Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Procedural Content Generation pada Level Gim Sokoban Menggunakan Model Hybrid GPT2 dan Algoritma Genetika

[1]Narandha Arya Ranggianto, [2]Akbar Pandu Segara, [3]Dwi Wijonarko, [4]Anang Andrianto, [5]M. Habibullah Arief
[1,2,3,4]Teknologi Informasi, Universitas Jember, Indonesia
[5]Sistem Informasi, Universitas Jember, Indonesia

[1]ranggi@unej.ac.id, [2]akbarpandu@unej.ac.id, [3]dwi-wijonarko@unej.ac.id, [4]anang.uptti@unej.ac.id, [5]m.habibullaharief@unej.ac.id

*Penulis Korespondensi
Diajukan : 07/08/2025
Diterima : 08/08/2025
Dipublikasi : 08/08/2025

ABSTRAK

Procedural Content Generation (PCG) yang berfokus pada level menjadi poin penting dalam mempengaruhi pengalaman pengguna dalam bermain gim. Salah satu gim puzzle khususnya Sokoban dapat diterapkan untuk pembangunan level secara otomatis karena dapat direpresentasikan secara sederhana. Dataset Sokoban biasanya direpresentasikan ke dalam string ASCII yang terdiri dari pemain (@), dinding (#), kotak ($), dan tujuan (.). Hal ini menjadikan level Sokoban dapat dikembangkan menggunakan dua pendekatan yaitu berbasis pencarian dan *machine learning*. Metode pencarian memiliki kelebihan dalam mengeksplorasi sebuah level yang *playable* namun menghasilkan level yang sama. Sedangkan pada pendekatan *machine learning* data digunakan untuk melakukan *training* dengan pola-pola tertentu sehingga memberikan kemampuan membangun level yang bervariatif. Kekurangan data dalam level gim menjadikan pendekatan *fine-tuning* GPT2 lebih unggul untuk digunakan dalam pembangunan level. Namun, karakteristik data yang tidak memiliki koherensi yang baik pada level Sokoban menjadikan GPT2 tidak dapat membangun level yang *playable*. Model Hybrid GPT2 dan Algoritma Genetika (GPT2-GA) dimana nilai penggabungan ini akan memberikan hasil yang optimal. Evaluasi untuk mengukur *accuracy, playability,* dan *diversity* yang menunjukkan performa lebih unggul dibandingkan GPT2. Model GPT2-GA menunjukkan hasil peningkatan *accuracy* dari 81,9% menjadi 90,1%, *playability* dari 41,3% menjadi 62,8%, dan *diversity* dari 88,2% menjadi 97,5%. Pendekatan model ini berhasil mengatasi kelemahan model generatif GPT2 dalam menghasilkan level yang fungsional dengan mempertahankan level yang unik yang dapat diselesaikan.

Kata Kunci: algoritma genetika, gpt2, *procedural content generation*, level gim, sokoban

I. PENDAHULUAN

Procedural Content Generation (PCG) adalah metode yang menggunakan algoritma untuk menciptakan konten gim secara otomatis, yang mencakup dunia gim yang luas dan beragam. Teknik ini memungkinkan pengembang untuk menghasilkan berbagai elemen permainan mulai dari karakter hingga lingkungan tanpa harus membuatnya secara manual satu per satu. PCG sendiri memiliki banyak jenis dari level, item, maupun karakter. Namun,` pada pengembangan level dapat meningkatkan nilai main ulang dan keragaman (Liapis, 2020). Pemanfaatan PCG dalam level diterapkan pada permainan Sokoban, sebuah permainan puzzle klasik berbasis petak (*tile-based*) dimana pemain harus mendorong kotak-kotak ke lokasi target yang telah ditentukan. Level dapat

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

direpresentasikan secara sederhana menggunakan karakter ASCII yang biasa dikembangkan dalam topik PCG.

PCG dalam Sokoban dilakukan dengan menggunaan pendekatan berbasis pencarian. Algoritma Genetika merupakan metode berbasis pencarian yang meniru proses evolusi biologis untuk menciptakan konten gim. Proses ini bekerja dengan cara memelihara sekumpulan "populasi" kandidat konten, seperti level atau item, dan secara berulang menyempurnakannya dari satu generasi ke generasi berikutnya (Farrokhi Maleki & Zhao, 2024). Penelitian lainnya menggunakan metode Monte Carlo Tree Search (MCTS) dengan menggunakan keadaan (*state*) konten dan setiap sisi *(edge)* melambangkan sebuah transisi atau aksi sehingga metode ini menjelajahi ruang kombinatorial yang direpresentasikan oleh struktur pohon yang cocok untuk gim RPG (Świechowski et al., 2023). Dalam pembuatan gim PixelArt biasanya menggunakan metode Wave Function Collapse (WFC) menggunakan sistem berbasis petak (*tile-based*) atau grid, di mana level atau dunianya dibangun dari potongan-potongan kecil yang mengikuti aturan penempatan tertentu (Wang & Chang, 2024). Namun pendekatan ini masih menghasilkan level yang sama dan fungsi evaluasi yang berbeda untuk tiap konten.

Pendekatan lainnya yaitu berbasis machine learning (PCGML) yang dapat melakukan representasi konten berdasarkan data baik dari gambar maupun teks. Pada penelitian lainnya model Generative Adversial Networks (GAN) dilakukan untuk membuat level pada Gim 2D dengan melatih data dari gim Zelda (Liu et al., 2021). GAN sendiri terdiri dari arsitektur generator yang bertugas menciptakan konten baru yang realistis, dan sebuah diskriminator yang dilatih untuk membedakan antara konten asli dari data pelatihan dengan konten buatan generator. Pembuatan level dapat menggunakan data teks dengan pendekatan machine translation yaitu konsep encoder dan decode. Konsep ini dilakukan oleh metode Transformers yang dapat mengubah sebuah input teks ke sebuah level dalam gim (Khan et al., 2022). Namun, kelemahan dari GAN dan Transformer terdapat pada representasi yang mengalami kesulitan dalam memahami fitur-fitur karena membutuhkan data yang sangat besar dalam melakukan model pelatihan (Todd et al., 2023). Oleh karena itu, penggunaan model yang telah dilatih menggunakan data besar lebih memberikan pemahaman konteks yang lebih baik.

Salah satu model yang telah dilatih beberapa data teks yaitu GPT yang memiliki arsitektur Transformer dengan mekanisme self-attention, mampu menangkap hubungan semantik yang rumit dan dependensi antara elemen-elemen yang berjauhan dalam sebuah urutan data dengan lebih baik. Penelitian sebelumnya menggunakan model GPT2 yang disebut dengan MarioGPT untuk membangun level pada platformer (Sudhakaran et al., 2023). Selain itu, penggunaan model GPT2 juga digunakan untuk membuat konten yang unik dalam dunia RPG. Namun, model GPT2 memiliki kelemahan dalam mengikuti *prompt* yang spesifik (van Stegeren & Myśliwiec, 2021). Hal ini juga terjadi ketika diimplementasikan pada gim Sokoban karena hasilnya tidak konsisten terhadap instruksi dan tidak dapat diandalkan agar semua level dapat dimainkan.

Oleh karena itu, pada penelitian ini pembangunan level tidak hanya mementingkan pembangunan level yang variatif menggunakan model GPT2, namun memastikan level dapat dimainkan. Penggunaan metode Algoritma Genetika membantu dalam membangun level berdasarkan fungsi *fitness* salah satunya nilai *solvability* dari populasi level yang dihasilkan oleh model GPT2. Berdasarkan hal tersebut, penelitian ini menggabungkan model GPT2 dan Algoritma Genetika (GPT2-GA) untuk membangun level pada Sokoban.

II. STUDI LITERATUR

Procedural Content Generation

Pada topik *Procedural Content Generation* (PCG) terdapat beberapa aspek konten gim yang mempengaruhi *gameplay*, salah satunya adalah pembuatan level pada gim. Penelitian sebelumnya menyebutkan pembangunan level yaitu *Procedural Level Generation* menjadi salah satu bagian yang sering dikembangkan dalam topik PCG sebesar 51% dibandingkan pembuatan konten lainnya seperti misi, karakter, atau item (Liapis, 2020). Salah satu jenis gim yang paling umum digunakan untuk pengembangan level adalah puzzle. Seperti contohnya adalah Sokoban yang memiliki

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

representasi sangat terstruktur dan sering kali dalam bentuk grid 2D atau larik ubin (*tile based*). Level Sokoban dapat direpresentasikan menggunakan karakter ASCII untuk memanipulasi dan menghasilkan level baru. Namun, level ini harus mempertimbangkan pemain agar dapat diselesaikan karena pemain akan menemukan tantang berupa dinding atau sudut yang sempit sehingga gim tidak dapat diselesaikan (Bazzaz & Cooper, 2025). Hal ini menjadi tantang dalam pembuatan level yang harus dapat diukur secara objektif melalui *playability* atau *solvability*.

PCG Berbasis Pencarian

Pembangunan level pada gim memiliki dua pendekatan yaitu berbasis pencarian dan berbasis *Machine Learning*. Pada metode berbasis pencarian adalah Perlin Noise yang dapat membangkitkan sebuah peta yang biasanya berbentuk bukit, gua, awan, serta menentukan penempatan objek yang tidak terlalu teratur namun tidak sepenuhnya acak untuk mencipatkan level secara prosedural (Ramadhan & Indriyanti, 2022). Metode lainnya adalah Algoritma Genetika pada gim Dungeon Driver yang berhasil digunakan untuk pembuatan level dengan waktu sekitar 0.08 hingga 0.3 detik dengan ukuran *tile* (32x32 atau 64x64). Penelitian tersebut melakukan pengujian yang menunjukkan bahwa penggunaan *seed* yang berbeda dengan *fill percentage* yang sama (45%) menghasilkan tata letak level yang bervariasi dan unik (Eka Wahyu Hidayat et al., 2024). Metode Algoritma Genetika juga digunakan untuk tema visual novel yang dapat membangun alur cerita yang efektif (Rikandi & Nudin, 2022). Penerapan PCG dalam pembuatan level gim juga diterapkan pada *Game Overcooked* yang berjalan secara multiplayer. Penulis menggunakan evaluasi *Game Experience Questionnaire* (GEQ) untuk mengukur pengalaman bermain dari level gim yang dibuat menggunakan Algoritma Genetika. Hasil dari kuesioner tersebut menjelaskan bahwa level yang dibangun dapat disesuaikan dengan tingkat keahlian pemainnya sehingga meningkatkan pengalaman (Baek et al., 2022). Pembangunan ini terbukti efektif dan efisien menghasilkan level yang dapat dimainkan.

PCG Berbasis *Machine Learning*

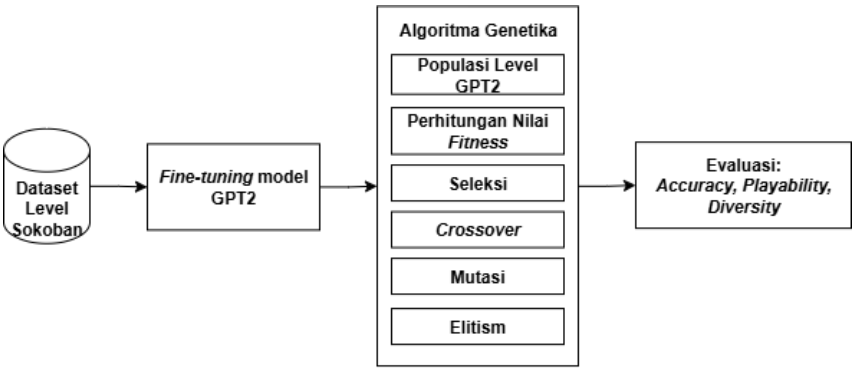
Berdasarkan hasil penelitian lain, PCG dilakukan dengan pendekatan machine learning yang disebut dengan *Procedural Content Generation Machine Learning* (PCGML). Penelitian sebelumnya menggunakan GAN untuk menghasilkan level untuk *gim first person shooter DOOM*. Peneliti melatih dua model yaitu GAN standar dan *Conditional GAN* (cGAN) yang menyertakan fitur-fitur topologi (seperti jumlah ruangan) untuk memberikan kontrol lebih. Pada gim tersebut ditemukan bahwa kurang dari 20% dari level yang dihasilkan untuk setiap gim tidak dapat dimainkan, yang menunjukkan tingkat keberhasilan fungsional yang tinggi (Mao et al., 2024). Namun, GAN masih kesulitan untuk mereproduksi detail level yang lebih kecil, kemungkinan besar karena adanya noise dalam proses generasi. Model yang lain yaitu Transformer khususnya pada Transfer Learning GPT2 yang mengungguli model LSTM, dimana *accuracy* prediksi ubin sebesar 93% dan akurasi prediksi jalur sebesar 91% (Sudhakaran et al., 2023). Penelitian lain yaitu gim berbasis visual novel diaman GPT2 digunakan membangun konten dalam percakapan NPC. Penelitian tersebut melakukan evaluasi pembuatan konten berdasarkan kuesioner yang terdiri dari *learnability, assets, educative, error,* dan *satisfaction*. Hasil penelitian tersebut menunjukkan bahwa pemain merasa tertarik dengan konten gim visual dan mudah dalam memainkan gim tersebut (Rikandi & Nudin, 2022). Berdasarkan penelitian sebelumnya, model GPT2 menunjukkan bahwa model ini adaptif untuk pembangunan level maupun konten dengan menggunakan dataset berbentuk teks.

III. METODE

Pada penelitian ini model Hybrid GPT2 dan Algoritma Genetika (GPT2-GA) ditunjukkan pada Gambar 1. Model GPT2 akan dilakukan *fine-tuning* menggunakan data Sokoban. Setelah itu, model GPT2 disimpan untuk digunakan dalam membangun level Sokoban. Pembangunan dilakukan proses iterasi untuk membuat populasi level yang digunakan dalam Algoritma Genetika. Tiap level tersebut akan dinilai berdasarkan nilai *fitness* yang berfungsi mengukur nilai tiap-tiap individu

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

berupa level Sokoban. Proses selanjutnya dalam Algoritma Genetika yaitu seleksi, *crossover*, mutasi, dan elitism.



Gambar 1. Model Hybrid GPT2 dan Algoritma Genetika untuk Level Gim Sokoban

Dataset Level Sokoban

Dataset yang digunakan yaitu Boxoban-levels dimana terdiri dari level Sokoban berukuran 10x10 yang direpresentasikan sebagai string ASCII, dimana setiap level berisi empat kotak dan empat target menggunakan pengkodean berikut: '#' untuk dinding, '@' untuk pemain, '$' untuk kotak, dan '.' untuk target. Dataset ini dikelompokkan ke dalam dua tingkat kesulitan yaitu *'medium'* dan *'hard'*. Prosedur pembuatan untuk setiap tingkat kesulitan didasarkan pada penelitian sebelumnya (Summerville et al., 2016). Pada penelitian ini kedua tingkatan tersebut digunakan untuk proses *fine-tuning*.

Terdapat beberapa file dalam Boxoban-levels dari setiap data *'medium'* dan *'hard'* dengan nama *file* "000.txt" yang masing-masing *file* berisi 1000 level sehinga total data yang digunakan adalah 2000 level. Data tersebut akan dibagi menjadi data *training*, *validation*, dan *testing*. Peneliti menambahkan atribut *input* yang berfungsi sebagai *prompt* untuk membangun sebuah level. Sebagai contoh *input* dituliskan dengan format "SOKOBAN_LEVEL: walls=<jumlah_papan> players=<jumlah_player> boxes=<jumlah_kotak> goals=<jumlah_tujuan> size=<ukuran_level> difficulty=<kesulitan_level>". *Input* ini akan membantu dalam proses *fine-tuning* sehingga model dapat memahami konteks yang spesifik dengan menuliskan jumlah untuk setiap karakteristik level Sokoban. Secara detail contoh dari dataset yang berupa *input* dan *output* ditunjukkan pada Tabel 1. Selanjutnya terdapat kolom Level sebagai contoh level berbentuk visualisasi grafis.

Tabel 1. Contoh Dataset Boxoban

Input	Output	Level
SOKOBAN_LEVEL: *walls=77 players=1 boxes=4 goals=4 size=10x10 difficulty=medium*	########## # .@###### #$$####### # ####### # ####### # ######## # ###### #.$.#### # . $ #### ##########	

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

SOKOBAN_LEVEL: *walls=62 players=1 boxes=4 goals=4 size=10x10 difficulty=medium*	########## # . $@ # # $ ## # # ##### # ######## # $. # #. # # .# ####### $# ###### # ##########	
SOKOBAN_LEVEL: *walls=64 players=1 boxes=4 goals=4 size=10x10 difficulty=medium*	########## ## # ## $. $ # ####### # ### . # ##.@$ #### # $ ##### #. ###### # ###### ##########	

***Fine-tuning* GPT2**

GPT2 (Generative Pre-trained Transformer 2) adalah model *autoregressive language* model yang menggunakan arsitektur Transformer dengan *decoder only architecture*. Pada dataset sokoban penulis menggunakan model "GPT2-small" dengan spesifikasi 12 *layers*, 12 *heads*, dan 768 *hidden dimension*. Dataset dari Sokoban yang akan diproses pada model GPT2 ditambahkan dengan token spesial pada model GPT2 yaitu <|startoftext|>, <|endoftext|>, <|pad|> sehingga data yang dihasilkan memiliki format sebagai berikut <|startoftext|>SPECIFICATION ⟶ LEVEL<|endoftext|>. Seperti contohnya pada Tabel 1 untuk baris 1 akan menghasilkan format data <|startoftext|>SOKOBAN_LEVEL: walls=77 players=1 boxes=4 goals=4 size=10x10 difficulty=medium ⟶ ##########\n# .@######\n#$$#######\n# #######\n# #######\n# ########\n# ######\n#.$.####\n# . $ ####\n##########. Token <|pad|> berfungsi untuk menyamakan panjang *sequence* dalam *batch training* sepanjang 512. Setelah itu, peneliti melakukan proses *fine-tuning* dengan *hyperparameter* yaitu *epoch* 8, *learning rate* 5e-4, *batch size* 2, *max length* 512, *warmup steps* 1e2, dan *optimizer* AdamW**.**

Pembangunan Level Model Hybrid GPT2 dan Algoritma Genetika

Peneliti menggunakan pendekatan GPT2 untuk membangun level sesuai dengan *prompt* yang dimasukkan oleh pemain, sedangkan Algoritma Genetika melakukan optimasi hasil pembangunan level dari GPT2. GPT2 akan membangun 5 bentuk level yang berbeda dari hasil model. Setiap level akan dievaluasi dan dioptimasi oleh Algoritma Genetika dengan melakukan beberapa langkah berikut.

1. Populasi Level GPT2

Proses pertama diawali dengan pembangunan level oleh model GPT2 sebanyak 20. Pembangunan ini dilakukan dengan mekanisme *autoregresif* dimana model memprediksi token berikutnya berdasarkan *sequence token* sebelumnya. Model menerima input (dalam konteks Sokoban berupa spesifikasi level), melakukan *forward pass* melalui *layers* untuk menghasilkan distribusi probabilitas atas *vocabulary*. Proses ini memahami token dari awal sampai bertemu dengan tanda '⟶', kemudian akan memprediksi token selanjutnya sampai mencapai kondisi berhenti (seperti token <|endoftext|> atau maksimal panjang *sequence*).

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

2. Nilai *Fitness*

Nilai *fitness* dalam konteks Algoritma Genetika berfungsi sebagai fungsi objektif untuk menghitung kualitas individu dalam populasi. Pada pembangunan Level Sokoban menggunakan beberapa nilai yaitu *validity* ($V(L)$), *solvabalitas* ($S(L)$), *balance* ($B(L)$), *complexity* ($C(L)$), *compactness* ($K(L)$) untuk menghasilkan level yang sesuai dengan prinsip desain gim dalam Sokoban. Secara matematis perhitungan nilai fitness ditunjukkan pada notasi (1), dimana setiap komponen memiliki bobot masing-masing. Penentuan bobot didasarkan pada faktor utama yang harus dipenuhi yaitu validity Jika nilai *validity* 0 maka nilai *fitness* akan mengembalikan nilai 0.

$$nilai_{fitness} = (V(L) \times 50\%) + (S(L) \times 30\%) + (B(L) \times 10\%) + (C(L) \times 5\%) + (K(L) \times 5\%) \quad (1)$$

Pada nilai *validity* $V(L)$ merupakan pengecekan dasar dengan melihat aturan struktrual dasar seperti pemain, keseimbangan, minimal penyelesaian, bentuk tepi, dan lebar peta. Setiap aspek tersebut dijelaskan pada Tabel 3 dengan aturan yang sesuai dengan kondisi dataset, seperti contohnya pada lebar peta yang terdiri dari 10 buah tembok. Secara notasi matematika maka dapat dituliskan sebagai variable $V(L)$, dimana ketika nilai valid maka akan bernilai 1 ketika tidak bernilai 0. Hal ini bertujuan untuk mengurangi komputasi ketika pembuatan level.

Table 2. Aturan *Validity*

Aturan	Visualisasi
Tepat satu player (@)	######### ← Batas atas (semua #) # # ← Pemain(@), Tujuan(.), Dinding(#) # $ $ # ← Kotak($) # .### .# ← Tujuan-tujuan(.) # $ $ # ← Kotak($) # . # ← Tujuan(.) # # ← Ruang kosong ######### ← Batas bawah (semua #)
Jumlah kotak sama dengan target	
Minimal satu kotak dan satu target	
Terdapat border dinding yang mengelilingi area	
Minimal 10 dinding	

Nilai selanjutnya adalah *solvability* $S(L)$ yang digunakan untuk mengukur kemungkinan suatu level dapat diselesaikan oleh pemain. Perhitungan dalam *solvability* menggunakan pendekatan *heuristic* sederhana dengan mengurangi nilai awal yaitu 1 dengan aturan yang mendukung level tidak dapat diselesaikan. Kondisi pertama terkait dengan deteksi tepi samping pada level dimana ketika kotak berada di posisi pojok namun tidak memiliki target, maka kondisi ini dinamakan *deadlock*. Kondisi kedua metode melakukan pencarian sederhana menggunakan algoritma *Breadth First Search (BFS)* dari posisi pemain untuk menghitung berapa banyak ruang kosong yang dapat dijangkau. Ketika ruang kosong seluruhnya dapat dijangkau maka nilainya adalah 1. Hasil nilai akhirnya dikalikan dengan bobot dari $S(L)$. Pendekatan ini menerima adanya *trade off* antara akurasi dan efisiensi, di mana beberapa level yang sebenarnya tidak dapat diselesaikan mungkin masih mendapat skor *solvability* yang tinggi.

Nilai *balance* $B(L)$ merupakan komponen *fitness* yang mengukur keseimbangan proporsi antar elemen dalam level Sokoban untuk memastikan level memiliki distribusi *gameplay* yang optimal dan menyenangkan. Komponen ini fokus pada aspek *space efficiency* dan keseimbangan penggunaan ruang, dimana level yang baik harus memiliki proporsi yang tepat antara elemen-elemen interaktif (kotak, target, pemain) dengan ruang kosong yang tersedia. Pada nilai ini terdapat perhitungan yang dinotasikan secara matematis (2) dan (3).

$$total_{interaktif} = jumlah_{kotak} + jumlah_{target} + jumlah_{pemain} \quad (2)$$

$$rasio_{ruang} = \frac{total\_interaktif}{ruang_{kosong}} \quad (3)$$

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

$$B(L) = 1 - \frac{|rasio_{ruang} - 0.5|}{0.5} \quad (4)$$

Complexity $C(L)$ merupakan komponen *fitness* yang mengukur seberapa sesuai tingkat kesulitan suatu level Sokoban dengan target desain yang diinginkan. Evaluasi jumlah kotak menjadi faktor pertama dalam menentukan *complexity score*. Sistem menetapkan target optimal sebanyak 4 kotak sebagai standar tingkat kesulitan yang diinginkan. Selain itu, evaluasi jumlah papan menjadi faktor kedua dengan target optimal 65 buah untuk level berukuran standar. Masing-masing evaluasi dapat dinotasikan sebagai berikut.

$$evaluasi_{kotak} = \frac{1 - |jumlah_{kotak_{aktual}} - 4|}{10} \quad (5)$$

$$evaluasi_{dinding} = \frac{1 - |jumlah_{dinding_{aktual}} - 65|}{100} \quad (6)$$

Berdasarkan notasi (5) dan (6) maka semakin jauh jumlah kotak atau papan dari target maka semakin rendah skornya. Pemilihan standar 4 didapatkan dari dataset yang hanya pasti terdiri dari 4 kotak, sedangkan standar 65 dipilih berdasarkan analisis bahwa jumlah ini menghasilkan keseimbangan yang baik antara ruang gerak yang cukup dan tantangan navigasi yang memadai. Setelah itu, kedua nilai akan di rata-rata menggunakan fungsi $max\ (0, avg_{kotak_{dinding}})$.

$$C(L) = max\ (0, avg_{kotak_{dinding}}). \quad (7)$$

Nilai *compactness* $K(L)$ merupakan komponen untuk mengevaluasi apakah elemen-elemen *gameplay* tersebar secara optimal tanpa menciptakan level yang terlalu *spread out* atau memiliki area kosong yang berlebihan di pinggiran. Konsep ini penting karena level yang terlalu tersebar akan mengurangi fokus pemain, sedangkan tantangan yang lebih tinggi akan menghasilkan pengalaman yang lebih fokus. Sistem melakukan *scanning* untuk menemukan koordinat minimum dan maksimum (min_row, max_row, min_col, max_col) dari semua sel yang bukan papan (#). Proses ini secara efektif membuat "kotak pembatas" yang melingkupi semua elemen *gameplay* aktif dalam level. Secara matematis perhitungan area yang digunakan ditunjukkan pada notasi (7), total area ditunjukkan pada notasi (8), dan nilai akhir *compactness* ditunjukkan pada notasi (9).

$$area_{digunakan} = (max_{row} - min_{row} + 1) \times (max_{col} - max_{col} + 1) \quad (8)$$

$$total_{area} = lebar_{level} \times tinggi_{level} \quad (9)$$

$$K(L) = \frac{area_{digunakan}}{total_{area}} \quad (10)$$

3. Seleksi

Proses seleksi pada Algoritm Genetika menggunakan *tournament_selection* dengan prinsip kompetisi sederhana di antara beberapa individu pada populasi. Proses dimulai dengan memilih sejumlah kandidat yang ditentukan oleh parameter *tournament size* dengan nilai 3. Kandidat sejumlah 3 melakukan kompetisi dimana nilai *fitness* yang tertinggi akan menjadi pemenangnya dan terpilih sebagai induk. Metode ini memiliki keunggulan dalam menjaga indvidu dengan nilai *fitness* yang rendah memungkinkan tetap dipertahankan karena individu berkompetisi dengan individu lainnya yang lebih rendah.

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

4. Mutasi

Proses mutasi berfungsi untuk memperkenalkan variasi genetik pada level Sokoban dengan probabilitas 15% per individu (*mutation rate*), dimana setiap gen internal (*non-border*) level dapat mengalami salah satu dari tiga strategi mutasi secara acak: *swap adjacent* (menukar karakter dengan sel tetangga), *change type* (mengubah jenis karakter dengan probabilitas *weighted*), atau *local shuffle* (mengacak distribusi karakter dalam area 3x3 lokal). Setelah mutasi, sistem secara otomatis memperbaiki struktur level dengan memastikan border tetap berupa dinding (#), menjaga jumlah player tepat satu, dan menyeimbangkan jumlah kotak dengan target, sehingga menghasilkan level yang valid namun memiliki variasi genetik untuk mencegah konvergensi dan memungkinkan eksplorasi ruang solusi yang lebih luas dalam proses evolusi.

5. Elitism

Strategi elitism ini memastikan bahwa level Sokoban dengan *fitness* tertinggi dari generasi sebelumnya tidak akan hilang akibat operasi genetik yang bersifat stokastik, sehingga algoritma dijamin selalu mempertahankan solusi terbaik yang pernah ditemukan dan mencegah degradasi kualitas populasi. Parameter *elite size* yang tetap dipertahankan sebesar 4, dimana individu terbaik langsung ke generasi berikutnya tanpa melalui proses *crossover* atau mutasi.

Evaluasi

Pada tahap ini penulis melakukan evaluasi terhadap proses *fine-tuning* model GPT2 serta pembangunan level menggunakan GPT2 dan Algoritma Genetika. Pada proses *fine-tuning* dilakukan perhitungan *loss* dan *accuracy* menggunakan data *training* dan *validation*, dimana nilai didapatkan dari perbandingan antara level yang dibangun oleh model dengan level pada dataset atau *ground truth*. Selanjutnya evaluasi untuk mengukur hasil dari pembangunan level dengan beberapa kriteria yaitu *accuracy, playability, dan diversity*.

Pada *accuracy* menggunakan persamaan matematis untuk pembangunan level yang telah dilakukan pada penelitian sebelumnya (Nasir & Togelius, 2023) yang ditunjukkan pada notasi (10). Berdasarkan notasi (10) $persentase_{prompt}$ adalah target persentase ubin pola yang dituliskan dalam *prompt* yang diberikan kepada model. $persentase_{hasil}$ adalah persentase aktual dari ubin pola yang ada pada level yang berhasil digenerasi oleh model. Nilai akurasi yang mendekati 1 (atau 100%) menunjukkan bahwa model berhasil menghasilkan jumlah ubin pola yang sangat sesuai dengan yang diinstruksikan dalam *prompt*. Pada *playability* menggunakan metode A* untuk mencari solusi, apabila A* tidak menemukan solusi dalam waktu tertentu maka level yang dibangun tidak dapat diselesaikan (Todd et al., 2023). Terakhir evaluasi mengenai *diversity* menggunakan pendekatan berbasis graf di mana setiap level yang dihasilkan menjadi sebuah simpul. Dua level dianggap "berbeda" dan dihubungkan oleh sebuah garis jika jarak string mereka di atas ambang batas k=5 maka level tersebut berbeda. Secara matematis ditunjukkan pada notasi (11).

$$accuracy = \frac{persentase_{prompt} - persentase_{hasil}}{persenase_{prompt}} \quad (10)$$

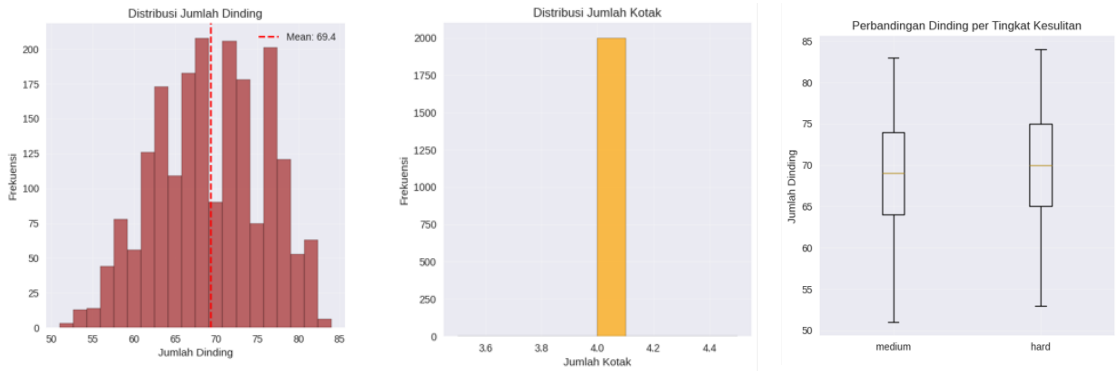
$$diversity = \frac{jumlah\ level\ diatas\ ambang\ batas}{jumlah\ total\ level\ yang\ dihasilkan} \quad (11)$$

IV. HASIL DAN PEMBAHASAN

Hasil Distribusi Dataset

Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

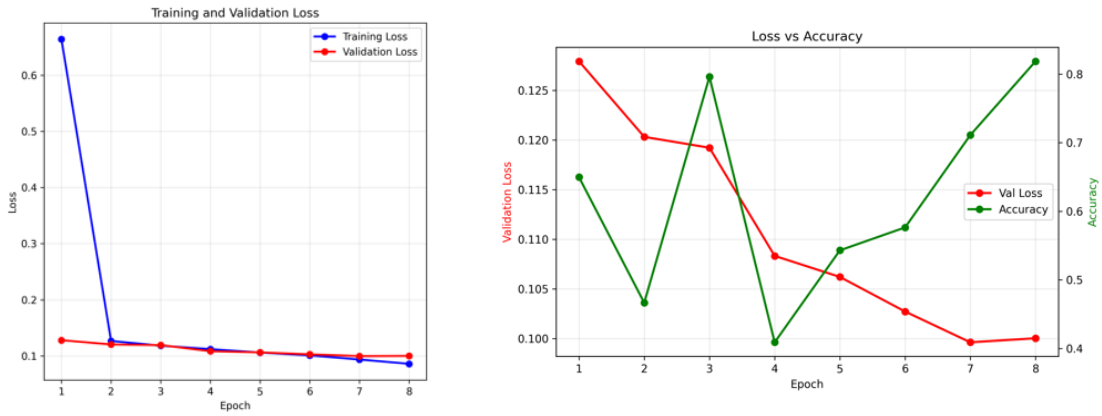
Peneliti melakukan pembagian dataset yang terdiri dari data *training, validation,* dan *testing*. Rasio pembagian dataset terdiri dari 70: 15: 15 sehingga data *testing* sebesar 1400, data *validation* sebesar 300, dan data *testing* sebesar 300. Hasil dari distribusi data Sokoban ditunjukkan pada Gambar 2, dimana data *training* memiliki nilai yang sama untuk jumlah kotak yaitu sebesar 4 yang ditunjukkan pada Gambar 2. Hal ini dikarenakan untuk keseluruhan dataset dari Boxoban-levels memiliki jumlah yang sama untuk kotaknya yaitu 4. Sedangkan untuk dindingnya menujukkan jumlah yang berbeda-beda dengan nilai rata-rata didapatkan sebesar 69.4. Persebaran jumlah dinding paling tinggi di tingkat kesulitan 'hard' yang ditunjukkan pada diagram kotak-garis



Gambar 2. Visualisasi Dataset *Training* Sokoban

Hasil *Fine-tuning* GPT2

Data yang telah dibagi akan digunakan melakukan *fine-tuning* pada model GPT2 agar model dapat belajar pola dari data Sokoban. Proses *fine-tuning* GPT-2 pada level Sokoban menunjukkan hasil yang cukup baik dari sisi penurunan *loss* dan peningkatan akurasi. *Training loss* turun drastis dari 0.65 ke 0.085, sedangkan *validation loss* turun stabil ke 0.10. Akurasi model meningkat dari awal yang fluktuatif hingga mencapai 81.9% di akhir training. Rata-rata akurasi sebesar 62.1% dengan deviasi 13.9% menunjukkan adanya ketidakstabilan di beberapa epoch awal, meskipun akhirnya model mampu memahami pola dalam data dengan cukup baik.



Gambar 3. Grafik *Loss* dan *Accuracy* untuk *Fine-tuning* GPT2

Evaluasi Hybrid GPT2 dan Algoritma Genetika

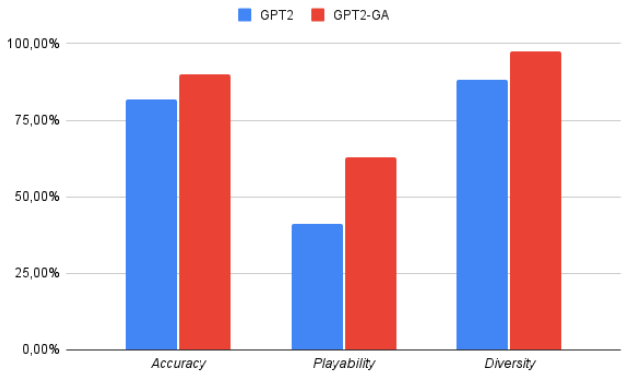
Remik: Riset dan E-Jurnal Manajemen Informatika Komputer
Volume 9, Number 3, Agustus 2025
http://doi.org/10.33395/remik.v9i3.15188
e-ISSN : 2541-1330
p-ISSN : 2541-1332

Berdasarkan hasil evaluasi Model Hybrid GPT2 dan Algoritma Genetika (GPT2-GA) menunjukkan peningkatan kinerja yang signifikan dibandingkan dengan model GPT2 pada semua metrik evaluasi utama. Hasil perbandingan menunjukkan bahwa pendekatan hybrid ini berhasil mengatasi keterbatasan yang dimiliki oleh model generatif dalam konteks generasi level gim Sokoban.

Pada Gambar 4 dapat dilihat bahwa GPT2 mampu menghasilkan level dengan tingkat diversitas yang cukup tinggi sebesar 88.2%, namun memiliki kelemahan signifikan dalam aspek *playability* dengan hanya mencapai 41.3%. Hal ini menunjukkan bahwa meskipun GPT2 dapat menghasilkan variasi level yang beragam berdasarkan pola *training* data, level yang dihasilkan sering kali tidak memenuhi aturan permainan yang diperlukan untuk dapat dimainkan. *Accuracy* sebesar 81.9% juga menunjukkan bahwa pemahaman pola dengan *ground truth* masih memerlukan optimisasi lebih lanjut.

Implementasi sistem hybrid GPT2-GA memberikan solusi yang efektif untuk mengatasi keterbatasan tersebut. Penggabungan kemampuan generatif GPT2 sebagai pembentukan level yang unik dan Genetic Algorithm sebagai mekanisme optimisasi berhasil meningkatkan *accuracy* menjadi 90.1%, yang merepresentasikan peningkatan 8.2% dari GPT2. Peningkatan ini menunjukkan bahwa Algorimta Genetika mampu mengoptimalkan pola level yang dihasilkan GPT2 untuk lebih sesuai dengan target pola yang diharapkan.

Peningkatan paling besar terlihat pada evaluasi *playability* yang melonjak dari 41.3% menjadi 62.8%, atau peningkatan sebesar 21.5%. Peningkatan ini sangat signifikan karena menunjukkan bahwa Algoritma Genetika berperan krusial dalam memastikan level yang dihasilkan memenuhi aturan mekanis permainan Sokoban, seperti keseimbangan jumlah kotak dan target serta *validity*. Tidak hanya itu, nilai *diversity* juga mengalami peningkatan dari 88.2% menjadi 97.5%, mencapai tingkat keunikan yang sangat tinggi dengan peningkatan 9.3%. Hasil ini menunjukkan bahwa kombinasi kreativitas GPT2 dalam menghasilkan pola dasar dengan eksplorasi ruang solusi yang dilakukan Algoritma Genetika menghasilkan variasi level yang lebih kaya dan beragam. Algoritma Genetika tidak hanya mengoptimalkan aturan dari gim Sokoban tetapi juga melakukan eksplorasi yang memperluas ruang desain level yang dapat dihasilkan model.



Gambar 3. Evaluasi GPT2 dan GPT2-GA

Diskusi

Hasil penelitian ini menunjukkan bahwa pendekatan gabungan GPT2-GA jauh lebih baik dibandingkan GPT2 biasa dalam membuat level Sokoban, dengan peningkatan *playability* 21.5%. Keberhasilan ini membuktikan bahwa menggabungkan kedua metode tersebut dapat mengatasi kelemahan model bahasa neural murni dalam menyelesaikan masalah dengan aturan khusus. GPT2

e-ISSN : 2541-1330
p-ISSN : 2541-1332

secara berkala berhasil mencegah konvergensi dini yang sering terjadi dalam Algoritma Genetika tradisional, sambil tetap menjaga kualitas melalui seleksi elitism.

Meskipun menunjukkan hasil yang bagus untuk seluruh evaluasi, penelitian masih berfokus pada domain Sokoban yang relatif sederhana dengan jumlah kotak yaitu 4 untuk seluruh data, perbedaan hanya pada jumlah dan tata letak dinding. Hal ini perlu ditambahkan jumlah kotak yang lebih bervariasi sehingga pemain dapat mengontrol jumlah kotak dan target pada model GPT2. Penggunaan nilai *fitness* masih berdasarkan peraturan yang terbatas yang memungkinan perlu didefinisikan kembali ketika terdapat gim puzzle yang lebih kompleks. Selain itu, diperlukan evaluasi langsung mengukur pengalaman pengguna pengukuran langsung oleh penggun seperti *Game Experience Questionnaire* (GEQ) sehingga akan mengetahui keberlanjutan pemain dalam menyelesaikan level Sokoban.

V. KESIMPULAN

Procedural Content Generation (PCG) khususnya pada pembuatan level memberikan pengalaman pengguna yang baik karena pemain mendapatkan tantangan baru. Penelitian ini menggunakan Hybrid GPT2-GA dalam membangun level Sokoban yang menunjukkan performa yang superior dibandingkan GPT2 standar dalam semua aspek evaluasi yaitu *accuracy*, *playability*, dan *diversity*. Pendekatan hybrid ini berhasil mengatasi kelemahan fundamental dari model generatif murni dengan mengintegrasikan mekanisme optimasi evolusioner yang memastikan level yang dihasilkan tidak hanya kreatif tetapi juga fungsional dan dapat dimainkan. Tingkat *playability* yang mencapai 62.8%, model ini dapat diandalkan untuk menghasilkan level yang dapat diselesaikan, sementara *diversity* 97.5% memastikan variasi konten yang kaya.

VI. REFERENSI

Baek, I.-C., Ha, T.-G., Park, T.-H., & Kim, K.-J. (2022). Toward Cooperative Level Generation in Multiplayer Games: A User Study in Overcooked! *2022 IEEE Conference on Games (CoG)*, 276–283. https://doi.org/10.1109/CoG51982.2022.9893581

Bazzaz, M., & Cooper, S. (2025). Analysis of Robustness of a Large Game Corpus. *Proceedings of the 20th International Conference on the Foundations of Digital Games*, 1–9. https://doi.org/10.1145/3723498.3723820

Eka Wahyu Hidayat, Euis Nur Fitriani Dewi, & Insan Saleh Ramadhan. (2024). APPLICATION OF PROCEDURAL CONTENT GENERATION SYSTEM IN FORMING DUNGEON LEVEL IN DUNGEON DIVER GAME. *Jurnal Teknik Informatika (Jutif)*, *5*(3), 873–881. https://doi.org/10.52436/1.jutif.2024.5.3.1465

Farrokhi Maleki, M., & Zhao, R. (2024). Procedural Content Generation in Games: A Survey with Insights on Emerging LLM Integration. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, *20*(1), 167–178. https://doi.org/10.1609/aiide.v20i1.31877

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, *54*(10s), 1–41. https://doi.org/10.1145/3505244

Liapis, A. (2020). 10 Years of the PCG workshop: Past and Future Trends. *International Conference on the Foundations of Digital Games*, 1–10. https://doi.org/10.1145/3402942.3409598

Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G. N., & Togelius, J. (2021). Deep learning for procedural content generation. *Neural Computing and Applications*, *33*(1), 19–37. https://doi.org/10.1007/s00521-020-05383-8

Mao, X., Yu, W., Yamada, K. D., & Zielewski, M. R. (2024). *Procedural Content Generation via Generative Artificial Intelligence*. http://arxiv.org/abs/2407.09013

Nasir, M. U., & Togelius, J. (2023). Practical PCG Through Large Language Models. *2023 IEEE Conference on Games (CoG)*, 1–4. https://doi.org/10.1109/CoG57401.2023.10333197

Ramadhan, D. A., & Indriyanti, A. D. (2022). Procedural Content Generation pada Game World Exploration Sandbox Menggunakan Alogoritma Perlin Noise. *Journal of Informatics and*

e-ISSN : 2541-1330
p-ISSN : 2541-1332

Computer Science (JINACS), *4*(01), 86–91. https://doi.org/10.26740/jinacs.v4n01.p86-91

Rikandi, N. C., & Nudin, S. R. (2022). Rancang Bangun Visual Novel Peduli Lingkungan dengan Metode Procedural Content Generation. *Journal of Informatics and Computer Science (JINACS)*, *4*(01), 131–142. https://doi.org/10.26740/jinacs.v4n01.p131-142

Sudhakaran, S., González-Duque, M., Glanois, C., Freiberger, M., Najarro, E., & Risi, S. (2023). *MarioGPT: Open-Ended Text2Level Generation through Large Language Models*. http://arxiv.org/abs/2302.05981

Summerville, A. J., Snodgrass, S., Mateas, M., & Ontañón, S. (2016). *The VGLC: The Video Game Level Corpus*. http://arxiv.org/abs/1606.07487

Świechowski, M., Godlewski, K., Sawicki, B., & Mańdziuk, J. (2023). Monte Carlo Tree Search: a review of recent modifications and applications. *Artificial Intelligence Review*, *56*(3), 2497–2562. https://doi.org/10.1007/s10462-022-10228-y

Todd, G., Earle, S., Nasir, M. U., Green, M. C., & Togelius, J. (2023). Level Generation Through Large Language Models. *Proceedings of the 18th International Conference on the Foundations of Digital Games*, 1–8. https://doi.org/10.1145/3582437.3587211

van Stegeren, J., & Myśliwiec, J. (2021). Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation. *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, 1–8. https://doi.org/10.1145/3472538.3472595

Wang, B.-Y., & Chang, C.-L. (2024). Wave Function Collapse Algorithm in Roguelite Game Level Design: An Analysis of Strategies and Challenges. *2024 10th International Conference on Applied System Innovation (ICASI)*, 359–361. https://doi.org/10.1109/ICASI60819.2024.10547926