

# IMPLEMENTASI NAVIGASI KENDARAAN DENGAN MENGGUNAKAN *EMBEDDED PC* DAN DETEKSI JALAN BERBASIS METODE *TEXTURE FILTER*

Johannes Mae<sup>1</sup>, Rudy Susanto<sup>2</sup>

<sup>1,2</sup>Program Studi Sistem Komputer, Fakultas Teknik, Universitas Bina Nusantara,  
Jln. K. H. Syahdan No. 9, Jakarta, 11480, Indonesia  
E-mail:<sup>1</sup>johannes@binus.edu, <sup>2</sup>mailtorudy@gmail.com

## **Abstrak**

Navigasi kendaraan adalah proses pengenalan jalan dan mengarahkan kendaraan pada lintasan yang diinginkan. Penelitian tentang navigasi kendaraan umumnya berfokus pada pengembangan Advanced Driver Assistance System (ADAS) untuk meningkatkan keselamatan dalam berkendara dengan tujuan jangka panjang untuk mewujudkan kendaraan otonom. Pada umumnya navigasi kendaraan dilakukan dengan menggunakan banyak jenis instrumen pengukuran yang berukuran besar dan sulit diaplikasikan pada kendaraan secara umum. Penelitian ini dilakukan untuk menguji ADAS dengan menggunakan instrumen sederhana dan mudah untuk dipasang pada kendaraan. Navigasi dimulai dengan proses deteksi sisi jalan dengan mengamati area jalan pada jarak 10m hingga 14m di depan kendaraan menggunakan teknik pengolahan citra. Hasil deteksi sisi jalan digunakan untuk menghasilkan rekomendasi arah gerak kendaraan yang diubah ke dalam bentuk rekomendasi putaran kemudi bagi pengemudi. Rekomendasi arah gerak kendaraan diperoleh melalui perhitungan sudut yang dibentuk oleh kendaraan terhadap titik tengah lajur jalan pada jarak yang telah ditentukan. Pengujian dilakukan secara real-time dengan menggunakan sebuah Embedded PC dapat mencapai waktu pengolahan citra 13,962 ms dengan tingkat keberhasilan deteksi jalan sebesar 89,7% dan estimasi tingkat kesesuaian data rekomendasi putaran kemudi sebesar 91,04% yang dibandingkan terhadap putaran kemudi oleh pengemudi.

**Kata kunci**—navigasi kendaraan, deteksi sisi jalan, pengolahan citra, Embedded PC, rekomendasi putaran kemudi

## **Abstract**

Vehicle navigation is process of detecting road area and directing the vehicle to desired trajectory. Research on vehicle navigation generally focused on Advanced Driver Assistance System (ADAS) development to improve driving safety with long-term goals is to establish an autonomous vehicle. Vehicle navigation generally implemented by using many types of large measuring instruments and difficult to be applied on common vehicle. This research is conducted to test ADAS by employing simple instruments and easy to be applied on vehicles. Navigation starts with road-side detection process by observing road area from 10m to 14m distance in front of the vehicle using image processing technique. The result of road-side detection is used to produce vehicle motion direction recommendation which is converted into steering-wheel rotation recommendation for the driver. Vehicle motion direction recommendation is obtained from angle calculation formed between the vehicle relative to road center point from a predetermined distance. Experiment is conducted in real-time on an Embedded PC can achieve image processing time of 13.962 ms with successful road detection

rate of 89.7% and estimation of steering-wheel recommendation conformity rate of 91.04% compared to steering-wheel rotation by the driver.

**Keywords**—vehicle navigation, road-side detection, image processing, Embedded PC, steering-wheel rotation recommendation

## 1. PENDAHULUAN

Perkembangan teknologi saat ini telah membuka banyak peluang penelitian terbaru seperti pengembangan *Advanced Driver Assistant System* (ADAS) pada teknologi kendaraan. Di masa depan, komputer yang dilengkapi dengan teknologi pintar akan mampu menggantikan peran manusia seutuhnya sebagai pengemudi. Kelebihan dari pemanfaatan komputer pada teknologi kendaraan adalah kemampuan untuk melakukan komputasi secara cepat apabila diberikan algoritma tertentu, namun tidak semua situasi yang dihadapi di jalan dapat dengan mudah diubah menjadi algoritma tertentu. Oleh sebab itu, tantangan bagi peneliti saat ini adalah menemukan algoritma sederhana untuk merespon situasi di jalan sehingga mampu diproses oleh komputer secara cepat dan dapat diaplikasikan dengan mudah.

Pada dasarnya komputer membutuhkan informasi yang sama dengan yang diterima oleh manusia pada saat mengemudi. Salah satunya adalah informasi tentang jarak kendaraan terhadap sebuah objek tertentu. Komputer mampu melakukan pengukuran jarak dengan tingkat akurasi yang tinggi dengan instrumen seperti laser [1] dengan metode *Light Detection and Ranging* (LIDAR), ultrasonik [2] yang memanfaatkan pemantulan gelombang suara, dan radar [3] yang memanfaatkan pemantulan gelombang radio. Instrumen laser dan radar sulit diaplikasikan pada kendaraan karena dimensi instrumen yang besar, sedangkan ultrasonik memiliki sudut deteksi yang terbatas. Oleh sebab itu, banyak penelitian terkini yang memanfaatkan kamera [4] dengan teknik pengolahan citra untuk menganalisa kondisi pada jalan.

Selain informasi jarak, manusia mampu melakukan klasifikasi terhadap objek yang ada di jalan dengan mudah, seperti trotoar, pohon, pejalan kaki dan kendaraan lainnya di jalan. Dengan teknik pembelajaran, banyak penelitian telah dilakukan dengan menggunakan kecerdasan buatan dan berhasil melakukan pengenalan jalan [5] maupun mengklasifikasikan objek-objek yang beragam, seperti rambu lalu lintas [6], pejalan kaki [7] dan lubang di jalan [8], namun metode pembelajaran seperti ini membutuhkan *dataset* dalam jumlah yang sangat banyak dan bervariasi, serta membutuhkan kemampuan komputasi yang tinggi sehingga sulit diaplikasikan pada komputer sederhana secara *real-time*.

Navigasi pada kendaraan dapat dilakukan apabila informasi area jalan yang dapat dilalui telah diperoleh. Area jalan dan posisi kendaraan pada bidang jalan dapat ditentukan apabila sisi jalan telah diketahui. Metode untuk mendeteksi sisi jalan umumnya terbagi menjadi 2 kategori berdasarkan basis analisa gambarnya, antara lain metode deteksi tepi [9] dan metode deteksi pola [10]. Kesulitan yang dihadapi dari kedua metode tersebut adalah bergantung pada marka jalan dan titik lenyap atau *vanishing point* agar dapat mendeteksi sisi jalan.

Berdasarkan hasil observasi di atas, penelitian ini mengusulkan sebuah algoritma sederhana menggunakan sebuah kamera yang mampu untuk mendeteksi sisi jalan. Hasil deteksi sisi jalan kemudian dilakukan *Inverse Perspective Mapping* (IPM) untuk memperoleh informasi orientasi kendaraan pada bidang jalan yang dipadukan dengan kecepatan kendaraan sehingga menghasilkan rekomendasi putaran kemudi. Parameter keberhasilan penelitian ini diukur dari kemampuan komputer untuk mendeteksi jalan secara *real-time* dan rekomendasi putaran kemudi yang dihasilkan oleh komputer mampu mengarahkan kendaraan menuju lintasan yang

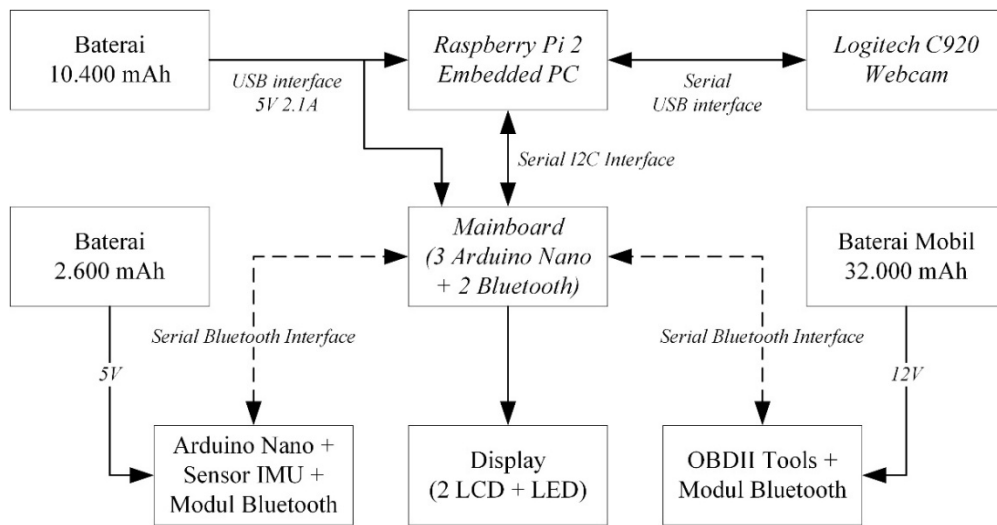
diinginkan. Rekomendasi putaran kemudi oleh komputer akan dibandingkan dengan pergerakan kemudi oleh pengemudi yang turut mengarahkan kendaraan agar selalu berada di posisi tengah pada lajur kanan jalan.

## 2. METODE PENELITIAN

Penelitian ini dilakukan secara *real-time* pada mobil yang sedang bergerak menggunakan *Embedded PC*. Perancangan sistem baik dari sisi perangkat keras maupun algoritma atau metode yang digunakan dilakukan agar menyesuaikan dengan tujuan penelitian ini. Beberapa tahapan yang dilakukan meliputi perancangan perangkat keras, kalibrasi kamera, metode deteksi sisi jalan dan metode navigasi kendaraan.

### 2.1 Perancangan Perangkat Keras

Penelitian ini memanfaatkan perangkat keras yang bersifat portabel dan mudah dipasang. Perangkat keras yang digunakan meliputi sebuah *Raspberry Pi 2* yang berperan dalam pengolahan citra dan kalkulasi sudut; sebuah *webcam Logitech C920* untuk mengambil gambar jalan; sebuah modul *mainboard* untuk mengambil data putaran kemudi dan kecepatan kendaraan dari sensor dan menampilkannya pada *Liquid Crystal Display (LCD)*; sebuah sensor *Inertial Measurement Unit (IMU)* untuk mengukur derajat putaran kemudi; dan sebuah alat *On-Board Diagnostic Tools (OBD)* untuk mengambil data kecepatan kendaraan. Keseluruhan perangkat keras yang digunakan serta hubungan antarmuka ditampilkan pada gambar 1.



Gambar 1 Diagram Blok Sistem

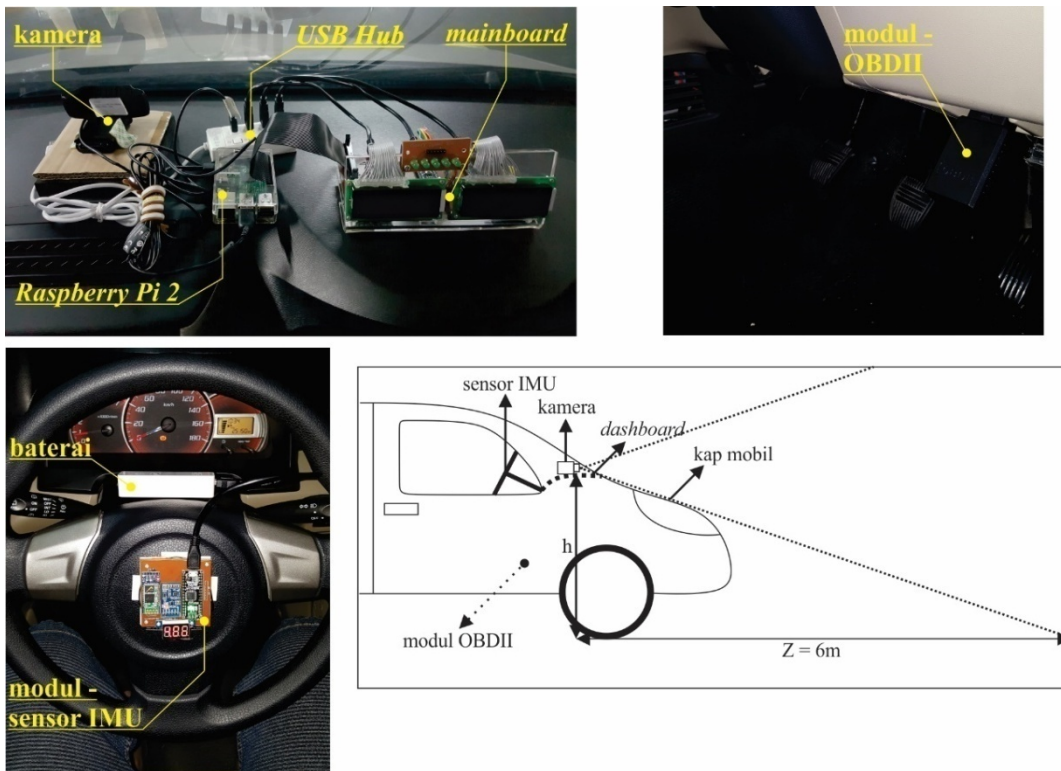
Penempatan perangkat keras di dalam kabin mobil ditampilkan pada gambar 2. Posisi kamera ditempatkan di tengah *dashboard* mobil dan diatur agar sejajar dengan bidang jalan. Kecepatan pengambilan gambar pada kamera dibatasi hingga *15 frame-per-second (FPS)* untuk meringankan beban komputasi dan menjaga agar ukuran video yang direkam tidak terlalu besar.

Sebuah sensor IMU dipasang pada poros putaran kemudi untuk mengukur derajat putaran kemudi. Bagian dari sensor IMU yang digunakan adalah *accelerometer* dan *gyroscope*. *Accelerometer* digunakan untuk mengukur percepatan dari sebuah objek. Dengan persamaan trigonometri (1), hasil pengukuran percepatan dapat dikonversi menjadi derajat rotasi pada bidang Z. Sedangkan *gyroscope* digunakan untuk mengukur derajat rotasi pada suatu sumbu dalam rentang waktu yang telah ditentukan. Pada penelitian ini, rentang waktu diatur selama 8ms atau 125Hz, sehingga besarnya rotasi pada sumbu-Z dapat dihitung dengan menggunakan

persamaan (2) dengan menggunakan faktor skala sensitivitas 65,5 LSB/(°/s). Masing-masing sensor memiliki kekurangan, dimana *accelerometer* sangat sensitif terhadap getaran dan *gyroscope* dapat mengalami akumulasi kesalahan atau *error drift* secara perlahan. Guna mengurangi kedua efek tersebut, maka diperlukan penggabungan hasil pengukuran kedua sensor, dimana pada penelitian ini diatur penggabungan dengan rasio masing-masing sebesar 50%.

$$Z_{accel} = atan2(aY, aX) = \begin{cases} atan\left(\frac{aY}{aX}\right), & \text{jika } aX > 0, \\ \frac{\pi}{2} - atan\left(\frac{aX}{aY}\right), & \text{jika } aY > 0, \\ -\frac{\pi}{2} - atan\left(\frac{aX}{aY}\right), & \text{jika } aY < 0, \\ atan\left(\frac{aY}{aX}\right) \pm \pi, & \text{jika } aX < 0, \\ \infty, & \text{jika } aX = 0 \text{ dan } aY = 0 \end{cases} \quad (1)$$

$$Z_{gyro(i)} = Z_{gyro(i-1)} + \left(\frac{gZ}{65,5} * 0,008\right) \quad (2)$$



Gambar 2 Penempatan Perangkat Keras Pada Mobil

## 2.2 Kalibrasi Kamera

Kalibrasi kamera dilakukan dengan menyediakan 20 gambar pola hitam putih menyerupai pola papan catur dengan resolusi 640x480 *pixel* yang diambil pada sudut yang berbeda-beda menggunakan kamera pada penelitian. Kalibrasi dilakukan dengan menggunakan *toolbox* *MatLabyang* dapat secara otomatis mendeteksi titik temu antara pola hitam putih. Ukuran pola hitam putih telah ditentukan yaitu 30mm. Hasil dari kalibrasi yang digunakan adalah parameter intrinsik dari kamera (3), dimana  $f_x, f_y$  adalah *focal length* dalam satuan *pixel*,  $s$  adalah *skew coefficient*, dan  $u_0, v_0$  adalah *principal point* dalam satuan *pixel*.

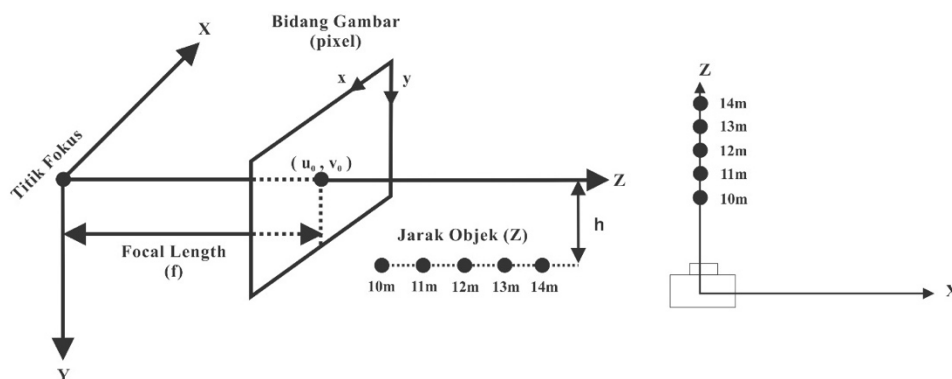
$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ u_0 & v_0 & 1 \end{bmatrix} = \begin{bmatrix} 616,6042 & 0 & 0 \\ 0 & 618,8103 & 0 \\ 322,4683 & 236,7301 & 1 \end{bmatrix}$$

Parameter intrinsik kamera diperlukan dalam proses IPM yaitu proses transformasi geometri yang berguna untuk menghilangkan efek perspektif pada gambar. Proses IPM dilakukan dalam 2 tahap transformasi, yaitu transformasi bidang gambar ke bidang riil  $I \rightarrow S$ , kemudian dilakukan transformasi kembali dari bidang riil ke bidang gambar  $S \rightarrow I$  sehingga menghasilkan objek dalam bidang gambar yang telah dihilangkan perspektifnya[11]. Ukuran riil sebuah objek pada bidang gambar 2-dimensi  $x_i, y_i$  dapat dihitung dengan menggunakan persamaan (4) dan memanfaatkan parameter intrinsik kamera yang telah diperoleh sebelumnya dan ketinggian kamera  $Y$  dari bidang jalan setinggi 1240mm yang diperoleh melalui proses pengukuran.

$$\begin{aligned} x_i &= \frac{f_x * X}{Z} + u_0 \\ y_i &= \frac{f_y * Y}{Z} + v_0 \end{aligned} \tag{4}$$

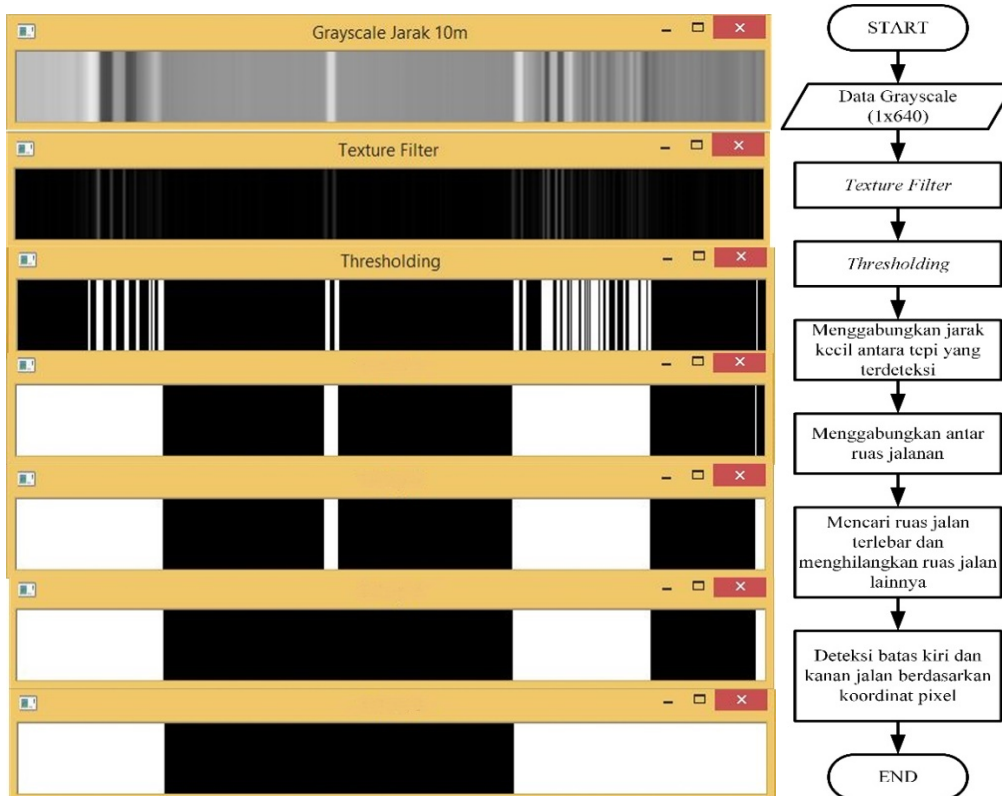
### 2.3 Metode Deteksi Sisi Jalan

Proses deteksi sisi jalan dilakukan terhadap 5 baris pada gambar yang mewakili jarak 10m, 11m, 12m, 13m dan 14m, seperti yang ditunjukkan oleh gambar 3. Penentuan jarak riil dalam bidang gambar dapat menggunakan persamaan (4). Penentuan jarak tersebut berdasarkan hasil percobaan sebelumnya bahwa pada jarak di bawah 10m, terdapat bagian sisi jalan yang tidak terlihat oleh kamera, sedangkan pada jarak di atas 14m, area jalan akan terlihat semakin kecil karena efek perspektif dan objek-objek lain seperti marka jalan yang tipis tidak dapat terlihat dengan jelas oleh kamera.



Gambar 3 Penentuan Jarak Objek Riil Pada Bidang Gambar

Pada masing-masing baris gambar yang ditentukan dilakukan proses *texture filter* untuk mencari batas-batas jalan. *Texture filter* dalam pengolahan grafik komputer adalah sebuah metode *smoothing* pada tekstur gambar dengan memanfaatkan informasi intensitas warna di sekitar *pixel* yang diamati. Metode ini dilakukan secara berulang pada gambar *grayscale* dengan jumlah perulangan sebesar lebar gambar secara horizontal. Hasil *texture filter* memperlihatkan area yang memiliki intensitas *pixel* yang berbeda jauh akan menghasilkan nilai filter yang besar, seperti yang diperlihatkan pada gambar 4. Dengan *texture filter*, area tepi pada sebuah baris gambar dapat dikenali. Hasil *texture filter* yang memuat besaran nilai antara 0-255 diberikan *thresholding* dengan sebuah nilai sensitifitas untuk memperjelas hasil filter. Selanjutnya hasil *thresholding* dipadukan dengan aturan logis untuk mengeliminasi celah kecil dan menggabungkan area yang dipisahkan oleh garis pemisah lajur jalan.

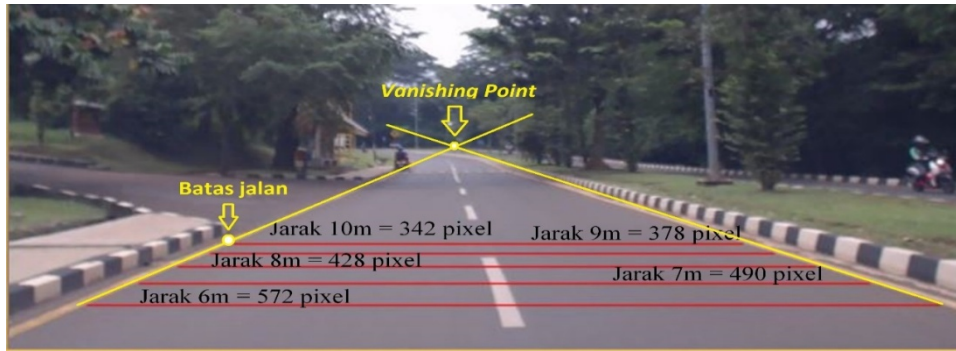


Gambar 4 Proses Deteksi Batas Jalan Pada Suatu Baris Gambar

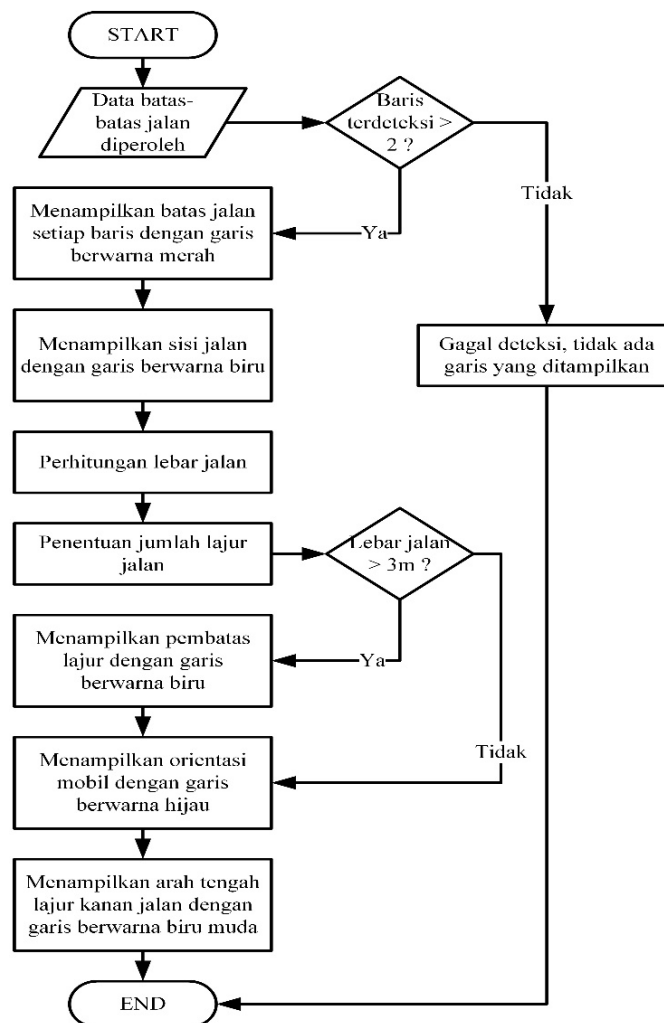
Deteksi sisi jalan dilakukan dengan menggabungkan batas-batas jalan telah diperoleh. Pada penelitian ini, deteksi sisi jalan dinyatakan berhasil apabila minimal 3 baris informasi batas jalan telah diketahui. Kombinasi dari posisi batas jalan kiri dan kanan akan membentuk perspektif dengan derajat kemiringan tertentu yang akan saling berpotongan di suatu titik lenyap / *vanishing point*. Efek perspektif dapat diukur dari jumlah *pixel* yang dihasilkan proses deteksi batas jalan. Karena terjadinya perspektif, maka area jalan yang lebih jauh akan memiliki jumlah *pixel* secara horizontal yang lebih sedikit dibandingkan dengan area jalan yang lebih dekat dari kamera meskipun lebar jalan secara riil adalah sama, seperti yang ditunjukkan oleh gambar 5 dengan contoh jarak peninjauan 6m hingga 10m. Dengan menggunakan parameter hasil kalibrasi kamera,  $f_x, u_0$  (3) dan  $x_i$  pada persamaan (4), maka jumlah *pixel* yang mewakili area jalan,  $\Delta x$ , pada jarak pengamatan tertentu,  $Z$ , dapat dikonversi menjadi informasi lebar jalan,  $\Delta X$ , yang digunakan untuk menentukan jumlah lajur pada jalan. Tabel 1 memperlihatkan perhitungan lebar jalan berdasarkan pengukuran jumlah *pixel* pada gambar 5, dimana hasil pengukuran riil lebar jalan di lapangan (*ground truth*) diperoleh selebar 5,5 meter. Pada penelitian ini, lebar sebuah lajur jalan ditentukan minimal 3 meter sehingga lebar jalan pada lokasi pengujian akan ditentukan sistem sebagai jalan dengan 2 lajur dan sistem akan menambahkan sebuah garis pemisah lajur. Keseluruhan proses deteksi sisi jalan dijelaskan pada gambar 6.

Tabel 1 Pengukuran Lebar Jalan Berdasarkan Jumlah *Pixel*

Jarak peninjauan( $Z$ ) dalam mm	Lebar <i>Pixel</i> ( $\Delta x$ )	Lebar area jalan( $\Delta X$ ) dalam mm
6000	572	5565.969
7000	490	5562.726
8000	428	5552.995
9000	378	5517.316
10000	342	5546.508



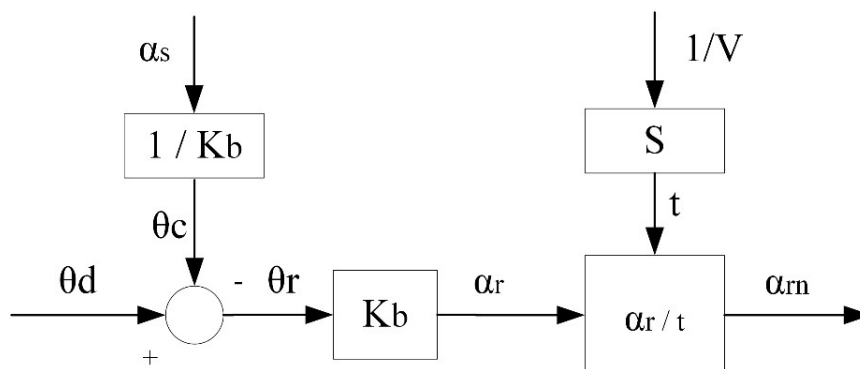
Gambar 5 Pengukuran Lebar Jalan Berdasarkan Jumlah Pixel



Gambar 6 Diagram Alir Proses Deteksi Sisi Jalan

#### 2.4 Metode Navigasi Kendaraan

Navigasi kendaraan diwujudkan dalam bentuk rekomendasi putaran kemudi oleh komputer. Proses navigasi kendaraan ditunjukkan melalui diagram blok pada gambar 7. Perhitungan sudut dilakukan pada bidang gambar hasil IPM seperti ilustrasi pada gambar 8 yang melibatkan empat parameter penting, yaitu orientasi kamera, orientasi roda, titik tujuan pada bidang jalan, dan kecepatan kendaraan saat ini.



Gambar 7 Diagram Blok Navigasi Kendaraan

Orientasi kamera merupakan acuan dalam perhitungan sudut rekomendasi karena menggambarkan posisi kendaraan relatif terhadap titik tujuan pada bidang jalan. Sudut yang dibentuk antara titik tengah kamera terhadap titik tujuan pada jarak 14m dilambangkan dengan simbol  $\theta_d$  dan dapat dihitung dengan persamaan (5). Orientasi rodadibentuk oleh pergerakan putaran kemudi. Besarnya orientasi roda dapat diukur dengan membaca sudut putaran kemudi  $\alpha_s$  yang dihasilkan oleh sensor IMU. Melalui proses *spooring* pada mobil yang digunakan, diperoleh sebuah konstanta  $K_b$  yaitu sebesar  $18^\circ$  yang memberikan hubungan antara putaran kemudi  $\alpha_s$  dengan putaran roda  $\theta_c$  sehingga orientasi roda dapat diestimasi dengan persamaan (6). Dengan demikian, sudut  $\theta_r$  yang perlu dibentuk oleh kendaraan agar dapat mencapai titik tujuan dapat ditentukan dengan persamaan (7) yang kemudian dikonversikan dalam besaran rekomendasi putaran kemudi  $\alpha_r$  dengan persamaan (8).

$$\theta_d = \tan^{-1} \frac{\Delta X}{14} \quad (5)$$

$$\theta_c = \frac{\alpha_s}{K_b} \quad (6)$$

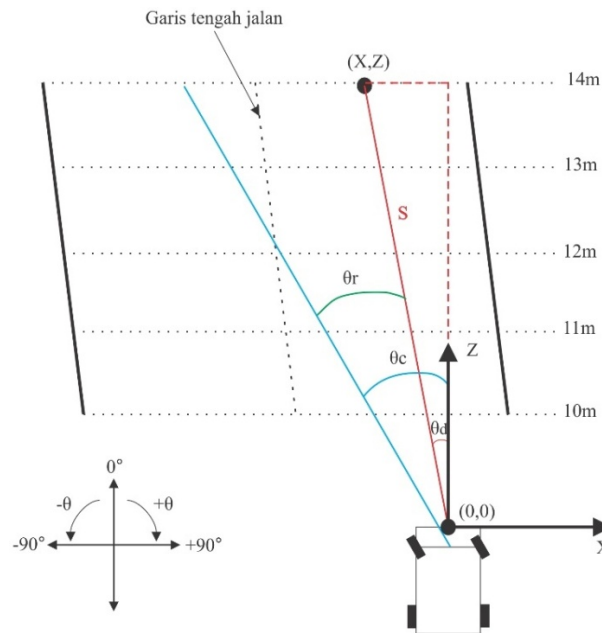
$$\theta_r = \theta_d - \theta_c \quad (7)$$

$$\alpha_r = \theta_r * K_b \quad (8)$$

Variabel kecepatan  $V$  yang diperoleh dari OBD turut mempengaruhi besaran sudut rekomendasi yang dihasilkan oleh sistem. Rekomendasi putaran kemudi akan dibagi secara gradual terhadap waktu capai target untuk menghindari perubahan nilai rekomendasi yang terlalu signifikan dalam waktu singkat. Perhitungan parameter waktu  $t$  sebagai pembagi menggunakan persamaan (9).

$$t = \frac{S}{V} = \frac{14}{V * \cos(\theta_d)} \quad (9)$$





**Gambar 8** Ilustrasi Navigasi Kendaraan Melalui Rekomendasi Putaran Kemudi

### 3. HASIL DAN PEMBAHASAN

Pengukuran keberhasilan dalam penelitian ini berdasarkan analisa dari data-data yang diperoleh, antara lain analisa data tingkat deteksi sisi jalan, analisa estimasi tingkat kesesuaian data rekomendasi putaran kemudi, dan analisa waktu proses rata-rata yang dibutuhkan.

#### 3.1 Analisa Data Tingkat Deteksi Sisi Jalan

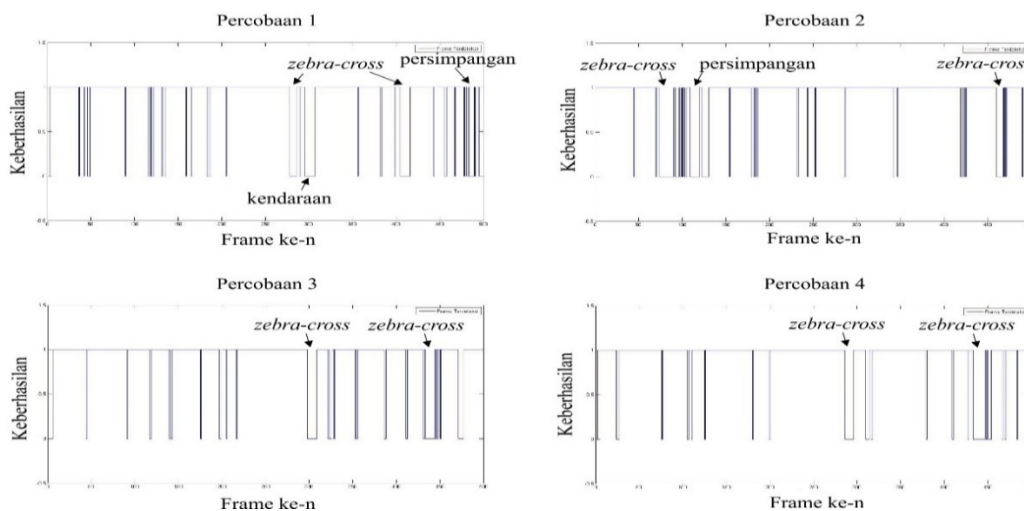
Tingkat deteksi sisi jalan ditentukan melalui banyaknya *frame* sisi jalan terdeteksi dari total 500 *frame* gambar yang diambil secara berurutan dalam satu kali percobaan. Data tingkat deteksi sisi jalan juga dikaitkan dengan banyaknya kendaraan, banyaknya persimpangan dan banyaknya area *zebra-cross* yang ada dalam suatu pengambilan data. Ketiga jenis rintangan tersebut dapat menyebabkan kegagalan deteksi jalan. Pengujian dilakukan di dalam lingkungan kampus dan terbatas pada lokasi tertentu yang telah diizinkan, dimana pada lokasi tersebut terdapat 2 buah area *zebra-cross*. Disebabkan pengujian dilakukan pada jalan umum, dalam beberapa percobaan ditemukan kendaraan lain yang turut melintas di jalan.

Percobaan dilakukan sebanyak 4 kali pada jalan raya beraspal yang memiliki 2 lajur dengan kondisi marka jalan yang tidak terlalu jelas. Percobaan dilakukan pada pagi hari dengan tingkat intensitas cahaya matahari yang sejuk dan kondisi jalan yang kering. Kecepatan kendaraan pada saat percobaan dipertahankan di sekitar 40 km/jam. Tingkat deteksi sisi jalan berikut dengan rintangan yang dilalui dituliskan pada tabel 2.

**Tabel 2** Tingkat Deteksi Sisi Jalan

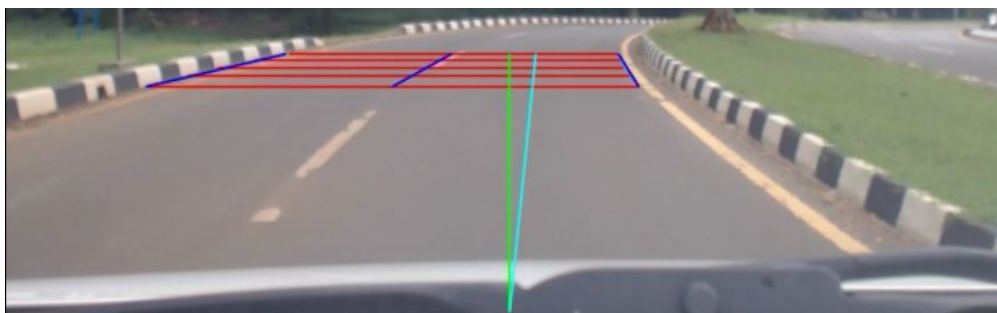
	Tingkat deteksi (%)	Jumlah <i>frame</i> terdeteksi	Banyak area persimpangan	Banyak kendaraan	Banyak area <i>zebra-cross</i>
Percobaan 1	88,4	442	1	1	2
Percobaan 2	88	440	1	0	2
Percobaan 3	90,6	453	0	1	2
Percobaan 4	91,8	459	0	0	2
Rata-rata	89,7	448,5	0,5	0,5	2

Pola deteksi sisi jalan oleh kamera ditunjukkan pada gambar 9 dengan keterangan sumbu-x menunjukkan urutan *frame* dari total 500 *frame* dan sumbu-y menunjukkan hasil deteksi dengan nilai 1 sebagai penanda sisi jalan terdeteksi dan 0 sebagai penanda sisi jalan gagal terdeteksi. Dari pola deteksi tersebut, terlihat bahwa sistem dapat mengalami kegagalan deteksi pada beberapa *frame* meskipun tidak ada rintangan berupa persimpangan, kendaraan, maupun *zebra-cross*. Dari hasil pengamatan pada beberapa *frame* gambar, diketahui bahwa hal ini disebabkan oleh variabel eksternal seperti faktor pencahayaan pada jalan, efek *autofocus* lensa kamera, maupun kesalahan sistem dalam mendeteksi batas jalan. Namun kegagalan deteksi tersebut tidak berpengaruh signifikan terhadap hasil rekomendasi putaran kemudi karena berlangsung dalam waktu yang sangat singkat dan terlalu cepat untuk dapat direspon oleh manusia.



Gambar 9 Pola Deteksi Sisi Jalan

Gambar 10 memperlihatkan sebuah *frame* pada percobaan dimana sisi jalan berhasil terdeteksi dengan baik. Garis merah pada gambar menandakan baris peninjauan batas jalan, garis biru menandakan sisi jalan dan pembatas jalan, garis hijau menandakan orientasi kamera, dan garis biru muda menandakan titik tujuan.



Gambar 10 Hasil Deteksi Sisi Jalan

### 3.2 Analisa Estimasi Tingkat Kesesuaian Data Rekomendasi Putaran Kemudi

Metode navigasi kendaraan dalam penelitian ini mampu mendeteksi jalan dan mengarahkan pengemudi menuju arah yang benar melalui rekomendasi putaran kemudi. Pengukuran tingkat kebenaran rekomendasi putaran kemudi yang diberikan oleh sistem dilakukan dengan mencari kesesuaian pergerakan kemudi yang dilakukan oleh pengemudi. Pada percobaan ini, pengemudi berhasil mempertahankan posisi kendaraan agar selalu berada di lajur

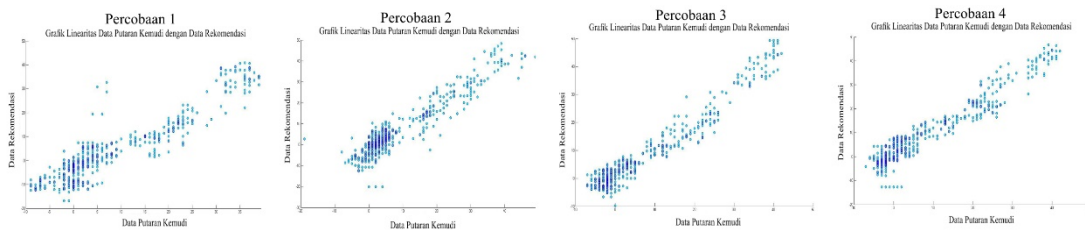
kanan jalan dan berhasil mengikuti jalan dengan baik, sehingga tingkat kesesuaian yang tinggi menandakan sistem mampu memberikan hasil yang sama dalam navigasi kendaraan. Tingkat kesesuaian pada tabel 3 dipengaruhi oleh tingkat keberhasilan deteksi sisi jalan dan kemampuan estimasi pengemudi dalam mengarahkan kendaraan ke tengah lajur kanan jalan.

Pengukuran tingkat kesesuaian dilakukan dengan mencari koefisien korelasi  $r$  menggunakan metode *Pearson Correlation Coefficient* pada persamaan (10), dimana  $r$  melambangkan koefisien korelasi;  $m, n$  adalah koordinat matriks;  $A, B$  adalah data yang dibandingkan;  $\bar{A}, \bar{B}$  adalah nilai rata-rata dari keseluruhan matriks tersebut. Teknik perhitungan koefisien korelasi yang digunakan mencari linearitas kedua data yang dibandingkan. Koefisien korelasi yang diperoleh memiliki rentang nilai antara  $-1 \leq r \leq +1$ . Agar dapat direpresentasikan dalam bentuk persentase, maka dilakukan kuadrasi pada  $r$ . Estimasi tingkat kesesuaian dalam bentuk persentase dituliskan pada tabel 3 dan grafik kesesuaian kedua data tersebut ditunjukkan pada gambar 11.

**Tabel 3** Estimasi Tingkat Kesesuaian Data Rekomendasi Putaran Kemudi

	Tingkat Kesesuaian (%)
Percobaan 1	86,3
Percobaan 2	89,42
Percobaan 3	95,9
Percobaan 4	92,54
Rata-rata	91,04

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (10)$$



**Gambar 11** Pola Linear Perbandingan Data Rekomendasi dan Putaran Kemudi

### 3.3 Analisa Waktu Proses Rata-Rata

Waktu proses yang diukur dalam penelitian adalah waktu yang dibutuhkan oleh komputer untuk mengolah 1 *frame* gambar, termasuk waktu proses algoritma deteksi sisi jalan dan perhitungan sudut rekomendasi. Tabel 4 menunjukkan waktu proses rata-rata terhadap 500 *frame* gambar untuk masing-masing percobaan.

**Tabel 4** Waktu Proses Rata-Rata

	Waktu proses rata-rata 500 <i>frame</i>
Percobaan 1	13,765
Percobaan 2	14,133
Percobaan 3	13,72
Percobaan 4	14,229
Rata-rata	13,962

#### 4. KESIMPULAN

Penelitian navigasi kendaraan yang dilakukan mampu menghasilkan sudut rekomendasi dengan waktu proses yang cukup singkat dengan rata-rata 13,962 ms dan tingkat keberhasilan deteksi sisi jalan rata-rata sebesar 89,7% yang dilakukan dengan melakukan peninjauan terhadap 5 baris pada gambar menggunakan metode *texture filter*, serta estimasi tingkat kesesuaian rekomendasi putaran kemudi rata-rata sebesar 91,04% yang dibandingkan dengan putaran kemudi oleh pengemudi. Penelitian yang dilakukan juga berhasil mengaplikasikan algoritma deteksi sisi jalan dan navigasi putaran kemudi ke dalam sebuah *Embedded PC* yaitu Raspberry Pi 2 dan perangkat keras yang portabel.

Keterbatasan pada metode yang digunakan dalam penelitian ini adalah seringnya terjadi kegagalan deteksi sisi jalan ketika area yang ditentukan terdapat kendaraan lain maupun marka jalan seperti *zebra-cross*. Pengembangan selanjutnya dapat dilakukan dengan perpaduan teknik klasifikasi objek dan penentuan area deteksi yang dinamis untuk meningkatkan keberhasilan deteksi jalan.

#### DAFTAR PUSTAKA

- [1] Q. Li, L. Chen, M. Li, S.-L. Shaw and A. Nuchter, "A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios," *IEEE Transactions on Vehicular Technology* (vol. 63), pp. 540-555, 2014.
- [2] F. Jimenez, J. E. Naranjo, O. Gomez and J. J. Anaya, "Vehicle Tracking for an Evasive Manoeuvres Assistant Using Low-Cost Ultrasonic Sensors," *Sensors*, pp. 22689-22705, 2014.
- [3] L. Hao, W. Jinfeng, L. Chao and W. Yi, "Research on Prediction of Vehicle Distance Based on Adaptive Kalman Filtering Principle," *International Conference on Measuring Technology and Mechatronics Automation*, pp. 38-40, 2017.
- [4] S. Godha, "On-road obstacle detection system for driver assistance," *Asia Pacific Journal of Engineering Science and Technology*, pp. 16-21, 2017.
- [5] M. Rezaee and Y. Zhang, "Road Detection Using Deep Neural Network In High Spatial Resolution Images," in *Urban Remote Sensing Event (JURSE)*, Dubai, 2017.
- [6] H. Luo, Y. Yang, B. Tong, F. Wu and B. Fan, "Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1-12, 2017.
- [7] W. Ouyang, X. Zeng and X. Wang, "Learning Mutual Visibility Relationship for Pedestrian Detection with a Deep Model," *International Journal of Computer Vision*, no. 1, pp. 14-27, 2016.
- [8] H. Maeda, Y. Sekimoto and T. Seto, "Lightweight Road Manager: Smartphone-based Automatic Determination of Road Damage Status by Deep Neural Network," in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, California, 2016.
- [9] J. Son, H. Yoo, S. Kim and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Systems with Applications*, pp. 1816-1824, 2015.
- [10] H. Kong, J. Y. Audibert and J. Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211 - 2220, April 2010.
- [11] M. Bertozzi, A. Broggi and A. Fascioli, "Stereo inverse prespective mapping: theory and applications," *Image and Vision Computing*, vol. 16, pp. 585-590, 1998.