PENERAPAN ALGORITMA MINIMAX DENGAN ALPHA-BETA PRUNING PADA PERMAINAN TIC-TAC-TOE MENGGUNAKAN FRAMEWORK FLUTTER

Yofhi Fauda Pradana, Yovi Litanianda
Teknik Informatika, Universitas Muhammadiyah Ponorogo
Jl. Budi Utomo No.10, Ronowijayan, Ponorogo, Jawa Timur
Yofhi132@gmail.com, yovi@umpo.ac.id

ABSTRAK

Tic-Tac-Toe merupakan permainan yang hanya dimainkan dengan kertas dan pensil atau alat tulis lainnya atau dengan kata lain masuk kategori genre *paper* and *pencil game*. Algoritma minimax dapat digunakan untuk membantu komputer dalam membuat keputusan optimal saat bermain Tic-Tac-Toe. Namun minimax dapat menjadi tidak efektif karena harus memeriksa setiap langkah secara menyeluruh pada permainan dengan banyak pilihan langkah. Penelitian ini bertujuan untuk menerapkan algoritma minimax dengan alpha-beta pruning dalam permainan Tic-Tac-Toe menggunakan *framework* Flutter yang mampu menghasilkan permainan Tic-Tac-Toe sebagai langkah pengambilan keputusan komputer untuk memenangkan permainan atau setidaknya seri sehingga membuat pemain merasa seperti berhadapan dengan orang lain. Pengujian dilakukan sebanyak 20 kali dengan komputer dan pengguna bergantian menjadi pemain pertama. Hasilnya, komputer berhasil memenangkan 9 dari 20 kali uji coba, sedangkan pada 11 kali pecobaan lainya mendapatkan hasil seri. Hal ini menunjukkan bahwa algoritma minimax dengan alpha-beta pruning efektif dalam menentukan langkah optimal.

Kata kunci : *tictactoe, minimax ,alpha-beta pruning, flutter*

1. PENDAHULUAN

Tic-Tac-Toe adalah permainan kuno yang dikenal pula dengan "X dan O", memiliki sejarah panjang dan konon berasal dari era kekaisaran Romawi. Meskipun asal-usulnya tidak pasti, permainan ini telah banyak digemari di berbagai belahan dunia.Dengan hanya membutuhkan selembar kertas dan pulpen, Tic-Tac-Toe dapat dimainkan oleh dua orang. Kesederhanaannya tidak mengurangi manfaatnya. Permainan ini diketahui mampu mengasah kemampuan berpikir, terutama dalam hal perencanaan strategis dan pengambilan keputusan[1]. Meskipun sederhana dalam konsepnya, Tic-Tac-Toe menantang pemain untuk mengembangkan strategi yang efektif dalam rangka untuk mencapai kemenangan**.**

Dalam permainan papan seperti Tic-Tac-Toe, algoritma minimax hadir untuk membantu pemain menentukan langkah terbaik. Algoritma ini bekerja dengan memeriksa keadaan permainan pada setiap tahapan dan menggunakan struktur pohon untuk memetakan semua tahapan yang dapat diselesaikan. Dalam pohon permainan, algoritma mencari jalur yang memaksimalkan keuntungan dan meminimalkan kerugian bagi pemain yang sedang bermain, dengan asumsi lawan juga bermain maksimal.[2].

Dengan mempertimbangkan nilai maksimum dan minimum dari setiap kemungkinan gerakan, algoritma minimax dapat memilih langkah terbaik untuk mencapai hasil permainan yang paling menguntungkan. Namun, minimax mengunjungi setiap node dalam pohon keputusan, termasuk yang jelas tidak optimal. Hal ini menyebabkan banyak waktu dan sumber daya yang terbuang, terutama untuk permainan dengan pohon keputusan yang sangat besar.

Minimax dapat menjadi tidak efektif karena harus memeriksa setiap langkah secara menyeluruh pada permainan dengan banyak pilihan langkah. Oleh karena itu, dalam permainan yang lebih kompleks dengan kedalaman pohon keputusan yang lebih besar, seperti catur, minimax dapat menjadi sangat lambat dan tidak praktis[3].

Untuk mengatasi kekurangan tersebut, algoritma minimax dapat dioptimalkan dengan teknik alpha-beta pruning. Alpha-beta pruning berfungsi untuk mengurangi jumlah simpul yang perlu dievaluasi dalam pohon permainan, dengan mengeliminasi langkah-langkah yang tidak diperlukan. Hal ini memungkinkan percepatan proses pengambilan keputusan tanpa mengorbankan kualitas hasil. Dengan demikian, penerapan algoritma minimax dengan alpha-beta pruning dapat meningkatkan efisiensi dan efektivitas dalam pengambilan keputusan, memberikan solusi yang lebih optimal dalam permainan Tic-Tac-Toe[4].

Penelitian ini bertujuan untuk mengimplementasikan algoritma minimax dengan alpha-beta pruning dalam permainan Tic-Tac-Toe menggunakan framework Flutter. Diharapkan bahwa pendekatan ini tidak hanya memberikan pengalaman bermain yang lebih luas dan menantang bagi pemain, tetapi juga meningkatkan efektivitas dan efisiensi dalam pengambilan keputusan komputer untuk memenangkan permainan atau setidaknya mencapai hasil imbang.

2. TINJAUAN PUSTAKA

2.1. Tic-Tac-Toe

Tic-Toe-Toe adalah permainan yang hanya dimainkan dengan kertas dan pensil atau alat tulis lainnya atau dengan kata lain masuk kategori genre

paper and pencil game. Nama lain dari permainan ini adalah *Nought* & *Cross*. Permainan ini juga bisa disebut dengan X dan O, dikarenakan permainan ini hanya menggunakan simbol X/O. Seperti permainan papan yang lainnya, tic tac toe juga memiliki beberapa aturan dalam cara bermainnya[1]**.** Peraturan ini ialah sebagai berikut:.

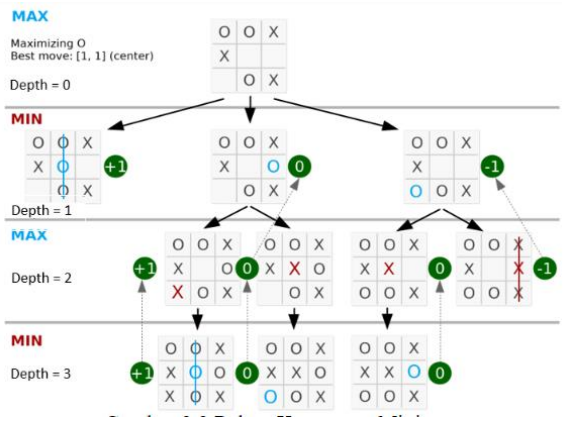
a. Pemain hanya boleh dilakukan oleh 2 orang
b. Awal permainan menggunakan papan kosong
c. Bermain secara bergantian menempatkan simbol X atau O yang terdapat di dalam kotak yang berukuran 3x3
d. Pemain pertama mendapatkan simbol X, dan pemain kedua mendapatkan simbol O
e. Dengan asumsi simbol X mendapat langkah pertama, pemain dengan simbol X memiliki satu lebih banyak dari jumlah simbol huruf O.
f. Permainan berakhir apabila salah satu pemain simbolnya berhasil ditempatkan tepat satu garis, baik garis secara horisontal, vertikal, dan diagonal atau saat kotak pada papan telah diisi penuh oleh simbol
g. Bila berakhir, kedua pemain tidak ada yang bergerak lagi[5].

Papan kosong saat permainan dimulai. Pemain X dan O akan menempatkan ke papan sekaligus. Pemain yang mampu menempatkan tiga biji dalam satu garis (vertikal, horizontal, atau diagonal) itulah yang menang. Selain itu, ketika papan penuh dan tidak ada yang menang, pertandingan dianggap seri[6]

2.2. Algoritma Minimax

Salah satu pendekatan yang paling umum digunakan dalam pengembangan permainan strategis Tic-Tac-Toe adalah algoritma minimax. Algoritma minimax dapat menghitung langkah-langkah dengan logis berdasarkan aturan permainan yang berlaku. Secara khusus, algoritma ini sering diterapkan pada permainan yang melibatkan dua pemain, di mana untuk setiap langkah, dilakukan perhitungan untuk menemukan langkah paling optimal dengan mempertimbangkan kemungkinan langkah yang akan diambil oleh lawan.[7].

Algoritma minimax menggunakan dua jenis simpul dalam repersentasi pohon, simpul max dan simpul min. Simpul max memilih langkah dengan nilai tertinggi, dan simpul min memilih langkah dengan nilai terendah. Untuk menentukan keputusan antara simpul max dan min, diperlukan nilai yang menunjukkan keuntungan atau kerugian yang akan diperoleh jika langkah tersebut dipilih. Di sini, fungsi heuristik digunakan. Untuk itulah disini digunakan sebuah fungsi heuristik[3].

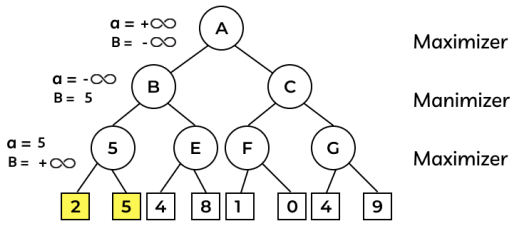


Gambar 1. Algoritma Minimax

2.3. Alpha-Beta Pruning

Alpha-beta pruning merupakan teknik optimasi yang di gunakan untuk algoritma minimax. Dengan menerapkan teknik alpha-beta pruning pencarian dapat memperoleh hasil yang sebanding dengan minimax tetapi dengan waktu yang lebih sedikit. Teknik alpha-beta pruning diterapkan kedalam algoritma minimax dengan menambahkan dua parameter baru yaitu alpha dan beta, yang dimana alpha adalah nilai terbaik yang telah didapatkan setelah menelusuri pohon pencarian, dan Beta adalah nilai terendah yang telah didapatkan setelah menelusuri pohon pecarian[4].

Dalam penerapan alpha-beta pruning memungkinkan pencarian lebih cepat dan bahkan masuk ke tingkat yang lebih tinggi di pohon permainan. Teknik tersebut memotong cabang di pohon permainan yang tidak perlu dicari karena sudah ada langkah yang lebih baik yang tersedia. Misalkan komputer sedang memeriksa langkah untuk permainan Tic-Tac-Toe. Komputer akan memulai dengan nilai alpha = -∞ dan beta = ∞. Komputer kemudian akan memeriksa semua kemungkinan langkah untuk pemain selanjutnya[1].



Gambar 2. Pohon Keputusan Alpha-Beta Pruning

2.4. Framework Flutter

Flutter adalah sebuah kerangka kerja *multiplatform* yang dikembangkan oleh Google untuk membangun antarmuka yang menarik bagi aplikasi *mobile*, seperti Android dan iOS. Bahasa pemrograman yang digunakan dalam kerangka kerja Flutter adalah Dart. Dart sendiri merupakan bahasa pemrograman yang dikembangkan oleh Google dan dapat digunakan untuk berbagai platform, termasuk Flutter, web, dan server. Struktur sintaksis Dart

menyerupai dengan bahasa pemrograman lain seperti C++, C#, Java, dan JavaScript.[8].

Flutter menerapkan kodenya melalui penggunaan widget. Widget dalam Flutter dapat berperan sebagai komponen visual atau hanya sebagai wadah bagi widget lainnya. Oleh karena itu, Flutter memiliki kode yang berstruktur hierarkis.[9].

Pada *platform* Android, aplikasi dibuat dengan menggunakan mesin bahasa C/C++ melalui Android NDK, Setelah itu, kode tersebut dikompilasi kembali menggunakan *dart compiler*. Sementara di platform iOS, kode aplikasi dikompilasi menggunakan LLVM (*Low Level Virtual Machine*) dan dijalankan menggunakan kumpulan instruksi yang spesifik untuk perangkat tersebut tanpa memerlukan interpretasi tambahan.[10].

3. METODE PENELITIAN

3.1. Pseudocode Algoritma Alpha-Beta Pruning

Untuk menyelesaikan permasalahan Tic-Tac-Toe dengan menggunakan algoritma minimax yang di optimasi dengan alpha-beta pruning, berikut ini merupakan bentuk dari pseudocode:

Tabel 1. Pseudocode Alpha-Beta Pruning

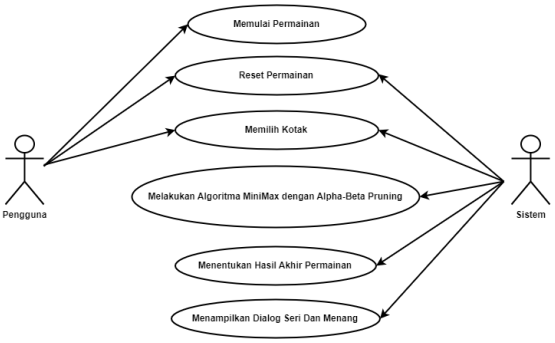
```
function minimax(depth, isMaximizingPlayer, alpha, beta):
    if game over in current state:
        return score
    if isMaximizingPlayer:
        bestScore = -INFINITY
        for each possible move:
         score = minimax(depth + 1, false, alpha, beta)
            bestScore = max(bestScore, score)
            alpha = max(alpha, bestScore)
            if beta <= alpha:
                break
        return bestScore
    else:
        bestScore = INFINITY
        for each possible move:
         score = minimax(depth + 1, true, alpha, beta)
            bestScore = min(bestScore, score)
            beta = min(beta, bestScore)
            if beta <= alpha:
                break
        return bestScore
```

Fungsi minimax memiliki beberapa parameter seperti depth yang menunjukkan kedalaman pencarian, isMaximizingPlayer untuk menentukan apakah pemain saat ini sedang memaksimalkan atau meminimalkan, serta alpha dan beta untuk teknik pemangkasan alpha-beta. Jika permainan mencapai kondisi terminal, fungsi tersebut akan mengembalikan nilai skor. Saat pemain dalam mode memaksimalkan, algoritma akan mencoba setiap langkah yang mungkin, menghitung skor untuk setiap langkah secara rekursif, dan memilih skor maksimum. Nilai alpha akan diperbarui dan pemangkasan dilakukan jika perlu. Di sisi lain, saat pemain sedang meminimalkan, algoritma juga akan mencoba setiap langkah yang mungkin, menghitung skor untuk setiap langkah secara rekursif, dan memilih skor minimum. Nilai beta akan diperbarui dan pemangkasan dilakukan jika perlu. Teknik pemangkasan alpha-beta membantu mengurangi jumlah simpul yang diperiksa dalam pohon permainan dengan menghapus cabang yang tidak relevan, berdasarkan nilai alpha (nilai terbaik yang ditemukan hingga saat ini pada level pemain maksimum) dan beta (nilai terbaik yang ditemukan hingga saat ini pada level pemain minimum). Ini meningkatkan efisiensi algoritma minimax dengan menghindari pemeriksaan simpul yang tidak berpengaruh pada hasil akhir permainan.

3.2. Use Case Diagram

Perancangan sistem pada penelitian ini memiliki use case yang menggambarkan bahwa pengguna dapat memulai permainan serta kemampuan untuk mengatur ulang skor permainan jika diperlukan. Sistem menggunakan algoritma minimax dengan alpha-beta pruning untuk menentukan langkah komputer saat bermain melawan pengguna. Ini memastikan bahwa komputer membuat langkah yang maksimal untuk mencapai hasil terbaik dalam permainan. Selanjutnya, sistem secara teratur memeriksa kondisi permainan untuk menentukan apakah ada pemenang setelah setiap langkah yang diambil oleh pemain. Jika ada pemenang, sistem akan menampilkan dialog pemenang untuk mengumumkan hasilnya kepada pengguna. Jika permainan berakhir dengan hasil seri, sistem juga menampilkan dialog khusus untuk mengumumkan hasil seri kepada pengguna.

Selain itu, pengguna dan sistem juga memiliki kemampuan untuk mengatur ulang permainan, sehingga menghapus semua langkah sebelumnya dan mengembalikan papan permainan keadaan awal, pengguna dan sistem juga dapat memilih kotak kosong pada papan permainan. Use case diagram untuk interaksi pengguna dengan sistem terlihat seperti ini:



Gambar 3 Use Case Diagram Permainan

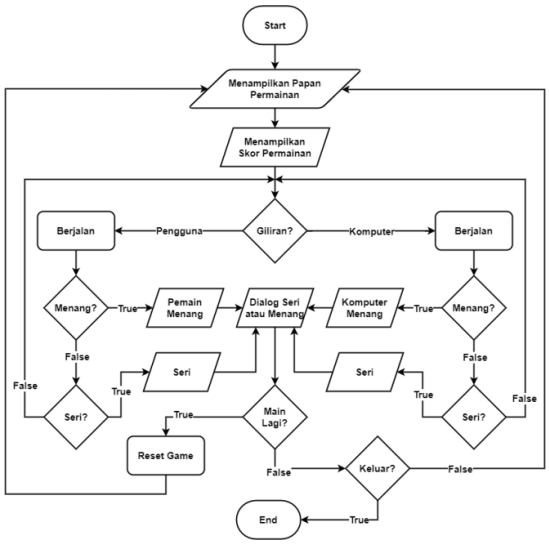
3.3. Flowchart Permainan

Flowchart permainan tic tac toe dimulai dengan menampilkan papan permainan dan skor saat ini. Sistem kemudian menentukan giliran, apakah itu

pengguna atau komputer. Jika giliran pengguna, pengguna melakukan langkahnya. Setelah itu, sistem memeriksa apakah pengguna menang. Jika pengguna menang, muncul pesan "Pemain Menang" dan dilanjutkan dengan dialog yang menanyakan apakah ingin bermain lagi. Jika belum menang, sistem memeriksa apakah permainan seri. Jika seri, muncul pesan "Seri" dan dilanjutkan ke dialog tadi. Jika tidak, giliran berikutnya ditentukan.

Jika giliran komputer, komputer melakukan langkahnya. Sistem kemudian memeriksa apakah komputer menang. Jika komputer menang, muncul pesan "Komputer Menang" dan dilanjutkan dengan dialog yang menanyakan apakah ingin bermain lagi. Jika belum menang, sistem memeriksa apakah permainan seri. Jika seri, muncul pesan "Seri" dan dilanjutkan ke dialog tadi. Jika tidak, giliran berikutnya ditentukan.

Setelah ada hasil menang atau seri, sistem menampilkan dialog yang menanyakan apakah pemain ingin bermain lagi. Jika pemain ingin bermain lagi, permainan direset dan papan permainan ditampilkan kembali. Jika pemain tidak ingin bermain lagi, sistem menanyakan apakah pemain ingin keluar. Jika iya, permainan berakhir. Jika tidak, kembali lagi ke dialog ingin bermain lagi.



Gambar 4. Flowchart Permainan

4. HASIL DAN PEMBAHASAN

4.1. Penerapan Algoritma Alpha-Beta Pruning

Untuk menyelesaikan permasalahan tic tac toe dengan menggunakan algoritma minimax yang di optimasi dengan alpha-beta pruning, berikut kode dari fungsi minimax yang di tulis dalam bahasa pemograman dart:

Tabel 2. Penerapan Alpha-Beta Pruning

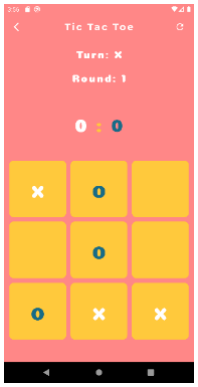
```
int _minimax(int depth, bool isMax, int alpha, int beta) {
  int score = _evaluate();
  if (score == 10) return score;
  if (score == -10) return score;
  if (_isMovesLeft() == false) return 0;
  if (isMax) {
    int best = -1000;
    for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
        if (_matrix[i][j] == '') {
          _matrix[i][j] = 'O';
          best = max( best,
_minimax(depth + 1, !isMax, alpha, beta) - depth);
          _matrix[i][j] = '';
          alpha = max(alpha, best);
          if (beta <= alpha) break;
        }}
    }
    return best;
  } else {
    int best = 1000;
    for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
        if (_matrix[i][j] == '') {
          _matrix[i][j] = 'X';
          best = min(
            best,
_minimax(depth + 1, !isMax, alpha, beta) + depth);
          _matrix[i][j] = '';
          beta = min(beta, best);
          if (beta <= alpha) break;
        }}
    }
    return best;
  }}
```

Parameter dari minimax adalah depth, isMax, alpha, dan beta. Depth adalah menunjukkan kedalaman saat ini dalam pencarian algoritma minimax. Ketika nilai kedalaman mencapai batas yang ditentukan, pencarian akan dihentikan, dan penilaian posisi akan dilakukan, isMax adalah menunjukkan apakah saat ini pemain sedang maksimalkan atau meminimalkan skor. Jika isMax adalah true, maka pemain sedang berusaha memaksimalkan skor (dalam hal ini pemain O), dan sebaliknya jika false, pemain sedang berusaha meminimalkan skor (dalam hal ini pemain X), alpha adalah nilai terbaik yang saat ini diketahui untuk pemain maksimal (pemain O). Ini merupakan batas bawah yang dijamin untuk skor terbaik yang dapat dicapai oleh pemain maksimal. Sedangkan beta adalah nilai terbaik yang saat ini diketahui untuk pemain minimal (pemain X). Ini merupakan batas atas yang dijamin untuk skor terbaik yang dapat dicapai oleh pemain minimal

4.2. Tampilan Papan Permainan

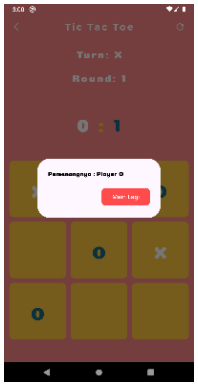
Pada tampilan utama permainan Tic-Tac-Toe, terdapat bagian atas dengan nama permainan "Tic Tac Toe". Terdapat tombol kembali dan reset untuk mengatur ulang permainan dan skor. Di bawahnya, ada indikator giliran pemain dengan simbol X dan O. Skor permainan ditampilkan dengan dua angka yang menunjukkan skor kedua pemain atau pengguna melawan komputer. Terdapat papan permainan 3x3

dengan kotak-kotak kuning tempat pemain menempatkan simbol selama permainan.



Gambar 5. Tampilan Utama Permaian Tic-Tac-Toe

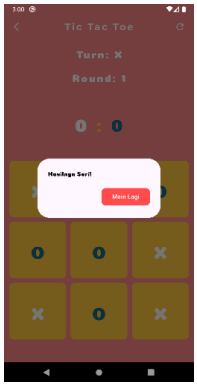
Ketika semua kotak terisi tanpa pemenang., maka akan muncul dialog di layar yang menyatakan bahwa pertandingan berakhir seri.



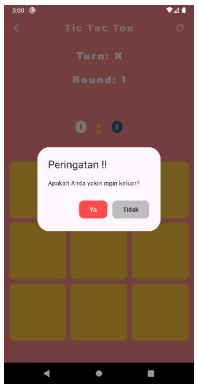
Gambar 6. Dialog Seri pada permainan

Sedangkan jika terdapat Pemain yang menandai tiga kotak berturut-turut, maka akan menampilkan dialog kemenangan.

Jika pengguna menekan tombool kembali maka akan muncul dialog yang menyatakan bahwa anda ingin keluar atau tidak.



Gambar 7. Dialog Kemangann



Gambar 8. Dialog Ketika Keluar

4.3. Pengujian Menggunakan *Blackbox Testing*

Pengujian terhadap permainan Tic-Tac-Toe akan dilakukan dengan menerapkan metode *Blackbox Testing*. Metode ini memungkinkan pengujian dilakukan tanpa perlu memahami detail internal perangkat lunak. Dengan n kata lain, penguji hanya akan berfokus pada *input* dan *output* permainan, tanpa perlu mengetahui bagaimana programnya bekerja secara detail. Berikut adalah tabel hasil dari pengujian fitur-fitur yang terdapat pada permianan Tic-Tac-Toe yang telah di bangun dapat berfungsi dengan baik sesuai dengan spesifikasi yang telah ditentukan.

Tabel 3. Pengujian Menggunakan *Blackbox Testing*

No	Skenario Pengujian	Hasil yang di harapkan	Keterangan
1	Membuka Aplikasi	Papan permainan kosong, Skor untuk kedua pemain kosong dan giliran pertama pemain.	Sesuai
2	Pengguna membuat langkah	Pengguna memilih kotak yang kosong	Sesuai
3	Komputer membuat langkah	Komputer memilih kotak yang masih kosong	Sesuai
4	Pemain menandai tiga kotak berturut-turut,	Menampilkan pesan kemenangan untuk pemain muncul, permainan berakhir.	Sesuai
5	Semua kotak terisi tanpa pemenang.	Menampilkan pesan seri untuk pemain muncul, permainan berakhir.	Sesuai
6	Menekan tombol bermain lagi ketika dialog kemenagan dan dialog seri muncul	Permainan akan mengosongkan kotak, memperbarui skor dan ronde permainan	Sesuai
7	Pemain memilih kotak yang sudah berisi.	Tidak ada perubahan pada papan	Sesuai
8	Menekan tombol Refresh.	Papan permainan kosong dan skor akan di atur ulang menjadi kosong	Sesuai
9	Pergantian Pemain	Pemain bergerak saling bergantian ketika sudah memilih kotak	Sesuai
10	Pergantian Ronde	Giliran pertama bergantian sesuai dengan aturan yang di	Sesuai

No	Skenario Pengujian	Hasil yang di harapkan	Keterangan
		tentukan dengan Pengguna bergerak pada ronde ganjil dan komputer bergerak pada ronde genap	
11	Menekan tombol kembali	Tampilan dialog konfirmasi Apakah Anda yakin ingin keluar?.	Sesuai
12	Menekan tombol Ya pada dialog keluar	Menutup aplikasi.	Sesuai
13	Menekan tombol Tidak pada dialog keluar	Menutup dialog dan kembali ke permainan.	Sesuai

4.4. Pengujian Permainan Tic-Tac-Toe

Dalam pengujian performa algoritma minimax dengan alpha-beta pruning pada permainan Tic-Tac-Toe, dilakukan serangkaian percobaan di mana komputer dan pengguna bergantian menjadi pemain pertama. Setiap percobaan dijalankan hingga permainan selesai, dengan catatan hasil kemenangan, kekalahan, atau seri. Pada setiap percobaan juga terdapat langkah permainan yang di ambil dari komputer dan pengguna. Hasil dari 20 percobaan permainan Tic Tac Toe adalah sebagai berikut:

Tabel 4. Pengujian permainan Tic-Tac-Toe

NO	Giliran Pertama	Langkah									Hasil Akhir
		1	2	3	4	5	6	7	8	9	
1	Pengguna	1,1	2,2	1,3	1,2	3,2	2,1	2,3	3,3	3,1	Seri
2	Komputer	1,2	2,3	1,3	1,1	2,2	3,2	3,1			Komputer Menang
3	Pengguna	3,1	2,2	1,1	2,1	2,3	1,2	3,2	3,3	1,3	Seri
4	Komputer	3,3	2,2	1,1	1,3	3,1	1,2	3,2			Komputer Menang
5	Pengguna	2,3	1,3	2,1	2,2	3,1	1,1	3,3	1,2		Komputer Menang
6	Komputer	2,2	1,1	1,2	3,2	2,1	2,3	1,3	3,1	3,3	Seri
7	Pengguna	1,1	2,2	3,3	1,2	3,2	3,1	1,3	2,3	2,1	Seri
8	Komputer	3,2	1,3	3,3	3,1	2,2	1,1	1,2			Komputer Menang
9	Pengguna	2,1	1,1	2,3	2,2	3,2	3,3				Komputer Menang
10	Komputer	1,3	2,2	1,1	1,2	3,2	2,1	2,3	3,3	3,1	Seri
11	Pengguna	3,1	2,2	1,3	1,2	3,2	3,3	1,1	2,1	2,3	Seri
12	Komputer	2,2	1,2	1,1	3,3	2,1	2,3	3,1			Komputer Menang
13	Pengguna	1,1	2,2	3,1	2,1	2,3	1,2	3,2	3,3	1,3	Seri
14	Komputer	1,2	2,2	1,1	1,3	3,1	2,1	2,3	3,3	3,2	Seri
15	Pengguna	3,3	2,2	1,1	1,2	3,1	3,2	1,3	2,3	2,1	Seri
16	Komputer	2,2	1,2	1,1	3,3	2,1	3,1	2,3			Komputer Menang
17	Pengguna	2,1	1,1	2,2	2,3	1,3	3,1	3,2	1,2	3,3	Seri
18	Komputer	2,2	1,3	1,1	3,1	3,3					Komputer Menang
19	Pengguna	3,1	2,2	1,3	1,2	3,2	3,3	1,1	2,1	2,3	Seri
20	Komputer	2,2	3,2	2,3	2,1	1,3	1,2	3,1			Komputer Menang

Pada pengujian permainan di atas, dilakukan 20 kali percobaan pada permainan Tic-Tac-Toe yang menggunankan algoritma minimax dengan alpha-beta pruning antara komputer dan pengguna yang saling bergantian menjadi giliran pertama. Dari hasil pengamatan 20 kali uji coba di atas, komputer mendapatkan 9 kali menang, sedangkan pada 11 kali pecobaan lainya mendapatkan hasil seri.

4.5. Analisa Hasil Pengujian

Berdasarkan hasil pengujian kinerja algoritma Minimax dengan alpha-beta pruning pada permainan Tic-Tac-Toe yang dilakukan sebanyak 20 kali, algoritma ini menunjukkan hasil yang memadai dalam pengambilan keputusan. Komputer berhasil memenangkan 9 dari 20 percobaan dan mendapatkan 3 kali hasil seri, sementara pengguna mendapatkan 8 kali hasil seri.

Hasil ini menunjukkan bahwa algoritma minimax dengan alpha-beta pruning efektif dalam menentukan langkah terbaik, sehingga komputer lebih sering menang dibandingkan pengguna. Lebih banyaknya kemenangan komputer menunjukkan bahwa algoritma yang diterapkan mampu memeriksa berbagai kemungkinan langkah dan memilih langkah terbaik untuk memaksimalkan peluang kemenangan atau meminimalkan peluang kekalahan.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan algoritma minimax dengan alpha-beta pruning pada permainan Tic-Tac-Toe. Melalui pengujian performa, dapat dilihat bahwa komputer mampu memberikan tantangan bagi pengguna. Dari 20 kali uji coba, komputer berhasil memenangkan 9 kali permainan, sedangkan 3 kali berakhir seri. Ini menunjukkan bahwa komputer mampu membuat keputusan yang baik dalam bermain Tic-Tac-Toe. Untuk penelitian selanjutnya, disarankan untuk menggunakan algoritma lain untuk di bandingkan dengan algoritma minimax pada permainan Tic-Tac-Toe

DAFTAR PUSTAKA

[1] N. R. Putri, N. Fitriani, V. H. Pranatawijaya,

and R. Priskila, "Kecerdasan Buatan dalam Permainan Tic-Tac- Toe Melalui Algoritma Minimax dengan Optimasi Alpha-Beta Pruning," *J. Minfo Polgan*, vol. 13, pp. 305–315, 2024, doi: 10.33395/jmp.v13i1.13594.

[2] H. Thamrin and R. W. A. Pamungkas, "Algoritma Minimax untuk Game Tic Tac Toe yang Menantang," *Indones. J. Comput. Sci.*, vol. 12, no. 3, pp. 1386–1398, Jun. 2023, doi: 10.33022/ijcs.v12i3.3214.

[3] T. A. Pracaya, A. B. Murti Wijaya, and H. Budiati, "Penerapan Algoritma Minimax Sebagai Kecerdasan Buatan Pada Permainan Tic Tac Toe," *J. Inform. UKRIM*, vol. 1, no. 1, pp. 1–12, 2020.

[4] B. Dinda Permatasari, H. Haryanto, E. Zuni Astuti, and E. Dolphina, "Peningkatan Kemenangan Non-Playable Character dalam Permainan Triple Triad Menggunakan Alpha-Beta Pruning," *J. Komputasi*, vol. 10, no. 1, pp. 95–104, 2022, doi: 10.23960/komputasi.v10i1.2952.

[5] A. A. Faqih and A. S. Fitrani, "Penerapan Game Tic-Tac-Toe dengan Metode Artificial Neural Network dalam Pengambilan Keputusan untuk Meraih Kemenangan," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 1, pp. 233–242, 2023.

[6] A. Christopher, E. Pratama, and L. Hakim, "Penerapan Algoritma Minimax Terhadap Permainan Tic-Tac-Toe Dengan Menggunakan Artificial Intelligence," *Ancol, Kec. Pademangan, Kota Jkt Utara*, vol. 05, no. 9, pp. 2657–1501, 2020, [Online]. Available: http://ejournal.ust.ac.id/index.php/JTIUST/article/view/941

[7] M. Masri, I. Uari, H. Alam, and M. I. Z. Firdaus, "Implementasi Algoritma Minimax Pada Permainan Checker Berbasis Artificial Intelligence," *J. Electr. Technol.*, vol. 1099, pp. 37–42, 2023.

[8] A. Alvin and Tukino, "RANCANG BANGUN GAME SUDOKU ANDROID BERBASIS FLUTTER," *Comput. Based Inf. Syst. J.*, vol. 01, pp. 52–62, 2024.

[9] S. Santoso, D. J. Surjawan, and E. D. Handoyo, "Pengembangan Sistem Informasi Tukar Barang Untuk Pemanfaatan Barang Tidak Terpakai dengan Flutter Framework," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 3, pp. 589–598, Dec. 2020, doi: 10.28932/jutisi.v6i3.3071.

[10] H. P. Ramadhan, C. Kartiko, and A. Prasetiadi, "Monitoring Kualitas Air Tambak Udang Menggunakan NodeMCU, Firebase, dan Flutter," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 1, pp. 102–114, 2020, doi: 10.28932/jutisi.v6i1.2365.