Metode Deteksi Titik Pendaratan Yang Aman Bagi Drone Dengan Pendekatan Point Cloud Perception

1st Agung Firmansyah Djalil
Jurusan Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom Surabaya, Indonesia
agungdjalil@student.telkomuniversity.ac.id

2nd Dr. Helmy Widyantara
Jurusan Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom Surabaya, Indonesia
helmywidyantara@telkomuniversity.ac.id

3rd Mohamad Yani
Jurusan Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom Surabaya, Indonesia
mohamadyani@telkomuniversity.ac.id

***Abstrak* — Penggunaan Drone terus meningkat dalam berbagai aplikasi seperti pencarian dan penyelamatan, pengiriman, pemantauan wilayah, serta misi otonom lainnya. Seiring dengan meningkatnya jumlah drone yang beroperasi diruang udara, kemampuan untukmelakukan pendaratan darurat secara aman menjadi aspek yang sangat penting, terutama ketika wahana mengalami kegagalan dan tidak dapat Kembali ke titik awal misi.oleh karena itu, penelitian Ketika wahana mengalami kegagalan dan tidak dapat Kembali ke titikawal misi. Oleh karena itu, penelitian ini mengusulkan metode deteksi titik pendaratan aman berbasis perpsesi point cloud untuk patform drone. Sistem memanfaatkan data point cloud mentah yang diperoleh dari kamera depth dan diproyeksikan ke sistem koordinat drone. Selanjutnya, dilakukan proses downsampling dan segmentasi menggunakan algoritma RANSAC untuk memisahkan bidang datar sebagai kandidat permukaan pendaratan dari objek lain di lingkungan. Titk- titik pada bidang hasil segmentasi dipetakan ke dalam grid dua dimensi, kemudian dianalisis berdasarkan kepadatan titik dan kemiringan permukaan untuk memastikan area yang data dan cukup luas. Evaluasi areapendaratan dilakukan menggunakan dua parameter utama, yaitu kepadatan titik pada setiap sel grid dan jaraknya terhadap objek penghalang di sekitarnya. Area dengan kepadatan tinggi dan jarak terjauh dari hambatan dipilih sebagai lokasi pendaratanpaling aman. Hasil pengujian menunjukkan bahwa metode yang diusulkan mampu mendeteksi titik pendaratan aman secarakonsisten pada berbagai kondisi lingkungan. Selain itu, penerapan downsampling secara signifikan mengurangi waktu komputasi, sehingga metode ini berpotensi diterapkan padasistem drone otonom secara real-time.**

Kata kunci— RANSAC Plane Segmentation, Unmanned Aerial Vehicle, Grid-Base Density, Obstacle Avoidance, Safe Landing Zoen

I. PENDAHULUAN

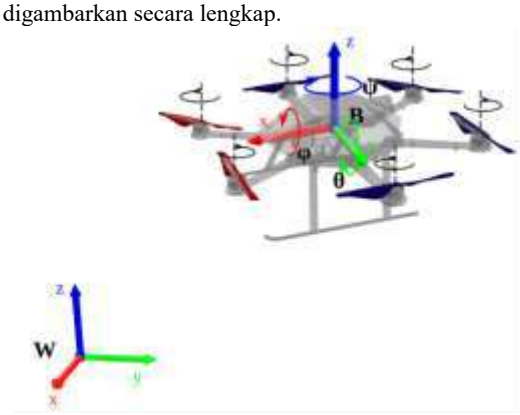
Unmanned Aerial Vehicle (UAV) atau pesawat tanpa awak merupakan wahana terbang yang dikendalikan secara jarak jauh maupun otonom berbasis komputer. Dengan struktur sederhana, bobot ringan, dan fleksibilitas operasional tinggi, UAV atau drone banyak digunakan karena efisiensi biaya dan kemudahan pengoperasiannya. Perkembangan teknologi UAV mendorong pemanfaatannya di berbagai bidang, seperti pengintaian dan pencarian korban bencana, pertanian presisi melalui pemantauan kondisi lahan, serta pengiriman pasokan medis ke daerah terpencil. Selain itu, UAV juga berperan strategis dalam aplikasi militer untuk pengawasan dan pengumpulan data intelijen. Seiring meningkatnya penggunaan drone, aspek keselamatan operasional, khususnya pada fase pendaratan, menjadi perhatian utama. Meskipun navigasi berbasis GPS umum digunakan, pendekatan ini memiliki keterbatasan akurasi dan rentan terhadap gangguan sinyal, terutama di lingkungan kompleks seperti kawasan perkotaan. Hambatan tak terduga seperti pepohonan, kabel listrik, dan bangunan dapat meningkatkan risiko kecelakaan saat pendaratan. Oleh karena itu, penelitian ini mengembangkan sistem pendeteksian zona pendaratan aman berbasis persepsi point cloud untuk drone tipe Vertical Take-Off and Landing (VTOL). Sistem memanfaatkan data point cloud tiga dimensi dari kamera stereo untuk mengekstraksi bidang datar sebagai kandidat area pendaratan melalui tahapan pemrosesan seperti *voxel grid filtering*, pemilihan *region of interest*, dan segmentasi bidang menggunakan algoritma RANSAC. Pendekatan ini diharapkan mampu meningkatkan keamanan dan keandalan pendaratan otonom pada lingkungan yang tidak terstruktur dan kondisi darurat.

II. KAJIAN TEORI

A. Orientasi Rotasi Dan Translasi Pada Drone

Orientasi dan rotas pada drone merupakan aspek fundamental dalam sistem navigasi, kendali dan persepsi lingkungan. Roll, pitch dan yaw berfungsi untuk menentukan arah hadap drone mengatur pergerakanyake segala arah, dan memastikan kestabilannya selama penerbangan. Tanpa informasi orientasi yang akurat, drone tidak mampu mempertahankan posisi, merespons gangguan seperti hembusan angin, maupun melakukan manuver secara presisi. Orientasi drone dalam ruang tiga dimensi dijelaskan melalui perubahan rotasi terhadap tiga sumbu utama, yaitu sumbu X, Y, dan Z. Ketiga rotasi ini dikenal sebagai *Euler Angles*, yang terdiri dari *roll, pitch,* dan *yaw,* dan digunakan secara luas untuk merepresentasikan sikap serta arah orientasi drone terhadap lingkungan sekitar. *Euler Angles* sendiri merupakan metode untuk menggambarkan rotasi suatu benda dalam suatu kerangka koordinat melalui tiga rotasi berurutan pada masing-masing sumbu utama. Rotasi tersebut biasa dinotasikan sebagai φ (phi) untuk rotasi terhadap sumbu X, $\emptyset$ $(theta)$ untuk rotasi terhadap sumbu Y, dan Ψ (psi) untukrotasi terhadap sumbu Z. Pada gambar 2.1 kombinasi

ketiga sudut ini memungkinkan setiap orientasi dalam ruang 3D dapat digambarkan secara lengkap.



GAMBAR 2. 1
(FRAME DRONE)

Untuk dapat merepresentasikan perubahan orientasi tersebut secara komputasional, diperlukan pendekatan matematis yangmampu menggambarkan bagaimana rotasi terjadi pada setiap sumbu. *Euler Angles* tidak hanya digunakn sebagai konsep deskriptif mengeani arah putar drone, tetapi juga dituangkan dalam bentuk matriks rotasi sebagai berikut:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_{\psi} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.\ 1)$$

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{\theta} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (2.\ 2)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = R_{\phi} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (2.\ 3)$$

Selain digunakan untuk menggambarkan orientasi, formulasi matematis tersebut juga menjadi dasar dalam menghitung besaran gerak pada drone, khususnya *linear velocity* dan *angular velocity*. Kedua besaran inipenting karena drone tidak hanya mengalami perubahan posisi, tetapi juga perubahan orientasi selama terbang. *Linear velocity* menggambarkan seberapa cepat drone berpindah posisi dalam ruang 3D. Sedangkan angular velocity menyatakan seberapa cepat drone berputar terhadap sumbu roll, pitch dan yaw. Perhitungan kedua besaran ini dilakukandengan melihat perubahan posisi maupun sudut orientasi terhadap waktu. Dengan mengetahui kecepatan linier serta kecepatansudut secara akurat, sistem kontrol drone dapat menyesuaikan respon motor untuk menjaga kestabilan, melakukan navigasi, serta mengatur manuver secara presisi. Secara matematis, linear velocity dihitung berdasarkan perubahan posisi drone dalam selang waktu tertentu, jika posisi drone pada waktu t_1 adalah (x_1, y_1, z_1) dan pada waktu t_2 berubah menjadi (x_2, y_2, z_2), maka kecepatan linier dapat dinyatakan sebagai:

$$\vec{v} = \frac{\begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}}{t_2 - t_1} \quad (2.\ 4)$$

Persamaan ini menunjukkan bahwa linear velocity merupakan turunan pertama dari posisi terhadap waktu, sehingga sangat terkait dengan pergerakan translasi drone.

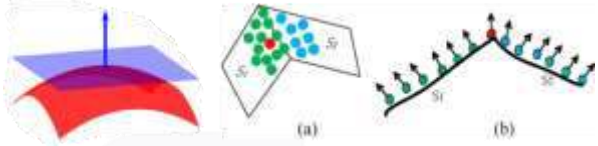
Sedangkan *angular velocity* diperoleh dari perubahan orientasi drone, yang direpresentasikan melalui perubaahn sudut *Euler* terhadap waktu. Jika sudut roll, pitch, dan yaw masing-masing berubah dari $\emptyset_1, \theta_1, \psi_1$ menjadi $\emptyset_2, \theta_2, \psi_2$ maka kecepatan sudut dapat dihitung menggunakan:

$$\vec{\omega} = \frac{\begin{bmatrix} \emptyset_2 - \emptyset_1 \\ \theta_2 - \theta_1 \\ \psi_2 - \psi_1 \end{bmatrix}}{t_2 - t_1} \quad (2.\ 5)$$

Besaran ini biasanya diukur langsung oleh gyroscpe pada IMU dalm satuan radian per detik (rad/s). *Angular velocity* sangat penting karena memberikan informasi kepada flight controller mengenai seberapa cepat drone mengalami rotasi, sehingga sistem dapat melakukan koreksi terhadap perubahan orientasi yang tidak diinginkan. Dengan demikian, perhitungan *linear* dan *angular velocity* yang memanfaatkan model matematika dari orientasi dan posisi drone memungkinkan sistem navigasi dan kontrol drone bekerja secara stabil dan responsif terhadap perubahan lingkungan.

B. Normal Vector

Normal vector adalah sebuah vektor yang tegak lurus terhadap suatu permukaan pada titik tertentu seperti pada gambar 2.7. Dalam konteks representasi 3D seperti point cloud, normal vector digunakan untuk menggambarkan arah orientasi lokal suatu permukaan. Setiap titik pada point cloud dapat dianalisis lingkungan sekitarnya untuk menentukan arah permukaan tempat titik tersebut berada, sehingga menghasilkan sebuah vektor normal n = (n_x, n_y, n_z) yang menunjukkan orientasi permukaan tersebut. Vektor normal ini sangat penting dalam berbagai proses analisis geometri 3D, seperti segmentasi, rekonstruksi, perhitungan kemiringan, dan deteksi bidang datar.



GAMBAR 2. 2
(NORMAL VEKTOR)

Gambar 2.2 bagian (a) memperlihatkan contoh point cloud yang berada pada dua permukaan berbeda, yaitu permukaan S_1dan S_2, di mana titik-titik berwarna hijau berada pada permukaan S_1, titik-titik berwarna biru berada pada permukaan S_2, dan titik berwarna merah terletak pada batas atau pertemuan kedua permukaan tersebut. Pada kondisi ini, penentuan *normal vector* menjadi lebih menantang karena tetangga terdekat titik merah berasal dari dua bidang dengan orientasi yang berbeda, sehingga kualitas normal yang dihasilkan sangat bergantung pada distribusi titik di sekitarnya. Jika titik-titik tetangga berada pada satu permukaan yang datar dan konsisten, normal vector yang diperoleh akan stabil, sedangkan pada area tepi, sudut, atau pertemuan dua bidang, normal yang dihitung cenderung ambigu dan kurang andal. Hal ini diperjelas pada Gambar 2.2 bagian (b), yang menunjukkan orientasi normal vector pada masing-masing permukaan, di mana titik-titik pada permukaan S_1dan S_2memiliki arah normal yang relatif seragam sesuai kemiringan bidangnya, sementara normal pada titik merah menyimpang akibat pengaruh campuran tetangga dari kedua permukaan. Fenomena ini menunjukkan bahwa normal vector tidak hanya merepresentasikan orientasi permukaan, tetapi juga dapat dimanfaatkan untuk mengidentifikasi struktur geometris seperti tepi, sudut, atau batas antar bidang yang umum dijumpai pada objek dan lingkungan yang kompleks.Untuk bisa mendapatkan normal vektor pada suatu permukaan, diperlukan informasi geometri dari titik-titik yang membentuk permukaan tersebut. Secara paling dasar, normal vektor dapat ditentukan menggunakan tiga titik yang tidak segaris dan berada pada bidang yang sama. Misalkan diberikan tiga titik $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, dan $P_3(x_3, y_3, z_3)$.

Dari ketiga titik tersebut dibentuk dua vektor yang terletak pada bidang, yaitu:

$$\vec{v}_1 = P_2 - P_1 = (x_2 - x_1,\ y_2 - y_1,\ z_2 - z_1) \quad (2.\ 6)$$

$$\vec{v}_2 = P_3 - P_1 = (x_3 - x_1,\ y_3 - y_1,\ z_3 - z_1) \qquad (2.\ 7)$$

Kedua vektor ini merepresentasikan dua arah berbeda pada permukaan bidang. Selanjutnya, vektor normal diperoleh dengan menghitung perkalian silang (cross product) antara kedua vektor tersebut, yang menghasilkan vektor yang tegak lurus terhadap bidang:

$$\vec{n} = (v_{1}v_{2z} - v_{1z}v_{2y},\ v_{1z}v_{2x} - v_{1x}v_{2z},\ v_{1x}v_{2y} - v_{1y}v_{2x}) \qquad (2.\ 8)$$

Vektor $\vec{n}$yang diperoleh dari proses ini memiliki arah yang tegak lurus terhadap bidang, namun panjang (magnitudo) vektor tersebut masih bergantung pada skala vektor pembentuknya. Oleh karena itu, diperlukan proses normalisasi agar vektor normal hanya merepresentasikan arah orientasi permukaan.
Normalisasi dilakukan dengan membagi vektor normal dengan panjangnya, yang dihitung menggunakan norma Euclidean:

$$||\vec{n}|| = \sqrt{n_x^2 + n_y^2 + n_z^2} \qquad (2.\ 9)$$

Sehingga vektor normal ter-normalisasi dapat dinyatakan sebagai:

$$\vec{n}_{norm} \frac{\vec{n}}{||\vec{n}||} \qquad (2.\ 10)$$

Vektor normal hasil normalisasi ini memiliki panjang satu satuan dan digunakan sebagai representasi orientasi lokal permukaan. Dengan menggunakan vektor normal ter-normalisasi, perbandingan orientasi antar permukaan dapat dilakukan secara konsisten, serta memungkinkan pendeteksian perubahan orientasi yang signifikan pada batas antar bidang.

III. METODE

A. Perangkat Lunak

Alur kerja lengkap dari sistem deteksi zona pendaratan pada drone dimulai dari proses akuisisi data sensor hingga pemilihan titikyang mencakup proses estimasi keadaan (State Estimation), serta pipeline Landing Spot Detection yang bertugas menganalisis point cloud untuk menemukan lokasi pendaratan yang memenuhi kriteria keselamatan. Kedua bagian ini bekerja secara pararel dan saling berkaitan untuk memastikan bahwa data lingkungan yang diterima drone dapat ditafsirkan secara konsisten dan digunakn sebagai dasar pengambilan keputusan.

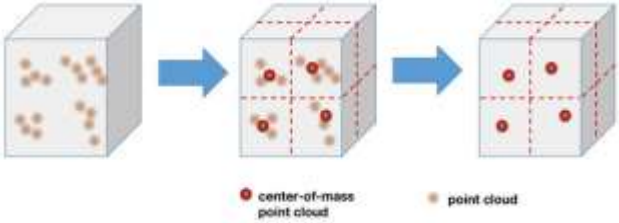


GAMBAR 3. 1
(ALUR PENGOLAHAN POINT CLOUD)

Pada sisi kiri diagram Gambar 3.1, blok Data Fusion menunjukkan bahwa sistem memanfaatkan beberapa sensor onboard pada Pixhawk, yaitu GPS, IMU, LiDAR, dan barometer. Sensor-sensor ini berfungsi sebagai sumber utama untuk memperkirakan keadaan UAV secara real-time. GPS menyediakan informasi posisi global, sementara IMU memberikan data inersia berupa percepatan linier dan kecepatan sudut. LiDAR digunakan untuk memperoleh informasi jarak dan struktur lingkungan di sekitar drone, sedangkan barometer berperan dalam menghitung ketinggian relatif berdasarkan perubahan tekanan udara. Seluruh data sensor tersebut digabungkan menggunakan algoritma *Extended Kalman Filter* (EKF) untuk menghasilkan estimasi posisi dan orientasi UAV yang stabil dan akurat. Estimasi state ini digunakan sebagai acuan dalam transformasi point cloud ke sistem koordinat yang konsisten. Pada bagian Landing Spot Detection, proses diawali dengan akuisisi point cloud dari kamera stereo atau sensor kedalaman. Point cloud mentah yang berada pada frame kamera kemudian ditransformasikan ke frame drone atau dunia melalui tahap Frame Transformation dengan memanfaatkan informasi pose UAV. Selanjutnya dilakukan Downsampling menggunakan metode *voxel grid* untuk mengurangi jumlah titik sehingga komputasi menjadi lebih efisien. Tahap Plane Detection dilakukan dengan algoritma RANSAC untuk mengekstraksi permukaan datar yang berpotensi menjadi area pendaratan. Permukaan ini kemudian dievaluasi pada tahap Landing Zone Evaluation berdasarkan kemiringan, kerataan, keberadaan rintangan, dan luas area. Hasil evaluasi membentuk *safety affordance map*, yang selanjutnya digunakan pada tahap Safety Point Selection untuk menentukan titik pendaratan paling aman dan optimal.

B. Downsampling Point Cloud

Downsampling point cloud merupakan tahap penting dalam proses pemrosesan data 3D yang bertujuan untuk mengurangi jumlah titik pada point cloud tanpa menghilangkan informasi geometris yang signifikan. Kamera stereo atau sensor depth biasanya menghasilkan point cloud dengan kepadatan titik yang sangat tinggi, sehingga jika diproses secara langsung dapat menyebabkan beban komputasi yang besar dan memperlambat kinerja sistem, khususnya pada Aplikasi yang membutuhkan pengolahan data secara real-time seperti pendeteksian zona pendaratan pada drone. Oleh karena itu, downsampling dilakukan untuk menyederhanakan representasi permukaan dengan tetap mempertahankan struktur dan bentuk utamanya.



GAMBAR 3. 2
(VOXEL GRID DOWNSAMPLING)

Metode downsampling yang umum digunakan adalah voxel grid filtering seperti pada gambar 3.2, dimana ruang 3D dibagi menjadi kotak-kotak kecil berukuran tertentu yang disebut voxel. Setiap voxel memiliki ukuran tetap, misalnya

$$v_x \times v_y \times v_z \qquad (3.\ 1)$$

Dimana v_x = ukuran pada sumbu X atau panjang voxel, v_y ukuran pada sumbu Y atau lebar voxel dan v_z ukuran pada sumbu Z atau tinggi voxel, seluruh titik yang berada di dalam voxel yang sama dikelompokkan bersama. Proses downsampling dilakukan dengan menghitung indeks voxel untuk setiap titik melalui operasi pembulatan kebawah seperti berikut

$$i = \frac{x}{v_x}, j = \frac{y}{v_y}, k = \frac{z}{v_z} \qquad (3.\ 2)$$

Dan setiap voxel menampung sekumpulan titik $v_{ijk} = \{p_1, p_2, \dots, p_n\}$. Dari setiap voxel hanya dipertahankan satu titik representatif, berupa centroid dari titik-titik yang berada di dalam voxel tersebut

$$c_x = \frac{1}{n}\sum_{t=1}^{n} x_t \qquad c_y = \frac{1}{n}\sum_{t=1}^{n} y_t \qquad c_z = \frac{1}{n}\sum_{t=1}^{n} z_t \qquad (3.\ 3)$$

Sehingga dihasilkan titik downsampling untuk voxel tersebut adalah $c = (c_x, c_y, c_z)$. Pendekatan ini menghasilkan point cloud yang lebih ringan, terstruktur dan seragam sehingga memudahkan tahap pemrosesan berikutnya seperti segmentasi bidang datar dan evaluasi

zona pendaratan. Selain mengurangi beban komputasi, downsampling juga membantu mengurangi noise pada data dan meningkatkan stabilitas algoritma dalam mendeteksi permukaan.

C. Deteksi Bidang Datar

Deteksi bidang datar merupakan komponen inti dalam proses analisis geometris point cloud, khususnya setelah kandidat region berhasil dididentifikasi melalui segmentasi spasial. Tahapan ini bertujuan untuk mengekstraksi struktur bidang datar dari kumpulan titik tiga dimensi sehingga sistem dapat menentukan apakah suatu area memenuhi syarat sebagai permukaan pendaratan yang layak bagi drone. Keberadaaan bidang datar sangat krusial dalam konteks autonomous landing, mengingat stabilitas dan kemanan pendaratan sangat bergantung pada kualitas permukaan yang dievaluasi.

Secara matematis, sebuah bidang tiga dimensi dapat direpresentasikan melalui persamaan linear:

$$ax + by + cz + d = 0 \qquad (3.4)$$

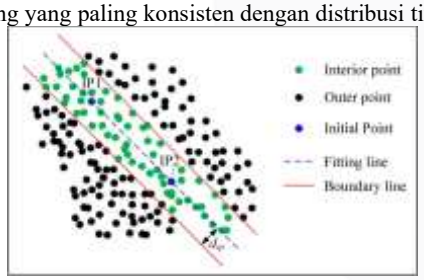
Dengan (a,b,c) sebagai komponen vektor normal n yang didapatkan dari persamaan 2.10 mendefinisikan orientasi bidang terhadap sistem koordinat global:

$$n = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad (3.5)$$

Sementara itu, $d$merupakan konstanta yang menentukan posisi bidang terhadap titik asal $(0, 0, 0)$. Konstanta ini berfungsi untuk menggeser bidang mendekati atau menjauhi titik asal tanpa memengaruhi orientasi bidang. Nilai $d$diperoleh dengan mensubstitusikan salah satu titik (x_0, y_0, z_0)yang terletak pada bidang ke dalam persamaan bidang, sehingga dirumuskan sebagai:

$$d = -(ax + by + cz) \qquad (3.6)$$

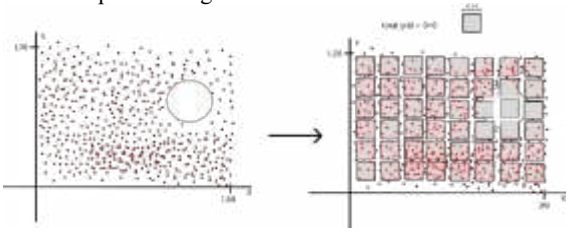
Estimasi parameter bidang tersebut membutuhkan proses yang robust, sehingga diperlukan mekanisme yang mampu memisahkan permukaan sebenarnya dari noise dan outlier. Prinsip kerja RANSAC digambarkan melalui alur pada gambar 3.3, di mana algoritma melakukan *iterative model sampling* untuk menemukan model bidang yang paling konsisten dengan distribusi titik.



GAMBAR 3.3
(KONSEP RANSAC)

D. Deteksi dan Evaluasi Zona Pendaratan

Setelah bidang datar terdeteksi, sistem kemudian memastikan apakah area tersebut benar-benar aman untuk pendaratan. Seluruh titik diselaraskan ke kerangka acuan bidang dan memproyeksikannya ke sebuah bidang dua dimensi untuk membentuk representasi grid.



GAMBAR 3.4
(PEMBAGIAN TITIK DENGAN GRID)

Gambar 3.4 menunjukkan bagaimana pembentukan grid, seluruh titik point cloud diproyeksikan terlebih dahulu ke bidang dua dimensi sehingga meimiliki koordinat (u_i, v_i) dalam satu kerangka acuan yang seragam. Dari kumpulan titik tersebut dihitung batas minimum dan maksimum ruang, yaitu u_{min} dan u_{max} pada sumbu horizontal serta v_{min} dan v_{max} pada sumbu vertikal, yang masing-masing merepresentasikan titik paling kiri, kanan, bawah dan atas dari persebaran data. Ruang berdimensi $(u_{max} - u_{min})$ dan $(v_{max} - v_{min})$ kemudian dibagi menjadi sel-sel grid dengan ukuran tetap Δ, yaitu panjang sisi satu kotak grid dalam satuan meter. Bentuk persamaan yang digunakan:

$$cols = \left\lceil \frac{u_{max} - u_{min}}{\Delta} \right\rceil \qquad (3.7)$$

dan

$$rows = \left\lceil \frac{v_{max} - v_{min}}{\Delta} \right\rceil \qquad (3.8)$$

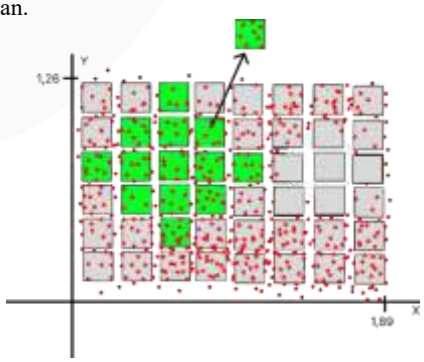
Selanjutnya point cloud dipetakan ke dalam sebuah grid 2D agar dapat dianalisis secara lebih trestruktur. Pada gambar 3.9 bagian kiri, titik-titik merah tersebar bebas dan tidak memiliki struktur, sehingga sulit untuk mengetahui bagian mana dari permukaan yang benar-benar penuh, kosong, atau tidak rata. Untuk itu, bidang dipartisi menjadi kotak-kotak kecil berukukan tetap (grid cell) seperti pada gambar 3.9 bagian kanan. Setiap titik (x_i, y_i) kemudian dimasukkan ke dalam sel grid tertentu berdasarkan koordinatnya. Indeks sel dihitung menggunakan:

$$i_x = \left\lfloor \frac{u_i - u_{min}}{\Delta} \right\rfloor \qquad (3.9)$$

Dan

$$i_y = \left\lfloor \frac{v_i - v_{min}}{\Delta} \right\rfloor \qquad (3.10)$$

Setiap sel pada grid kemudian dianalisis untuk melihat seberapa padat titik yang menempatinya sehingga dapat dikethui bagian mana yang benar-benar membentuk permukan yang utuh. Grid ini selanjutnya diperiksa untuk menutup celah-celah kecil serta mengidentifikasi area yang berpotensi menjadi rintangan, seperti tepi permukaan atau objek yang tidak terdeteksi secara konsisten. Dengan menggunakan pola cakram berukuran tertentu, sistem menilai setiap posisi potensial sebagi pusat zona pendaratan dengan melihat apakah area sekelilingnya cukup penuh, bebas dari rintangan, dan memilki jarak aman terhadap batas permukaan. Dari seluruh kandidat, dipilih pusat yang paling bersih dan paling jauh dari rintangan.



GAMBAR 3.5
(PENCARIAN TITIK PENDARATAN YANG AMAN)

Titik tersebut kemudian divalidasi kembali melalui beberapa frame untuk memastikan ke kestabilannya terhadap gangguan sensor. Jika lokasi tersebut konsisten aman, barulah sistem menetapkannyasebagi titik pendaratan yang sah dan mempublikasikannya sebagai referensi bagi controller drone.

IV. HASIL DAN PEMBAHASAN

A. Lingkungan Pengujian

Environment 1 merupakan area terbuka dengan permukaan tanah berlapis rumput yang relatif datar, namun tetap memiliki variasi elevasi kecil akibat kontur alami dan tekstur vegetasi terlihat pada gambar 4.1. Di dalam area ini terdapat beberapa obstacle berukuran kecil hingga sedang, seperti traffic cone, sebuah kotak hitam, serta semak-semak dan ranting tanaman yang meninggalkan bentuk permukaan tidak rata pada titik-titik tertentu.



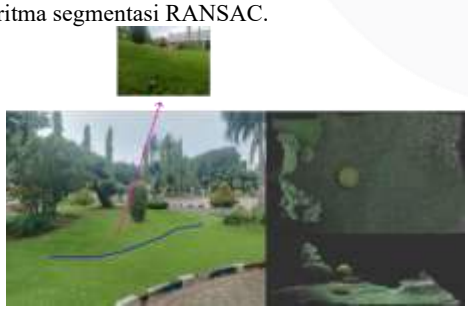
GAMBAR 4. 1
(LINGKUNGAN PENGUJIAN 1)

Lingkungan II pada Gambar 4.2 dikarakterisasikan sebagai taman perkotaan dengan kepadatan objek sedang hingga tinggi dan struktur semi-unstructured. Area ini didominasi vegetasi dengan variasi ketinggian yang signifikan, mulai dari rumput hingga semak dan pohon. Berdasarkan point cloud ZED2i, sebagian besar permukaan memiliki *roughness* alami yang tinggi dan tidak membentuk bidang datar sempurna, serta mengalami tingkat *occlusion* yang cukup besar akibat kepadatan vegetasi. Meskipun terdapat beberapa area rumput yang relatif datar, kedekatannya dengan objek lain dan variasi elevasi di sekitarnya meningkatkan risiko pendaratan. Oleh karena itu, lingkungan ini tergolong menantang bagi pendaratan drone otonom dan memerlukan analisis point cloud yang lebih ketat.



GAMBAR 4. 2
(LINGKUNGAN PENGUJIAN II)

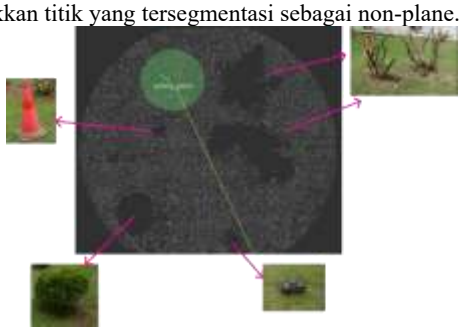
Lingkungan Uji Coba 3 pada gambar 4.3 merepresentasikan skenario medan alami bergelombang yang secara visual didominasi oleh permukaan tanah yang ditutupi oleh rumput atau vegetasi penutup tanah yang rendah. Karakteristik utama Point Cloud ini adalah sifatnya yang non-planar dan bergelombang, dengan variasi ketinggian yang terlihat jelas di seluruh area bidang tanah, yang secara kritis menguji ambang batas toleransi kerataan dari algoritma segmentasi RANSAC.



GAMBAR 4. 3
(LINGKUNGAN PENGUJIAN III)

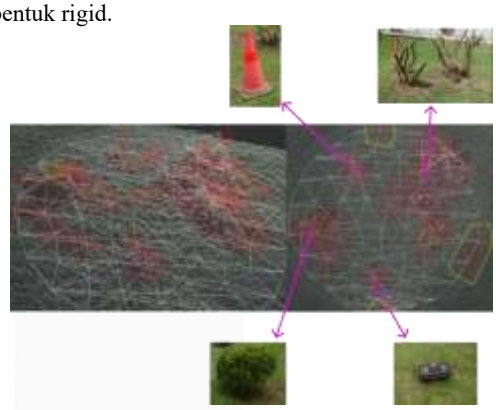
B. Hasil Segmentasi RANSAC dan GNG

Perbandingan dilakukan untuk mengevaluasi kinerja GNG dan RANSAC dari segi akurasi, ketahanan terhadap noise, dan efisiensi komputasi, dengan data yang sama agar hasil objektif. GNG unggul dalam memodelkan struktur data secara adaptif, sedangkan RANSAC efektif mengekstraksi model geometris robust terhadap outlier. Pada Lingkungan 1 (Gambar 4.4), RANSAC berhasil memisahkan permukaan tanah datar dari obstacle, dengan titik putih mewakili point cloud yang dianalisis dan area kosong menunjukkan titik yang tersegmentasi sebagai non-plane.



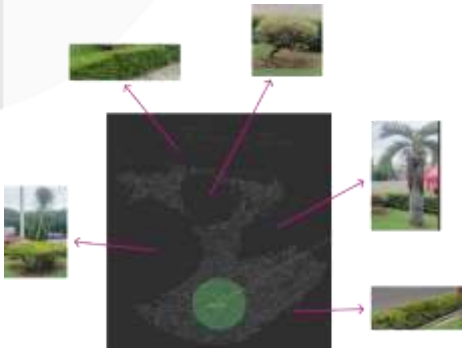
GAMBAR 4. 4
(HASIL SEGMENTASI RANSAC DI LINGKUNGAN 1)

Hasil segmentasi menunjukkan bahwa RANSAC mampu mengidentifikasi objek utama di permukaan rumput. Area kosong memanjang di kiri atas terkait semak gundul, area gelap bulat di kiri bawah adalah semak kecil, dan traffic cone di kiri juga terdeteksi sebagai outlier karena tinggi dan permukaannya tegak. Objek kotak hitam di kanan bawah juga berhasil dikenali sebagai non-plane, menunjukkan RANSAC efektif menangkap objek berukuran rendah dengan bentuk rigid.



GAMBAR 4. 5
(HASIL CLUSTERING GNG PADA LINGKUNGAN 1)

Pada lingkungan pengujian yang sama, GNG mengklasifikasikan objek kompleks seperti vegetasi dan kerucut sebagai non-planar karena adaptif terhadap bentuk, namun permukaan datar kadang salah klasifikasi akibat noise dan variasi titik. Sebaliknya, RANSAC efektif memisahkan permukaan tanah planar dari vegetasi dan rintangan, sehingga lebih handal untuk membedakan ground dan non-ground, sementara GNG lebih sensitif terhadap variasi bentuk dan distribusi titik.



GAMBAR 4. 6
(HASIL SEGMENTASI RANSAC PADA LINGKUNGAN II)

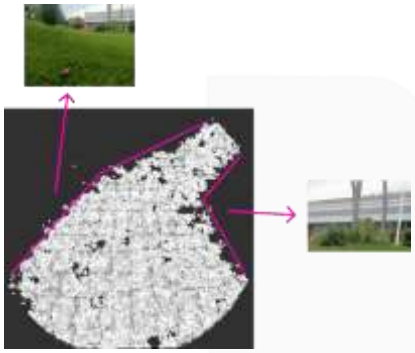
Berbeda dengan RANSAC, GNG pada Gambar 4.18 gagal mengenali bidang datar. Variasi kecil pada permukaan rumput dianggap sebagai fitur penting, sehingga area datar diklasifikasikan sebagai bidang kompleks berwarna merah. GNG berhasil

memetakan pohon dan pagar tanaman sebagai bidang kompleks karena sifatnya adaptif mengikuti kontur objek, namun tidak memisahkan ground dan rintangan, sehingga seluruh area terlihat menyatu dalam satu klasifikasi kompleks.



GAMBAR 4. 7
(HASIL CLUSTERING GNG PADA LINGKUNGAN II)

Hasil segmentasi pada gambar 4.8 menggunakan RANSAC menunjukkan bahwa algoritma ini mampu mendeteksi bidang datar permukaan tanah secara konsisten. Bidang planar yang dihasilkan merepresentasikan permukaan tanah dengan baik meskipun terdapat area lain yang memiliki kemiringan atau gangguan dari objek di sekitarnya. Hal ini menunjukkan kemampuan RANSAC dalam mengekstraksi model geometris sederhana dari data point cloud yang kompleks.



GAMBAR 4. 8
(HASIL SEGMENTASI RANSAC PADA LINGKUNGAN III)

Hasil segmentasi menggunakan Growing Neural Gas (GNG) pada lingkungan yang sama (Gambar 4.8) menunjukkan karakteristik berbeda dibanding RANSAC. GNG mampu mengelompokkan titik dengan kepadatan dan struktur lokal serupa, sehingga objek non-linear seperti semak terdeteksi jelas melalui node dan koneksi yang padat. Namun, GNG gagal mendeteksi bidang datar atau miring secara eksplisit karena permukaan tanah yang homogen tidak membentuk klaster dominan, sehingga node tersebar merata dan bidang tidak tersegmentasi sebagai entitas utuh.



GAMBAR 4. 8
(HASIL CLUSTERING GNG PADA LINGKUNGAN III)

C. Hasil Pengujian Sistem

TABEL 4. 1
(HASIL PENGUJIAN SISTEM)

Lingkungan Pengujian Ke-	Percobaan Ke-	Berhasil Mendarat Dengan Aman	Total Waktu (s)
1	1	Berhasil	57
	2	Berhasil	70
	3	Berhasil	54
	4	Berhasil	59
	5	Berhasil	65
Rata-Rata Waktu			61
2	1	Berhasil	35
	2	Berhasil	26
	3	Berhasil	35
	4	Berhasil	19
	5	Berhasil	27
Rata-Rata Waktu			28,4
3	1	Berhasil	77
	2	Berhasil	68
	3	Berhasil	63
	4	Berhasil	75
	5	Berhasil	60
Rata-Rata Waktu			68,6

Berdasarkan Tabel 4.1, sistem berhasil melakukan pendaratan aman di tiga lingkungan. Lingkungan 1 memiliki waktu rata-rata 61 detik, sedikit di atas threshold karena banyak area datar potensial. Lingkungan 2 tercepat dengan rata-rata 28,4 detik karena area aman terbatas. Lingkungan 3 rata-rata 68,6 detik, melebihi threshold akibat kompleksitas lingkungan dan banyak kandidat area datar, sehingga segmentasi dan validasi lebih lama.

D. Waktu Komputasi RANSAC dan GNG

TABEL 4. 2
(DATA PERBANDINGAN RANSAC DAN GNG DENGAN DOWNSAMPLING)

Lingkungan Pengujian	Algoritma	Min (s)	Max (s)	Average (s)
1	GNG	0.20	1.09	0.62
	RANSAC	0.14	0.22	0.16
2	GNG	0.23	1.04	0.64
	RANSAC	0.24	0.71	0.43
3	GNG	0.20	0.95	0.50
	RANSAC	0.13	0.30	0.20

Berdasarkan Tabel 4.2, RANSAC lebih efisien dan konsisten dibanding GNG pada tiga lingkungan. RANSAC mencatat rata-rata waktu 0,16–0,43 s, lebih rendah dan stabil dibanding GNG yang 0,5–0,64 s, menunjukkan ketahanan RANSAC terhadap variasi kondisi lingkungan.

TABEL 4. 3
(DATA PERBANDINGAN RANSAC DAN GNG TANPA DOWNSAMPLING)

Lingkungan Pengujian	Algoritma	Min (s)	Max (s)	Average (s)
1	GNG	1.05	2.65	1.79
	RANSAC	0.28	0.63	0.36
2	GNG	0.98	2.46	1.73
	RANSAC	0.66	1.11	0.83
3	GNG	0.91	2.44	1.68
	RANSAC	0.42	1.39	0.7

Berdasarkan Tabel 4.3, RANSAC lebih cepat dan konsisten dibanding GNG pada tiga lingkungan, dengan rata-rata waktu 0,36–0,70 s, sedangkan GNG lebih lambat 1,68–1,79 s, menunjukkan efisiensi RANSAC dalam segmentasi permukaan datar di berbagai kondisi.

E. Waktu Komputasi Setiap Proses

TABEL 4. 4
(TABEL WAKTU KOMPUTASI PIPELINE TANPA DOWNSAMPLING)

Lingkungan Pengujian	Algoritma	Min (s)	Max (s)	Average (s)
1	RANSAC	0.32	0.5	0.3
	PencarianTitik Aman	0.06	0.1	0.08
2	RANSAC	0.42	1.06	0.67
	Pencarian Titik Aman	0.04	0.07	0.05
3	RANSAC	0.43	0.92	0.64
	Pencarian Titik Aman	0.03	0.05	0.04

Verifikasi pipeline tanpa downsampling menunjukkan bahwa RANSAC mendominasi beban komputasi dengan rata-rata 0,30–0,67 s tergantung kompleksitas lingkungan, sedangkan pencarian titik aman tetap ringan (0,04–0,08 s). Hasil ini menandakan RANSAC sangat dipengaruhi kondisi lingkungan, sementara pencarian titik aman relatif stabil.

TABEL 4. 5
(TABEL WAKTU KOMPUTASI PIPELINE DENGAN DOWNSAMPLING)

Lingkungan Pengujian	Algoritma	Min (s)	Max (s)	Average (s)
1	RANSAC	0.13	0.36	0.16
	PencarianTitik Aman	0.03	0.06	0.04
	Downsampling	0.01	0.03	0.02
2	RANSAC	0.14	0.68	0.41
	Pencarian Titik Aman	0.02	0.04	0.02
	Downsampling	0.01	0.04	0.02
3	RANSAC	0.15	0.31	0.21
	Pencarian Titik Aman	0.02	0.05	0.03
	Downsampling	0.01	0.04	0.02

Berdasarkan Tabel 4.5, downsampling memiliki waktu komputasi rendah dan stabil (~0,02 s) pada tiga lingkungan, efektif menekan waktu RANSAC tanpa menurunkan kualitas segmentasi. RANSAC tetap andal mendeteksi bidang datar, meski waktu meningkat di lingkungan lebih kompleks, sementara pencarian titik aman relatif ringan (0,02–0,04 s).

V. KESIMPULAN

Secara keseluruhan, sistem yang dikembangkan mampu melakukan pemrosesan point cloud untuk mendeteksi bidang datar dan menentukan titik pendaratan aman secara efektif pada berbagai lingkungan pengujian. Hasil eksperimen menunjukkan bahwa algoritma RANSAC memiliki kinerja yang andal dalam mendeteksi bidang datar, khususnya setelah dipadukan dengan tahapan downsampling yang secara signifikan menurunkan waktu komputasi tanpa mengurangi kualitas hasil segmentasi. Sementara itu, Growing Neural Gas (GNG) mampu merepresentasikan struktur lingkungan yang kompleks seperti vegetasi atau semak, namun kurang optimal dalam membedakan bidang datar dan bidang miring secara eksplisit untuk kebutuhan pendaratan. Integrasi tahapan downsampling, segmentasi bidang, dan pencarian titik aman terbukti meningkatkan efisiensi pipeline secara keseluruhan, sehingga sistem dapat bekerja lebih cepat dan stabil.

REFERENSI

[1] J. Oluoch and M. S. Alam, "A survey of safe landing zone detection techniques for autonomous UAVs," *Expert Systems with Applications,* p. 115091, 2021.

[2] R. Hossain, "A Short Review of the Drone Technology," *International Journal of Mechatronics and Manufacturing Technology,* p. 53–68, 2022.

[3] Y. Cao, M. Li, L. Xu and M. Wei, "Autonomous Landing Spot Detection for Unmanned Aerial Vehicles Based on Monocular Vision," *ISPRS Archives (The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences),* p. 51–58, 2024.

[4] W. Lyu, W. Ke, H. Sheng, X. Ma and H. Zhang, "Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering," *Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering,* p. 3160 (article number), 2024.

[5] G. Loureiro, A. Dias, A. Martins and J. Almeida, "Emergency Landing Spot Detection Algorithm for Unmanned Aerial Vehicles," *Remote Sensing,* p. 1930, 2021.

[6] K. Backman, D. Kulić and H. Chung, "Learning to Assist Drone Landings," *IEEE Robotics and Automation Letters,* 2021.

[7] S. N. M. Isa, S. A. Abdul Shukor, N. A. Rahim, I. Maarof, Z. R. Yahya, A. Zakaria, A. H. Abdullah and R. Wong, "Point Cloud Data Segmentation Using RANSAC and Localization," *IOP Conference Series: Materials Science and Engineering,* p. 012004, 2019.

[8] S. Lee and Y. Kwon, "Safe Landing of Drone Using AI-based Obstacle Avoidance," *International Journal of Mechanical Engineering and Robotics Research,* p. 1495–1501, 2020.

[9] S. N. M. Isa, S. A. Abdul Shukor, N. A. Rahim, I. Maarof, Z. R. Yahya, A. Zakaria, A. H. Abdullah and R. Wong, "Point Cloud Data Segmentation Using RANSAC and Localization," *IOP Conference Series: Materials Science and Engineering,* p. 012004, 2019.

[10] T. Foote, "tf: The Transform Library," *(Conference) Technologies for Practical Robot Applications (TePRA),* 2013.

[11] K. Fang, K. Xu, Z. Wu, T. Huang and Y. Yang, "Three-Dimensional Point Cloud Segmentation Algorithm Based on Depth Camera for Large Size Model Point Cloud Unsupervised Class Segmentation," *Sensors,* p. 112, 2024.

[12] A. A. Saputra, W. H. Chin, Y. Toda, N. Takesue and N. Kubota, "Dynamic Density Topological Structure Generation for Real-Time Ladder Affordance Detection," pp. 1-6, 2018.