

## IMPLEMENTASI *DEEP LEARNING* PADA SIMULASI *AUTONOMOUS DRIVE* MENGUNAKAN *AIRSIM*

Violla Gunova<sup>\*1)</sup>

Sekolah Teknik Elektro dan Informatika, Program Studi Teknologi Media Digital dan Game  
Institut Teknologi Bandung  
Bandung, Indonesia  
viollagunova123@gmail.com<sup>1</sup>

### Abstract

*Autonomous Driving is one of the technologies available in Artificial Intelligence. This technology aims to facilitate human work related to driving. The application of this technology can be done by using one of the main capabilities found in Artificial Intelligence, namely Machine Learning. By using several methods found in Machine Learning, Autonomous Driving Technology can be formed and developed. This paper will discuss how to make Autonomous Driving technology using available methods and show the results obtained from using these methods.*

*Keywords: Autonomous Driving, Machine Learning.*

### Abstrak

*Autonomous Driving* merupakan salah satu teknologi yang ada pada Artificial Intelligence. Teknologi ini memiliki tujuan untuk mempermudah pekerjaan manusia yang berhubungan mengemudi. Penerapan teknologi ini dapat dilakukan dengan menggunakan salah satu kemampuan utama yang terdapat pada Artificial Intelligence, yaitu Machine Learning. Dengan menggunakan beberapa metode yang terdapat pada Machine Learning, Teknologi *Autonomous Driving* dapat dibentuk dan dikembangkan. Paper ini akan membahas bagaimana membuat teknologi *Autonomous Driving* dengan menggunakan metode yang tersedia serta menunjukkan hasil yang didapat dari penggunaan metode tersebut.

*Kata kunci: Autonomous Driving, Machine Learning.*

### 1. Pendahuluan

Seiring dengan perkembangan zaman, teknologi yang dimiliki oleh manusia juga semakin berkembang. Tujuan utama dari pengembangan teknologi yang sudah ada ialah untuk semakin mempermudah manusia untuk menjalani kehidupannya. *Autonomous Drive* merupakan salah satu contoh dari pengembangan teknologi yang sudah ada. Teknologi ini membuat sebuah mobil mampu melihat dan merasakan lingkungan disekitarnya serta menjalankan dirinya sendiri secara benar dan aman dengan sedikit atau tanpa kontrol oleh pengemudi [1], [2]. Dengan kata lain, keberadaan teknologi ini memungkinkan orang-orang yang tidak atau belum bisa mengemudi mobil untuk berpergian dengan mobil yang mereka miliki.

Teknologi *Autonomous Drive* dibentuk dengan cara menanamkan kecerdasan buatan atau *Artificial Intelligence* (AI) kedalam mobil. Fungsi AI yang ditanam mirip dengan otak manusia, dimana AI yang ditanam ini akan menentukan tindakan mobil sesuai dengan input yang diterima, baik yang berasal dari lingkungan disekitarnya atau yang berasal dari pengemudinya. AI sendiri sebenarnya memiliki beberapa kemampuan, namun kemampuan AI yang akan diterapkan pada teknologi *Autonomous Drive* ini ialah *Machine Learning* (ML) dimana mesin akan berkembang melalui penggunaan data dan pengalaman yang dilalui oleh mesin [3].

Didalam ML sendiri terdapat berbagai metode dan algoritma. Pada *Autonomous Drive* ini, pendekatan metode yang akan digunakan adalah *Deep Learning* (DL) yang menggunakan algoritma *Artificial Neural Network* (ANN). Hal ini disebabkan karena teknologi *Autonomous Drive* memiliki ukuran data yang cukup besar serta jumlah data yang cukup banyak dalam pengaplikasiannya. Disisi lain, metode DL yang menggunakan algoritma ANN memiliki kapabilitas untuk mengolah data yang memiliki ukuran yang besar serta jumlah yang banyak. Ini bisa terjadi karena algoritma ANN memiliki struktur yang mirip dengan otak manusia dimana neuron yang terdapat pada otak manusia diganti dengan neuron buatan yang berbentuk node yang memiliki input dan output yang berfungsi untuk mengolah data dan menghasilkan pekerjaan yang akan dilakukan. Struktur algoritma seperti ini memungkinkan untuk mengolah data yang jumlahnya sangat banyak sehingga metode DL cocok untuk digunakan dalam pengaplikasian *Autonomous Driving*.

#### A. *Autonomous Drive*

*Autonomous Drive*, atau bisa juga disebut *autonomous vehicle*, *self-driving car*, *driverless car*, atau *robocar* merupakan sebuah kendaraan yang mampu melihat atau merasakan lingkungan disekitarnya dan

bergerak sesuai dengan apa yang dirasakan atau dilihat oleh kendaraan tersebut dengan sedikit atau tanpa bantuan dari pengemudi. *Autonomous Drive* merupakan gabungan dari berbagai tipe sensor untuk merasakan keadaan sekitarnya, seperti radar, lidar, sonar, GPS, odometri, dan inertial measurement units, serta *control-system* tingkat lanjut yang memiliki berbagai informasi yang dapat digunakan oleh kendaraan untuk perjalanan secara otomatis [1], [4].

Terdapat 6 tingkatan dalam *Autonomous Drive* yang mana setiap tingkatan dibagi sesuai dengan seberapa banyak sistem yang dikendalikan otomatis oleh mobilnya sendiri. 6 tingkatan itu terdiri dari level 0, level 1, level 2, level 3, level 4, dan level 5 [5], [6], [7]. Klasifikasi ini dibuat oleh persatuan insinyur automotif atau Society of Automotive Engineers International (SAE International), dimana penjelasan dari setiap klasifikasinya dapat dilihat pada Tabel I.

Tabel I Level *Autonomous Drive* berdasarkan SAE

Level SAE	Nama	Penjelasan		Pengendali Setir dan Kecepatan	Pengawasan terhadap Lingkungan Sekitar	Pengendali Kendaraan terhadap Tugas Mengemudi yang Mendadak	Mode Mengemudi (Kapabilitas Sistem)
<i>Human driver monitors the driving environment</i>							
0	No Automation (All On)	Setiap tugas yang ada dalam menyetir dilakukan oleh manusia sekalipun sudah ada sensor atau sistem pemberitahuan		Pengemudi			Tidak Ada
1	Driver Assistance (Hands On)	Salah satu kontrol dasar dalam mengemudi (setir atau kecepatan) dilakukan oleh sistem	Pengontrolan dilakukan dengan menggunakan informasi yang didapat oleh mobil dengan catatan untuk tugas	Sistem dan Pengemudi			Beberapa Mode Mengemudi
2	Partial Automation (Hands Off)	Semua kontrol dasar dalam mengemudi (setir dan kecepatan) dilakukan oleh sistem	pengemudian mendadak ( <i>Dynamic Driving Task</i> ) masih dilakukan oleh pengemudi	Sistem	Pengemudi	Pengemudi	
<i>Automated driving system monitors the driving environment</i>							
3	Conditional Automation (Eyes Off)	Semua kontrol dalam mengemudi baik yang dasar maupun yang tidak (termasuk aspek mengemudi mendadak) dikendalikan oleh sistem		Dengan catatan pengemudi harus siap siaga untuk mengambil alih kemudi jika terjadi suatu kejadian yang tak terduga		Pengemudi	Beberapa Mode Mengemudi
4	High Automatin (Mind Off)			Pengemudi tidak harus siap siaga untuk mengambil alih kemudi karena mobil sudah diarahkan sistem untuk menjalankan tindak pencegahan	Sistem	Sistem	Hampir Semua Mode Mengemudi

Level SAE	Nama	Penjelasan		Pengendali Setir dan Kecepatan	Pengawasan terhadap Lingkungan Sekitar	Pengendali Kendaraan terhadap Tugas Mengemudi yang Mendadak	Mode Mengemudi (Kapabilitas Sistem)
5	Full Automation (Steering Wheel Optional)	Semua kontrol dalam mengemudi baik yang dasar maupun yang tidak (termasuk aspek mengemudi mendadak) dikendalikan oleh sistem	Pengendalian mobil sudah pada tahapan pilihan pengemudi, apakah pengemudi yang menyetir atau tidak	Sistem	Sistem	Sistem	Semua Mode Mengemudi

**B. Artificial Intelligence**

Artificial Intelligence (AI) adalah kecerdasan yang dimiliki oleh mesin. Kecerdasan ini mirip dengan kecerdasan yang dimiliki oleh manusia, dimana AI membuat mesin memiliki kemampuan untuk memahami serta mengambil keputusan berdasarkan lingkungan sekitarnya [8]. Tujuan utama dari pembuatan AI adalah untuk membuat mesin memiliki fungsi yang memiliki kriteria-kriteria kecerdasan didalamnya sehingga mesin tersebut mampu melakukan pekerjaan manusia yang lebih kompleks, tergantung dari tingkat kecerdasan AI yang digunakan. Dalam penerapannya, terdapat 6 kemampuan utama yang dapat diklasifikasikan sebagai AI [9]. Kemampuan tersebut antara lain :

- Representasi Informasi/Pengetahuan (*Knowledge Representation*)
- Perencanaan (*Planning*)
- Persepsi (*Perception*)
- *Machine Learning*
- Pemahaman Bahasa (*Natural Language Processing*)
- *Robotics*

Meskipun 6 kemampuan diatas memiliki fungsi yang berbeda-beda, kemampuan-kemampuan tersebut saling berhubungan satu sama lain saat pengaplikasiannya. Bahkan hubungan antar kemampuan utama diatas akan membentuk kemampuan baru.

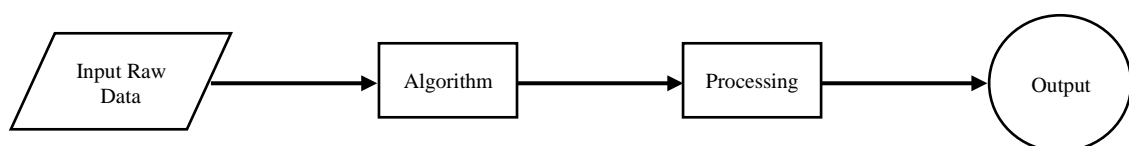
**C. Machine Learning**

*Machine Learning* (ML) merupakan salah satu aspek atau kemampuan dalam AI. Aspek ini terinspirasi dari salah satu cara kerja otak manusia yaitu belajar (*learning*). ML membuat mesin atau komputer belajar melalui berbagai input dataset dan pengalaman yang dialami oleh mesin. Pembelajaran ini akan dilakukan secara terus-menerus oleh mesin hingga mesin mencapai level kecerdasan tertentu.

Dalam ML, terdapat beberapa metode atau pendekatan yang dapat digunakan mesin untuk melakukan pembelajaran. Namun secara garis besar, terdapat 3 metode *Machine Learning* yang dipisahkan berdasarkan tipe input atau dataset dan cara pelatihannya. Metode-metode tersebut antara lain :

- *Unsupervised Learning* (UL)

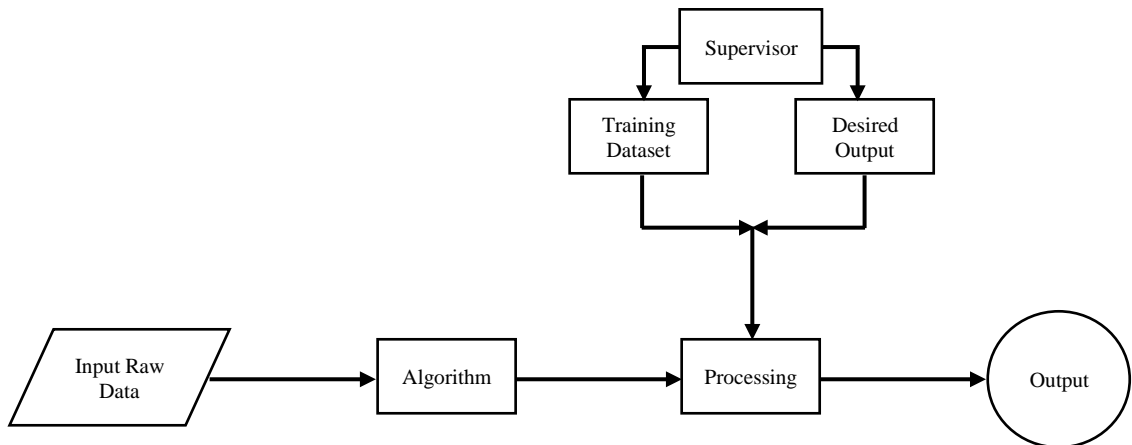
*Unsupervised Learning* (UL) merupakan sebuah metode ML dimana metode ini hanya memberikan data input saja kedalam mesin. Setelah menerima input, mesin kemudian akan memberikan output berdasarkan pola data input yang diterima. Metode ini cocok digunakan untuk penyelesaian masalah pengelompokkan data, baik itu *association* maupun *clustering*.



Gambar 1 Cara Kerja *Unsupervised Learning* (UL)

- *Supervised Learning (SL)*

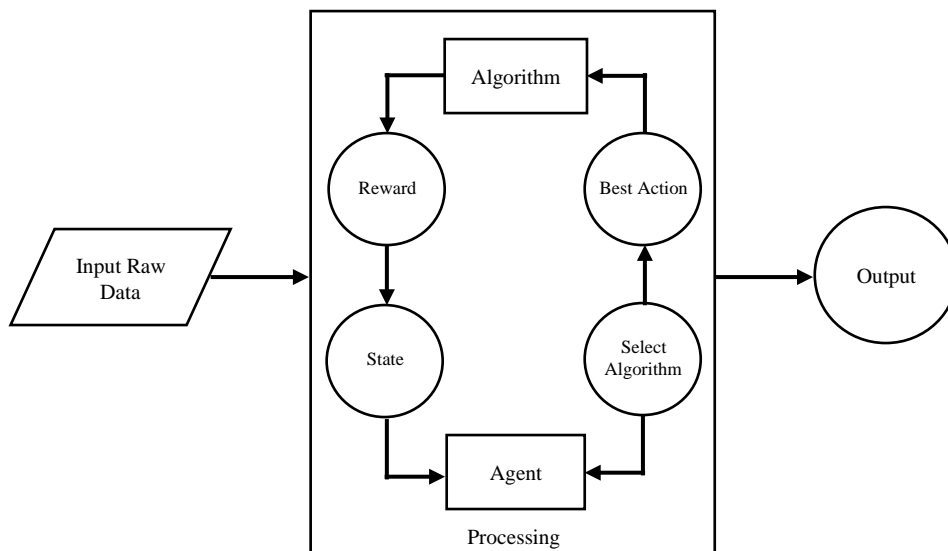
*Supervised Learning (SL)* merupakan sebuah metode ML dimana metode ini memberikan dataset yang berlabel dan data input kedalam mesin. Maksud dari dataset yang berlabel ini ialah untuk setiap tipe/bentuk dataset yang telah diberikan, outputnya telah ditentukan. Setelah menerima input, mesin kemudian akan memberikan output berdasarkan dataset yang dilabeli. Metode ini cocok digunakan untuk penyelesaian masalah *classification* dan *regression*.



Gambar 2 Cara Kerja *Supervised Learning (SL)*

- *Reinforcement Learning (RL)*

*Reinforcement Learning (RL)* merupakan sebuah metode ML dimana metode ini membuat mesin akan mencoba segala macam kemungkinan output yang ada berdasarkan dari input yang diterima dan keadaan lingkungan disekitar mesin. Setiap output yang dihasilkan oleh mesin akan diberikan *reward* atau *penalty* berdasarkan kedekatan output yang diinginkan dengan yang dihasilkan. Metode ini cocok digunakan untuk penyelesaian masalah yang menggunakan sistem *reward-based*.

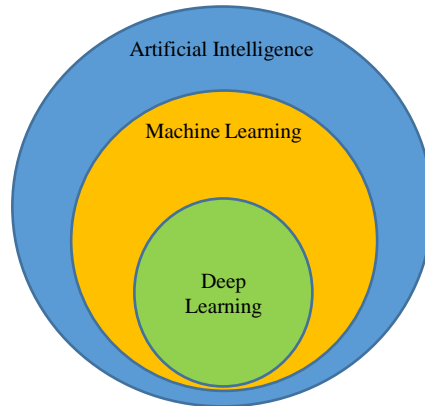


Gambar 3 Cara Kerja *Reinforcement Learning (RL)*

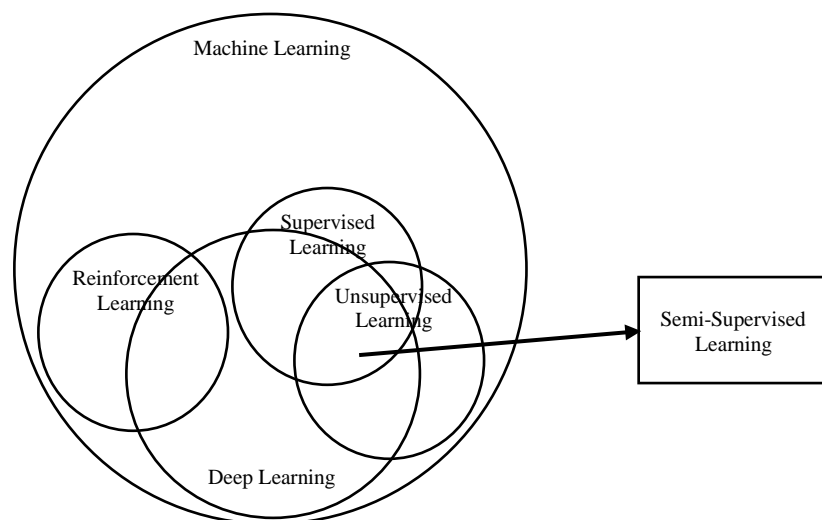
D. *Deep Learning*

Setiap teknik pembelajaran dalam ML yang menggunakan algoritma *Artificial Neural Network (ANN)* disebut dengan metode *Deep Learning (DL)*. Cara kerja metode ini adalah dengan menggunakan data-data yang sudah tersimpan diberbagai layer network yang ada di ANN, kemudian membentuk output atau informasi

sebelum ada input (pre-processing step), setelah itu membuat prediksi output atau keputusan berdasarkan input yang sudah diterima. Meskipun DL merupakan metode yang berbeda dibandingkan dengan metode-metode ML lainnya, namun dalam penerapannya beberapa metode saling tumpang tindih dengan metode lainnya.



Gambar 4 Hubungan antara AI, ML dan DL



Gambar 5 Salah Satu Hubungan antar metode yang dalam ML (UL, SL, RL, dan DL)

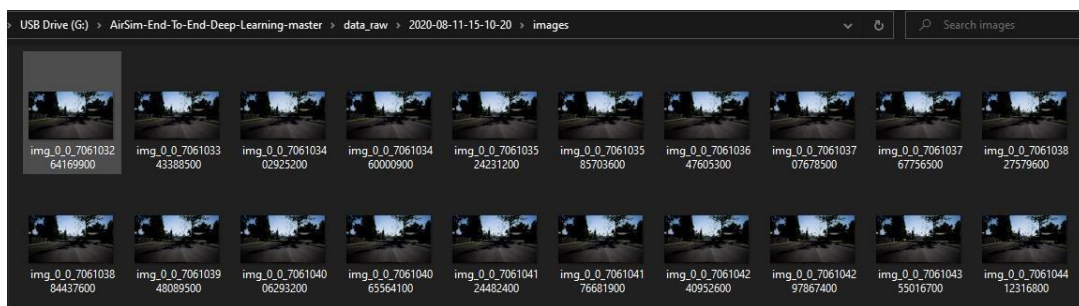
## 2. Metode Penelitian

Pada paper ini, teknologi *Autonomous Driving* akan dibentuk dengan menggunakan bantuan AirSim yang merupakan *software* untuk simulasi mengemudi mobil. Penggunaan *software* ini dilakukan dengan tujuan untuk meminimalisir biaya yang terjadi saat terjadi kesalahan atau miscalculated pada output yang dihasilkan. Untuk tahap awal, simulasi akan dilakukan pada lintasan yang sudah tersedia yang berbentuk kotak dengan mobil yang terparkir disetiap sisi jalan kecuali mobil pengguna. Penerapan teknologi *Autonomous Driving* ini akan dilakukan dengan menggunakan metode DL dikarenakan jumlah dataset yang diolah sangat banyak. Sedangkan untuk cara pelatihannya, metode yang bisa digunakan hanyalah metode SL dan RL, sedangkan metode UL tidak dapat digunakan untuk teknologi ini. Alasan kenapa metode UL tidak dapat digunakan ialah dikarenakan metode ini tidak bisa mengirim data yang dibutuhkan pada teknologi *Autonomous Driving*.



Gambar 6 Simulasi yang Terdapat pada AirSim

Baik metode UL maupun metode SL, keduanya memiliki kemiripan pada data input dan proses pelatihnannya. Perbedaannya terdapat pada output data yang dihasilkan. Bila diterapkan pada teknologi *Autonomous Driving*, kedua metode ini sama-sama mendapatkan input berupa rekaman video mobil yang dikemudikan secara manual. Rekaman tersebut diambil dari sudut pandang mobil terhadap lingkungan sekitarnya dan akan menjadi data mentah untuk pelatihan teknologi ini. Data mentah yang akan disimpan terbagi menjadi 2, yaitu *Data Images* yang merupakan gambar per *frame* dari video rekaman yang sudah diambil, dan *Data Log* yang berbentuk tabel dimana isinya berupa informasi-informasi dari setiap gambar yang ada pada *Data Images*, yaitu nama *file*, *speed*, *steering*, *throttle*, dan *brakes*. Perbedaan muncul setelah output pelatihan telah dihasilkan. Pada metode SL, output pelatihan berupa nilai *speed*, *steering*, *throttle*, dan *brakes* untuk setiap *frame*. Informasi dari output inilah yang nantinya akan diterima oleh teknologi *Autonomous Driving* dimana mobil akan berjalan sesuai dengan output yang diterima. Sedangkan pada metode UL, output yang akan diberikan bukanlah data-data untuk setiap *frame*, akan tetapi data yang diterima akan berupa data-data *images* yang dikelompokkan sesuai kemiripannya. Hal ini disebabkan karena dalam metode UL, tidak adanya pengawas atau *supervisor* yang memberitahu ke mesin nilai-nilai output yang harus dihasilkan untuk setiap jenis gambar yang serupa sehingga metode ini hanya akan menghasilkan pengelompokkan gambar yang tak memiliki informasi yang dibutuhkan oleh *Autonomous Driving*. Oleh karena itulah metode ini tidak bisa digunakan untuk teknologi ini.



Gambar 7 Contoh Data Image

G:\AirSim-End-To-End-Deep-Learning-master\data\_raw\2020-08-11-15-10-20\airsim\_rec.txt - Notepad++

1	Timestamp	Speed (kmph)	Throttle	Steering	Brake	Gear	ImageName
2	706103270	0	0.000000	0.000000	0.000000	N	img_0_0_706103264169900.png
3	706103326	0	0.000000	0.000000	0.000000	N	img_0_0_706103343388500.png
4	706103385	0	0.000000	0.000000	0.000000	N	img_0_0_706103402925200.png
5	706103443	0	0.000000	0.000000	0.000000	N	img_0_0_706103460000900.png
6	706103508	0	0.000000	0.000000	0.000000	N	img_0_0_706103524231200.png
7	706103568	0	0.000000	0.000000	0.000000	N	img_0_0_706103585703600.png
8	706103630	0	0.000000	0.000000	0.000000	N	img_0_0_706103647605300.png
9	706103693	0	0.000000	0.000000	0.000000	N	img_0_0_706103707678500.png
10	706103753	0	0.000000	0.000000	0.000000	N	img_0_0_706103767756500.png

Gambar 7 Contoh DataLog

Metode RL memiliki cara yang berbeda, dimana pada metode ini kita harus menentukan terlebih dahulu lintasan yang harus dilalui oleh mobil, kemudian mobil akan dibiarkan belajar sendiri untuk menempuh lintasan yang sudah

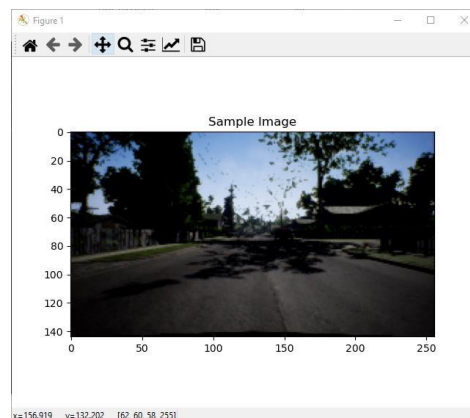
ditentukan. Setiap mobil menabrak atau keluar dari jalur lintasan akan mengaktifkan *trigger* yang membuat mobil harus menempuh lagi dari titik semula. Meskipun metode ini akan menghasilkan output yang bagus, metode ini memiliki kelemahan dimana waktu yang diperlukan untuk melatih mobil hingga bisa melaju sesuai lintasan yang ditentukan cukup lama. Hal ini disebabkan karena metode RL ini membuat mesin mencoba segala kemungkinan yang tersedia. Sebagai contoh, jika kita membuat lintasan berupa maju kedepan sepanjang 50 meter, maka mobil akan mencoba setiap kemungkinan yang ada, seperti jalan zig-zag hingga titik tujuan, maju 10 meter kemudian berputar 1 kali sebelum maju hingga tujuan, dan sebagainya. Oleh karena itu, metode yang akan digunakan pada paper ini ialah metode SL.

Metode SL memiliki kelebihan dimana waktu pelatihan yang dibutuhkan untuk membuat mobil melaju sesuai dengan lintasan yang ditentukan cukup sebentar. Hal ini disebabkan karena metode ini telah memberikan terlebih dahulu contoh mengemudi untuk menempuh lintasan yang sudah ditentukan sehingga mesin hanya perlu bergerak sesuai dengan contoh yang diberikan. Hanya saja, metode ini memiliki beberapa kelemahan dimana jumlah data rekaman yang menjadi input harus lebih banyak agar pergerakan mobil semakin akurat. Selain itu, parameterparameter yang terdapat pada saat *training* data juga harus diubah-ubah untuk mendapatkan hasil yang lebih bagus. Meskipun begitu, metode SL ini cocok digunakan saat baru pertama kali membuat teknologi *Autonomous Driving*.

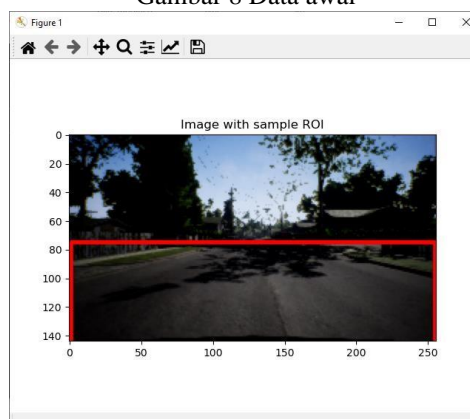
Seperti yang sudah dijelaskan sebelumnya bahwa data input untuk metode SL berupa *Data Images* dan *Datalog*. Setelah mendapatkan data input ini, data kemudian akan diolah terlebih dahulu sebelum ditraining. Setelah data mentah didapat, data kemudian diolah terlebih dahulu sebelum ditraining. Data olahan ini didapat dengan cara memberi label pada setiap masing-masing gambar sesuai dengan label yang ada pada data log. Setelah data diolah, kemudian data dilatih dengan mengubah 3 parameter yang mempengaruhi output pelatihan, yaitu

:

- ROI = Merupakan batas area yang akan diambil dari data images. Area yang diambil ini yang akan digunakan untuk pelatihan dengan tujuan memperingan pelatihan dan membuat mobil tidak terlalu kebingungan dengan obyek lain selain jalan. Pengubahan batas ini juga dilakukan pada saat pengolahan data.



Gambar 8 Data awal



Gambar 9 Data dengan batas ROI

- Brighten-Range = Tingkat kecerahan untuk setiap gambar yang akan dilatih □
- Zero\_Drop\_Percentage = Presentasi data yang berlabel 0 yang akan dihilangkan.



Data yang sudah dilatih membentuk model output. Output dari proses ini adalah berupa nilai-nilai seperti parameter yang terdapat pada *DataLog* untuk setiap data raw yang diproses oleh sistem. Pada pengujian kali ini, nilai output hanya terdiri dari angle dan throttle. Nilai angle dan throttle yang didapat kemudian dikirim ke AirSim dimana program akan menjalankan mobil sesuai dengan nilai yang dikirim.

```

    Anaconda Prompt (Anaconda3) - conda deactivate - conda deactivate - python autocontrol.py
    Sending steering = -4.67, throttle = 1.0
    Sending steering = -5.0, throttle = 1.0
    Sending steering = -4.88, throttle = 1.0
    Sending steering = -5.07, throttle = 1.0
    Sending steering = -5.27, throttle = 1.0
    Sending steering = -4.82, throttle = 1.0
    Sending steering = -2.86, throttle = 1.0
    Sending steering = -0.69, throttle = 1.0
    Sending steering = -0.5, throttle = 1.0
    Sending steering = -0.27, throttle = 1.0
    Sending steering = -0.33, throttle = 1.0
    Sending steering = -0.21, throttle = 1.0
    Sending steering = -0.64, throttle = 1.0
    Sending steering = 0.12, throttle = 1.0
    
```

Gambar 11 Nilai steering angle dan throttle yang dikirim

### 3. Hasil dan Pembahasan

Pada bagian ini akan disajikan data hasil pengujian, analisis, dan evaluasi untuk metode SL dimana pengujian akan dilakukan dengan mengubah tiga parameter pada tahap pelatihan (ROI, Brightness, Zero\_Drop) yang kemudian akan dievaluasi berdasarkan waktu training dan seberapa jauh mobil bisa berjalan. Selain itu, pengujian juga dilakukan dengan mengubah jumlah rekaman yang dijadikan referensi yang digunakan oleh mesin saat melakukan pelatihan. Hasil pelatihan dapat dilihat pada Tabel II dan III.

Tabel II pengujian dengan menggunakan 1 rekaman sebagai data pengujian

Training			Output		Catatan
ROI	Brightness	Zero_Drop	Time Training	Mobil yang terlewati	
75,150,0,255	0,9	75%	06:05,5	0	
75,150,0,255	0,8	75%	06:05,7	0	
75,150,0,255	0,7	75%	06:04,8	0	Nabrak pas dimobil
75,150,0,255	0,6	75%	06:04,1	1	
75,150,0,255	0,5	75%	06:05,2	1	Nabrak diantara mobil 1 dan 2 (Waktu sebelum nabrak = 00:13,9)
75,150,0,255	0,4	75%	06:04,1	1	
75,150,0,255	0,3	75%	06:03,5	1	Nabrak didepan mobil
75,150,0,255	0,2	75%	06:04,2	0	
75,150,0,255	0,1	75%	06:04,0	0	
75,150,0,255	0,9	50%	07:07,8	0	
75,150,0,255	0,8	50%	07:08,5	0	
75,150,0,255	0,7	50%	07:08,3	0	Nabrak pas dimobil
75,150,0,255	0,6	50%	07:07,5	1	
75,150,0,255	0,5	50%	07:06,7	1	
75,150,0,255	0,4	50%	07:07,2	2	Nabrak didepan mobil (Waktu sebelum nabrak = 00:16,6)



75,150,0,255	0,3	50%	07:06,8	1	
75,150,0,255	0,2	50%	07:06,3	0	
75,150,0,255	0,1	50%	07:06,1	0	
76,135,0,255	0,9	95%	03:31,6	0	
76,135,0,255	0,8	95%	03:32,3	0	
76,135,0,255	0,7	95%	03:31,7	0	
76,135,0,255	0,6	95%	03:31,6	1	Nabrak didepan mobil
76,135,0,255	0,5	95%	03:31,3	1	Nabrak agak jauh sedikit (Waktu sebelum nabrak = 00:12,4)
76,135,0,255	0,4	95%	03:31,3	0	Nabrak dibelakang mobil
76,135,0,255	0,3	95%	03:30,8	0	
76,135,0,255	0,2	95%	03:30,7	0	
76,135,0,255	0,1	95%	03:30,8	0	
76,135,0,255	0,9	70%	04:15,3	0	
76,135,0,255	0,8	70%	04:15,8	0	
76,135,0,255	0,7	70%	04:14,9	0	Nabrak dibelakang mobil
76,135,0,255	0,6	70%	04:14,6	1	
76,135,0,255	0,5	70%	04:14,5	1	
76,135,0,255	0,4	70%	04:14,3	1	Nabrak dibelakang mobil 2 (Waktu sebelum nabrak = 00:15,2)
76,135,0,255	0,3	70%	04:13,8	1	
76,135,0,255	0,2	70%	04:12,9	0	
76,135,0,255	0,1	70%	04:12,6	0	

Tabel III pengujian dengan menggunakan 2 rekaman sebagai data pengujian

Training			Output		Catatan
ROI	Brightness	Zero_Drop	Time Training	Mobil yang terlewati	
76,135,0,251	0,9	90%	06:57,2	0	
76,135,0,252	0,8	90%	06:56,7	0	
76,135,0,253	0,7	90%	06:56,1	0	
76,135,0,254	0,6	90%	06:56,0	1	
76,135,0,255	0,5	90%	06:55,9	2	Nabrak didepan mobil (Waktu sebelum nabrak = 00:17,0)
76,135,0,256	0,4	90%	06:55,6	1	
76,135,0,257	0,3	90%	06:55,7	1	

76,135,0,258	0,2	90%	06:55,4	0	
76,135,0,259	0,1	90%	06:55,1	0	

Berdasarkan tabel diatas, dapat dilihat bahwa setiap perubahan nilai parameter pelatihan, yaitu ROI, Brightness, dan Zero\_Drop, akan mempengaruhi nilai outputnya, baik itu waktu pelatihannya maupun jarak tempuhnya yang diwakili seberapa banyak mobil yang terparkir disisi jalan yang terlewati dipinggir jalan. Berdasarkan tabel III, dapat dilihat pula bahwa pengujian dengan jumlah rekaman yang lebih banyak akan menghasilkan output yang lebih baik dibanding dengan rekaman yang lebih sedikit.

#### 4. Kesimpulan

Hasil dari penelitian ini ialah metode SL dapat digunakan untuk membuat teknologi *Autonomous Driving* yang tidak perlu memakan waktu cukup lama dalam pelatihannya. Akan tetapi metode ini membutuhkan perubahan nilai pada parameter pelatihannya serta membutuhkan jumlah referensi rekaman yang lebih banyak untuk mendapatkan hasil yang lebih baik. Metode ini bagus untuk diimplementasikan pada teknologi *Autonomous Driving* saat baru akan membuatnya, namun jika tujuannya untuk mendapatkan hasil mengemudi yang mendekati sempurna, maka teknologi ini membutuhkan rekaman yang lebih banyak serta perubahan nilai parameter pelatihannya.

#### 5. Daftar Rujukan

- [1] Hu, Junyan; et, al (2020). "Cooperative control of heterogeneous connected vehicle platoons: An adaptive leader-following approach". *IEEE Robotics and Automation Letters*. **5** (2): 977–984.
- [2] Gehrig, Stefan K.; Stein, Fridtjof J. (1999). "Dead reckoning and cartography using stereo vision for an automated car". *IEEE/RSJ International Conference on Intelligent Robots and Systems*. **3**. Kyongju. pp. 1507–1512.
- [3] Mitchell, Tom. "Machine Learning". New York: McGraw Hill, USA, 1997
- [4] Lassa, Todd (January 2013). "The Beginning of the End of Driving". *Motor Trend*. Retrieved 1 September 2014.
- [5] "Adaptive system classification and glossary on Automated driving" (PDF). Archived from the original (PDF) on 7 October 2017. Retrieved 11 September 2017.
- [6] "Automated Driving Levels of Driving Automation are Defined in New SAE International Standard J3016" (PDF). 2017. Archived from the original(PDF) on 20 November 2016.
- [7] "Automated Driving – Levels of Driving Automation are Defined in New SAE International Standard J3016" (PDF). *SAE International*. 2014. Archived (PDF) from the original on 1 July 2018.
- [8] Aleksander, Igor (1995). "Artificial Neuroconsciousness: An Update". IWANN. Archived from the original on 2 March 1997. BibTex Archived 2 March 1997 at the Wayback Machine.
- [9] Nilsson, Nils. "Artificial Intelligence: A New Synthesis", 1998