Eksplorasi Implementasi Algoritma Optimal Permainan Dots and Boxes Menggunakan Teori Graf dan Algoritma Minimax

Noumisyifa Nabila Nareswari - 13523058[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]noumisyifa@gmail.com, 13523058@std.stei.itb.ac.id

Makalah ini membahas aplikasi strategi optimal untuk penyelesaian permainan strategi bernama Dots and Boxes dengan representasi graf. Permainan ini meskipun dengan aturan yang sederhana, kompleksitas strategi akan semakin bertambah seiring dengan bertambahnya ukuran papan. Penelitian ini memodelkan komponen-komponen permainan sebagai berbagia komponen graf. Titik direpresentasikan sebagai simpul pada graf, garis direpresentasikan sebagai sisi pada graf, sedangkan kotak direpresentasikan sebagai siklus pada graf. Algoritma pemilhan langkah menggunakan gabungan dari algoritma Minimax, Pemangkasan Alfa-Beta untuk optimasi, dan dasar strategi penyelesaian Dots and Boxes untuk menentukan nilai dari tiap kemungkinan langkah. Hasil uji pada papan permainan berukuran 3x3 menunjukkan kemampuan algoritma yang dapat memenfaatkan kelemahan langkah lawan dan memaksimalkan skor komputer. Hasil menunjukkan bahwa pendekatan berbasis teori grafdan pohon keputusan yaitu Algoritma Minimax menghasilkan strategi permainan yang optimal.

***Keywords*—Dots and Boxes, Graf, Algoritma Minimax, Pemangkasan Alfa-Beta**

I. PENDAHULUAN

Evolusi permainan berdasarkan konsep dan teori matematika telah mengalami pertumbuhan yang sangat pesat dalam beberapa dekade terakhir. Permainan-permainan tersebut menumbuhkan tren permainan yang tidak hanya memiliki fungsi rekreasi, namun juga fungsi edukasi. Prinsip permainan seperti ini mendorong manusia dari sejak dini untuk menumbuhkan kemampuan untuk berpikir kritis dan melakukan eksplorasi mandiri mendalam baik secara akademik, terutama matematika, dan juga teknologi. Tidak sedikit konsep matematika dan algoritma suatu permainan dikembangkan menjadi basis untuk pengembangan algoritma dan inovasi teknologi. Sebagai contoh, algoritma A* yang berawal dikembangkan untuk permainan dengan mekanika pencarian jalur kini diadaptasi untuk sistem navigasi dan optimasi rute pada GPS. Algoritma Minimax yang dikembangkan sebagai teknik permainan strategis seperti catur kini diimplementasikan dalam strategi trading finansial. Masih banyak lagi teori dan algoritma matematika suatu permainan yang menjadi solusi praktis untuk berbagai permasalahan di kehidupan sehari-hari manusia. Hal ini mendorong pertumbuhan yang pesat pada penelitian konsep dan algoritma penyelesaian permainan dengan pendekatan matematis.

Dots and Boxes adalah permainan *turn-based* klasik yang dimainkan oleh dua pemain di atas kertas dengan petak berukuran tertentu. Permainan ini pertama kali dipublikasikan pada abad ke-19 oleh matematikawan asal Prancis bernama Edouard Lucas. Permainan ini melibatkan tiga komponen utama, yaitu titik, garis, dan kotak.Pemain secara bergiliran menggambarkan garis yang menyambungkan dua titik dan secara kompetitif membentuk sebanyak mungkin kotak dari garis-garis tersebut, Pemain yang membentuk kotak terbanyak adalah pemenangnya. Permainan ini adalah permainan yang tampak sedrhana dan telah dimainkan oleh banyak sebatas untuk rekreasi. Namun, walau tampak sederhana, permainan ini memiliki kompleksitas cukup tinggi seiring bertambahnya ukuran grid permainan, meninggalkan ruang luas untuk eksplorasi strategi.

Berangkat dari latar belakang tersebut, makalah ini disusun dengan tujuan untuk melakukan eksplorasi lebih dalam strategi atau algoritma optimal dalam permainan Dots and Boxes menggunakan dasar teori graf. Analisis permainan ini cocok menggunakan pendekatan graf karena komponen pada permainan ini dapat diwakilkan dengan berbagai komponen penyusun graf. Hal ini memungkinkan analisis yang sistematis terhadap struktur permainan. Fokus utama penelitian adalah melakukan eksplorasi algoritma untuk berbagai ukuran grid dan posisi pemain menggunakan pemodelan sesuai teori graf.

II. STUDI LITERATUR

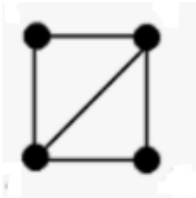
A. Teori Graf

A.1. Definisi Graf

Graf adalah suatu konsep di bidang Matematika yang digunakan untuk merepresentasikan beberapa objek diskrit dan hubungan antara objek tersebut. Graf dibangun oleh gabungan dari dua elemen utama, yaitu simpul (*vertices*) dan sisi (*edges*). Seminimalnya sebuah graf harus memiliki satu sisi sehingga sebuah graf juga perlu seminimalnya dua simpul karena untuk membentuk satu sisi diperlukan dua simpul.Sebuah graf G didefinisikan sebagai berikut

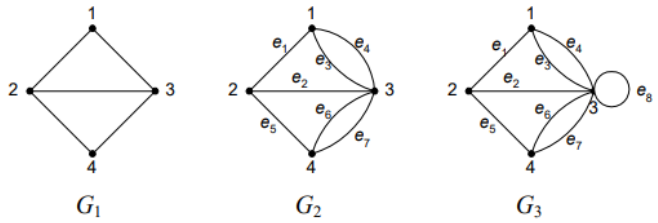
$$G = (V, E)$$
$$V = \{v1, v2, \ldots vn\}$$
$$E = \{e1, e2, \ldots en\}$$

Berikut adalah contoh graf dengan empat simpul dan lima sisi.



Gambar 1. Contoh Graf

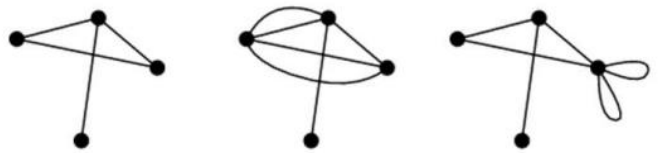
Graf dapat dikelompokkan melalui berbagai perspektif pengelompokkan. Pertama, berdasarkan ada tidaknya gelang atau sisi ganda di dalam graf, graf dapat dikelompokkan menjadi tiga jenis yaitu graf sederhana, graf ganda, dan graf semu.



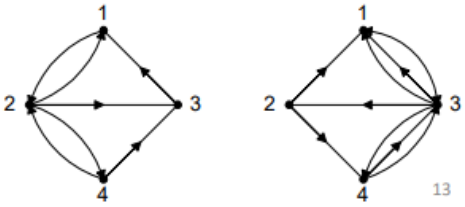
Gambar 2. Contoh graf sederhana (G1), graf ganda (G2), dan graf semu (G3)

Pada gambar di atas G1 merepresentasikan contoh dari graf sederhana, yaitu graf tanpa cincin/gelang (sisi dengan simpul asal dan simpul tujuan adalah simpul yang sama) dan juga tanpa ada dua sisi atau lebih dengan simpul asal dan simpul tujuan yang sama. Selanjutnya, G2 merepresentasikan contoh dari graf ganda yaitu graf dengan sisi-ganda. Sisi-ganda adalah beberapa sisi yang menghubungi dua buah simpul yang sama, dalam kasus G2, sisi e3 = (1,3) dan sisi e4 = (1,3) adalah sisi-ganda pada graf G2. Lalu, G3 merepresentasikan contoh dari graf semu yaitu graf dengan gelang (*loop*). Sisi e8 = (3,3) adalah gelang pada G3.

Selain pengelompokkan melalui ada tidaknya gelang atau sisi ganda di dalam graf, graf juga dapat dikelompokkan berdasarkan orientasi arah pada sisi. Berdasarkan pengelompokkan ini, graf terbagi menjadi dua jenis, yaitu graf tak-berarah dan graf berarah



Gambar 3. Contoh graf tak berarah

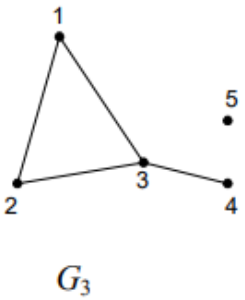


Gambar 4. Contoh graf berarah

Graf tak-berarah adalah graf dengan sisi yang tidak memiliki orientasi arah. Orientasi arah pada graf umumnya direpresentasikan dengan tanda panah. Lalu, graf berarah adalah graf yang setiap sisinya memiliki orientasi arah, seperti yang ditunjukkan pada gambar 4. Pada makalah ini, representasi graf yang akan digunakan adalah graf tak-berarah sederhana karena pada permainan Dots and Boxes arah tidak memengaruhi hasil dari permainan. Selain itu, mengapa digunakan representasi graf sederhana adalah karena dalam aturan permainan Dots and Boxes, tiap dua simpul tidak boleh direpresentasikan dengan lebih dari satu simpul.

A.2. Ketetanggan

Dalam graf, dua buah simpul dinyatakan bertetangaan (memiliki sifat ketetanggan) bila kedua simpul terhubung secara langsung oleh suatu sisi.



Gambar 5. Contoh graf G3

Pada graf G3 yang tergambar pada Gambar 5, simpul 4 bertetangaan dengan simpul 3 karena terhubung oleh suatu sisi. Namun, simpul 4 tidak bertetangaan dengan simpul 1, 2, dan 5. Simpul 5 pada G3 tidak bertetangaan dengan simpul apapun.

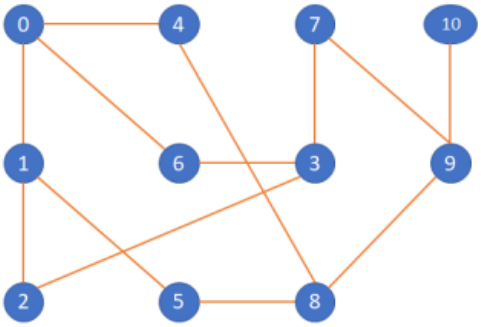
A.3. Derajat Simpul

Derajat sebuah simpul di suatu graf adalah angka yang menyatakan jumlah sisi yang bersisian (terhubung) dengan simpul tersebut. Tinjau Kembali graf G3 pada Gambar 5. Simpul 1 memiliki derajat sebesar 2 karena simpul 1 memiliki dua sisi yang terhubung ke dirinya. Simpul 4 memiliki derajat 1 karena simpul 4 hanya terhubung dengan satu sisi yang menghubungakan dirinya ke simpul 3. Lalu, simpul 5 memiliki derajat 0 karena tidak terhubung dengan sisi apapun. Konsep derajat ini cukup penting dalam berbagai teori graf karena

banyak teori dasar graf lainnya yang bergantung dengan nilai derajat simpul-simpul yang ada dalam suatu graf.

A.4. Lintasan dan Siklus Graf

Graf dengan lintasan yang memiliki Panjang ndari simpul awal ke simpul tujuan dalam graf G adalah barisan berselang-seling simpul-simpul dan sisi-sisi v0, e1, v1, e2, v2, ..., v(n –1), en , vn sedemikian sehingga e1 = (v0, v1), e2 = (v1, v2), ... , en = (v(n-1), vn) adalah sisi-sisi dari graf G.



Gambar 6. Contoh graf dengan panjang lintasan lima

Tinjau graf pada gambar 6, lintasan 0, 6, 3, 7, 9, 10 adalah lintasan dari simpul 0 ke 10. Lintasan tersebut melalui lima sisi yaitu sisi (0, 6), (6,3), (3,7), (7, 9), (9, 10), sehingga lintasan tersbut memiliki Panjang lima.

Lalu, siklus adalah lintasan yang berawal dan berakhir pada simpul yang sama. Tinjau Kembali graf pada Gambar 6. Lintasan yang melalui simpul 0, 4, 8, 5, 1, lalu kembali ke simpul 0 adalah contoh dari sebuah siklus. Panjang siklus adalah jumlah sisi yang terlibat dalam siklus tersebut. Pada contoh sebelumnya, siklus memiliki panjang bernilai lima. Pada permainan Dots and Boxes, satu *box* (kotak) dinyatakan sebagai satu siklus dengan panjang empat dan graf haruslah sebuah graf sederhana tak-berarah.

A.5. Graf Teratur

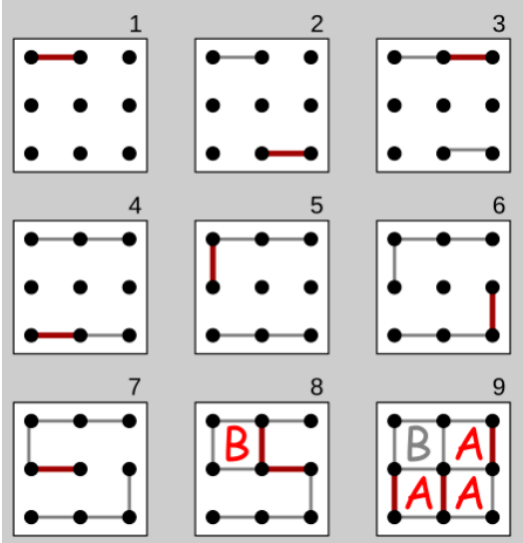
Graf teratur adalah graf yang setiap simpulnya memiliki derajat yang sama. Formula berikut menjelaskan hubungan antara jumlah simpul, jumlah sisi,dan derajat pada suatu graf teratur:

$$2n = re$$

Pada formula di atas, e merepresentasikan jumlah sisi, n merepresentasikan jumlah simpul, dan r merepresentasikan derajat tiap simpul yang nilainya sama pada tiap simpul.

B. Permainan Dots and Boxes

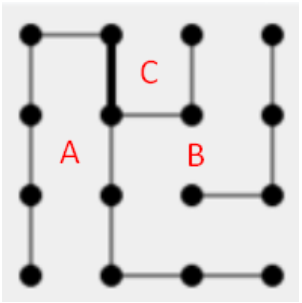
Permainan Dots and Boxes adalah permainan yang dibangun atas dasar matematika yang memiliki tiga komponen utama, yaitu titik, garis, dan boks. Permainan ini pertama kali dipublikasi pada abad ke-19 oleh seorang matematikawan asal Prancis bernama Édouard Lucas dengan nama La Pipopipette. Dots and Boxes adalah permainan dengan dua pemain membentuk garis pada grid titik-titik dengan ukuran nxn satu-satu secara bergiliran dengan tujuan untuk membentuk kotak sebanyak mungkin. Pemain yang membentuk kotak terbanyak adalah pemain yang menang. Permainan ini juga dikenal dengan nama Dots and Dashes, Game of Dots, dan Dot to Dot Grid.



Gambar 7. Contoh permainan Dots and Boxes dengan Grid 3 x 3

Permainan ini dimulai dengan grid berisikan n kali n titik yang belum terhubung. Selanjutnya, pemain secara bergiliran menambahkan garis yang menghubungkan dua titik dengan posisi tepat di samping secara vertikal atau horizontal, namun tidak boleh secara diagonal. Hubungan dua titik hanya boleh direpresentasikan oleh satu garis. Setiap seorang pemain membentuk satu kotak menggunakan garis yang baru saja digambar, pemain mendapatkan satu poin dan mendapatkan satu giliran tambahan. Permainan berakhir ketika grid sudah penuh, yaitu dimana sudah tidak ada lagi garis horizontal atau vertikal antara dua titik bertetangga yang dapat dibentuk. Saat permainan berakhir, pemain denganpoin terbanyak adalah pemenangnya. Dalam contoh permainan Dots and Boxes pada Gambar 7, pemain A adalah pemenangnya karena pemain A berhasil membentuk tiga kotak.

Permainan ini dibangun atas strategi yang melibatkan matematika, khususnya teori kombinatorika, graf, dan strategi pengambilan keputusan. Strategi utama dalam memenangkan permainan ini terletak pada kemampuan pemain untuk mengendalikan ranta, yaitu rangkaian jalur yang terbentuk dari garis-garis yang telah digambar.



Gambar 8. Contoh representasi rantai pada permainan Dots and Boxes

Pada Gambar 8, terdapat tiga rantai yang terbentuk, yaitu lintasan A, B, dan C. Rantai A memiliki Panjang tiga karena merepresentasikan tiga kotak,rantai B memiliki panjang lima karena merepresentasikan lima kotak, sedangkan rantai C memiliki panjang satu karena merepresentasikan satu kotak. Pemain yang dapat memanfaatkan kesempatan untuk membuka rantai Panjang dan dengan sekaligus mengambil semua skor (membentuk beberapa kotak secara sekaligus) sekaligus menghalangi lawan untuk membentuk rantai Panjang untuk skornya adalah pemenang yang kemungkinan besar untuk menang.

Waktu paling krusial untuk memenangkan permainan ini terletak akhir permainan. Ketika permainan mendekati akhirnya, pemain perlu dengan teliti memprediksi beberapa langkah ke depan yang mungkin di ambil oleh lawan dan memerhatikan bagaimana rantai giliran tambahan kemungkinan akan berjalan. Apabila kemungkinan besar suatu rantai panjang diambil oleh lawan, strategi terbaik yang dapat dilakukan oleh pemain adalah dengan memotong rantai tersebut menjadi rantai yang lebih kecil.

C. Algoritma Minimax

Algoritma Minimax adalah algoritma rekursif yang untuk pengambilan keputusan yang menggunakan prinsip *zero-sum*, yaitu prinsip yang menyatakan keuntungan satu pemain setimbang dengan kerugian pemain lain. Algoritma ini digunakan untuk memaksimalkan hasil yang didapatkan pemain yang sedang memainkan gilirannya sekaligus meminimalkan keuntungan lawan. Algoritma ini mengeksplorasi semua kemungkinan yang mungkin dilakukan pada giliran saat tertentu. Eksplorasi kemungkinan langkah dinyatakan dalam bentuk sebuah pohon keputusan dengan keadaan permaianan dinyatakan dalam simpul dan langkah yang akan diambil dinyatakan dengan cabang. Kelemahan dari algoritma ini adalah kompleksitas yang eksponensial yaitu $O(b^m)$ dengan b adalah dan m adalah tinggi pohon keputusan. Oleh karena itu diperlukan algoritma pemangkasan untuk menyeimbangkan kompleksitas yang tinggi .

D. Pemangkasan Alfa-Beta

Pemangkasan Alfa-Beta adalah algoritma untuk optimasi yang dalam makalah ini akan diterapkan untuk algoritma Minimax. Algoritma ini memiliki fungsi untuk mengurangi jumlah simpul yang dievaluasi pada pohon keputusan dengan mengevaluasi kemungkinan apa yang memang perlu dieksplorasi dan yang tidak. Algoritmaini menggunakan dua parameter utamayaitu Alfa yang merepresentasikan nilai maksimum yang dapat dijamin oleh pemain dan juga Beta yang merepresentasikan nilai minimum yang dapat dijamin oleh pemain lawan. Algoritma ini akan memangkas sebuah simpul dan seluruh turunannya pada pohon keputusan jika simpul tertentu tidak menghasilkan suatu hasil yang lebih baik dari nilai Alfa atau Beta saat ini.

III. HASIL DAN PEMBAHASAN

A. Representasi Permainan dalam Bentuk Graf

Seperti yang sudah dijelaskan pada landasan teori, permainan Dots and Boxes ini dibentuk oleh tiga komponen utama, yaitu titik, garis, dan kotak. Untuk menganalisis mengggunakan pendekatan graf, ketiganya perlu ditentukan persamaannya dalam komponen graf. Untuk komponen titik, dalam graf akan direpresentasikan sebagai simpul (*vertex*). Pemain akan memasukkan ukuran dari papan yang diinginkan, kemudian program akan membentuk sebuah graf dengan jumlah simpul sebanyak (n+1) x (n+1) dengan n adalah jumlah kotak secara horizontal dan vertikal yang diinginkan pemain.

```
def __init__(self, size: int):
        self.size = size
        # Create graph where vertices are
dots and edges are lines
        self.graph = nx.Graph()
        self.boxes = np.zeros((size,
size))  # Completed boxes
        self.scores = {1: 0, 2: 0}
        self.current_player = 1

        # Initialize graph vertices (dots)
        for i in range(size + 1):
            for j in range(size + 1):
                self.graph.add_node((i, j))

        # Track which edges belong to which
potential boxes
        self.edge_to_boxes: Dict[Tuple,
List[Tuple]] = {}
        self._initialize_potential_boxes()
```

Selanjutnya garis direpresentasikan sebagai sisi dalam suatu graf (*edge*) dengan bentuk (v1, v2) dengan v1 dan v2 adalah titik yang bersebelahan, menyatakan titik apa yang terhubung oleh garis tersebut. Lalu, sebuah kotak (*box*) direpresentasikan oleh empat sisi dengan aturan berikut

$$v1 = (i, j) \quad v2 = (i, j+1)$$
$$v3 = (i+1, j) \quad v4 = (i+1, j+1)$$
$$box = [(v1, v2), (v2, v4), (v4, v3), (v3, v1)]$$

Selanjutnya Permainan berakhir ketika papan permainan sudah penuh, yaitu ketika papan sudah membentuk sebuah graf teratur dengan jumlah simpul $(n+1)^2$ dengan semua titik berderajat empat. Atau dapat dinyatakan juga sebagai jumlah sisi yang terbentuk adalah sebagai berikut

$$\frac{2n}{r} = garis$$

Namun, dalam program yang disusun, permainan berakhir ketika semua kotak sudah dibuat.yaitu ketika status/nilai dari semua kotak yang disimpan bernilai 0.

```
def is_game_over(self) -> bool:
        return np.all(self.boxes != 0)
```

B. Algoritma Optimasi Kemenangan

Penyusunan algoritma ini menggunakan Bahasa Python dan beberapa *library* network untuk representasi graf, numpy untuk beberapa operasi perhitungan, dan juga typing untuk membentuk struktur data tuple, list, dan dictionary. Semua kotak dan informasi titik pembentuknya disimpan dalam self.edge_to_boxes dalam bentuk dictionary yang berisikan tuple dan list of tuple yang merupakan representasi list of garis. Dictionary ini berada dalam kelas DotsAndBoxesGraph.

Selanjutnya, untuk menentukan keputusan atau langkah terbaik, algoritma ini menggunakan algoritma Minimax. Algoritma rekursif khusus untuk *decision-making* sebuah permainan dengan dua pemain. Algoritma ini akan mensimulasikan siapa yang pemain manusia dan siapa yang pemain berupa komputer. Kunci algoritma ini adalah *recursive depth-search* yang akan berjalan sebanyak kedalaman tertentu. Pada setiap kedalamannya, algoritma akan memaksimalkan hasil untuk pemain komputer dan meminimalkan skor pemain manusia hingga sudah mencapai kedalaman maksimal yang ditentukan atau ketika permainan sudah berakhir. Agar algoritma ini bekerja dengan lebih kompleksitas waktu dan memori yang lebih baik, algoritma ini dioptimisasi dengan algoritma Pemangkasan Alfa-Beta. Nilai Alfa mewakilkan nilai maksimal dan Beta mewakilkan nilai minimum. Algoritma ini akan memangkas kemungkinan yang harus dicek oleh algoritma Minimax dan semua anak kemungkinannya ketika nilai Beta lebih kecil atau sama dengan nilai Alfa. Hal ini menandakan bahwa kemungkinan itu tidak signifikan dan tidak perlu ditelusuri lebih lanjut.

```
FUNCTION minimax_graph(game, depth, alpha,
beta, maximizing_player)
    IF depth == 0 OR game.is_game_over()
        RETURN game score difference, None

    available_moves ←
game.get_available_moves()
    SORT available_moves BY chain value
(descending for maximizing, ascending for
minimizing)

    IF maximizing_player
        max_eval ← -∞
        FOR each move IN available_moves DO
            APPLY move TO game copy
            eval_score, _ ←
minimax_graph(game copy, depth - 1, alpha,
beta, False)
```

```
minimax_graph(game copy, depth - 1, alpha,
beta, False)
            max_eval ← MAX(max_eval,
eval_score)
            alpha ← MAX(alpha, eval_score)
            IF beta ≤ alpha THEN BREAK
        RETURN max_eval, best_move

    ELSE
        min_eval ← +∞
        FOR each move IN available_moves DO
            APPLY move TO game copy
            eval_score, _ ←
minimax_graph(game copy, depth - 1, alpha,
beta, True)
            min_eval ← MIN(min_eval,
eval_score)
            beta ← MIN(beta, eval_score)
            IF beta ≤ alpha THEN BREAK
        RETURN min_eval, best_move
END FUNCTION
```

Selanjutnya, untuk menilai setiap kemungkinan langkah, digunakan fungsi evaluate_chain_value yang akan mengembalikan nilai dari 0 hingga 1 seberapa direkomendasikan langkah tersbut berdasarkan rantai yang dipengaruhi oleh garis yang kemungkinan akan ditambahkan.

Fungsi ini pertama-tama memeriksa apakah komputer pemain pertama atau kedua. Lalu, fungsi ini memeriksa kotak-kotak yang terpengaruhi oleh garis baru menggunakan fungsi pemetaan edge_to_boxes. Untuk setiap kotak yang belum terbentuk lengkap, fungsi akan secara rekursif mencari rantai

kotak yang saling terhubung dengan fungsi find_chain.

```
def find_chain(box_coord, visited):
            nonlocal chain_length
            i, j = box_coord
            if (i, j) in visited or
self.boxes[i, j] != 0:
                return

            visited.add((i, j))
            edges = self._get_box_edges(i,
j)
            completed_edges = sum(1 for e in
edges if self.graph.has_edge(*e))
```

```
            if completed_edges == 3:
                chain_length += 1
                # Check adjacent boxes for
chain continuation
                for next_box in
self._get_adjacent_boxes(i, j):
                    find_chain(next_box,
visited)
```

Lalu algoritma mengecek untuk setiap kotak yang mungkin terpengaruhi oleh garis baru merupakan bagian dari rantai yang belum dievaluasi, jika iya, fungsi akan menjalankan Kembali fungsi find_chain. Kemudian algoritma mulai mengecek nilai bobot suatu langkah berdasarkan apakah computer pemain pertama atau kedua. Apabila komputer permain pertama, rantai dengan panjang genap akan dianggap membawa kerugian sedangkan rantai dengan panjang ganjil dianggap menguntungkan, hal ini berlaku sebaliknya bila komputer bergerak sebagai pemain kedua. Algoritma ini juga berusaha untuk membuat kesempatan memaksa lawan (manusia) untuk membuat jumlah rantai sesuai dengan jumlah rantai apa yang menguntungkan bagi komputer.

C. Hasil Uji

C.1. Hasil Uji Papan 3 x 3 Komputer Sebagai Pemain Kedua

Berikut adalah hasil uji algoritma optimasi kemenangan permainan Dots and Boxes dalam papan dengan kotak 3 x 3 dan komputer sebagai pemain kedua.

```
Dots and Boxes - Graph Theory Implementation
========================================
Enter grid size (2-3 recommended for reasonable computation
 time): 3
Do you want to play first (1) or second (2)? 1

Game Instructions:
Enter moves as coordinates of two dots to connect
Example: '0,0 0,1' connects dots at (0,0) and (0,1)
• • • •

• • • •

• • • •

• • • •

Score - Player 1: 0, Player 2: 0
```

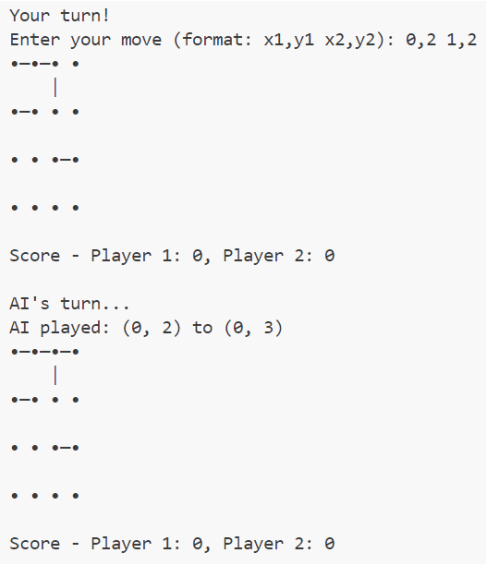
Gambar 8. Permainan dimulai dengan manusia sebagai pemain urutan pertama

```
Your turn!
Enter your move (format: x1,y1 x2,y2): 2,2 2,3
• • • •

• • • •

• • •–•

• • • •

Score - Player 1: 0, Player 2: 0

AI's turn...
AI played: (0, 0) to (0, 1)
•–• • •

• • • •

• • •–•

• • • •

Score - Player 1: 0, Player 2: 0
```

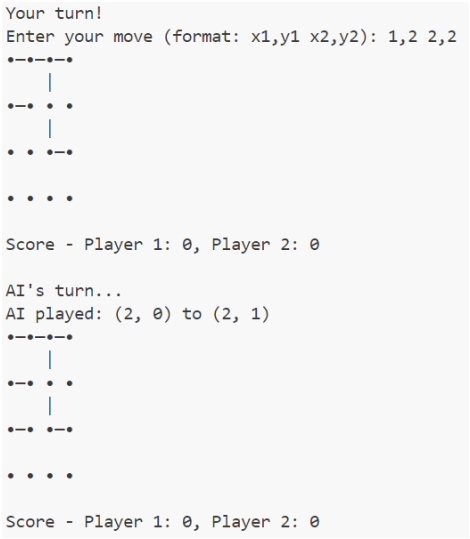
Gambar 8. Permainan pada babak pertama

```
Your turn!
Enter your move (format: x1,y1 x2,y2): 1,0 1,1
•–• • •

•–• • •

• • •–•

• • • •

Score - Player 1: 0, Player 2: 0

AI's turn...
AI played: (0, 1) to (0, 2)
•–•–• •

•–• • •

• • •–•

• • • •

Score - Player 1: 0, Player 2: 0
```

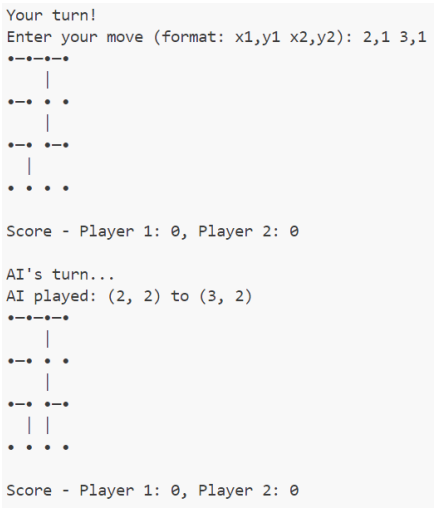
Gambar 8. Permainan pada babak kedua



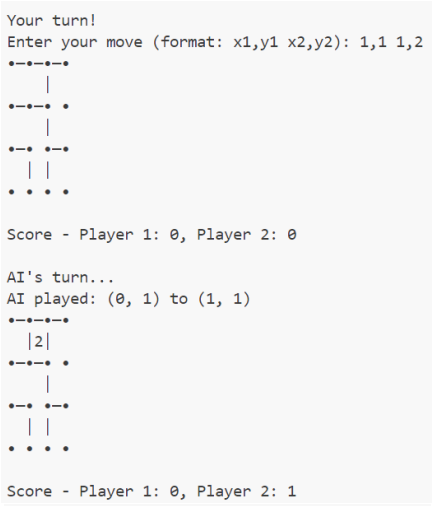
Gambar 8. Permainan pada babak ketiga

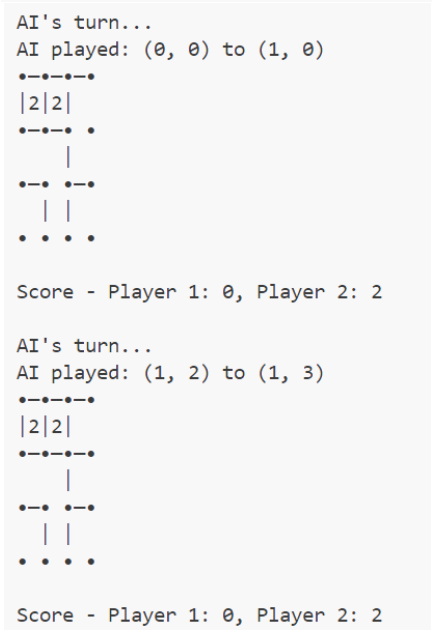


Gambar 8. Permainan pada babak keempat

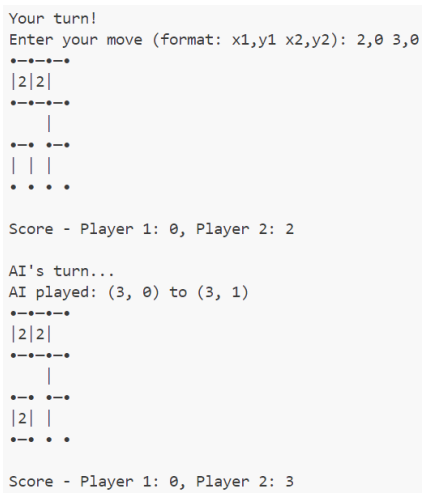


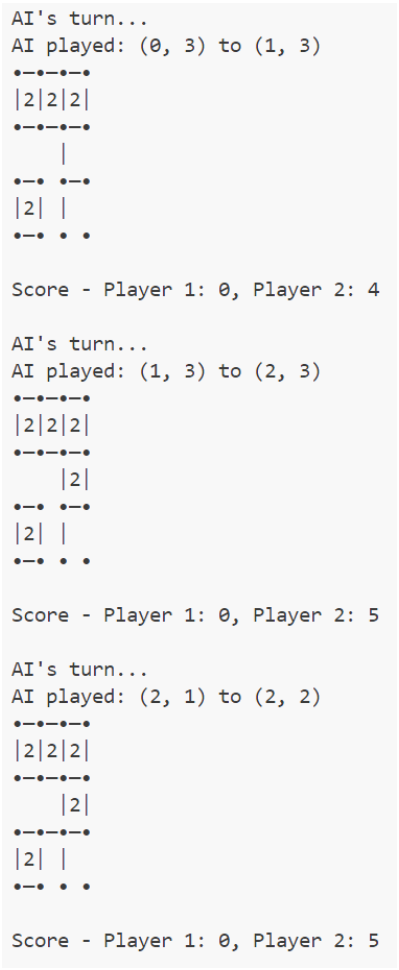
Gambar 8. Permainan pada babak kelima



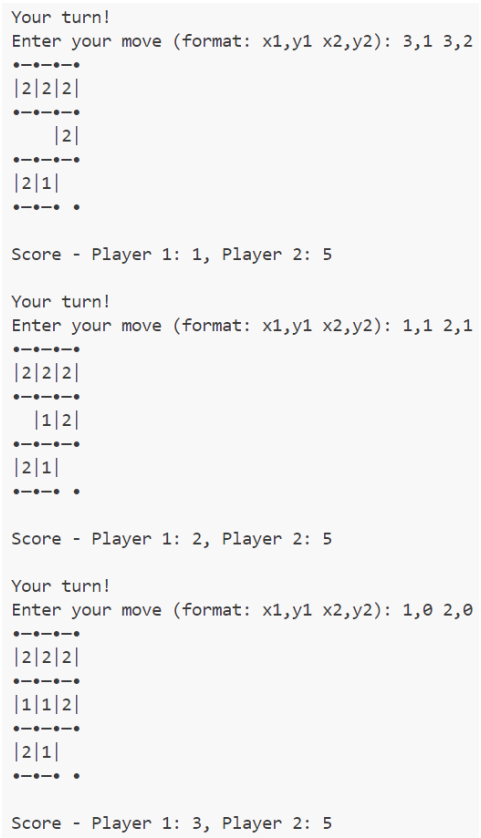


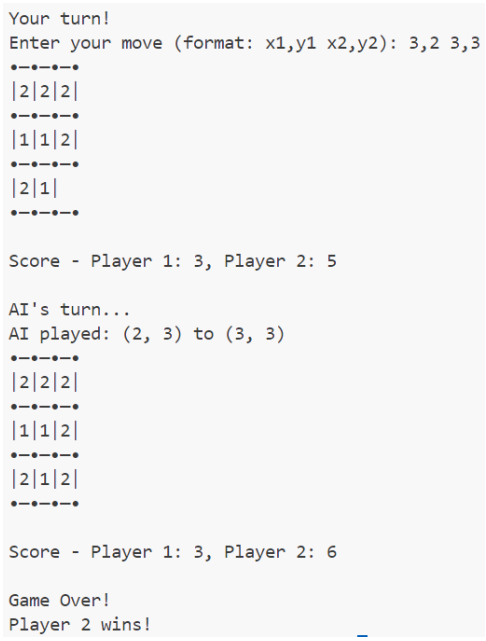
Gambar 9. Permainan pada babak keenam





Gambar 9. Permainan pada babak ketujuh





Gambar 9. Permainan pada babak terakhir

Permainan berakhir dengan computer (AI) menang terhadap pemain manusia dengan skor komputer mendapatkan skor sebanyak enam dan manusia mendapatkan skor sebanyak tiga. Hal ini menunjukkan bahwa program berhasil memprediksi dan memaksimalkan skor yang dapat dihasilkan berdasarkan posisi garis yang dibentuk oleh lawan.

Program menunjukkan kemampuan algoritma untuk mengeksploitasi kelemahan lawan dan memanfaatkan sebaik mungkin. Ketika pemain manusia membuka kesempatan berupa rantai genap, algoritma dengan cepat menebak langkah paling optimal untuk memanfaatkannya sekaligus mengkalkulasi bagaimana mempertahankan posisi skkornya agar tetap aman.

IV. KESIMPULAN

Dapat disimpulkan bahwa pendekatan representasi graf untuk mengimplementasikan algoritma yang optimal untuk memenangkan permainan Dots and Boxes sangatlah relevan dan optimal. Menggunakan gabungan representasi papan permainan dengan graf, algoritma Minimax sebagai pohon keputusan, ditambah dengan optimisasi algoritma Pemangskasan Alfa-Beta, program dapat menentukan langkah terbaik dalam permainan ini dengan efisien. Uji coba pada papan kotak 3 x 3 dilakukan, hasil uji coba menunjukkan bahwa algoritma (sisi computer) dapat memanfaatkan kelemahan giliran yang dilakukan oleh pemain manusia dan memenangkan pemain dengan strategi optimal. Hasil ini menunjukkan adanya potensi dalam penerapan teori graf ke berbagai permainan berbasis strategi lainnya. Di masa depan, penelitian mengenai hal ini dapat dilakukan dengan kasus uji coba yang lebih variative. Akibat keterbatasan jumlah halaman untuk pengumplan tugas ini, tidak memungkinkan untu kpenulis memaparkan hasil uji coba yang lebih ekstensif.

VII. UCAPAN TERIMA KASIH

Pertama-tama, saya ingin menyampaikan terima kasih yang sebesar-besarnya kepada bapak Rila Mandala, dosen untuk mata

kuliah Matematiak Diskrit IF1220 K-02. Berkat bimbingan, ketulusan, keahlian, dan dedikasi Beliau dalam mengajar selama satusemester ini, saya berhasil menyelesaikan proyek ini dan mendapatkan banyak ilmu baru melampaui dari ekspektasi saya ketika pertama kali masuk jurusan Teknik Informatika ini.

Saya juga ingin mengucapkan terima kasih yang paling tulus kepada orang tua saya yang berkat dorongan moral dan doa mereka yang tanpa henti saya selalu mendapatkan semangat kuat untuk menuntut ilmu dengan giat. Hal ini menjadi dorongan mental yang tak tergantikan bagi saya.

Saya juga ingin mengucapkan terima kasih yang special untuk semua teman-teman Teknik Informatika ITB Angkatan 2023, terutama Diyah, Anella, dan Lala. Berkat mereka saya dapatmenjalani perkuliahan yang terkenal dengan beban akademik yang berat ini dengan lebih menyenangkan.

REFERENCES

[1] Munir, R. (2024). *Teori Graf Bagian 1*. Diakses dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf pada 8 Januari 2025.

[2] Nilsson, N. J. (1983). *Artificial intelligence prepares for 2001*. *Artificial Intelligence*, *21*(1-2), 31-34. Diakses dari https://www.sciencedirect.com/science/article/abs/pii/0004370283900012 pada 8 Januari 2025.

[3] Pearl, J. (1983). *Heuristics: Intelligent search strategies for computer problem solving*. *Artificial Intelligence*, *21*(3), 165-191. Diakses dari https://www.sciencedirect.com/science/article/abs/pii/S0004370283800204 pada 8 Januari 2025.

[4] Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2018). *Deterministic policy gradients: Theory and application*. *Artificial Intelligence*, *287*, 103-123. Diakses dari https://www.sciencedirect.com/science/article/pii/S0004370218303485 pada 8 Januari 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 27 Desember 2024

Noumisyifa Nabila Nareswari
13523058