

Nama : Rhonni Irama Noorhuda

NIM : 1103210176

Kelas : TK – 45 – G09

## HASIL ANALISIS SIMULASI

### 1. Analisis terhadap tugas yang telah dikerjakan dari tutorial belajar Rust di Playlist Youtube Learn Rust With Me

Belajar Rust menggunakan Visual Studio Code (VS Code) memberikan pengalaman yang kaya dan efisien bagi pemula dan pengembang berpengalaman. VS Code adalah editor yang ringan dan mudah disesuaikan dengan berbagai ekstensi, salah satunya adalah rust-analyzer yang menyediakan fitur seperti autocompletion, linting, dan debugging untuk pengembangan Rust. Proses pertama yang harus dilakukan adalah menginstal Rust dan mengonfigurasi VS Code dengan ekstensi yang sesuai. Rust, dengan alat manajemen pakatnya, Cargo, memungkinkan pengguna untuk membuat proyek baru, mengelola dependensi, dan membangun aplikasi Rust dengan mudah.

Setelah instalasi, tutorial sering kali mulai dengan mengajarkan cara membuat proyek pertama menggunakan perintah `cargo new`, yang akan menghasilkan struktur proyek dengan file `Cargo.toml` untuk mengelola dependensi dan file `src/main.rs` untuk kode utama. Proses ini memberi pemula pemahaman tentang bagaimana proyek diatur di Rust dan bagaimana manajemen dependensi dilakukan menggunakan Cargo. Meskipun sederhana, ini adalah langkah penting karena membantu pengguna memahami struktur dasar dan pentingnya file konfigurasi dalam proyek Rust.

Selanjutnya, tutorial berfokus pada penulisan kode dasar dalam Rust, seperti program "Hello, World!". Di sini, pengguna diperkenalkan pada konsep dasar pemrograman Rust, seperti deklarasi variabel, tipe data, dan struktur kontrol. Dengan penekanan pada kepastian tipe dan pengelolaan memori yang ketat, Rust mendorong pengembang untuk menulis kode yang lebih aman dan efisien. Proses ini memungkinkan pemula untuk memahami dasar-dasar bahasa sambil menghindari kesalahan umum yang dapat terjadi dalam bahasa lain yang lebih longgar dalam hal tipe data dan manajemen memori.

Seiring dengan berjalannya waktu, tutorial lebih lanjut mengajarkan fitur lebih lanjut seperti pengolahan kesalahan dengan `Result` dan `Option`, konsep kepemilikan dan peminjaman yang unik dalam Rust, serta pengelolaan memori yang tidak memerlukan garbage collection. Tutorial ini memberikan wawasan penting tentang bagaimana Rust memaksa pengembang untuk menulis kode yang lebih aman melalui kompilasi yang ketat. Seiring dengan berjalannya waktu, pemahaman yang lebih dalam mengenai konsep-konsep ini akan membantu pengembang menulis kode yang lebih efisien dan dapat diandalkan.

Dengan penggunaan Visual Studio Code yang didukung oleh ekstensi Rust yang kuat, belajar Rust menjadi lebih menyenangkan dan produktif. Pengguna dapat dengan mudah mengakses dokumentasi dan mendapatkan saran secara langsung melalui fitur

autocompletion dan linting. Namun, tantangan utama bagi pemula mungkin terletak pada penguasaan konsep-konsep fundamental Rust yang lebih kompleks, seperti kepemilikan memori, tetapi dengan pendekatan yang tepat, VS Code menyediakan alat yang sempurna untuk menguasai bahasa ini. Secara keseluruhan, tutorial Rust dengan menggunakan Visual Studio Code memberikan pengalaman yang efisien, mendalam, dan menyeluruh dalam menguasai pengembangan perangkat lunak menggunakan Rust.

## **2. Analisis terhadap tugas membuat dan menjalankan kode-kode robotika di Rust dengan bantuan Generative AI**

- **Perencanaan Jalur Sederhana**  
Prompt ini menekankan pembuatan algoritma perencanaan jalur dalam matriks 2D untuk robot. Fokusnya adalah menghindari rintangan dengan cara menentukan jalur bebas dari titik awal ke tujuan. Ini dapat menggunakan algoritma seperti Breadth-First Search (BFS) atau A\* untuk menyelesaikan masalah.
- **Gerakan Robot dengan Input Pengguna**  
Prompt ini meminta interaktivitas di mana pengguna dapat memasukkan perintah untuk memindahkan robot ke posisi baru. Fungsionalitas ini sangat cocok untuk sistem kontrol manual, yang juga mencetak status posisi terkini robot sebagai umpan balik kepada pengguna.
- **Simulasi Robot Menghindari Rintangan**  
Di sini, robot dirancang untuk bergerak melalui peta 2D sambil menghindari rintangan. Prompt ini memerlukan implementasi algoritma navigasi dinamis, memungkinkan simulasi pergerakan yang lebih realistis dari titik awal ke tujuan.
- **Penjadwalan Robot dengan Prioritas**  
Prompt ini menambahkan kompleksitas dengan mengatur sistem antrian berbasis prioritas untuk tugas robot. Hal ini berguna dalam aplikasi seperti manajemen logistik, di mana robot perlu menyelesaikan tugas berdasarkan kepentingan.
- **Robotik dengan Sistem Event-Driven**  
Sistem berbasis event-driven dalam robotika memungkinkan respons adaptif terhadap perubahan lingkungan. Prompt ini memerlukan implementasi sistem pemantauan dan reaksi terhadap peristiwa, seperti deteksi rintangan atau perubahan tujuan.
- **Robot dengan Model Probabilistik**  
Prompt terakhir mengarah pada penggunaan metode probabilistik, seperti filter partikel atau jaringan Markov. Ini sangat relevan untuk aplikasi yang melibatkan ketidakpastian sensor, di mana robot harus mengidentifikasi jalur terbaik dengan data yang tidak sempurna.