

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan referensi atau tinjauan pustaka pada penelitian ini, berdasarkan uraian latar belakang sebelumnya, berikut tinjauan pustaka dalam penelitian ini :

2.1.1 Tinjauan Pustaka Terhadap Literatur 1

Oleh (Paxton et al., 2019) melakukan penelitian dengan judul “*Visual Robot Task Planning*”. Dalam penelitian ini, penulis memanfaatkan model generatif untuk mereplikasi hasil dari berbagai aksi tingkat tinggi dalam berbagai lingkungan dan menilai efektivitas aksi-aksi tersebut menggunakan variasi dari teknik pencarian pohon Monte Carlo, guna mengidentifikasi solusi yang efektif untuk masalah khusus yang ditargetkan. Pendekatan yang diambil memfasilitasi visualisasi dari tujuan-tujuan grafis menengah dan mengembangkan kemampuan untuk merencanakan kegiatan kompleks, sehingga memungkinkan generasi dan visualisasi dari rencana-rencana tugas dalam simulasi penumpukan blok sebagai contoh. Peneliti menawarkan suatu solusi untuk memvisualisasikan dampak dari tindakan-tindakan berlevel tinggi di berbagai domain yang berbeda. Hal ini dapat diintegrasikan sebagai komponen dari algoritma perencanaan yang meninjau berbagai kemungkinan masa depan yang potensial, dengan tujuan untuk memilih serangkaian aksi terbaik yang bisa diambil. Peneliti juga mengaplikasikan model pencarian pohon *Monte Carlo* (MCTS) untuk mengidentifikasi aksi-aksi yang akan membawa pada keberhasilan pelaksanaan tugas baru.

2.1.2 Tinjauan Pustaka Terhadap Literatur 2

Oleh (Xiao & Xu, 2019) melakukan penelitian dengan judul “*Web-Based Robot Control Interface*”. Pada penelitian ini, sebuah antarmuka sistem kontrol robot berbasis web dirancang. Antarmuka ini dapat digunakan oleh pengguna untuk mengontrol gerakan robot target, mengamati model 3D, dan memantau nilai sendi robot secara *real-time* melalui PC atau perangkat seluler. Keunggulan sistem ini terletak pada keserbagunaan dan portabilitasnya, memungkinkan pengguna untuk membuka antarmuka untuk mengontrol robot target di mana saja. Sistem ini terutama menggunakan sistem operasi robot (ROS), HTML5, C++, *JavaScript* dan teknologi PHP. Pustaka *JavaScript* yang disediakan oleh otoritas ROS digunakan untuk membangun antarmuka *front-end*. PHP digunakan untuk membuat sistem login pendaftaran pengguna. *Ros3djs* digunakan untuk mewujudkan pembentukan model dinamis simulasi robot. Pustaka C++ yang disediakan oleh ROS digunakan untuk menyinkronkan komunikasi antara simpul ROS dan robot. Akhirnya, sistem interaksi komputer manusia dari tangan robot 16-DOF berhasil diimplementasikan. Menurut hasil, sistem memiliki interaktivitas yang baik, fleksibilitas dan portabilitas. Teknologi Html dan JS yang populer dan library JS yang didukung oleh sistem ROS diadopsi dalam sistem ini untuk mewujudkan sistem kontrol web robot. Dibandingkan dengan sistem kontrol robot tradisional, karena pemisahan klien dan server, sistem dapat digunakan tanpa konfigurasi perangkat lunak dan lingkungan perangkat keras yang rumit, yang mudah digunakan dan cocok untuk dipopulerkan secara luas. Karena independensi *platform* hanya teknologi web, dan kompatibilitas sempurna dengan komputer dan perangkat seluler, sistem ini

memiliki fitur lintas *platform*, yang meningkatkan situasi geografis untuk menggunakan sistem kontrol robot.

2.1.3 Tinjauan Pustaka Terhadap Literatur 3

Oleh (SUCIPTO et al., 2021) melakukan penelitian dengan judul “Gerak Robot Berkaki Dua menggunakan ROS dan RViz sebagai *Visualisasi Interaktif*”. Pada penelitian ini, *visualisasi* interaktif gerak robot berkaki dua menggunakan ROS (*Robot Operation System*) berbasis *library* dan beberapa tools sedangkan RViz merupakan *visualisasi* 3D yang digunakan untuk memvisualisasikan robot dan data sensor *dynamixel*. Dengan membuat beberapa simulasi gerakan yang dilakukan pada RViz kemudian diimplementasikan pada robot. Tingkat keberhasilan dari perencanaan gerakan ini memiliki rata rata error sebesar 1.8%. Pada gerakan condong ke kiri memiliki rata-rata error sebesar 0.83%. Kemudian untuk gerakan condong ke kanan memiliki rata-rata error sebesar 0.84%. Sedangkan untuk gerakan mengangkat satu kaki memiliki rata-rata error sebesar 1.71%. Selain itu, gerakan kaki kanan kedepan memiliki rata-rata error sebesar 3.83%. *Error* tersebut dapat terjadi karena faktor mekanik yang terdapat pada robot seperti *backlash* pada aktuator.

2.1.4 Tinjauan Pustaka Terhadap Literatur 4

Oleh (Jain et al., 2020) melakukan penelitian dengan judul “*Development of Web-based Application for Mobile Robot using IOT Platform*”. Pada penelitian ini, untuk melakukan operasi dengan robot bergerak, sistem otonom dan dikendalikan komputer memberikan solusi terbaik untuk berbagai aplikasi. Untuk memenuhi kebutuhan tersebut, aplikasi berbasis aplikasi web dan seluler memainkan peran penting dalam mengoperasikan robot dan menghadirkan cara baru untuk

melakukannya tugas robot secara mandiri. Robot berbasis aplikasi dikendalikan web dapat digunakan di berbagai bidang seperti industri, pertambangan, medis, operasi militer, pertanian, dan sebagainya. Dalam makalah ini, aplikasi web untuk robot seluler dirancang untuk memahami setiap aspek fungsional robot dan semua kontrol dan visual yang diperlukan dari data sensor dan aktuator oleh pengguna. Antarmuka pengguna dan aplikasi web sedang dikembangkan untuk menentukan kemacetan di sepanjang jalur robot yang dapat mengontrol aplikasi web yang mengaktifkan *internet-of-thing* (IoT) dan untuk menghentikan dan pergerakan kendaraan robot. Pesanan diproses berbasis aplikasi web akan ditransmisikan ke unit kontrol robot melalui wi-fi dan robot ini dapat dikontrol melalui aplikasi web melalui ponsel, Dalam makalah ini dibahas tentang aplikasi mobile robotik berbasis web dengan menggunakan *platform* IoT yang juga dirancang arsitektur sistem untuk memantau dan mengendalikan posisi robot dan kualitas udara di lingkungan. Robot seluler terhubung ke mikrokontroler dan node MCU untuk komunikasi nirkabel dengan sistem jaringan cloud. Di sistem jaringan cloud ini, data berhasil disimpan dan dapat diakses dari jarak jauh melalui sistem berbasis web. Dengan mengembangkan dan mengoperasikan robot seluler melalui aplikasi web, menunjukkan bahwa robot dapat dikendalikan melalui aplikasi seluler Android di mana saja.

2.1.5 Tinjauan Pustaka Terhadap Literatur 5

Oleh (Kashyap et al., 2020) melakukan penelitian dengan judul “Rancang Bangun Gerakan Berjalan pada Robot *Bipedal Antropomorphic* menggunakan *Inverse Kinematics* dan Gait Data”. Dalam penelitian ini membahas mengenai anggota gerak pada robot *bipedal* yaitu kaki. Dimana anggota gerak paling penting

dari robot *humanoid* itu sendiri yang didesain menyerupai layaknya manusia. Oleh karenanya peneliti menggunakan metode *Inverse Kinematic* dan *Forward Kinematic* tujuannya agar dapat menentukan urutan kontrol dari gerakan kaki robot *humanoid*, digunakannya data Antropomorphic yang dijadikan sebagai parameter dari gerak robot. Dan sistem kontrol robot *bipedal* yang menyerupai manusia dengan parameter *gait cycle*. Hasil pengujian dari penelitian ini dinyatakan bahwa Data Trayektori pada gerakan berjalan dapat digunakan sebagai Parameter Kontrol pada robot. Error pada setiap sudut dengan apa yang dihasilkan pada robot yaitu 29,67% untuk hip, 23,45% pada knee dan ankle 11,34%. Perbedaan panjang pada kaki subjek manusia tidak mempengaruhi pada *forward Kinematic* nya, namun diperlukan toleransi pada implementasi robot.

2.2 Sistem

Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain, dan terpadu bertujuan untuk memenuhi suatu proses tertentu (Sutabri et al., 2022).

2.3 Monitoring

Monitoring adalah sebuah siklus kegiatan yang meliputi proses pengumpulan, peninjauan ulang pelaporan dan tindakan atas informasi suatu proses yang sedang diimplementasikan (Sulaeman & Permana, 2021).

2.4 Robot *Humanoid*

Robot *humanoid* adalah robot yang termasuk dalam kategori robot berkaki dengan penampilan keseluruhannya dibentuk menyerupai tubuh manusia yaitu: memiliki dua buah kaki, dua buah tangan, badan, dan kepala. mampu melakukan

interaksi dengan peralatan maupun lingkungan yang *based on human* atau sesuai dengan manusia, robot *humanoid* memiliki fitur-fitur sebagai berikut (Elmer P. Dadios & Ron-Ron G. Garcia, D. Johnson, 2012):

1. Robot *humanoid* bisa bekerja di lingkungan manusia, karena lingkungan yang sudah di desain untuk manusia sehingga akan memakan banyak biaya yang lebih jika harus lingkungan yang menyesuaikan robot.
2. Robot *humanoid* bisa menggunakan peralatan manusia, sama seperti lingkungan, peralatan-peralatan di desain untuk manusia juga.
3. Robot *humanoid* memiliki bentuk seperti manusia.



Gambar 2. 1 Robot *Humanoid*

2.5 CSS

CSS adalah kode yang digunakan untuk mengatur tampilan dokumen HTML seperti: warna, jenis font, spasi antar baris dan lain-lain yang berhubungan untuk meningkatkan tampilan website (Rudjiono & Saputro, 2020).


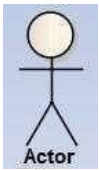
2.6 *Java Script*

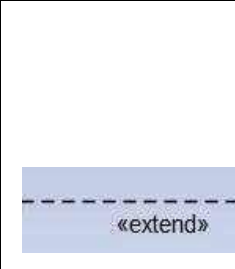




JavaScript adalah bahasa pemrograman atau kode script yang ditempatkan dengan kode HTML atau terpisah yang digunakan untuk membuat tampilan website lebih dinamis (Rudjiono & Saputro, 2020).

2.7 *Use Case Diagram*

Use case diagram (use case) merupakan model untuk perilaku sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi-fungsi apa saja yang ada pada suatu sistem informasi dan siapa yang berhak menggunakan fungsi-fungsi tersebut. Simbol simbol yang digunakan dalam pembuatan *use case* disajikan pada tabel 2.1 berikut (Hafiz Irsyad, 2018).

Tabel 2. 1 *Use Case Diagram*

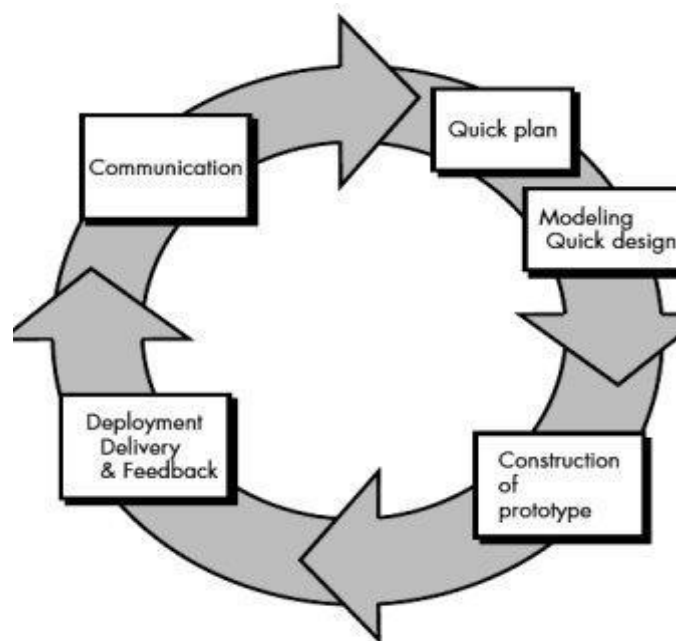
Nama	Simbol	Keterangan
<i>Use case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja frase nama use case.
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat.

Asosiasi		Komunikasi antara aktor dengan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>usecase</i> yang memiliki interaksi dengan aktor.
Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
<i>Generalisasi</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<i>Include</i> (menggunakan)		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang memerlukan usecase ini untuk menjalankan fungsinya atau sebagai syarat dijalkannya <i>usecase</i> ini.
Asosiasi (<i>Association</i>)		Relasi antar kelas dengan makna umum. Biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah (<i>Direct Association</i>)		Relasi antar kelas dengan makna kelas satu digunakan oleh kelas lainnya. Biasanya juga disertai dengan <i>multiplicity</i>

<i>Generalisasi</i>		Relasi antar kelas dengan makna generalisasi-spesialisasi
Ketergantungan (<i>Dependency</i>)		Relasi antar kelas dengan makna kelas kebergantungan antar kelas

2.8 Metode *Prototype*

Prototype merupakan versi awal dari sistem perangkat lunak yang digunakan untuk merancang dan menemukan lebih banyak masalah dan kemungkinan solusi, metode *prototyping* yang digunakan dalam penelitian ini bertujuan untuk mendapatkan gambaran umum tentang aplikasi yang akan dibangun melalui desain aplikasi prototipe terlebih dahulu kemudian akan dievaluasi oleh pengguna. Prototipe aplikasi yang telah dievaluasi oleh pengguna berikutnya akan digunakan sebagai referensi untuk membuat aplikasi yang digunakan sebagai produk akhir sebagai keluaran dari penelitian ini, dapat dilihat pada gambar 2.2 (Nugraha & Syarif, 2018).



Gambar 2. 2 Metode *Prototype*

(Sumber: Nugraha & Syarif, 2018)

Tahapan metode *prototype* dapat dilihat dalam penjelasan dibawah ini :

1. *Communication*

Dimulai dengan tahap komunikasi, tahap ini bertujuan untuk mengidentifikasi berbagai kebutuhan aplikasi yang akan dirancang nantinya dengan melibatkan klien terkait sehingga selama proses desain dapat memberikan hasil yang tepat sesuai keinginan klien yang bersangkutan.

2. *Quick Plan*

Pada tahap *quick plan* ini, perancang perangkat lunak akan melakukan quick plan sesuai dengan spesifikasi kebutuhan pengguna berdasarkan data yang telah dikumpulkan pada tahap komunikasi dengan merancang

desain antarmuka yang dibutuhkan dan kebutuhan pendukung dalam proses ini.

3. *Modeling Quick Design*

Pada tahap ini penulis membuat model yang dibutuhkan dengan waktu perencanaan yang efektif untuk mendeskripsikan kebutuhan pelanggan berdasarkan analisis yang telah dilakukan sebelumnya.

4. *Costruction of Prototype*

Selanjutnya pada tahap ini desainer akan mulai membangun perangkat lunak berdasarkan data-data yang telah dikumpulkan sebelumnya, proses pengembangan ini lebih menitikberatkan pada aspek-aspek utama dari perangkat lunak dengan maksud agar pada proses selanjutnya desainer dapat dengan cepat mendapatkan *feedback* dari *client*. tentang perangkat lunak yang dibuat.

5. *Depluyment Deelivery & Feedback*

Pada tahap ini *prototype* akan diserahkan kepada *client* untuk mendapatkan *feedback* dari hasil *prototype*, *feedback* tersebut akan dijadikan dasar untuk perbaikan *prototype* agar sesuai dengan spesifikasi kebutuhan *client*.