

Manuver Autonomous Car ITS di Bundaran atau U-Turn Menggunakan Deep Reinforcement Learning

1st Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.

Department of Computer Engineering

*Faculty of Intelligent Electrical
and Informatics Technology*

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

hery@ee.its.ac.id

2nd Dr. I Ketut Eddy Purnama, ST., MT.

Department of Computer Engineering

*Faculty of Intelligent Electrical
and Informatics Technology*

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

ketut@ee.its.ac.id

3rd Muhtadin, ST., MT.

Department of Computer Engineering

*Faculty of Intelligent Electrical
and Informatics Technology*

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

muhtadin@te.its.ac.id

4rd Muhammad Roychan Meliaz

Department of Computer Engineering

*Faculty of Intelligent Electrical
and Informatics Technology*

Institut Teknologi Sepuluh Nopember

Surabaya, Indonesia 60111

meliaz.17072@mhs.its.ac.id

Abstract—Autonomus Car atau kendaraan otonom merupakan kendaraan yang memiliki kemampuan untuk berkendara secara mandiri layaknya dikendalikan manusia dengan menggunakan rangkaian kecerdasan buatan. Pada penelitian ini kami mengajukan riset pengembangan kendaraan otonom iCar ITS (Intelligent Car Institut Teknologi Sepuluh Nopember) dengan mengembangkan sistem manuver kendaraan otonom di bundaran atau u-turn dalam lingkungan yang disimulasikan. Dalam lingkungan simulasi, model yang digunakan adalah model kendaraan yang disesuaikan dengan iCar. Pengembangan sistem navigasi dan manuver kendaraan otonom dilakukan menggunakan metode Deep Reinforcement Learning, salah satu cabang dari Machine Learning. Pada penelitian ini, didapatkan hasil model reinforcement learning yang mampu melakukan manuver bundaran simpang empat dan bundaran tanpa simpang dengan nilai rerata deviasi sudut dari jalurnya masing-masing senilai 27.011° dan 30.068° , mampu bermanuver tanpa collision selama rerata 13.3 detik dan 7.9 detik, serta dengan kecepatan rerata 27.0 kmpj dan 28.5 kmpj.

Index Terms—Kendaraan otonom, Reinforcement Learning, Simulasi

I. LATAR BELAKANG

TEKNOLOGI kendaraan otonom memiliki sejarah yang cukup panjang. Purwarupa pertama yang dapat berfungsi dengan baik diciptakan pada tahun 1980. Dengan menggunakan kamera, purwarupa ini berhasil menempuh 100km jalan kosong tanpa perlu dikemudikan oleh manusia. Dengan keberhasilan ini, muncul banyak proyek pada tahun 80-an dan 90-an menggunakan sistem serupa yang digunakan untuk menyetir melalui jalan raya, baik pada lalu lintas ringan atau tidak ada sama sekali. Dalam pengembangannya, kendaraan otonom dapat memecahkan masalah keselamatan berkendara dan efisiensinya. Maka dari itu, tujuan utama dilakukan peneli-

tian adalah untuk mencegah atau mengurangi kecelakaan lalu lintas, mengurangi waktu orang berkendara, serta mengurangi emisi karbon. [1]

Salah satu produk *autonomous car* yang tengah dikembangkan adalah iCar ITS (Intelligent Car Institut Teknologi Sepuluh Nopember). iCar ITS merupakan purwarupa mobil yang dilengkapi dengan fitur pengemudiannya secara otonom sebagai hasil riset kolaborasi dari para peneliti ITS dengan berbagai bidang keahlian. [2] iCar ITS dioperasikan sebagai mobil komuter yang melayani perjalanan penumpang menuju berbagai tujuan dalam kampus ITS. [3]

Metode yang lazim digunakan untuk mengembangkan kendaraan otonom adalah *reinforcement learning*, sebuah bagian dari *machine learning* yang juga merupakan bagian dari *artificial intelligence*. *Reinforcement Learning* adalah sebuah metode pembelajaran mengenai apa yang meski dilakukan (mengimplementasikan aksi kedalam situasi) pada sebuah masalah/problem untuk mendapatkan hasil/reward yang maksimal. *Agent* tidak diberi *clue* mengenai aksi apa yang harus dilakukan. *Agent* akan mempelajari aksi dengan prinsip *Trial and Error*, lalu mengambil keputusan berdasarkan reward yang didapatkan (*reward* maksimal).

Dalam pengaplikasianya, iCar ITS masih menggunakan metode tradisional rule-based tanpa memanfaatkan metode *machine learning*. Metode tersebut tidak cukup baik karena pengembang harus memprogram secara manual setiap skenario yang akan dihadapi oleh iCar. Metode tersebut tidak akan bertahan lama mengingat banyaknya skenario dunia nyata yang hampir tidak mungkin untuk diberi solusi secara manual.

II. DESAIN DAN IMPLEMENTASI SISTEM

Pada *paper* ini dijelaskan mengenai penerapan metode *Deep Reinforcement Learning* yang bertujuan untuk menciptakan algoritma pada mobil otonom yang mampu melakukan manuver di bundaran/u-turn dengan efisien.

A. Persiapan Lingkungan Simulasi

Menggunakan simulator CARLA, digunakan map Town03 CARLA, yang memiliki bundaran dan u-turn untuk memfasilitasi penggerjaan penelitian. Terdapat dua jenis bundaran yang digunakan pada penelitian ini.

1) *Bundaran Simpang Empat*: Digunakan bundaran berjumlah simpang empat seperti pada Gambar 1. Pada lingkungan ini, agent akan *spawn* dari empat titik *spawn* di sekitar bundaran.



Fig. 1: Bundaran Simpang Empat

2) *Bundaran Tanpa Simpang*: Digunakan bundaran tanpa simpang seperti pada Gambar 2. Pada lingkungan ini, agent akan *spawn* dari sebuah titik *spawn* awal bundaran.



Fig. 2: Bundaran Tanpa Simpang

B. Sensor

Sensor menggunakan sensor kamera yang diletakkan secara fisik di bagian depan bawah dari agen. Sensor kamera yang digunakan adalah sensor kamera segmentasi. Sensor kamera segmentasi adalah sensor kamera yang memisahkan object-object di simulator menjadi berbagai warna unik yang solid. Citra yang dihasilkan dari sensor kamera segmentasi pada Gambar 4 adalah citra lanjutan yang telah di proses dari citra RGB pada gambar 3

C. Akuisisi Data

Data citra diambil dari sensor dengan ukuran 480x270. Citra yang digunakan adalah citra segmentasi yang telah disediakan oleh sensor kamera segmentasi dari CARLA Simulator. Terlihat pada Gambar 3 merupakan citra awal. Kemudian dilakukan segmentasi pada citra tersebut sehingga setiap obyek dipresentasikan dengan warna solid seperti pada Gambar 4



Fig. 3: Citra RGB



Fig. 4: Citra Segmentasi

Ada dua jenis citra yang digunakan dalam penelitian ini, yaitu citra segmentasi grayscale dan citra segmentasi lanjutan.

Citra segmentasi grayscale merupakan citra segmentasi yang kemudian diberi filter grayscale agar dapat meminimalkan data yang dianalisa namun tetap mempertahankan fitur yang ada.

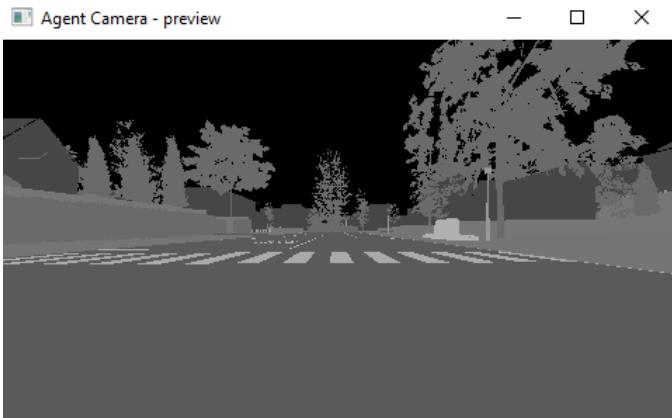


Fig. 5: Citra Segmentasi Grayscale

Citra segmentasi lanjutan merupakan citra segmentasi, yang kemudian di segmentasi kembali menjadi dua jenis obyek, yaitu *drivable* dan *non-drivable*. Obyek *drivable* dipresentasikan dengan warna putih dan obyek *non-drivable* dipresentasikan dengan warna hitam.

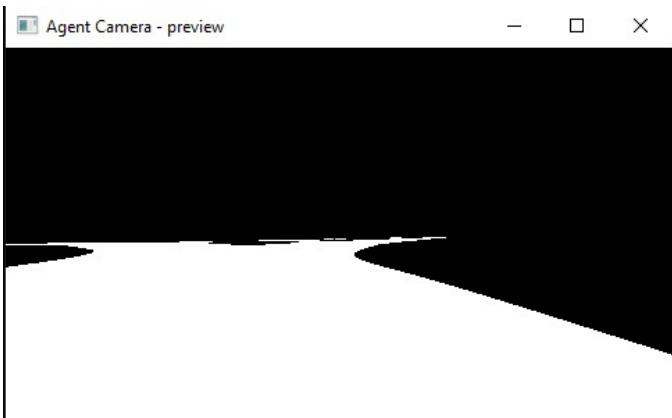


Fig. 6: Citra Segmentasi Lanjutan

Citra segmentasi grayscale dan citra segmentasi lanjutan akan menjadi hasil akhir akuisisi citra yang kemudian akan diberikan ke algoritma DQN untuk melakukan *training* model dan/atau *inference* model.

D. Action

Ada 3 *action* yang bisa dilakukan oleh *agent*. Diantaranya:

- 1) *forward*
Agent *throttle* dengan nilai 1 dan *steer* dengan nilai 0. Dengan demikian agent akan melakukan gerakan manuver maju.
- 2) *forward_left*
Agent *throttle* dengan nilai 1 dan *steer* dengan nilai -1. Dengan demikian agent akan melakukan gerakan manuver maju ke depan kiri.
- 3) *forward_right*

Agen *throttle* dengan nilai 1 dan *steer* dengan nilai 1. Dengan demikian agent akan melakukan gerakan manuver maju ke depan kanan.

E. Reward Function

Fungsi reward dirancang agar mobil otonom mampu bergerak sepanjang lintasan dengan cepat dan aman.

1) *Bundaran Simpang Empat*: Ada dua buah reward yang ditetapkan dimana masing-masing reward memperhatikan referensi *target lane*. Target lane adalah garis imajiner yang menjadi target gerakan mobil otonom. Terlihat pada Gambar 7, *target lane* digambarkan dengan titik-titik merah di sekitar bundaran.



Fig. 7: Target lane

- 1) *Reward 1*: Deviasi sudut dari *target lane* Reward yang digunakan pada Reward 1 adalah $1/\alpha$. Alpha adalah sudut agent yang menyimpang dari *target lane*. Teknis alpha dijelaskan pada Gambar 8.

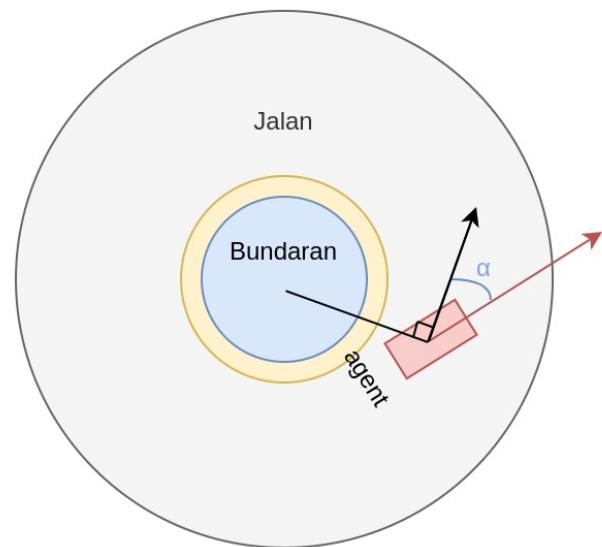


Fig. 8: Reward 1

Garis hitam merupakan vektor tegak lurus antara titik pusat bundaran ke titik pusat agen. Lalu garis merah

adalah vektor arah mobil. Dari kedua nilai tersebut, didapat alpha yang merupakan sudut dari kedua vektor tersebut. Kemudian nilai reward didefinisikan dengan $1/\alpha$. Nilai tertinggi reward dibatasi menjadi 1. Hal ini dilakukan agar fluktuasi reward yang diberikan ke agent tidak terlalu tinggi untuk perubahan nilai yang kecil.

- 2) *Reward 2:* Deviasi jarak dari *target lane*. Reward yang digunakan pada reward 2 adalah $1/\text{deviasi_jarak} \times 10$. Deviasi jarak adalah jarak terkecil agen terhadap *target lane* dalam meter.

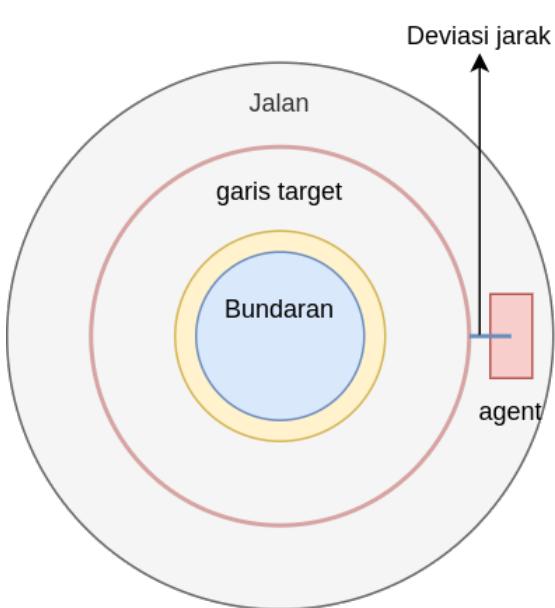


Fig. 9: Reward 2

Nilai tertinggi reward dibatasi menjadi 1. Hal ini dilakukan agar fluktuasi reward yang diberikan ke agent juga tidak terlalu tinggi untuk perubahan nilai yang kecil, seperti pada Reward 1.

Selain reward yang bernilai positif, diperlukan juga reward yang bernilai negatif untuk mengurangi peluang agent melakukan hal yang sebaiknya tidak dilakukan. Ada dua buah reward bernilai negatif yang diberikan.

- 3) *Reward 3:* Sentuhan dengan obyek lain Reward senilai -1 akan diberikan pada *agent* jika agent menyentuh obyek lain selain jalan raya, tanah, dan trotoar.
- 4) *Reward 4:* Deviasi jarak terlalu besar Reward senilai -0.5 akan diberikan pada *agent* jika jarak antara *agent* dan bundaran lebih besar dari 30 meter. Jarak 30 meter dari bundaran dapat dilihat di Gambar 10



Fig. 10: Batasan jarak

- 5) *Reward Total* Total nilai reward didefinisikan dengan $\text{Reward 1} + \text{Reward 2} + \text{Reward 3} + \text{Reward 4}$.
- 2) *Bundaran Tanpa Simpang:* Pada bundaran tanpa simpang, fungsi reward yang diberikan berbeda. Pada kasus ini disediakan titik-titik jalur sebagai *waypoint* yang harus diikuti oleh *agent*.

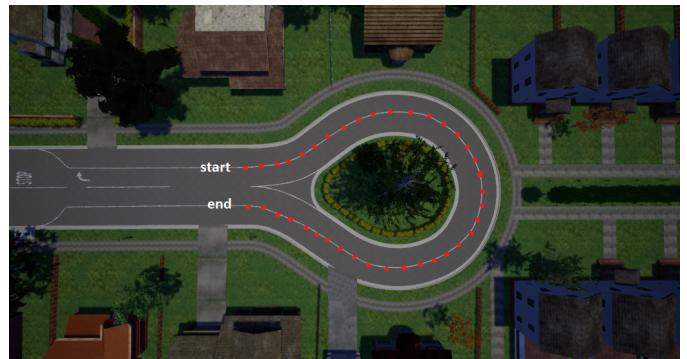


Fig. 11: Waypoint

Waypoint merupakan titik-titik pada map sebagai target tujuan jalannya *agent*. Terlihat pada Gambar 11, titik-titik merah merupakan ilustrasi dari *waypoint*. *Waypoint* berjumlah 100 titik dimana setiap titiknya berjarak satu meter.

Reward didefinisikan dengan $1/\alpha$. Alpha merupakan deviasi sudut arah *agent* terhadap arah titik *waypoint* terdekat dari *agent* + 5.



Fig. 12: Waypoint dari Sisi Agent

Gambar 12 menunjukkan bahwa *waypoint* terdekat + 5 dari *agent* berjarak enam meter dari pusat koordinat *agent*.

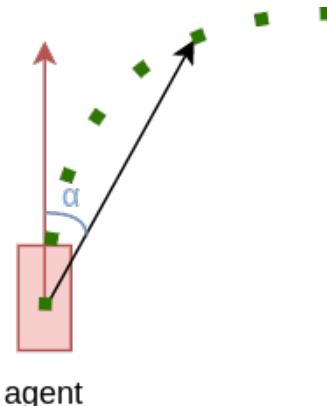


Fig. 13: Reward Alpha

Ilustrasi fungsi reward dapat dilihat pada Gambar 13, di mana titik-titik hijau adalah *waypoints*, garis merah adalah arah dari *agent*, garis hitam adalah arah dari *agent* ke *waypoint*+5 dari *agent*, serta alpha adalah perbedaan sudut dari kedua nilai sudut tersebut. Nilai tertinggi reward dibatasi menjadi 1. Hal ini dilakukan agar fluktuasi reward yang diberikan ke agent juga tidak terlalu tinggi untuk perubahan nilai yang kecil, seperti pada Reward 1 dan Reward 2 pada bundaran simpang empat.

3) *End Episode*: Waktu maksimal yang diberikan pada agent untuk melakukan proses learning setiap episodenya adalah 10 detik.

Episode akan berakhir jika waktu maksimal episode berakhir atau agent menyentuh object lain selain jalan raya, tanah, dan trotoar.

F. Parameter DQN

Penentuan nilai hyperparameter yang tepat merupakan salah satu langkah yang penting dilakukan untuk mendapatkan model pembelajaran mesin yang baik. Dalam machine learning, hyperparameter adalah parameter yang digunakan untuk mengatur jalannya proses pembelajaran mesin. Berbeda dengan parameter model yang nilainya mengalami perubahan seiring berjalannya pembelajaran mesin, hyperparameter perlu didefinisikan di awal dan umumnya bernilai tetap sepanjang proses pembelajaran. Dalam penelitian ini, hyperparameter dari algoritma DQN didefinisikan sebagai berikut:

Hyperparameter	Nilai	Deskripsi
MINIBATCH_SIZE	16	Jumlah sampel pembelajaran yang diproses oleh perhitungan SGD (stochastic gradient) algoritma DQN
PREDICTION_BATCH_SIZE	1	Jumlah sampel yang di prediksi di saat yang bersamaan
TRAINING_BATCH_SIZE	8	Jumlah sampel yang di fit di saat yang bersamaan (lebih besar lebih cepat)
TRAINER_MEMORY_FRACTION	0.6	

TABLE I: Hyperparameter model.

1) Model:

Hyperparameter	Nilai	Deskripsi
DISCOUNT	0.99	Nilai faktor discount dalam perhitungan reward algoritma DQN
REPLAY_MEMORY_SIZE	20_000	Berapa banyak step terakhir yang disimpan untuk training model
MIN_REPLAY_MEMORY_SIZE	5_000	Jumlah step minimum dalam memori untuk memulai training
OPTIMIZER_LEARNING_RATE	0.001	Laju pembelajaran yang digunakan pada optimizer
OPTIMIZER_DECAY	0.0	Pengurangan laju pembelajaran pada optimizer setiap episodenya

TABLE II: Hyperparameter DQN.

2) DQN:

Hyperparameter	Nilai	Deskripsi
START_EPSILON	1	Nilai epsilon saat pertamakali mulai training
EPSILON_DECAY	0.9995	Penurunan nilai epsilon di setiap episodenya
MIN_EPSILON	0.1	Epsilon terendah yang diperbolehkan

TABLE III: Hyperparameter epsilon.

3) *Epsilon*: Penentuan parameter epsilon ditentukan oleh kondisi berikut. Epsilon akan bernilai 1 saat pertamakali memulai learning atau pada saat episode 0. Setiap dimulainya episode baru, nilai epsilon akan dikali dengan nilai 0.9995 hingga pada akhirnya akan bernilai 0.1 setelah 4700 episode. Pengurangan nilai epsilon akan berhenti setelah sampai pada nilai 0.1.

Penentuan parameter *epsilon-greedy* ini dilakukan agar agent mampu melakukan kegiatan eksplorasi dan eksplorasi dengan tepat. Sehingga hasil yang didapatkan akan baik dalam waktu cepat.

G. Arsitektur Model

Berikut adalah arsitektur yang digunakan:

Layer (type)	Output Shape	Param #
conv2d_1_input (InputLayer)	(None, 270, 480, 1)	0
conv2d_1 (Conv2D)	(None, 270, 480, 64)	640
activation_1 (Activation)	(None, 270, 480, 64)	0
average_pooling2d_1 (AveragePooling2D)	(None, 90, 160, 64)	0
conv2d_2 (Conv2D)	(None, 90, 160, 64)	36928
activation_2 (Activation)	(None, 90, 160, 64)	0
average_pooling2d_2 (AveragePooling2D)	(None, 30, 54, 64)	0
conv2d_3 (Conv2D)	(None, 30, 54, 64)	36928
activation_3 (Activation)	(None, 30, 54, 64)	0
average_pooling2d_3 (AveragePooling2D)	(None, 10, 18, 64)	0
flatten_1 (Flatten)	(None, 11520)	0
kmh_input (InputLayer)	(None, 1)	0
concatenate_1 (Concatenate)	(None, 11521)	0
dense_1 (Dense)	(None, 256)	2949632
dense_2 (Dense)	(None, 3)	771

TABLE IV: Arsitektur model.

Diberikan konvolusi kepada citra image sebanyak 3 kali. Setiap konvolusi dilakukan dengan average pooling. Proses selanjutnya setelah konvolusi adalah menambahkan data kecepatan agent. Kemudian pada akhir dari proses akan didapatkan output action berjumlah 3.

H. Training

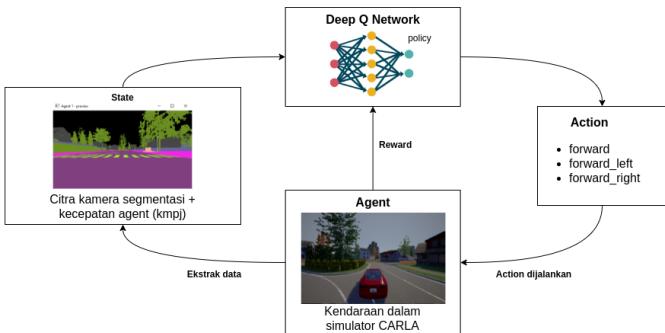


Fig. 14: Diagram Blok Metodologi

Proses training dilakukan setelah seluruh konfigurasi selesai dilakukan. Metodologi training dari penelitian ini terdapat pada Gambar 14. *Agent* yang berada pada simulasi melakukan suatu *action* yang menyebabkan berubahnya *state*. *State* yang berupa citra segmentasi dan kecepatan agent (kmpj) serta *reward* dari *agent* kemudian dikirimkan ke DQN untuk dilakukan training. Hasil dari training berupa *policy*, akan menentukan *action* yang kemudian akan dilaksanakan oleh *agent*. Kemudian proses tersebut akan terulang kembali menjadi sebuah siklus yang tak henti.

III. TESTING AND RESULT

A. Training DQN

Training algoritma DQN dilakukan dengan menggunakan hardware berikut:

CPU	Intel Core i5
GPU	Nvidia GTX 1060 - 6GB VRAM
RAM	8GB
Disk	256GB SSD M.2
OS	Windows 10

TABLE V: Hardware yang digunakan untuk learning.

Learning dilakukan dengan mesin lokal menggunakan GPU. Lingkungan software machine learning yang digunakan pada learning berupa:

Software	Version
tensorflow-gpu	1.13.1
keras	2.2.5
h5py	3.1
python	3.7.9

TABLE VI: Lingkungan software yang digunakan untuk learning.

Hasil dari proses training tersebut adalah model jaringan DQN yang merepresentasikan sistem perencanaan gerakan mobil otonom. Model tersebut disimpan dalam sebuah file berformat .model yang selanjutnya dapat divisualisasikan dengan tensorboard untuk melihat hasil training algoritma DQN.

B. Training DQN pada Bundaran Simpang Empat dengan Segmentasi Grayscale

Training algoritma DQN pada bundaran simpang empat dengan segmentasi grayscale berlangsung selama 20 jam. Berikut adalah hasil visualisasi training algoritma DQN:

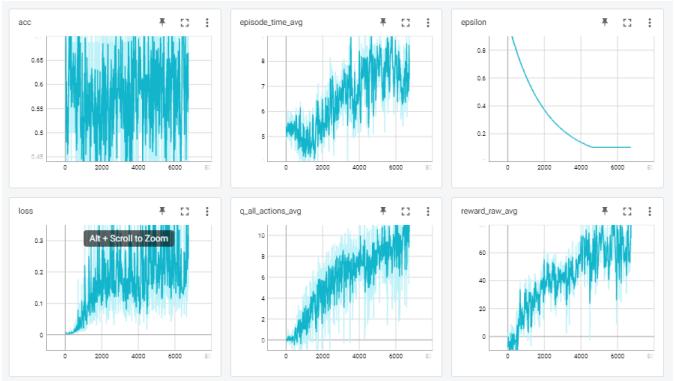


Fig. 15: Tensorboard pada Bundaran Simpang Empat dengan Segmentasi Grayscale

Dari hasil visualisasi training DQN pada Gambar 15, dapat dilihat bahwa proses training algoritma DQN telah berjalan dengan baik. Hal itu dapat dilihat dari nilai akurasi, episode time, dan reward berkendara mobil otonom yang mengalami tren kenaikan seiring dengan berjalannya proses training.

C. Training DQN pada Bundaran Simpang Empat dengan Segmentasi Lanjutan

Training algoritma DQN pada bundaran simpang empat dengan segmentasi lanjutan berlangsung selama 13 jam, 30 menit, dan 48 detik. Berikut adalah hasil visualisasi training algoritma DQN:

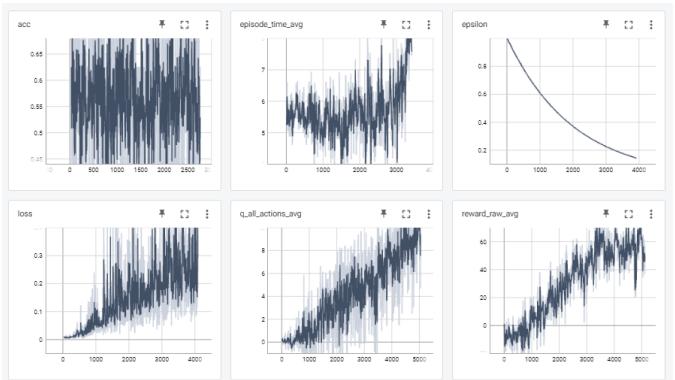


Fig. 16: Tensorboard pada Bundaran Simpang Empat dengan Segmentasi Lanjutan

Dari hasil visualisasi training DQN pada Gambar 17, dapat dilihat bahwa proses training algoritma DQN telah berjalan dengan baik. Hal itu dapat dilihat dari nilai akurasi, episode time, dan reward berkendara mobil otonom yang mengalami tren kenaikan seiring dengan berjalannya proses training.

D. Training DQN pada Bundaran Tanpa Simpang dengan Segmentasi Lanjutan

Training algoritma DQN pada bundaran tanpa simpang dengan segmentasi lanjutan berlangsung selama 26 jam dan 25 detik. Berikut adalah hasil visualisasi training algoritma DQN:

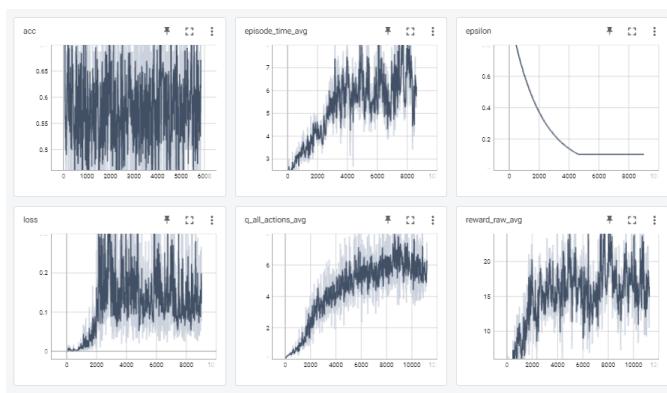


Fig. 17: Tensorboard pada Bundaran Tanpa Simpang dengan Segmentasi Lanjutan

Dari hasil visualisasi training DQN pada Gambar 17, dapat dilihat bahwa proses training algoritma DQN telah berjalan dengan baik. Hal itu dapat dilihat dari nilai akurasi, episode time, dan reward berkendara mobil otonom yang mengalami tren kenaikan seiring dengan berjalannya proses training.

IV. PENGUJIAN MODEL DQN

Pada bagian ini, akan dilakukan pengujian model hasil dari proses training DQN. Pengujian dilakukan saat lingkungan berkendara dalam kondisi normal.

A. Pengujian Model DQN pada Bundaran Simpang Empat dengan Segmentasi Grayscale

episode_length	avg_speed	avg_angle_diff	avg_dist_diff	avg_reward
9.005	34.983	20.001	5.812	0.166
4.772	28.927	12.14	0.91	0.448
8.524	33.052	10.65	1.993	0.398
12.678	35.053	15.259	3.264	0.222
13.621	37.03	32.642	12.394	0.122
4.851	30.102	10.241	1.253	0.471
6.555	20.433	21.37	2.775	0.318
8.694	31.494	14.927	3.381	0.235
20.184	34.142	58.799	45.228	-0.23
10.026	28.305	12.737	2.411	0.332
5.147	29.648	17.682	1.707	0.289
8.521	32.238	13.255	2.253	0.289
7.99	32.617	13.939	2.661	0.342
5.206	30.142	13.541	1.077	0.409
6.072	23.77	36.641	3.617	0.17
5.426	31.166	12.926	0.925	0.418
5.176	27.477	12.276	1.194	0.457
8.085	33.884	11.883	1.224	0.343
10.048	27.918	15.146	3.086	0.306
7.443	33.063	12.29	2.298	0.383
29.772	38.947	65.564	58.657	-0.255
6.617	30.019	20.405	1.658	0.425
4.931	27.362	12.078	1.821	0.361
14.428	29.973	18.9	3.066	0.272
5.445	30.364	10.48	0.951	0.523
8.794	31.964	15.54	2.635	0.294
8.568	33.002	13.103	1.95	0.409
20.471	24.614	64.263	15.817	-0.076
5.276	31.54	14.182	1.594	0.349
7.645	29.516	39.559	1.573	0.349

TABLE VII: 30 Sampel Hasil Model DQN pada Bundaran Simpang Empat dengan Segmentasi Grayscale.

Dari hasil pengambilan sampel tersebut didapatkan kesimpulan seperti pada tabel VIII.

avg_episode_length	avg_speed	avg_angle_diff	avg_dist_diff	avg_reward
9.332	30.758	21.413	6.306	0.284

TABLE VIII: Kesimpulan Hasil Model DQN pada Bundaran Simpang Empat dengan Segmentasi Grayscale.

B. Pengujian Model DQN pada Bundaran Simpang Empat dengan Segmentasi Lanjutan

episode_length	avg_speed	avg_angle_diff	avg_dist_diff	avg_reward
7.431	38.229	10.205	2.0	0.34
8.094	30.83	24.743	1.187	0.392
8.418	30.337	20.039	2.106	0.257
21.587	37.044	14.712	2.588	0.28
7.019	35.495	11.015	0.849	0.465
19.252	35.029	20.848	3.259	0.267
18.607	29.382	58.62	6.881	0.071
6.529	36.715	10.4	1.301	0.348
14.337	36.066	16.105	0.99	0.421
17.025	34.249	45.883	4.252	0.15
9.511	41.363	12.647	1.503	0.32
36.864	34.917	68.418	8.128	0.069
9.973	38.902	13.901	1.56	0.304
7.34	35.594	13.333	1.547	0.45
14.864	31.482	59.456	3.588	0.112
9.987	41.68	12.595	1.952	0.297
9.767	39.018	15.001	1.341	0.373
7.011	36.749	10.62	1.488	0.321
17.4	26.631	79.254	13.336	-0.07
12.798	40.187	15.749	1.645	0.285
17.809	31.448	46.125	2.951	0.268
11.504	36.149	21.827	1.347	0.355
6.395	37.127	9.601	0.895	0.418
23.591	30.945	60.991	4.07	0.153
6.957	37.284	8.505	0.959	0.438
9.431	40.093	11.436	1.98	0.311
35.604	27.912	80.507	11.311	-0.171
9.089	42.26	10.891	1.286	0.395
7.032	36.387	11.418	1.278	0.418
8.578	34.796	15.489	1.456	0.316

TABLE IX: 30 Sampel Hasil Model DQN pada Bundaran Simpang Empat dengan Segmentasi Lanjutan.

Dari hasil pengambilan sampel tersebut didapatkan kesimpulan seperti pada tabel X.

avg_episode_length	avg_speed	avg_angle_diff	avg_dist_diff	avg_reward
13.326	35.476	27.011	2.967	0.278

TABLE X: Kesimpulan Hasil Model DQN pada Bundaran Simpang Empat dengan Segmentasi Lanjutan.

C. Pengujian Model DQN pada Bundaran Tanpa Simpang dengan Segmentasi Lanjutan

episode_length	avg_speed	avg_alpha	avg_reward
8.374	32.202	31.098	0.113
8.796	29.909	32.023	0.093
12.195	29.678	40.528	0.068
12.439	30.384	36.557	0.12
10.246	29.72	42.027	0.063
3.138	19.86	32.557	0.085
9.982	30.777	35.319	0.129
8.185	30.64	24.322	0.112
9.719	31.814	31.563	0.081
10.908	33.314	38.641	0.097
3.299	21.882	22.22	0.122
9.27	31.347	26.635	0.12
4.27	26.317	16.402	0.152
3.765	23.032	32.14	0.169
10.776	31.581	37.949	0.091
5.588	28.784	29.461	0.155
3.219	22.536	20.392	0.186
8.508	30.489	24.669	0.122
4.457	23.68	23.546	0.146
3.824	22.575	36.989	0.105
10.599	31.869	23.992	0.132
12.438	31.08	38.071	0.088
7.873	31.305	21.448	0.15
3.562	22.864	16.225	0.136
8.758	27.877	37.786	0.076
4.641	26.732	25.345	0.106
10.472	32.074	33.322	0.101
10.937	31.784	33.864	0.095
9.244	30.508	28.196	0.12

TABLE XI: 30 Sampel Hasil Model DQN pada Bundaran Tanpa Simpang dengan Segmentasi Lanjutan.

Dari hasil pengambilan sampel tersebut didapatkan kesimpulan seperti pada tabel XII.

avg_episode_length	avg_speed	avg_alpha	avg_reward
7.958	28.533	30.068	0.115

TABLE XII: Kesimpulan Hasil Model DQN pada Bundaran Tanpa Simpang dengan Segmentasi Lanjutan.

V. CONCLUSION

A. Kesimpulan

Dari perancangan dan pengujian sistem yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut. Kendala dan kekurangan yang penulis hadapi juga penulis tuliskan pada bagian saran dengan harapan untuk membantu pengembangan penelitian selanjutnya.

nolistsep

- 1) Sistem perencanaan gerakan mobil otonom berbasis algoritma DQN yang dirancang mampu melakukan akselerasi dan *steer* yang mampu mengontrol mobil otonom dan memahami kondisi sekitarnya.
- 2) Penggunaan *state* citra dengan segmentasi lanjutan; yaitu menghasilkan output citra *drivable* dan *non-drivable* menghasilkan hasil yang lebih baik daripada segmentasi biasa dengan performa *average_episode_length* 42.8% lebih baik, *average_speed*

15.3% lebih baik, dan *average_distance_difference* 112.4% lebih baik.

B. Saran

Untuk pengembangan selanjutnya pada topik penelitian perencanaan gerakan mobil otonom dengan menggunakan algoritma DQN, terdapat beberapa saran yang diberikan, antara lain sebagai berikut:

nolistsep

- 1) Sensor yang digunakan untuk navigasi dari *agent* menggunakan algoritma DQN dapat ditambah menjadi lebih banyak seperti GPS serta LIDAR, agar *agent* dapat bertindak dengan parameter yang lebih lengkap.
- 2) Proses training dapat dilakukan dengan waktu yang lebih panjang serta menggunakan mesin dengan tenaga komputasi yang lebih kuat agar didapatkan model yang konvergen.

REFERENCES

- [1] M. V. Rajasekhar and A. K. Jaswal, "Autonomous vehicles: The future of automobiles," pp. 1–6, 2015.
- [2] menristekbrin, "Menristek/kepala brin luncurkan icar its," *Biro Kerja Sama dan Komunikasi Publik Kemenristek/BRIN*, Aug 2020. [Online]. Available: <https://www.ristekbrin.go.id/menristek-kepala-brin-luncurkan-icar-its/>
- [3] itsnews, "I-car, its special gift for indonesia," *ITS News*, Aug 2020. [Online]. Available: <https://www.its.ac.id/news/en/2020/08/18/i-car-its-special-gift-for-indonesia/>