	30,00	35,46	40,69	45,63	50,65
5,03	10,05	15,38	20,59	25,34	30,29	35,44	40,26	45,24	50,60
5,01	10,08	15,38	20,20	25,22	30,42	35,48	40,56	45,61	50,10

## B. Koding Sensor HC-SR04 (arduino UNO)

```
#define echoPin 12 //Echo Pin
#define trigPin 11 //Trigger Pin
#define LEDPin 13 //Led default dari Arduino uno

int maximumRange = 200; //kebutuhan akan maksimal range
int minimumRange = 00; //kebutuhan akan minimal range
long duration, distance; //waktu untuk kalkulasi jarak

void setup() {
  Serial.begin (9600); //inisialisasi komunikasi serial
  //deklarasi pin
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  /* Berikut siklus trigPin atau echo pin yang digunakan
  untuk menentukan jarak objek terdekat dengan memantulkan
  gelombang suara dari itu. */
  digitalWrite(trigPin, LOW);delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);

  //perhitungan untuk dijadikan jarak
  float distance = duration/58.2;
  float real_distance = (-0.0000301830225832412*(pow(distance,3)) +
  0.002971945737084*(pow(distance,2)) + 0.954323644592859*(distance) +
  0.215679022247498)-8;
  //distance = (34*duration)/2

  /* Kirim angka negatif ke komputer dan Turn LED ON
  untuk menunjukkan "di luar jangkauan" */

  if (real_distance >= maximumRange || real_distance <= minimumRange)
  {
    Serial.println("-1");digitalWrite(LEDPin, HIGH);
    delay (500);
```

```

}
else {

/*Kirim jarak ke komputer menggunakan Serial protokol, dan
menghidupkan LED OFF untuk menunjukkan membaca sukses. */
Serial.println(real_distance);
digitalWrite(LEDPin, LOW);

//waktu tunda 50mS
//delay(50);
delay(100);
}}

```

### C. Koding Sensor HC-SR04 (Jetson nano board)

```

import serial
#import csv

arduino = serial.Serial('/dev/ttyACM0', 9600, timeout=10)
#with open('demo1/datajarak/calibrationnyoba.txt', 'w') as f:
while True:
    data = arduino.readline()
    distance = data.decode('utf-8')
    intDistance = float(distance)

    print(realDistance)
    #f.write(distance)

```

### D. Koding Object detection

```

import cv2
import math
import csv
import numpy as np
import jetson.inference
import jetson.utils

net = jetson.inference.detectNet(argv=['--model=demo1/ssd-mobilenet.onnx', '--
labels=models/train3/labels.txt', '--input-blob=input_0', '--output-cvg=scores', '--
output-bbox=boxes'], threshold=0.5)

cap = cv2.VideoCapture(0)

```

```

cap.set(3,640)
cap.set(4,360)

dataDistance = []
fourcc = cv2.VideoWriter_fourcc('M','J','P','G')
out = cv2.VideoWriter('takevideo/jarak/5cm.avi', fourcc, 30.0, (640, 360))

while True:
    ret, img = cap.read()
    if ret == True:
        imgCuda = jetson.utils.cudaFromNumpy(img)
        detections = net.Detect(imgCuda, overlay = "OVERLAY_NONE")
        confidence = (d.Confidence)
        cx, cy = int(d.Center[0]), int(d.Center[1])
        x1,y1,x2,y2 = int(d.Left), int(d.Top), int(d.Right), int(d.Bottom)

        cv2.rectangle(img, (x1,y1),(x2,y2),(0,255,0),2)
        cv2.circle(img, (cx,cy), 2, (0,255,0), cv2.FILLED)
        garis = cv2.line(img,(cx,cy),(320,360), (0,255,0),1)
        cv2.putText(img, className, (x1+15, y1+15), cv2.FONT_HERSHEY_DUPLEX, 0.5,
(0,0,53), 2)
        cv2.putText(img, f'{confidence}', (x1+125, y1+15),
cv2.FONT_HERSHEY_DUPLEX, 0.5, (0,0,53), 2)
        cv2.putText(img, f'FPS:{int(net.GetNetworkFPS())}', (30,30),
cv2.FONT_HERSHEY_DUPLEX, 1, (255,0,0), 2)

        if (className=='mobil'):
            pos_x, pos_y = cx-320, 360-cy

        out.write(img)
        cv2.imshow("Video", img)
        cv2.waitKey(1)

cap.release()
out.release()
cv2.destroyAllWindows()

```

## BIODATA PENULIS



Penulis bernama lengkap Akhmad Ibnu Hija, dilahirkan di Blitar, 15 Mei 1996. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis telah menyelesaikan pendidikan di SDN Semen 01 pada tahun 2008, SMPN 1 Wlingi pada tahun 2011, SMAN 1 Talun pada tahun 2014 dan S1 Teknik Fisika ITS pada tahun 2019. Penulis di terima sebagai mahasiswa S2 Teknik Fisika ITS dengan bidang keahlian rekayasa instrumentasi industri pada tahun 2019. Selama kuliah penulis aktif di Himpunan Mahasiswa Teknik Fisika pada tahun 2016 serta asisten dari Laboratorium Embedded System and Cyberphysics. Penulis fokus pada bidang minat rekayasa instrumentasi dan kontrol serta mengambil topik penelitian *“Rancang Bangun Sistem Presepsi Berbasis SSD (Single Shot Multibox Detector) Dengan Sensor Fusion Kamera Mono dan Ultrasonik”*. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini dapat menghubungi melalui [hija.ibnu@gmail.com](mailto:hija.ibnu@gmail.com)