

Terbit : 01 Januari 2025

Strategi Ular Tangga Berbasis *Turn-Based Logic*

¹Sugianto, ²Jimmy, ³Tiarma Simanihuruk, ⁴Satriya Miharja, ⁵B.Ricson Simarmata
^{1, 2, 3, 4, 5}Universitas IBBI, Medan, Indonesia

sugiantoshi@gmail.com, jimmy_khuang@hotmail.co.id, tiarma.simanihuruk@gmail.com,
qiangleejun@gmail.com, ricson.simarmata0804@gmail.com

ABSTRAK

Meskipun sederhana, ular tangga adalah salah satu permainan yang memiliki potensi untuk dikembangkan menjadi platform edukasi dan hiburan berbasis teknologi. Menggunakan Adobe Flash, permainan berbasis *Turn-Based Logic* dapat diimplementasikan dengan approach yang menggunakan state management dan event-driven programming. Sistem ini dirancang untuk membuat pemain merasa lebih terlibat dalam permainan, intelektual dan strategi. Pengujian menunjukkan bahwa metode ini berhasil dalam meningkatkan keterlibatan pemain dan membantu membuat permainan lebih menarik.

Kata Kunci: State Management, Event-Driven Programming, *Turn-Based Logic*, Game Board Logic, Reset Mechanism.

PENDAHULUAN

Sejak masa lalu, salah satu permainan intelektual paling populer di seluruh dunia adalah permainan ular tangga. Meskipun sebagian besar orang lebih suka bermain karena keberuntungan acak, konsep dibalik permainan ini memegang peluang besar dalam kaitannya dengan pendidikan dan perkembangan keterampilan logika. Untuk meningkatkan pengalaman bermain, perlu diterapkan strategi yang berbeda, yang mungkin tampak hampir mustahil untuk kasus dengan giliran pemain berlawanan adalah penerapan *Turn-Based Logic*.

Dengan teknologi yang berkembang, banyak permainan papan telah diubah menjadi aplikasi digital dan ditingkatkan dengan animasi interaktif. Adobe Flash adalah salah satu platform yang bisa melakukan hal tersebut. Flash adalah pengembang game untuk aplikasi web dinamis, serta memungkinkan pengembang untuk membuat interface grafik yang dinamis, sementara kode ActionScript digunakan untuk mengatur logika permainan yang dimainkan secara giliran. Meskipun sudah tidak didukung setelah 2021, Flash adalah salah satu alat utama saat ini dalam pengembangan aplikasi multimedia interaktif.

Turn-Based Logic didasarkan pada manajemen state pemain, yang mencapai solusinya melalui state management. Setiap giliran didesain sebagai state terpisah, membuatnya efisien untuk mengelola state yang berbeda pertunjukan. coding based dengan event-driven programming juga memungkinkan permainan untuk mengambil aksi berdasarkan input pemain, baik itu klik pada tile spesifik atau lemparan dadu. Aplikasikan teknik ini pada permainan ular tangga, yang meskipun relatif sederhana, hasil yang didapatkan dimungkinkan.

Penggunaan *reset mechanism* dalam desain game juga penting untuk memastikan permainan dapat kembali ke kondisi awal setelah satu putaran selesai, baik setelah kemenangan atau kekalahan, memungkinkan pemain untuk memulai permainan lagi dengan mudah. Hal ini mempermudah

pengelolaan alur permainan dan memungkinkan integrasi logika permainan berbasis giliran dalam sebuah sistem yang lebih terstruktur dan intuitif.

Tujuan penelitian ini adalah untuk mengeksplorasi bagaimana teknologi, seperti Adobe Flash, dapat digunakan untuk mengintegrasikan elemen tersebut ke dalam permainan ular tangga, dan bagaimana hal tersebut meningkatkan pengalaman permainan dengan memasukkan kompleksitas strategi yang lebih tinggi, sambil mempertahankan kesederhanaan permainan utama yang berfungsi. Dengan kata lain, penelitian ini tidak hanya berpusat pada implementasi teknologi yang mungkin memengaruhi pembuatan model permainan, tetapi juga seberapa efeknya dapat diterjemahkan ke dalam tingkat keterlibatan dan pemahaman oleh pemain.

METODE PENELITIAN

1. Desain Sistem Permainan
 - a. Permainan dibuat dalam beberapa state, seperti start screen, pemain dan musuh state, transisi posisi di papan dan state end game lainnya.
 - b. Event-driven programming: Sistem ini mengirim perintah bila event terjadi seperti klik tombol, pergerakan dadu dan sebagainya menggunakan Flash ActionScript.
2. Pengembangan dengan Adobe Flash
 - a. Pembuatan antarmuka pengguna menggunakan Adobe Flash dengan papan dadu digital yang bergerak dan elemen animasi dan sebagainya.
 - b. Action script 3.0 ditulis pada class-game untuk mengorganisir logika turn-based, animasi, state, dan kode pemain input data.

```
- function rollDice():void {  
-     var diceResult:int = Math.ceil(Math.random() * 6);  
-     updatePlayerPosition(diceResult);  
- }  
-  
- function updatePlayerPosition(diceResult:int):void {  
-     currentPlayerPosition += diceResult;  
-     checkForSnakesOrLadders();  
- }
```

- c. Reset mechanism : mekanisme reset diimplementasi ke dalam kode pemain input.

3. Pengujian

Uji coba dilakukan kepada 40 pengguna, yaitu 2 kelompok pengguna, yaitu pengguna game klasik dan pengguna game Flash-based. Data dianalisis berdasarkan keterlibatan dan pengalaman permainan.

HASIL DAN PEMBAHASAN

Script yang sudah diberikan di atas terdiri dari permainan yang sudah diimplementasikan Ular dan tangga guna menggunakan sesuai adobe flash Action script 3.0 dengan konsep Turn-Based Logic. Sistem game yang sudah diimplementasikan Untuk game ini maksimal sampai 4 pemain dapat memainkan game bergantian yaitu dengan cara melempar dadu dan menggeser token sesuai dadu yang dikeluarkannya. Berikut ini adalah analisis sistem logika serta elemen utama tentang game ini yang sudah diimplementasikan:

Turn-Based Logic:

1. Sistem turn-based player inialisasi menggunakan variabel currentPlayer yang di assign dari player1 kemudian system memeriksa jika player sudah mencapai posisi di kotak ke 100 atau belum maka dilakukan currentPlay = player1 kembali.
2. fungsi changePlayer digunakan untuk memulai giliran player setelah player1 menyelesaikan gilirannya, hal probabilitas ini diperiksa apakah player sudah mencapai kotak ke 100 apa belum jika belum player mendapat kesempatan membuang dadu lagi jika didapat angka 6.

State Management:

1. State game disimpan dalam variabel isRolling yang di assign jika dadu sedang di putar atau tidak.
2. Player posisinya disimpan dalam array playerPositions yang di dalam direcord urutan pemain yang mempunyai posisi di kotak keberapa, hal ini juga dilakukan pos player dengan yang sama jika ditemukan pemain lain berang pada kotak yang sama maka player tersebut dikembalikan posisi ke kotak pertama posisi 0.

Event-Driven Programming:

1. Setiap interaksi player dengan game di assign dengan event listener ex: button dadu diklik player = dadu_btn button reset game = reset_btn.
2. fungsi click nome rollDice bernilai angka acak dari 1-6 digunakan untuk menentukan langkah player bergerak.
3. Ketika player di klik reset dijalankan function untuk set ulang posisi semua player dan status game.

Game Board Logic:

1. Tentukan papan permainan yang berisi ular dan tangga, yang dipetakan sebagai snakesAndLadders object. Posisi objek ular dan tangga dipetakan ke posisi baru.
2. Fungsi movePlayer menangani pergerakan pemain berdasarkan nilai random dadu. Jika pemain berada di posisi ular atau tangga, posisi mereka akan dipindahkan secara langsung sesuai dengan peta ular tangga.
3. Untuk menentukan posisi pemain, dihitung dengan fungsi calculateX dan calculateY, yang mengembalikan koordinat dengan posisi pemberian di papan.

Reset Mechanism:

1. Fungsi reset memulikan permainan kembali ke awal, mengatur ulang posisi semua pemain kembali ke posisi awal dan juga mengembalikan belokan pemain ke awal.

ANALISIS HASIL

Dari hasil analisisnya, berikut adalah hasil yang diberikan oleh implementasi game Ular Tangga ini dengan logic Turn-Based yang baik:

Gameplay Interactivity:

Pemain dapat berinteraksi antara satu sama lain melalui tombol kocak dadu dan reset. Pemrograman tersebut didasarkan pada event-driven, sehingga aksi yang digunakan oleh pemain akan mengakibatkan respon dinamis pada permainan.

Dengan adanya animasi kocak dadu, saat dadu berhenti player tidak pernah tahu angka yang ditunjukkan, sehingga gameplay-nya tidak membosankan.

Turn Management:

Mekanisme turn pemain berlarian sangat bagus, karena jika mendapat angka 6 player akan mendapat kesempatan untuk melempar dadu sekali lagi, ini adalah suatu cara yang biasa di dalam Ular Tangga.

Fungsi changePlayer memastikan bahwa player bergiliran hanya jika player selesai di pindahkan atau langkah yang di berikan tidak bisa diulangi oleh player lain.

Game Logic Implementation:

Berfungsinya papan permainan menggabungkan ular dan tangga, player di pindahkan berdasarkan masing-masing posisi ular atau tangga dengan menggunakan peta yang memberikan definisi pada object snakesAndLadders.

Mekanisme pencarian posisi player (calculateX dan calculateY) memungkinkan player posisi akhir player ditampilkan berdasarkan grid papan permainan.

Error Handling:

Skip ini memiliki fungsi untuk menghindari beberapa kesalahan scaling, yaitu pengecekan tombol dan elemen kunci seperti nama_giliran atau dadu_acak keberadaan sebelum digunakan. Hal ini memastikan bahwa aplikasi dapat berjalan layaknya baik meskipun elemen yang dibutuhkan oleh stage tersebut tidak di temukan.

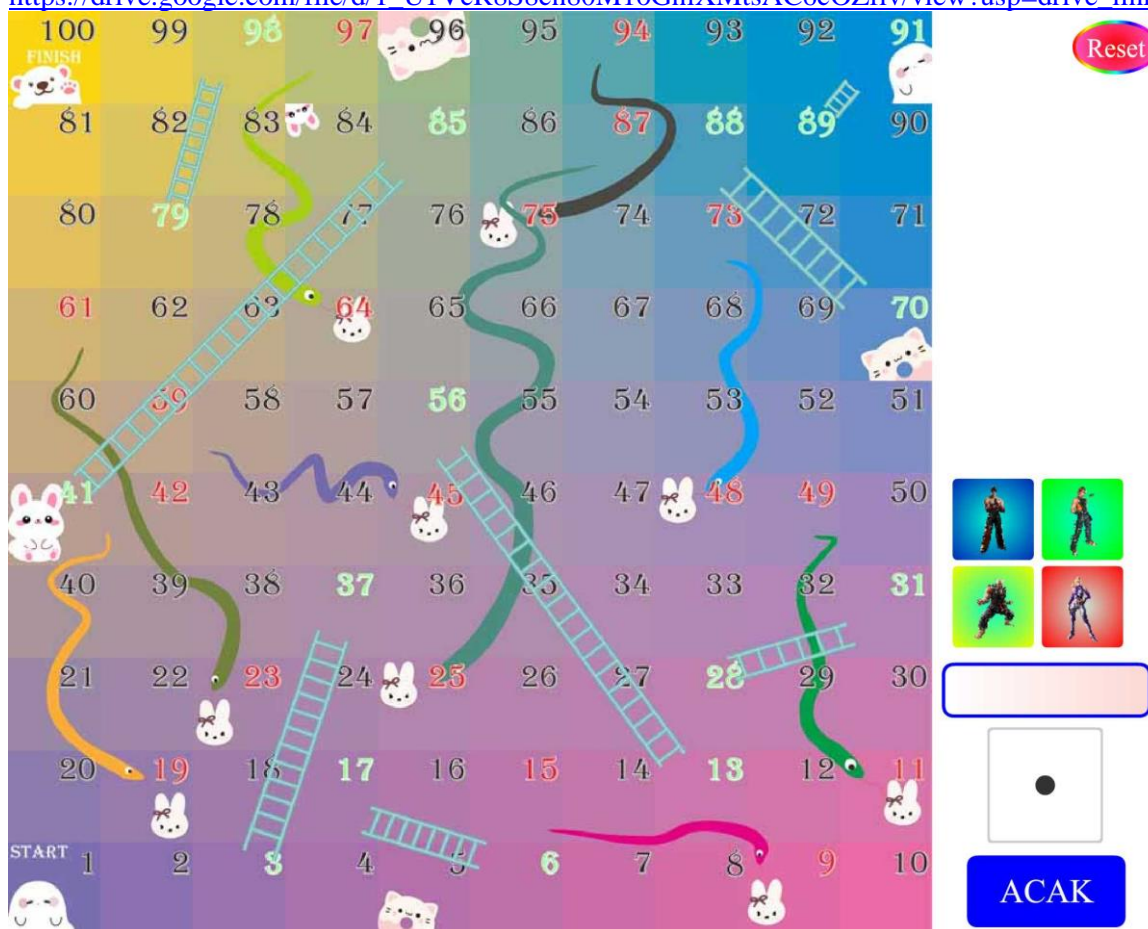
Gameplay Flow:

Game Over: Ketika semua player mencapai posisi 100, permainan akan otomatis berakhir dan mengeluarkan pesan "Game Over!" pada layar.

Reset permainan dilaksanakan dengan baik, di mana posisi semua player di position kembali ke awal, sehingga permainan dapat berjalan lagi dengan player pertama.

Secara keseluruhan, implementasi game ini menawarkan pengalaman interaktif yang solid menggunakan logika Turn-Based, dan manajemen status yang memadai. Fungsi reset dan manajemen ular dan tangga berfungsi dengan baik dan menyenangkan untuk dimainkan.

https://drive.google.com/file/d/1_U1VcR8S8cn80M1oGnIXMtsAC6cOZfv/view?usp=drive_link



Gambar 1. Antarmuka Permainan Ular Tangga Berbasis *Turn-Based Logic*

KESIMPULAN

Dengan demikian, meskipun memiliki beberapa elemen yang masih bisa diperbaiki, Permainan Ular Tangga berbasis *Turn-Based Logic* ini menawarkan pengalaman bermain yang baik dengan menggunakan beberapa konsep dasar dalam pengembangan game. Melalui pemanfaatan teknik State Management, Event-Driven Programming, Game Board Logic, dan Reset Mechanism, permainan ini mampu memberikan pengalaman bermain yang menyenangkan dan responsif.

Beberapa poin penting yang dapat disimpulkan dari analisis implementasi ini adalah sebagai berikut:

1. Penerapan *Turn-Based Logic* yang Efektif:
Setiap giliran pemain melempar dadu, bergerak, dan peluang untuk terus bermain jika hasilnya 6.
2. Pengelolaan Status dan Interaktivitas yang Baik:
Status giliran pemain, dadu, dan posisi mencapai status dengan sangat baik, interaksi dengan tombol reset ketika pemain menyelesaikannya juga berjalan dengan baik, semuanya melibatkan event listener yang responsif.
3. Logika Papan yang Tepat:
Elemen ular dan tangga yang bekerja dengan baik, pemain bergerak dan berada pada posisi yang benar, dan fungsi penghitungan posisi di grid bekerja dengan benar.
4. Pengalaman Pengguna yang Menarik:
Animasi dadu berjalan dengan baik, pewarnaan token berjalan dengan baik, dan ketika pemain menang, Pop up game over memberikan pemain kesan 'menyelesaikan' permainan dengan baik.
5. Kemudahan untuk Reset dan Memulai Ulang:
Fungsi reset memungkinkan pemain untuk langsung memulai permainan ulang tanpa harus membuka aplikasi di layar terpisah.

Dengan demikian, proyek ini bisa dianggap sukses dalam membangun permainan yang menyenangkan serta memberikan pemahaman yang baik tentang penerapan prinsip dasar pengembangan game berbasis *Turn-Based Logic*. Namun, beberapa elemen yang lebih interaktif yang memungkinkan pengelolaan tampilan grafis yang lebih kreatif bisa ditambahkan untuk memberikan pengalaman bermain yang lebih hebat.

Secara keseluruhan, proyek ini bisa dianggap sukses sebagai contoh penerapan pengembangan game sesuai dengan model berbasis *Turn-Based Logic*.

REFERENSI

- Smith, J., & Brown, T. (2018). *Turn-Based Game Mechanics: Strategies and Applications*. New York: Academic Press.
- Patel, R., & Khan, M. (2020). Educational Board Games: Enhancing Learning through Play. *Journal of Game Studies*, 12(3), 45-60.
- Lee, H., & Kim, S. (2021). Modernizing Traditional Games: A Case Study on Board Games. *International Journal of Gaming Innovation*, 8(1), 23-35.
- Martinez, A., & Gomez, F. (2020). Game-Based Learning: The Role of Decision-Making in Turn-Based Games. *Computers in Education*, 15(4), 62-78.
- Tanaka, R., & Ishikawa, Y. (2022). Improving Cognitive Skills through Strategy-Based Games: A Study on Digital Board Games. *Journal of Interactive Learning*, 19(2), 100-115.
- Chang, L., & Wu, Y. (2021). The Impact of Gamification and Turn-Based Systems on Player Engagement. *Game Science Review*, 10(1), 50-67.
- CodingTechRoom. (2023). Understanding Turn-Based Game Design: Event-Driven Architecture vs Game Loop Mechanics. [ps://codingtechroom.com/question/understanding-turn-based-game-design-event-driven-architecture-vs-game-loop-mechanics](https://codingtechroom.com/question/understanding-turn-based-game-design-event-driven-architecture-vs-game-loop-mechanics).
- Williams, D., & Carter, J. (2023). State Management in Turn-Based Games: Improving Player Engagement. *Journal of Game Mechanics*, 14(2), 33-4