

Eksplorasi Penggunaan Large Language Model (LLM) dalam Pembangunan Permainan Minesweeper dengan Python Programming

Qurhanul Rizqie¹, Nurul Afifah^{2*}, Ali Bardadi³

¹Teknik Informatika, Fakultas Ilmu Komputer Universitas Sriwijaya, Palembang, Indonesia

²Teknik Komputer, Fakultas Ilmu Komputer Universitas Sriwijaya, Palembang, Indonesia

³Sistem Informasi, Fakultas Ilmu Komputer Universitas Sriwijaya, Palembang, Indonesia

*Korespondensi: nurul@unsri.ac.id

ARTICLE INFO

Article History:

- Received 01 July 2023
- Received in revised form 25 August 2023
- Accepted 19 September 2023
- Available online 30 October 2023

ABSTRAK

Pembangunan perangkat lunak semakin melibatkan kecerdasan buatan (Artificial Intelligence), khususnya dalam konteks penggunaan LLM. Artikel ini mengeksplorasi potensi dan batasan penerapan ChatGPT 3.5 dalam pengembangan permainan Minesweeper menggunakan bahasa pemrograman Python. Dengan mengadopsi tiga fase pengembangan bertahap, eksperimen ini bertujuan untuk mengilustrasikan evolusi kompleksitas permainan seiring dengan instruksi-instruksi yang diberikan kepada model. Melalui proses ini, kami menganalisis kode yang dihasilkan oleh ChatGPT 3.5 dan membandingkannya dengan implementasi Minesweeper yang telah tersedia di domain publik.

Kata Kunci: Kecerdasan Buatan, ChatGPT 3.5, Bahasa Besar, Python, Minesweeper

ABSTRACT

The development of software increasingly involves Artificial Intelligence (AI), particularly in the context of using Large Language Models (LLM). This article explores the potential and limitations of applying ChatGPT 3.5 in developing Minesweeper games using Python. Adopting a three-phase incremental development approach, this experiment aims to illustrate the evolution of the game's complexity alongside the instructions given to the model. Throughout this process, we analyze the code generated by ChatGPT 3.5 and compare it with publicly available implementations of Minesweeper.

Keywords: Artificial Intelligence, ChatGPT 3.5, Natural Language, Python, Minesweeper

1. PENDAHULUAN

Salah satu tren yang sedang berkembang dalam pembelajaran mesin adalah Kecerdasan Buatan Generatif, yang merupakan algoritma pembelajaran mesin yang dirancang untuk mempelajari konten seperti teks, gambar, dan audio, serta menghasilkan artefak baru sebagai konten.[1] Salah satu Kecerdasan Buatan generatif yang paling populer saat ini adalah LLM (Large Language Model) yaitu model jaringan syaraf yang didasarkan pada data teks dengan

jumlah yang besar, dan dirancang untuk menghasilkan output yang menyerupai manusia, termasuk namun tidak terbatas pada prosa, puisi, dan bahkan kode program.[2]

Perkembangan ini memberikan tantangan bagi praktisi dan pendidik bidang ilmu komputer, di mana pemrograman adalah salah satu disiplin inti dari ilmu komputer.[3] Kecerdasan buatan generative telah mampu memberikan performa yang baik dalam suatu lingkungan yang terkontrol, namun mungkin akan kurang efektif dalam penerapan pada dunia nyata.[4] Namun, perkembangan Kecerdasan Buatan generatif juga memberikan peluang yang memungkinkan pengembangan sebuah program dilakukan dengan pengetahuan koding yang minimal.[5] Dalam penelitian ini, kami mengeksplorasi kemampuan salah satu Large Language Model untuk membuat kode menggunakan instruksi kustom. Kami menginstruksikan ChatGPT 3.5 untuk membuat permainan minesweeper dengan kompleksitas instruksi yang meningkat dalam 3 fase, kemudian kami membandingkan hasilnya dengan kode sumber permainan minesweeper yang tersedia secara publik di internet.

2. TINJAUAN PUSTAKA

2.1. LLM

Artificial Intelligence telah menjadi pendorong utama inovasi di berbagai bidang, termasuk pemrograman komputer. Dalam konteks ini, pemahaman terhadap LLM, seperti ChatGPT 3.5, telah memberikan dimensi baru dalam mengintegrasikan kecerdasan buatan ke dalam proses pembangunan perangkat lunak. Python, dengan sintaksis yang bersahabat dan beragamnya perpustakaan, menjadi pilihan utama dalam mengimplementasikan algoritma kecerdasan buatan. Keberagaman perpustakaan seperti TensorFlow, PyTorch, dan scikit-learn membuat Python menjadi bahasa yang ideal untuk mengembangkan dan menguji model kecerdasan buatan. LLM, seperti yang diwakili oleh ChatGPT 3.5, telah menunjukkan kemampuan untuk menghasilkan kode program yang bermakna. Eksplorasi kemampuan ini dalam konteks pengembangan perangkat lunak, termasuk pembangunan permainan seperti Minesweeper, memberikan wawasan tentang potensi model ini sebagai alat bantu pemrograman. Meskipun LLM menawarkan potensi besar dalam menciptakan kode, tantangan muncul terutama terkait keakuratan dan keamanan. Pengguna harus memahami batasan model dan peran kritis pengetahuan dasar pemrograman dalam memastikan kode yang dihasilkan sesuai dengan kebutuhan dan standar. Penggunaan AI dalam pembelajaran pemrograman juga menjadi fokus perhatian.

LLM dapat menjadi alat efektif untuk mendukung pembelajaran interaktif, memberikan solusi, dan mengeksplorasi konsep pemrograman secara kontekstual. Pengembangan permainan, seperti Minesweeper, menunjukkan potensi besar untuk menerapkan AI. Pemodelan kecerdasan buatan dapat digunakan untuk menciptakan agen pintar, meningkatkan pengalaman pengguna, dan mengoptimalkan berbagai aspek dari permainan. Meskipun AI dapat menghasilkan kode, peran pemrogram tetap krusial. Pemahaman konsep pemrograman, logika algoritma, dan penanganan kesalahan tetap menjadi tanggung jawab utama pemrogram, sementara AI dapat menjadi alat bantu yang efektif. Penerapan Kecerdasan Buatan dalam Python Programming, terutama melalui LLM, membawa dampak signifikan dalam pengembangan perangkat lunak. Pemahaman menyeluruh terhadap potensi dan tantangan dalam penerapan ini penting untuk memastikan hasil yang optimal dan sesuai dengan kebutuhan industri pemrograman.



Gambar 1. Ilustrasi Large Language Model

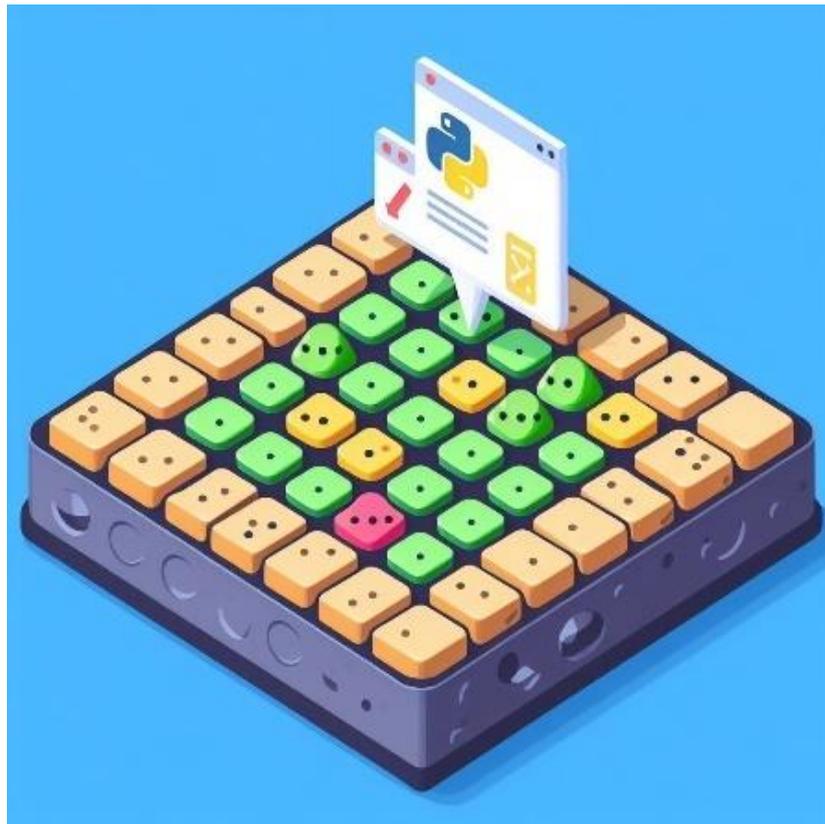
2.2. Minesweeper

Minesweeper adalah permainan teka-teki yang dimainkan pada papan persegi yang terbagi menjadi sel-sel kecil. Tujuan utama permainan ini adalah untuk mengungkap semua sel di papan kecuali yang berisi ranjau. Jika pemain mengungkap sel yang berisi ranjau, permainan berakhir. Papan Minesweeper dibagi menjadi sel-sel kecil, di mana setiap sel dapat berisi ranjau atau tidak. Ukuran papan dan jumlah ranjau dapat bervariasi, memberikan tingkat kesulitan yang berbeda. Sel yang tidak berisi ranjau dapat berisi angka, yang menunjukkan berapa banyak sel di sekitarnya yang berisi ranjau. Pemain dapat menggunakan informasi ini untuk membuat keputusan strategis tentang di mana ranjau mungkin berada. Minesweeper mengandalkan pemikiran logis dan strategi untuk menyelesaikan papan tanpa mengungkap sel-sel yang berisi ranjau. Pemain perlu mempertimbangkan angka di sekitar setiap sel untuk menentukan keberadaan ranjau.

Pemain perlahan membangun kepercayaan terhadap keputusan mereka saat mereka mengungkap sel-sel. Mereka dapat memperkirakan keberadaan ranjau berdasarkan informasi yang telah mereka kumpulkan selama permainan. Meskipun Minesweeper melibatkan elemen keberuntungan, keterampilan dan pemikiran logis memainkan peran penting dalam kesuksesan. Pemain yang terampil mampu membuat keputusan yang lebih baik berdasarkan informasi yang tersedia. Seiring berjalannya waktu, banyak algoritma dan strategi yang telah dikembangkan untuk menyelesaikan permainan Minesweeper secara otomatis. Beberapa menggunakan pendekatan rekursif, sementara yang lain mengandalkan logika matematika. Minesweeper dapat berfungsi sebagai alat pendidikan yang baik untuk mengembangkan keterampilan kognitif seperti pemikiran logis, perencanaan, dan analisis situasi. Ini juga dapat

meningkatkan kemampuan pemain dalam mengelola risiko. Minesweeper dapat memiliki efek psikologis, termasuk meningkatkan ketegangan dan kehati-hatian.

Pemain seringkali mengalami rasa puas dan kelegaan saat berhasil menyelesaikan papan tanpa mengungkap ranjau. Meskipun sederhana, Minesweeper telah menjadi salah satu permainan komputer paling ikonik dan populer sejak pertama kali diperkenalkan. Ini telah berevolusi dari versi awalnya menjadi variasi yang lebih kompleks dengan berbagai tingkat kesulitan. Dengan demikian, Minesweeper tidak hanya merupakan permainan yang menyenangkan, tetapi juga menciptakan lingkungan di mana pemain dapat mengembangkan dan menguji keterampilan kognitif dan strategis mereka.



Gambar 2. Minesweeper

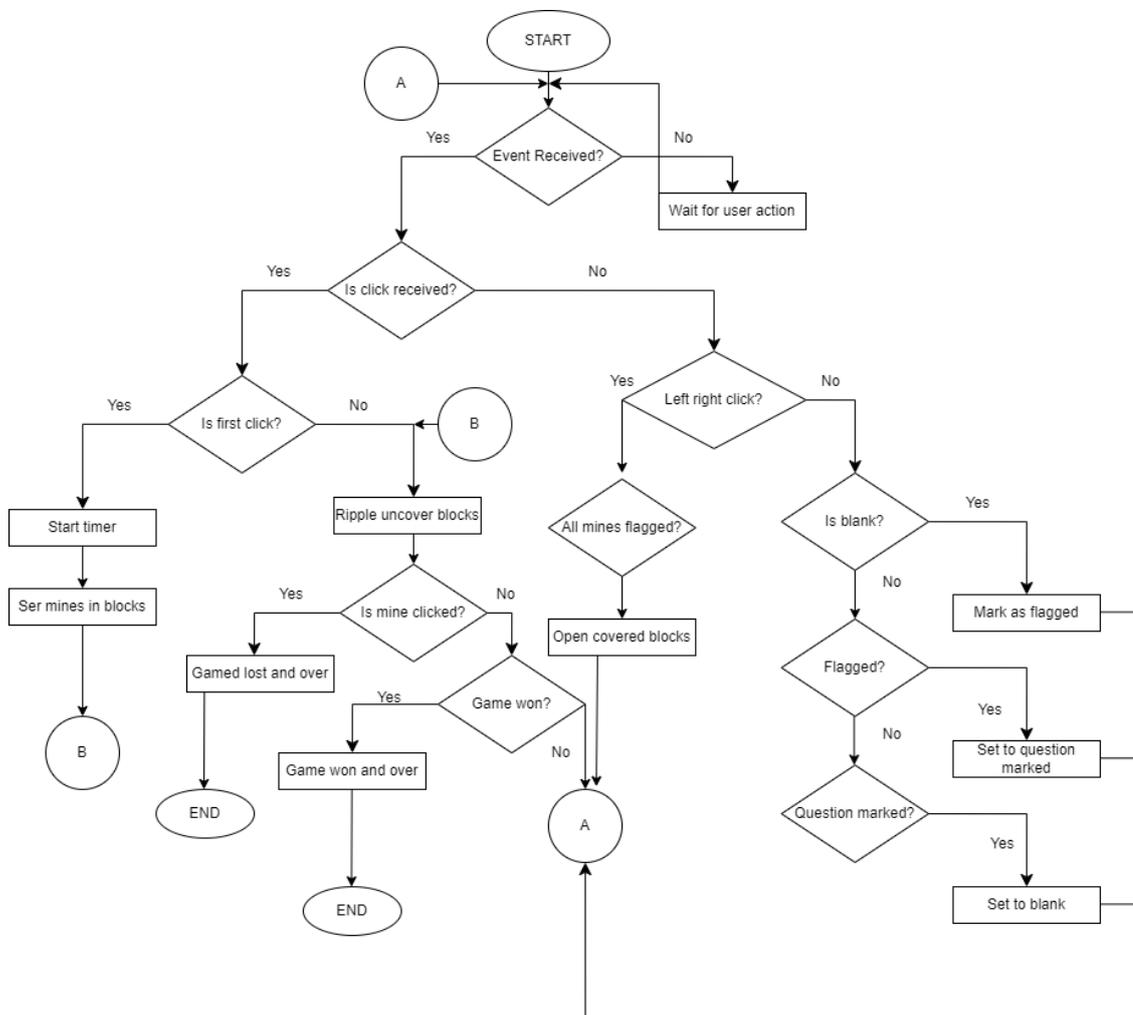
3. METODELOGI PENELITIAN

3.1 Eksplorasi Minesweeper

Dalam artikel ini, kami menginstruksikan ChatGPT 3.5 untuk menggenerasi kode untuk permainan Minesweeper pada bahasa pemrograman Python. Instruksi dimulai dengan perintah sederhana yang kemudian ditambahkan dengan fitur yang lebih kompleks untuk setiap fase.



Gambar 3. Fase Minesweeper



Gambar 4. Flowchart Minesweeper

Fase 1: Grid Dasar

Instruksi pertama yang diberikan pada ChatGPT 3.5 adalah untuk melakukan generasi kode permainan Minesweeper sederhana berukuran 5x5 yang mudah dimengerti. Adapun Prompt yang diberikan adalah sebagai berikut:

```
create 5x5 grid minesweeper game in phyton
```

Fase 2: Tingkat Kesulitan Terkendali

Tujuan berikutnya adalah memperkenalkan tingkat kesulitan yang terkendali, untuk meningkatkan daya tarik permainan. Diputuskan untuk mengelompokan tingkat kesulitan permainan dalam 3 tingkatan yaitu mudah, normal, dan sulit. Tingkatan kesulitan dinyatakan sebagai jumlah grid dari permainan minesweeper tersebut yaitu ukuran grid 3x3 sebagai tingkat mudah, ukuran grid 5x5 sebagai tingkat normal, dan ukuran grid 9x9 sebagai tingkat sulit. Adapun Prompt yang diberikan adalah sebagai berikut:

```
create nxn grid minesweeper game in phyton
constraint :
1. mines number <= 1/3 grid column size
2. if chosen cells selected have no neighboring mine cells reveal all neighboring cells
3. divided into 3 level
   3.1. easy, n = 3
   3.2 normal, n = 5
   3.3 hard, n = 9
```

Fase 3: Pilihan Main Lagi

Terakhir kami menambahkan instruksi pada ChatGPT 3.5 untuk memodifikasi kode dengan menambahkan opsi untuk memulai permainan kembali. Setelah menyelesaikan permainan, pemain diberi pilihan untuk memulai kembali dengan posisi mine dan tingkat kesulitan yang berbeda. Adapun Prompt yang diberikan adalah sebagai berikut:

```
create nxn grid minesweeper game in phyton
constraint :
1. mines number <= 1/3 grid column size
2. if chosen cells selected have no neighboring mine cells reveal all neighboring cells
3. divided into 3 level
   3.1. easy, n = 3
   3.2 normal, n = 5
   3.3 hard, n = 9
4. if end ask if want to play again
```

Kode yang digenerasi berdasarkan ketiga prompt diatas kemudian akan dibandingkan dengan kode yang tersedia secara publik. Kode tersebut ditemukan menggunakan perangkingan teratas pada pencarian google. [6]

4. HASIL DAN DISKUSI

ChatGPT 3.5 dapat melakukan generasi kode permainan minesweeper sesuai dengan instruksi yang diberikan. Prompt pertama menghasilkan kode permainan minesweeper dengan ukuran grid 5x5, dibandingkan dengan kode pada artikel [6] kode yang digenerasi oleh ChatGPT 3.5 secara umum memiliki struktur yang sama yaitu pengacakan posisi mines, pengecekan jumlah mines pada tetangga, pengecekan game over, dan looping permainan. Pada fase 2 kami menambahkan instruksi untuk membagi tingkat permainan menjadi mudah, normal, dan sulit, dengan asumsi tingkat kesulitan berdasarkan besar ukuran grid. Pada fase ini kode yang digenerasi oleh ChatGPT3.5 memberikan syntax error yaitu kekurangan pada salah satu lambang tutup kurung pada perulangan. Setelah syntax error diperbaiki program berjalan sesuai instruksi yang diberikan. Pemain dapat memilih tingkat kesulitan kemudian melakukan permainan minesweeper dengan ukuran grid yang sesuai dengan pilihan pemain tersebut.

Pada fase 3 kami menambahkan instruksi untuk memberikan pilihan pada pemain untuk bermain kembali dengan setelan tingkat kesulitan dan posisi mines yang berbeda. Pada fase ini kode yang digenerasi oleh ChatGPT3.5 juga memberikan syntax error yang sama yaitu kekurangan pada salah satu lambang tutup kurung pada perulangan. Setelah syntax error diperbaiki program berjalan sesuai instruksi yang diberikan. Setelah menyelesaikan permainan, pemain dapat bermain kembali dengan setelan yang berbeda.

Pseudocode Yang Digenerasi ChatGPT 3.5 Berdasarkan Prompt 3

```
import random

def create_board(size, num_mines):
    board = [[' ' for _ in range(size)] for _ in range(size)]
    # Place mines randomly on the board
    mines_placed = 0
    while mines_placed < num_mines:
        row, col = random.randint(0, size - 1), random.randint(0, size - 1)
        if board[row][col] != 'X':
            board[row][col] = 'X'
            mines_placed += 1
    return board
```

5. KESIMPULAN

Berdasarkan hasil dan diskusi diatas dapat disimpulkan bahwa LLM seperti ChatGPT 3.5 memiliki kemampuan untuk mengenerasi kode dengan fungsionalitas yang sama dengan contoh kode yang tersedia secara publik. ChatGPT 3.5 juga mampu untuk memodifikasi kode berdasarkan instruksi tambahan yang diberikan sehingga kode yang dihasilkan relative berbeda dengan kode yang disediakan secara publik. Namun demikian kode yang digenerasi oleh ChatGPT 3.5 masih rawan menghasilkan kesalahan sintaks, untuk itu pengetahuan dasar dan logika pemrograman diperlukan untuk dapat menggunakan LLM sebagai alat bantu pemrograman.

DAFTAR PUSTAKA

- [1].Sun, J., Liao, Q.V., Muller, M., Agarwal, M., Houde, S., Talamadupula, K., & Weisz, J.D. (2022, March). Investigating explainability of generative AI for code through scenario-based design. In 27th International Conference on Intelligent User Interfaces (pp. 212-228).
- [2].Denny, P., Prather, J., Becker, B.A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B.N., Santos, E.A., & Sarsa, S. (2023). Computing Education in the Era of Generative AI. arXiv preprint arXiv:2306.02608.
- [3].Becker, B.A., & Quille, K. (2019). 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), 338–344. <https://doi.org/10.1145/3287324.3287432>
- [4].Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). The impact of AI on developer productivity: Evidence from GitHub Copilot. arXiv preprint arXiv:2302.06590.
- [5].Becker, B.A., Prather, J., Denny, P., Luxton-Reilly, A., FinnieAnsley, J., et al. (2023). Programming Is Hard - Or at Least It Used to Be: Educational Opportunities And Challenges of AI Code Generation. In Proceedings of the 54th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '23). Association for Computing Machinery.
- [6].AskPython. (n.d.). Create Minesweeper using Python. Diakses dari <https://www.askpython.com/python/examples/create-minesweeper-using-python>