

Deteksi Ruang Kosong pada Jalan Menggunakan *Semantic Segmentation* pada Mobil Otonom

Agil Bintang Pratama, Rusdhianto Effendi Abdul Kadir, dan Mohamad Abdul Hady
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: ditto@ee.its.ac.id

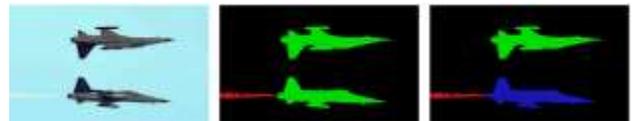
Abstrak—Perkembangan mobil otonom tentunya tidak terlepas dari masalah navigasi, dimana navigasi untuk menentukan apakah jalan di depannya aman dan bebas dari halangan. Beberapa masalah pada mobil otonom yang diteliti belakangan ini seperti deteksi halangan dan deteksi garis jalan hanya melakukan pendeteksian dan pengenalan objek namun tidak memberikan informasi mengenai jalan yang dapat dilewati. Oleh karena itu, diperlukan deteksi ruang kosong pada jalan yang aman dari halangan agar tidak terjadi tabrakan. Pada umumnya deteksi ruang kosong jalan yang ada masih menggunakan sensor kamera saja, sehingga hasil yang didapat kurang baik. Pada penelitian ini, metode *Semantic Segmentation* dengan menggunakan data *fusion* antara sensor LiDAR dan kamera diterapkan untuk menghasilkan suatu sistem yang dapat mendeteksi ruang kosong jalan. Algoritma yang digunakan pada sistem ini adalah SNE-RoadSeg. Deteksi ruang kosong yang dihasilkan adalah berupa segmentasi pada jalan untuk memberikan informasi pengenalan dan klasifikasi jalan di tingkat piksel. Performa sistem kemudian diuji menggunakan dataset jalan yang berasal dari KITTI pada kelima jenis *backbone* yaitu, ResNet-18, ResNet-34, ResNet-50, ResNet-101, dan ResNet-152. Algoritma sistem dapat menghasilkan skor *F1 score* terkecil dengan nilai 0.9548 pada ResNet-18 dan skor *F1 score* terbesar dengan nilai 0.9675 untuk ResNet-101. Skor *precision* ResNet-152 dengan nilai 0.9686 lebih tinggi 0.0024 dibandingkan dengan ResNet-101. Sistem yang diajukan memiliki rata-rata kecepatan deteksi pada tiap jenis ResNet sekitar 0.343 detik.

Kata Kunci—Mobil Otonom, ResNet, Ruang kosong, *Semantic Segmentation*, SNE-RoadSeg.

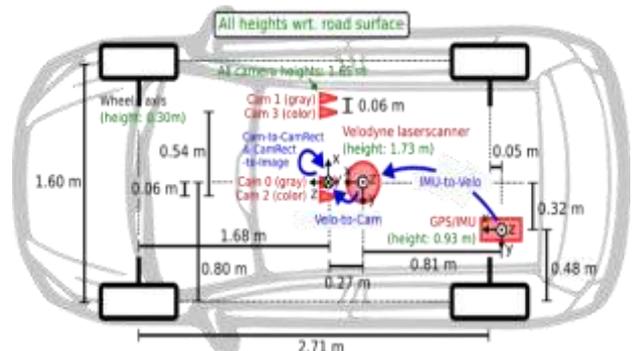
I. PENDAHULUAN

PADA jaman yang maju dan berkembang seperti saat ini, transportasi merupakan susunan penting dalam keberlangsungan hidup manusia. Oleh karenanya jumlah kendaraan lalu lintas yang beroperasi meningkat secara pesat hingga saat ini. Peningkatan jumlah kendaraan tersebut diikuti dengan naiknya angka kecelakaan lalu lintas. Salah satu alasan terjadinya kecelakaan lalu lintas karena kelalaian pengemudi. Didukung dengan berkembangnya teknologi dan akan memasuki revolusi industri 5.0 sehingga para peneliti mulai mengembangkan kendaraan otonom. Mobil otonom telah menjadi salah satu topik yang banyak diteliti dan dikembangkan oleh para civitas akademisi maupun industri. Berbagai penelitian mengenai bagian-bagian mobil otonom dilakukan dengan berbagai macam pendekatan. Beberapa pihak ketiga bahkan telah meluncurkan prototipe mobil otonomnya yang dapat digunakan seperti Tesla, Waymo, Voyage, Uber, AutoX, dan masih banyak lagi.

Mobil otonom merupakan kendaraan yang dapat mengenali keadaan sekitarnya dan beroperasi tanpa kendali manusia. Mobil otonom dapat pergi kemana pun seperti mobil pada umumnya dan dapat melakukan hal yang dilakukan pengemudi handal sekalipun. National Highway Traffic Safety Administration (NHTSA) Amerika Serikat, mengklasifikasikan mobil otonom kedalam empat kategori yaitu, *function specific automation*, *combined function*



Gambar 1. Sampel gambar, semantic segmentation, instance segmentation, dari kiri ke kanan.



Gambar 2. Pengaturan tata letak yang digunakan untuk pengambilan dataset KITTI.

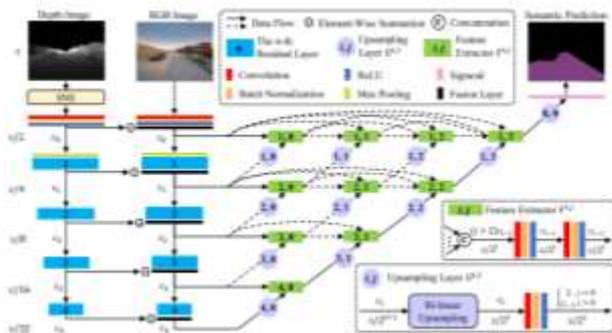


Gambar 3. Gambar RGB, ground truth, dan depth pada dataset KITTI berurutan dari atas ke bawah.

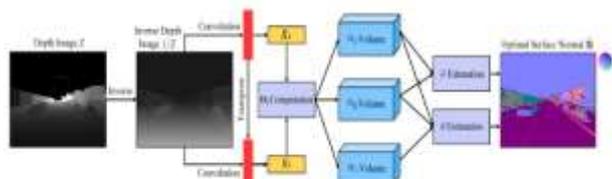
automation, *limited self-driving automation*, dan *fully self-driving automation* [1]. Terdapat empat fokus utama pada mobil otonom, yaitu persepsi, *motion planning*, navigasi, dan *behavior*.

Persepsi merupakan kemampuan mobil otonom untuk mengenali dan memahami lingkungannya. Hal ini akan membantu mobil otonom untuk menghindari terjadinya tabrakan dan juga untuk mengetahui berbagai macam rintangan yang ada di jalan. Dalam menjalankan fungsi persepsi, mobil otonom dibantu oleh beberapa sensor, seperti Light Detection and Ranging (LiDAR), Kamera, Radar, GPS, dan lainnya. LiDAR dan kamera merupakan salah satu sensor penting pada mobil otonom dimana LiDAR berfungsi untuk mendeteksi lingkungan sekitar dan kamera berfungsi untuk melihat.

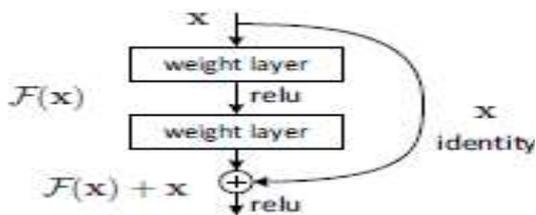
Perkembangan mobil otonom tentunya tidak terlepas dari masalah navigasi, dimana navigasi untuk menentukan apakah



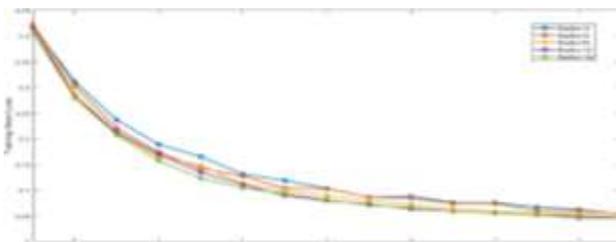
Gambar 4. Arsitektur SNE-RoadSeg.



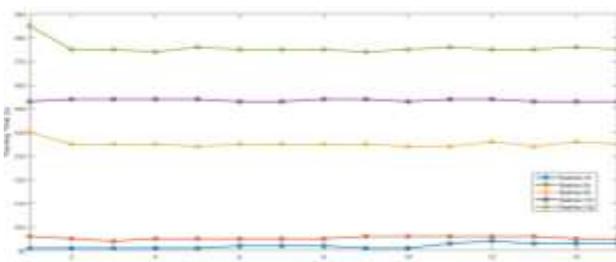
Gambar 5. Arsitektur modul SNE.



Gambar 6. Diagram blok ResNet.



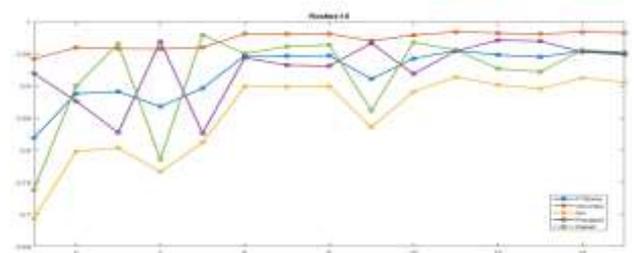
Gambar 7. Grafik loss training pada dataset KITTI.



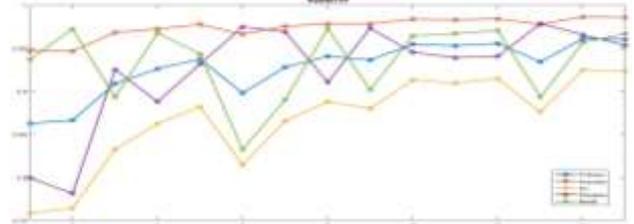
Gambar 8. Durasi training tiap epoch pada dataset KITTI.

jalan di depannya aman dan bebas dari halangan. Beberapa masalah seperti deteksi garis jalan seperti pada penelitian dan deteksi halangan seperti yang dilakukan hanya sebatas melakukan pendeteksian dan pengenalan objek [2-3]. Untuk mengestimasi posisi dari garis jalan tanpa mengetahui apakah lajur diluar garis tersebut aman untuk menyeberang atau tidak [3]. Untuk itu dibutuhkan sebuah pengenalan dan pendeteksian dari ruang kosong pada jalan yang bebas dari rintangan dan halangan agar tidak terjadi tabrakan pada mobil otonom.

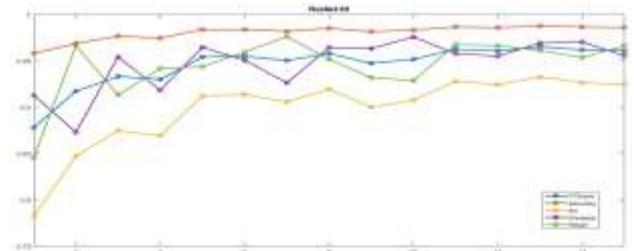
Beberapa tahun belakangan ini, telah terdapat beberapa penelitian mengenai deteksi ruang kosong jalan. Telah dapat menentukan estimasi free space yang direpresentasikan dalam curve batas jalan dan telah mampu melakukan deteksi pada beberapa lajur [4-5]. Namun dari kedua penelitian tersebut masih menggunakan data yang berasal dari kamera dimana



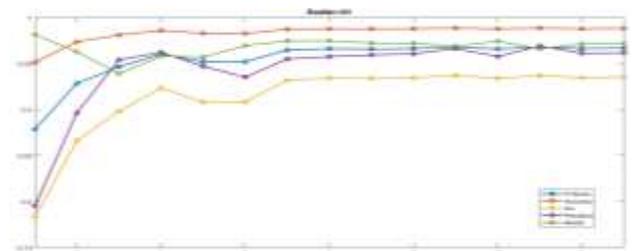
Gambar 9. Hasil evaluasi model untuk ResNet-18.



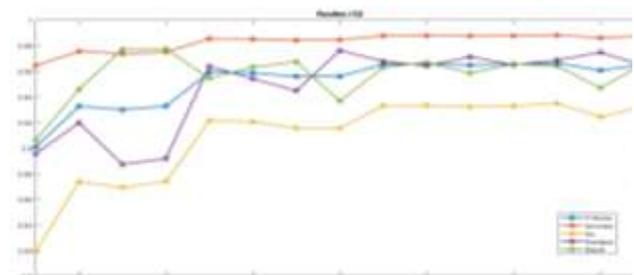
Gambar 10. Hasil evaluasi model untuk ResNet-34.



Gambar 11. Hasil evaluasi model untuk ResNet-50.



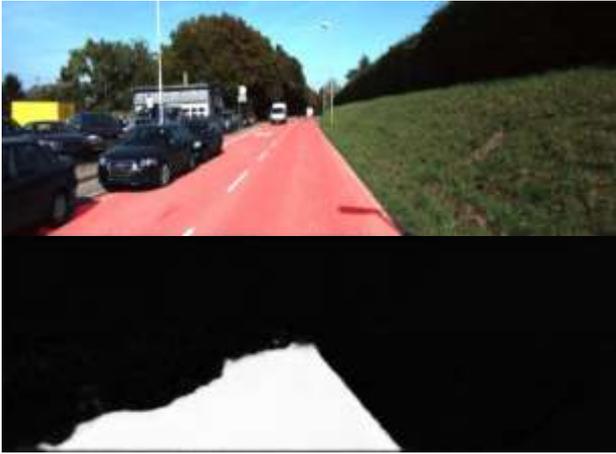
Gambar 12. Hasil evaluasi model untuk ResNet-101.



Gambar 13. Hasil evaluasi model untuk ResNet-152.

didapatkan hasil yang kurang maksimal. Pada penelitian ini, dilakukan deteksi ruang kosong jalan menggunakan fusion data LiDAR dan kamera untuk mendapatkan hasil deteksi yang lebih baik. Image segmentation belakangan ini menjadi salah satu topik hangat dalam bidang computer vision dan machine learning. Image segmentation merupakan proses mengklasifikasikan setiap pixel dalam gambar yang termasuk dalam suatu kelas tertentu. Image Segmentation merupakan salah satu langkah penting dalam membangun suatu sistem robot yang kompleks, salah satunya adalah mobil otonom. Terdapat 2 tipe image segmentation, yaitu instance segmentation dan semantic segmentation (Gambar 1) [6].

Semantic segmentation mengaitkan setiap pixel gambar dengan suatu kelas label tertentu. Yang membedakan antara kedua jenis *image segmentation* ini yaitu, pada *semantic*



Gambar 14. Hasil deteksi pada ResNet-152 berupa gambar prediksi dan gambar segmentasi berurutan dari atas ke bawah.



Gambar 15. Kegagalan deteksi pada ResNet-152.

segmentation, memperlakukan beberapa objek pada satu kelas yang sama sebagai satu entitas sedangkan, *instance segmentation* memperlakukan beberapa objek dari kelas yang sama sebagai entitas individu yang berbeda. Pada penelitian ini, deteksi ruang kosong pada jalan akan menggunakan metode *semantic segmentation* (Gambar 2).

Sejak munculnya Deep Neural Network (DNN), *semantic segmentation* telah membuat kemajuan yang luar biasa. Sebelum adanya DNN, berbagai fitur yang digunakan untuk *semantic segmentation* seperti Pixel Color, Histogram of Oriented Gradients (HOG), Scale-invariant feature transform (SIFT), dan lainnya [7]. Setelah adanya DNN, berbagai fitur baru yang dapat digunakan untuk *semantic segmentation* seperti Fully Convolutional Networks (FCN), Encoder-Decoder Architecture (U-Net), hingga pendekatan menggunakan Recurrent Neural Networks (RNNs) [6].

II. PERANCANGAN ARSITEKTUR

Bagian ini menjelaskan mengenai sistem yang dibangun seperti perancangan data *training*, serta arsitektur yang digunakan dalam penelitian ini.

A. Data Masukan (Training)

Data masukan atau dataset yang digunakan pada penelitian ini, yaitu dataset KITTI. Data masukan ini berasal dari dataset KITTI Road/Lane Detection Evaluation 2013. Dataset ini terdiri dari 289 data training dan 290 data tes yang mencakup tiga kategori adegan jalan yaitu jalan tanpa marka, jalan dengan marka, dan jalan dengan banyak marka [8]. Dataset ini diambil menggunakan kendaraan Volkswagen Passat B6 yang telah dimodifikasi seperti pada Gambar 2.

Data direkam menggunakan komputer delapan inti Intel Core-i7 yang dilengkapi dengan sistem RAID, menjalankan Ubuntu Linux, dan database realtime. Sensor yang digunakan untuk pengambilan data ini adalah sebagai berikut: 1 buah Inertial Navigation System (IMU): OXTS RT 3003, 1 buah

Tabel 1.
Spesifikasi GPU yang Digunakan

GPU Architecture	NVIDIA Pascal
NVIDIA CUDA Cores	3584
Double Precision Performance	5.3 TeraFLOPS
Single Precision Performance	10.6 TeraFLOPS
Half Precision Performance	21.2 TeraFLOPS
GPU Memory	16 GB CoWoS HBM2
Memory Bandwidth	732 GB/s

Tabel 2.
Hasil Evaluasi Epoch terbaik

ResNet	Epoch	F1 Score	Accuracy	IoU	Precision	Recal
18	11	0.9548	0.9838	0.9135	0.9539	0.9557
34	14	0.9609	0.9861	0.9247	0.9654	0.9567
50	13	0.9651	0.9876	0.9325	0.9691	0.9611
101	11	0.9675	0.9884	0.9371	0.9662	0.9681
152	13	0.9663	0.9881	0.9349	0.9686	0.9641

Tabel 3.
Hasil Uji Coba Kecepatan Deteksi

Backbone	Waktu pemrosesan (detik)		
	Rata-rata	Tercepat	Terlama
ResNet-18	0.345	0.341	0.352
ResNet-34	0.342	0.340	0.346
ResNet-50	0.341	0.338	0.347
ResNet-101	0.344	0.335	0.350
ResNet-152	0.342	0.338	0.346

Laserscanner (LIDAR): Velodyne HDL-64E, 2 buah kamera grayscale, 1.4 MP: FLIR FL2-14S3M-C, 2 buah kamera berwarna, 1.4 MP: FLIR FL2-14S3C-C, 4 buah lensa Varifocal, 4-8 mm: Edmund Optics NT59-917.

Lidar berputar dengan frekuensi 10 putaran per detik, dan menangkap sekitar 100 ribu poin per siklus putarannya. Resolusi vertikal LiDAR adalah 64 poin. Kamera dipasang kira-kira sejajar dengan bidang tanah. Gambar kamera dipangkas ke ukuran 1382 x 512 piksel menggunakan mode format 7 libdc. Setelah dilakukan filter dan pemotongan, gambar menjadi sedikit lebih kecil. Kamera dipicu untuk menangkap gambar dengan frekuensi 10 frame per detik oleh LiDAR (saat menghadap ke depan) dengan periode yang disesuaikan.

Pada tiap dataset KITTI baik training maupun testing terdapat data kalibrasi atau intrinsik parameter kamera untuk tiap gambarnya. Pada dataset training, terdapat gambar *ground truth* untuk tiap data. Pada penelitian ini, dataset KITTI *left color image* yang digunakan dibagi lagi menjadi 235 untuk training, 54 untuk validasi, dan 222 untuk testing. *Depth image* dataset KITTI yang digunakan telah dikonversi. Gambar dataset KITTI yang digunakan dapat dilihat pada Gambar 3.

B. Arsitektur SNE-RoadSeg

Dengan adanya metode *semantic segmentation* berbasis Deep Convolutional Neural Network (DCNN) seperti FCN dan DeepLab telah meningkatkan performa dari segmentasi jalan. Namun, DCNN mungkin masih kurang dalam kinerjanya ketika ada gangguan visual seperti pencahayaan berlebih, tampilan ambigu, dan gambar buram. Penggabungan LiDAR dan informasi visual untuk deteksi jalan telah dilakukan pada penelitian sebelumnya [9]. Namun, pendekatan yang memanfaatkan data LiDAR untuk deteksi jalan masih jauh dari efektif dan hanya memberikan peningkatan terbatas dibandingkan metode deteksi jalan berbasis citra visual. Oleh karena itu, pada penelitian ini mengusulkan teknik *data-fusion* berbasis arsitektur CNN yang dapat mengekstrak dan menggabungkan antara fitur pada

gambar RGB (Red-Green-Blue) dan *depth image* yang didapat dari data LiDAR untuk deteksi ruang kosong yang akurat.

Arsitektur SNE-RoadSeg yang digunakan secara garis besar memiliki dua bagian utama, yaitu modul SNE dan RoadSeg. Arsitektur SNE-RoadSeg secara lengkap dapat dilihat pada Gambar 4. Modul SNE digunakan untuk mengestimasi permukaan normal menggunakan input dari *depth image*.

Arsitektur dari SNE sendiri dapat dilihat pada Gambar 5. Pada perspektif model kamera, titik 3D $\mathbf{P} = [X, Y, Z]^T$ dalam sistem koordinat Euclidean dapat dihubungkan dengan piksel dari gambar 2D $\mathbf{p} = [x, y]^T$ menggunakan:

$$Z \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{P} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{P} \quad (1)$$

dimana \mathbf{K} merupakan intrinsik parameter dari kamera; $\mathbf{p}_0 = [x_0, y_0]^T$ merupakan tengah gambar; f_x dan f_y merupakan panjang fokus kamera dalam pixel. Cara termudah untuk mengestimasi permukaan normal $\mathbf{n} = [n_x, n_y, n_z]^T$ dari \mathbf{P} adalah dengan menyesuaikan pada bidang lokal:

$$n_x X + n_y Y + n_z Z + d = 0 \quad (2)$$

Dengan $\mathbf{N}_p^+ = [\mathbf{P}, \mathbf{N}_p]^T$, dimana $\mathbf{N}_p = [\mathbf{Q}_1, \dots, \mathbf{Q}_k]^T$ merupakan kumpulan titik yang berdekatan dari \mathbf{P} . Dengan menggabungkan Persamaan 1 dengan Persamaan 2 didapatkan Persamaan 3.

$$\frac{1}{Z} = -\left(\frac{1}{d}\right) \left(\frac{n_x(x - x_0)}{f_x} + \frac{n_y(y - y_0)}{f_y} + n_z \right) \quad (3)$$

Kemudian dilakukan penurunan terhadap x dan juga y pada Persamaan 3, diperoleh Persamaan 4.

$$g_x = \frac{\partial \left(\frac{1}{Z}\right)}{\partial x} = -\frac{n_x}{df_x}, \quad g_y = \frac{\partial \left(\frac{1}{Z}\right)}{\partial y} = -\frac{n_y}{df_y} \quad (4)$$

Seperti yang diilustrasikan pada Gambar 2, masing-masing untuk g_x dan g_y dapat didekati dengan konvolusi invers dari *depth image* $1/Z$ dengan menggunakan horizontal dan vertical gradien filter.

Dengan menyusun ulang hasil yang diperoleh pada Persamaan 4 dalam nilai n_x dan n_y diperoleh:

$$n_x = -df_x g_x, \quad n_y = -df_y g_y \quad (5)$$

dengan diberikan sembarang himpunan $\mathbf{Q}_i \in \mathbf{N}_p$, dapat dihitung n_{z_i} dengan mensubstitusi Persamaan 5 ke Persamaan 2 diperoleh Persamaan 6 (Gambar 6).

$$n_{z_i} = \frac{d(f_x \Delta X_i g_x + f_y \Delta Y_i g_y)}{\Delta Z_i} \quad (6)$$

Sebuah kumpulan titik k yang berdekatan \mathbf{N}_p dari \mathbf{P} dapat menghasilkan nilai k yang dinormalisasi pada permukaan normal $\overline{\mathbf{n}}_1, \dots, \overline{\mathbf{n}}_k$, dimana

$$\overline{\mathbf{n}}_i = \frac{n_i}{\|n_i\|_2} = [\overline{n_{x_i}}, \overline{n_{y_i}}, \overline{n_{z_i}}]^T \quad (7)$$

Karena tiap normalisasi pada permukaan normal diproyeksikan pada bidang bola dengan titik tengah $(0, 0, 0)$ dan radius 1, sehingga permukaan normal optimal $\hat{\mathbf{n}}$ untuk \mathbf{P} juga diproyeksikan disuatu tempat pada bidang bola tersebut, dimana proyeksi dari $\overline{\mathbf{n}}_1, \dots, \overline{\mathbf{n}}_k$ terdistribusi paling intensif. Penulisan $\hat{\mathbf{n}}$ pada koordinat bola pada Persamaan 8:

$$\hat{\mathbf{n}} = [\sin\theta \cos\varphi, \sin\theta \sin\varphi, \cos\theta]^T \quad (8)$$

dimana $\theta \in [0, \pi]$ menunjukkan kemiringan dan $\varphi \in [0, 2\pi]$ menunjukkan azimuth. Nilai φ dapat dihitung dengan Persamaan 9.

$$\varphi = \arctan\left(\frac{f_y g_y}{f_x g_x}\right) \quad (9)$$

Sudut antara pasangan normalisasi permukaan normal kurang dari $\pi/2$, sehingga $\hat{\mathbf{n}}$ dapat diestimasi dengan meminimalkan $E = -\sum_{i=1}^k \hat{\mathbf{n}} \cdot \overline{\mathbf{n}}_i$, $\frac{\partial E}{\partial \theta} = 0$ diperoleh:

$$\theta = \arctan\left(\frac{(\sum_{i=1}^k \overline{n_{x_i}} \cos\varphi + \sum_{i=1}^k \overline{n_{y_i}} \sin\varphi)}{\sum_{i=1}^k \overline{n_{z_i}}}\right) \quad (10)$$

dengan mensubstitusi θ dan φ ke Persamaan 8 diperoleh hasil permukaan normal optimal $\hat{\mathbf{n}}$ seperti pada Gambar 5.

RoadSeg menggunakan arsitektur enkoder dan dekoder. Pada bagian encoder, informasi permukaan normal yang dihasilkan melalui proses SNE kemudian akan diproses dan digabungkan tiap elemennya dengan gambar RGB yang telah sama-sama diekstrak sebelumnya melalui proses konvolusi, normalisasi batch, dan proses aktivasi ReLU. Normalisasi *batch* merupakan teknik *training* yang membagi input ke lapisan konvolusi untuk tiap batch kecil yang dapat menstabilkan proses belajar model. Sedangkan ReLU merupakan fungsi aktivasi yang akan mengeluarkan nilai output jika input bernilai positif dan output nol jika input bernilai negatif. ReLU digunakan karena lebih mudah dalam pembelajaran model dan dapat mencapai performa yang baik.

Enkoder pada RoadSeg menggunakan jaringan *backbone* (tulang belakang) Residual Network (ResNet). Kemudian untuk decoder, RoadSeg menggunakan dua jenis modul yaitu *feature extractors* $F^{i,j}$ dan *upsampling layers* $U^{i,j}$ yang terhubung untuk mewujudkan perpaduan fitur yang fleksibel. *Feature extraction* merupakan konversi dari data masukan ke dalam kumpulan fitur. Dalam *machine learning*, *feature extraction* dikaitkan dengan pengurangan jumlah informasi yang diperlukan untuk mendefinisikan kumpulan informasi yang sangat besar tanpa kehilangan informasi penting dan relevan. *Upsampling* merupakan teknik yang digunakan untuk meningkatkan resolusi gambar keluaran lebih baik dari gambar masukan serta mengurangi chanel pada fitur map. Jenis *Upsampling layer* yang digunakan merupakan *Bilinear Upsampling*.

Arsitektur RoadSeg menggunakan ResNet dengan c_n yang merepresentasikan jumlah chanel fitur map atau filter pada level yang berbeda. Koneksi pintar antar lapisan (*layer*) pada ResNet menghasilkan kemampuan untuk dapat melatih suatu model *neural network* lebih baik dibandingkan dengan menggunakan *neural network* biasa karena adanya suatu faktor *skip connection*. Diagram blok dari ResNet pada Gambar 6. Penggunaan dari *skip connection* bertujuan untuk menghindari masalah degradasi [10].

C. Overlay Image

Setelah melalui algoritma SNE-RoadSeg didapatkan gambar prediksi atau *probability image*. Hasil prediksi map yang telah diperoleh akan digabungkan dengan gambar yang asli sehingga menghasilkan gambar output dengan segmentasi jalan yang dapat dilewati.

III. HASIL DAN PEMBAHASAN

Pada bagian ini menjelaskan mengenai hasil pengujian algoritma deteksi ruang kosong jalan yang telah dirancang pada bagian sebelumnya serta dilakukan analisis data.

A. Training Neural Network

Pada penelitian ini, *training* model *neural network* dilakukan menggunakan *Google Collaboratory Platform Pro* dengan spesifikasi hardware komputasi GPU seperti pada

Tabel 1. Spesifikasi GPU yang diberikan oleh platform bersifat tidak tetap bergantung dengan banyaknya penggunaan dalam satu waktu tertentu. Namun pada penelitian ini, spesifikasi GPU yang diperoleh sama.

Training neural network dilakukan pada lima jenis *backbone*, yaitu ResNet-18, ResNet-34, ResNet-50, ResNet-101, dan ResNet-152 dengan konfigurasi *neural network* yang sama serta spesifikasi GPU yang sama. Training dilakukan menggunakan dataset KITTI dengan 235 data masukan. Representasi rata-rata *loss training* dataset KITTI pada kelima jenis penggunaan *backbone* dapat dilihat pada Gambar 7.

Berdasarkan Gambar 7 tersebut, diketahui bahwa *loss* pada *epoch* pertama pada tiap jenis *backbone* adalah yang terbesar. Hal ini dikarenakan model *neural network* tersebut baru saja melakukan *training*. Pada proses *training* dengan kelima jenis *backbone* tersebut, laju penurunan *loss* cenderung menjadi lebih sedikit setelah memasuki *epoch* ke-12. Pada 3 *epoch* berikutnya terlihat bahwa perubahan *loss* sangatlah kecil dan nilai *loss* semakin mendekati nilai nol, sehingga proses *training* dapat dicukupkan hingga *epoch* ke-15.

Berdasarkan Gambar 8, dapat diketahui bahwa rata-rata waktu/durasi *training* model *neural network* pada ResNet-152 merupakan yang tertinggi/terlama dibandingkan dengan lainnya. Durasi *training* model menggunakan ResNet-152 rata-rata bernilai 175 detik. Pada penggunaan ResNet-101, durasi *training* model *neural network* rata-rata bernilai 153 detik.

Sedangkan pada penggunaan ResNet-50, durasi *training* model *neural network* rata-rata bernilai 135 detik. Durasi/waktu *training* model menggunakan ResNet-34 rata-rata bernilai 95 detik dan ResNet-18 rata-rata bernilai 91 detik. Semakin banyak lapisan pada ResNet yang digunakan, waktu yang dibutuhkan untuk proses *training* juga meningkat.

B. Testing Neural Network

Hasil evaluasi model *neural network* menggunakan ResNet-18 pada dataset KITTI dapat dilihat pada Gambar 9. Nilai perhitungan *F1 score* digunakan untuk menentukan *epoch* terbaik hasil *training*, dimana diperoleh skor terbaik pada *epoch* ke-11 dengan skor *F1 score* sebesar 0.9548, skor *accuracy* 0.9838, skor *IoU* sebesar 0.9135, skor *precision* sebesar 0.9539, dan skor *recall* sebesar 0.9557.

Hasil evaluasi model *neural network* menggunakan ResNet-34 pada dataset KITTI dapat dilihat pada Gambar 10. Skor *F1 score* terbaik pada *epoch* ke-14 dengan nilai 0.9609, skor *accuracy* sebesar 0.9861, skor *IoU* sebesar 0.9247, skor *precision* sebesar 0.9654, dan skor *recall* sebesar 0.9564. Hasil evaluasi model *neural network* menggunakan ResNet-50 pada dataset KITTI dapat dilihat pada Gambar 11. Skor *F1 score* terbaik pada *epoch* ke-13 dengan nilai sebesar 0.9651, skor *accuracy* sebesar 0.9876, skor *IoU* sebesar 0.9325, skor *precision* sebesar 0.9691, dan skor *recall* sebesar 0.9611.

Hasil evaluasi model *neural network* menggunakan ResNet-101 pada dataset KITTI dapat dilihat pada Gambar 12. Skor *F1 score* terbaik pada *epoch* ke-11 dengan nilai 0.9675, skor *accuracy* sebesar 0.9884, skor *IoU* sebesar 0.9371, skor *precision* sebesar 0.9662, dan skor *recall* sebesar 0.9688. Sedangkan hasil evaluasi pada ResNet-152 dapat dilihat pada Gambar 13.

Skor *F1 score* terbaik pada *epoch* ke-13 dengan nilai sebesar 0.9663, skor *accuracy* sebesar 0.9881, skor *IoU*

sebesar 0.9349, skor *precision* sebesar 0.9686, dan skor *recall* sebesar 0.9641.

Berdasarkan hasil evaluasi model yang telah dilakukan pada dataset KITTI untuk tiap ResNet yang digunakan, didapatkan *epoch* terbaik dengan skor parameter yang dapat dilihat pada Tabel 2. Skor parameter yang dihasilkan menunjukkan ResNet-101 dan ResNet-152 memiliki skor yang lebih tinggi apabila dibandingkan dengan ResNet-18, ResNet-34, dan ResNet-50. Hal ini dapat dikaitkan dengan jumlah lapisan pada ResNet-101 dan ResNet-152 lebih banyak dibandingkan dengan ResNet-18, ResNet-34, dan ResNet-50 sehingga performa dalam mendeteksi ruang kosong jalan yang dataset ujinya tidak terlalu representatif terhadap dataset *training* lebih baik. Skor ResNet-101 yang lebih baik namun tidak terlalu jauh jika dibandingkan dengan ResNet-152 hal ini dikarenakan *loss* saat *training* dan juga dataset evaluasi yang digunakan namun, kedua ResNet ini dapat digunakan dengan baik untuk deteksi ruang kosong pada jalan.

C. Hasil Demonstrasi Algoritma Deteksi Ruang Kosong Jalan

Pada demonstrasi algoritma deteksi ruang kosong jalan, akan ditampilkan beberapa hasil data kualitatif yang terdiri dari data hasil deteksi ruang ksoong jalan pada dataset KITTI yang berupa foto. Hasil demonstrasi akan menampilkan data tes dari model untuk ResNet-18, ResNet-34, ResNet-50, ResNet-101, dan ResNet-152. Contoh hasil deteksi pada dataset KITTI dapat dilihat pada Gambar 14. Terdapat kesalahan deteksi pada trotoar (*False Positive*), hal ini dikarenakan trotoar memiliki struktur dan tekstur yang hampir sama seperti jalan sehingga modul SNE mendeteksinya sebagai jalan. Kegagalan deteksi juga terjadi pada jalan dengan rel kereta, hal ini karena data *training* yang tidak representatif terhadap kondisi lingkungan. Kegagalan deteksi dapat dilihat pada Gambar 15.

D. Kecepatan Deteksi Algoritma Deteksi Ruang Kosong Jalan

Uji coba kecepatan deteksi dilakukan untuk mengetahui banyak waktu yang diperlukan dalam proses pengolahan gambar dengan algoritma SNE-RoadSeg. Uji coba dilakukan untuk kelima jenis *backbone* yang digunakan selama lima kali. Seluruh data kecepatan deteksi diperoleh menggunakan GPU Nvidia Tesla P100 dengan hasil uji coba yang dapat dilihat pada Tabel 3. Rata-rata waktu pemrosesan gambar pada algoritma SNE-RoadSeg dengan GPU Nvidia Tesla P100 sebesar 0.343 detik

IV. KESIMPULAN

Dalam penelitian ini, tingkat *F1 score* dari algoritma deteksi ruang kosong jalan bergantung pada jenis *backbone* ResNet yang digunakan. Sehingga semakin banyak *layer* pada tipe ResNet yang digunakan semakin baik skor *F1 score* yang diperoleh. Hal ini dibuktikan dengan skor *F1 score* pada ResNet-101 mencapai 0.9675 sedangkan skor *F1 score* pada ResNet-18 hanya mampu mencapai nilai 0.9548. Pada beberapa gambar terdapat kesalahan deteksi (*False Positive*) dan juga kegagalan deteksi dikarenakan kesamaan struktur trotoar dan data *training* yang kurang reperesentatif terhadap lingkungan. Kecepatan deteksi algoritma berada pada kisaran 0.343 detik untuk tiap gambar yang diproses bergantung pada *hardware* yang digunakan.

DAFTAR PUSTAKA

- [1] M. V. Rajasekhar and A. K. Jaswal, "Autonomous Vehicles: The Future of Automobiles," in *IEEE International Transportation Electrification Conference, ITEC-India 2015*, 2015, pp. 1–6, doi: 10.1109/ITEC-India.2015.7386874.
- [2] S. C. Widiyanto, I. R. E. A. K., and A. Santoso, "Deteksi Lajur Mobil Otonom pada Kondisi Gambar yang Terdistorsi dan Kurang Pencahayaan," Institut Teknologi Sepuluh Nopember.
- [3] A. Firmansyah, A. A. Firmansyah, R. E. AK, and A. Santoso, "Deteksi halangan menggunakan metode stereo R-CNN pada mobil otonom," *J. Tek. ITS*, vol. 9, no. 2, pp. E160–E166, 2021, doi: 10.12962/j23373539.v9i2.53687.
- [4] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, and R. Urtaşun, "Estimating Drivable Collision-Free Space from Monocular Video," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 420–427, doi: 10.1109/WACV.2015.62.
- [5] F. Pizzati and F. Garcia, "Enhanced Free Space Detection in Multiple Lanes Based on Single CNN with Scene Identification," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2019, vol. 2019-June, pp. 2536–2541, doi: 10.1109/IVS.2019.8814181.
- [6] I. Ulku and E. Akagunduz, "A Survey on deep learning-based architectures for semantic segmentation on 2D images," *Appl. Artif. Intell.*, vol. 10, no. 1, 2019, doi: 10.1080/08839514.2022.2032924.
- [7] X. Liu, Z. Deng, and Y. Yang, "Recent progress in semantic image segmentation," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1089–1106, 2019, doi: 10.1007/s10462-018-9641-3.
- [8] J. Fritsch, T. Kuhn, and A. Geiger, "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms," in *IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1693–1700, doi: 10.1109/ITSC.2013.6728473.
- [9] L. Xiao, R. Wang, B. Dai, Y. Fang, D. Liu, and T. Wu, "Hybrid conditional random field based camera-LIDAR fusion for road detection," *Inf. Sci. (Ny)*, vol. 432, pp. 543–558, 2018, doi: 10.1016/j.ins.2017.04.048.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.