



## Algoritma Minimax untuk Game Tic Tac Toe yang Menantang

Husni Thamrin<sup>1</sup>, Ramadhana Wahid Aji Pamungkas<sup>1</sup>

Email husni.thamrin@ums.ac.id, l200190190@student.ums.ac.id

<sup>1</sup>Universitas Muhammadiyah Surakarta

---

### Informasi Artikel

Diterima : 23 Mei 2023

Direview : 13 Jun 2023

Disetujui : 30 Jun 2023

---

### Kata Kunci

minimax, kecerdasan buatan, tic tac toe, permainan papan

---

### Abstrak

Algoritma minimax adalah algoritma yang digunakan untuk menentukan pilihan dengan memperkecil kemungkinan kehilangan nilai maksimal. Dalam konteks game, algoritma minimax meminimalkan kemungkinan kekalahan dan memaksimalkan kemungkinan kemenangan. Algoritma minimax dapat diterapkan dalam game berbasis papan permainan seperti tic tac toe dan catur. Penerapan algoritma minimax membuat komputer tidak terkalahkan sedangkan penggunaan metode random membuat komputer memilih langkah secara tidak logis dan dengan mudah dikalahkan. Untuk mendapatkan game yang menantang di mana pemain dapat memperoleh kemenangan, hasil seri, dan kekalahan, maka kedua metode dikombinasikan. Algoritma minimax memperhitungkan semua kemungkinan langkah pemain dan komputer sehingga waktu eksekusinya bergantung jumlah kotak kosong yang ada pada papan tic tac toe. Kompleksitas algoritma bersifat logaritmis atau  $O(xn)$ . Untuk jumlah kotak kosong lebih dari 9 maka waktu eksekusi berada pada orde detik dan ketika kotak kosong berjumlah 13 waktu eksekusi berada pada orde jam. Karena itu algoritma minimax dapat diterapkan pada papan tic tac toe 3 x 3 namun tidak dapat digunakan pada papan 4 x 4. Untuk diterapkan pada papan 4 x 4 diperlukan perbaikan pada algoritma atau perubahan dalam penerapan algoritma.

---

### Keywords

Minimax, artificial intelligence, tic tac toe, board game

---

### Abstract

*The minimax algorithm is a decision-making tool that minimizes the risk of losing the maximum value. The minimax algorithm maximizes the probability of winning and minimizing the probability of losing a game. Board games like tic tac toe and chess may implement the minimax algorithm. The random technique lets the computer pick illogical steps and is easily defeated whereas the implementation of the minimax algorithm makes the computer invincible. This study combines the two techniques to create a challenging game where players may win, draw, and lose. The minimax algorithm takes into account all possible moves by the player and the computer so that the execution time depends on the number of empty squares on the tic-tac-toe board. The algorithm's complexity is logarithmic, or  $O(xn)$ . If there are more than 9 empty boxes, the execution time is on the order of seconds, and when there are 13 empty boxes, the execution time is on the order of hours. Therefore the minimax algorithm is applicable to a 3 x 3 tic tac toe board but impractical to a 4 x 4 board. To make it work on a 4 x 4 board, modification of the algorithm is necessary.*

## **A. Pendahuluan**

Game di era modern ini telah menjadi sangat populer dan terus berkembang sebagai dampak perkembangan teknologi yang pesat dan semakin canggih. Saat ini terdapat banyak jenis game yang dapat dimainkan, seperti game pertarungan, game strategi, game casual, dan game papan [1]. Game telah banyak berkembang dan dapat dimainkan di mana saja karena merebaknya teknologi handphone, laptop, console dan perangkat permainan lainnya [2].

Game dapat menjadi teman di kala sendirian dan menjadi obat penyegar otak untuk menghindari stres. Game memiliki dampak positif dari meningkatkan kemampuan berpikir, memberi motivasi, dan meningkatkan hubungan sosial [3], [4]. Perkembangan game telah memberikan lapangan pekerjaan bagi pengembang game dan penyedia perangkat bermain. Game memiliki manfaat namun dapat memberi dampak negatif bagi orang yang kecanduan bermain game. Bahaya yang dialami pecandu game adalah rasa malas untuk mengerjakan kegiatan lain, merusak mata karena terlalu sering menatap layar komputer [5]–[7]. Bahaya bermain game dapat dihindari dengan membuat jadwal bermain game sehingga ada keteraturan serta menumbuhkan komitmen untuk mendapatkan manfaat dari bermain game [8].

Game yang menarik dan menyenangkan adalah yang dapat memberikan tantangan sesuai dengan kemampuan pemain. Game yang terlalu mudah akan membuat cepat bosan sedangkan game yang terlalu sulit akan membuat frustrasi pemain, sehingga pemain tidak ingin memainkan game itu lagi. Game yang disukai harus memiliki kemungkinan untuk dimenangkan oleh pemain dan memiliki kemungkinan yang sama untuk kalah atau menang. Pemain tidak boleh menunggu terlalu lama untuk mendapat giliran [9], [10]. Akan lebih disukai apabila game memiliki banyak level sehingga tantangan bermain terus meningkat.

Pada game papan seperti catur dan tic tac toe, telah dikembangkan algoritma minimax yang bekerja dengan mengevaluasi keadaan permainan pada setiap langkah dan menggunakan struktur pohon untuk merepresentasikan semua kemungkinan langkah yang dapat diambil. Dalam pohon permainan, algoritma mencari jalur yang mengoptimalkan keuntungan maksimal dan kerugian minimal bagi pemain yang sedang bermain, dengan asumsi bahwa pemain lawan juga bermain secara optimal. Dengan mempertimbangkan nilai maksimal dan minimal dari setiap kemungkinan langkah, algoritma minimax dapat memilih langkah terbaik yang akan diambil untuk mencapai hasil yang paling menguntungkan dalam permainan [11], [12].

Tulisan ini mendeskripsikan penelitian yang mencoba mengembangkan game tic tac toe yang menantang (*challenging*) baik pada papan tic tac toe 3 x 3 maupun 4 x 4. Hasil penelitian menunjukkan perlunya penambahan mode random ketika komputer memilih langkah, sehingga langkah komputer dapat berjalan lebih cepat dan komputer masih dapat dikalahkan oleh pemain.

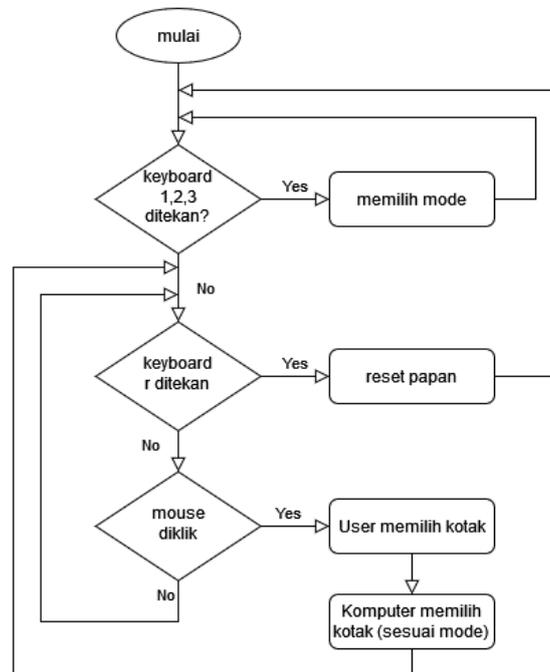
## **B. Metode Penelitian**

### **2.1 Pengembangan Game**

Penelitian dimulai dengan pengembangan game tic tac toe 3 x 3. Game dikembangkan dengan tiga mode yaitu permainan antara manusia dengan manusia (mode 0), manusia dengan komputer yang memilih langkah dengan algoritma minimax (mode 1), dan manusia dengan komputer yang memilih langkah secara random (mode 2). Pengamatan dilakukan ketika manusia bermain melawan komputer.

Selanjutnya dikembangkan game tic tac toe 4 x 4. Game pada awalnya juga dikembangkan dengan tiga mode seperti pada papan 3 x 3. Namun karena waktu eksekusi algoritma minimax terlalu lama, maka implementasi algoritma minimax tidak murni diterapkan pada papan 4 x 4. Peneliti menerapkan mode 3 yaitu memilih langkah secara random di awal permainan dan menerapkan algoritma minimax di langkah selanjutnya.

Game dibangun dengan bahasa pemrograman Python dengan memanfaatkan modul pygame. Adapun alur jalannya game diperlihatkan pada Gambar 1. Pada awal permainan, pemain dapat memilih mode dengan menekan tombol 0, 1, 2, atau 3. Selanjutnya pemain mengklik sebuah kotak kosong yang akan terisi dengan sebuah marker yaitu tanda silang. Jika permainan melawan komputer, maka komputer akan memilih kotak kosong sesuai mode permainan, yaitu secara random atau dengan algoritma minimax. Permainan akan berakhir jika salah satu pemain atau komputer memenangkan pertandingan atau semua kotak sudah terisi marker, yang artinya permainan berakhir seri. Pemain atau komputer dinyatakan menang jika dapat mengisi kotak dengan marker membentuk garis 3 atau 4 kotak berturut-turut.



**Gambar 1.** Alur program game tic tac toe

Ketika komputer memilih kotak kosong untuk ditempati marker, secara prinsip digunakan algoritma seperti yang diperlihatkan pada Gambar 2. Algoritma bersifat

rekursif, yaitu dengan memeriksa semua kemungkinan meletakkan marker pada kotak kosong. Algoritma minimax yang diterapkan terlihat pada Gambar 3. Langkah terbaik ditentukan menggunakan skor untuk setiap pemilihan langkah. Langkah yang mengarah kepada kemenangan diberi nilai positif dan langkah yang mengarah kepada kekalahan diberi nilai negatif.

```
function findBestMove(board):
    bestMove = NULL
    for each move in board:
        if current move is better than bestMove
            bestMove = current move
    return bestMove
```

**Gambar 2.** Prinsip menentukan langkah terbaik

```
function minimax(board, depth, isMaximizingPlayer):
    if current board state is a terminal state:
        return value of the board

    if isMaximizingPlayer:
        bestVal = -INFINITY
        for each move in board:
            value = minimax(board, depth+1, false)
            bestVal = max( bestVal, value)
        return bestVal

    else :
        bestVal = +INFINITY
        for each move in board:
            value = minimax(board, depth+1, true)
            bestVal = min( bestVal, value)
        return bestVal
```

**Gambar 3.** Algoritma minimax

## 2.2 Pengambilan data

Penelitian mengambil data berupa waktu yang diperlukan komputer dalam menghitung satu langkah yaitu mengisi tanda silang atau lingkaran ketika komputer mendapat giliran jalan. Pengambilan data dilakukan ketika komputer memilih langkah dengan algoritma minimax, dan tidak perlu dilakukan ketika komputer memilih langkah secara random yang diasumsikan akan terjadi secara sangat cepat. Data diambil baik untuk permainan papan 3 x 3 maupun 4 x 4. Dapat diduga bahwa ketika kotak kosong masih banyak akan diperlukan waktu yang lebih lama dalam menentukan langkah terbaik. Data diambil sebanyak 5 kali untuk setiap kotak kosong yang berbeda.

Waktu eksekusi program komputer dipengaruhi oleh hardware yang digunakan. Pada penelitian ini, komputer yang digunakan untuk melakukan pengujian memiliki spesifikasi sebagai berikut.

- *Prosesor* : 12<sup>th</sup> Gen Intel® Core™ i5-12500H 2.50 GHz
- *RAM* : 16 GB
- *Sistem Operasi* : *Windows 11 Home Single Language*
- *Storage* : 512 GB M.2 NVMe™ Pcle 3.0 SSD
- *Display* : 14-inch, 2.8K (2880 x 1800) OLED
- *Graphics* : Intel® Iris Xe Graphics
- *Memory* : 8 GB DDR4 on board + 8 GB DDR4 SO-DIMM

### 2.3 Pengujian

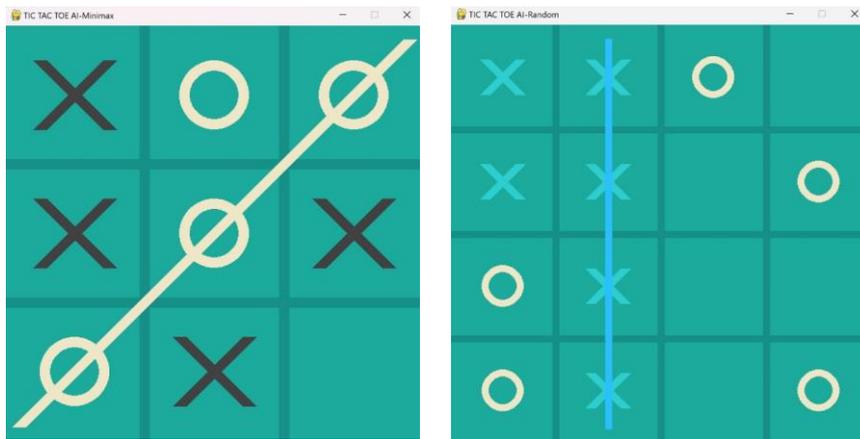
Game yang dibangun selanjutnya diuji dengan dimainkan oleh pemain manusia. Lima orang diminta memainkan game secara berulang. Papan 3 x 3 dengan mode random dimainkan 3 kali, papan 3 x 3 dengan mode minimax dimainkan sebanyak 1 kali, dan papan 4 x 4 dimainkan 2 kali.

Mode permainan pada papan 4 x 4 adalah campuran (mixed) di mana komputer di awal permainan memilih langkah secara random dan selanjutnya memilih langkah dengan algoritma minimax. Campuran mode ini diterapkan karena pada papan 3 x 3 mode random, tercatat bahwa pemain manusia selalu menang. Selanjutnya pada papan 3 x 3 mode minimax, pemain manusia tidak pernah menang, yaitu manusia hanya bisa kalah atau seri. Fakta yang sama diasumsikan akan terjadi pada papan 4 x 4, sehingga tidak dilakukan pengujian permainan dengan mode random dan minimax pada papan 4 x 4. Mode campuran diterapkan yaitu komputer memilih langkah secara random pada tiga langkah pertama dan memilih menggunakan algoritma minimax pada langkah selanjutnya.

### C. Hasil dan Pembahasan

Game Tic Tac Toe 3 x 3 dan 4 x 4 berhasil dibuat sesuai rancangan yaitu dibangun dengan bahasa Python dengan modul pygame. Game memiliki 3 mode permainan yaitu manusia lawan manusia, manusia lawan komputer yang memilih langkah secara *random*, dan manusia lawan komputer yang memilih langkah menggunakan algoritma Minimax.

Secara default game berjalan pada mode minimax dengan marker silang untuk pemain manusia dan marker lingkaran untuk komputer. Beberapa tombol dapat ditekan untuk merubah mode, dan melakukan reset papan permainan. Penekanan tombol 0 merubah mode permainan menjadi manusia lawan manusia, penekanan tombol 1 merubah mode menjadi minimax, penekanan tombol 2 merubah mode menjadi random, dan penekanan tombol 3 merubah mode menjadi campuran. Penekanan tombol R akan mereset papan permainan. Kemenangan diperoleh jika salah satu pemain dapat meletakkan marker dengan membentuk garis horisontal, vertikal atau diagonal. Contoh posisi menang diperlihatkan pada Gambar 4, di mana gambar kiri menunjukkan komputer menang pada permainan papan 3 x 3, sedangkan gambar kanan memperlihatkan pemain manusia menang pada papan 4 x 4.



**Gambar 4.** Tampilan papan Tic Tac Toe 3 x 3 dan 4 x4;  
Kiri: komputer menang; Kanan: pemain manusia menang

### 3.1 Implementasi algoritma minimax

Kode Python yang mengimplementasikan algoritma minimax terlihat pada Gambar 5. Fungsi mengembalikan nilai 1 jika status papan menyatakan bahwa pemain manusia menang, mengembalikan -1 jika komputer menang dan mengembalikan 0 jika tidak ada yang menang (seri). Kode program yang menentukan status papan terlihat pada baris 119-132. Jika status papan belum sampai kondisi akhir, maka dilakukan proses minimax untuk setiap kemungkinan mengisi kotak kosong oleh komputer dan oleh pemain manusia secara bergantian (lihat kode program pada baris 134-162).

```

114 # --- MINIMAX ---
115
116 def minimax(self, board, maximizing):
117     # terminal case
118     case = board.final_state()
119
120     # player 1 wins
121     if case == 1:
122         return 1, None # eval, move
123
124     # player 2 wins
125     if case == 2:
126         return -1, None
127
128     # draw
129     elif board.isfull():
130         return 0, None
131
132     if maximizing:
133         max_eval = -100
134         best_move = None
135         empty_sqrs = board.get_empty_sqrs()
136
137         for (row, col) in empty_sqrs:
138             temp_board = copy.deepcopy(board)
139             temp_board.mark_sqr(row, col, 1)
140             eval = self.minimax(temp_board, False)[0]
141             if eval > max_eval:
142                 max_eval = eval
143                 best_move = (row, col)
144         return max_eval, best_move
145
146     elif not maximizing:
147         min_eval = 100
148         best_move = None
149         empty_sqrs = board.get_empty_sqrs()
150
151         for (row, col) in empty_sqrs:
152             temp_board = copy.deepcopy(board)
153             temp_board.mark_sqr(row, col, self.player)
154             eval = self.minimax(temp_board, True)[0]
155             if eval < min_eval:
156                 min_eval = eval
157                 best_move = (row, col)
158         return min_eval, best_move
159
160
161
162
163

```

**Gambar 5.** Kode Python untuk implementasi algoritma Minimax

### 3.2 Waktu eksekusi fungsi minimax

Penentuan waktu dalam menjalankan satu langkah *minimax* dilakukan dengan menyisipkan fungsi `time.perf_counter()` sebelum dan sesudah menjalankan fungsi memilih langkah sebagaimana terlihat pada baris 177 dan 179 pada kode program (lihat Gambar 6). Fungsi `perf_counter` membantu menghitung waktu yang diperlukan untuk mengeksekusi program dengan memperhatikan faktor-faktor hardware dan sistem operasi yang berjalan. Untuk mengurangi efek acak dari kondisi di luar eksekusi program, pengamatan dilakukan berulang sebanyak 5 kali.

```

163
164 # --- MAIN EVAL ---
165
166 def eval(self, main_board):
167     if self.level == 0:
168         # random choice
169         evalm = 'random'
170         move = self.rnd(main_board)
171
172     elif self.level == 1:
173         # minimax algo choice
174         # menghitung waktu eksekusi
175         start_time = time.perf_counter()
176         evalm, move = self.minimax(main_board, False)
177         end_time = time.perf_counter()
178
179         total_time = end_time - start_time
180

```

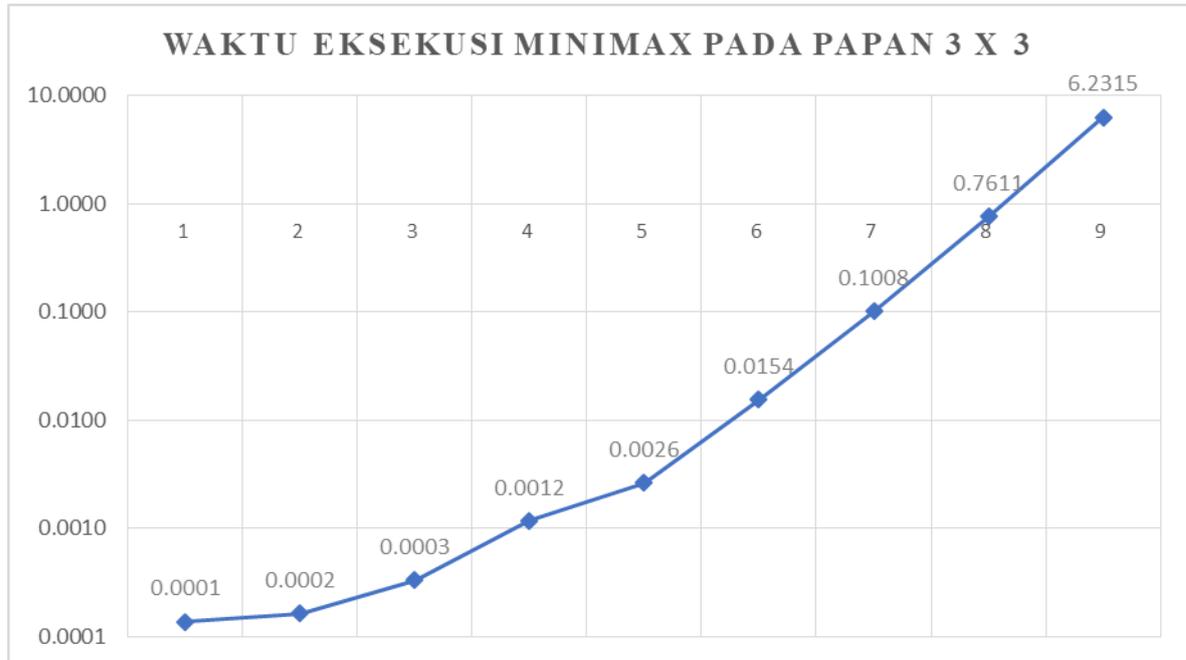
**Gambar 6.** Kode program untuk menentukan waktu eksekusi minimax

**Tabel 1.** Waktu eksekusi minimax pada papan 3 x 3

Jumlah Kotak Kosong	Waktu untuk tiap pengamatan (detik)					Rata-rata Waktu
	1	2	3	4	5	
1	0,0001281	0,0001213	0,0001481	0,0001751	0,000114	0,0001
2	0,0002799	0,0001248	0,0001696	0,0001246	0,000126	0,0002
3	0,0002812	0,0002938	0,0003451	0,0004602	0,0002766	0,0003
4	0,0008689	0,0009575	0,0023146	0,0008158	0,0009054	0,0012
5	0,0023911	0,0024172	0,0023204	0,0033052	0,0027079	0,0026
6	0,0108053	0,0126877	0,0209706	0,0194441	0,0132192	0,0154
7	0,0808614	0,1013003	0,1229037	0,1015065	0,0975181	0,1008
8	0,7556259	0,7948625	0,7342209	0,8368133	0,6840023	0,7611
9	6,6227222	5,6913418	6,761632	6,5179045	5,5637297	6,2315

Hasil pengamatan waktu eksekusi fungsi minimax oleh komputer pada papan 3 x 3 dapat dilihat pada Tabel 1. Komputer memerlukan waktu dalam orde sepersepuluh milidetik untuk memilih kotak yang akan diisi ketika jumlah kotak kosong hanya satu sampai tiga kotak. Waktu yang diperlukan untuk memilih kotak akan semakin besar jika kotak kosong semakin banyak. Sebaliknya waktu yang diperlukan akan semakin sedikit jika kotak kosong semakin sedikit. Ketika kotak kosong ada sembilan, komputer memerlukan waktu hingga 6 detik untuk memutuskan kotak yang dipilihnya.

Lamanya waktu dalam memilih kotak berubah secara tidak linier terhadap perubahan jumlah kotak kosong. Jika digambarkan dalam bentuk grafik, waktu eksekusi fungsi minimax menjadi relatif linier jika koordinat waktu digambar dengan skala logaritmis (lihat Gambar 7). Fakta ini mengindikasikan bahwa waktu eksekusi fungsi minimax bertambah secara eksponensial terhadap jumlah kotak kosong yang perlu diisi.



**Gambar 7. Grafik waktu eksekusi fungsi minimax pada papan 3 x 3 terhadap banyaknya kotak kosong yang harus dipilih**

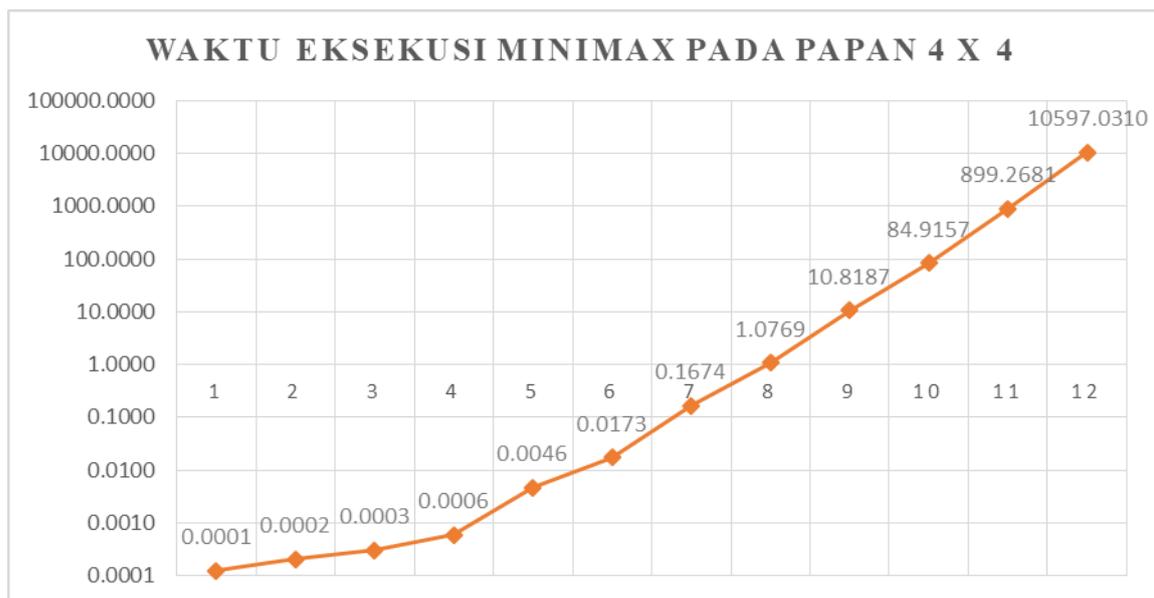
Hasil pengamatan waktu eksekusi fungsi minimax pada papan 4 x 4 disajikan pada Tabel 2. Jika dibandingkan dengan hasil pengamatan untuk papan 3 x 3, terdapat kesamaan waktu eksekusi untuk jumlah kotak kosong yang kecil. Namun secara umum waktu eksekusi pada papan 4 x 4 lebih lama dibanding aksi yang sama pada papan 3 x 3. Komputer memerlukan waktu dalam orde sepersepuluh milidetik untuk memilih kotak yang akan diisi ketika jumlah kotak kosong hanya satu sampai empat kotak. Waktu yang diperlukan untuk memilih kotak akan semakin besar jika kotak kosong semakin banyak.

Ketika kotak kosong berjumlah lebih dari 9 buah, komputer memerlukan waktu yang cukup lama untuk membuat langkah. Bahkan ketika kotak kosong berjumlah 12 buah, waktu yang diperlukan mencapai 10597 detik atau lebih dari 3 jam. Peneliti tidak mencoba eksekusi ketika kotak kosong berjumlah 13-16 buah karena waktu yang diperlukan sudah terlalu lama. Tentu saja tidak akan ada orang yang bersedia memainkan game ketika komputer memerlukan waktu bermenit-menit atau berjam-jam ketika melangkah.

**Tabel 2.** Waktu eksekusi minimax pada papan 4 x 4

Jumlah kotak kosong	Rata-rata waktu eksekusi (detik)
1	0,0001
2	0,0002
3	0,0003
4	0,0006
5	0,0046
6	0,0173
7	0,1674
8	1,0769
9	10,8187
10	84,9157
11	899,2681
12	10597,0310

Jika data pada Tabel 2 disajikan secara grafis, terlihat kenaikan waktu eksekusi yang cukup linier terhadap jumlah kotak kosong. Kecenderungan linier terlihat karena koordinat waktu digambarkan secara logaritmis. Berdasarkan bentuk grafik ini dapat dinyatakan bahwa waktu eksekusi algoritma minimax bersifat eksponensial terhadap jumlah kotak kosong yang sedang dipilih. Jika dinyatakan dalam notasi big-O, kompleksitas algoritmanya adalah  $O(x^n)$ .



**Gambar 8.** Grafik waktu eksekusi fungsi minimax pada papan 4 x 4 terhadap banyaknya kotak kosong

### 3.3 Pengujian dengan pemain

Pengujian game selanjutnya dilakukan dengan dimainkan oleh lima pemain manusia yang memainkan game secara berulang. Pemain memainkan papan 3 x 3 dengan mode random, papan 3 x 3 dengan mode minimax, dan papan 4 x 4 dengan mode campuran. Hasil pengujian disajikan pada Tabel 3.

**Tabel 3.** Hasil pengujian game kepada pemain manusia

Orang ke	Jumlah kemenangan		
	Papan 3 x 3 random	Papan 3 x 3 minimax	Papan 4 x 4 campuran
1	3	0	1
2	3	0	0
3	3	0	0
4	3	0	0
5	3	0	0
Rata-rata hasil	100% menang	80% seri 20% kalah	90% seri 10% menang

Tabel 3 memperlihatkan bahwa game tic tac toe yang dimainkan oleh komputer dengan mode random tidak mampu memainkan game dengan baik. Keseluruhan permainan dimenangkan oleh pemain manusia. Sebaliknya komputer yang menjalankan langkah dengan algoritma minimax tidak pernah mengalami kekalahan. Pemain manusia hanya dapat memperoleh hasil seri atau kalah. Selanjutnya komputer yang menjalankan langkah dengan mode campuran dapat memperoleh hasil menang, seri atau kalah. Variasi hasil terjadi tergantung seberapa banyak komputer menjalankan mode random dalam satu permainan. Hasil pada tabel diperoleh ketika komputer memainkan papan 4 x 4 dengan mode random pada 3 langkah pertama dan menjalankan algoritma minimax untuk langkah-langkah berikutnya.

### 3.4 Pembahasan

Tic tac toe adalah permainan papan yang dimainkan dengan meletakkan marker secara bergantian antara dua pemain. Bentuk marker untuk pemain pertama berbeda dengan pemain kedua, misalnya yang satu marker silang dan yang satu marker lingkaran. Dalam bentuk fisik tic tac toe dimainkan oleh dua orang manusia. Sedangkan dalam bentuk game komputer, salah satu pemain dapat diperankan oleh komputer.

Pemain game biasanya menginginkan game yang menantang, di mana pemain berpeluang lebih banyak mendapatkan kemenangan, tapi bisa mendapatkan hasil seri atau kalah. Pemain tidak menyukai game yang terlalu mudah dimenangkan atau game yang tidak dapat dimenangkan [9]. Pada permainan tic tac toe, komputer dapat memainkan game dengan mode random artinya komputer memilih kotak secara acak. Pada mode random, komputer dengan mudah dikalahkan oleh manusia dan komputer meletakkan marker secara tidak logis. Data penelitian ini membuktikan pemain manusia 100% dapat mengalahkan komputer.

Komputer dapat diberi kecerdasan dalam memainkan tic tac toe yaitu dengan menerapkan algoritma minimax ketika memilih langkah. Algoritma minimax memperhitungkan semua kemungkinan langkah sendiri dan langkah lawan dan berusaha memaksimalkan poin menuju kemenangan dan meminimalkan poin menuju kekalahan. Penerapan algoritma minimax

menyebabkan komputer tidak dapat dikalahkan dalam permainan tic tac toe. Data penelitian menunjukkan bahwa komputer memenangkan 20% permainan dan sisanya menghasilkan skor seri sedangkan pemain manusia tidak pernah mendapat kemenangan.

Untuk menghasilkan permainan yang menantang pada tic tac toe, diperlukan kombinasi penerapan teknik random dan algoritma minimax. Hasil akhir permainan akan tergantung seberapa banyak langkah komputer ditentukan secara random dan seberapa yang ditentukan dengan algoritma minimax. Pada penelitian ini dicoba membuat game tic tac toe berukuran 4 x 4 di mana 3 langkah pertama komputer ditentukan secara random dan langkah sisanya ditentukan dengan minimax. Hasilnya menunjukkan 10% permainan dimenangkan pemain manusia dan sisanya menghasilkan seri. Peneliti menduga jika penentuan langkah diatur sedemikian rupa, akan dapat diperoleh permainan yang lebih menantang dengan persentase kemenangan, seri dan kekalahan yang lebih berimbang.

Penerapan algoritma minimax memiliki kelemahan yaitu waktu komputasi yang lama terutama ketika jumlah kotak kosong masih banyak. Kelemahan ini terutama terasa untuk game tic tac toe berukuran 4 x 4 atau lebih. Untuk kotak kosong berjumlah lebih dari 9 maka waktu yang diperlukan menjalankan algoritma minimax sudah berada dalam orde detik. Untuk kotak kosong berjumlah 13, diperlukan waktu eksekusi hingga 3 jam. Tentunya pemain manusia tidak ingin menunggu komputer bermain terlalu lama. Untuk itu perlu diupayakan perbaikan dalam algoritma minimax. Di antara teknik yang dapat diterapkan adalah alpha beta pruning di mana komputer dapat mengabaikan posisi papan tertentu berdasarkan probabilitas bahwa tidak mungkin suatu langkah akan diambil [13], [14]. Teknik ini dapat menurunkan waktu eksekusi minimax sebesar 25%. Namun penurunan 25% belum memadai. Teknik-teknik lain seperti penggunaan memori untuk mengingat posisi papan yang sudah diuji dinyatakan dapat lebih banyak menurunkan waktu eksekusi. Penggunaan kecerdasan buatan telah pula dicoba [15]. Pada penelitian ini, ketika jumlah kotak kosong lebih dari 9, maka langkah komputer dilakukan secara random. Teknik ini memungkinkan game berjalan dengan cukup cepat. Berdasarkan data pengamatan maka jumlah kemenangan pemain manusia pada permainan tic tac toe berukuran 4 x 4 masih tergolong kecil tapi pemain memiliki kemungkinan untuk menang, seri atau kalah. Permainan seperti ini lebih menantang dan disukai daripada game di mana manusia selalu menang atau tidak pernah menang [9].

Gambar 6 dan Gambar 8 memperlihatkan grafik yang waktu eksekusi algoritma *minimax* ketika diterapkan pada papan 3 x 3 dan 4 x 4. Gambar disajikan secara *logaritmis* agar data terlihat jelas karena kenaikan waktu eksekusi setiap penambahan kotak kosong sangat besar. Bentuk grafik menunjukkan bahwa semakin banyak jumlah kotak kosong maka waktu semakin lama, sebaliknya semakin sedikit kotak kosong maka semakin cepat waktu untuk komputer memilih langkah. Peningkatan waktu eksekusi terhadap jumlah kotak kosong bersifat eksponensial sehingga kompleksitas algoritma minimax adalah  $O(x^n)$  [16].

#### **D. Simpulan**

Penelitian ini telah menghasilkan game tic tac toe dengan ukuran 3 x 3 dan 4 x 4. Game memiliki 3 mode permainan yaitu manusia lawan manusia, manusia lawan komputer yang memilih langkah secara random, dan manusia lawan komputer dengan algoritma minimax. Komputer yang memainkan game tic tac toe dengan mode random dengan mudah dikalahkan oleh pemain manusia. Jika komputer menggunakan algoritma minimax ketika bermain, maka pemain manusia tidak pernah menang dan maksimal mendapatkan hasil seri. Pemain game cenderung tidak suka bermain selalu menang atau tidak pernah menang dan lebih menyukai permainan yang menantang, yaitu yang memungkinkan untuk menang, kalah atau seri. Penggabungan mode random dan mode minimax menghasilkan game yang memenuhi kriteria sebagai game yang menantang.

### E. Referensi

- [1] A. Chesham, P. Wyss, R. M. Müri, U. P. Mosimann, T. Nef, and others, "What older people like to play: genre preferences and acceptance of casual games," *JMIR serious games*, vol. 5, no. 2, p. e7025, 2017.
- [2] I. Rahmawati and T. Sibuea, "Penelitian Eksperimen Siswa Kelas Delapan SMP PGRI Kalimulya: Keuntungan Memanfaatkan Situs Game Pembelajaran sebagai Media Pembelajaran Berbasis Web," *J. Manaj. Pendidik. dan Ilmu Sos.*, vol. 2, no. 2, pp. 858–869, 2021.
- [3] M. Quwaider, A. Alabed, and R. Duwairi, "The impact of video games on the players behaviors: A survey," *Procedia Comput. Sci.*, vol. 151, pp. 575–582, 2019.
- [4] S. Mulyati, "Serious Game to Training Focus for Children with Attention Deficit Hyperactivity Disorder: 'Tanji Adventure to the Diamond Temple,'" *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 9, no. 1, 2023.
- [5] N. Aziz, M. J. Nordin, S. J. Abdulkadir, and M. M. M. Salih, "Digital addiction: systematic review of computer game addiction impact on adolescent physical health," *Electronics*, vol. 10, no. 9, p. 996, 2021.
- [6] V. Rosendo-Rios, S. Trott, and P. Shukla, "Systematic literature review online gaming addiction among children and young adults: A framework and research agenda," *Addict. Behav.*, p. 107238, 2022.
- [7] E. Sudarmilah, U. Fadlilah, H. Supriyono, F. Y. Al Irsyadi, Y. S. Nugroho, and A. Fatmawati, "A review: Is there any benefit in serious games?," in *AIP Conference Proceedings*, 2018, vol. 1977, no. 1, p. 20059.
- [8] O. Lopez-Fernandez and D. J. Kuss, "Preventing harmful internet use-related addiction problems in Europe: A literature review and policy options," *Int. J. Environ. Res. Public Health*, vol. 17, no. 11, p. 3797, 2020.
- [9] W. Kramer, "What makes a game good," *Game & Puzzle Des. vol. 1, no. 2, 2015*, p. 84, 2000.
- [10] M. Bond and R. Beale, "What makes a good game? Using reviews to inform design," *People Comput. XXIII Celebr. People Technol.*, pp. 418–422, 2009.

- 
- [11] D. J. Hemanth and others, “Analysis of Minimax Algorithm Using Tic-Tac-Toe,” 2020.
- [12] J. Madake, C. Deotale, G. Charde, and S. Bhatlawande, “CHESS AI: Machine learning and Minimax based Chess Engine,” in *2023 International Conference for Advancement in Technology (ICONAT)*, 2023, pp. 1–6.
- [13] S. Mandadi, B. Tejashwini, and S. Vijayakumar, “Implementation of sequential and parallel alpha-beta pruning algorithm,” *Int. J. Innov. Eng. Res. Technol.*, vol. 7, no. 08, pp. 98–104, 2020.
- [14] C. Felstiner, “Alpha-Beta Pruning.” Whitman College, 2019.
- [15] D. Perez-Liebana, S. Samothrakis, J. Togelius, S. M. Lucas, and T. Schaul, “General video game ai: Competition, challenges and opportunities,” *ojs.aaai.org*, 2016.
- [16] J. He, D. Zhou, and Q. Gu, “Nearly minimax optimal reinforcement learning for discounted MDPs,” *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 22288–22300, 2021.