

## BAB III

### RANCANGAN DAN IMPLEMENTASI SISTEM

#### 3.1. Standar NHTSA untuk Pengujian Kendaraan Otonom

Pada tugas akhir ini digunakan standar *National Highway Traffic Safety Administration* (NHTSA) sebagai standar utama yang mengatur mengenai kendaraan otonom. NHTSA merupakan badan keselamatan jalan raya Amerika Serikat. Standar NHTSA ini merupakan standar yang sering digunakan dalam observasi kendaraan otonom. Standar NHTSA mencakup sistem keamanan dari kendaraan otonom baik sistem keamanan *hardware* maupun sistem keamanan *software*. Selain itu, NHTSA juga mengatur mengenai keamanan pengujian dari kendaraan otonom.

Berikut hal-hal yang perlu diperhatikan untuk pengujian kendaraan otonom menggunakan standar NHTSA.

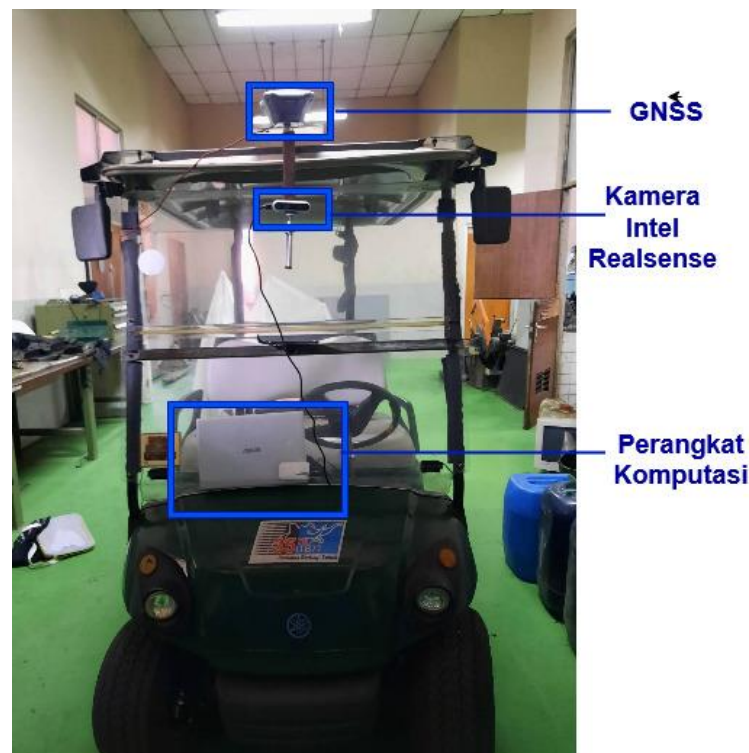
- a. Memastikan pengujian kendaraan otonom meminimalkan bahaya ke pengguna jalan yang lain
- b. Menyesuaikan kondisi pengujian lingkungan sesuai batasan dari kendaraan otonom yang diuji coba
- c. Memonitor performansi kendaraan otonom pada saat uji coba
- d. Memastikan bahwa transisi dari mode otomatis ke mode manual aman, sederhana dan cepat.
- e. Kendaraan otonom dapat mendeteksi, merekam dan menginformasikan kepada pengguna jika terjadi kesalahan pada sistem
- f. Memastikan sistem keamanan pada kendaraan otonom dalam kondisi aktif
- g. Memastikan kendaraan otonom merekam status informasi dari sistem kontrol pada saat terjadi kecelakaan

Kendaraan otonom yang digunakan pada penelitian memiliki sistem keamanan berupa *emergency button* agar kendaraan otonom dapat berhenti ketika terjadi masalah saat pengujian mode otonom.

Selain menggunakan standar NHTSA untuk standar pengujian. NHTSA juga digunakan sebagai standar akurasi untuk penentuan posisi pada tugas akhir ini yaitu penentuan posisi dan pemetaan dengan ORB-SLAM2. NHTSA memberikan standar untuk akurasi dari penentuan posisi [35] yaitu sebesar 1,500 meter.

### 3.2. Arsitektur Sistem Penentuan Posisi

Kendaraan otonom yang digunakan berupa mobil golf yang dimodifikasi sehingga dapat berjalan secara otonom. Energi yang digunakan pada mobil golf berasal dari empat baterai dengan total tegangan sekitar 48 Volt. Terdapat beberapa sensor yang akan digunakan pada tugas akhir ini yaitu kamera, GNSS dan takometer. Ilustrasi dari arsitektur kendaraan otonom beserta sensor-sensornya ditunjukkan pada Gambar 3.1.

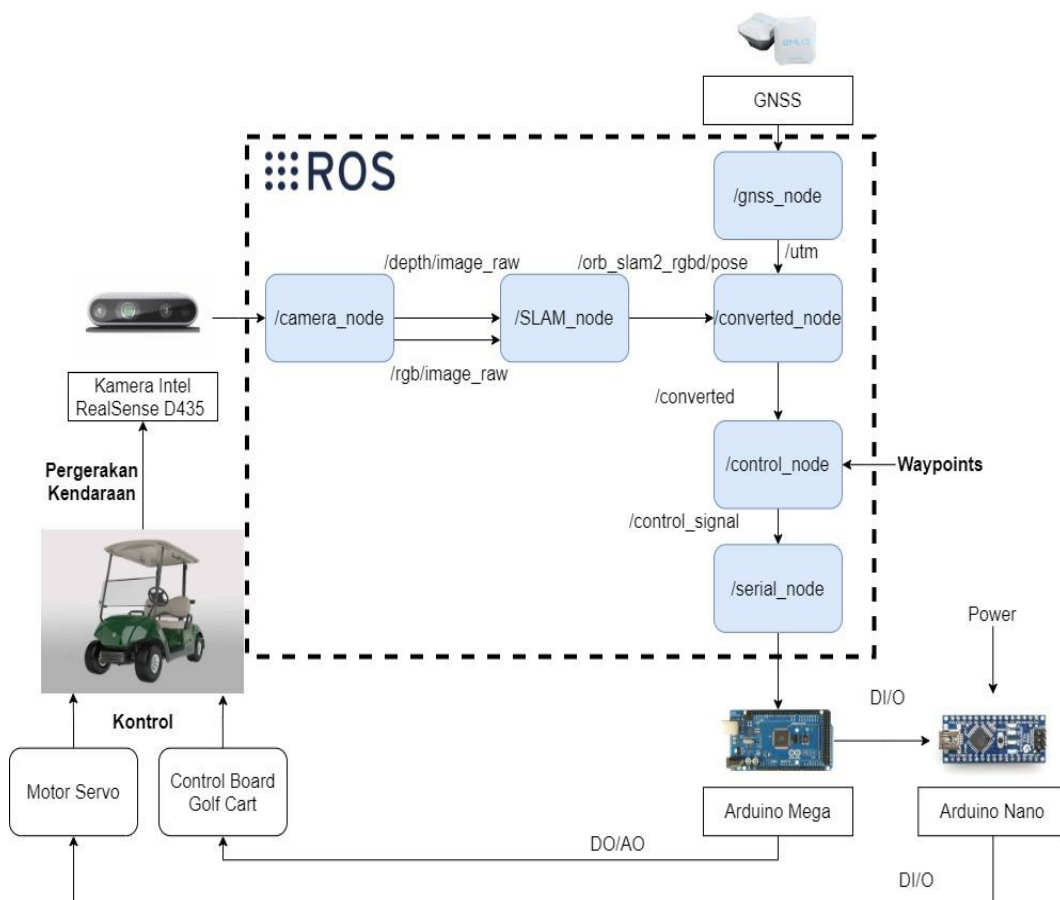


**Gambar 3.1.** Arsitektur mobil otonom

Kamera Intel RealSense merupakan sensor yang digunakan untuk mengimplementasikan algoritma ORB-SLAM2. Sedangkan untuk evaluasi

digunakan GNSS sebagai *ground truth* untuk evaluasi posisi dan kecepatan. GNSS memberikan estimasi posisi yang akurat jika digunakan pada kondisi lingkungan yang terbuka dan pada cuaca yang bagus. Selain itu, juga akan digunakan sensor takometer untuk mengukur kecepatan melalui perputaran dari roda. Hasil pengukuran dari takometer kemudian dibandingkan dengan hasil estimasi kecepatan dari ORB-SLAM2.

Perangkat lunak yang digunakan untuk pengambilan data dan untuk mengeksekusi program ke sensor adalah ROS (*Robot Operating System*). ROS memiliki beberapa kelebihan antara lain dapat menerima masukan informasi lebih dari satu sensor. Selain itu, ROS juga dapat melakukan eksekusi beberapa bahasa pemrograman yang berbeda seperti python dan C++. Gambar 3.2. memperlihatkan arsitektur *software* dan sensor yang digunakan untuk aplikasi ORB-SLAM2 pada kendaraan otonom.



**Gambar 3.2.** Arsitektur *software* dan sensor untuk aplikasi ORB-SLAM2

### 3.2.1. Mobil Golf

Kendaraan otonom yang digunakan pada tugas akhir ini berupa mobil golf Yamaha Golf Car YDRE 2011 (YDRE) yang telah dimodifikasi sehingga dapat bergerak secara otonom. Kendaraan otonom tersebut memiliki sumber energi dari baterai (listrik). Spesifikasi dari mobil golf ditunjukkan pada Tabel 3.1.

**Tabel 3.1.** Spesifikasi mobil golf

No	Spesifikasi	Detail
1	Dimensi	2395 x 1200 x 1190 mm
2	Berat (tanpa baterai)	270 kg
3	Kelajuan Maks.	24 km/jam
4	<i>Turning Radius</i>	2,8 m
5	Baterai	48 V
6	Steering	Rack & Pinion (1,85 Turn)
7	Throttle	Electric
8	Brake	Mechanical, internal transaxle disc-brake
9	Motor	3,7 HP Shunt-wound Brushless DC
10	Transmisi	Differential, rear-wheel drive

### 3.2.2. Perangkat Komputasi

Laptop Asus Vivobook digunakan sebagai perangkat komputasi pada proses pengambilan data, pengirim data serta eksekusi program. Spesifikasi laptop asus vivobook 14 tipe A412DA ditunjukkan pada Tabel 3.2.

**Tabel 3.2.** Spesifikasi Asus Vivobook 14 tipe A412DA

Spesifikasi	Keterangan
Prosesor	AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
RAM	8,00 GB
Penyimpanan	512GB SSD
Kartu Grafis	AMD Radeon Vega 8
Layar	14 Inch Full HD (1920x1080)

### 3.2.3. Kamera Intel RealSense D435

Sensor yang digunakan untuk menjalankan ORB-SLAM2 adalah kamera Intel RealSense D435, memiliki tiga lensa yang terdiri dari RGB, *infrared* (IR), dan *infrared* laser projector. Tiga lensa tersebut dimanfaatkan untuk mendeteksi

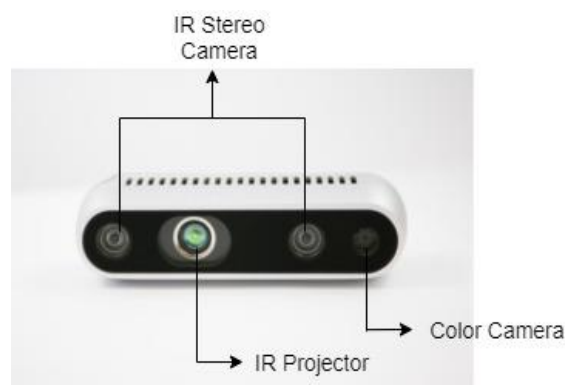
kedalaman dari objek dengan menerima pantulan sinar *infrared* dari objek. Spesifikasi dari kamera Intel RealSense D435 ditunjukkan pada Tabel 3.3.

**Tabel 3.3.** Spesifikasi kamera Intel RealSense D435

Dimensi	90 mm x 25 mm x 25 mm
Berat	72 gr
Teknologi Sensor Gambar (RGB)	<i>Global Shutter</i> , pixel $3\mu\text{m} \times 3\mu\text{m}$
Teknologi kedalaman ( <i>depth</i> )	Inframerah stereo aktif
RGB <i>Field of View</i>	$69,4^\circ \times 42,5^\circ (\pm 3^\circ)$
Kedalaman <i>Field of View</i>	$86^\circ \times 57^\circ (\pm 3^\circ)$
RGB <i>Frame Rate</i> Maksimum	30 fps
Kedalaman <i>Frame Rate</i> Maksimum	90 fps
Jarak minimum inframerah*	0.2 m
Jarak maksimum inframerah*	~10 m

Pada kamera Intel RealSense terdapat dua mode kamera yang digunakan yaitu stereo dan RGB-D. Dalam mode stereo, masukan yang diterima oleh algoritma ORB-SLAM2 adalah citra dari kamera stereo *infrared* kiri dan kamera stereo *infrared* kanan pada kamera Intel RealSense D435. Citra dari kamera RGB (*color camera*) tidak digunakan. Lensa-lensa yang digunakan pada mode stereo adalah lensa IR stereo kamera dan IR projector.

Dalam mode RGB-D, data yang digunakan adalah data kedalaman (*depth stream*) dan data kamera RGB pada kamera Intel RealSense D435. Data kedalaman didapatkan dari pengolahan kedua kamera stereo *infrared*. Ilustrasi dari lensa kamera Intel RealSense ditunjukkan pada Gambar 3.3.



**Gambar 3.3.** Visualisasi lensa Intel RealSense.

#### 3.2.4. GNSS EMLID Reach RS+

GNSS bekerja dengan akurat di lingkungan yang terbuka (tidak tertutup oleh gedung dan pohon) dan pada penelitian ini GNSS digunakan sebagai *ground truth* dari hasil penentuan posisi ORB-SLAM2. GNSS bekerja diawali dengan satelit yang melakukan *pinpoint* ke GNSS *receiver*. Kemudian, satelit mengirim sinyal berupa posisi dan waktu ke GNSS *receiver*. Spesifikasi dari GNSS yang digunakan terlampir pada Tabel 3.4. Sedangkan gambar dari GNSS ditunjukkan pada Gambar 3.4.

**Tabel 3.4.** Spesifikasi GNSS EMLID Reach RS+

Spesifikasi	Keterangan
Dimensi	145 mm x 145 mm x 85 mm
Massa	690 g
<i>Band frekuensi</i>	<i>Single-band</i>
Radio	LoRa 868/915 MHz
Sinyal GNSS tersedia	GPS/QZSS L1, GLONASS G1, BeiDou B1, Galileo E1, SBAS
<i>Tracking channels</i>	72
Frekuensi cuplik	5 Hz



**Gambar 3.4.** GNSS (*Global Navigation Satellite System*)

#### 3.2.5. Takometer

Takometer digunakan untuk mengukur kecepatan dari mobil golf dengan menghitung rotasi dari roda kendaraan otonom. Takometer yang digunakan terpasang di mobil golf dari awal. Pada uji coba yang dilakukan, hasil pengukuran dari takometer akan dibandingkan dengan hasil estimasi kecepatan ORB-SLAM2.

### 3.3. Package ROS yang Digunakan

Package dalam ROS merupakan *folder* dari ROS yang dapat berisi sekumpulan program. Package dari ROS yang digunakan pada tugas akhir ini antara lain

- nmea\_navsat\_driver  
Package nmea\_navsat\_driver mengolah data GNSS menjadi koordinat LLA. Koordinat LLA dapat diakses di topik /fix yang berisi *message* tipe sensor\_msgs/NavSatFix.
- gps\_common  
Package gps\_common mengkonversi koordinat LLA yang *publish* oleh package nmea\_navsat\_driver ke koordinat UTM dalam sumbu x dan sumbu y. Hasil konversi dapat diakses di topik /utm yang memiliki tipe *message* nav\_msgs/Odometri
- roserial\_python  
Package roserial\_python berfungsi untuk menghubungkan ROS dengan arduino.
- orb\_slam2\_ros  
Package orb\_slam2\_ros merupakan program ORB-SLAM2 yang digunakan. Hasil dari package ini adalah penentuan posisi berupa sumbu x,y,z dan sudut quaternion. Hasil penentuan posisi dapat diakses di topik /orb\_slam2\_rgbd/pose untuk mode rgbd dan /orb\_slam2\_stereo/pose untuk mode stereo.
- realsense\_ros  
Package realsense\_ros merupakan package yang menghubungkan kamera dan ROS. Hasil gambar *depth* dan RGBD dapat diakses pada topik /camera/color/image\_raw dan topik /camera/depth/image\_rect\_raw.
- rviz  
Package rviz menyediakan fitur yang dapat digunakan untuk menampilkan gambar yang diolah oleh ORB-SLAM2.
- golfi

Package golfi berisi program untuk me-*launch* GNSS dan UKF. Topik yang dapat diakses untuk memperoleh nilai estimasi UKF adalah dengan menggunakan topik /ukf\_states

- pkg\_ta

pkg\_ta berisi program untuk menjalankan kontrol kendaraan secara otonom

### 3.4. Implementasi Penggunaan ORB-SLAM2

Jika ditinjau secara umum, terdapat tiga tahap dalam proses implementasi penggunaan ORB-SLAM2 pada kendaraan otonom yang digunakan yaitu pembuatan peta, kalibrasi parameter untuk transformasi koordinat peta ke koordinat GNSS dan terakhir penentuan posisi atau lokalisasi menggunakan peta yang telah tersimpan. Pada proses pemetaan, dilakukan pengumpulan informasi posisi dan *frame* untuk membentuk peta. Hasil pemetaan kemudian disimpan dan nantinya akan dimuat kembali untuk proses penentuan posisi. Proses lokalisasi sebenarnya bisa dilakukan secara langsung tanpa menyimpan peta terlebih dahulu. Namun, hasil lokalisasi tanpa penyimpanan peta terlebih dahulu memberikan hasil pengukuran yang kurang presisi dibanding dengan melakukan penyimpanan peta terlebih dahulu. Pada Gambar 3.5, terdapat diagram alur implementasi dari ORB-SLAM2.



**Gambar 3.5.** Diagram alur implementasi dari ORB-SLAM2

#### 3.4.1. Pemetaan

Terdapat beberapa hal yang perlu dipertimbangkan dalam melakukan pemetaan yaitu banyaknya informasi yang masuk serta menghindari kondisi *track-loss* yaitu kondisi dimana ORB-SLAM2 kehilangan konektivitas antara *frame* yang sekarang dengan *frame* yang sebelumnya. Semakin banyak informasi yang masuk, semakin akurat pula peta yang dihasilkan. Untuk mendapatkan informasi yang banyak,

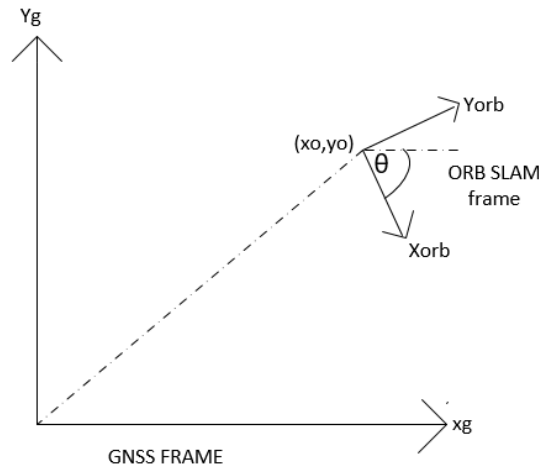


pemetaan dilakukan dengan kecepatan yang pelan. Hal ini akan membuat jumlah *frame* pada suatu lingkungan yang masuk ke kamera lebih banyak.

Selain itu, pada proses pemetaan dapat dilakukan pemetaan berulang kali pada suatu tempat yang sama. Semakin banyak pengulangan semakin akurat peta yang dihasilkan. Hal ini dikarenakan pada ORB-SLAM2 terdapat algoritma optimisasi jika terjadi pemetaan *closed loop*. *Closed loop* terjadi apabila pemetaan lingkungan kembali di titik dimana telah dilakukan pemetaan sebelumnya. Pemetaan perlu dilakukan setiap sebelum dilakukan implementasi ORB-SLAM2 karena variasi kondisi pencahayaan disetiap objek pada lingkungan dan karena perubahan objek pada lingkungan.

### 3.4.2. Kalibrasi Parameter untuk Mentransformasikan Koordinat ORB-SLAM2 ke Koordinat GNSS

Pengubahan koordinat ORB-SLAM2 ke koordinat GNSS dilakukan karena algoritma kontrol yang digunakan adalah menggunakan koordinat GNSS. Pengubahan koordinat GNSS dilakukan pada setiap pembuatan peta. Hal ini dikarenakan koordinat GNSS berubah-ubah setiap periode waktu tertentu. Koordinat dari ORB-SLAM2 dan GNSS diilustrasikan seperti pada Gambar 3.6.

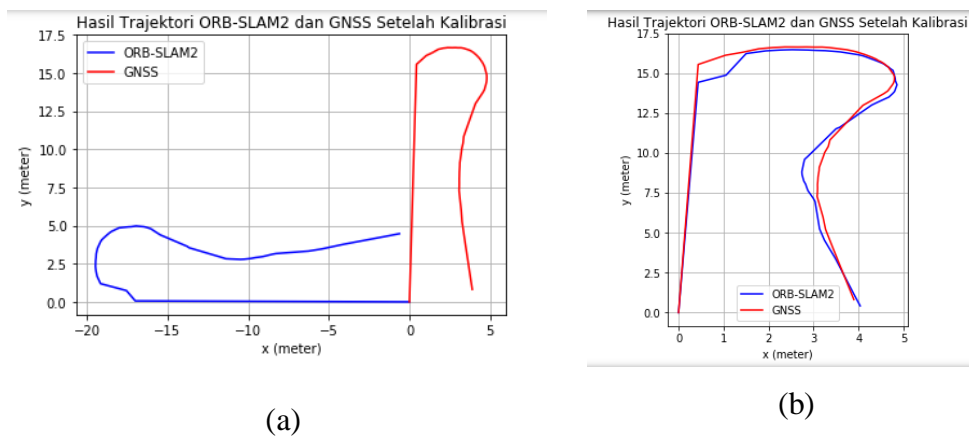


**Gambar 3.6.** Koordinat GNSS dan Koordinat ORB-SLAM2

Transformasi dilakukan dengan menggunakan persamaan sebagai berikut

$$\begin{pmatrix} x_g \\ y_g \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x_{orb} \\ y_{orb} \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \quad (3.1)$$

Dengan  $(x_g, y_g)$  adalah koordinat SLAM yang telah diubah ke dalam koordinat GNSS,  $(s_x, s_y)$  adalah skala pada sumbu x dan pada sumbu y,  $\theta$  adalah sudut antara koordinat GNSS dan koordinat SLAM,  $(x_{orb}, y_{orb})$  adalah koordinat SLAM sebelum ditransformasikan, dan  $(x_0, y_0)$  adalah titik (0,0) koordinat SLAM dalam bentuk koordinat GNSS. Kalibrasi diawali dengan pengambilan data trajektori GNSS dan data trajektori ORB-SLAM2. Parameter yang dilakukan kalibrasi adalah parameter  $\theta$  dan  $(S_x, S_y)$  sehingga trajektori GNSS dan trajektori ORB-SLAM2 berhimpit. Perubahan koordinat sebelum dan sesudah transformasi ditunjukkan pada Gambar 3.7.



**Gambar 3.7.** Trajektori sebelum ditransformasikan (a) dan setelah ditransformasikan (b)

### 3.4.3. Penentuan Posisi

Langkah akhir dan tujuan utama dari implementasi ORB-SLAM2 adalah penentuan posisi. Penentuan posisi dilakukan dengan memuat peta yang telah disimpan. Hasil penentuan posisi dari ORB-SLAM2 kemudian diubah ke dalam koordinat GNSS menggunakan hasil kalibrasi yang sebelumnya telah dilakukan pada sebuah program.

### 3.5. Sistem Kontrol Kendaraan Otonom

Kontrol longitudinal dari kendaraan otonom diatur oleh sistem traksi dan pengereman sedangkan kontrol lateral dari kendaraan otonom diatur oleh sistem penyetiran. Tipe kontrol yang digunakan antara lain *high level controller* dan *low level controller* [36]. *High level controller* menerima persepsi dari mobil otonom kemudian membandingkannya dengan keadaan yang ingin dicapai kemudian

memberikan perintah keadaan yang perlu dicapai oleh *low level controller* berupa sudut setir, posisi rem dan gas.

Sistem *high level controller* terletak pada program python sedangkan *low level controller* ditanamkan pada arduino mega 2560. *Software* yang digunakan adalah ROS (*Robot Operating System*). ROS dapat menerima masukan data dari beberapa sensor dan mengirim perintah baik arduino maupun sensor lainnya. Umpan balik yang digunakan pada tugas akhir ini adalah hasil penentuan posisi dari ORB-SLAM2. Sistem penentuan posisi dari ORB-SLAM2 yang digunakan adalah posisi, kecepatan dan yaw. GNSS masih digunakan namun hanya untuk diambil data awalnya saja guna mentransformasikan koordinat ORB-SLAM2 ke koordinat GNSS.

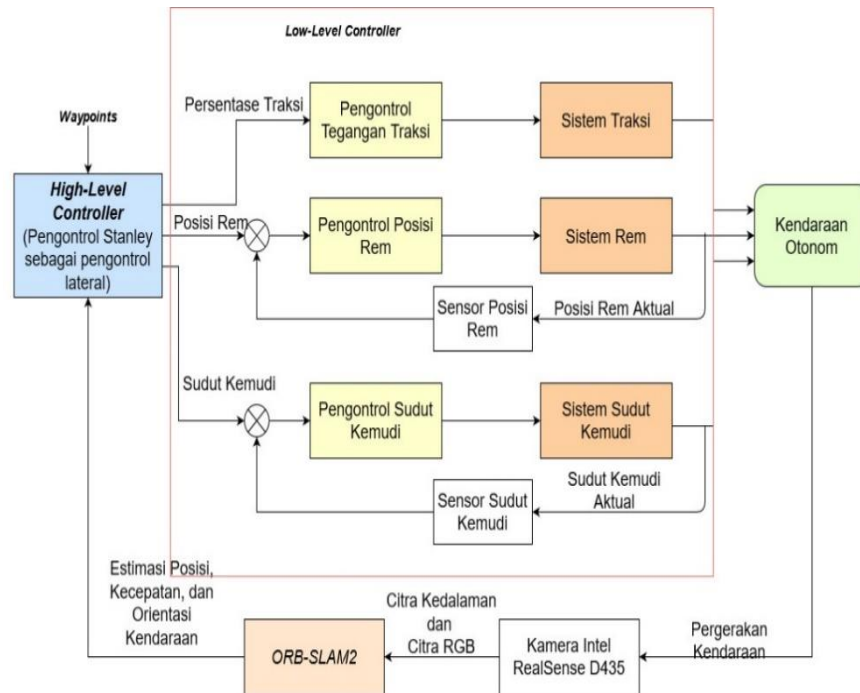
Input kontrol lateral seberapa besar sudut setir yang diterapkan dihitung menggunakan persamaan Stanley dengan asumsi *setpoint* orientasi konstan. Persamaan yang digunakan adalah ditunjukkan sebagai berikut dengan  $l_h, \kappa_t \in [-1, 1]$  dan  $\delta_t \in [-35^\circ, 28^\circ]$ ,  $\kappa_T$  adalah kurvatur jalur,  $e(t)$  adalah jarak antara mobil otonom dan jalur,  $v_f$  adalah kecepatan yang searah dengan sudut setir sedangkan  $l_h$  jarak antara roda depan dan belakang.

$$\delta_{t,s} = \arcsin(l_h, \kappa_{T,i_t}) + \psi_t + \arctan\left(\frac{ke(t)}{v_f}\right) \quad (3.2)$$

Sedangkan untuk pengontrol longitudinal menggunakan persamaan sebagai berikut dengan  $e_v \in \mathbb{R}$  adalah galat kelajuan dan  $E_v \in \mathbb{R}$  adalah nilai integral dari  $e_v$ ,  $u_k$  adalah input dari sistem traksi dan rem.

$$u_k = k_p e_{v,k} + k_i E_{v,k} + k_d \frac{e_{v,k} - e_{v,k-1}}{\Delta T} \quad (3.3)$$

Pada Gambar 3.8, terdapat diagram blok hubungan ORB-SLAM2 dengan sistem kontrol.



**Gambar 3.8.** Diagram blok hubungan ORB-SLAM2 dengan pengontrol pada kendaraan