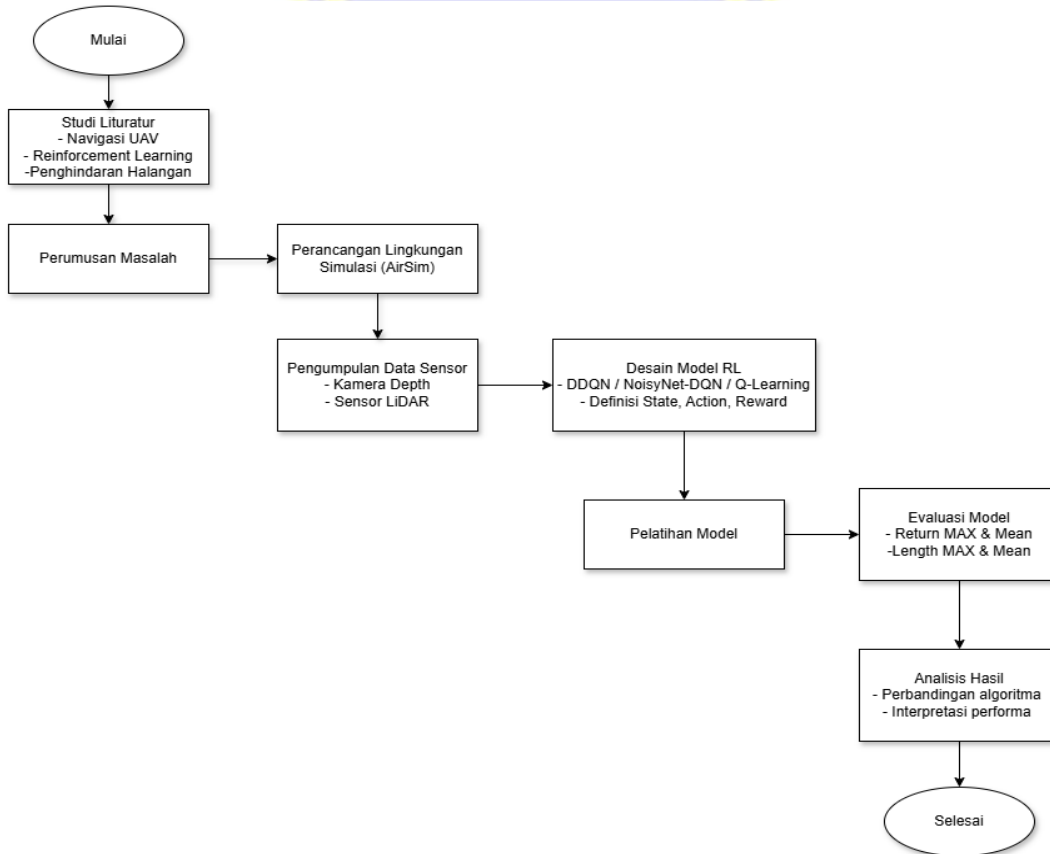


BAB III METODOLOGI PENELITIAN

3.1 Desain penelitian

Pada tahap ini dilakukan perancangan sistem navigasi UAV berbasis Reinforcement Learning yang bertujuan untuk memungkinkan UAV dapat bergerak secara mandiri dan menghindari halangan di lingkungan dinamis. Sistem ini dibangun menggunakan pendekatan Reinforcement Learning, yang memungkinkan agent UAV belajar melalui interaksi langsung dengan lingkungan simulasi.



Gambar 3 Flowchart Metode Penelitian

Secara umum, alur metode penelitian yang diusulkan dapat dilihat pada **Gambar 3**. Diagram tersebut memperlihatkan tahapan utama mulai dari tinjauan pustaka, perancangan lingkungan simulasi, pelatihan model Reinforcement Learning, hingga evaluasi performa UAV dalam penghindaran halangan

Perancangan sistem dimulai dengan mendefinisikan lingkungan simulasi yang merepresentasikan area navigasi UAV. Simulasi ini dibuat menggunakan platform seperti AirSim atau Gazebo, yang menyediakan visualisasi 3D lingkungan lengkap dengan berbagai jenis halangan, seperti bangunan, pohon, atau objek bergerak.

UAV sebagai agent akan mengamati lingkungan melalui data sensor, seperti citra kamera atau peta kedalaman (depth map). Data ini digunakan untuk membentuk representasi keadaan (state), yang mencakup posisi UAV, arah gerak, kecepatan, serta posisi relatif terhadap halangan dan tujuan. Berdasarkan keadaan tersebut, UAV akan menentukan aksi terbaik, seperti bergerak maju, berbelok, atau naik turun, untuk mencapai tujuan sambil menghindari tabrakan.

Sistem ini menggunakan algoritma Reinforcement Learning seperti Double Deep Q-Network (DDQN), NoisyNet-DQN, Q-learning. Jaringan ini dilatih untuk memetakan setiap keadaan ke aksi yang optimal dengan tujuan memaksimalkan reward kumulatif, di mana reward diberikan saat UAV berhasil mendekati tujuan atau menghindari halangan, dan penalti diberikan saat terjadi tabrakan.

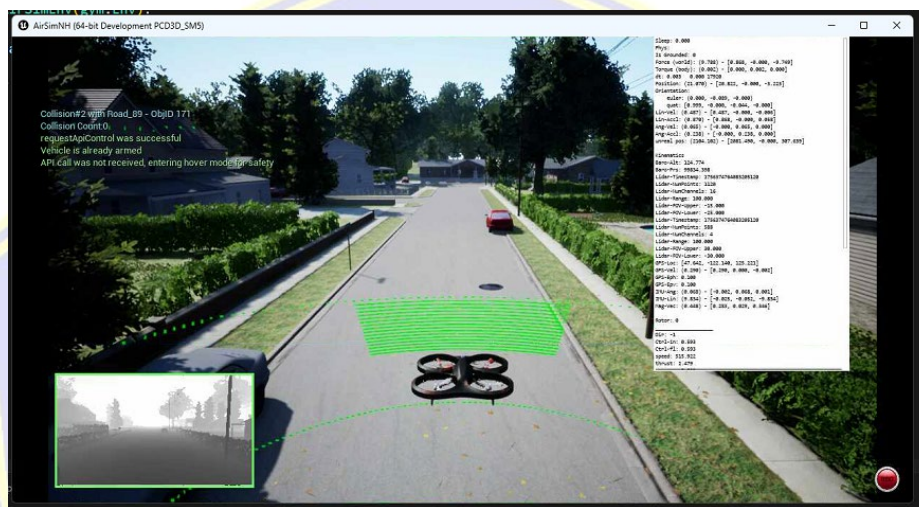
Desain sistem ini diimplementasikan dalam framework Python menggunakan pustaka pembelajaran mesin seperti PyTorch. Komponen utama sistem terdiri dari modul pengolahan data sensor, modul agent RL, serta modul simulasi dan evaluasi. Interaksi antara UAV dan lingkungan terus berlangsung selama proses pelatihan, sehingga UAV dapat mempelajari strategi navigasi yang efisien secara bertahap.

3.2 Lingkungan simulasi

Pada penelitian ini, lingkungan simulasi (Environment) dikembangkan menggunakan platform Airsim yang berbasis Unreal Engine sebagai media pelatihan dan pengujian algoritma Reinforcement Learning. Airsim dipilih karena mampu menyediakan visualisasi tiga dimensi (3D) yang realistis. Alternatif lain seperti Gazebo juga dapat digunakan detail visual dan integrasi sensor yang lebih baik untuk UAV.

Lingkungan simulasi ini terdiri dari 2 komponen utama, yaitu agent (UAV) dan environment (lingkungan virtual).

1. Agent berperan sebagai actor utama yang melakukan navigasi, mendeteksi halangan, dan mengambil Keputusan berdasarkan state yang diterima dari lingkungan. UAV dikontrol melalui script Python menggunakan AirSim API, yang mengatur parameter penerbangan seperti posisi, kecepatan, dan orientasi.
2. Environment mencakup berbagai element visual dan fisik seperti medan perbukitan, pepohonan, bangunan, serta objek dinamis yang berfungsi sebagai halangan. Lingkungan ini memberikan data sensorik berupa gambar RGB, Depth Map, dan sensor lidar yang digunakan sebagai masukan (state) bagi agent.



Gambar 4 Lingkungan simulasi UAV.

Cara kerja sistem pada Airsim digambarkan pada **Gambar 4**, dimana UAV (agent) berinteraksi langsung dengan lingkungan secara berulang melalui siklus berikut :

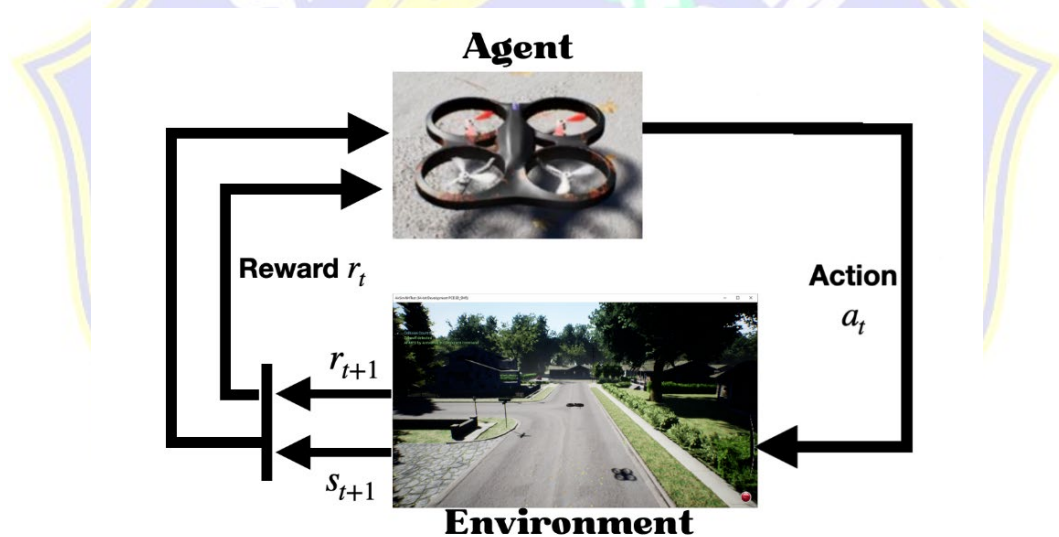
1. UAV melakukan aksi (action) berdasarkan kebijakan (policy) yang dihasilkan oleh model Reinforcement Learning.
2. Lingkungan Airsim kemudian merespons aksi tersebut dan memperbarui kondisi dunia simulasi, termasuk posisi UAV serta jarak terhadap halangan.
3. Sensor virtual (Kamera dan LIDAR) mengirimkan Kembali data state ke agent, yang berisi informasi terbaru dari lingkungan.
4. Berdasarkan hasil observasi tersebut, system memberikan nilai *reward* kepada agent.

5. Proses ini terus berlangsung secara iterative hingga UAV mampu mempelajari pola navigasi yang efisien dan aman.

Dengan pendekatan ini, UAV dapat dilatih secara intensif dalam lingkungan virtual yang aman dan terkendali tanpa risiko kerusakan perangkat keras.

3.3 Model Algoritma Reinforcement Learning

Model algoritma Reinforcement Learning (RL) yang digunakan dalam penelitian ini dirancang untuk memungkinkan Unmanned Aerial Vehicle (UAV) belajar secara mandiri melalui interaksi berulang dengan lingkungan simulasi. Dalam konteks ini, UAV berperan sebagai Agent yang berupaya mencapai tujuan sambil menghindari halangan seperti bangunan, pepohonan, maupun objek dinamis lainnya.



Gambar 5 Model Interaksi Agent dan environment pada Reinforcement Learning.

Pada **Gambar 5** Model Interaksi Agent dan environment pada Reinforcement Learning, proses RL terdiri dari 4 komponen utama yaitu berikut ini:

1. **State** : merepresentasikan kondisi UAV pada waktu tertentu, misalnya posisi, kecepatan, orientasi, dan jarak terhadap halangan.
2. **Action** : Keputusan yang yang diambil UAV, seperti bergerak maju, berbelok, naik, atau turun.
3. **Reward** : Merupakan umpan balik numerik yang menunjukkan seberapa baik aksi tersebut.

4. **Policy** : strategi pengambilan Keputusan yang dipelajari UAV untuk memaksimalkan reward kumulatif selama misi berlangsung.

Penelitian ini mengimplementasikan dan membandingkan 3 algoritma utama dalam kerangka RL, yaitu Double Deep Q-Network (DDQN), NoisyNet-DQN, Q-Learning masing- masing memiliki keunggulan tersendiri dalam konteks navigasi UAV:

1. Double Deep Q-Network (DDQN)

$$Y_t^{\text{DoubleDQN}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \arg \max_a Q(S_{t+1}, a; \theta_t), \theta_t^-) \quad (2)$$

Persamaan ini mendefinisikan nilai target yang digunakan dalam algoritma Double Deep Q-Network (DDQN). Jaringan *online* dengan parameter θ_t digunakan untuk memilih aksi dengan nilai tertinggi pada keadaan berikutnya, sedangkan jaringan target dengan parameter θ_t^- digunakan untuk memperkirakan nilai Q yang sesuai. Pemisahan antara kedua jaringan ini membantu mengurangi *overestimation bias* yang sering muncul pada metode DQN standar, sehingga meningkatkan stabilitas dan akurasi proses pembelajaran (Fortunato et al., 2019).

2. NoisyNet-DQN

$$y \stackrel{\text{def}}{=} (\mu^w + \sigma^w \odot \varepsilon^w)x + \mu^b + \sigma^b \odot \varepsilon^b \quad (3)$$

Persamaan ini menjelaskan mekanisme parameterized noise yang digunakan dalam NoisyNet-DQN untuk proses eksplorasi. Dalam formulasi ini, μ^w and σ^w masing- masing merepresentasikan nilai rata-rata dan simpangan baku dari parameter bobot, sedangkan ε^w merupakan variabel acak yang diambil dari distribusi Gaussian. Operator \odot menunjukkan operasi perkalian elemen demi elemen (element-wise multiplication). Dengan menyuntikkan noise langsung ke dalam parameter jaringan, NoisyNet meningkatkan efisiensi eksplorasi dibanding strategi tradisional seperti $\varepsilon - greedy$ (Fortunato et al., 2019).

3. Q-Learning

$$\theta_{t+1} = \theta_t + \alpha(Y_t^Q - Q(S_t, A_t; \theta_t))\nabla_{\theta}Q(S_t, A_t; \theta_t) \quad (4)$$

Persamaan ini merepresentasikan proses pembaruan parameter jaringan Q (Q-network) dalam pembelajaran penguatan (Reinforcement Learning). Di sini θ_t menyatakan parameter model pada iterasi ke t, dan α Adalah learning rate yang mengatur besarnya angka pembaruan. Selisih antara nilai target Y_t^Q dan nilai prediksi $Q(S_t, A_t; \theta_t)$ berfungsi sebagai sinyal kesalahan, kemudian disebarkan kembali melalui jaringan menggunakan gradien $\nabla_{\theta}Q(S_t, A_t; \theta_t)$ untuk mengoptimalkan parameter (van Hasselt et al., 2015).

Pada penelitian ini dilakukan modifikasi dari repotasi DRL-Pytorch dengan mengganti replay buffer default menjadi Hindsight Experience Reply (HER). HER digunakan karena mampu memperbaiki efisiensi pembelajaran pada lingkungan dengan reward yang jarang (sparse reward), seperti navigasi UAV yang jarang mencapai goal di awal pelatihan.

Tabel 3 Parameter system Navigasi UAV Berbasis Reinforcement Learning.

Kategori	Parameter	Nilai / Deskripsi
Lingkungan Simulasi	Simulator	AirSim (Unreal Engine)
	Mode Simulasi	Outdoor dengan obstacle statis & dinamis
	Jenis Halangan	Pohon, bangunan, objek bergerak
Sensor UAV	Depth Camera (RGB-D)	16-bit depth map
	LiDAR	360° radius
	Jarak Maksimal Deteksi	20–30 meter
	Jarak Aman Minimum	1–2 meter
	Data Sensor	Depth frame, LiDAR polar distance
Representasi State	State Vector	Posisi UAV, kecepatan, jarak terhadap obstacle
	Tipe Action	Diskrit (6 aksi)
Aksi (Action Space)	Daftar Aksi	Maju, belok kiri, belok kanan, naik, turun

Kategori	Parameter	Nilai / Deskripsi
	Frekuensi Update	10–20 Hz
	Tipe RL	Value-based deep RL
Algoritma RL	Metode yang Digunakan	DDQN, NoisyNet-DQN, Q-Learning
	Replay Buffer	Hindsight Experience Replay (HER)
	Ukuran Replay Buffer	50.000 –100.000 transitions
	Batch Size	64
	Optimizer	Adam
	Learning Rate	1e-4 – 5e-4
	Discount Factor (γ)	0.99
	Target Network Update	Setiap 1000 step
	Reward Mendekati Goal	+1 sampai +5 per step
Reward Function	Reward Mencapai Goal	100
	Penalti Tabrakan	-100
	Penalti Terlalu Dekat Obstacle	-10
	Shaping Reward	Distance-based shaping
	Jumlah Episode	10.000–50.000
Training Setup	Max Step per Episode	500
	Device	GPU dan CPU
	Framework	PyTorch (DRL-Pytorch Repo)
	Goal Sampling Strategy	final
Parameter HER	HER Ratio	0.2 – 0.4
	Goal Encoding	State vector / koordinat target UAV

3.4 Rencana Evaluasi

Evaluasi dilakukan untuk mengukur performa sistem navigasi UAV berbasis Reinforcement Learning yang telah dirancang dan dilatih dalam lingkungan simulasi. Tujuan dari evaluasi ini adalah untuk mengukur kemampuan

UAV dalam menghindari halangan, mencapai tujuan secara efisien dalam berbagai kondisi lingkungan simulasi.

Evaluasi kinerja dilakukan berdasarkan lima parameter utama yang dihasilkan dari proses pelatihan dan pengujian, yaitu:

a. Return Max (Reward Maksimum)

Menunjukkan reward tertinggi yang dicapai oleh model selama proses simulasi. Semakin tinggi nilai ini, semakin efektif strategi navigasi yang dipelajari.

b. Return Mean (Reward Rata-rata)

Menggambarkan konsistensi performa UAV selama seluruh episode pelatihan.

c. Length Max (Panjang Episode Maksimum)

Menunjukkan jumlah Langkah terbanyak yang ditempuh UAV dalam 1 episode. Nilai tinggi dapat mengindikasikan kemampuan UAV bertahan lebih lama tanpa tabrakan, namun perlu dievaluasi Bersama efisiensi jalur.

d. Length Mean (Panjang Episode Rata-rata)

Menunjukkan efisiensi rata-rata navigasi UAV dalam menyelesaikan misi.

e. Success Rate(%)

Menunjukkan presentase keberhasilan UAV dalam mencapai target tanpa mengalami tabrakan.

Setiap algoritma diuji dalam beberapa skenario simulasi dengan tingkat kompleksitas berbeda. Hasil dari setiap pengujian dibandingkan untuk menilai:

- a. Efektivitas penghindaran halangan.
- b. Efisiensi jalur berdasarkan Panjang episode dan reward.
- c. Konsistensi performa antar episode.

Analisis dilakukan dengan membandingkan nilai rata-rata dan maksimum dari setiap parameter, serta melihat korelasi antara reward tingkat keberhasilan. Nilai Reward kumulatif dijadikan indikatorr utama dalam menilai efektivitas strategi yang dipelajari oleh Agent.