

IMPLEMENTASI ALGORITMA *SIMULTANEOUS LOCALIZATION AND MAPPING* BERBASIS *COMPUTER VISION* UNTUK PEMETAAN JALUR PADA KENDARAAN OTONOM

IMPLEMENTATION OF *SIMULTANEOUS LOCALIZATION AND MAPPING* ALGORITHM BASED ON *COMPUTER VISION* FOR PATH MAPPING IN *AUTONOMOUS VEHICLES*

Jefri Salmon Tande
Prodi S1 Teknik Elektro
Fakultas Teknik Elektro
Universitas Telkom

Fiky Yosef Suratman
Prodi S1 Teknik Elektro
Fakultas Teknik Elektro
Universitas Telkom

Arief Suryadi Satyawan
Prodi S1 Teknik Elektro
Fakultas Teknik Elektro
Universitas Telkom

Abstrak

Kendaraan otonom merupakan sistem transportasi cerdas yang mampu bergerak tanpa kendali langsung manusia. Salah satu tantangan utama dalam pengembangan kendaraan ini adalah bagaimana membangun peta lingkungan serta menentukan posisinya secara bersamaan. Untuk menjawab tantangan ini, algoritma *Simultaneous Localization and Mapping* (SLAM) berbasis visual, khususnya ORB-SLAM2, diimplementasikan pada modul *JetRacer AI Kit* dengan bantuan kamera monokular IMX 219-160. Metode yang digunakan mencakup studi literatur, perancangan perangkat keras dan lunak, implementasi algoritma, serta pengujian sistem. Algoritma ORB-SLAM2 dijalankan pada sistem *Jetson Nano* untuk memproses citra dari kamera dan menghasilkan pemetaan 3D serta estimasi lintasan kamera. Pengujian dilakukan dalam dua kondisi pencahayaan (terang dan redup) dengan variasi parameter *nFeatures* untuk menganalisis performa deteksi fitur, waktu inisialisasi, serta jumlah *map points* yang dihasilkan. Hasil penelitian menunjukkan bahwa algoritma ORB-SLAM2 mampu untuk membentuk lingkungan secara *real-time* dengan tingkat akurasi dan efisiensi yang tinggi. Pada kondisi terang, performa optimal tercapai pada *nFeatures* 700-800 dengan *map points* tertinggi sebanyak 1734 dan waktu inisialisasi 50 ms. Sedangkan pada kondisi redup, performa terbaik tercapai pada *nFeatures* 900 dengan 2543 *map points* dan efisiensi deteksi mencapai 99,89 %.

Kata Kunci : *Computer Vision, JetRacer AI Kit, Kendaraan Otonom, ORB Feature Detection, ORB-SLAM2, Visual SLAM.*

Abstarct

Autonomous vehicles are intelligent transportation systems capable of moving without direct human control. One of the main challenges in developing such vehicles is the ability to simultaneously map the environment and determine their own position. To address this issue, a visual-based Simultaneous Localization and Mapping (SLAM) algorithm, specifically ORB-SLAM2, was implemented on the JetRacer AI Kit module using a monocular camera (IMX 219-160). The research methods include literature studies, hardware and software design, algorithm implementation, and system testing. ORB-SLAM2 was executed on the Jetson Nano Platform to process image data from the camera and generate 3D

mapping along with camera trajectory estimation. Tests were conducted under two lighting conditions (bright and dim) with various nFeatures parameters to analyze feature detection performance, initialization time, and the number of generated map points. The results show that ORB-SLAM2 can reconstruct environment in real time with high accuracy and efficiency. Under bright conditions, optimal performance was achieved at nFeatures 700-800, producing up to 1734 map points with a 50 ms initialization time. Under dim lighting, the best performance was observed at nFeatures 900, generating 2543 map points with a feature detection efficiency of 99,89%.

Keywords : *Computer Vision, Jetracer AI Kit, Autonomous Vehicle, ORB Feature Detection, ORB-SLAM2, Visual SLAM.*

1. Pendahuluan

Kendaraan otonom (autonomous vehicle) merupakan kendaraan yang dapat beroperasi secara mandiri, tanpa memerlukan pengendalian langsung dari manusia untuk bergerak menuju tujuan yang telah ditetapkan. Supaya kendaraan otonom dapat beroperasi tanpa kendali langsung dari manusia, kendaraan otonom diharuskan mampu menganalisis situasi sekitar, memperkirakan arah pergerakan objek-objek dalam jangkauannya, dan merencanakan jalur yang aman menuju tujuan akhir dengan menghindari segala halangan yang ada [1].

Permasalahan yang sering dihadapi dari kendaraan otonom adalah bagaimana kendaraan otonom dapat melaju ke tempat tujuan tanpa menimbulkan kerusakan pada area sekitar. Tantangan terbesar pada kendaraan otonom adalah memahami peta lingkungan yang dilalui selama beroperasi, serta mampu untuk menentukan posisinya sendiri pada peta [2]. Solusi dari permasalahan ini adalah kendaraan otonom perlu membangun peta lingkungannya sendiri. Untuk membangun peta tersebut, diperlukan lokalisasi untuk mengetahui posisi kendaraan, dan algoritma *Simultaneous Localization and Mapping* (SLAM) memungkinkan kedua tugas ini dilakukan secara bersamaan [3].

SLAM merupakan metode untuk robot dan kendaraan otonom lainnya untuk membuat peta dari lingkungan yang belum dikenal sambil secara bersamaan menentukan posisi mereka di dalam lingkungan tersebut [4]. Dengan mendapatkan informasi dari sensor visual kamera monokular, dapat digunakan untuk pemetaan dan menentukan posisi kendaraan otonom itu sendiri. ORB-SLAM (*Oriented FAST and Rotated BRIEF SLAM*) adalah algoritma SLAM yang mencocokkan fitur, memperkirakan posisi kamera, dan membangun struktur 3D dari lingkungan dengan

menggunakan karakteristik titik-titik kunci yang diambil dari gambar kamera. Bekerja dalam tiga proses yaitu pelacakan, lokalisasi, dan *close loop*, ORB-SLAM mampu memetakan area sekitar kendaraan otonom [5-6].

Topik dalam penelitian tugas akhir ini adalah implementasi algoritma ORB-SLAM pada kendaraan otonom berupa modul *JetRacer AI Kit*, untuk memetakan lingkungan sekitar yang belum diketahui oleh kendaraan otonom. Diharapkan dengan penggunaan algoritma ORB-SLAM yang diintegrasikan dengan kamera monokular modul *JetRacer AI Kit* dapat memenuhi tujuan penulis dalam pembuatan tugas akhir ini.

2. Kajian teori

2.1 Computer Vision

Dengan pemanfaatan kecerdasan buatan terbaru, sistem komputer memiliki kemampuan untuk memahami dunia visual dengan lebih baik, bahkan melebihi kemampuan manusia dalam beberapa tugas tertentu. Kemampuan ini berasal dari sistem computer vision yang terdiri dari algoritma dan perangkat keras canggih, yang memungkinkan melatih komputer untuk mengerti dunia fisik melalui penglihatan [7]. Tujuan utama dari computer vision adalah untuk membuat komputer dapat memahami dunia visual seperti halnya manusia. Pengembangan algoritma computer vision mengupayakan komputer untuk mengenali objek, menentukan lokasi, dan memahami keseluruhan isi sebuah gambar.

2.2 SLAM (*Simultaneous Localization and Mapping*)

Simultaneous Localization And Mapping (SLAM) merupakan suatu teknologi dimana dapat membangun peta dalam lingkungan yang tidak diketahui dan melokalisasi sensor di peta dengan fokus pada operasi secara real time [8]. Diantara berbagai sensor yang digunakan sebagai sumber informasi, kamera merupakan sensor yang sangat aktif dibahas, dengan alasan kamera menyediakan informasi lingkungan yang kaya, dan memungkinkan pengenalan tempat yang kuat dan akurat. Metode ini secara khusus disebut sebagai visual SLAM (vSLAM) dikarenakan input dari SLAM berupa informasi visual [9].

2.3 ORB-SLAM

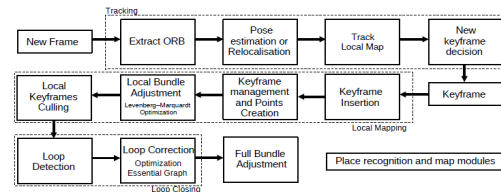
ORB-SLAM (Oriented FAST and Rotated BRIEF SLAM) memiliki struktur yang terdiri dari tiga unit eksekusi yang bekerja secara paralel Tracking, Local Mapping, dan Loop Closing. Sistem ORB-SLAM ini menerima input berupa gambar, dimana tugas pertama adalah untuk mengekstraksi fitur ORB. Proses dimulai dengan menemukan fitur tepi dalam frame input menggunakan metode ekstraksi fitur yang disebut FAST (Features from Accelerated Segment Test). Langkah berikutnya membuat deskriptor untuk fitur-fitur tersebut dengan menggunakan metode yang disebut BRIEF (Binary Robust Independent Elementary Features). Penggunaan deskriptor BRIEF adalah metode pendekatan langsung dengan menghitung string biner dari patch gambar yang diekstraksi.

2.4 ORB-SLAM2

ORB-SLAM2 merupakan pengembangan dari

algoritma ORB-SLAM, dapat dikatakan sebagai algoritma SLAM berbasis fitur yang terbaik diantara algoritma yang lain. Algoritma ORB-SLAM2 bekerja dalam tiga unit eksekusi yaitu pelacakan, pemetaan lokal, dan *close loop*. Algoritma ORB-SLAM2 bekerja dalam tiga unit eksekusi yaitu pelacakan, pemetaan lokal, dan *close loop*. Unit eksekusi yang pertama adalah pelacakan, bertugas untuk menemukan lokasi sensor dengan mencari korespondensi fitur dan mengurangi kesalahan

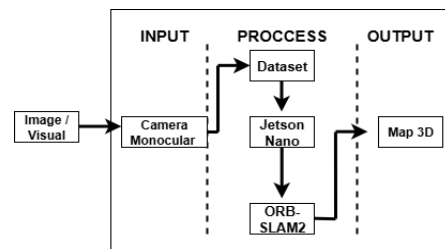
proyeksi ulang. Unit kedua merupakan pemetaan lokal, bertanggung jawab mengenai operasi manajemen pada peta. Unit terakhir, *close loop* memiliki tugas mendeteksi loop baru dan memperbaiki kesalahan drift dalam loop. Gambar 1, merupakan diagram yang merepresentasikan algoritma ORB-SLAM2



GAMBAR 1
Diagram Algoritma ORB-SLAM2

3. Metode

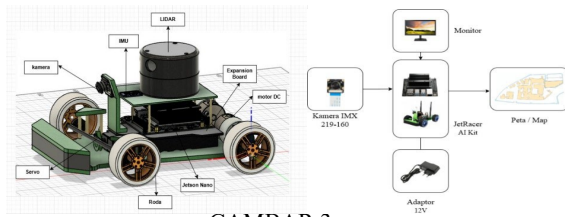
3.1 Diagram Blok Sistem



GAMBAR 2
Diagram Blok Sistem

Gambar 2, menunjukkan diagram blok sistem pemetaan pada kendaraan otonom menggunakan algoritma ORB-SLAM2, yang akan diimplementasikan pada modul *JetRacer AI Kit*. Input atau masukan dari pengembangan sistem ini merupakan citra atau visual. Citra atau visual yang diperoleh kamera monokular akan dikumpulkan menjadi satu dataset. Dataset yang telah dikumpulkan melalui kamera pada kendaraan otonom, menjadi informasi bagi algoritma ORB-SLAM2 untuk dapat memproses informasi lingkungan. Dengan mencocokkan fitur dari frame ke frame secara berurutan, Algoritma ORB-SLAM2 mampu untuk memperkirakan posisi secara akurat dan bersamaan membangun peta lingkungan sekitar kendaraan otonom.

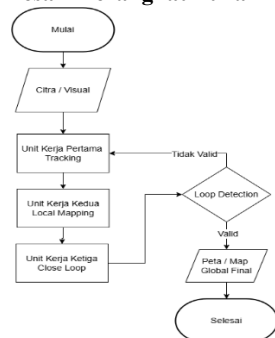
3.2 Desain Perangkat Keras



GAMBAR 3
Desain Perangkat Keras Sistem

Gambar 3, memperlihatkan desain perangkat keras sistem pemetaan kendaraan otonom. Pada sistem ini, kamera berfungsi sebagai sensor utama dalam menangkap data citra atau visual. Data yang diperoleh dari kamera dikirimkan ke jetson nano sebagai mikrokomputer untuk memproses algoritma ORB-SLAM2. Jetson nano terhubung dengan baterai dan sasis kendaraan otonom yang lain untuk membantu kamera dalam memperoleh citra atau visual. Selain itu, perangkat seperti monitor dan adaptor turut serta dalam menampilkan informasi pemrosesan dan peta hasil secara visual.

3.3 Desain Perangkat Lunak



GAMBAR 4
Flowchart sistem pemetaan ORB-SLAM2

Berdasarkan gambar 4, menggambarkan proses algoritma SLAM (Simultaneous Localization and Mapping) yang dimulai dari input berupa citra atau visual dari kamera. Proses ini terdiri dari beberapa langkah, Tracking mengekstraksi fitur penting dari gambar untuk memperkirakan posisi kamera saat ini. Jika pelacakan berhasil, sistem melanjutkan ke langkah berikutnya, tetapi jika tidak, sistem mencoba memperbaiki pelacakan. Local Mapping memperbarui peta lokal dengan menambahkan keyframe baru dan titik-titik peta, serta mengoptimalkan peta lokal. Loop Detection memeriksa apakah kamera kembali ke lokasi yang pernah dilihat sebelumnya. Jika loop terdeteksi, sistem menghitung dan memvalidasi transformasi loop. Terakhir, Close Loop memperbaiki peta global jika loop valid, untuk mengurangi kesalahan dan memastikan konsistensi. Hasil akhirnya adalah peta global yang akurat dan dapat digunakan untuk kendaraan otonom dalam menuju tempat tujuan.

4. Hasil dan Pembahasan

4.1 Pengujian Kalibrasi Kamera IMX 219-160

Dalam pengujian ini, kalibrasi kamera IMX 219-

160 dilakukan untuk mendapatkan parameter intrinsik (matriks kamera) dan koefisien distorsi supaya meningkatkan akurasi sistem vision komputer. Matriks kamera yang dihasilkan mencakup focal length (f_x , f_y) dan principal point (c_x , c_y), sedangkan koefisien distorsi (k_1 , k_2 , p_1 , p_2 , k_3) digunakan untuk memperbaiki distorsi radial dan tangensial. Proses kalibrasi menggunakan pola chessboard dengan ukuran 7×5 . Dalam percobaan yang dilakukan mengambil sebanyak 50 gambar yang diperlukan dalam kalibrasi dengan ukuran 640×480 piksel. Jetracer AI Kit yang telah dilengkapi dengan kamera IMX 219 – 160, dihadapkan pada papan kalibrasi dengan berbagai jarak dimulai dari jarak 20 cm, 25 cm, 30 cm, 35 cm, dan sampai 40 cm. Hasil pengujian kalibrasi kamera IMX 219-160 didapatkan sebagai berikut :

$$K = \begin{bmatrix} 246,04 & 0 & 305,10 \\ 0 & 328,82 & 253,88 \\ 0 & 0 & 1 \end{bmatrix}$$

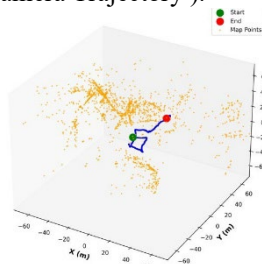
$$D = [-0,523 \quad 1,232 \quad -0,003 \quad -0,008 \quad -1,33]$$

Berdasarkan dari nilai matriks kamera dan koefisien distorsi dari 50 gambar yang telah dikalibrasi, dapat mengetahui kamera dapat berjalan dan berfungsi sesuai dengan standar yang telah ditetapkan pada kamera IMX 219-160. Pertama dimulai dari nilai focal length (f_x , f_y) atau panjang fokus dalam satuan piksel untuk suatu kamera, dari data yang didapatkan nilai focal length arah horizontal (x) sebesar 246,04 dan vertikal (y) sebesar 328,82, dimana nilai tersebut memiliki perbedaan yang cukup signifikan antara nilai f_x dan f_y . Nilai focal length atau panjang fokus pada kamera tidak harus sama dan tidak terlalu jauh selisih antara nilai keduanya, sehingga hasil yang didapatkan masih dapat diterima sebagai batas normal dalam parameter kalibrasi kamera IMX 219-160.

Dengan nilai dari data yang dihasilkan oleh kalibrasi yaitu sebesar c_x sebesar 305,10 dan c_y sebesar 253,88 nilai principal point termasuk dalam kategori sangat baik dan berada pada tengah frame. Selanjutnya dalam parameter koefisien distorsi dari kalibrasi yang dilakukan mencakup nilai k_1 , k_2 , p_1 , p_2 , k_3 , terbagi menjadi 2 bagian yaitu koefisien distorsi radial (k_1 , k_2 , k_3) dan koefisien distorsi tangensial (p_1 , p_2). Dari kalibrasi yang telah dilakukan, mendapatkan data nilai koefisien distorsi radial berturut-turut $k_1 = -0,523$, $k_2 = 1,232$, dan $k_3 = -1,330$, nilai yang dihasilkan cukup besar untuk kamera menghasilkan gambar melengkung kedalam maupun keluar. Sementara untuk koefisien distorsi tangensial memiliki nilai $p_1 = -0,003$ dan $p_2 = -0,0087$, nilai yang dihasilkan dari kalibrasi termasuk kecil, menandakan bahwa lensa sejajar dengan sensor kamera IMX 219-160.

4.2 Pengujian Algoritma ORB-SLAM2

Pengujian ini bertujuan untuk menguji algoritma ORB-SLAM2 dalam proses pemetaan lingkungan sekitar yang ditangkap oleh kamera monokular. Pada pengujian ini, akan menghasilkan keluaran peta 3D dan lintasan lingkungan sekitar dari pergerakan kamera monokular. Jetracer AI Kit yang dilengkapi dengan kamera monokular diuji dalam lingkungan indoor. Kamera tersebut menangkap citra lingkungan secara real-time, yang kemudian diproses menggunakan algoritma ORB-SLAM2 untuk menghasilkan rekonstruksi peta 3D (Map Points) serta melacak lintasan pergerakan kamera (Camera Trajectory).



GAMBAR 5

Peta 3D Map algoritma ORB-SLAM2

Berdasarkan hasil pengujian yang ditunjukkan pada gambar 5, ORB-SLAM2 berhasil merepresentasikan lingkungan melalui rekonstruksi map points dan estimasi posisi kamera. Representasi map points yang dihasilkan oleh algoritma ini menjadi dasar dalam pelacakan estimasi posisi kamera. Untuk optimasi akurasi pemetaan area, diperlukan implementasi loop detection untuk meminimalisir akumulasi error pada rekonstruksi 3D lingkungan.

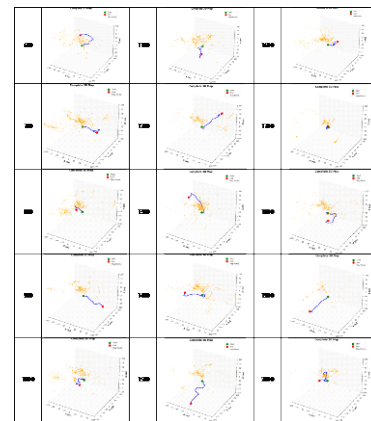
4.3 Pengujian Parameter ORB Feature Detection Pada Lingkungan Indoor

Pengujian ini bertujuan untuk menganalisis variasi parameter nFeatures dalam ORB Feature Detection terhadap performa sistem ORB-SLAM2 dalam merekonstruksi lingkungan indoor dengan dua kondisi pencahayaan, dengan fokus pada optimasi kualitas deteksi fitur dan efisiensi inisialisasi sistem. Pada skenario pengujian ini, Jetracer AI Kit yang telah dilengkapi dengan kamera monokular akan melewati suatu lintasan berbentuk huruf U yang disiapkan di dalam ruangan uji. Pengujian dilakukan dalam dua kondisi pencahayaan yaitu terang dan redup. Parameter nilai nFeatures divariasikan dari 600 hingga 2000 dengan interval 100, sementara scale factor 1.2 dan nLevel adalah 8 untuk seluruh percobaan.

TABEL 1

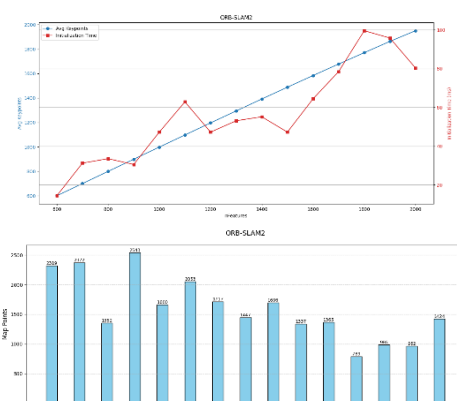
Tabel Feature ORB Detection kondisi terang

nFeatures	Avg Keypoints	Map Points	Initialization Time
600	599,366	1165	23,13 ms
700	698,747	1724	28,12 ms
800	797,798	1734	34,72 ms
900	896,056	1231	51,01 ms
1000	993,821	1049	38,08 ms
1100	1089,921	1667	43,49 ms
1200	1185,57	1171	63,73 ms
1300	1278,264	1437	62,01 ms
1400	1369,189	1341	48,09 ms
1500	1454,677	1101	62,77 ms
1600	1534,032	1399	78,51 ms
1700	1608,535	683	79,24 ms
1800	1679,135	1446	76,38 ms
1900	1746,972	473	101,68 ms
2000	1810,536	1196	98,21 ms



GAMBAR 6

Peta Map 3D kondisi terang



GAMBAR 7

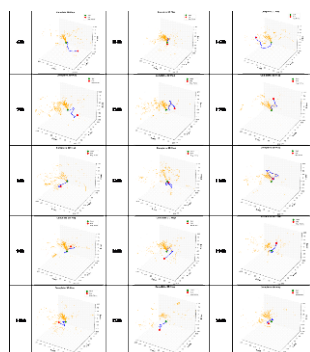
Grafik pengujian ORB Feature Detection kondisi terang

Hasil pengujian parameter features ORB dalam kondisi redup menunjukkan pola yang konsisten dengan hasil kondisi cahaya terang, sebagaimana yang terlihat pada tabel 1 dan gambar 6. Kondisi pencahayaan redup memengaruhi kestabilan deteksi keypoint, dimana pada kondisi redup memiliki nilai efisiensi yang lebih tinggi dibandingkan dengan kondisi terang. Pada pengujian dengan nFeatures = 2000, efisiensi deteksi bertahan pada 97,52%. Dalam kondisi

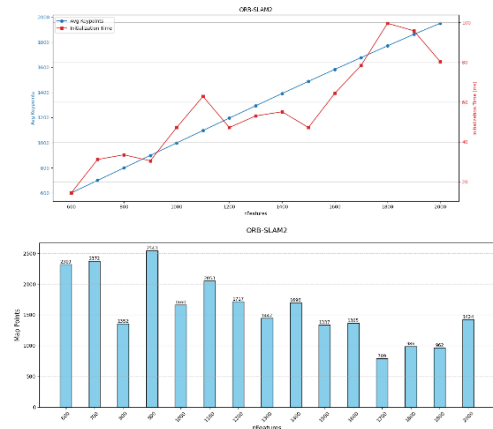
pencahayaan redup dengan $nFeatures = 600$, waktu inisialisasi mencapai nilai tercepat (14,4 ms) dibandingkan seluruh percobaan dengan beberapa kondisi pencahayaan. Berdasarkan gambar 4.13, merupakan distribusi *map points* dalam kondisi redup, dengan nilai tertinggi *map points* dicapai pada $nFeatures = 900$ dengan 2543 *points* dan nilai terendah *map points* terjadi pada $nFeatures = 1700$ dengan 789 *points*. Pada kondisi pencahayaan redup, rentang $nFeatures$ 600 – 900 menjadi titik optimal algoritma ORB-SLAM2, dimana pada rentang ini mampu mempertahankan jumlah *map point* yang maksimal dengan waktu inisialisasi yang stabil berada < 40 ms. Pola penurunan jumlah *map point* bertahap pada rentang $nFeatures$ 1000 – 1600, meskipun tetap mempertahankan nilai *map points* yang relatif tinggi, sementara pada rentang 1700 – 2000 mengalami penurunan jumlah *map points* yang cukup drastis dengan jumlah *map point* konsisten dibawah 1000. Peningkatan performa *map points* dalam kondisi pencahayaan yang redup dijelaskan oleh mekanisme adaptasi ORB-SLAM2 yang lebih selektif dalam memilih fitur berkualitas tinggi. Kondisi pencahayaan yang menantang memaksa sistem untuk melakukan *filtering* lebih ketat terhadap *keypoint* yang terdeteksi. Hal ini meningkatkan sensitivitas terhadap parameter $nFeatures$, dimana nilai yang terlalu tinggi justru dapat menurunkan performa sistem, akibat memproses terlalu banyak fitur yang mengandung *noise*.

TABEL 2
Tabel Feature ORB Detection kondisi redup

$nFeatures$	Avg Keypoints	Map Points	Initialization Time
600	599,797	2319	14,4 ms
700	699,603	2372	31,24 ms
800	799,346	1352	33,56 ms
900	898,783	2543	30,54 ms
1000	998,205	1660	47,25 ms
1100	1097,133	2053	62,88 ms
1200	1196,118	1717	47,25 ms
1300	1293,991	1447	53,1 ms
1400	1391,779	1696	55,17 ms
1500	1488,235	1337	47,26 ms
1600	1583,656	1365	64,41 ms
1700	1677,885	789	78,51 ms
1800	1770,857	986	99,59 ms
1900	1862,907	962	95,81 ms
2000	1950,589	1424	80,34 ms



GAMBAR 8
Peta Map 3D kondisi redup



GAMBAR 9
Grafik Pengujian ORB Feature Detection kondisi redup

Hasil pengujian parameter *features* ORB dalam kondisi redup menunjukkan pola yang konsisten dengan hasil kondisi cahaya terang, sebagaimana yang terlihat pada tabel 7 dan gambar 8 dan 9. Kondisi pencahayaan redup memengaruhi kestabilan deteksi *keypoint*, dimana pada kondisi redup memiliki nilai efisiensi yang lebih tinggi dibandingkan dengan kondisi terang. Pada pengujian dengan $nFeatures = 2000$, efisiensi deteksi bertahan pada 97,52%. Dalam kondisi pencahayaan redup dengan $nFeatures = 600$, waktu inisialisasi mencapai nilai tercepat (14,4 ms) dibandingkan seluruh percobaan dengan beberapa kondisi pencahayaan. Berdasarkan gambar 4.13, merupakan distribusi *map points* dalam kondisi redup, dengan nilai tertinggi *map points* dicapai pada $nFeatures = 900$ dengan 2543 *points* dan nilai terendah *map points* terjadi pada $nFeatures = 1700$ dengan 789 *points*. Pada kondisi pencahayaan redup, rentang $nFeatures$ 600 – 900 menjadi titik optimal algoritma ORB-SLAM2, dimana pada rentang ini mampu mempertahankan jumlah *map point* yang maksimal dengan waktu inisialisasi yang stabil berada < 40 ms. Pola penurunan jumlah *map point* bertahap pada rentang $nFeatures$ 1000 – 1600, meskipun tetap mempertahankan nilai *map points* yang relatif tinggi, sementara pada rentang 1700 – 2000 mengalami penurunan jumlah *map points* yang cukup drastis dengan jumlah *map point* konsisten dibawah 1000. Peningkatan performa *map points* dalam kondisi pencahayaan yang redup dijelaskan oleh mekanisme adaptasi ORB-SLAM2 yang lebih selektif dalam memilih fitur berkualitas tinggi. Kondisi pencahayaan yang menantang memaksa sistem untuk melakukan *filtering* lebih ketat terhadap *keypoint* yang terdeteksi. Hal ini meningkatkan sensitivitas terhadap parameter $nFeatures$, dimana nilai yang terlalu tinggi justru dapat menurunkan performa sistem, akibat memproses terlalu banyak fitur yang mengandung *noise*.

5. Kesimpulan

Hasil penelitian menunjukkan bahwa algoritma ORB-SLAM2 mampu untuk membentuk lingkungan secara real – time dengan tingkat

akurasi dan efisiensi yang tinggi. Pada kondisi terang, performa optimal tercapai pada nFeatures 700-800 dengan map points tertinggi sebanyak 1734 dan waktu inisialisasi 50 ms. Sedangkan pada kondisi redup, performa terbaik tercapai pada nFeatures 900 dengan 2543 map points dan efisiensi deteksi mencapai 99,89 %.

Daftar Pustaka

- [1] M. Aria, “Algoritma Perencanaan Jalur Kendaraan Otonom di Lingkungan Perkotaan dari Sudut Pandang Filosofi Kuhn dan Filosofi Popper,” *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 7, no. 2, 2020, doi: 10.34010/telekontran.v7i2.2627.
- [2] A. A. Fikri and L. Anifah, “Mapping And Localization System Pada Mobile Robot Menggunakan Metode SLAM Berbasis LiDAR,” *Journal of Information Engineering and Educational Technology*, vol. 5, no. 1, 2021, doi: 10.26740/jieet.v5n1.p27-33.
- [3] S. Kusmiawati, E. Setiawan, and E. R. Widasari, “Simulasi Algoritme Hector SLAM untuk Pemetaan 2D pada Quadcopter berbasis ROS,” 2022.
- [4] A. Rahman, “Penerapan SLAM Gmapping dengan Robot Operating System Menggunakan Laser Scanner pada Turtlebot,” *Jurnal Rekayasa ElektriKa*, vol. 16, no. 2, 2020, doi: 10.17529/jre.v16i2.16491.
- [5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Transactions on Robotics*, vol. 31, no. 5, 2015, doi: 10.1109/TRO.2015.2463671.
- [6] L. Li, L. Schulze, and K. Kalavadia, “Promising SLAM Methods for Automated Guided Vehicles and Autonomous Mobile Robots,” in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 2867– 2874. doi: 10.1016/j.procs.2024.02.103.
- [7] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, “Computer vision algorithms and hardware implementations: A survey,” *Integration*, vol. 69. Elsevier B.V., pp. 309–320, Nov. 01, 2019. doi: 10.1016/j.vlsi.2019.07.005.
- [8] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, “A Comprehensive Survey of Visual SLAM Algorithms,” *Robotics*, vol. 11, no. 1. MDPI, Feb. 01, 2022. doi: 10.3390/robotics11010024.
- [9] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual SLAM algorithms: A survey from 2010 to 2016,” *IPSN Transactions on Computer Vision and Applications*, vol. 9. Springer, 2017. doi: 10.1186/s41074-017-0027-2.