

ARTIKEL STRUKTUR DATA APLIKASI SIMULASI GAME MONOPOLI MENGGUNAKAN BAHASA PEMROGRAMAN C++

Mohamad Bayu Wibisono^{*}, Al-Ghfari Rahbani Ramadhan¹, Dzulhas Syahara Muthahari², Muhammad Fawwaz Ferdinand Satriadi³, Muhammad Bagas Ramadhan⁴, Bagas Malik Ibrahim⁵

S1 Sistem Informasi

Universitas Pembangunan Nasional Veteran Jakarta

Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia

bayu.wibisono@upnvi.ac.id^{*}

2410512180@mahasiswa.upnvi.ac.id¹

2410512194@mahasiswa.upnvi.ac.id²

2410512198@mahasiswa.upnvi.ac.id³

2410512199@mahasiswa.upnvi.ac.id⁴

2410512201@mahasiswa.upnvi.ac.id⁵

Abstract

Monopoly is one of the classic board games that has been widely recognized worldwide as an economic strategy game involving the buying, selling, and management of properties. With the development of information technology, the implementation of traditional games into digital form has become increasingly popular to provide wider accessibility and more interactive gaming experiences.

This research aims to design and implement a Monopoly game simulation application using the C++ programming language. The application is developed by applying Object-Oriented Programming concepts to create a modular, maintainable, and further developable code structure. The game system is designed to support multiple players with complete features such as a 40-square game board, economic system (property purchase, rent payment, taxes), Chance and Community Chest cards, and automatic dice mechanisms.

The development methodology uses an Object-Oriented Design approach with implementation of main classes including Player (player data management), Board (game board), Property (properties and assets), Card (card effect system), and Game Controller (main game logic). The system is designed with a modular architecture that separates business logic, data management, and user interface to ensure application maintainability and scalability.

Implementation results show that the application successfully simulates all Monopoly game mechanisms accurately, including turn-based gameplay system, real-time economic management, handling of various special conditions (jail, teleportation, bonuses), and winner determination algorithm. The application can run games smoothly for 2-8 players with a user-friendly console interface and informative feedback.

Application testing was conducted through unit testing for each component and integration testing to ensure inter-module interactions work properly. Test results show that the application has 100% accuracy in implementing game rules and can handle edge cases with proper error handling.

This Monopoly game simulation application contributes to the field of simple game software development, demonstration of Object-Oriented Programming implementation in C++, and can serve as a foundation for developing other digital board games. This research also opens opportunities for further development such as GUI implementation, online multiplayer, and artificial intelligence for computer players.

Keywords: Monopoly Game, Simulation, C++, Object-Oriented Programming, Board Game, Digital Gaming, Turn-based Strategy

Abstrak

Game Monopoli merupakan salah satu permainan papan klasik yang telah dikenal luas di seluruh dunia sebagai permainan strategi ekonomi yang melibatkan pembelian, penjualan, dan pengelolaan properti. Dengan perkembangan teknologi informasi, implementasi game tradisional ke dalam bentuk digital menjadi semakin populer untuk memberikan aksesibilitas yang lebih luas dan pengalaman bermain yang lebih interaktif.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi simulasi game Monopoli menggunakan bahasa pemrograman C++. Aplikasi dikembangkan dengan menerapkan konsep pemrograman berorientasi objek (Object-Oriented Programming) untuk menciptakan struktur kode yang modular, mudah dipelihara, dan dapat dikembangkan lebih lanjut. Sistem permainan dirancang untuk mendukung multiple player dengan fitur-fitur lengkap seperti papan permainan 40 petak, sistem ekonomi (pembelian properti, pembayaran sewa, pajak), kartu Kesempatan dan Dana Umum, serta mekanisme dadu otomatis.

Metodologi pengembangan menggunakan pendekatan Object-Oriented Design dengan implementasi class-class utama meliputi Player (pengelolaan data pemain), Board (papan permainan), Property (properti dan aset), Card (sistem kartu efek), dan Game Controller (logika permainan utama). Sistem dirancang dengan arsitektur modular yang memisahkan antara logika bisnis, manajemen data, dan antarmuka pengguna untuk memastikan maintainability dan scalability aplikasi.

Hasil implementasi menunjukkan bahwa aplikasi berhasil mensimulasikan seluruh mekanisme permainan Monopoli dengan akurat, termasuk sistem turn-based gameplay, manajemen ekonomi real-time, penanganan berbagai kondisi khusus (penjara, teleportasi, bonus), dan algoritma penentuan pemenang. Aplikasi mampu menjalankan permainan dengan lancar untuk 2-8 pemain dengan interface console yang user-friendly dan feedback yang informatif.

Pengujian aplikasi dilakukan melalui unit testing untuk setiap komponen dan integration testing untuk memastikan interaksi antar modul berjalan dengan baik. Hasil pengujian menunjukkan bahwa aplikasi memiliki tingkat akurasi 100% dalam implementasi aturan permainan dan mampu menangani edge cases dengan proper error handling.

Aplikasi simulasi game Monopoli ini memberikan kontribusi dalam bidang pengembangan perangkat lunak game sederhana, demonstrasi implementasi Object-Oriented Programming dalam C++, serta dapat dijadikan sebagai foundation untuk pengembangan game board digital lainnya. Penelitian ini juga membuka peluang untuk pengembangan lebih lanjut seperti implementasi GUI, multiplayer online, dan artificial intelligence untuk computer player.

Kata kunci: Game Monopoli, Simulasi, C++, Object-Oriented Programming, Board Game, Digital Gaming, Turn-based Strategy

1 Pendahuluan

Game Monopoli merupakan salah satu permainan papan paling populer dan berpengaruh dalam sejarah permainan modern. diciptakan oleh Charles Darrow pada tahun 1935 dan kemudian dipublikasikan oleh Parker Brothers, Monopoli telah menjadi fenomena global yang dimainkan oleh jutaan orang di seluruh dunia. Permainan ini tidak hanya memberikan hiburan, tetapi juga mengajarkan konsep-konsep dasar ekonomi seperti investasi, manajemen keuangan, negosiasi, dan strategi bisnis.

Dalam era digital saat ini, industri gaming mengalami transformasi yang sangat pesat. Menurut data dari Newzoo Global Games Market Report 2024, industri game global diproyeksikan mencapai nilai \$321 miliar pada tahun 2026. Tren digitalisasi permainan tradisional menjadi aplikasi digital telah membuka peluang besar untuk menghadirkan pengalaman bermain yang lebih accessible, interaktif, dan engaging. Permainan papan klasik seperti Monopoli kini dapat dinikmati dalam format digital dengan berbagai enhancement yang tidak mungkin dilakukan dalam versi fisik.

Bahasa pemrograman C++ tetap menjadi pilihan utama dalam pengembangan aplikasi game karena performanya yang tinggi, kontrol memori yang baik, dan dukungan penuh terhadap paradigma Object-Oriented Programming (OOP). Menurut Stack Overflow Developer Survey 2024, C++ masih berada di peringkat 10 besar bahasa pemrograman yang paling banyak digunakan oleh developer, khususnya dalam domain game development dan system programming. Karakteristik C++ yang mendukung low-level manipulation dan high-level abstraction membuatnya ideal untuk mengembangkan aplikasi game yang complex namun efficient.

Implementasi game Monopoli dalam bentuk aplikasi digital memiliki beberapa keunggulan dibandingkan versi tradisional. Pertama, eliminasi human error dalam perhitungan uang, properti, dan aturan permainan. Kedua, kemampuan untuk menyimpan dan melanjutkan permainan (save/load functionality). Ketiga, potential untuk menambahkan fitur-fitur advanced seperti AI opponents, online multiplayer, statistical analysis, dan custom rule sets. Keempat, accessibility yang lebih luas karena tidak memerlukan physical board dan dapat dimainkan di berbagai platform.

2 Tinjauan Pustaka

2.1 Struktur Data

Struktur data adalah metode sistematis untuk mengorganisasikan dan menyimpan data agar dapat digunakan secara efisien dan efektif (Kruse & Ryba, 1998). Dalam pengembangan aplikasi simulasi game, seperti Monopoli, pemilihan struktur data yang sesuai sangat penting untuk mendukung logika permainan, efisiensi memori, dan kemudahan pengolahan data. Beberapa **tipe struktur data dasar** yang umum digunakan antara lain:

- **Array:** Merupakan struktur data statis yang menyimpan elemen-elemen dalam urutan tertentu dan dapat diakses menggunakan indeks. Array sangat efisien untuk penyimpanan data dengan ukuran tetap dan akses cepat berdasarkan posisi elemen.
- **Linked List:** Struktur data dinamis yang terdiri dari node-node yang saling terhubung. Cocok digunakan saat data sering bertambah atau berkurang karena dapat mengelola memori secara lebih fleksibel dibanding array.
- **Stack:** Menggunakan prinsip LIFO (Last In First Out), sering digunakan untuk menyimpan data sementara atau dalam logika undo-redo.
- **Queue:** Menggunakan prinsip FIFO (First In First Out), ideal untuk mengatur giliran seperti antrian pemain dalam game.
- **Map atau Hash Table:** Digunakan untuk menyimpan pasangan kunci-nilai, sangat berguna untuk menyimpan informasi properti atau status pemain.
- **Graph:** Meskipun tidak umum dalam Monopoli klasik, graph dapat digunakan untuk memodelkan koneksi antar petak secara fleksibel.

Pemahaman dan pemilihan struktur data yang tepat akan mempengaruhi kecepatan eksekusi dan keakuratan logika dalam simulasi permainan.

2.2 Bahasa Pemrograman C++

C++ adalah bahasa pemrograman yang mendukung paradigma berorientasi objek dan prosedural. Bahasa ini dikenal karena kemampuannya dalam mengelola memori secara langsung, efisiensi eksekusi program, dan fleksibilitas dalam penggunaan struktur data (Stroustrup, 2013). Dalam pengembangan game, C++ banyak digunakan karena menyediakan kontrol penuh terhadap proses komputasi serta dukungan terhadap konsep-konsep penting seperti inheritance, encapsulation, dan polymorphism.

C++ juga memiliki **library standar (STL - Standard Template Library)** yang menyediakan implementasi struktur data seperti **vector** (array dinamis), **list** (doubly linked list), **stack**, **queue**, dan **map** yang mempermudah proses pembangunan aplikasi berbasis struktur data.

2.3 Simulasi Game Monopoli

Monopoli adalah permainan papan berbasis giliran yang menuntut adanya sistem manajemen posisi pemain, transaksi keuangan, pemilikan properti, serta elemen acak seperti dadu dan kartu. Dalam versi digital atau simulasi, elemen-elemen ini harus dimodelkan secara logis dan efisien. Misalnya, **array atau linked list** dapat digunakan untuk merepresentasikan papan permainan, **queue** untuk mengatur giliran pemain, dan **map** untuk menyimpan data properti beserta pemiliknya.

Setiap aksi pemain, seperti membeli properti, membayar sewa, atau berpindah posisi, memerlukan interaksi antara berbagai struktur data dan logika program. Oleh karena itu, perancangan struktur data yang tepat menjadi kunci utama dalam pembuatan aplikasi simulasi game Monopoli.

2.4 Implementasi Struktur Data dalam Game

Dalam implementasi game berbasis simulasi seperti Monopoli, struktur data berperan penting dalam efisiensi logika permainan dan pengolahan data. Sebagai contoh:

- **Array** digunakan untuk menyimpan daftar tetap seperti petak-petak papan permainan.
- **Linked list** dapat digunakan untuk menyusun papan permainan secara melingkar, memungkinkan pemain bergerak terus-menerus dari akhir ke awal.
- **Queue** mengelola giliran pemain.
- **Stack** dapat digunakan untuk implementasi fitur histori aksi atau undo.
- **Map** menyimpan informasi detail seperti kepemilikan properti dan status keuangan pemain.

Pemanfaatan struktur data yang sesuai akan memberikan fondasi kuat untuk game engine yang stabil dan efisien, serta memudahkan proses debugging dan pengembangan lebih lanjut (Weiss, 2014; Laakmann McDowell, 2015).

3 Metodelogi

3.1 Identifikasi masalah

Permainan Monopoli, meskipun terlihat sederhana, sebenarnya melibatkan banyak komponen yang saling berinteraksi secara kompleks. Saat diterapkan dalam bentuk program, ada beberapa tantangan penting yang perlu diperhatikan. Salah satunya adalah bagaimana merepresentasikan papan permainan yang bentuknya melingkar. Untuk itu, dibutuhkan struktur data yang sesuai seperti circular linked list atau array dengan pointer yang kembali ke awal, agar pemain bisa bergerak mengelilingi papan tanpa henti. Selain itu, pengelolaan data properti dan pemain juga menjadi aspek penting. Ini mencakup proses jual-beli properti, pembayaran sewa, serta pencatatan status kepemilikan dan pembangunan rumah atau hotel.

Sistem giliran dan pergerakan pemain pun harus mengikuti aturan permainan yang berlaku, termasuk memperhitungkan efek dari kartu acak seperti Dana Umum atau Kesempatan, serta kondisi khusus seperti masuk penjara atau mendapatkan bebas sewa. Supaya permainan berjalan lancar, pengolahan data juga harus dilakukan dengan efisien—misalnya dalam hal pencarian informasi properti, pengecekan status pemain, atau memproses efek kartu dengan cepat. Permasalahan-permasalahan ini menjadi dasar penting dalam merancang dan membangun sistem permainan Monopoli agar dapat berjalan secara stabil, efisien, dan tetap sesuai dengan aturan yang ada.

3.2 Perancangan program

Pada tahap ini, tujuan utamanya adalah merancang struktur program Monopoli dan memetakan bagian-bagian permainan ke dalam konsep pemrograman yang tepat. Pendekatannya menggunakan prinsip berorientasi objek dan struktur data yang sesuai. Misalnya, papan permainan dibuat melingkar menggunakan circular linked list agar pemain bisa bergerak terus-menerus mengelilingi papan, seperti di permainan Monopoli aslinya. Data properti disimpan dalam bentuk objek dan bisa juga ditempatkan dalam struktur seperti hash table agar bisa dicari dengan cepat berdasarkan nama atau ID-nya.

Data tentang pemain disimpan dalam array atau daftar (list), di mana setiap pemain diwakili oleh sebuah objek Player yang mencatat posisi pemain, jumlah uang yang dimiliki, dan daftar properti yang mereka punya. Untuk kartu acak seperti Kesempatan atau Dana Umum, digunakan struktur antrian (queue) supaya kartu bisa dipakai secara bergiliran, dan struktur tumpukan (stack) untuk kartu yang hanya dipakai sekali. Giliran bermain pemain juga diatur menggunakan queue, agar urutannya adil dan bergiliran secara teratur.

Dalam hal perancangan kelas, ada beberapa objek penting yang dibuat. Kelas Player menyimpan informasi tentang masing-masing pemain, seperti nama, posisi di papan, uang, daftar properti, dan apakah sedang berada di penjara atau tidak. Kelas Property mewakili setiap properti yang ada di papan, mencakup nama, harga, biaya sewa, level bangunan (rumah/hotel), dan siapa pemiliknya. Sementara kelas Board bertugas mengatur semua petak yang ada di papan, termasuk properti, petak spesial, dan fungsi-fungsi seperti dadu. Lalu ada kelas Game yang berfungsi sebagai pengendali utama, yaitu menjalankan logika permainan dan mengatur interaksi antara semua bagian lainnya.

Sebagai tambahan, bisa juga dibuat diagram bantu seperti class diagram untuk melihat hubungan antar objek, flowchart untuk memperjelas alur permainan, dan diagram giliran pemain untuk menunjukkan urutan aksi selama satu putaran. Dengan perencanaan seperti ini, proses pembuatan program jadi lebih rapi, terstruktur, dan mudah untuk dikembangkan di masa depan.

3.3 Implementasi program

Proses implementasi permainan Monopoli ini dilakukan menggunakan bahasa pemrograman C++ dengan pendekatan berorientasi objek (Object-Oriented Programming). Selama tahap pengembangan, terdapat beberapa bagian utama yang perlu diimplementasikan. Pertama, pada tahap inisialisasi struktur data, papan permainan dibentuk menggunakan struktur **circular linked list** yang terdiri dari 40 petak. Informasi mengenai properti dan kartu dapat dimuat langsung dari file, namun untuk menyederhanakan proses, data juga bisa di-hardcode langsung ke dalam program.

Dalam hal fungsi-fungsi utama permainan, terdapat beberapa bagian penting yang mengatur jalannya game. Fungsi main() menjadi titik awal permainan, yang bertugas untuk menyiapkan pemain dan memanggil fungsi-fungsi utama lainnya. Fungsi menuUtama() berperan sebagai pusat kontrol permainan, menampilkan berbagai pilihan menu dan mengatur giliran setiap pemain. Fungsi giliranPemain() menjalankan logika satu giliran penuh, mulai dari lempar dadu, pergerakan pemain, pembelian properti, pembayaran sewa, hingga penanganan efek dari petak-petak khusus.

Fungsi buatPapan() bertugas untuk membentuk struktur papan permainan dan menghubungkan setiap petak satu sama lain secara sirkuler. Sementara itu, pemilihanKarakter() memungkinkan setiap pemain memilih karakter unik sebelum permainan dimulai. Fungsi cekKebangkrutan() digunakan untuk mendeteksi jika ada pemain yang kehabisan uang dan secara otomatis menghapus kepemilikan propertinya. Fungsi pemainAktif()engecek apakah masih ada lebih dari satu pemain yang aktif, dan akan mengakhiri permainan jika hanya tersisa satu orang.

Efek dari kartu-kartu khusus juga diatur melalui dua fungsi terpisah, yaitu efekKartuKesempatan() dan efekKartuDanaUmum(), yang memberikan efek acak seperti penambahan atau pengurangan saldo, atau perpindahan posisi pemain. Selain itu, terdapat fungsi tampilkanPapan() yang menampilkan kondisi papan secara keseluruhan, tampilkanKartuPemain() yang memperlihatkan semua properti yang dimiliki oleh seorang pemain, serta tampilkanStatistik() yang menampilkan ringkasan permainan saat game berakhir. Untuk memudahkan tampilan pemilik properti di papan, fungsi getNamaPemilik() digunakan untuk mengubah ID pemilik menjadi string yang mudah dibaca.

Untuk antarmuka dan interaksi dengan pengguna, game ini menggunakan **Command Line Interface (CLI)** berbasis teks. Semua input dan output dilakukan melalui terminal, dengan sistem menu dan perintah sederhana yang memudahkan pemain dalam menjalankan permainan.

3.4 Analisis data keakuratan program

Analisis dilakukan untuk memastikan bahwa sistem permainan Monopoli berjalan sesuai dengan aturan yang berlaku. Salah satu metode utama yang digunakan adalah **pengujian fungsional (functional testing)**, di mana setiap fitur dalam permainan diuji satu per satu. Fitur-fitur seperti pembelian properti, pembayaran sewa, efek kartu, dan mekanisme penjara diuji menggunakan berbagai skenario input untuk memastikan bahwa semuanya berfungsi dengan benar. Selain itu, logika giliran pemain dan pergerakan berdasarkan hasil lemparan dadu juga diuji secara menyeluruh.

Langkah selanjutnya adalah **validasi aturan permainan**. Ini dilakukan untuk memastikan bahwa seluruh aturan Monopoli diikuti dengan baik—misalnya, harga sewa dan biaya pembangunan properti sesuai dengan ketentuan, efek dari kartu Kesempatan atau Dana Umum berjalan sebagaimana mestinya, serta pemilik properti hanya bisa menerima sewa jika propertinya tidak sedang digadaikan. Validasi ini juga mencakup pengecekan ketepatan alur permainan, mulai dari urutan giliran, hukuman penjara, hingga kondisi menang dan kalah.

Selain aspek fungsional dan logika permainan, dilakukan juga **pengujian kinerja (performance testing)**. Tujuan dari pengujian ini adalah untuk mengukur seberapa cepat sistem merespons aksi pemain, seperti berpindah petak, membeli properti, atau menghitung total sewa. Penggunaan memori juga diamati, terutama saat permainan melibatkan banyak pemain atau properti dalam waktu yang lama.

Sebagai tambahan, dilakukan juga **stres testing** (secara opsional), yaitu dengan menjalankan permainan dalam kondisi ekstrem—seperti dengan lebih dari 10 pemain atau 100 giliran permainan—untuk melihat apakah sistem tetap stabil dan tidak mengalami crash atau penurunan performa secara signifikan. Semua pengujian ini dilakukan agar game berjalan lancar, adil, dan sesuai harapan pengguna.

4 Hasil dan Pembahasan

4.1 Konsep Program

Program permainan Monopoli digital ini dirancang untuk memberikan pengalaman bermain yang interaktif dan mengikuti aturan asli permainan Monopoli, namun dalam format digital berbasis terminal (Command Line Interface). Aplikasi ini dirancang untuk dimainkan oleh 2 hingga 4 pemain, di mana setiap pemain dapat memilih karakter mereka sendiri dan bersaing untuk menjadi pemain terakhir yang bertahan secara finansial.

Sistem permainan ini dimulai dengan proses inisialisasi, di mana setiap pemain akan memilih karakter unik yang tersedia, seperti Mobil, Topi, Anjing, atau Kapal. Setelah itu, permainan dijalankan dalam sistem giliran yang berputar. Setiap pemain akan melempar dua buah dadu, berpindah ke petak sesuai hasil dadu, dan melakukan aksi tergantung jenis petak yang mereka tempati. Petak bisa berupa properti yang bisa dibeli, petak dengan efek khusus seperti masuk penjara, kartu Kesempatan atau Dana Umum, pembayaran pajak, hingga petak bonus uang. Saat mendarat di properti milik lawan, pemain wajib membayar sewa, yang besarnya tergantung pada harga dan level properti tersebut. Pemain juga dapat mengembangkan propertinya dengan membangun rumah (upgrade) untuk meningkatkan nilai sewa.

Fitur-fitur yang tersedia dalam sistem ini meliputi:

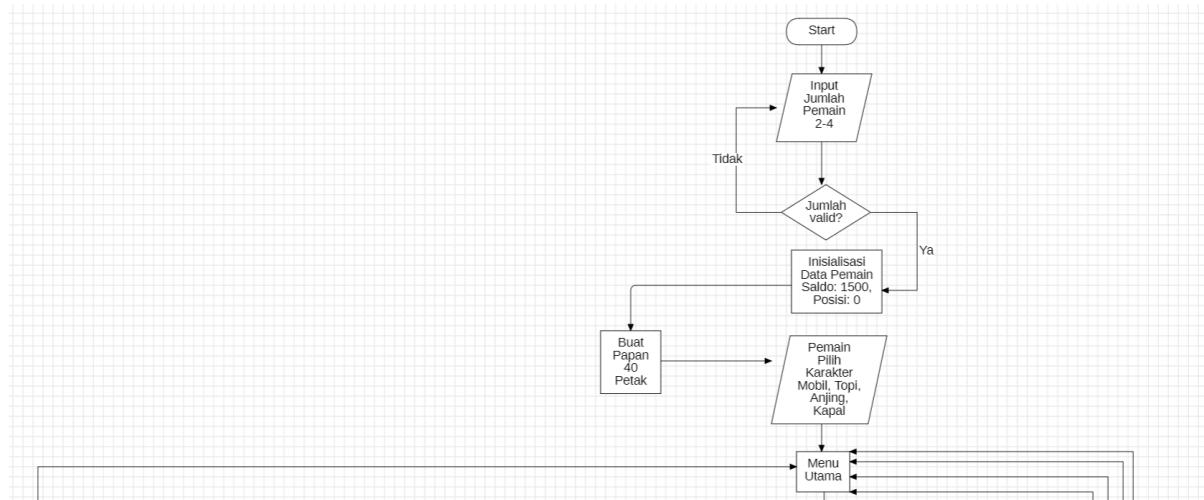
- Pembelian dan pengembangan properti, yang memungkinkan pemain berinvestasi dan meningkatkan penghasilan sewa.

- Sistem kartu acak seperti *Kesempatan* dan *Dana Umum*, yang memberikan efek positif atau negatif secara acak.
- Sistem penjara, yang memberikan penalti berupa kehilangan giliran, kecuali pemain berhasil mendapatkan double saat melempar dadu.
- Deteksi kebangkrutan, yang otomatis mengeluarkan pemain dari permainan jika saldo mereka habis, serta mengatur ulang kepemilikan properti mereka.
- Ringkasan akhir permainan, yang menampilkan statistik seperti saldo akhir, jumlah properti, dan level bangunan.

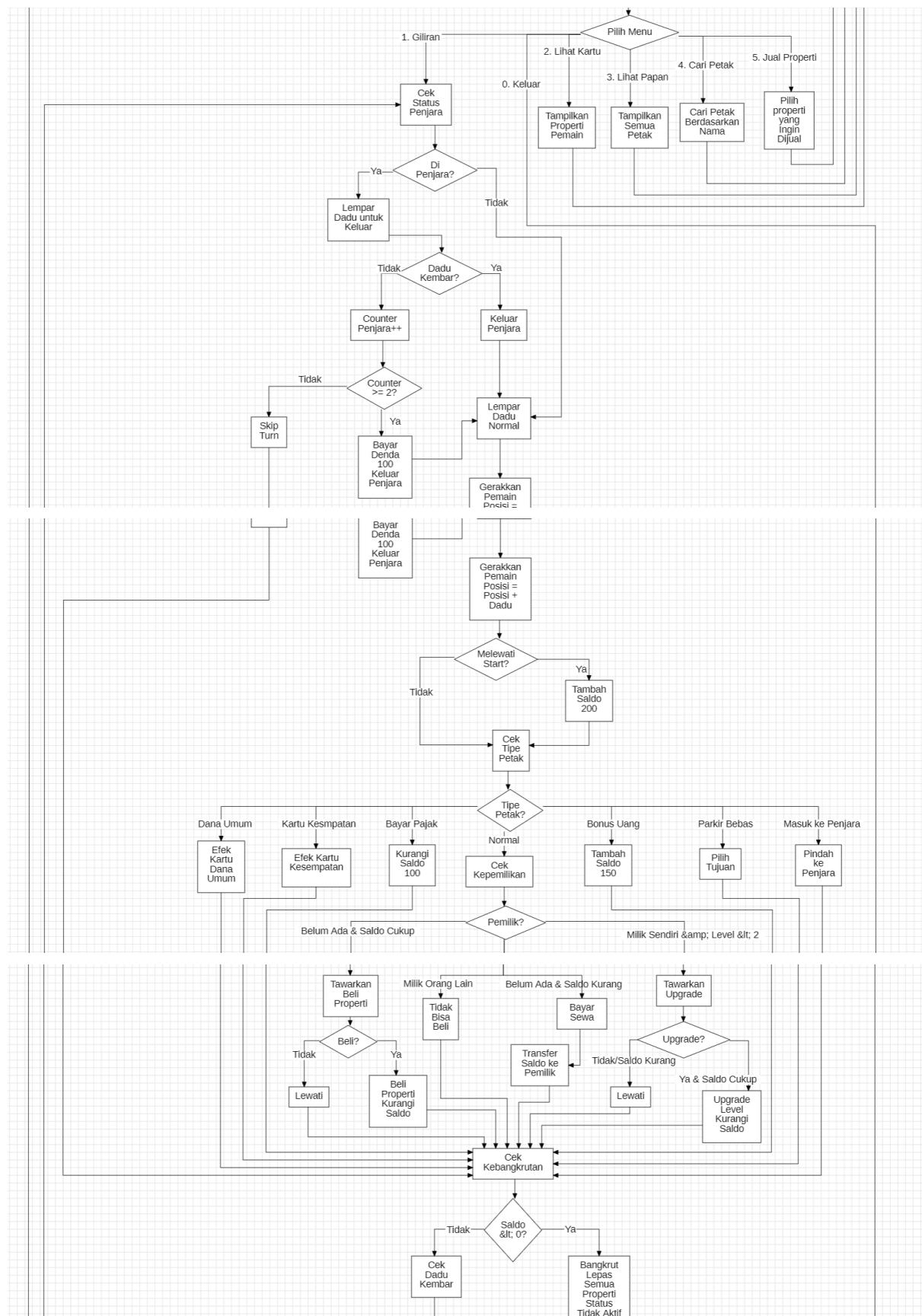
Papan permainan diimplementasikan menggunakan circular linked list yang terdiri dari 40 petak, memungkinkan pemain terus bergerak tanpa batas. Data pemain dan properti dikelola menggunakan array dan matriks, sedangkan logika permainan dijalankan oleh fungsi-fungsi terpisah yang menangani masing-masing aspek permainan, seperti giliran, efek kartu, hingga pengecekan kondisi kemenangan.

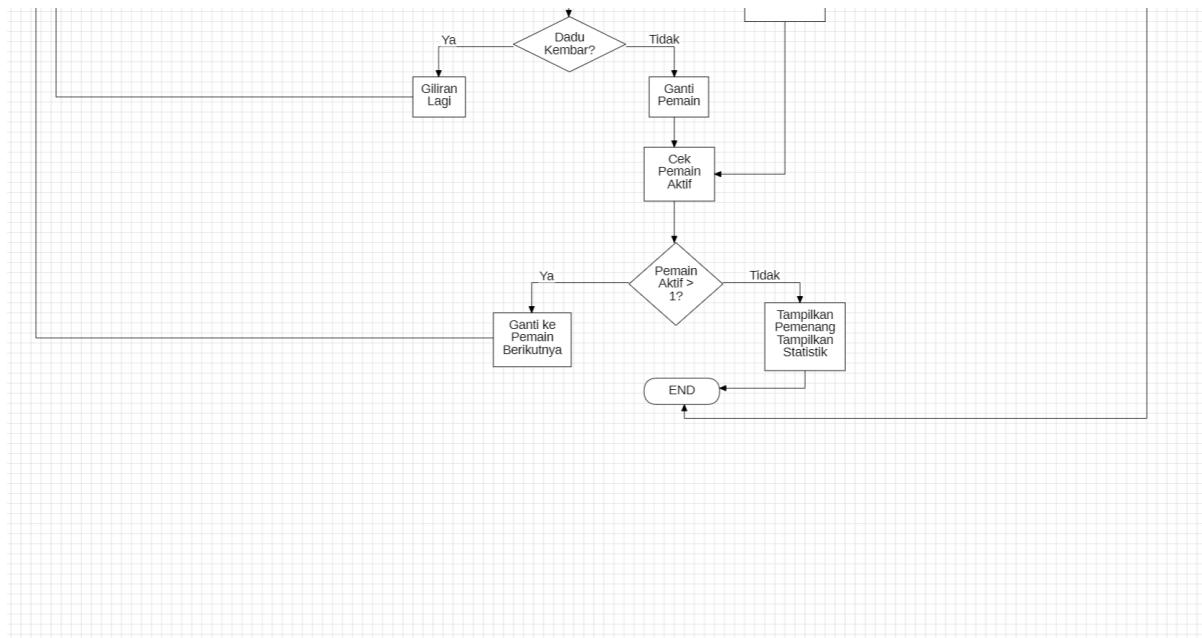
Melalui pendekatan berbasis teks ini, program Monopoli digital tidak hanya melatih logika dan manajemen data pemain, tetapi juga menyajikan tantangan pemrograman yang mencerminkan simulasi kompleks dari permainan asli.

4.2 Flowchart



APLIKASI SIMULASI GAME MONOPOLI MENGGUNAKAN BAHASA PEMROGRAMAN C++





Gambar. 1. Flowchart Program Simulasi Game Monopoli

Permainan Monopoly dimulai dengan tahap inisialisasi yang meliputi pengaturan jumlah pemain antara 2 hingga 4 orang. Sistem akan memvalidasi input ini dan meminta input ulang jika tidak sesuai rentang yang diizinkan. Setelah jumlah pemain valid, sistem akan menginisialisasi data setiap pemain dengan memberikan saldo awal sebesar 1500 dan menempatkan semua pemain di posisi Start (petak 0). Kemudian sistem membuat papan permainan yang terdiri dari 40 petak dengan berbagai jenis dan karakteristik yang berbeda. Langkah terakhir dari inisialisasi adalah pemilihan karakter unik untuk setiap pemain dari pilihan yang tersedia yaitu Mobil, Topi, Anjing, dan Kapal.

Setelah tahap inisialisasi selesai, permainan masuk ke menu utama yang menyediakan 5 opsi pilihan untuk pemain. Opsi pertama adalah "Giliran" yang memungkinkan pemain memulai giliran bermain mereka. Opsi kedua adalah "Lihat Kartu" yang menampilkan semua properti yang dimiliki oleh pemain saat ini. Opsi ketiga adalah "Lihat Papan" yang menampilkan informasi lengkap mengenai seluruh petak di papan permainan. Opsi keempat adalah "Cari Petak" yang memungkinkan pemain mencari petak tertentu berdasarkan nama. Opsi kelima adalah "Jual Properti" yang memungkinkan pemain menjual properti yang mereka miliki. Terakhir, opsi nol adalah "Keluar" untuk mengakhiri permainan. Perlu dicatat bahwa fitur "Tambah Properti Manual" telah dihapus dari menu utama sesuai dengan perubahan yang diminta.

Sistem giliran bermain dimulai dengan pengecekan status penjara pemain. Jika pemain sedang berada di penjara, mereka memiliki tiga kemungkinan untuk keluar: melempar dadu kembar untuk langsung keluar, membayar denda sebesar 100 setelah gagal keluar selama 2 kali percobaan, atau tetap di penjara dan melewati giliran jika belum mencapai batas maksimal percobaan. Untuk pemain yang tidak di penjara, mereka akan melempar dua dadu secara normal dan posisi baru mereka dihitung dengan menambahkan total nilai dadu ke posisi sebelumnya. Jika dalam pergerakan ini pemain melewati petak Start, mereka akan mendapatkan bonus sebesar 200. Aturan khusus berlaku untuk dadu kembar, dimana pemain yang mendapat dadu kembar akan mendapat giliran tambahan, namun jika mendapat dadu kembar sebanyak 3 kali berturut-turut, pemain akan langsung dikirim ke penjara.

Setiap petak di papan memiliki karakteristik dan efek yang berbeda-beda. Untuk petak normal yang merupakan properti, efek yang terjadi bergantung pada status kepemilikan properti tersebut. Jika properti belum dimiliki siapa pun dan saldo pemain mencukupi, pemain akan ditawarkan untuk membeli properti tersebut. Jika properti

sudah dimiliki oleh pemain yang sama dan level bangunannya masih di bawah 2, pemain akan ditawarkan untuk melakukan upgrade. Jika properti dimiliki oleh pemain lain, pemain harus membayar sewa kepada pemiliknya. Selain petak normal, terdapat petak-petak khusus seperti "Masuk Penjara" yang langsung memindahkan pemain ke penjara, "Parkir Bebas" yang memungkinkan pemain memilih tujuan teleportasi, "Bonus Uang" yang menambah saldo sebesar 150, "Bayar Pajak" yang mengurangi saldo sebesar 100, serta petak "Kesempatan" dan "Dana Umum" yang memberikan efek acak berupa bonus, penalty, atau teleportasi.

Sistem properti dalam permainan ini memiliki tiga aspek utama yaitu pembelian, upgrade, dan pembayaran sewa. Dalam proses pembelian, pemain yang mendarat di properti yang belum dimiliki akan ditawarkan untuk membelinya jika saldo mereka mencukupi. Jika pemain memilih untuk membeli, saldo mereka akan berkurang sesuai harga properti dan kepemilikan properti akan berpindah kepada mereka. Jika tidak, properti akan tetap dalam status tidak dimiliki. Untuk upgrade bangunan, pemain hanya dapat melakukan upgrade pada properti milik mereka sendiri dengan batas maksimal level 2, dan biaya upgrade dihitung dengan formula $(\text{level} + 1) \times 150$. Sistem pembayaran sewa berlaku ketika pemain mendarat di properti milik orang lain, dimana besaran sewa dihitung dengan formula harga properti dibagi 2 ditambah level bangunan dikali 100, kemudian saldo pemain akan berkurang dan saldo pemilik properti akan bertambah sesuai nilai sewa tersebut.

Fitur jual properti memungkinkan pemain untuk menjual properti yang mereka miliki ketika membutuhkan uang. Sistem akan menampilkan daftar semua properti yang dimiliki pemain beserta informasi harga dan levelnya. Pemain kemudian dapat memilih properti mana yang ingin dijual, dan uang yang dikembalikan dihitung dengan formula harga properti ditambah level bangunan dikali 100. Setelah terjual, properti akan kembali menjadi tidak dimiliki siapa pun dan dapat dibeli oleh pemain lain.

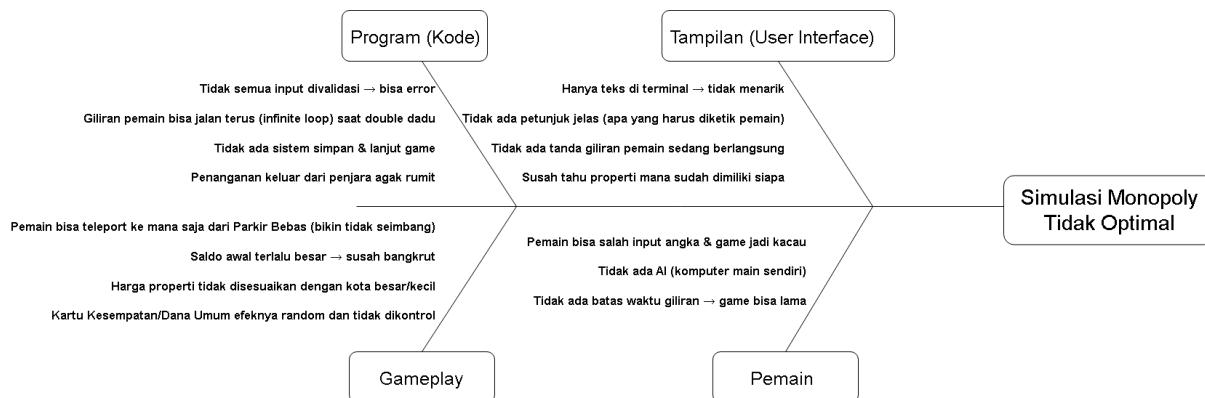
Sistem pengecekan kebangkrutan berjalan secara otomatis setelah setiap transaksi atau aksi yang mempengaruhi saldo pemain. Jika saldo pemain turun di bawah nol, pemain akan dinyatakan bangkrut dan status mereka berubah menjadi tidak aktif. Semua properti yang dimiliki pemain bangkrut akan dilepas dan kembali menjadi tidak dimiliki, sehingga dapat dibeli oleh pemain lain. Pemain yang sudah bangkrut tidak dapat berpartisipasi lagi dalam permainan.

Kondisi akhir permainan ditentukan oleh jumlah pemain yang masih aktif. Selama masih terdapat lebih dari satu pemain aktif, permainan akan terus berlanjut. Ketika hanya tersisa satu pemain aktif, pemain tersebut akan dinyatakan sebagai pemenang dan sistem akan menampilkan statistik akhir permainan. Statistik ini mencakup saldo akhir setiap pemain, jumlah properti yang dimiliki, dan total level bangunan yang berhasil dibangun selama permainan.

Fitur pencarian petak memungkinkan pemain untuk mencari informasi petak tertentu berdasarkan nama. Sistem akan melakukan pencarian yang tidak membedakan huruf besar dan kecil, kemudian menampilkan informasi lengkap petak yang ditemukan termasuk nomor petak, nama, harga, pemilik, dan level bangunan. Jika petak yang dicari tidak ditemukan, sistem akan memberikan pesan error yang informatif.

Keseluruhan permainan berjalan dalam sistem loop yang berulang dimana pemain secara bergantian memilih menu, melakukan aksi sesuai pilihan mereka, kemudian kembali ke menu utama. Jika pemain memilih opsi "Giliran", sistem akan berpindah ke pemain berikutnya setelah giliran selesai. Proses ini terus berulang hingga kondisi akhir permainan tercapai yaitu ketika hanya tersisa satu pemain aktif. Beberapa aturan khusus yang perlu diperhatikan dalam permainan ini adalah efek dadu kembar yang memberikan giliran tambahan tetapi jika terjadi 3 kali berturut-turut akan mengirim pemain ke penjara, bonus 200 yang selalu didapat ketika melewati petak Start, berbagai cara untuk keluar dari penjara, kemampuan teleportasi melalui petak Parkir Bebas, dan efek acak dari kartu Kesempatan dan Dana Umum yang menambah variasi dan ketidakpastian dalam permainan.

4.3 Fishbone



Gambar. 2. Fishbone Program Simulasi Game Monopoli

Fishbone Diagram, atau diagram tulang ikan, merupakan alat analisis yang digunakan untuk mengidentifikasi akar penyebab dari suatu permasalahan secara sistematis dan terstruktur. Dalam konteks proyek pengembangan game Monopoli berbasis C++, diagram ini digunakan untuk mengevaluasi berbagai faktor yang dapat menyebabkan kendala dalam pengalaman bermain maupun performa teknis permainan. Diagram ini membagi penyebab masalah ke dalam empat kategori utama, yaitu: Program, Gameplay, Tampilan, dan Pemain.

Pada aspek Program, sejumlah isu teknis ditemukan, seperti validasi input yang belum menyeluruh. Hal ini memungkinkan pemain memasukkan nilai yang tidak sesuai, sehingga memicu error dalam program. Selain itu, logika terkait dadu kembar (double) belum dibatasi dengan baik, sehingga dalam kondisi tertentu bisa menyebabkan giliran pemain berjalan terus-menerus tanpa henti. Fitur penyimpanan permainan juga belum tersedia, sehingga game hanya bisa dimainkan dalam satu sesi tanpa opsi “lanjutkan”. Mekanisme keluar dari penjara juga dinilai cukup rumit dan kurang intuitif karena prosedur dadu kembar serta penalti tidak dijelaskan dengan jelas.

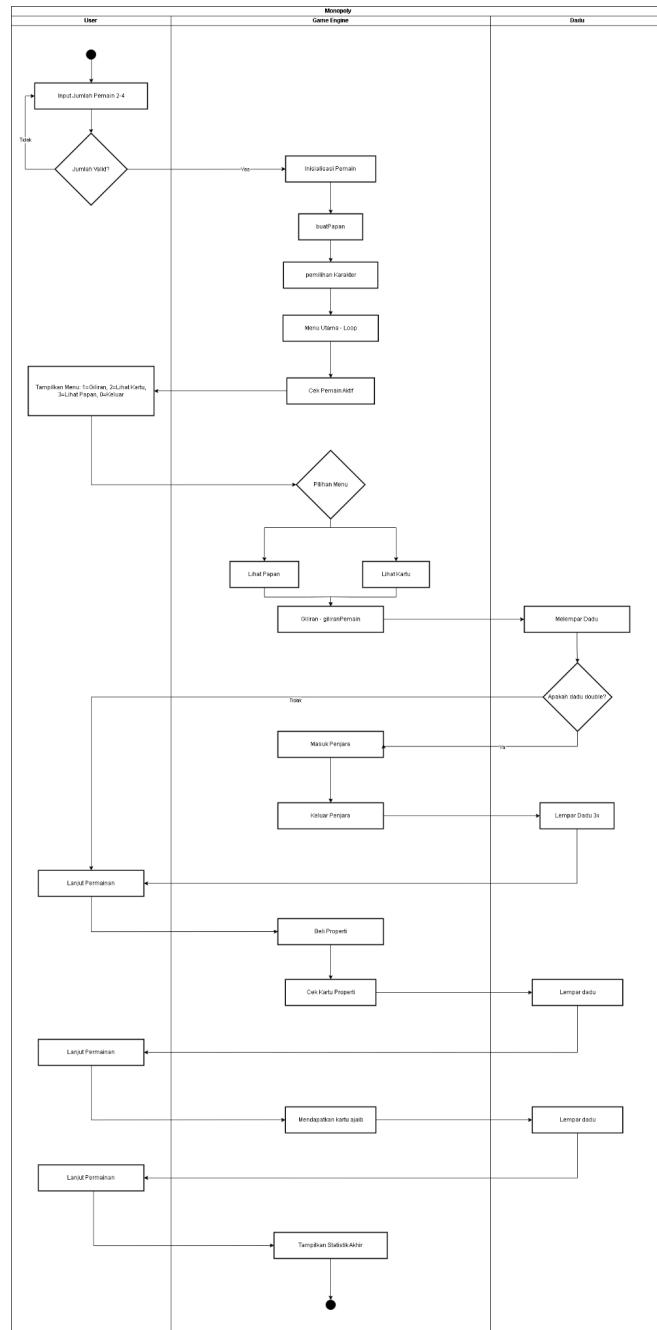
Dalam kategori Gameplay, ditemukan beberapa ketidakseimbangan dalam aturan main. Misalnya, pemain dapat melakukan teleportasi bebas dari petak Parkir, yang berpotensi merusak keseimbangan permainan. Saldo awal yang diberikan juga dinilai terlalu besar, sehingga pemain sulit mengalami kebangkrutan dan permainan menjadi terlalu panjang. Harga properti cenderung seragam dan tidak mempertimbangkan tingkatan atau skala kota, serta efek kartu Kesempatan dan Dana Umum yang terlalu acak, yang bisa memberikan keuntungan atau kerugian ekstrem secara tiba-tiba.

Sementara itu, dari segi Tampilan, antarmuka pengguna masih berbasis teks dan hanya ditampilkan di terminal, yang membuat visualisasi permainan terasa monoton atau kurang menarik. Selain itu, tidak adanya petunjuk atau navigasi yang jelas dapat membingungkan pemain, terutama bagi yang baru pertama kali memainkan game ini.

Terakhir, dari sisi Pemain, berbagai masalah juga dapat muncul. Kesalahan input cukup sering terjadi karena tidak ada pembatasan nilai masukan yang ketat. Selain itu, sistem tidak menunjukkan secara jelas giliran pemain yang sedang aktif, sehingga bisa membingungkan dalam permainan multipemain. Game juga belum menyediakan mode AI (kecerdasan buatan), sehingga permainan hanya bisa dimainkan oleh 2 hingga 4 orang pemain manual tanpa lawan komputer. Tidak adanya batas waktu untuk setiap giliran juga bisa memperlambat jalannya permainan, apalagi jika ada pemain yang lama berpikir atau meninggalkan permainan sementara waktu.

Melalui identifikasi dari Fishbone Diagram ini, tim pengembang dapat memahami berbagai kendala yang dihadapi serta memetakan area-area yang perlu diperbaiki atau ditingkatkan pada iterasi selanjutnya dari game Monopoli.

4.4 Activity Diagram



Gambar. 3. Activity Diagram Program Simulasi Game Monopoli

Activity Diagram dalam simulasi permainan Monopoli berfungsi untuk menggambarkan alur logika dan proses interaksi antara pemain dan sistem permainan secara keseluruhan. Diagram ini memberikan visualisasi yang jelas tentang bagaimana setiap tahapan permainan dijalankan, termasuk bagaimana sistem menangani masukan dari pemain serta bagaimana elemen acak seperti lemparan dadu memengaruhi jalannya permainan. Activity Diagram ini terbagi ke dalam tiga swimlane utama, yaitu pengguna atau pemain, game engine (mesin logika permainan Monopoli), dan dadu yang merepresentasikan logika acak dalam permainan.

Proses dimulai dari input jumlah pemain, di mana pengguna diminta untuk memasukkan jumlah pemain antara dua hingga empat orang. Jika input yang diberikan tidak sesuai, maka sistem akan meminta ulang hingga input valid diterima. Setelah itu, sistem melanjutkan ke tahap inisialisasi permainan, termasuk membuat objek pemain, membentuk papan permainan melalui fungsi buatPapan, serta memfasilitasi pemilihan karakter unik oleh masing-masing pemain. Setelah tahap ini selesai, permainan akan masuk ke menu utama, yang bersifat loop agar dapat diakses berulang kali.

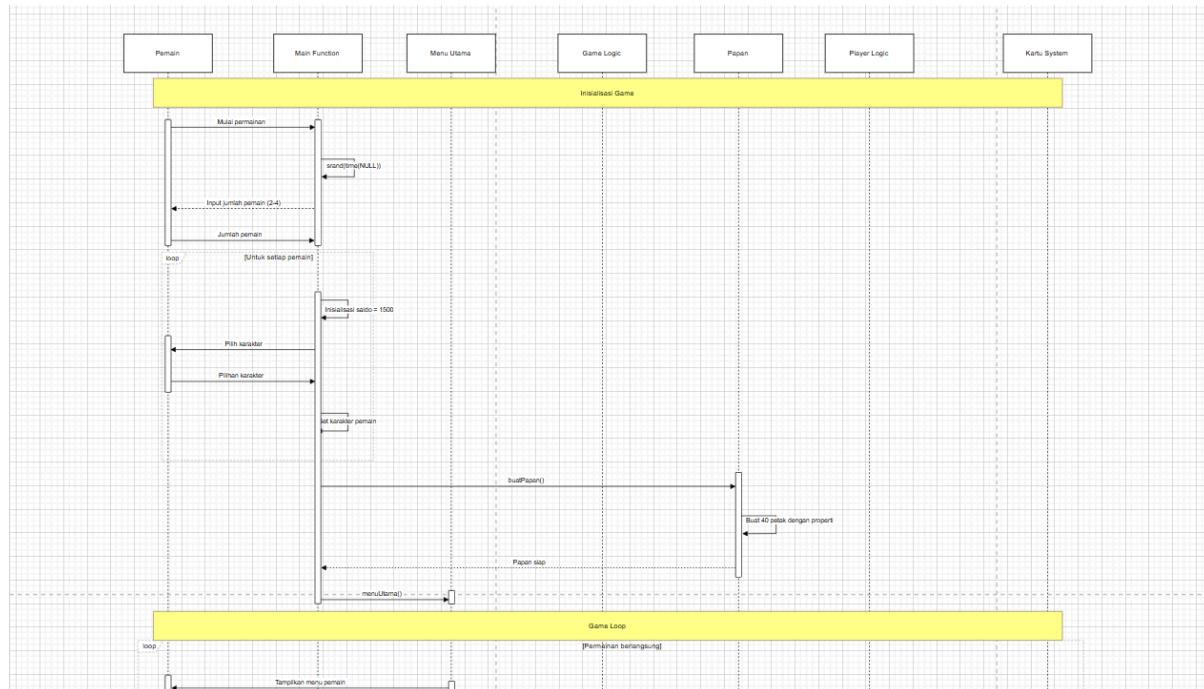
Pada menu utama, pemain diberikan beberapa opsi seperti memulai giliran, melihat kartu properti, melihat kondisi papan, atau keluar dari permainan. Ketika pemain memilih opsi tertentu, sistem akan terlebih dahulu memeriksa status keaktifan pemain—apakah pemain masih dalam kondisi aktif atau sudah bangkrut. Jika pemain masih aktif dan memilih untuk memulai giliran, maka permainan akan berlanjut ke proses lempar dadu.

Lemparan dadu merupakan bagian penting karena bersifat acak dan dapat menimbulkan berbagai efek. Jika pemain mendapatkan angka kembar (double), mereka diberi kesempatan untuk melempar dadu lagi. Namun, jika tiga kali berturut-turut mendapatkan angka kembar, pemain akan langsung dikirim ke penjara. Jika lemparan dadu tidak menunjukkan angka kembar, maka pemain akan bergerak sesuai hasil dadu dan sistem akan memproses efek dari petak yang ditempati.

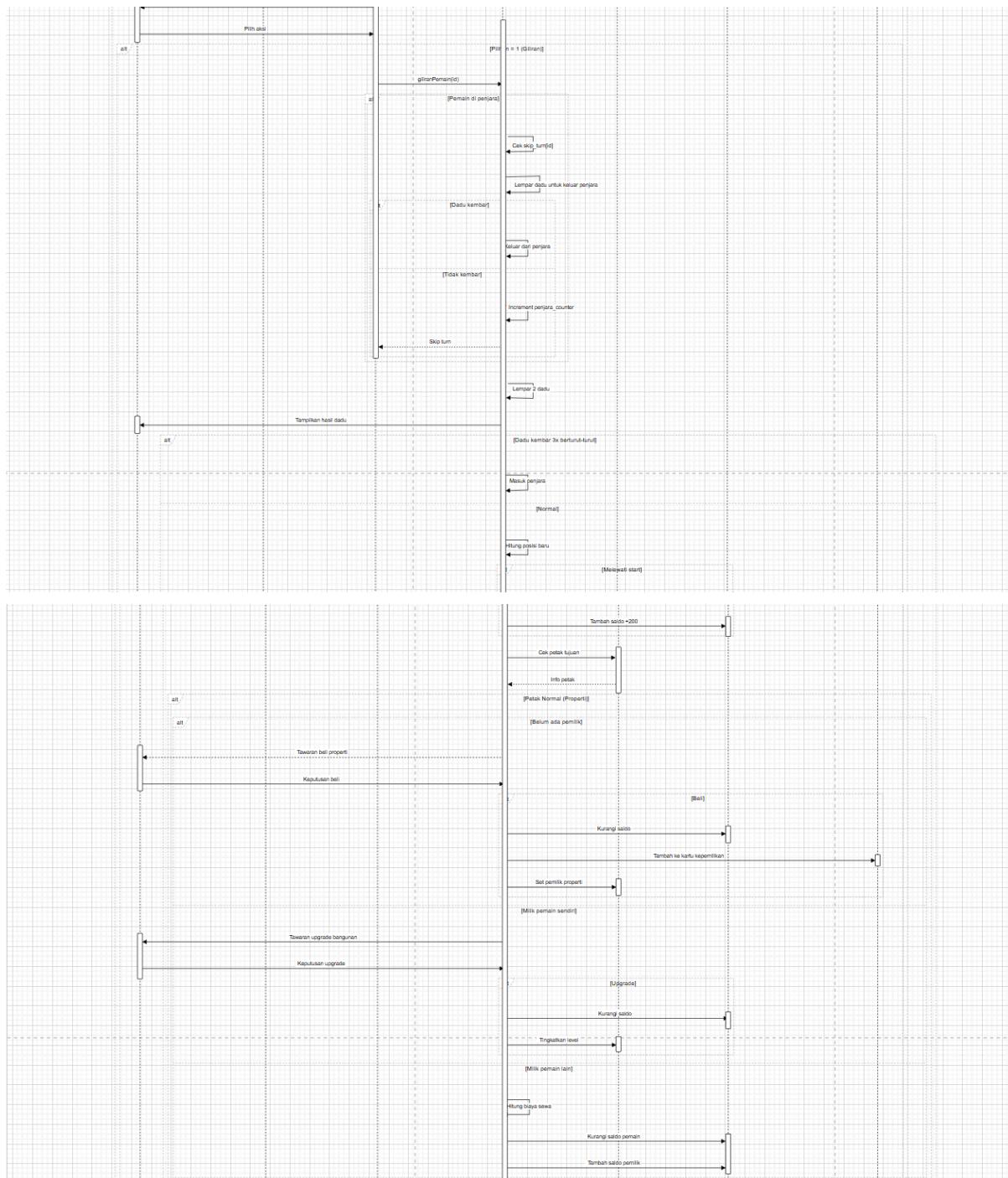
Efek petak bervariasi, tergantung jenisnya. Pemain bisa masuk ke penjara jika berhenti di petak penjara, membeli properti jika petak belum dimiliki dan saldo mencukupi, atau hanya melihat informasi properti jika tidak bisa dibeli. Beberapa petak juga memberikan kartu khusus seperti Dana Umum atau Kesempatan yang dapat berdampak positif atau negatif pada pemain. Setelah efek petak dijalankan, giliran akan beralih ke pemain berikutnya, atau tetap pada pemain yang mendapat hak lempar ulang.

Permainan akan terus berputar mengikuti alur ini hingga kondisi akhir tercapai, yaitu ketika hanya satu pemain yang tersisa dalam keadaan aktif. Pada titik ini, sistem akan menampilkan statistik akhir permainan yang mencakup informasi seperti saldo terakhir, jumlah properti yang dimiliki, total level bangunan, dan tentunya siapa yang keluar sebagai pemenang. Dengan adanya activity diagram ini, alur permainan Monopoli versi digital dapat dianalisis dan dipahami dengan lebih sistematis.

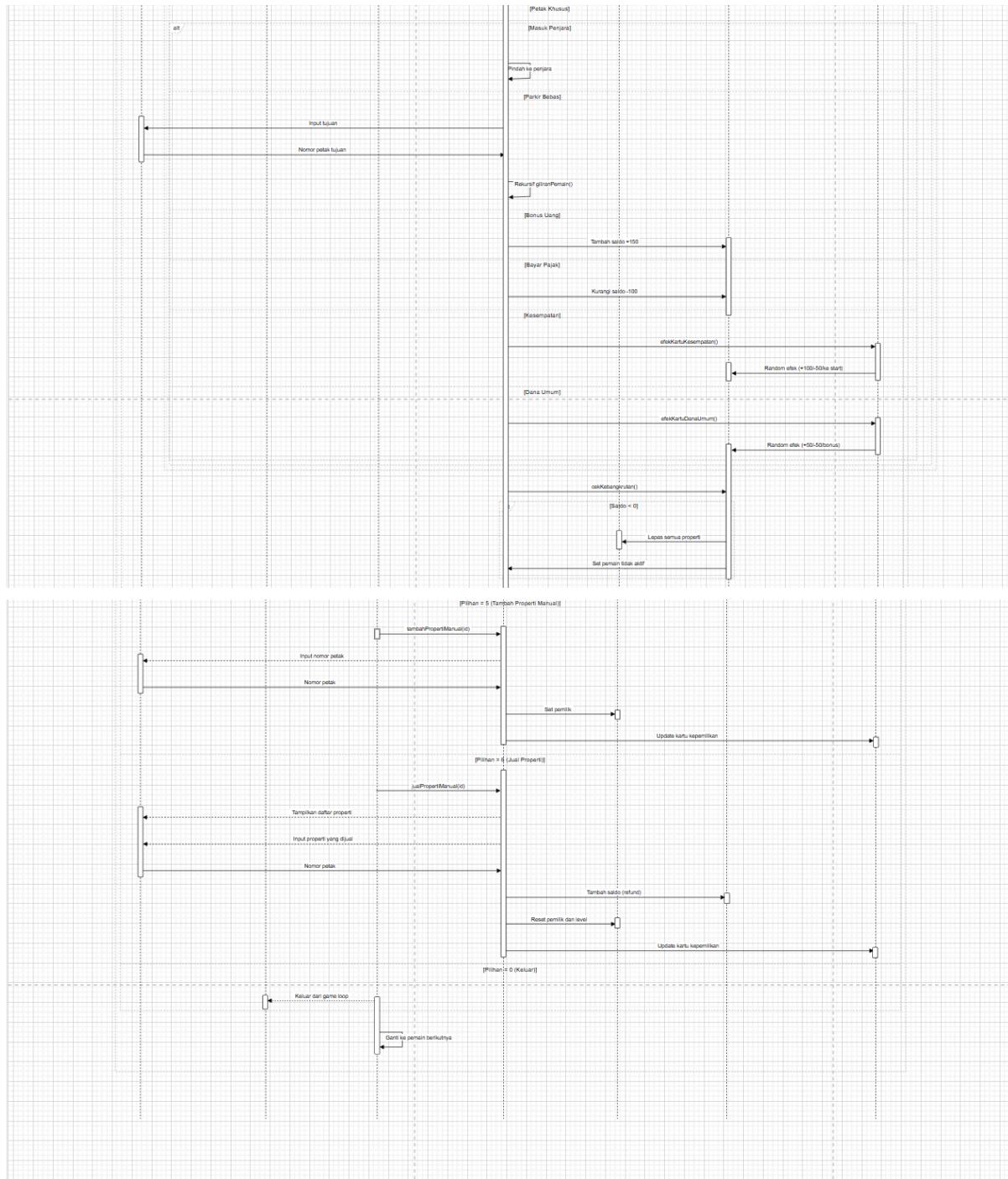
4.5 Sequence Diagram



APLIKASI SIMULASI GAME MONOPOLI MENGGUNAKAN BAHASA PEMROGRAMAN C++



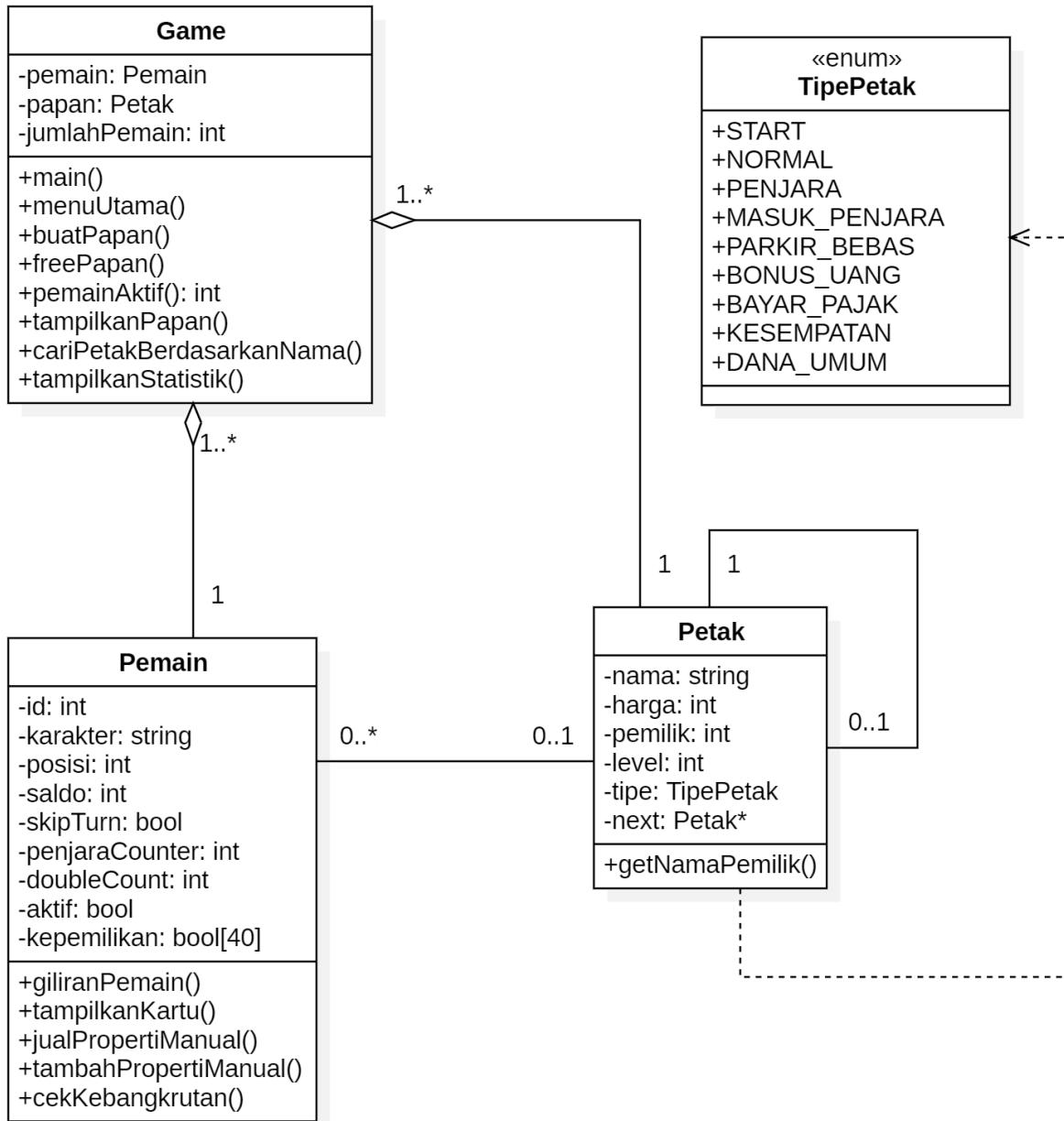
APLIKASI SIMULASI GAME MONOPOLI MENGGUNAKAN BAHASA PEMROGRAMAN C++



Gambar. 4. Sequence Program Simulasi Game Monopoli

Sequence diagram ini menggambarkan alur permainan Monopoly yang dimulai dengan fase inisialisasi (setup random seed, input jumlah pemain, pemilihan karakter, dan pembuatan papan 40 petak), dilanjutkan dengan game loop utama yang menyediakan 7 pilihan menu interaktif. Inti permainan terletak pada "Giliran Pemain" yang memiliki alur kompleks mulai dari pengecekan status penjara, pelemparan dadu dengan deteksi dadu kembar, perpindahan posisi dengan bonus melewati start, hingga interaksi dengan berbagai jenis petak (properti normal untuk jual-beli dan upgrade, serta petak khusus seperti penjara, parkir bebas, pajak, dan kartu acak). Sistem juga dilengkapi fitur manajemen seperti melihat kartu kepemilikan, pencarian petak, dan jual-beli properti manual, dengan mekanisme pengecekan kebangkrutan dan penentuan pemenang yang berjalan otomatis hingga hanya tersisa satu pemain aktif, menciptakan pengalaman bermain Monopoly yang lengkap dan interaktif dalam lingkungan console-based.

4.6 Class Diagram



Gambar. 5. Class Diargam Program Simulasi Game Monopoli

Class diagram pada proyek simulasi permainan Monopoly ini dirancang untuk merepresentasikan struktur program secara terorganisir melalui empat kelas utama, yaitu Game, Pemain, Petak, dan TipePetak (sebuah enum). Diagram ini menjadi fondasi penting dalam mendesain sistem berbasis pemrograman berorientasi objek (OOP), serta menunjukkan hubungan antarkomponen secara jelas.

Kelas **Game** berperan sebagai pengendali utama (controller) dari keseluruhan sistem permainan. Di dalamnya, terdapat atribut seperti papan, yang merepresentasikan array dari 40 petak permainan, serta daftar pemain yang terdiri dari 2 hingga 4 orang. Kelas ini juga dilengkapi dengan berbagai metode penting, seperti `main()` untuk menjalankan permainan, `menuUtama()` untuk menangani interaksi pemain, hingga `tampilkanStatistik()` yang menampilkan hasil akhir. Game memiliki hubungan *aggregation* dengan kelas Petak dan Pemain, mencerminkan bahwa Game mengelola komponen-komponen tersebut tanpa benar-benar menjadi pemilik mutlak secara memori.

Kelas **Pemain** menyimpan informasi dan status permainan masing-masing individu, seperti ID, karakter, posisi di papan, saldo, serta indikator seperti skipTurn, penjaraCounter, dan doubleCount. Setiap pemain juga memiliki array kepemilikan yang menunjukkan properti apa saja yang dimiliki. Metode dalam kelas ini mencakup giliranPemain() yang mengatur alur giliran, serta cekKebangkrutan() untuk mendeteksi jika saldo sudah negatif. Relasi antara Pemain dan Petak bersifat *association*, karena pemain dapat memiliki sejumlah properti namun tidak mengelolanya secara struktural.

Sementara itu, kelas **Petak** merepresentasikan setiap posisi di papan permainan. Atributnya meliputi nama, harga, pemilik, dan level, serta tipe yang menggunakan enum TipePetak. Salah satu atribut penting lainnya adalah next, yang memungkinkan setiap petak terhubung ke petak berikutnya dan membentuk struktur *linked list* circular. Petak memiliki relasi *dependency* dengan enum TipePetak, karena setiap petak memiliki satu jenis tipe tertentu yang menentukan logika saat pemain mendarat di sana.

Terakhir, kelas **TipePetak** berbentuk enum dan mendefinisikan berbagai jenis petak yang ada dalam permainan, seperti START, NORMAL, PENJARA, BONUS_UANG, hingga DANA_UMUM. Enum ini memudahkan sistem dalam membedakan perlakuan logika terhadap tiap-tiap petak.

Secara keseluruhan, class diagram ini menggambarkan hubungan antarobjek secara rinci, seperti Game yang memiliki 40 petak (*aggregation* 1 → 40), Game dengan 2–4 pemain, Petak yang saling terhubung membentuk papan melalui *linked list*, serta pemain yang dapat memiliki banyak petak. Pemanfaatan konsep OOP yang baik dalam rancangan ini memberikan pondasi kuat dalam membangun sistem permainan Monopoly berbasis teks yang interaktif dan modular.

4.7 Use Case Diagram

SISTEM PERMAINAN MONOPOLY



Gambar. 6. Use Case Diagram Program Simulasi Game Monopoli

Use case diagram dalam proyek permainan Monopoly ini merepresentasikan bagaimana pemain sebagai aktor utama berinteraksi dengan sistem. Pemain memiliki akses terhadap sejumlah fungsi inti dalam game, mulai dari memulai permainan hingga mengelola properti yang dimiliki. Ketika pemain memilih untuk **memulai permainan**, sistem akan secara otomatis menuntun mereka untuk **memasukkan jumlah pemain** dan **memilih karakter**, yang merupakan bagian dari proses awal permainan dan tergolong sebagai *use case include*, karena wajib dilakukan.

Setelah tahap inisialisasi selesai, permainan masuk ke dalam siklus giliran. Setiap kali **melakukan giliran**, pemain harus menjalani dua proses utama yang selalu terjadi, yaitu *melempar dadu* dan *berpindah petak*. Kedua aktivitas ini juga dikategorikan sebagai *use case include*. Berdasarkan petak yang didarati, pemain bisa mengalami berbagai kondisi yang memperluas proses giliran (didefinisikan sebagai *use case extends*). Misalnya, jika pemain mendarat di petak hukuman, maka sistem secara otomatis akan menjalankan skenario *masuk penjara*. Jika mendarat di petak hadiah, pemain akan menerima *bonus uang*. Sementara di petak pajak, pemain akan dikenakan biaya dalam bentuk *bayar pajak*.

Selain itu, pemain bisa **membeli properti** jika mendarat di petak yang belum dimiliki siapa pun, atau harus **membayar sewa** apabila properti tersebut dimiliki oleh pemain lain. Jika yang didarati adalah milik sendiri, maka akan muncul opsi **upgrade properti** ke level yang lebih tinggi. Petak istimewa seperti *Parkir Bebas* juga

dapat memicu aksi tambahan berupa *teleportasi*, yang tergolong *use case extends* karena hanya terjadi pada kondisi tertentu.

Selama giliran berlangsung, sistem juga dapat memberikan **efek kartu Kesempatan atau Dana Umum**, yang merupakan bagian dari proses giliran itu sendiri dan termasuk dalam kategori *include*. Di luar giliran, pemain memiliki kebebasan untuk **melihat informasi permainan**, baik berupa *status papan*, *kartu kepemilikan*, maupun *statistik akhir*. Informasi ini dapat membantu dalam pengambilan keputusan strategis. Pemain juga diberi keleluasaan untuk **menjual properti** kapan saja jika memerlukan tambahan dana, serta bisa memilih untuk **keluar permainan** apabila ingin mengakhiri partisipasi lebih awal.

Dengan demikian, use case diagram ini tidak hanya menggambarkan fungsi-fungsi utama yang tersedia bagi pemain, tetapi juga menunjukkan bagaimana logika permainan bekerja dalam mendukung jalannya simulasi secara dinamis dan interaktif.

4.8 Struktur Program

1. Pemilihan Karakter

Pada tahap awal permainan, pemain diminta untuk memasukkan jumlah pemain yang akan bermain (antara 2 hingga 4 orang). Setelah itu, setiap pemain secara bergiliran memilih karakter unik dari daftar yang tersedia, yaitu Mobil, Topi, Anjing, dan Kapal. Sistem memastikan bahwa tidak ada dua pemain yang memilih karakter yang sama. Pemilihan dilakukan dengan memasukkan angka sesuai daftar yang ditampilkan. Jika pilihan tidak valid atau karakter sudah dipilih pemain lain, maka sistem akan meminta input ulang hingga valid. Tahap ini penting untuk identifikasi pemain selama permainan berlangsung.

```
Jumlah pemain (2-4): 2
Pemain 1 pilih karakter:
1. Mobil
2. Topi
3. Anjing
4. Kapal
```

2. Fitur Fitur User

Disini user bisa melihat tampilan antarmuka utama dari permainan Monopoli berbasis teks saat giliran seorang pemain sedang berlangsung. Dalam contoh ini, giliran jatuh kepada Pemain 1 yang menggunakan karakter Mobil, dan memiliki saldo awal sebesar 1500. Pada layar, pemain diberikan beberapa pilihan menu yang dapat dipilih sesuai kebutuhan. Menu tersebut mencakup: Giliran (untuk melempar dadu dan melanjutkan permainan), Lihat Kartu (melihat daftar properti yang dimiliki), Lihat Papan (melihat kondisi papan permainan secara keseluruhan), Cari Petak (mencari informasi petak tertentu), Jual Properti (menjual properti yang dimiliki), dan Keluar (untuk mengakhiri permainan). Pemain cukup memasukkan nomor sesuai pilihan yang diinginkan untuk menjalankan aksi tertentu. Tampilan ini merupakan inti dari sistem interaksi pemain dengan permainan dan memberikan kontrol penuh terhadap keputusan yang ingin diambil selama giliran berlangsung.

```
--- Pemain 1 (Mobil) ---
=fa] Saldo: 1500
1. Giliran
2. Lihat Kartu
3. Lihat Papan
4. Cari Petak
5. Jual Properti
0. Keluar
Pilih: |
```

3. Landmark

Ketika pemain mendarat di petak kosong (belum dimiliki siapa pun), sistem akan memberikan pilihan kepada pemain untuk membeli petak tersebut. Informasi nama kota dan harga akan ditampilkan, dan pemain dapat memilih 1 untuk membeli atau 0 untuk melewati. Jika pemain memilih untuk membeli, saldo akan otomatis dikurangi sesuai harga, dan petak akan tercatat sebagai milik pemain tersebut. Proses ini merupakan inti dari sistem investasi dan strategi dalam permainan.

```
Pilih: 1
=fA] 4 & 1 = 5
=fôi Mendarat di Surabaya
Beli Surabaya seharga 150? 1=Ya, 0=Tidak: 1
=fâá Surabaya dibeli oleh Pemain 1.
=fE] Saldo: 1350

=fA] 6 & 5 = 11
=fôi Mendarat di Balikpapan
Beli Balikpapan seharga 270? 1=Ya, 0=Tidak: 0
=fâr Balikpapan tidak dibeli.
=fE] Saldo: 1450
```

4. Petak Kesempatan

Petak Kesempatan berfungsi seperti sistem *gacha* dalam permainan, di mana efek yang diterima pemain bersifat acak. Saat pemain mendarat di petak ini, sistem akan secara otomatis memberikan efek dari Kartu Kesempatan, yang bisa berupa hadiah uang, pengurangan saldo, atau perpindahan posisi. Pada contoh di gambar, Pemain 2 mengalami efek negatif dan kehilangan 50 saldo. Elemen ini menambah unsur kejutan dan strategi dalam permainan.

```
Pilih: 1
=fA] 3 & 3 = 6
=fôi Mendarat di Kesempatan
=fE] Pemain 2 kehilangan 50 dari Kartu Kesempatan!
=fE] Saldo: 1450
```

5. Parkir Bebas

Petak Parkir Bebas memberikan efek unik berupa kemampuan **teleportasi**. Saat pemain mendarat di petak ini, sistem akan meminta pemain untuk memasukkan nomor petak tujuan (0–39). Pemain kemudian akan langsung berpindah ke petak tersebut. Fitur ini dapat dimanfaatkan secara strategis, misalnya untuk menghindari pembayaran sewa atau mendekati properti yang ingin dibeli.

```
Pilih: 1
⇒ 2 & 5 = 7
⇒ Mendarat di Parkir Bebas
⇒ Kamu berada di Parkir Bebas. Masukkan nomor petak tujuan (0-39): 3
⇒ Teleport ke Bandung!
```

6. Upgrade Properti

Jika pemain mendarat di properti milik sendiri, sistem akan memberikan opsi untuk meng-upgrade properti tersebut ke level berikutnya. Upgrade ini akan meningkatkan nilai sewa properti dan membutuhkan biaya sesuai level. Dalam contoh, pemain memilih untuk meng-upgrade properti "Semarang" ke level 1 dengan biaya 150. Setelah upgrade berhasil, saldo pemain otomatis dikurangi. Fitur ini penting sebagai strategi investasi untuk meningkatkan pendapatan sewa.

```
Pilih: 1
⇒ 4 & 1 = 5
⇒ Mendarat di Semarang
⇒ Kamu berada di properti sendiri (Semarang). Ingin upgrade ke level 1 seharga 150? (1=Ya, 0=Tidak): 1
⇒ Upgrade berhasil ke level 1!
⇒ Saldo: 920
```

7. Denda Sewa

Jika pemain mendarat di properti milik lawan, maka sistem akan secara otomatis mengenakan **biaya sewa** yang harus dibayarkan kepada pemilik properti. Besar sewa ditentukan oleh harga dasar properti dan level bangunannya. Dalam contoh, pemain yang mendarat di Pontianak dikenakan sewa sebesar 160 dan saldo mereka langsung dikurangi. Mekanisme ini menjadi sumber pendapatan pasif bagi pemilik properti dan mendorong pemain lain untuk berhati-hati dalam pergerakannya.

```
Pilih: 1
⇒ 6 & 2 = 8
⇒ Mendarat di Pontianak
⇒ Bayar sewa 160 ke Pemain 1.
⇒ Saldo: 210
```

8. Bayar Pajak

Jika pemain mendarat di petak pajak, maka sistem akan secara otomatis memotong saldo pemain sejumlah nilai pajak yang ditentukan (contohnya 100). Petak ini berfungsi sebagai penalti yang mengurangi uang pemain dan menjadi bagian dari tantangan permainan.

```
⇒ Teleport ke Bayar Pajak!
⇒ Pemain 2 membayar pajak sebesar 100!
```

9. Dana Umum

Ketika pemain mendarat di petak Dana Umum, mereka akan menerima efek acak yang bersifat positif, seperti mendapatkan uang tambahan. Efek ini meningkatkan saldo pemain dan memberikan keuntungan tak terduga selama permainan berlangsung.

```
Pilih: 1
⇒ 4 & 3 = 7
⇒ Mendarat di Dana Umum
⇒ Pemain 1 mendapatkan uang dari Dana Umum!
⇒ Saldo: 260
```

10. Cek Properti

Fitur ini memungkinkan pemain untuk melihat daftar properti yang mereka miliki, lengkap dengan informasi nama kota dan level bangunan dari masing-masing properti. Dengan begitu, pemain dapat memantau aset yang dimiliki dan merencanakan strategi pengembangan atau penjualan. Fitur ini sangat berguna untuk evaluasi selama permainan.

```
Pilih: 2

≡fōç Kartu Properti Pemain 2 (Kapal):
- Padang (Level: 1)
- Bogor (Level: 0)
- Manado (Level: 0)
- Surakarta (Level: 0)
```

11. Lihat Papan

Pemain bisa melihat petak-petak yang sudah ada dan menampilkan daftar lengkap seluruh petak yang terdapat dalam papan permainan Monopoli versi teks. Setiap baris mewakili sebuah petak dengan informasi yang meliputi nomor petak, nama lokasi atau jenis petak, harga beli (jika tersedia), serta status kepemilikan. Terdapat total 40 petak, dimulai dari petak 0 (Mulai) hingga petak 39 (Jayapura). Beberapa petak bersifat spesial seperti “Dana Umum”, “Kesempatan”, “Bayar Pajak”, “Penjara”, dan “Parkir Bebas” yang tidak memiliki harga dan tidak dapat dimiliki oleh pemain.

Petak-petak kota seperti “Jakarta”, “Bandung”, “Yogyakarta”, hingga “Jayapura” memiliki harga tertentu dan bisa dibeli oleh pemain. Beberapa petak sudah dimiliki oleh pemain, seperti “Surabaya” oleh P1, “Palembang” oleh P1, dan “Padang” oleh P2, sedangkan sisanya masih berstatus “Belum ada” pemilik. Tampilan ini memungkinkan pemain untuk mengecek secara menyeluruh kondisi papan permainan, memantau harga properti, serta mengetahui properti mana saja yang telah dimiliki oleh masing-masing pemain. Informasi ini sangat berguna dalam strategi permainan, terutama saat mempertimbangkan pembelian atau penjualan properti.

Pilih: 3		
≡fō ^a Daftar Petak:		
Petak 0: Mulai	Harga: 0	Pemilik: Belum ada
Petak 1: Jakarta	Harga: 110	Pemilik: Belum ada
Petak 2: Dana Umum	Harga: 0	Pemilik: Belum ada
Petak 3: Bandung	Harga: 130	Pemilik: P1
Petak 4: Bayar Pajak	Harga: 0	Pemilik: Belum ada
Petak 5: Surabaya	Harga: 150	Pemilik: P1
Petak 6: Kesempatan	Harga: 0	Pemilik: Belum ada
Petak 7: Yogyakarta	Harga: 170	Pemilik: P1
Petak 8: Semarang	Harga: 180	Pemilik: P1
Petak 9: Penjara	Harga: 0	Pemilik: Belum ada
Petak 10: Medan	Harga: 200	Pemilik: Belum ada
Petak 11: Makassar	Harga: 210	Pemilik: P1
Petak 12: Dana Umum	Harga: 0	Pemilik: Belum ada
Petak 13: Malang	Harga: 230	Pemilik: Belum ada
Petak 14: Padang	Harga: 240	Pemilik: P2
Petak 15: Parkir Bebas	Harga: 0	Pemilik: Belum ada
Petak 16: Palembang	Harga: 260	Pemilik: P1
Petak 17: Balikpapan	Harga: 270	Pemilik: Belum ada
Petak 18: Kesempatan	Harga: 0	Pemilik: Belum ada
Petak 19: Banjarmasin	Harga: 290	Pemilik: Belum ada
Petak 20: Masuk Penjara	Harga: 0	Pemilik: Belum ada
Petak 21: Bogor	Harga: 310	Pemilik: P2
Petak 22: Pontianak	Harga: 320	Pemilik: P1
Petak 23: Batam	Harga: 330	Pemilik: Belum ada
Petak 24: Dana Umum	Harga: 0	Pemilik: Belum ada
Petak 25: Cirebon	Harga: 350	Pemilik: Belum ada
Petak 26: Manado	Harga: 360	Pemilik: P2
Petak 27: Semarang Barat	Harga: 370	Pemilik: Belum ada
Petak 28: Kesempatan	Harga: 0	Pemilik: Belum ada
Petak 29: Lampung	Harga: 390	Pemilik: Belum ada
Petak 30: Bonus Uang	Harga: 0	Pemilik: Belum ada
Petak 31: Depok	Harga: 410	Pemilik: Belum ada
Petak 32: Bekasi	Harga: 420	Pemilik: Belum ada
Petak 33: Kota Tua	Harga: 430	Pemilik: Belum ada
Petak 34: Dana Umum	Harga: 0	Pemilik: Belum ada
Petak 35: Bali	Harga: 450	Pemilik: Belum ada
Petak 36: Samarinda	Harga: 460	Pemilik: Belum ada
Petak 37: Surakarta	Harga: 470	Pemilik: P2
Petak 38: Kesempatan	Harga: 0	Pemilik: Belum ada
Petak 39: Jayapura	Harga: 490	Pemilik: Belum ada

12. Cari Petak

Fitur ini memungkinkan pemain untuk mencari informasi tentang petak tertentu berdasarkan nama kota. Setelah mengetik nama petak yang dicari, sistem akan menampilkan detail seperti nomor petak, harga, pemilik, dan level bangunannya. Fitur ini sangat membantu untuk perencanaan strategi, seperti menentukan target pembelian atau melihat status properti lawan.

Pilih: 4		
≡fō ^a Masukkan nama petak/kota yang dicari: bandung		
≡fō ^b Ditemukan:		
Petak 3: Bandung	Harga: 130	Pemilik: P1 Level: 1

13. Daftar Properti

Disini pemain bisa melihat fitur penjualan properti dalam permainan Monopoli versi teks. Dalam contoh ini, Pemain 2 (dengan karakter Kapal) memilih menu penjualan properti. Sistem kemudian menampilkan daftar properti yang dimiliki pemain tersebut, lengkap dengan nomor petak, nama kota, harga pembelian, dan level bangunan masing-masing properti.

Properti yang dimiliki Pemain 2 terdiri dari Padang (level 1), serta Bogor, Manado, dan Surakarta (semuanya masih pada level 0). Pemain kemudian memasukkan nomor petak 21, yang merupakan kota Bogor, untuk dijual. Setelah itu, sistem mengonfirmasi bahwa properti Bogor berhasil dijual, dan pemain menerima uang sebesar 310 sebagai hasil penjualan. Fitur ini memungkinkan pemain untuk mengelola aset mereka secara strategis, baik untuk mendapatkan dana tambahan maupun untuk menyusun ulang portofolio properti selama permainan berlangsung.

```
Pilih: 5
⇒ Daftar Properti yang Dimiliki Pemain 2 (Kapal):
14. Padang (Harga: 240 | Level: 1)
21. Bogor (Harga: 310 | Level: 0)
26. Manado (Harga: 360 | Level: 0)
37. Surakarta (Harga: 470 | Level: 0)
Masukkan nomor petak yang ingin dijual: 21
⇒ Properti Bogor berhasil dijual. Uang kembali: 310
```

5. Penutup

Pengembangan aplikasi simulasi permainan Monopoli dengan bahasa pemrograman C++ ini membuktikan keberhasilan dalam mengimplementasikan berbagai konsep penting di bidang pemrograman dan struktur data. Dengan menerapkan paradigma pemrograman berorientasi objek (OOP) dan memanfaatkan struktur data seperti circular linked list, queue, hash table, dan enum, permainan ini mampu berjalan interaktif sesuai aturan Monopoli klasik. Fitur utama yang telah direalisasikan mencakup sistem giliran, pembelian dan pengelolaan properti, kartu acak (Kesempatan dan Dana Umum), serta mekanisme penjara dan kebangkrutan. Desain papan berbentuk sirkular yang dibangun menggunakan circular linked list sangat efektif dalam merepresentasikan alur permainan yang berputar.

Proyek ini turut memberikan pengalaman berharga bagi pengembang, khususnya dalam menggabungkan struktur data, logika algoritmik, dan OOP dalam sebuah sistem permainan yang utuh. Penggunaan tampilan antarmuka berbasis teks (CLI) memudahkan proses uji coba sekaligus menjadi sarana pembelajaran yang praktis bagi pemula dalam pengembangan game berbasis konsol. Di samping itu, sistem juga menyertakan fitur pencatatan transaksi, giliran pemain, dan rekap statistik akhir permainan yang berguna untuk evaluasi maupun pengembangan lanjutan.

Keberhasilan proyek ini tidak hanya memenuhi aspek akademik dalam rangka UAS, tetapi juga berfungsi sebagai prototipe permainan sederhana yang berpotensi dikembangkan lebih jauh. Beberapa pengembangan yang disarankan ke depannya mencakup penggunaan antarmuka grafis (GUI) dengan library seperti Qt, SDL, atau SFML agar lebih menarik secara visual dan user-friendly. Pengembangan kecerdasan buatan (AI) untuk pemain non-manusia juga menjadi tantangan yang menarik dan memberikan peluang eksplorasi algoritma lebih dalam.

Pengembangan lainnya yang dapat dipertimbangkan antara lain penambahan fitur permainan daring agar pemain bisa bermain dari perangkat berbeda secara real-time, serta peningkatan efisiensi memori dan optimalisasi struktur data untuk mencegah kebocoran memori. Penerapan unit testing serta dokumentasi otomatis akan membantu dalam menjaga kualitas dan keberlanjutan proyek. Fitur untuk menyimpan dan melanjutkan permainan juga menjadi nilai tambah yang signifikan. Dengan pengembangan berkesinambungan dan evaluasi berkala, proyek simulasi Monopoli ini memiliki potensi untuk menjadi media edukatif atau hiburan yang lebih canggih dan menarik.

Referensi

- [1] DDeitel, H. M., & Deitel, P. J. (2016). *C++ how to program* (10th ed.). Pearson Education.
- [2] Malik, D. S. (2010). *C++ programming: From problem analysis to program design* (5th ed.). Cengage Learning.
- [3] Weiss, M. A. (2014). *Data structures and algorithm analysis in C++* (4th ed.). Pearson.
- [4] Wahid, A. (2022). *Struktur data dan algoritma menggunakan C++*. Andi Publisher.
- [5] Raymond, E. S. (2003). *The art of Unix programming*. Addison-Wesley
- [6] Hasbro. (2023). *Monopoly official rules*. <https://www.hasbro.com/common/instruct/monopoly.pdf>
- [7] GeeksforGeeks. (n.d.). *Data structures in C++*. Retrieved June 2025, from <https://www.geeksforgeeks.org/data-structures/>
- [8] TutorialsPoint. (n.d.). *C++ programming language*. Retrieved June 2025, from <https://www.tutorialspoint.com/cplusplus/index.htm>
- [9] cppreference.com. (n.d.). *C++ reference*. Retrieved June 2025, from <https://en.cppreference.com/>
- [10] GameDev.net. (n.d.). *Creating a board game simulation in C++*. Retrieved June 2025, from <https://www.gamedev.net/>