

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGAMBILAN KEPUTUSAN TREM OTONOM

TESIS

**Karya tulis sebagai salah satu syarat
untuk memperoleh gelar Magister dari
Institut Teknologi Bandung**

**Oleh
KHANSA SALSABILA SUHAIMI
NIM: 23220006
(Program Studi Magister Teknik Elektro)**



**INSTITUT TEKNOLOGI BANDUNG
Januari 2023**

ABSTRAK

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGAMBILAN KEPUTUSAN TREM OTONOM

Oleh

Khansa Salsabila Suhaimi

NIM: 23220006

(Program Studi Magister Teknik Elektro)

Trem dan kereta api berbeda dalam hal lingkungan operasionalnya. Trem dapat melewati lingkungan terbuka dimana terdapat pengguna jalan raya. Hal ini membuat pengemudi trem dituntut untuk selalu memperhatikan situasi disekitar dan memprediksi kemungkinan situasi yang muncul. Namun tingginya beban kerja dan kelelahan dapat mengurangi kesadaran situasi pengemudi trem terhadap situasi sekitar. Oleh karena itu, tugas untuk selalu memprediksi perilaku dari pengguna jalan raya dan merencanakan aksi yang harus diambil menjadi masalah yang paling umum bagi pengemudi trem.

Penelitian ini bertujuan untuk membangun sistem pengambilan keputusan berdasarkan evaluasi penilaian risiko tabrakan objek terhadap trem yang dapat diimplementasikan dan diuji pada sistem *embedded*. Tahapan pengambilan keputusan yang dilakukan yaitu menilai risiko tabrakan menggunakan metode *risk assessment*, mengambil keputusan menggunakan metode *finite state machine* (FSM), dan merencanakan kecepatan menggunakan metode *model predictive control* (MPC) dan *fuzzy logic*. Sistem pengambilan keputusan dirancang agar trem mampu berjalan dari satu stasiun ke stasiun lainnya dan mampu mengatasi situasi dimana terdapat trem lain atau pengguna jalan raya di rel atau situasi dimana pengguna jalan raya melintasi rel. Maka aksi yang dilakukan adalah melakukan pengendali penjelajahan aktif, penghindar tabrakan, dan pengereman darurat.

Sistem pengambilan keputusan yang dirancang disimulasikan pada simulator Carla dan diimplementasikan pada NVIDIA Drive AGX Pegasus. Hasil simulasi menunjukkan bahwa trem dapat berjalan secara otonom dengan persentase keselamatan 80%, yaitu sistem dapat memprediksi lintasan dari setidaknya 11 objek dan mampu menghindari tabrakan di situasi persimpangan yang padat. Pengambilan keputusan pengendali penjelajahan aktif, penghindar tabrakan, dan pengereman darurat dapat menjaga keselamatan trem dengan persentase masing-masing sebesar 96.94%, 100%, dan 100%. Sistem tersebut telah diujikan pada trem di rel PT INKA dalam situasi terbatas, yaitu situasi aman dan situasi dimana pejalan kaki berada di rel. Hasilnya trem mampu berjalan secara otonom, yaitu mampu berjalan sesuai kecepatan maksimum jalur dan mampu menghindari tabrakan.

Kata kunci: sistem pengambilan keputusan, trem otonom.

ABSTRACT

DESIGN AND IMPLEMENTATION OF THE AUTONOMOUS TRAM DECISION-MAKING SYSTEM

By

Khansa Salsabila Suhaimi

NIM: 23220006

(Master's Program in Electrical Engineering)

Trams and trains differ in their operational environment. Trams can pass through open environments where there are road users. This causes tram drivers to always pay attention to the situation around them and predict possible situations that may arise. However, high workloads and fatigue can reduce tram drivers' situational awareness of the situation around them. Therefore, the task of always predicting the behavior of road users and planning actions to be taken are the most common problems for tram drivers.

This study aims to build a decision-making system based on an evaluation of the risk assessment of object collisions with trams, which can be implemented and tested on embedded systems. The stages of decision-making include assessing the risk of collision using the risk assessment method, making decisions using the finite state machine (FSM) method, and planning speed using the predictive control model (MPC) and fuzzy logic methods. The decision-making system is designed so that the tram can run from one station to another and overcome situations where there are other trams or road users on the rails or where road users cross the rails. The actions taken include adaptive cruise control, collision avoidance, and emergency braking.

The designed decision-making system was simulated using the Carla simulator and implemented in NVIDIA Drive AGX Pegasus. The simulation results show that the tram can run autonomously with a safety percentage of 80%; that is, the system can predict the trajectory of at least 11 objects and avoid collisions in busy intersection situations. The decision-making of adaptive cruise control, collision avoidance, and emergency braking can maintain tram safety with percentages of 96.94%, 100%, and 100%, respectively. The system was tested on trams on the PT INKA rail in limited situations, namely, safe situations and situations where pedestrians were on the rails. The result is that the tram is able to run autonomously; that is, it can run according to the maximum speed of the track and avoid collisions.

Keywords: *autonomous tram, decision-making system.*

PERANCANGAN DAN IMPLEMENTASI SISTEM PENGAMBILAN KEPUTUSAN TREM OTONOM

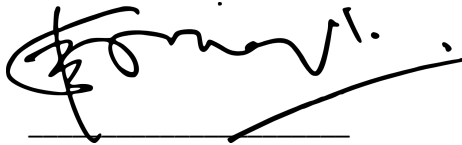
Oleh
Khansa Salsabila Suhaimi
NIM: 23220006
(Program Studi Magister Teknik Elektro)

Institut Teknologi Bandung

Menyetujui
Tim Pembimbing

Tanggal 23 Januari 2023

Ketua



Prof. Dr. Ir. Bambang Riyanto Trilaksono

Anggota

Ir. Arief Syaichu R, M.Eng.Sc.,Ph.D.

PEDOMAN PENGGUNAAN TESIS

Tesis Magister yang tidak dipublikasikan terdaftar dan tersedia di Perpustakaan Institut Teknologi Bandung, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada penulis dengan mengikuti aturan HaKI yang berlaku di Institut Teknologi Bandung. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin penulis dan harus disertai dengan kaidah ilmiah untuk menyebutkan sumbernya.

Sitasi hasil penelitian Tesis ini dapat di tulis dalam bahasa Indonesia sebagai berikut:

Suhami, K. S. (2023): *Perancangan dan Implementasi Sistem Pengambilan Keputusan Trem Otonom*, Tesis Program Magister, Institut Teknologi Bandung.

dan dalam bahasa Inggris sebagai berikut:

Suhami, K. S. (2023): *Design and Implementation of The Autonomous Tram Decision-making System*, Master's Thesis, Institut Teknologi Bandung.

Memperbanyak atau menerbitkan sebagian atau seluruh tesis haruslah seizin Dekan Sekolah Pascasarjana, Institut Teknologi Bandung.

HALAMAN PERUNTUKAN

Alhamdulillah robbil 'alamiin

*Dipersembahkan kepada Ayah, Umi, Adik-adik serta keluarga besarku tercinta
yang senantiasa mendukung lahir dan batin.*

KATA PENGANTAR

Puji syukur kehadiran Allah SWT. atas rahmat dan hidayah-Nya penulis dapat menyelesaikan tesis yang berjudul “Perancangan dan Implementasi Sistem Pengambilan Keputusan Trem Otonom” dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Magister Institut Teknologi Bandung.

Adapun selama proses penelitian dan penyusunan tesis ini, penulis mendapatkan bantuan dan dukungan dari berbagai pihak, oleh karena itu penulis menyampaikan terimakasih kepada:

1. Bapak Prof. Dr. Ir. Bambang Riyanto Trilaksono dan Bapak Ir. Arief Syaichu Rohman, M.Eng. Sc., Ph.D. sebagai tim pembimbing yang telah memberikan arahan, ilmu, serta waktunya dalam pengerjaan tesis ini.
2. Ayah, Umi, dan Adik-adik yang selalu memberikan dukungan.
3. Tim trem dan pihak-pihak lain yang turut membantu pada penyelesaian tesis ini yang tidak dapat penulis sebutkan satu-persatu.

Penulis menyadari bahwa penulisan tesis ini masih memiliki banyak kekurangan, oleh karena itu kritik dan saran yang membangun akan sangat penulis harapkan. Semoga tesis ini dapat bermanfaat bagi semua pihak, khususnya bagi pembaca.

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
HALAMAN PENGESAHAN	iii
PEDOMAN PENGGUNAAN TESIS	iv
HALAMAN PERUNTUKAN	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR GAMBAR DAN ILUSTRASI	ix
DAFTAR TABEL	xi
DAFTAR SINGKATAN DAN LAMBANG	xii
Bab I Pendahuluan	1
I.1. Latar Belakang	1
I.2. Rumusan Masalah	4
I.3. Tujuan Penelitian	5
I.4. Batasan Masalah	5
I.5. Metodologi	5
I.6. Sistematika Penulisan	6
Bab II Tinjauan Pustaka	7
II.1. Trem Baterai dan Wilayah Operasionalnya	7
II.2. Arsitektur Mengemudi Otonom	9
II.3. <i>Finite State Machine</i>	10
II.4. <i>Time to Collision</i>	11
II.5. <i>Model Predictive Control</i>	12
II.6. <i>Fuzzy Logic</i>	14
Bab III Perancangan Sistem	15
III.1. Spesifikasi Sistem Pengambilan Keputusan	15
III.2. Perancangan Perangkat Keras	15
III.3. Perancangan Perangkat Lunak Sistem Pengambilan Keputusan	17
III.4. Subsistem Penilai Risiko Tabrakan	19
III.4.1. Estimator Rel	20
III.4.2. Prediktor Lintasan	22
III.4.3. Penilaian Keselamatan	23
III.5. Subsistem Pengambilan Keputusan dan Perencanaan	24
III.5.1. Pengambilan Keputusan	26
III.5.2. Perencanaan Kecepatan	27
III.5.2.1. <i>Model Predictive Control</i> Sebagai Metode Perencanaan Kecepatan	28
III.5.2.2. <i>Logika Fuzzy</i> Sebagai Metode Perencanaan Kecepatan ...	29
Bab IV Pengujian dan Analisis	31
IV.1. Simulasi	31
IV.1.1. Simulasi lalu lintas sederhana	32
IV.1.2. Simulasi lalu lintas campuran	38
IV.2. Pengujian Testbed	43
IV.3. Pengujian Trem Otonom	49
Bab V Kesimpulan	51

DAFTAR PUSTAKA	53
----------------------	----

DAFTAR GAMBAR DAN ILUSTRASI

Gambar I.1. Diagram blok Rancangan Besar Trem Otonom.....	2
Gambar II.1. Trem baterai yang menjadi kendaraan uji.	7
Gambar II.2. Arsitektur mengemudi otonom. Secara berurutan dari atas, arsitektur mengemudi otonom adalah <i>sequential planning</i> , <i>behavior-aware planning</i> , dan <i>end-to-end planning</i>	9
Gambar II.3. Diagram FSM.	10
Gambar II.4. Struktur dasar MPC.	13
Gambar III.1. Rancangan Perangkat Keras Trem Otonom.....	16
Gambar III.2 Perangkat keras. (a) konfigurasi sensor; (b) peletakan perangkat didalam trem.	17
Gambar III.3. Diagram blok sistem pengambilan keputusan.	18
Gambar III.4. Diagram blok subsistem penilai risiko tabrakan.	18
Gambar III.5. Diagram alir subsistem penilai risiko tabrakan.	19
Gambar III.6. Analisa skor ancaman pada variasi kecepatan trem.	23
Gambar III.7. Diagram blok subsistem pengambilan keputusan dan perencanaan.	24
Gambar III.8. Diagram alir subsistem pengambilan keputusan dan perencanaan.	25
Gambar III.9. Diagram FSM.....	27
Gambar III.10. Fungsi keanggotaan variabel DTC.....	29
Gambar III.11. Fungsi keanggotaan variabel TTC.	30
Gambar III.12. Fungsi keanggotaan variabel tingkat pengereman.	30
Gambar IV.1. Lingkungan pengujian di simulator.	31
Gambar IV.2. Situasi yang disimulasikan di Carla. (a) situasi aman; (b) situasi terdapat pengguna jalan raya di rel; (c) situasi terdapat pengguna jalan raya melintasi rel.....	33
Gambar IV.3. Profil kecepatan situasi aman. (<i>merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM</i>).	34
Gambar IV.4. Profil kecepatan situasi terdapat pengguna jalan raya di rel. (<i>merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM</i>).	34
Gambar IV.5. Profil jarak relatif pengguna jalan raya yang berada di rel terhadap trem. (<i>hitam: jarak relatif, magenta: jarak batas aman</i>).	35
Gambar IV.6. Profil kecepatan situasi terdapat pengguna jalan raya melintasi rel. (<i>merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM</i>).	36
Gambar IV.7. Profil kecepatan situasi darurat. (<i>merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM</i>)	36
Gambar IV.8. <i>ground truth</i> posisi objek.	37
Gambar IV.9. Prediksi lintasan dibandingkan <i>ground truth</i> . (a) objek yang akan berbelok; (b) objek yang bergerak lurus. (<i>hitam: ground truth, merah: pengukuran, magenta: prediksi lintasan</i>).	37
Gambar IV.10. (a) trem menemui mobil yang diam di rel; (b) trem menemui objek yang menyebrang	40
Gambar IV.11. Kendaraan <i>testbed</i>	43
Gambar IV.12. Pengujian skenario pertama. (a) profil kecepatan; (b) state FSM dan perintah ke DbW. (<i>biru: set point kecepatan, merah: kecepatan aktual, hitam: state FSM, dan magenta: perintah ke DbW</i>).	45

Gambar IV.13. Pengujian skenario kedua. (a) profil kecepatan; (b) state FSM dan perintah ke DbW; (c) jarak relatif pengguna jalan raya terhadap trem; (d) TTC dan DTC.....	47
Gambar IV.14. Pengujian skenario ketiga. (a) profil kecepatan; (b) state FSM dan perintah ke DbW; (c) jarak relatif pengguna jalan raya terhadap trem.....	48
Gambar IV.15. Lingkungan pengujian di PT INKA.....	49
Gambar IV.16. Pengujian situasi aman dan situasi terdapat pengguna jalan raya diam di pinggir rel. (<i>merah: kecepatan aktual, hitam: state FSM</i>).....	50

DAFTAR TABEL

Tabel II.1. Spesifikasi Trem Baterai (<i>sumber: dokumentasi PT INKA</i>)	8
Tabel II.2. Pemetaan level <i>powering</i> dan <i>braking</i> (<i>sumber: dokumentasi PT INKA</i>).....	8
Tabel III.1. Parameter estimator rel.	21
Tabel IV.1. RMSD prediksi lintasan terhadap <i>ground truth</i>	38
Tabel IV.2. Hasil simulasi pada lalu lintas campuran.....	41
Tabel IV.3. Perbandingan <i>state</i> FSM yang terjadi dan target <i>state</i> FSM.	42

DAFTAR SINGKATAN DAN LAMBANG

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
LRV	<i>light rail vehicle</i>	1
KAI	PT. Kereta Api Indonesia	1
FSM	<i>Finite state machine</i>	8
TTC	<i>time-to-collision</i>	9
MPC	<i>Model predictive control</i>	10
DbW	<i>drive-by-wire</i>	14
AI	<i>artificial intellegent</i>	14
SoC	<i>system-on-chip</i>	14
GPU	<i>Graphics Processing Unit</i>	14
TOPS	<i>Tera Operations per Second</i>	14
DTC	<i>distance-to-collision</i>	19
KFCV	<i>kalman filter constant velocity</i>	21
TTS	<i>time to stop</i>	23
RMSD	<i>Root mean square deviation</i>	37
LAMBANG		
$\mathbf{X}$	posisi	10
$\dot{\mathbf{X}}$	Kecepatan	10
I_o	Panjang objek	10
$TTC_t(t)$	Time to collision dari trem	10
t_{alert}	Durasi sirine dibunyikan	21
v_t	Kecepatan trem	21
a_{min}	Percepatan minimum	21
a_{max}	Percepatan maksimum	21
s_{brake}	Jarak berhenti terpanjang	21
v_{max}	Kecepatan maksimum rel	21
s_{guard}	Jarak tambahan	21
s_{alert}	Jarak yang ditempuh selama memberi peringatan	21
s_H	Jarak area <i>rail horizon</i>	21
x_k	Vektor state kalman filter	22
F	Matriks transisi kalman filter	22
H	Matriks pengukuran kalman filter	22
a_e	Perlambatan maksimum	23
φ_{o_j}	Skor ancaman	23
σ	Ketidakpastian skor ancaman	23
Δt_e	selisih antara TTC dan TTS_{a_e}	24
d_o	Jarak relatif objek didepan trem	26

LAMBANG

Nama

Pemakaian pertama kali pada halaman

d_{ref}	Batas aman jarak relatif objek terhadap trem	26
A	Matriks <i>state-space</i>	27
B	Matriks <i>state-space</i>	27
C	Fungsi biaya	27
w_d	Bobot jarak	27
w_v	Bobot kecepatan	27
w_{rate}	Bobot laju percepatan	27
w_a	Bobot percepatan	27
$d_{rel,i+1}$	Jarak relatif objek terhadap trem	27
$v_{rel,i+1}$	Kecepatan relatif objek terhadap trem	27
a_{i+1}	Percepatan trem	27

Bab I Pendahuluan

I.1. Latar Belakang

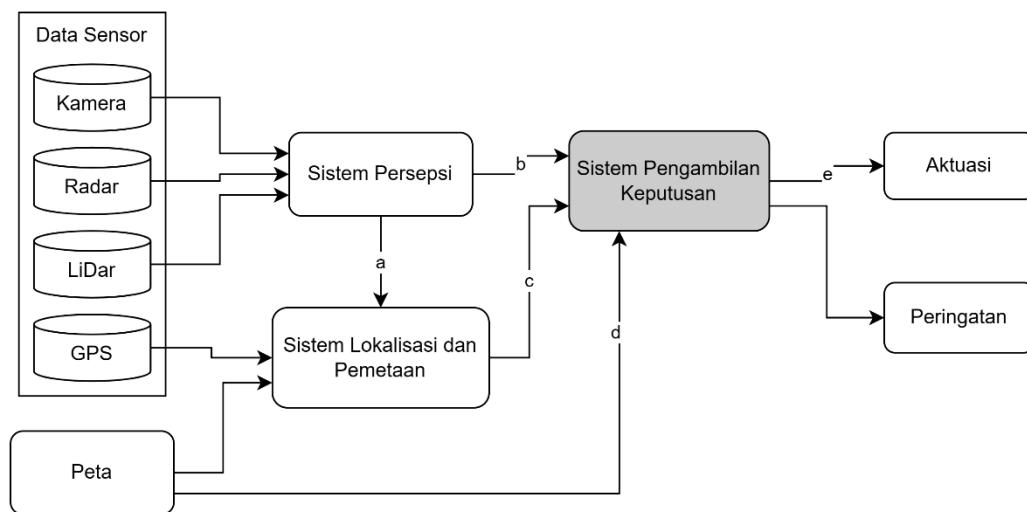
Transportasi umum berbasis rel sedang menjadi tren dalam beberapa tahun terakhir. Salah satu transportasi umum berbasis rel adalah trem. Trem merupakan kereta api ringan atau *light rail vehicle* (LRV) yang beroperasi di dalam kota. Jaringan rel yang dimiliki trem dapat terdiri dari rel khusus yang terpisah dari jalan raya, rel yang berdampingan dengan jalan raya namun dipisahkan oleh pembatas dan melintasi perlintasan sebidang, dan rel yang berbaur dengan jalan raya.

Dalam penelitian mengenai sistem manajemen jalur trem untuk Indonesia, Damayanti dkk (2015) memaparkan bahwa kesadaran akan keselamatan dan disiplin dari pengguna jalan masih rendah. PT. Kereta Api Indonesia (KAI) mencatat telah terjadi 395 kali kecelakaan kereta api pada tahun 2018, 260 kali pada tahun 2019, dan 198 kali hingga awal bulan oktober di tahun 2020 (Wahyudi, E, 2019). Kecelakaan tersebut terjadi di perlintasan sebidang, baik yang terdapat palang pintu perlintasan ataupun tidak terdapat palang pintu perlintasan. Menurut Di Palma dkk (2020) kecelakaan yang melibatkan trem disebabkan oleh kelalaian pengguna jalan raya seperti melintas tanpa memperhatikan jalur trem dan salah memperhitungkan waktu berhenti trem.

Pada sisi lain, kecelakaan yang melibatkan trem juga disebabkan oleh tingkat kesadaran situasi dari pengemudi trem. Studi pada kecelakaan trem di Australia yang dilakukan oleh Naweed dan Rose (2015) mengungkapkan bahwa rendahnya kesadaran situasi pengemudi trem terhadap sekitar memiliki peran utama dalam kecelakaan. Selama berjalan, pengemudi trem dituntut untuk selalu memperhatikan situasi disekitar dan memprediksi kemungkinan situasi yang muncul. Pengemudi trem harus mampu memprediksi perilaku dari trem lain, kendaraan di jalan raya, dan pejalan kaki yang melintas. Berkurangnya tingkat konsentrasi pengemudi akibat faktor beban kerja dan kelelahan dapat mempengaruhi kesadaran pengemudi terhadap situasi disekitar. Kesadaran situasi akan mempengaruhi keputusan yang diambil oleh pengemudi dan berdampak pada keselamatan trem. Dalam beberapa

kasus, kurangnya kesadaran situasi terkait dengan faktor teknis manusia, seperti tidak dapat melihat sinyal karena pandangan pengemudi terhalang, ataupun pemahaman terhadap situasi. Namun, masalah pengemudi trem yang paling umum berkaitan dengan prediksi perilaku dari pengguna jalan raya dan perencanaan kedepan (Naweed dan Rose, 2015).

Saat ini, teknologi komputer dan sensor dianggap mampu mengurangi faktor kesalahan manusia dalam mengemudi karena lebih akurat dan konsisten. Penelitian terkait bantuan mengemudi dan kendaraan otonom telah banyak dilakukan (Ibañez-Guzmán dkk., 2012, Sun dkk., 2022, Liu dkk., 2021). Mereka memanfaatkan sensor-sensor untuk mempersepsi lingkungan sekitar dan membuat keputusan berupa aksi untuk merespon situasi yang ditemui. Secara umum arsitektur kendaraan otonom terdiri dari persepsi, lokalisasi, perencanaan, dan kendali. Teknologi otonom tersebut dapat dimanfaatkan untuk menggantikan peranan pengemudi trem dalam memprediksi perilaku pengguna jalan raya dan mengambil keputusan terhadap situasi yang dihadapi.



Gambar I.1. Diagram blok Rancangan Besar Trem Otonom.

Rancangan besar trem otonom dalam program riset produktif-inovatif (RISPRO) berjudul “Pengembangan Sistem Otonomi dengan Menggunakan Kecerdasan Artifisial untuk Trem” mengintegrasikan sistem persepsi, sistem lokalisasi dan

pemetaan, dan sistem pengambilan keputusan. Diagram blok rancangan besar trem otonom tersebut seperti yang ditunjukkan pada Gambar I.1.

Sinyal-sinyal pada Gambar I.1 sebagai berikut:

1. Sinyal “a” merupakan *masking* rel yang dihasilkan oleh sistem persepsi untuk menjadi masukan bagi sistem lokalisasi dan pemetaan.
2. Sinyal “b” adalah sinyal masukan untuk sistem pengambilan keputusan berupa hasil deteksi objek di sekitar yang telah dilakukan pengolahan oleh sistem persepsi. Data yang diterima dalam sistem koordinat global terhadap peta.
3. Sinyal “c” adalah sinyal masukan untuk sistem pengambilan keputusan berupa pose dan kecepatan trem dalam sistem koordinat global terhadap peta.
4. Sinyal “d” merupakan masukan ke sistem pengambilan keputusan berupa *waypoint* dari stasiun awal ke stasiun tujuan.
5. Sinyal “e” merupakan keluaran dari sistem pengambilan keputusan berupa *set point* kecepatan yang harus dicapai oleh sistem aktuasi.

Sistem persepsi bertugas mengolah data langsung dari tiap sensor untuk menghasilkan deteksi, klasifikasi, dan pelacakan dari objek-objek disekitar. Hasil pengolahan yang dilakukan dimanfaatkan oleh sistem pengambilan keputusan untuk memahami situasi disekitar trem. Segmentasi juga dilakukan oleh sistem persepsi untuk melakukan *masking* rel yang hasilnya dimanfaatkan oleh sistem lokalisasi dan pemetaan. Selain itu, terdapat sistem lokalisasi dan pemetaan yang bertugas untuk melakukan lokalisasi posisi trem dan memperbarui informasi peta secara terus-menerus. Setelah pengenalan terhadap situasi sekitar dilakukan dan mengetahui pose trem, trem perlu memilih aksi yang tepat sesuai dengan kedua hal tersebut. Pemilihan aksi menjadi tugas dari sistem pengambilan keputusan. Hasil keputusan yang diambil kemudian dihitung *set point* kecepatan yang dibutuhkan sesuai pemetaan tingkat *master control*. Tingkat *master control* tersebut kemudian diberikan ke sistem aktuasi trem agar pengendali tingkat rendah trem menjalankan perintah yang diberikan.

Pembahasan utama pada penelitian ini mengenai sistem pengambilan keputusan yang merupakan bagian dari rancangan besar trem otonom. Studi terkait trem otonom telah dilakukan oleh peneliti-peneliti sebelumnya. Penelitian yang dilakukan Patlins dkk (2010) mengusulkan sistem penghindaran tabrakan untuk objek didepan berdasarkan jarak berhenti trem dan jarak terhadap objek dimana aksi yang diberikan berupa pengereman, sedangkan penelitian lainnya yang dilakukan oleh Lüy dkk (2018) dan Katz dan Schulz (2013) mengusulkan sistem penghindaran tabrakan menggunakan metode *risk assessment* untuk mengestimasi kemungkinan tabrakan dimana sistem akan memberikan peringatan kepada pengemudi apabila terdapat kemungkinan tabrakan. Selain itu, penelitian yang dilakukan Li (2018) mempertimbangkan tabrakan depan, tabrakan belakang, tabrakan samping, dan kelebihan kecepatan untuk trem dengan mengoptimasi keselamatan, biaya energi, kenyamanan, ketepatan berhenti, dan ketepatan jadwal. Penelitian-penelitian tersebut berada dalam tahap bantuan mengemudi dan telah diuji pada lingkungan terbatas, kecuali Li (2018) yang belum memaparkan hasil implementasinya. Namun, tahap bantuan mengemudi dirasa kurang karena sistem terbatas pada memberi peringatan dan menghentikan trem jika tidak ada reaksi dari pengemudi trem. Sistem juga perlu menangani perjalanan antar stasiun, penghindaran tabrakan, dan pengereman darurat. Untuk tugas yang lebih kompleks tersebut, sistem perlu mempertimbangkan prediksi lintasan dari objek-objek disekitar. Hal ini belum dilakukan oleh penelitian-penelitian sebelumnya.

I.2. Rumusan Masalah

Mengemudikan trem di lalu lintas campuran merupakan tugas yang kompleks bagi pengemudi trem, karena beban kerjanya tinggi dan membutuhkan kerja parallel antara mewaspadaai situasi-situasi di sekitar dan memanipulasi *throttle* seperti momentum dan kompensasi gradien. Mengemudikan trem berbeda dengan mengemudikan kereta api yang beroperasi dalam lingkungan tertutup, juga berbeda dengan mengemudikan kendaraan karena karakteristik fisik trem berbeda. Maka, penghindaran tabrakan untuk trem di situasi lalu lintas campuran sangat menantang karena tidak dapat bermanuver dan tidak dapat melambat secepat mobil.

Sistem pengambilan keputusan perlu dirancang dengan menyesuaikan karakteristik trem sehingga trem mampu berjalan di lalu lintas campuran dengan selamat. Selain itu, juga dibutuhkan sistem pengambilan keputusan yang dapat diimplementasikan ke sistem *embedded* bersama dengan sistem lainnya yang mendukung kinerja otonom.

I.3. Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka penelitian ini bertujuan untuk:

1. Membangun sistem pengambilan keputusan berdasarkan evaluasi penilaian risiko tabrakan dari setiap objek yang berada di sekitar terhadap trem sehingga sistem mampu melakukan perjalanan antar stasiun sambil menjaga jarak apabila terdapat objek di rel dan mengantisipasi objek yang akan menyebrangi rel.
2. Mengimplementasikan sistem pengambilan keputusan pada trem.

I.4. Batasan Masalah

Penelitian ini terbatas pada hal-hal sebagai berikut:

1. Fokus penelitian pada sistem pengambilan keputusan.
2. Lingkungan uji trem pada jalur lurus yang berada di lalu lintas campuran.
3. Mengatasi situasi aman, situasi pengguna jalan raya menyebrang, situasi bertemu trem lain atau pengguna jalan raya di rel, dan situasi darurat.

I.5. Metodologi

Pada pengerjaan laporan penelitian ini digunakan metodologi sebagai berikut:

1. Identifikasi kebutuhan
Pada tahap ini dilakukan studi mengenai kebutuhan pada pengembangan sistem trem otonom.
2. Studi Literatur
Pada tahap ini dilakukan studi literatur dari buku, jurnal, tesis, dan situs internet mengenai pengambilan keputusan dan penghindaran tabrakan yang dapat diterapkan pada trem.
3. Perancangan dan Spesifikasi Sistem

Pada tahap ini dilakukan penentuan spesifikasi sistem yang diinginkan dan penentuan desain sistem.

4. Simulasi

Pada tahap ini dilakukan simulasi sistem pengambilan keputusan pada simulator Carla.

5. Implementasi

Pada tahap ini dilakukan implementasi sistem pengambilan keputusan pada sistem *embedded*.

6. Pengujian dan Analisis

Pada tahap ini dilakukan uji coba sistem pengambilan keputusan secara bertahap dari tahap simulasi, kendaraan *testbed* hingga uji coba sistem pengambilan keputusan di trem.

7. Kesimpulan dan Saran

Pada tahap ini dilakukan analisis dari hasil yang diperoleh selama proses pengujian hingga dapat ditarik kesimpulan dari penelitian.

I.6. Sistematika Penulisan

Penulisan laporan penelitian ini dibuat dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, tujuan yang ingin dicapai, batasan masalah, metodologi, dan sistematika penulisan dari laporan penelitian.

2. BAB II TINJAUAN PUSTAKA

Bab ini membahas mengenai landasan teori dan penelitian-penelitian terkait yang menjadi acuan penelitian.

3. BAB III PERANCANGAN SISTEM

Bab ini berisi perancangan sistem pengambil keputusan.

4. BAB IV PENGUJIAN DAN ANALISIS

Bab ini membahas mengenai simulasi dan implementasi sistem pengambilan keputusan, pengujian sistem, dan pembahasan mengenai data-data pengujian.

5. BAB V KESIMPULAN DAN SARAN

Bab ini berisi simpulan dari hasil yang didapat pada penelitian ini, serta saran peneliti untuk pengembangan lebih lanjut.

Bab II Tinjauan Pustaka

II.1. Trem Baterai dan Wilayah Operasionalnya

Trem merupakan kereta api ringan atau light rail vehicle (LRV) yang beroperasi di dalam kota. Seperti kereta api pada umumnya, trem berjalan diatas rel dan membutuhkan manipulasi throttle seperti momentum dan kompensasi gradien. Perbedaannya terletak pada lingkungan operasional trem yang lebih kompleks. Trem dapat melewati lingkungan tertutup dengan lintasan khusus yang terpisah dari jalan raya. Namun juga dapat melewati lingkungan terbuka dimana terdapat pengguna jalan raya. Hal ini membuat tuntutan dalam mengemudikan trem mirip dengan mengemudikan kendaraan.



Gambar II.1. Trem baterai yang menjadi kendaraan uji.

Trem yang menjadi kendaraan uji adalah trem baterai milik PT INKA. Trem yang menjadi kendaran uji ditunjukkan oleh Gambar II.1. Trem tersebut memiliki berat 48 ton dengan kecepatan maksimum operasi 60 km/jam seperti yang ditunjukkan oleh Tabel II.1. Pada pengoperasiannya, trem dikendalikan oleh *master control* yang memiliki tujuh level *powering*, lima level *braking*, dan netral. Masinis akan mengganti level *master control* melalui tuas kemudi yang terdapat pada dasbor dan

master control akan mengkodekan level yang diminta untuk menggerakkan mesin trem. Selain itu, trem tersebut sudah dilengkapi oleh *drive-by-wire* yang akan meneruskan sinyal kendali ke *master control*. *Master control* diatur sebagai kendali umpan balik yang melacak kecepatan yang diberikan. Tujuh level *powering* dipetakan sebagai kecepatan, sedangkan level *braking* dipetakan sebagai bukaan katup. Pemetaan *powering* dan *braking* ditunjukkan oleh Tabel II.2.

Tabel II.1. Spesifikasi Trem Baterai (*sumber: dokumentasi PT INKA*)

Parameter	Nilai
Berat	48 ton
Kecepatan maksimum operasi	60 <i>km/jam</i>
Akselerasi	0.22 <i>m/s²</i>
Koefisien adhesi	0.172

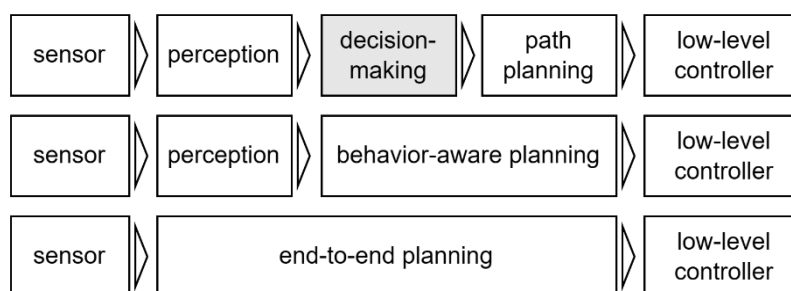
Tabel II.2. Pemetaan level *powering* dan *braking* (*sumber: dokumentasi PT INKA*).

Parameter	Satuan	Nilai	Keterangan
P1	m/s	0.511	Nilai kecepatan dari level <i>powering</i> master control P1
P2	m/s	1.102	Nilai kecepatan dari level <i>powering</i> master control P2
P3	m/s	1.666	Nilai kecepatan dari level <i>powering</i> master control P3
P4	m/s	2.179	Nilai kecepatan dari level <i>powering</i> master control P4
P5	m/s	3.844	Nilai kecepatan dari level <i>powering</i> master control P5
P6	m/s	4.613	Nilai kecepatan dari level <i>powering</i> master control P6
P7	m/s	5.126	Nilai kecepatan dari level <i>powering</i> master control P7

B1	%	0.25	Persentase kekuatan pengereman dari level braking master control B1
B2	%	0.25	Persentase kekuatan pengereman dari level braking master control B2
B3	%	0.5	Persentase kekuatan pengereman dari level braking master control B3
B4	%	0.75	Persentase kekuatan pengereman dari level braking master control B4
B5	%	1.0	Persentase kekuatan pengereman dari level braking master control B5

II.2. Arsitektur Mengemudi Otonom

Mengemudi otonom merupakan integrasi kompleks dari banyak teknologi. Mengemudi otonom melibatkan serangkaian teknologi yang sangat beragam, seperti persepsi, pengambilan keputusan, sistem operasi waktu nyata, komputasi heterogen, komputasi awan, pemrosesan video/gambar, dan sebagainya. Dari sisi algoritma, teknologi yang terlibat adalah penginderaan menggunakan sensor-sensor, persepsi yang melokalisasi kendaraan ego dan memahami lingkungan sekitar, serta pengambilan keputusan untuk mencapai tujuan dengan selamat.



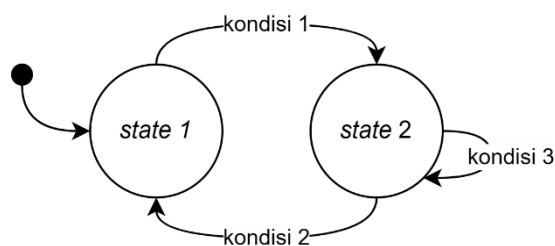
Gambar II.2. Arsitektur mengemudi otonom. Secara berurutan dari atas, arsitektur mengemudi otonom adalah *sequential planning*, *behavior-aware planning*, dan *end-to-end planning*.

Pada tinjauan arsitektur mengemudi otonom yang dilakukan oleh Schwarting dkk (2018), terdapat tiga arsitektur yang berbeda untuk mengemudi otonom, yaitu

sequential planning, *behavior-aware planning*, dan *end-to-end planning*. Arsitektur *sequential planning* memisahkan *perception* atau persepsi, *decision-making* atau pengambilan keputusan, dan *path planning* atau perencanaan jalur dalam tiga tahap berurutan. Pada arsitektur *behavior-aware planning*, perencanaan jalur dilakukan oleh pengambilan keputusan, sedangkan *end-to-end planning* mengintegrasikan persepsi, pengambilan keputusan, dan perencanaan jalur. Perbedaan tiga arsitektur mengemudi otonom seperti yang ditunjukkan oleh Gambar II.2. Penelitian ini mengadaptasi arsitektur *sequential planning* untuk membangun trem otonom dimana tahap yang menjadi fokus penelitian adalah tahap pengambilan keputusan. Bagian ini disorot pada Gambar II.2.

II.3. Finite State Machine

Finite state machine (FSM) adalah model sistem dengan dinamika diskrit yang setiap reaksinya memetakan penilaian masukan ke penilaian keluaran dimana kumpulan *state* yang mungkin adalah terbatas (Lee dan Seshia, 2017). FSM dapat digambarkan dengan diagram seperti yang ditunjukkan pada Gambar II.3.



Gambar II.3. Diagram FSM.

Pada Gambar II.3, kumpulan *state* terdiri dari *state 1* dan *state 2* dimana *state* awal ditunjuk oleh panah dengan dot hitam. Antar *state* pada FSM dapat bertransisi dari satu *state* ke *state* lainnya apabila memenuhi kondisi transisi. Pada Gambar, kondisi transisi ditunjukkan oleh panah berlabel kondisi 1, kondisi 2, dan kondisi 3. Panah tersebut menunjukkan transisi dari satu *state* ke *state* lainnya. Transisi juga dapat dimulai dan berakhir di *state* yang sama seperti kondisi 3.

Selain digambarkan dalam diagram, FSM dapat ditulis dalam persamaan matematika dengan 5-*tuple*. **states** adalah kumpulan terbatas dari *state*, **inputs** adalah kumpulan penilaian masukan, **outputs** adalah kumpulan penilaian keluaran, **update** adalah fungsi pembaruan yang memetakan *state* dan penilaian input ke *state* selanjutnya dan penilaian output, **initialState** adalah *state* awal.

$$(\mathbf{states}, \mathbf{inputs}, \mathbf{outputs}, \mathbf{update}, \mathbf{initialState})$$

Pada tiap reaksi, terdapat *state* saat ini yang dapat bertransisi ke *state* berikutnya. *State* berikutnya tersebut kemudian menjadi *state* saat ini dari reaksi berikutnya.

Misalkan fungsi yang memberikan *state* FSM pada reaksi $n \in \mathbb{N}$ adalah $s: \mathbb{N} \rightarrow \mathbf{states}$ dengan *state* awal $s(0) = \mathbf{initialState}$. $x: \mathbb{N} \rightarrow \mathbf{inputs}$ dan $y: \mathbb{N} \rightarrow \mathbf{outputs}$ menunjukkan penilaian masukan dan keluaran pada tiap reaksi. Maka $x(0) \in \mathbf{inputs}$ adalah penilaian masukan pertama dan $y(0) \in \mathbf{outputs}$ adalah penilaian keluaran pertama. Persamaan (II.1) menunjukkan dinamika FSM yaitu *state* dan keluaran berikutnya dipandang dari segi *state* dan masukan saat ini. Fungsi pembaruan mengkodekan semua transisi, kondisi, dan spesifikasi keluaran dalam FSM.

$$(s(n+1), y(n)) = \mathbf{update}(s(n), x(n)) \quad (\text{II.1})$$

II.4. Time to Collision

Waktu menuju tabrakan atau *time-to-collision* (TTC) adalah indikator keselamatan standar berbasis waktu. TTC diperkenalkan oleh Hayward (1971) sebagai rentang waktu yang tersisa sebelum dua objek atau lebih bertabrakan jika tidak ada aksi penghindaran yang dilakukan. TTC menjadi parameter utama dalam mendesain sistem penghindaran tabrakan (Van der horst dan Hogema, 1993) terutama ketika dihadapkan pada pertanyaan kapan dan bagaimana pengemudi harus melakukan percepatan atau perlambatan (Wang dkk., 2008). Selain itu, TTC digunakan oleh beberapa peneliti sebagai indikasi untuk menentukan aksi yang harus diambil (Lefèvre dkk., 2014).

TTC mengasumsikan kendaraan ego dan objek berjalan dengan kecepatan konstan hingga terjadi tabrakan, sehingga percepatan dan perlambatan dari kedua kendaraan tersebut diabaikan. Persamaan untuk menghitung TTC dituliskan oleh Saffarzadeh dkk (2013) sebagai fungsi waktu yang mempertimbangkan posisi kendaraan ego dan objek, kecepatan keduanya, dan panjang badan kendaraan target dalam arah longitudinal. Persamaan TTC diberikan oleh (II.2). X adalah posisi dan $\dot{X}$ adalah kecepatan. Subskrip t adalah kendaraan ego, yaitu trem dalam penelitian ini, dan subskrip o adalah objek. I_o adalah Panjang objek.

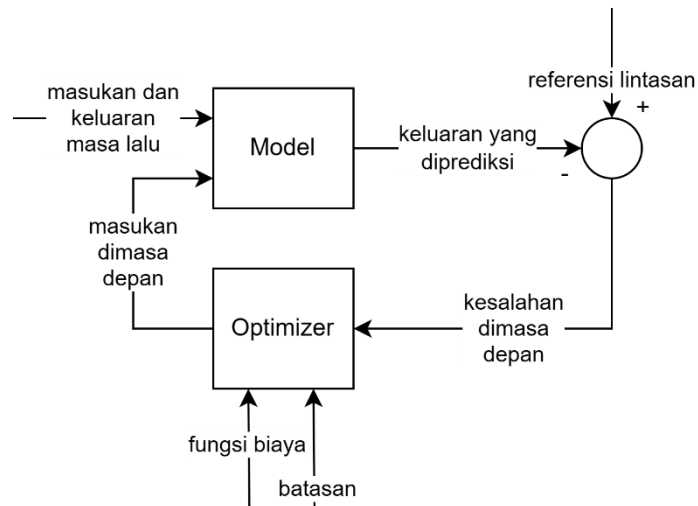
$$TTC_t(t) = \frac{X_o(t) - X_t(t) - I_o}{\dot{X}_t(t) - \dot{X}_o(t)} \quad (II.2)$$

Waktu untuk mengambil aksi biasanya ditentukan oleh suatu batas ambang. Batas ambang tersebut menjadi pembeda antara situasi aman dan tidak aman. Nilai batas ambang TTC bergantung pada perilaku pengemudi sehingga tidak ada nilai pasti yang memisahkan antara situasi aman dan tidak aman. Studi sistem peringatan tabrakan yang dilakukan Hirst dan Graham (1997) menunjukkan penggunaan batas ambang 4 atau 5 detik terjadi banyak alarm palsu dibandingkan batas ambang 3 detik. Studi yang dilakukan Hogema dan Janssen (1996) pada pengemudi yang memiliki bantuan mengemudi dan yang tidak memiliki bantuan mengemudi menunjukkan batas ambang minimal untuk pengemudi tanpa bantuan mengemudi adalah 3 – 5 detik, sedangkan batas ambang untuk pengemudi dengan bantuan pengemudi adalah 2.6 detik.

II.5. Model Predictive Control

Model predictive control (MPC) adalah metode kontrol yang menggunakan model proses secara eksplisit untuk mendapatkan sinyal kontrol dengan meminimalkan fungsi objektif (Camacho dan Bordons, 1999). Struktur dasar MPC terdiri dari model dan pengoptimal atau *optimizer* seperti yang ditunjukkan pada Gambar II.4. Model digunakan untuk memprediksi keluaran *plant* selanjutnya berdasarkan nilai masukan dan keluaran sebelumnya, serta tindakan kendali optimal selanjutnya.

Tindakan tersebut dihitung oleh pengoptimal dengan mempertimbangkan fungsi biaya serta batasannya.



Gambar II.4. Struktur dasar MPC.

Model proses menjadi komponen yang sangat penting dalam pengendali. Model yang dipilih harus mampu menangkap dinamika proses sehingga dapat memprediksi keluaran selanjutnya secara tepat. Selain itu model juga harus sederhana untuk diterapkan dan dipahami. Terdapat beberapa jenis model yang digunakan dalam formulasinya, diantaranya adalah model fungsi alih dan *state-space*.

Pengoptimal adalah bagian yang fundamental karena menghasilkan tindakan kendali. Apabila fungsi biaya adalah fungsi kuadratik, maka nilai minimumnya dapat diperoleh sebagai fungsi eksplisit (linear) dari masukan dan keluaran sebelumnya serta lintasan referensi selanjutnya. Apabila batasan adalah *inequality* maka solusinya harus diperoleh dengan algoritma numerik yang lebih berat secara komputasi. Ukuran masalah optimisasi bergantung pada jumlah variabel dan cakrawala prediksi yang digunakan dan biasanya menjadi masalah optimisasi yang relatif sederhana. Namun jumlah waktu yang dibutuhkan untuk kasus yang dibatasi dan *robust* dapat lebih tinggi dari yang dibutuhkan untuk kasus yang tidak dibatasi.

II.6. Fuzzy Logic

Fuzzy logic adalah logika penalaran samar-samar atau antara yang mirip dengan logika penalaran manusia (Kusumadewi dan Purnomo, 2004). Teori kumpulan fuzzy didefinisikan oleh fungsi keanggotaan yang memiliki variasi halus dari pada batas tegas seperti pada nilai crisp. Untuk tiap variabel fuzzy, fungsi keanggotaan mendefinisikan suatu nilai real dalam rentang $[0,1]$ yang disebut derajat kebebasan. Derajat kebebasan 1 menunjukkan suatu nilai masukan merupakan bagian dari suatu himpunan fuzzy, derajat kebebasan 0 menunjukkan nilai masukan bukan termasuk bagian dari suatu himpunan fuzzy, dan derajat kebebasan diantaranya menunjukkan keanggotaan sebagian.

Bab III Perancangan Sistem

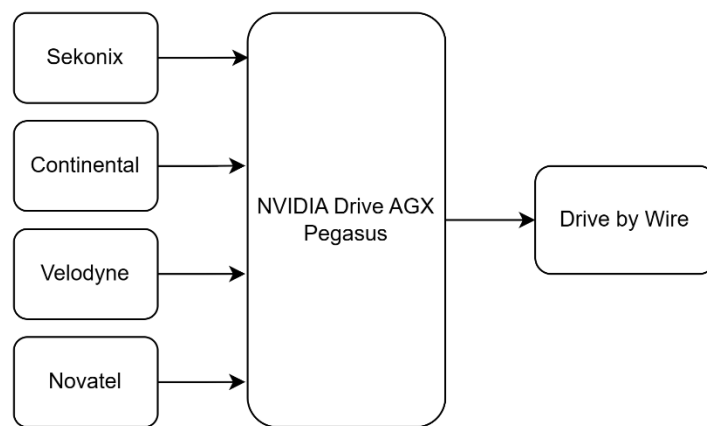
III.1. Spesifikasi Sistem Pengambilan Keputusan

Berikut merupakan spesifikasi sistem pengambilan keputusan untuk trem otonom yang dibangun:

1. Sistem mampu bekerja pada tiga jenis situasi, yaitu situasi aman (pengguna jalan tidak ada disekitar atau berjalan berdampingan searah trem), situasi bertemu trem lain atau pengguna jalan raya di rel, dan situasi pengguna jalan menyebrang rel.
2. Pada situasi aman, sistem mengambil keputusan untuk mengikuti kecepatan maksimum rel yang diperbolehkan.
3. Pada situasi bertemu trem lain atau pengguna jalan raya di rel, sistem mengambil keputusan untuk menjaga jarak.
4. Pada situasi pengguna jalan menyebrang rel, sistem mengambil keputusan untuk menurunkan kecepatan apabila terdapat potensi tabrakan.
5. Menaikkan atau menurunkan kecepatan dilakukan dengan memberikan perintah berupa tingkat *master control* ke *drive-by-wire* (DbW).

III.2. Perancangan Perangkat Keras

Sistem pengambilan keputusan untuk trem otonom menggunakan perangkat komputasi utama NVIDIA Drive AGX Pegasus. NVIDIA Drive AGX Pegasus adalah platform komputasi *artificial intellegent* (AI) yang dirancang khusus untuk mengemudi otonom level 5 dan mengemudi otomatis level 4 (NVIDIA Drive overview, 2019). Komputer ini tersusun dari dua *system-on-chip* (SoC) Xavier dan dua *Graphics Processing Unit* (GPU) NVIDIA Turing dengan kemampuan komputasi 320 *Tera Operations per Second* (TOPS). NVIDIA Drive AGX Pegasus dikonfigurasi dengan sensor-sensor yang didukung, yaitu kamera sekoni SF3324, LiDAR velodyne, radar continental ARS430RDI, dan GNSS+IMU NovAtel. Selain itu juga dikonfigurasi dengan sistem *drive-by-wire* untuk kendali mekanis trem. Diagram blok perangkat keras seperti yang ditunjukkan oleh Gambar III.1.



Gambar III.1. Rancangan Perangkat Keras Trem Otonom.

NVIDIA Drive AGX Pegasus akan menerima data dari tiap sensor secara langsung. Dari sensor kamera diperoleh citra lingkungan yang ditangkap, sedangkan dari radar dan LiDAR diperoleh data deteksi lingkungan dalam bentuk *point cloud*. Data-data tersebut akan diolah oleh sistem persepsi untuk memberikan informasi pada sistem pengambilan keputusan mengenai lingkungan sekitar. Selain itu NVIDIA Drive AGX Pegasus juga menerima tipe data string dari GNSS+IMU untuk lokalisasinya. Tipe data ini akan diolah oleh sistem lokalisasi dan pemetaan untuk memberikan informasi pada sistem pengambilan keputusan mengenai *state* trem. Komputasi sistem persepsi, sistem lokalisasi dan pemetaan, dan sistem pengambilan keputusan dijalankan di NVIDIA Drive AGX Pegasus. Keluaran dari sistem pengambilan keputusan dikirimkan ke *drive-by-wire* (DbW) melalui standar protokol modbus.

Pada trem otonom yang dirancang memiliki delapan kamera, satu LiDAR, satu radar, dan satu GNSS+IMU. Kamera, LiDAR, dan radar dipasang pada bagian luar trem dengan menggunakanudukan khusus. Kamera disusun dengan pengaturan 5-3, yaitu tiga buah kamera dipasang pada bagian muka trem untuk menangkap citra depan dengan sudut pandang yang rendah, dan lima lainnya dipasang dibagian kepala trem untuk menangkap citra depan dengan sudut pandang yang lebih tinggi. LiDAR dipasang di bagian kepala trem untuk menangkap deteksi lingkungan dalam arah depan dan radar dipasang di bagian muka trem dimana dapat mendeteksi

lingkungan dalam arah 90° . Selain itu, GNSS memiliki dua antenna yang diletakkan pada bagian luar depan trem, dan luar belakang trem, sedangkan perangkat IMU diletakkan di dalam trem. Peletakan sensor-sensor seperti yang ditunjukkan pada Gambar III.2 a. dan peletakan perangkat didalam trem seperti yang ditunjukkan pada Gambar III.2 b.



(a)



(b)

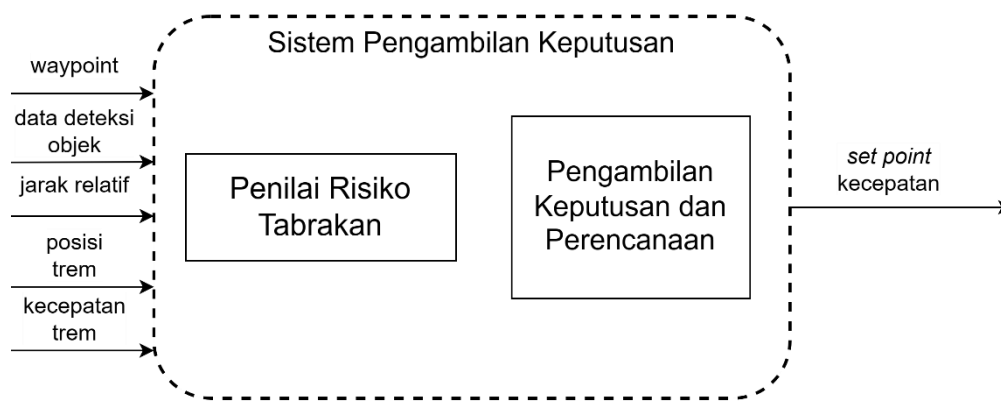
Gambar III.2 Perangkat keras. (a) konfigurasi sensor; (b) peletakan perangkat didalam trem.

III.3. Perancangan Perangkat Lunak Sistem Pengambilan Keputusan

Pengembangan perangkat lunak sistem pengambilan keputusan dilakukan di komputer host pendamping dengan GPU Nvidia RTX 2070. Sesuai spesifikasi sistem pengambilan keputusan pada subbab III.1, sistem pengambilan keputusan perlu menentukan aksi yang tepat ketika menemui tiga situasi, yaitu situasi aman, situasi bertemu trem lain atau pengguna jalan raya di rel, dan situasi pengguna jalan menyebrang rel.

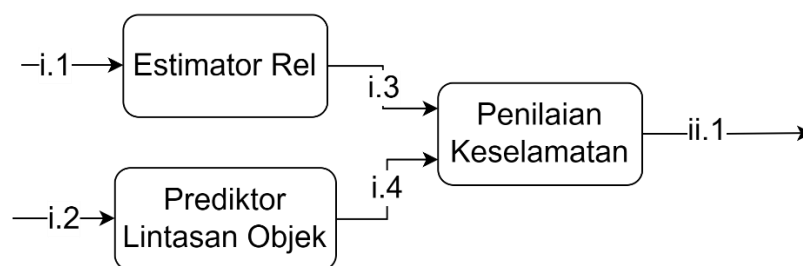
Perancangan sistem pengambilan keputusan pada penelitian ini menggunakan metode *risk assessment*. Metode *risk assessment* menilai risiko tabrakan dari tiap

objek yang terdeteksi dan menentukan aksi berdasarkan penilaiannya (Li dkk., 2021). Diagram blok sistem pengambilan keputusan seperti yang ditunjukkan pada Gambar III.3. Data yang diterima oleh sistem pengambilan keputusan merupakan keluaran dari sistem persepsi, sistem lokalisasi dan pemetaan, dan peta, secara berturut-turut yaitu data deteksi dan jarak relatif objek, data pose dan kecepatan trem, dan *waypoint*. Keluaran dari sistem pengambilan keputusan adalah *set point* kecepatan.



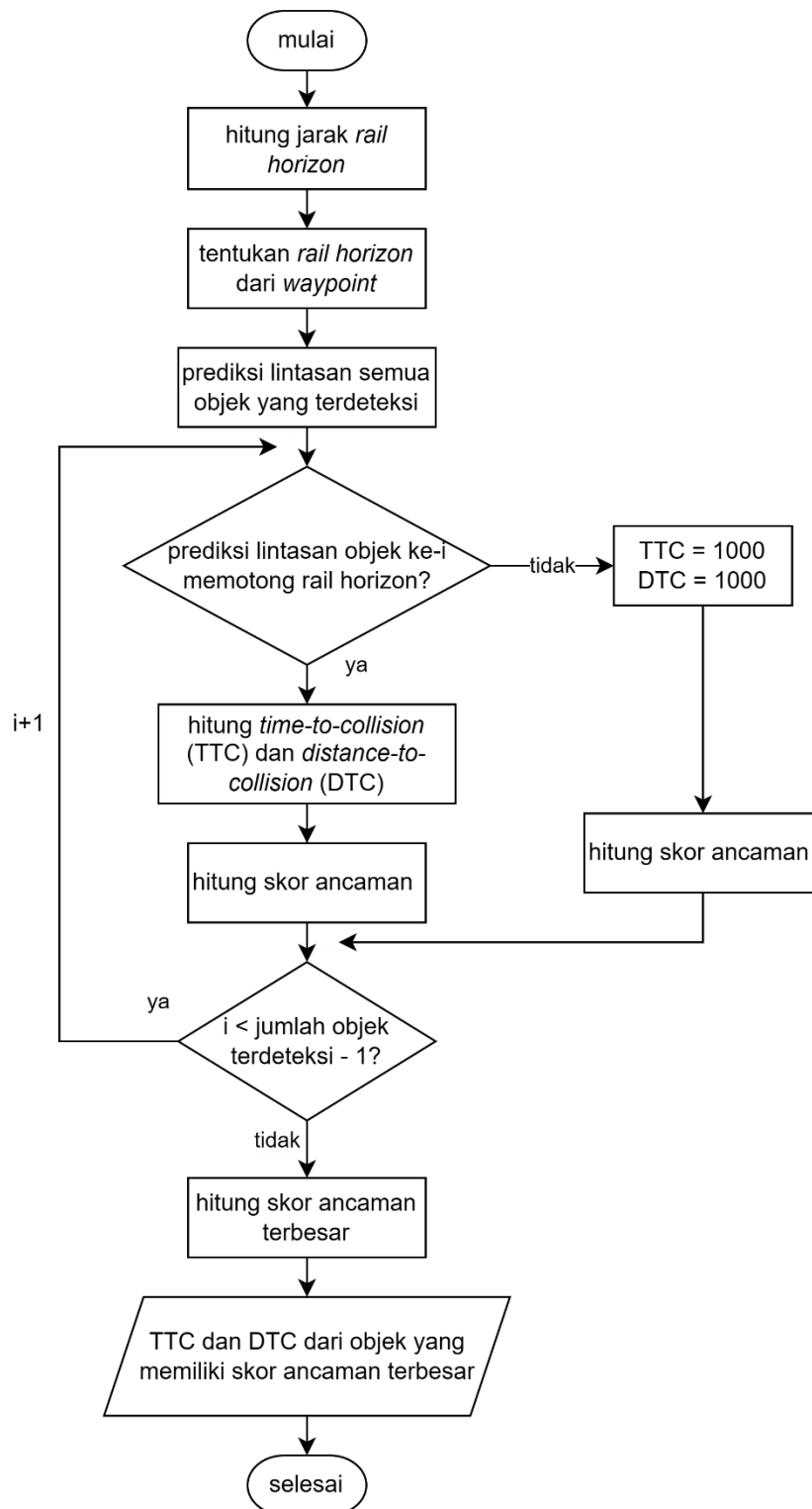
Gambar III.3. Diagram blok sistem pengambilan keputusan.

Sistem pengambilan keputusan terdiri dari subsistem penilai risiko tabrakan dan subsistem pengambilan keputusan dan perencanaan. Subsistem penilai risiko tabrakan menilai risiko tabrakan dari tiap objek yang terdeteksi. Hasilnya berupa waktu ke titik tabrakan atau *time-to-collision* (TTC) dan jarak ke titik tabrakan atau *distance-to-collision* (DTC). Hasil tersebut kemudian menjadi pertimbangan dalam memutuskan aksi oleh subsistem pengambilan keputusan dan perencanaan.



Gambar III.4. Diagram blok subsistem penilai risiko tabrakan.

III.4. Subsistem Penilai Risiko Tabrakan



Gambar III.5. Diagram alir subsistem penilai risiko tabrakan.

Subsistem penilai risiko tabrakan terdiri dari estimator rel, prediktor lintasan, dan penilaian keselamatan seperti yang ditunjukkan pada Gambar III.4. Pada Gambar tersebut, bagian estimator rel memanfaatkan data *waypoint*, pose dan kecepatan trem (i.1) untuk menentukan *region of interest planning* atau disebut sebagai *rail horizon* (i.3). Bagian prediktor lintasan memanfaatkan data deteksi objek (i.2) untuk memprediksi lintasan gerak objek tersebut (i.4). Keluaran dari estimator rel dan prediktor lintasan tersebut digunakan oleh bagian penilaian keselamatan untuk menilai seberapa besar kemungkinan trem bertabrakan dengan objek tersebut. Keluaran bagian penilaian keselamatan (ii.1) adalah TTC dan DTC.

Proses yang dilakukan subsistem ini dimulai dengan menghitung jarak *rail horizon* yang diperlukan lalu menentukan daerah rel yang didefinisikan oleh *waypoint* menjadi *rail horizon*. Kemudian seluruh objek yang terdeteksi diprediksi lintasan geraknya. *Rail horizon* dan prediksi lintasan dari seluruh objek dibandingkan untuk mengecek adanya titik perpotongan. Kemudian TTC dan DTC dapat dihitung untuk tiap prediksi lintasan gerak objek yang memotong *rail horizon* dan dipilih objek yang paling berbahaya. Diagram alir dari subsistem penilai risiko tabrakan seperti yang ditunjukkan pada Gambar III.5.

III.4.1. Estimator Rel

Tugas yang dilakukan estimator rel adalah menentukan *region of interest planning* atau disebut sebagai *rail horizon* dari *waypoint* antar stasiun. Panjang *rail horizon* bergantung pada kecepatan trem dengan parameter kecepatan maksimum rel dan deselerasi trem yang ditentukan.

Jarak *rail horizon* dihitung dengan mengadaptasi konsep *alert and advisory (command)* pada penelitian yang dilakukan oleh Lehner dkk (2008). Konsep *alert and advisory (command)* mempertimbangkan jarak untuk menentukan kapan sistem penghindaran tabrakan kereta api memberi peringatan ke masinis untuk melakukan pengereman. Jarak yang dipertimbangkan adalah jarak yang akan ditempuh selama memberi peringatan dan jarak yang akan ditempuh selama

pengereman. Kemudian untuk memastikan jarak pengereman cukup untuk kereta api berhenti, ditambahkan jarak tambahan.

Pada penelitian ini, jarak *rail horizon* perlu mempertimbangkan jarak yang dibutuhkan untuk memberi peringatan pada pengguna jalan raya dan jarak yang dibutuhkan trem untuk berhenti. Pertama-tama, sirine harus dibunyikan untuk memperingati pengguna jalan raya disekitar selama waktu tetap t_{alert} . Kemudian kecepatan diturunkan setelah t_{alert} detik untuk menghindari tabrakan. Maka jarak yang dibutuhkan untuk memperingati pengguna jalan raya bergantung pada kecepatan trem v_t dan waktu yang dibutuhkan untuk memberi peringatan t_{alert} . Sedangkan jarak yang dibutuhkan trem hingga berhenti bergantung pada kecepatan awal ketika akan melambat dan deselerasi yang mampu digunakan oleh trem. Jarak berhenti terpanjang terjadi apabila trem melambat dengan deselerasi terkecilnya a_{min} , sedangkan jarak berhenti terpendek terjadi apabila trem melambat dengan deselerasi terbesarnya a_{max} . Jarak berhenti terpanjang dari kecepatan trem saat itu adalah s_{brake} , sedangkan jarak berhenti terpendek dengan kecepatan maksimum yang mungkin di rel v_{max} menjadi jarak tambahan yaitu s_{guard} . Maka jarak *rail horizon* terdiri dari jarak berhenti terpanjang s_{brake} , jarak tambahan s_{guard} , serta jarak yang ditempuh saat memberi peringatan s_{alert} . Persamaan untuk menghitung jarak *rail horizon* diberikan oleh persamaan (III.1). Selain itu parameter yang digunakan seperti pada Tabel III.1.

$$s_H = s_{alert} + s_{brake} + s_{guard} \quad (III.1)$$

$$\text{Dimana } s_{alert} = t_{alert}v_t ; s_{brake} = \frac{v_t^2}{2a_{min}} ; s_{guard} = \frac{v_{max}^2}{2a_{max}}$$

Tabel III.1. Parameter estimator rel.

Parameter	Nilai
t_{alert}	2 detik
a_{min}	0.22 m/s^2
a_{max}	1.0 m/s^2

III.4.2. Prediktor Lintasan

Tugas yang dilakukan prediktor lintasan adalah memprediksi lintasan gerak dari tiap objek yang terdeteksi melalui data historis tiap objek. Metode prediksi lintasan yang digunakan adalah *kalman filter constant velocity* (KFCV). Dengan menggunakan metode tersebut, posisi dan kecepatan objek digambarkan dalam vektor *state* x_{kf} dimensi 4 yang terdiri dari posisi x_k , y_k , serta kecepatan dalam arah x $\dot{x}_k$ dan y $\dot{y}_k$, seperti pada persamaan (III.2) dan diinisialisasi dengan nilai 0. Hal ini karena pada inisialisasi posisi dan kecepatan objek belum diketahui. Selain itu ketidakpastian diinisialisasikan dengan nilai yang besar, agar bobot terhadap pengukuran lebih besar.

$$x_{kf} = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} \quad (III.2)$$

Untuk melakukan ekstrapolasi vektor *state*, matriks transisi dibangun dari model dinamika gerak dari posisi dan kecepatan. Model dinamika tersebut diimplementasikan dengan pendekatan diskrit menggunakan pencuplikan waktu Δt . Model dinamika tersebut ditunjukkan oleh persamaan (III.3). Dengan menggunakan model dinamika tersebut matriks transisi F ditunjukkan oleh persamaan (III.4).

$$\begin{aligned} x_{k+1} &= x_k + \dot{x}_k \Delta t \\ y_{k+1} &= y_k + \dot{y}_k \Delta t \\ \dot{x}_{k+1} &= \dot{x}_k \\ \dot{y}_{k+1} &= \dot{y}_k \end{aligned} \quad (III.3)$$

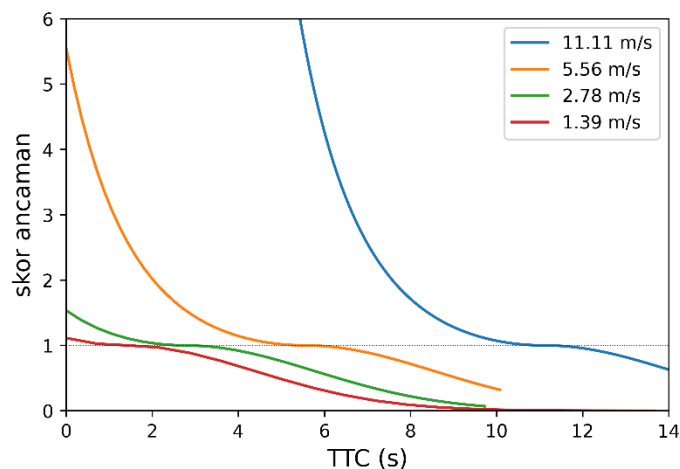
$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.4)$$

Pada prediksi lintasan gerak objek, data pengukuran yang digunakan adalah posisi x dan y dalam sistem koordinat global, sehingga matriks pengukuran H seperti yang ditunjukkan oleh persamaan (III.5).

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{III.5})$$

III.4.3. Penilaian Keselamatan

Tugas penilaian keselamatan adalah menghitung *time-to-collision* (TTC) dan *distance-to-collision* (DTC) dari tiap objek yang telah diprediksi lintasannya dan menentukan objek yang paling berbahaya. Ambang batas TTC yang digunakan adalah 4 detik sesuai studi yang telah dilakukan oleh Hirst dan Graham (1997) dan Hogema dan Janssen (1996). Studi mengenai ambang batas TTC telah dipaparkan pada bab II. 4.



Gambar III.6. Analisa skor ancaman pada variasi kecepatan trem.

Objek yang paling berbahaya ditentukan dengan menggunakan *threat energy function* yang diadaptasi dari penelitian yang dilakukan oleh Li dkk (2021). fungsi ini mempertimbangkan TTC dan waktu berhenti atau *time to stop* (TTS) untuk menghitung skor ancaman. TTS dihitung berdasarkan kecepatan trem dengan menggunakan perlambatan maksimum a_e yang dapat dilakukan. Gambar III.6

menunjukkan skor ancaman untuk kecepatan trem yang berbeda-beda dengan variasi TTC. Melalui gambar di ketahui bahwa skor ancaman 1 adalah keadaan saat TTC dan TTS sama, sedangkan skor ancaman > 1 menunjukkan kemungkinan tabrakan yang tidak dapat dihindari karena $TTC < TTS$ dan skor ancaman < 1 menunjukkan kemungkinan tabrakan yang masih dapat dihindari karena $TTC > TTS$.

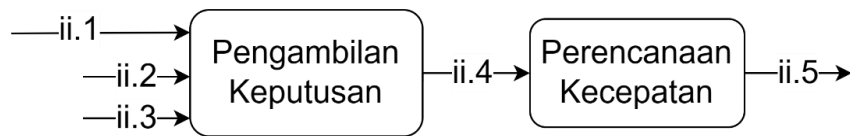
$$\varphi_{o_j}(\tau = D | TTC, TTS_{a_e}) = \begin{cases} \exp\left(\frac{\Delta t_e^2}{2\sigma^2}\right), & \Delta t_e < 0 \\ \exp\left(-\frac{\Delta t_e^2}{2\sigma^2}\right), & otherwise \end{cases} \quad (III.6)$$

$$\Delta t_e = TTC - TTS_{a_e}$$

$$\delta = \underset{j}{\operatorname{argmax}}(\varphi_{o_j}) \quad (III.7)$$

Skor ancaman untuk masing-masing objek yang terdeteksi φ_{o_j} diberikan oleh persamaan *theat energy function* yang ditunjukkan oleh persamaan (III.6). σ adalah ketidakpastian dari skor ancaman dan Δt_e adalah selisih antara TTC dan TTS_{a_e} . Kemudian, objek yang paling berbahaya ditentukan oleh skor ancaman terbesar seperti yang ditunjukkan persamaan (III.7).

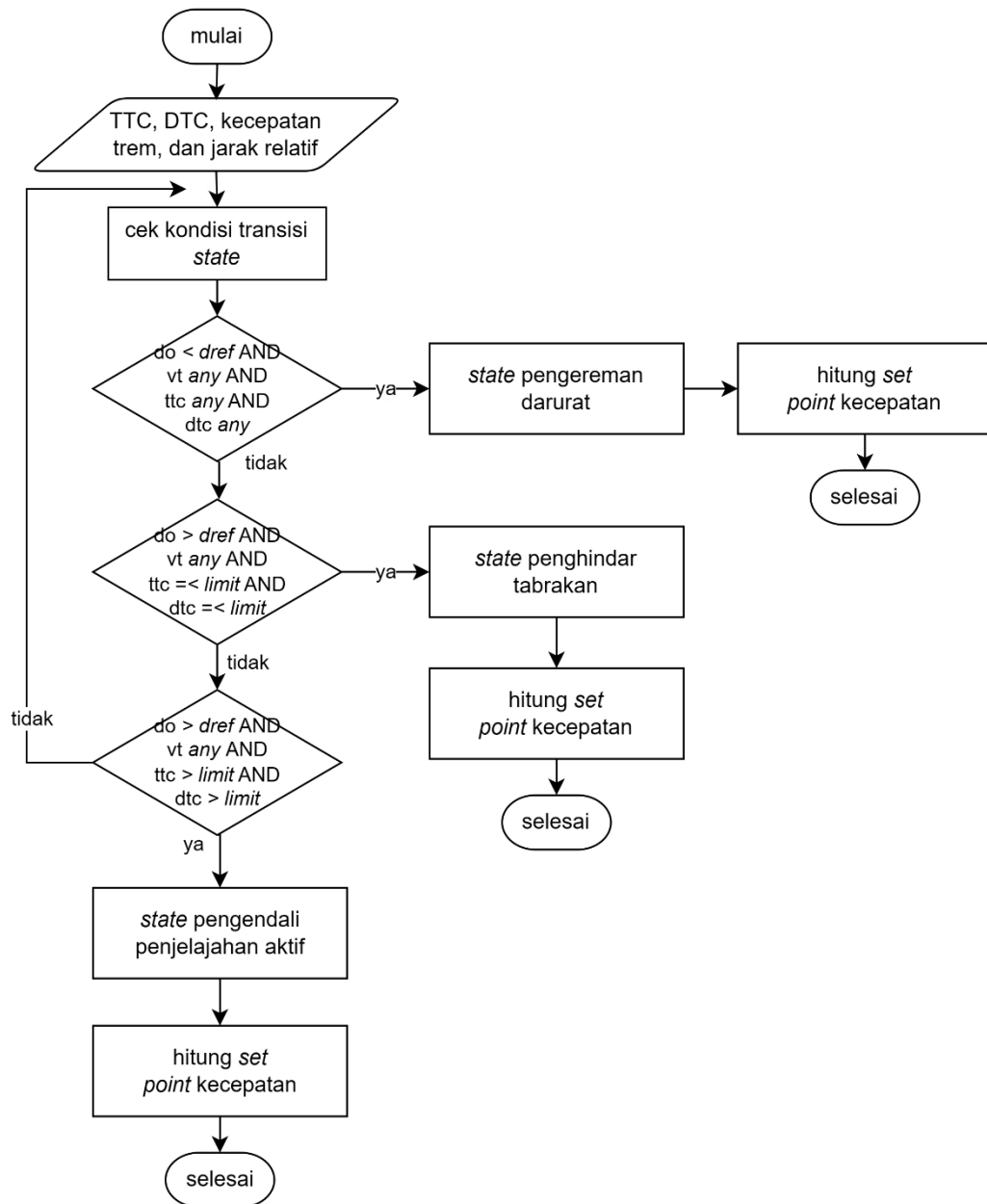
III.5. Subsistem Pengambilan Keputusan dan Perencanaan



Gambar III.7. Diagram blok subsistem pengambilan keputusan dan perencanaan.

Subsistem pengambilan keputusan dan perencanaan terdiri dari bagian pengambilan keputusan dan perencanaan kedepan seperti yang ditunjukkan pada Gambar III.7. Bagian pengambilan keputusan menerima masukan berupa TTC dan DTC (ii.1), jarak relatif trem lain atau pengguna jalan raya yang berada didepan dan

di rel (ii.2), dan kecepatan trem (ii.3). Keluarannya berupa aksi yang akan dijalankan (ii.4). Berdasarkan aksi yang dipilih, bagian perencanaan kecepatan akan menghitung *set point* kecepatan yang diperlukan (ii.5).



Gambar III.8. Diagram alir subsistem pengambilan keputusan dan perencanaan.

Diagram alir subsistem keputusan dan perencanaan seperti yang ditunjukkan pada Gambar III.8. Setelah menerima masukan berupa TTC, DTC, kecepatan trem v_t ,

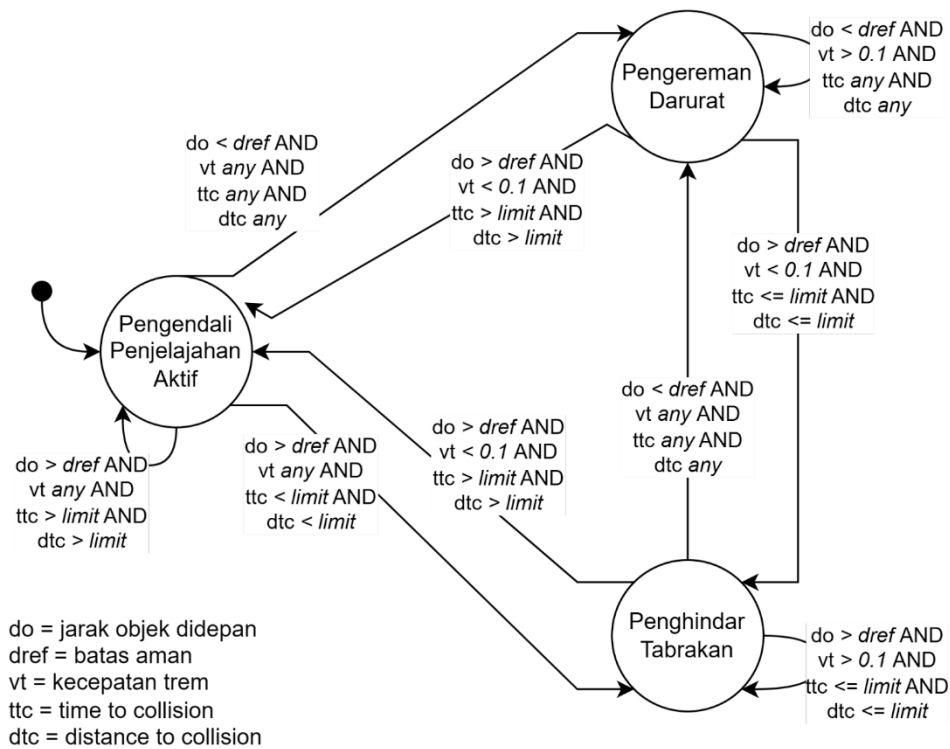
dan jarak relatif objek didepan terhadap trem d_o , keputusan aksi akan ditentukan berdasarkan data masukan. Kemudian *set point* kecepatan dihitung berdasarkan aksi yang dipilih.

III.5.1. Pengambilan Keputusan

Metode pengambilan keputusan yang digunakan adalah *finite state machine* (FSM). Metode ini telah digunakan oleh beberapa peneliti dalam “*DARPA urban challenge*” (Buehler dkk., 2009, Urmson dkk., 2008, Leonard dkk., 2008, dan Montemerlo dkk., 2008). Skema pembangkitan perilaku berbasis FSM intuitif dan efisien, meskipun memiliki beberapa kekurangan yaitu kurangnya ketahanan dalam menangani situasi yang tidak terduga dan kurangnya solusi/prilaku kreatif.

Pada penelitian ini metode FSM dipilih karena kendaraan otonom yang dilombakan pada “*DARPA urban challenge*” menggunakan metode FSM dan terbukti mampu mengambil keputusan terhadap kondisi yang telah dirancang. Selain itu, karakteristik kereta api yang dimiliki trem menyederhanakan masalah pengambilan keputusan karena gerak trem dibatasi oleh rel sehingga secara umum keputusan yang diambil berupa menaikkan kecepatan, menurunkan kecepatan, atau mempertahankan kecepatan.

Sesuai spesifikasi pada subbab III.1, FSM dirancang memiliki tiga *state*, yaitu pengendali penjelajahan aktif, pengereman darurat, dan penghindar tabrakan. Pengendali penjelajahan aktif adalah aksi yang diambil untuk menjelajah rel menggunakan kecepatan maksimum rel yang diperbolehkan, namun dapat secara adaptif menurunkan kecepatan apabila terdapat trem lain atau pengguna jalan raya yang berada di depan trem dan di rel. Pengereman darurat adalah aksi yang diambil ketika terdapat trem lain atau pengguna jalan raya yang berada sangat dekat didepan trem dan di rel yang tidak mampu diatasi oleh pengendali penjelajahan aktif dan penghindar tabrakan. Penghindar tabrakan adalah aksi yang diambil ketika terdapat pengguna jalan raya yang akan melintasi rel dan terdapat potensi tabrakan. Diagram FSM yang dirancang seperti yang ditunjukkan pada Gambar III.9.



Gambar III.9. Diagram FSM.

Kondisi transisi dari FSM ditentukan oleh jarak relatif objek didepan ke trem d_o , kecepatan trem v_t , TTC, dan DTC. *State* awal FSM adalah pengendali penjelajahan aktif. Pada situasi aman *state* akan berada di pengendali penjelajahan aktif. Situasi aman ditandai oleh jarak relatif objek didepan terhadap trem lebih dari batas aman d_{ref} , serta TTC dan DTC lebih dari batas tertentu. Apabila terdapat situasi darurat, *state* akan bertransisi ke *state* pengereman darurat. Situasi darurat ditandai oleh jarak relatif objek didepan terhadap trem kurang dari batas aman. Apabila terdapat potensi tabrakan yang disebabkan oleh pengguna jalan raya yang akan menyebrangi rel, *state* akan bertransisi ke *state* penghindar tabrakan. Potensi tabrakan ditandai oleh TTC dan DTC yang kurang dari batas tertentu.

III.5.2. Perencanaan Kecepatan

Masing-masing *state* FSM memiliki perencanaan kecepatan. Perencanaan kecepatan pada aksi pengendali penjelajahan aktif menggunakan metode *model*

predictive control (MPC), sedangkan aksi penghindar tabrakan menggunakan metode *fuzzy logic*. Untuk aksi pengereman darurat, *set point* kecepatan adalah nol.

III.5.2.1. *Model Predictive Control* Sebagai Metode Perencanaan Kecepatan

Model predictive control (MPC) digunakan untuk mengendalikan jarak dan kecepatan seperti yang dilakukan oleh Chen dkk (2017) dan Dani dkk (2017). Model *state-space* dipilih untuk menangkap dinamika proses sehingga dapat memprediksi keluaran selanjutnya. Model menggunakan *state* kecepatan trem dan jarak relatif antar kendaraan yang diimplementasikan dengan pendekatan diskrit. Model dinamika yang digunakan seperti yang ditunjukkan pada persamaan (III.8). v_k menyatakan kecepatan trem, a_k menyatakan sinyal kendali percepatan, dan Δt menyatakan perubahan waktu, sedangkan $d_{rel,k}$ dan $v_{rel,k}$ secara berturut-turut menyatakan jarak dan kecepatan relatif antara trem dan objek. Model dinamika tersebut kemudian menjadi model *state-space* seperti yang ditunjukkan pada persamaan (III.9).

$$\begin{aligned} v_{k+1} &= v_k + a_k \Delta t \\ d_{rel,k+1} &= d_{rel,k} - v_{rel,k} \Delta t - \frac{1}{2} a_k \Delta t^2 \end{aligned} \quad (III.8)$$

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} \\ B &= \begin{bmatrix} \Delta t \\ -0.5 \Delta t^2 \end{bmatrix} \end{aligned} \quad (III.9)$$

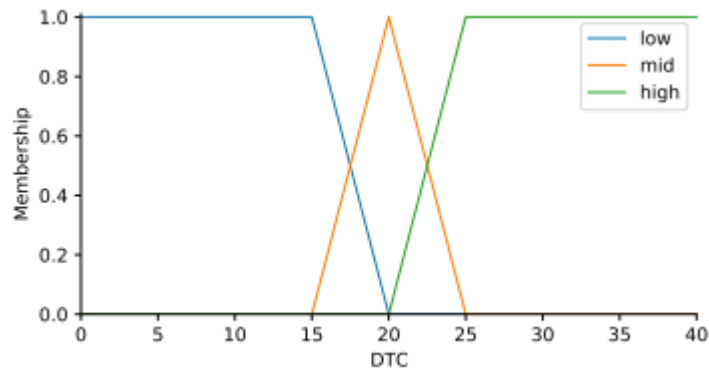
Kumpulan sinyal kendali selanjutnya dihitung dengan mengoptimalkan kriteria menggunakan fungsi biaya yang diadaptasi dari penelitian Gonzalez dkk (2020) agar sedekat mungkin dengan lintasan referensi. Fungsi biaya tersebut merupakan fungsi kuadratik dari galat antara sinyal keluaran yang diprediksi dan lintasan referensi. Lintasan referensi yang digunakan adalah ambang batas jarak aman d_{ref} dan kecepatan maksimum rel v_{max} . Batasan yang digunakan adalah percepatan dengan tujuan membatasi perencanaan kecepatan yang nyaman bagi penumpang. Persamaan fungsi biaya seperti yang ditunjukkan pada persamaan (III.10).

Pembobotan dalam fungsi biaya tersebut terdapat bobot jarak w_d , bobot kecepatan w_v , dan bobot percepatan w_a .

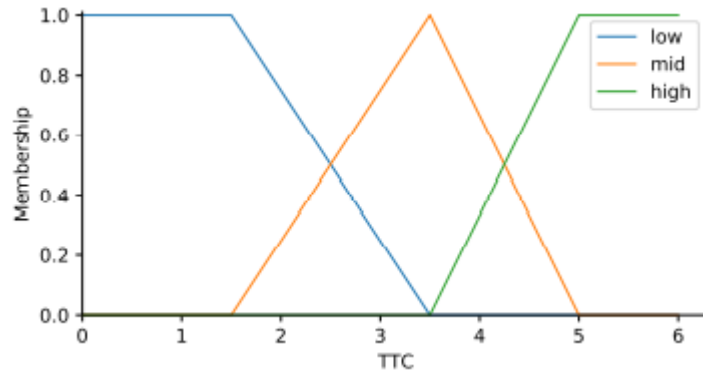
$$C = \sum_{k=0}^N \left(w_d (d_{rel,k+1} - d_{ref}) \right)^2 + \left(w_v (v_{rel,k+1} - v_{max}) \right)^2 + \left(w_a (a_{k+1}) \right)^2 \quad (III.10)$$

III.5.2.2. Logika Fuzzy Sebagai Metode Perencanaan Kecepatan

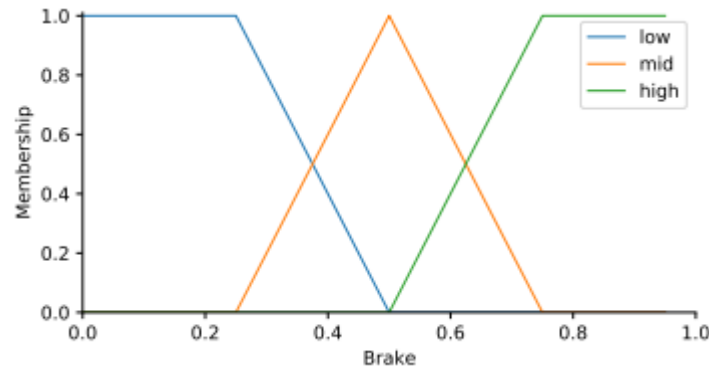
Fuzzy logic diadaptasi untuk menghasilkan penurunan kecepatan yang halus seperti pada penelitian yang dilakukan oleh Tomar dan Kumar (2021). Pada penelitian ini digunakan *fuzzy logic* metode Mamdani. Pada tahap fuzzifikasi terdapat tiga variabel fuzzy yang digunakan, yaitu DTC, TTC, dan pengereman. Variabel fuzzy tersebut dimodelkan dengan menggunakan fungsi keanggotaan trapesium dan segitiga. Fungsi keanggotaan untuk variabel DTC, TTC, dan tingkat pengereman seperti yang ditunjukkan pada Gambar III.10 – Gambar III.12.



Gambar III.10. Fungsi keanggotaan variabel DTC.



Gambar III.11. Fungsi keanggotaan variabel TTC.



Gambar III.12. Fungsi keanggotaan variabel tingkat pengereman.

Inferensi sistem fuzzy diperoleh dari kumpulan dan korelasi antar aturan. Inferensi yang digunakan adalah metode max, yaitu mengambil nilai maksimum aturan seperti yang ditunjukkan persamaan (III.11) dimana μ menyatakan derajat keanggotaan. Kemudian defuzzifikasi dilakukan dengan menggunakan metode Centroid. Solusi dari metode Centroid diperoleh dengan cara mengambil titik pusat daerah fuzzy. Hasilnya berupa tingkat pengereman yang ditunjukkan oleh persamaan (III.12).

$$\mu_{sf}(x_i) = \max(\mu_{kf}(x_i), \mu_{kf}(x_i)) \quad (\text{III.11})$$

$$brake_{CA} = \frac{\int \mu(z)zdz}{\int \mu(z)dz} \quad (\text{III.12})$$

Bab IV Pengujian dan Analisis

Perancangan sistem pengambilan keputusan diujikan pada simulator, kendaraan *testbed*, dan trem. Bab ini akan membahas tiga tahap pengujian yang dilakukan.

IV.1. Simulasi

Simulasi sistem pengambilan keputusan dilakukan di simulator Carla. Pada Carla, kendaraan pemadam api digunakan sebagai kendaraan ego yang mewakili trem. kendaraan tersebut diatur untuk berjalan pada satu jalur untuk mensimulasikan trem yang berjalan dari satu stasiun ke stasiun lain. Berat kendaraan pemadam api diubah sesuai berat trem pada Tabel II.1, yaitu 48 ton. Untuk keperluan simulasi, digunakan kendali PID yang telah tersedia dalam simulator untuk menjejak *set point* kecepatan yang diberikan. Lingkungan simulasi seperti yang ditunjukkan pada Gambar IV.1.



Gambar IV.1. Lingkungan pengujian di simulator.

Lingkungan simulasi yang dipilih adalah Town02 dari simulator Carla. Seperti terlihat pada Gambar IV.1, lingkungan simulasi terdiri dari pepohonan, bangunan,

jalur lurus, tikungan, persimpangan, dan kendaraan-kendaraan seperti mobil, sepeda, dan motor. Rel yang akan dilalui ditunjukkan oleh titik-titik hitam pada gambar.

Pengujian dilakukan untuk tiga macam situasi yang ditargetkan, yaitu situasi aman, situasi terdapat pengguna jalan raya di rel, dan situasi terdapat pengguna jalan raya yang menyebrang. Selain itu, terdapat pengujian terhadap situasi darurat, yaitu situasi yang muncul sebagai implikasi apabila keselamatan trem terancam ketika perencanaan kecepatan untuk situasi lainnya belum mampu menyelamatkan trem.

IV.1.1. Simulasi lalu lintas sederhana

Pada pengujian pertama, tiap *state* FSM diuji menggunakan situasi terkait yang dapat memicu transisi ke *state* yang diuji. Situasi yang disimulasikan seperti yang ditunjukkan pada Gambar IV.2. Pada gambar tersebut trem diwakili oleh mobil pemadam kebakaran berwarna merah, sedangkan pengguna jalan raya yang menjadi halangan adalah mobil. Garis putus-putus berwarna merah pada tiap subgambar adalah *rail horizon* dari *waypoint* yang akan dilalui, sedangkan yang berwarna magenta adalah prediksi lintasan dari mobil yang terdeteksi. Subgambar (a) menggambarkan situasi aman, yaitu mobil yang terdeteksi tidak berada di rel dan tidak akan menyebrangi rel. Lalu subgambar (b) menggambarkan situasi dimana terdapat mobil yang berada di rel. Kemudian subgambar (c) menggambarkan situasi dimana terdapat mobil yang akan melintasi rel.

State pertama dari *state* FSM yang dirancang adalah pengendali penjelajahan aktif. *State* pengendali penjelajahan aktif terpicu untuk situasi aman dan situasi terdapat pengguna jalan raya di rel. Ketika berada di *state* ini, perencanaan kecepatan yang menggunakan metode MPC secara adaptif menaikkan *set point* kecepatan trem hingga kecepatan maksimum rel yang diperbolehkan atau menurunkan *set point* kecepatan. Perencanaan kecepatan yang menaikkan *set point* kecepatan trem hingga kecepatan maksimum rel terjadi saat situasi aman, sedangkan menurunkan *set point* kecepatan terjadi apabila menemui mobil yang berada di rel dengan target jarak 15 meter. Perencanaan kecepatan tersebut dihitung oleh metode MPC dengan bobot kecepatan trem dan jarak relatif mobil yang telah dirancang pada bab III.



(a)



(b)



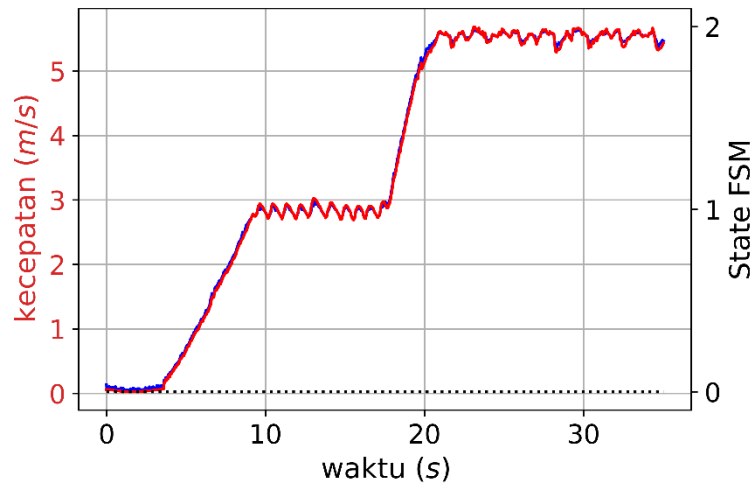
(c)

Gambar IV.2. Situasi yang disimulasikan di Carla. (a) situasi aman; (b) situasi terdapat pengguna jalan raya di rel; (c) situasi terdapat pengguna jalan raya melintasi rel.

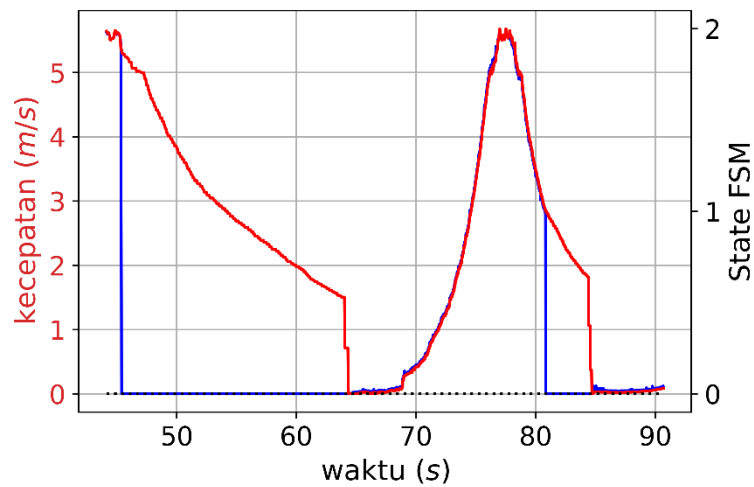
Pada simulasi situasi aman yang memicu *state* pengendali penjelajahan aktif, terdapat dua kecepatan maksimum rel, yaitu 2.78 m/s dan 5.56 m/s. Maka MPC membangun profil kecepatan untuk berada di kecepatan maksimum rel yang sesuai. Hasil profil kecepatan dari perencanaan kecepatan saat situasi aman seperti yang ditunjukkan pada Gambar IV.3. Pada gambar tersebut *set point* kecepatan yang dihitung menggunakan metode MPC ditunjukkan oleh grafik biru.

Transisi *state* FSM ke *state* pengendali penjelajahan aktif juga terjadi ketika menemui situasi dimana terdapat mobil di rel. Skenario simulasi yaitu trem yang sedang berjalan pada kecepatan 5.56 m/s mendeteksi adanya mobil pada jarak 100 m yang bergerak dengan kecepatan 2.78 m/s. Semakin lama, jarak relatif mobil terhadap trem semakin kecil. Hasilnya *set point* kecepatan trem diturunkan ketika jarak relatif mobil berada di bawah 50 m dan dinaikkan ketika jarak relatif tersebut membesar. Hasil profil kecepatan dari simulasi tersebut seperti yang ditunjukkan

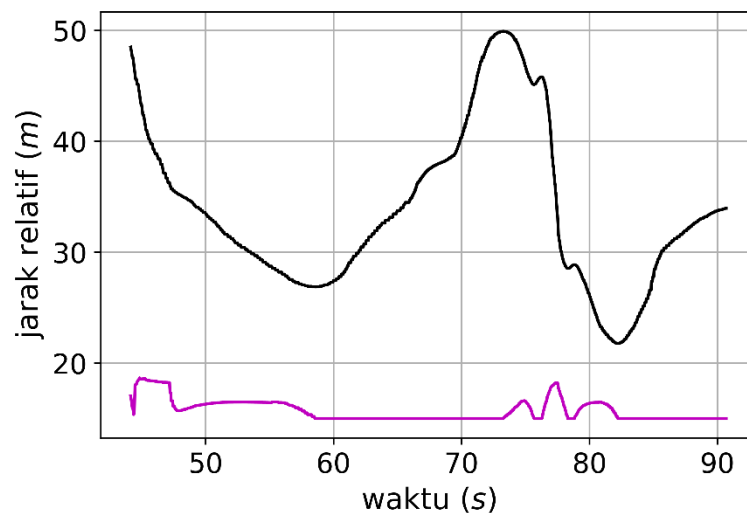
pada Gambar IV.4 dan profil jarak relatif seperti yang ditunjukkan pada Gambar IV.5.



Gambar IV.3. Profil kecepatan situasi aman. (*merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM*).



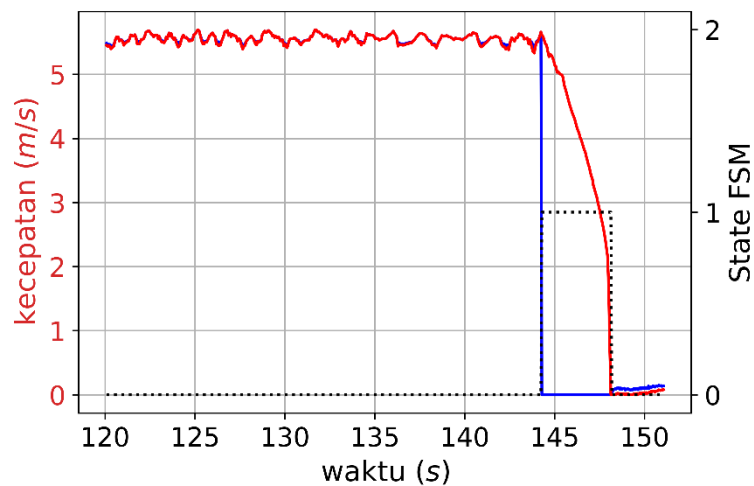
Gambar IV.4. Profil kecepatan situasi terdapat pengguna jalan raya di rel. (*merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM*).



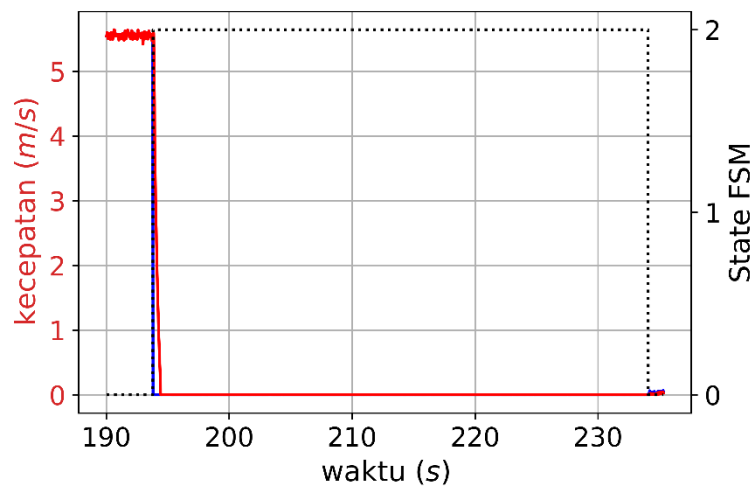
Gambar IV.5. Profil jarak relatif pengguna jalan raya yang berada di rel terhadap trem. (hitam: jarak relatif, magenta: jarak batas aman).

State FSM berikutnya adalah penghindar tabrakan. *State* ini dirancang untuk situasi dimana terdapat mobil yang akan menyebrangi rel. Simulasi dilakukan pada persimpangan town02 dengan mengatur agar mobil menyebrang ketika trem mendekati persimpangan seperti yang digambarkan pada Gambar IV.2 c. Skenario simulasi yaitu pada saat trem berjalan dengan kecepatan 5.56 m/s dan mendekati persimpangan, terdapat mobil yang bergerak ke arah rel dan berbelok, sehingga memotong rel trem. Pada situasi tersebut trem dan mobil akan bertabrakan apabila trem mempertahankan kecepatannya. Melalui pengamatan simulasi, trem dapat menghindari tabrakan dengan menurunkan kecepatannya seperti yang ditunjukkan pada Gambar IV.6. Gambar tersebut menunjukkan hasil profil kecepatan trem saat berada di *state* penghindar tabrakan.

State FSM pengereman darurat terpicu ketika trem bertemu situasi dimana terdapat mobil di rel pada jarak kurang dari jarak aman yang dirancang pada bab III. Simulasi dilakukan dengan memunculkan mobil di depan trem secara tiba-tiba pada saat trem berjalan dengan kecepatan 5.56 m/s. Hasil simulasi yaitu perencanaan kecepatan untuk *state* pengereman darurat memberikan *set point* nol seperti yang ditunjukkan pada Gambar IV.7.

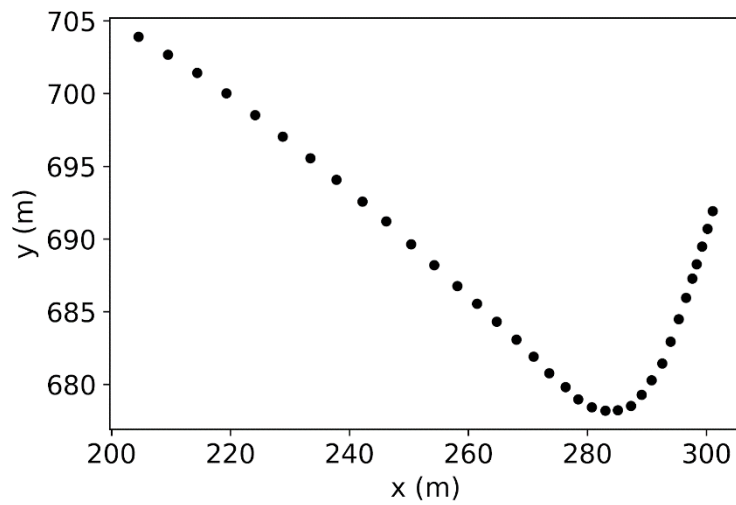


Gambar IV.6. Profil kecepatan situasi terdapat pengguna jalan raya melintasi rel. (merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM).

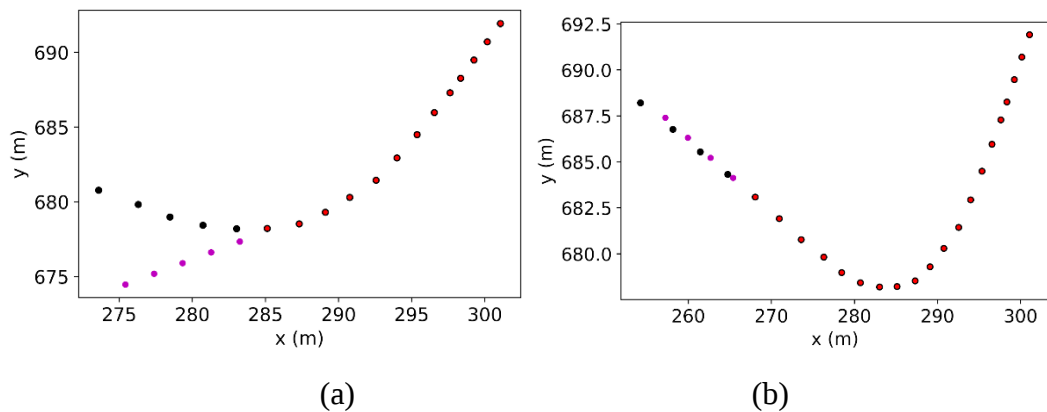


Gambar IV.7. Profil kecepatan situasi darurat. (merah: kecepatan trem, biru: set point kecepatan, dan hitam: state FSM)

Pada simulasi lalu lintas sederhana juga diamati hasil prediksi lintasan objek. Salah satu lintasan objek yang menjadi *ground truth* seperti yang ditunjukkan pada Gambar IV.8. yaitu objek yang berbelok dipersimpangan. Prediksi lintasan untuk objek tersebut diamati pada dua keadaan, yaitu ketika objek akan berbelok dan setelah berbelok. Hasil prediksi seperti yang ditunjukkan pada Gambar IV.9.



Gambar IV.8. *ground truth* posisi objek.



Gambar IV.9. Prediksi lintasan dibandingkan *ground truth*. (a) objek yang akan berbelok; (b) objek yang bergerak lurus. (*hitam: ground truth, merah: pengukuran, magenta: prediksi lintasan*).

Pada Gambar IV.9a terlihat bahwa prediksi lintasan ketika objek akan berbelok jauh dari *ground truth*. *Root mean square deviation* (RMSD) untuk posisi x dan y dari objek yang akan berbelok adalah 11.570 dan 21.751. Sedangkan objek yang bergerak lurus dapat diprediksi lintasannya dengan lebih baik seperti yang ditunjukkan pada gambar IV.9b dengan RMSD posisi x dan y adalah 3.527 dan 0.249. Ketidakmampuan sistem dalam memprediksi lintasan dari objek yang akan berbelok atau bermanuver merupakan batasan dari metode KFCV karena

menggunakan asumsi kecepatan konstan. Data prediksi lintasan, *ground truth* dan RMSD dari objek yang akan berbelok dan bergerak lurus seperti yang ditunjukkan pada Tabel IV.1.

Tabel IV.1. RMSD prediksi lintasan terhadap *ground truth*.

objek yang akan berbelok			RMSD
x	gt	280.721, 278.466, 276.307, 273.593, 270.96	11.570
	p	283.244, 281.291, 279.336, 277.382, 275.428	
y	gt	678.432, 678.978, 679.822, 680.77, 681.909	21.751
	p	677.333, 676.617, 675.900, 675.184, 674.467	
objek yang bergerak lurus			RMSD
x	gt	264.743, 261.442, 258.137, 254.251	3.527
	p	265.399, 262.675, 259.952, 257.228	
y	gt	684.316, 685.543, 686.771, 688.207	0.249
	p	684.128, 685.220, 686.311, 687.403	

IV.1.2. Simulasi lalu lintas campuran

Simulasi lalu lintas campuran dilakukan dengan memunculkan banyak pengguna jalan raya, yaitu 40 kendaraan (mobil, motor, truck, bus, sepeda) dan 40 pejalan kaki. Pengguna jalan raya tersebut digerakkan dan diatur perilakunya oleh modul pengatur lalu lintas yang terdapat dalam simulator Carla. Pengguna jalan raya dapat berjalan mengelilingi kota sesuai batas kecepatan tiap jalan, menyebrangi jalan, berhenti, dan negosiasi di persimpangan. Pada simulasi ini, modul pengatur lalu lintas secara mandiri mengatur perilaku pengguna jalan raya sehingga tiap kejadian muncul secara acak. Situasi-situasi yang muncul dalam simulasi sebagai berikut:

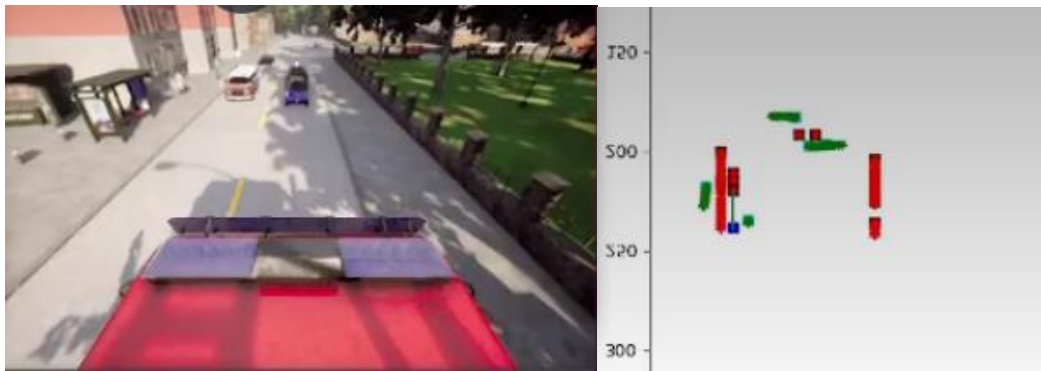
1. Kendaraan (mobil, motor, truck, bus, dan sepeda) berjalan di rel dengan kecepatan yang lebih rendah atau tinggi dari trem,
2. Kendaraan (mobil, motor, truck, bus, dan sepeda) berbelok di persimpangan dan mengambil jalan di rel,
3. Kendaraan (mobil, motor, truck, bus, dan sepeda) menyebrangi rel,

4. Kendaraan (mobil, motor, truck, bus, dan sepeda) berhenti secara tiba-tiba,
5. Kemacetan di persimpangan,
6. Pejalan kaki menyebrangi rel.

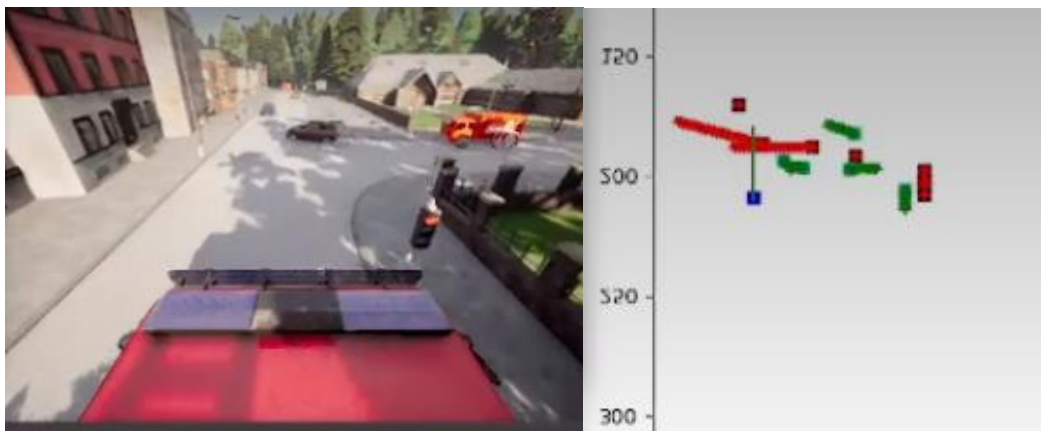
Lintasan yang akan dilalui trem sama seperti simulasi situasi sederhana, namun trem tidak akan berhenti di titik selesai melainkan melanjutkan perjalanannya hingga kembali ke titik mulai. Simulasi dimulai dari keadaan trem diam di titik mulai dimana *state* awal adalah pengendali penjelajahan aktif. Pada tiap iterasi program, *rail horizon* dihitung sesuai kecepatan trem. Panjang *rail horizon* akan menentukan apakah suatu objek (jika ada) disekitar trem perlu diperhatikan atau tidak. Apabila tidak terdapat objek dalam area *rail horizon* sehingga memenuhi kondisi untuk transisi diri sesuai rancangan FSM pada Gambar III.9. maka perencanaan kecepatan akan secara adaptif mengikuti kecepatan maksimum rel (5.56 m/s).

Pada tiap iterasi, sistem pengambilan keputusan menerima data posisi dan kecepatan dari Python API yang disediakan simulator Carla. Data yang diterima dibatasi pada jarak 100 m untuk menjaga pengenalan terhadap lingkungan simulasi sesuai dengan pengenalan lingkungan yang didukung oleh sensor. Setidaknya terdapat 11 objek yang terdeteksi saat trem mendekati persimpangan. Semua objek tersebut dapat diprediksi lintasanya sehingga trem memiliki pengetahuan akan arah gerak dari objek-objek disekitar. Plot posisi objek dan posisi prediksi lintasanannya seperti yang ditunjukkan oleh Gambar IV.10. Gambar kiri menunjukkan situasi lalu lintas di simulator, sedangkan gambar kanan adalah plot posisi dan prediksi lintasan objek (merah untuk kendaraan dan hijau untuk pejalan kaki). Selain itu posisi trem dan daerah *rail horizon* ditunjukkan di gambar kanan oleh titik berwarna biru dan *rail horizon* ditunjukkan oleh garis hijau. Subgambar (a) menunjukkan sistem memprediksi lintasan dari 11 objek (mobil dan pejalan kaki) disekitar dimana objek yang menimbulkan bahaya bagi trem adalah mobil yang berhenti di rel, sedangkan subgambar (b) menunjukkan sistem memprediksi lintasan dari 11 objek yang terdeteksi dimana objek yang menimbulkan bahaya adalah mobil yang menyebrang.

Analisa simulasi dalam lalu lintas campuran menggunakan tiga pendekatan, yaitu jumlah keberhasilan simulasi dengan tetap menjaga keselamatan, jumlah kemunculan *state* FSM yang dapat menjaga keselamatan trem, dan pemetaan *state* FSM yang dijalankan terhadap ekspektasi *state* yang muncul untuk suatu situasi. Masing-masing pendekatan dijelaskan sebagai berikut.



(a)



(b)

Gambar IV.10. (a) trem menemui mobil yang diam di rel; (b) trem menemui objek yang menyebrang.

Pada kondisi terdapat banyak objek yang terdeteksi seperti pada Gambar IV.10, sistem pengambilan keputusan dapat bertransisi dari satu *state* ke *state* lainnya sehingga trem dapat menghindari tabrakan dengan objek-objek lainnya. Sebanyak

50 kali simulasi dilakukan untuk menguji tingkat keberhasilan algoritma sistem pengambilan keputusan untuk menjalankan trem secara otonom. Diketahui sebanyak 10 simulasi trem gagal menjaga keselamatannya. Hal ini membawa tingkat keberhasilan simulasi sistem pengambilan keputusan terhadap keselamatan sebesar 80%. Persentase ini kurang baik dalam sistem otonom karena masih terdapat risiko keselamatan trem.

Lebih khusus, keberhasilan simulasi sistem pengambilan keputusan ditentukan oleh *state* apa yang diambil pada suatu situasi yang muncul. Untuk itu, pada tiap simulasi dilakukan pengamatan pada *state* FSM apa saja yang dijalankan dan apakah pemilihan *state* tersebut dapat menjaga keselamatan trem. *State* FSM yang diambil terhadap keberhasilan menjaga keselamatan trem ditunjukkan oleh Tabel IV.2. Tabel tersebut menunjukkan dalam 50 kali simulasi yang dilakukan, terdapat 458 kali *state* FSM berada di *state* pengendali penjelajahan aktif. Keputusan yang diambil tersebut tepat sebanyak 444 kali karena berhasil tetap menjaga keselamatan trem, namun 14 kali menyebabkan tabrakan. Selain itu, *state* FSM berada di *state* penghindar tabrakan dan *state* pengereman darurat, masing-masing sebanyak 411 kali dan 23 kali. Keputusan untuk berada di *state* penghindar tabrakan dan pengereman darurat berhasil menjaga keselamatan trem. Persentase capaian keselamatan saat berada di *state* pengendali penjelajahan aktif adalah 96.94%, sedangkan capaian keselamatan *state* penghindar tabrakan dan *state* pengereman darurat adalah 100%. Sehingga diketahui bahwa tabrakan terjadi ketika sistem berada di *state* pengendali penjelajahan aktif.

Tabel IV.2. Hasil simulasi pada lalu lintas campuran.

<i>State</i> FSM	Tidak Menabrak	Menabrak	Capaian keselamatan
Pengendali penjelajahan aktif	444	14	96.94%
Penghindar tabrakan	411	0	100%
Pengereman darurat	23	0	100%

Pada simulasi, kemunculan tiap situasi dapat dengan mudah diamati dan dapat diekspektasikan aksi mana yang seharusnya diambil. Ekspektasi aksi dapat digunakan sebagai kebenaran dasar untuk menentukan apakah aksi yang dipilih oleh sistem tepat. Ketika disekitar trem tidak terdapat objek atau objek-objek tersebut tidak berada di rel atau objek-objek tersebut tidak menyebrang maka ekspektasi aksi yang diambil sistem adalah pengendali penjelajahan aktif. Apabila terdapat objek di rel maka ekspektasi aksi yang diambil adalah pengendali penjelajahan aktif atau pengereman darurat, sedangkan ekspektasi aksi yang diambil ketika terdapat objek yang akan menyebrang adalah penghindar tabrakan. Tabel IV.3 menunjukkan pemetaan *state* FSM yang dijalankan terhadap ekspektasi *state* yang muncul untuk suatu situasi. Keputusan yang diambil sesuai dengan ekspektasi *state* 444 kali, sedangkan 198 kali bertransisi ke *state* penghindar tabrakan. Transisi yang dilakukan ini dinilai tidak diperlukan karena mobil, motor, sepeda, dan pejalan kaki yang menyebabkan transisi sejatinya tidak membahayakan keselamatan trem. Kondisi transisi ke *state* penghindaran tabrakan terpenuhi karena terdapat prediksi lintasan dari objek-objek tersebut yang memotong rel. Meskipun 198 kali keputusan transisi tersebut tidak sesuai dengan ekspektasi aksi, keputusan tersebut tidak membahayakan keselamatan trem.

Tabel IV.3. Perbandingan *state* FSM yang terjadi dan target *state* FSM.

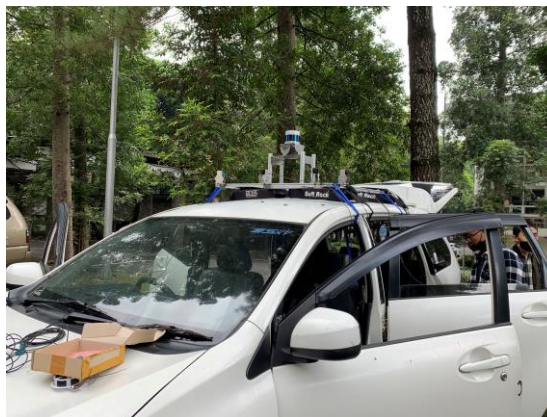
State FSM yang terjadi	Target state berdasarkan situasi		
	Pengendali penjelajahan aktif	Penghindar tabrakan	Pengereman darurat
Pengendali penjelajahan aktif	444	11	3
Penghindar tabrakan	198	213	0
Pengereman darurat	0	0	23

Ekspektasi aksi untuk situasi terdapat pengguna jalan raya melintas atau berbelok di persimpangan dan mengambil jalan di rel adalah berada di *state* penghindar tabrakan. Keputusan yang diambil sesuai dengan ekspektasi sebanyak 213 kali dan mampu menjaga keselamatan trem. Namun pada situasi tersebut juga terjadi 11 kali kesalahan dalam pengambilan keputusan, yaitu *state* FSM bertransisi ke pengendali penjelajahan aktif. Hal ini menyebabkan terjadi tabrakan antara trem dan mobil yang akan melintas. Pengamatan simulasi menunjukkan, kesalahan tersebut terjadi ketika mobil akan melintas pada saat trem dalam keadaan diam.

Ekspektasi aksi untuk situasi darurat adalah berada di *state* pengereman darurat. Keputusan yang diambil sesuai dengan target sebanyak 23 kali, sedangkan 3 lainnya bertransisi ke *state* pengendali penjelajahan aktif. Pengamatan simulasi menunjukkan 3 kali kesalahan tersebut terjadi ketika posisi mobil melintang di rel. Kesalahan ini mengakibatkan terjadinya tabrakan antara trem dengan mobil.

Keputusan yang diambil oleh sistem bergantung pada kondisi yang didefinisikan untuk tiap *state*. Dari pengamatan simulasi diketahui bahwa sistem pengambilan keputusan mampu bekerja pada situasi-situasi yang muncul dalam simulasi. Kecuali situasi dimana terdapat mobil yang melintang di rel dan mobil yang menyebrang pada saat kecepatan trem nol.

IV.2. Pengujian *Testbed*



Gambar IV.11. Kendaraan *testbed*.

Pengujian di kendaraan *testbed* dilakukan dengan mengimplementasikan algoritma sistem pengambilan keputusan ke NVIDIA Drive AGX Pegasus. Sistem ujung-ke-ujung yang dijalankan adalah sistem persepsi untuk mengolah stream langsung dari satu kamera sekonix dan sistem pengambilan keputusan. Data stream langsung dari kamera sekonix diolah oleh sistem persepsi untuk melakukan deteksi dan klasifikasi kelas objek. Objek yang dimaksud adalah pengguna jalan raya. Kendaraan *testbed* yang digunakan seperti yang ditunjukkan pada Gambar IV.11. Pengujian dilakukan di jalan dalam kampus ITB dengan mengatur lingkungan tertutup yang hanya terdapat satu halangan.

Hasil pengolahan data yang diterima oleh sistem pengambilan keputusan terbatas pada posisi objek dalam kerangka lokal sensor. Kemudian belum dapat dikirimnya data kecepatan objek dan dibedakan identitas dari tiap objek menyebabkan skenario pengujian terbatas pada satu objek. Pada pengujian ini objek yang digunakan adalah satu orang pejalan kaki yang menjalankan skenario sebagai berikut:

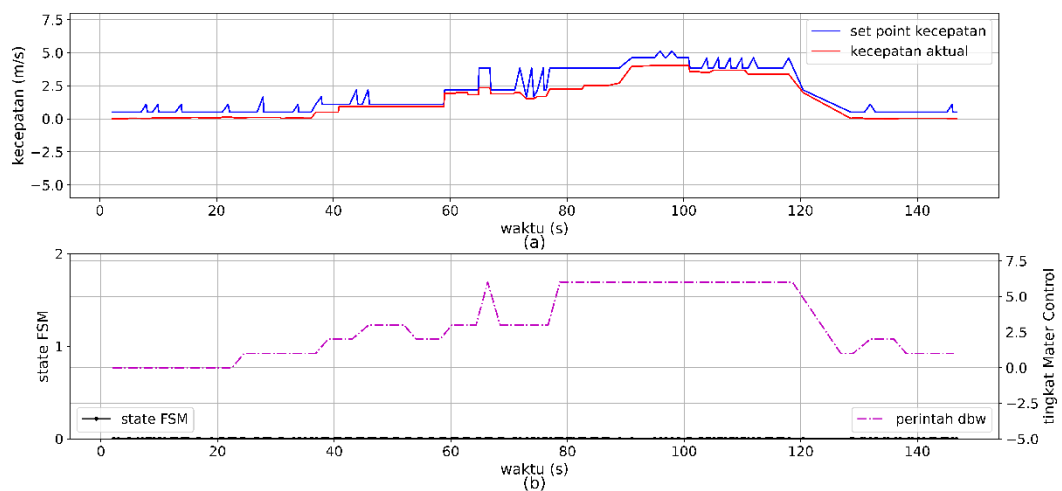
1. Skenario pertama yaitu pejalan kaki tidak berada dalam radius deteksi sensor kamera selama pengujian dijalankan,
2. Skenario kedua yaitu pejalan kaki diam di rel pada jarak 20 meter dari kendaraan *testbed*.
3. Skenario ketiga yaitu pejalan kaki diam di pinggir rel pada jarak 20 meter dari kendaraan *testbed*.

Pengujian di kendaraan *testbed* dilakukan dengan tanpa memberikan sinyal kendali pada kendaraan. Namun, kecepatan kendaraan *testbed* dinaikkan secara manual menggunakan kemudi sesuai perintah dari sistem pengambilan keputusan. Hasil pengujian untuk skenario pertama, kedua, dan ketiga secara berturut-turut ditunjukkan pada Gambar IV.12, Gambar IV.13, dan Gambar IV.14.

Pada Gambar IV.12, data pengujian diambil untuk skenario pertama yaitu tidak terdapat pejalan kaki dalam radius deteksi sensor kamera. Skenario ini mensimulasikan situasi aman. Hasilnya *state* FSM berada di *state* pengendali penjelajahan aktif yang dapat dilihat pada subgambar (b) yaitu garis hitam dengan

nilai vertikal *state* FSM 0. Pada sekitar 22 detik pertama pengujian, sistem otonom belum diaktifkan sehingga perintah DbW yang diberikan 0, yaitu netral. Perintah DbW digambarkan oleh garis putus-putus berwarna magenta pada subgambar (b). Setelah sistem otonom diaktifkan, perintah DbW berubah menjadi 1, yaitu P1 dan kemudian dinaikkan secara bertahap hingga P6. Pada pengujian ini, kecepatan maksimum rel adalah 4.72 m/s. Profil kecepatan ditunjukkan pada subgambar (a) dimana garis berwarna biru adalah *set point* kecepatan dan garis berwarna merah adalah kecepatan kendaraan *testbed* yang terukur.

Gambar IV.13 merupakan hasil pengujian untuk skenario kedua, yaitu terdapat pejalan kaki yang diam di rel. Gambar tersebut terdiri dari 4 subgambar yaitu subgambar (a) berisi profil kecepatan dari pengujian yang dilakukan, subgambar (b) berisi *state* FSM yang aktif dan perintah DbW yang diberikan, subgambar (c) berisi profil jarak relatif dari pejalan kaki yang terdeteksi, dan subgambar (d) berisi profil TTC dan DTC yang disebabkan oleh pejalan kaki yang terdeteksi.



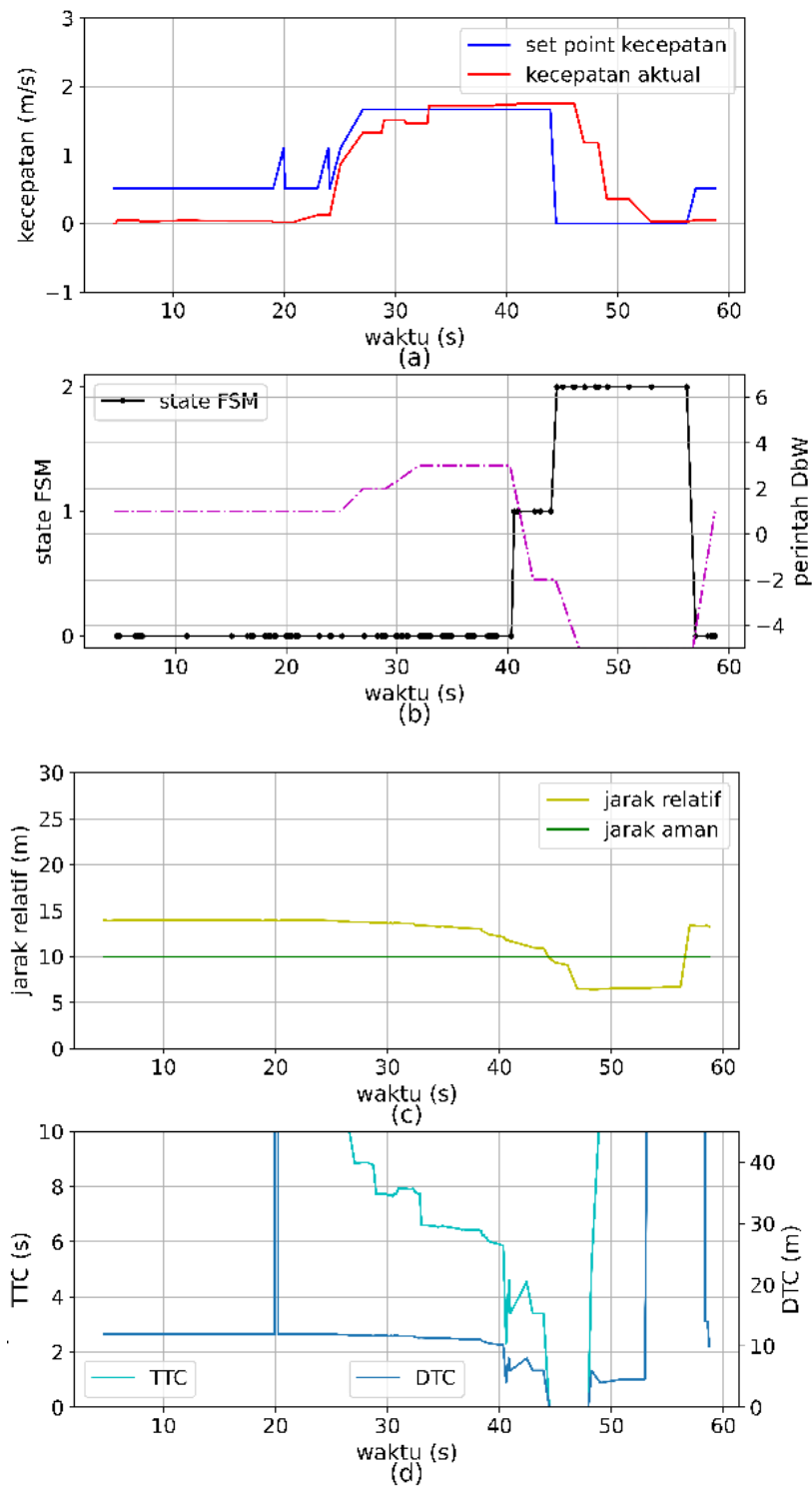
Gambar IV.12. Pengujian skenario pertama. (a) profil kecepatan; (b) state FSM dan perintah ke DbW. (biru: *set point* kecepatan, merah: *kecepatan aktual*, hitam: *state* FSM, dan magenta: *perintah* ke DbW).

Pengujian skenario kedua berjalan sebagai berikut. Pada keadaan awal, pejalan kaki terdeteksi berada di jarak 14 meter, *state* FSM berada di pengendali penjelajahan

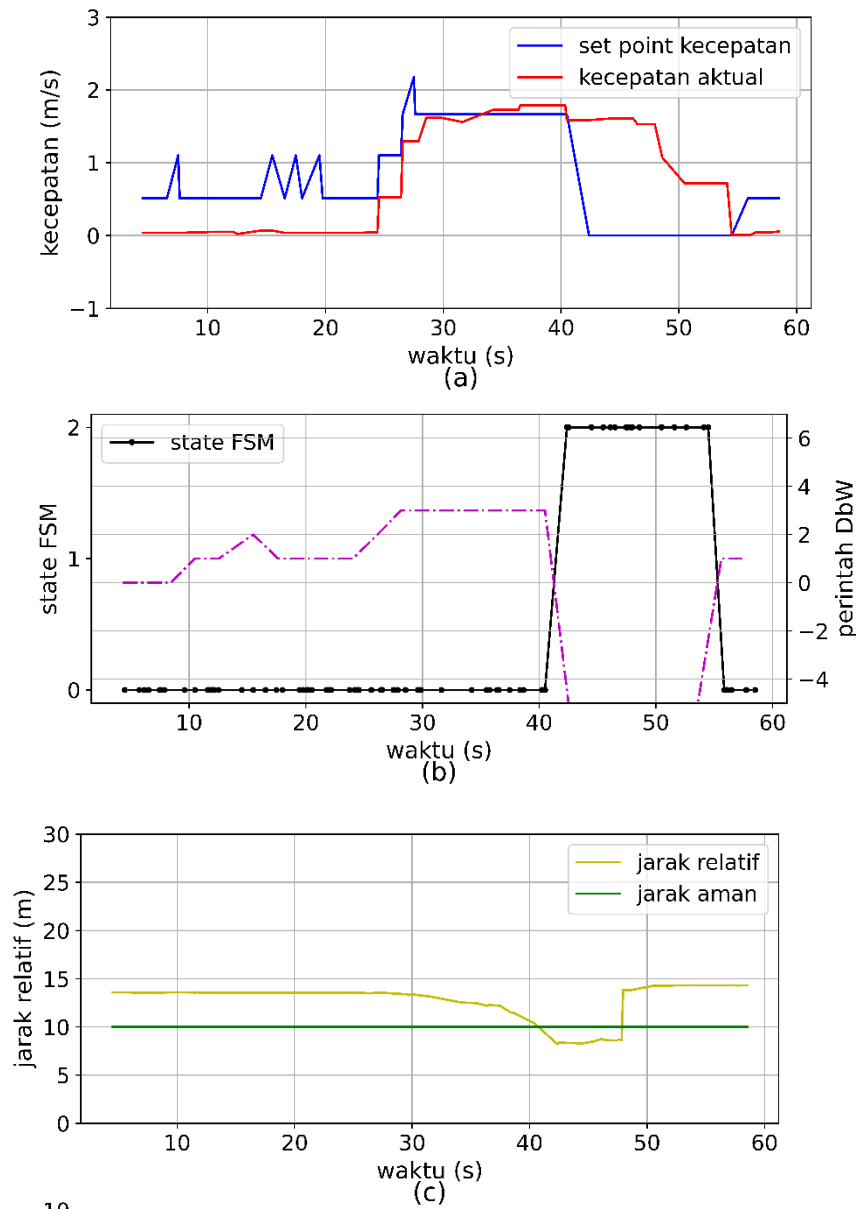
aktif. Maka pada 40 detik pertama pengujian, perintah DbW adalah memberikan P1 hingga P3 (subgambar (b)). Pada saat kendaraan *testbed* berjalan, jarak relatif pejalan kaki terhadap kendaraan *testbed* semakin kecil yaitu ditunjukkan oleh garis kuning di subgambar (c). Bersamaan dengan semakin kecilnya jarak pejalan kaki, pejalan kaki diprediksi bergerak ke arah kendaraan *testbed* dan menyebabkan tabrakan. Profil TTC dan DTC pada subgambar (d) menunjukkan bahwa nilai TTC dan DTC semakin kecil, hingga memenuhi kondisi transisi ke *state* penghindar tabrakan (ditandai dengan *state* FSM 1 pada subgambar (b)). Pada waktu ini, Kendaraan *testbed* akan mencapai titik tabrakan pada TTC 2.3 detik dan DTC 1 meter.

Pada saat berada di *state* penghindar tabrakan, perintah DbW adalah memberikan pengereman B2 untuk menurunkan kecepatan. Kemudian karena jarak relatif pejalan kaki terhadap kendaraan *testbed* semakin kecil, hingga kurang dari batas aman yaitu ditunjukkan pada subgambar (c) ketika jarak relatif (garis kuning) berada dibawah batas aman (garis hijau), *state* FSM bertransisi ke pengereman darurat. Pada waktu ini terlihat grafik *state* FSM dalam subgambar (b) bernilai 2, yaitu *state* pengereman darurat.

Pada pengujian skenario ketiga, pejalan kaki diam di pinggir rel pada jarak 14 meter dari kendaraan *testbed*. Seperti pengujian di dua skenario sebelumnya, *state* FSM berada di pengendali penjelajahan aktif yang memberikan perintah DbW untuk menaikkan kecepatan. Hal ini dapat dilihat pada Gambar IV.14 b. Seiring dengan meningkatnya kecepatan kendaraan *testbed*, jarak relatif objek terhadap kendaraan *testbed* mengecil hingga berada dibawah batas aman (Gambar IV.14 c). Pada waktu ini *state* FSM bertransisi ke pengereman darurat. Profil kecepatan dari pengujian ini ditunjukkan pada Gambar IV.14 a.



Gambar IV.13. Pengujian skenario kedua. (a) profil kecepatan; (b) state FSM dan perintah ke DbW; (c) jarak relatif pengguna jalan raya terhadap trem; (d) TTC dan DTC.



Gambar IV.14. Pengujian skenario ketiga. (a) profil kecepatan; (b) state FSM dan perintah ke DbW; (c) jarak relatif pengguna jalan raya terhadap trem.

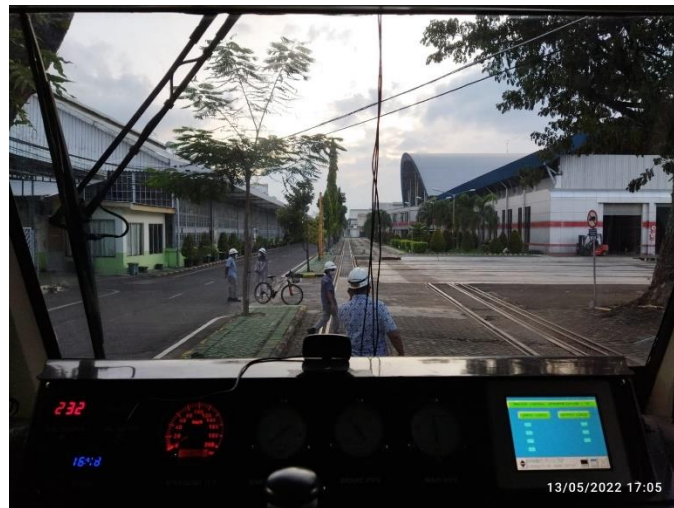
Pengujian yang dilakukan di kendaraan *testbed* pada tiga skenario menunjukkan bahwa *state* FSM akan bertransisi dari satu *state* ke *state* lainnya apabila kondisi transisinya terpenuhi. Objek diam di rel dianggap bergerak ke arah *testbed* karena posisi objek dalam kerangka lokal, dimana seharusnya dalam kerangka global.

Walaupun kecepatan objek dianggap nol, namun ketika semakin dekat kondisi state penghindar tabrakan terpenuhi.

Pada pengujian ini, meskipun situasi pengujian terbatas pada objek diam di rel dan di pinggir rel, namun semua state FSM dapat diamati perilakunya. Khususnya untuk situasi objek yang melintas dapat diwakili oleh objek yang diam di rel. Hal ini dapat dilakukan karena keterbatasan posisi objek dalam koordinat lokal, sehingga objek yang diam di rel dianggap bergerak ke kearah *testbed*.

IV.3. Pengujian Trem Otonom

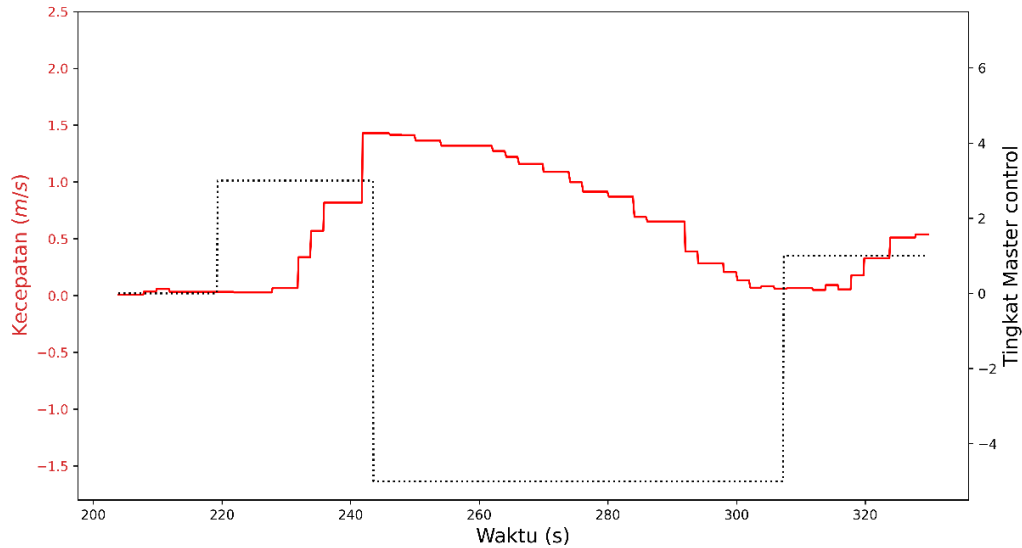
Pengujian pada trem dilakukan di rel PT INKA. Lingkungan pengujian seperti yang ditunjukkan pada Gambar IV.15. Pada pengujian ini hanya dua situasi yang dapat disimulasikan, yaitu situasi aman dan situasi pejalan kaki diam di pinggir rel. Pada pengujian ini NVIDIA drive AGX Pegasus telah terhubung dengan DbW menggunakan protokol komunikasi modbus, sehingga perintah sesuai tingkat *master control* dapat dijalankan oleh trem.



Gambar IV.15. Lingkungan pengujian di PT INKA.

Saat pengujian dengan menerapkan kendali ke trem, perintah *powering* dan *braking* dibatasi agar tidak terjadi kegagalan dalam sistem tingkat rendah trem. *Powering*

dibatasi pada P3 sedangkan *braking* diganti dengan perintah netral. Kecepatan maksimum rel adalah 1.5 m/s.



Gambar IV.16. Pengujian situasi aman dan situasi terdapat pengguna jalan raya diam di pinggir rel. (merah: kecepatan aktual, hitam: state FSM)

Gambar IV. 16 adalah hasil profil kecepatan dalam pengujian sistem pengambilan keputusan yang diimplementasikan di trem. sekitar 240 detik pertama, trem berada di *state* pengendali penjelajahan aktif dan menaikkan kecepatan ke kecepatan maksimum rel. kemudian pada 243 detik setelah memulai pengujian, terdeteksi pejalan kaki pada jarak kurang dari jarak aman. Hal ini menyebabkan *state* bertransisi ke *state* pengereman darurat. Pada Gambar IV.16 meskipun tingkat *master control* yang diberikan oleh sistem pengambilan keputusan adalah pengereman B5, namun pengereman yang dijalankan sesungguhnya adalah netral. Hasilnya trem dapat berjalan sesuai kecepatan maksimum rel yang telah ditentukan dan menghindari tabrakan secara otonom.

Bab V Kesimpulan

V.1. Kesimpulan

Sistem pengambilan keputusan untuk trem otonom yang dirancang memiliki kemampuan untuk menilai risiko tabrakan dari tiap objek yang terdeteksi dan menentukan aksi berdasarkan penilaian tersebut. Sistem yang dirancang terdiri dari penilai risiko tabrakan dan pengambilan keputusan dan perencanaan. Penilai risiko tabrakan terdiri dari estimator rel yang bertugas menentukan *rail horizon*, prediktor lintasan yang bertugas memprediksi lintasan gerak objek-objek yang terdeteksi, dan penilaian keselamatan yang bertugas menentukan objek yang dapat membahayakan trem. Pengambilan keputusan dan perencanaan bertugas untuk menentukan aksi berdasarkan penilaian yang telah dilakukan dan merencanakan kecepatan yang sesuai.

Pengujian yang dilakukan disimulasi menunjukkan bahwa sistem pengambilan keputusan yang dirancang dapat menjalankan trem secara otonom. Sistem dapat memprediksi lintasan dari setidaknya 11 objek dan mampu menjaga keselamatan saat berada di persimpangan yang padat. Persentase capaian keselamatan simulasi adalah 80% dari 50 kali simulasi. Ditinjau dari tiap *state* FSM-nya, persentase capaian keselamatan saat berada di *state* pengendali penjelajahan aktif adalah 96.94%, sedangkan capaian keselamatan *state* penghinder tabrakan dan *state* pengereman darurat adalah 100%. Pengamatan simulasi menunjukkan kasus tabrakan 3.06% terjadi karena mengambil keputusan pengendali penjelajahan aktif ketika trem menemui mobil yang melintang di rel dan ketika terdapat mobil yang akan melintas saat trem dalam keadaan diam. Selain itu, meskipun pengambilan keputusan penghinder tabrakan 100% mampu menjaga keselamatan trem, namun sebagian dari keputusan yang diambil kurang tepat meskipun tidak mengancam keselamatan trem, yaitu mengambil keputusan penghinder tabrakan ketika terdapat mobil, motor, sepeda, dan pejalan kaki yang salah diprediksi lintasannya memotong rel.

Sistem pengambilan keputusan yang dirancang mampu di implementasikan pada NVIDIA Drive AGX Pegasus. Pengujian yang dilakukan pada kendaraan *testbed* menunjukkan perilaku pengambilan keputusan siap diujikan pada trem yang menerapkan sinyal kendali. Hasil pengujian pada trem di rel PT INKA menunjukkan trem mampu berjalan secara otonom, yaitu mampu berjalan sesuai kecepatan maksimum jalur dan mampu menghindari tabrakan.

V.2. Saran

Untuk penelitian selanjutnya dan pengembangan penelitian ini, penulis menyarankan beberapa hal, yaitu:

1. Memperbaiki akurasi prediksi lintasan untuk objek-objek sekitar agar meminimalisir aksi penghindaran tabrakan yang tidak diperlukan.
2. Memperbaiki kondisi transisi yang menggunakan indikator keselamatan *time-to-collision* (TTC) karena indikator ini memiliki kelemahan pada saat kecepatan trem 0.
3. Mempertimbangkan ukuran kendaraan dan pejalan kaki yang terdeteksi.

DAFTAR PUSTAKA

- Buehler, M., Iagnemma, K., dan Singh, S. (2009): The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, in *Series Springer Tracts in Advanced Robotics*, Springer-Verlag, vol. 56.
- Camacho, E. F., dan Bordons, C. (1999): Model Predictive Control, *Advanced Textbooks in Control and Signal Processing*, doi:10.1007/978-1-4471-3398-8
- Chen, H., Liu, L., Zhang, Y., dan Tian, Y. (2017): Adaptive speed control of autonomous vehicle under changing operation conditions, *2017 36th Chinese Control Conference (CCC)*, pp. 3522-3526, DOI: 10.23919/ChiCC.2017.8027903.
- Damayanti, M., Malkhamah, S., dan Walker, K. (2015): Tramway Management System in Indonesia, *Journal of the Civil Engineering Forum* 1(1), hal. 23–28.
- Dani, S., Sonawane, D., Ingole, D., dan Patil, S. (2017): Performance evaluation of PID, LQR and MPC for DC motor speed control, *2017 2nd International Conference for Convergence in Technology (I2CT)*, Mumbai, India, pp. 348-354, doi: 10.1109/I2CT.2017.8226149.
- Data Kecelakaan di Perlindungan Kereta 2 Tahun Terakhir, diperoleh melalui situs internet: <https://bisnis.tempo.co/read/1244503/kai-catat-655-kecelakaan-di-perlintasan-kereta-2-tahun-terakhir>. Diunduh pada tanggal 7 Desember 2022.
- Data Kecelakaan di Perlindungan Sebidang KA, diperoleh melalui situs internet: https://www.kai.id/information/full_news/4045-kedisiplinan-rendah-terjadi-198-kecelakaan-di-perlintasan-sebidang-ka. Diunduh pada tanggal 7 Desember 2022.
- Di Palma, C., Galdi, V., Calderaro, V. dan De Luca, F. (2020): Driver Assistance System for Trams: Smart Tram in Smart Cities. *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, hal. 1–6.
- Gonzalez, L., Matute-Peaspan, J. A., Rastelli, J. P., dan Calvo, I. (2020): Longitudinal Collision Avoidance Based on Model Predictive Controllers and Fuzzy Inference Systems, *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-6, DOI: 10.1109/ITSC45102.2020.9294584.
- Hayward, J. C. (1971): Near Misses as a Measure of Safety at Urban Intersections, *PhD thesis*, Department of Civil Engineering, Pennsylvania State University, University Park, PA, USA.
- Hirst, S., dan Graham, R. (1997): The format and presentation of collision warnings, In *Ergonomics and Safety of Intelligent Driver Interfaces* (Noy YI (ed.). Lawrence, Erlbaum Associates, Mahwah, NJ, USA, pp. 1–22.
- Hogema, J., dan Janssen, W. (1996): Effects of Intelligent Cruise Control on Driving Behaviour: a Simulator Study, *TNO Human Factors*, Soesterberg, the Netherlands, Report tm-1996-c-12.

- Ibañez-Guzmán, J., Laugier, C., Yoder, J., dan Thrun, S. (2012): Autonomous Driving: Context and State-of-the-Art, in *Handbook of Intelligent Vehicles*, Eskandarian Azim, Ed., London: Springer London, pp. 1271-131, DOI: 10.1007/978-0-85729-085-4_50.
- Katz, R., dan Schulz, R. (2013): Towards the development of a laserscanner-based collision avoidance system for trams, *2013 IEEE Intelligent Vehicles Symposium (IV)*, hal. 725–729.
- Kusumadewi, S., dan Purnomo, H. (2004): Aplikasi logika fuzzy untuk pendukung keputusan / Sri Kusumadewi, Hari Purnomo, Yogyakarta : Graha Ilmu. 491 hlm.
- Lee, E. A., dan Seshia, S. A. (2017): Introduction to Embedded Systems - A Cyber-Physical Systems Approach, *MIT Press*, Second Edition.
- Lefèvre, S., Vasquez, D., dan Laugier, C. (2014): A survey on motion prediction and risk assessment for intelligent vehicles, *Robomech J* **1**, 1. <https://doi.org/10.1186/s40648-014-0001-z>.
- Lehner, A., Strang, T., dan Rico-García, C. (2008): A reliable surveillance strategy for an autonomous Rail Collision Avoidance System, *Proceedings of the ITS World Congress 2008*.
- Leonard, J., How, J., Teller, S., Berger, M., Campbell S., dkk. (2008): A perception-driven autonomous urban vehicle, *J. Field Robot*, 25:727–74
- Li, G., Yang, Y., Zhang, T., Qu, X., Cao, D., Cheng, B., dan Li, K. (2021): Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios, *Transportation Research Part C: Emerging Technologies*, vol.122, pp. 102820, DOI: 10.1016/j.trc.2020.102820.
- Li, K. (2018): The Challenges and Potential of Risk Assessment for Active Safety of Unmanned Tram. *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, hal. 22–27.
- Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., dan Shi, W. (2021): Computing Systems for Autonomous Driving: State of the Art and Challenges, in *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469-6486, DOI: 10.1109/JIOT.2020.3043716.
- Lüy, M., Çam, E., Ulaş, F., Uzun, İ., dan Akın, S.İ. (2018): Initial Results of Testing a Multilayer Laser Scanner in a Collision Avoidance System for Light Rail Vehicles, *Applied Sciences* 8(4).
- Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., dkk. (2008): Junior: the Stanford entry in the Urban Challenge, *J. Field Robot*, 25:569–97.
- Naweed, A. dan Rose, J. (2015): “It’s a Frightful Scenario”: A Study of Tram Collisions on a Mixed-traffic Environment in an Australian Metropolitan Setting. *Procedia Manufacturing* 3, hal. 2706–2713.
- NVIDIA Drive overview. (2019): www.nvidia.com/drive.
- Palmer, A. W., Sema, A., Martens, W., Rudolph, P., dan Waizenegger, W. (2020): The Autonomous Siemens Tram, *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1-6, DOI: 10.1109/ITSC45102.2020.9294699.
- Patlins, A., Kunicina, N., Zhiravecka, A., dan Shukaeva, S. (2010): LIDAR Sensing Technology Using in Transport Systems for Tram Motion Control, *Elektronika Ir Elektrotechnika*, 101, 13-16.

- Saffarzadeh, M., Nadimi, N., Naseralavi, S., dan Mamdoohi, A.R. (2013): A general formulation for time-to-collision safety indicator. *Proceedings of the ICE – Transport*, 166, 294-304, 10.1680/tran.11.00031.
- Schwarting, W., Alonso-Mora, J., dan Rus, D. (2018): Planning and decision-making for autonomous vehicles, *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187-210.
- Sun, Q., Huang, X., Gu, J., Williams, B. C., dan Zhao, H. (2022): M2I: From Factored Marginal Trajectory Prediction to Interactive Prediction, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 6533-6542, doi: 10.1109/CVPR52688.2022.00643.
- Tomar, B., dan Kumar, N. (2021): Fuzzy Logic based Braking System for Metro Train, *2021 International Conference on Intelligent Technologies (CONIT)*, Hubli, India, pp. 1-4, doi: 10.1109/CONIT51480.2021.9498360.
- Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., dkk. (2008): Autonomous driving in urban environments: Boss and the Urban Challenge, *J. Field Robot*, 25:425–66.
- Van der horst, R., dan Hogema, J. (1993): Time-to-collision and collision avoidance systems. *Proceedings of the 6th ICTCT Workshop: Safety Evaluation of Traffic Systems: Traffic Conflicts and Other Measures*, Salzburg, Austria.
- Wang, L. J., Zhang, H., Meng, H. D., dan Wang, X. Q. (2008): A model based on TTC to describe how drivers control their vehicles, *European Physical Journal B*, 66(1): 149–153.