**LAPORAN PRAKTIKUM KE-1**

**PENGOLAHAN CITRA DIGITAL**



DISUSUN OLEH

Nama : - Salah Nasser Hasan Meqdam

* Nursita

Npm : G1A023095

Dosen pengampu:

1. Arie Vatresia، ST، M.TI، Ph.D.

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**

**2023**

**LANDASAN TEORI**

**1. Java**

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama Sun Microsystems, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama “Oak”, namun pada tahun 1995 diganti namanya menjadi “Java”.

Pengertian Java Menurut definisi Sun Microsystem, di dalam buku M. Shalahuddin dan Rosa A.S. (2010 : 1) Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (standalone) ataupun pada lingkungan jaringan. Java berdiri di atas sebuah mesin penterjemah (interpreter) yang diberi nama Java Virtual Machine (JVM). JVM inilah yang akan membaca kode bit (bytecode) dalam file .class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa Java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada system operasi tersebut terdapat JVM. Alasan utama pembentukan bahasa Java adalah untuk membuat aplikasi-aplikasi yang dapat diletakkan di berbagai macam perangkat elektronik, sehingga Java harus bersifat tidak bergantung pada platform (platform independent). Itulah yang menyebabkan dalam dunia pemrograman Java dikenal adanya istilah „write once, run everywhere‟, yang berarti kode program hanya ditulis sekali, namun dapat dijalankan di bawah kumpulan pustaka (platform) manapun, tanpa harus melakukan perubahan kode program.

**2. Arsitektur Java**

Secara arsitektur, Java tidak berubah sedikitpun sejak awal mula Bahasa tersebut dirilis. Compiler Java (yang disebut dengan javac atau Java Compiler) akan mentransformasikan kode-kode dalam bahasa Java ke dalam suatu kode bit. Dimana bytecode adalah sekumpulan perintah hasil kompilasi yang kemudian dapat dieksekusi melalui sebuah mesin komputer abstrak, yang disebut dengan JVM (Java Virtual Machine). JVM juga sering dinamakan sebagai interpreter, karena sifatnya yang selalu menerjemahkan kode-kode yang tersimpan dalam kode bit dengan cara baris demi baris. Untuk menjalankan program Java, maka file dengan ekstensi .java harus dikompilasi menjadi file kode bit. Dimana untuk menjalankan kode bit tersebut dibutuhkan JRE (Java Runtime Environment) yang memungkinka pemaka untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi JRE berisi JVM dan pustaka Java yang digunakan.

**3. Tic Tac Toe**

Permainan tic-tac-toe merupakan permainan klasik berjenis permainan papan (board-game) dengan ukuran 3x3. Cara memainkan Permainan tersebut dengan memberikan Nilai X atau O pada tiap kotak papan. Dalam Permainan ini hasil permainan yang didapat berupa menang, kalah, atau seri. Permainan ini mempunyai beberapa varian, antara lain 3D (3 dimensi), 2D (2 dimensi) dan juga board Permainans (permainan papan). Sejak tahun 1980an, permainan 3 dimensi semakin banyak peminatnya karena mnemberikan efek visualisasi yang berbeda bagi pengguna dan user experience yang berbeda pula jika dibandingkan dengan permainan 2 dimensi sehingga saat ini permainan 3 dimensi semakin dikembangkan dengan visual dan sound yang dapat membuat permainan lebih *real.*

**4. Tic Tac Toe Dengan Java**

Pembuatan permainan Tic-Tac-Toe dengan java , menerapkan beberapa konsep utamadalam java, antara lain :

* 1. Pemrograman Berorientasi Objek (OOP):

Dalam pembuatan permainan ini, digunakan konsep OOP untuk merancang kelas-kelas yang sesuai, seperti kelas TicTacToeBoard untuk merepresentasikan papan permainan dan kelas Player untuk mengelola pemain.

* TicTacToeBoard: Kelas ini bertanggung jawab untuk merepresentasikan papan permainan, mengelola posisi simbol, dan memeriksa kondisi permainan seperti pemenang dan seri.
* Player: Kelas ini digunakan untuk mengelola pemain dalam permainan. Setiap pemain memiliki atribut seperti nama dan simbol (X atau O).
  1. Input Pengguna:

Kelas Scanner digunakan untuk menerima input dari pengguna, seperti nama pemain dan posisi gerakan yang diinginkan.

* 1. Exception Handling:

Penting untuk menangani pengecualian yang mungkin terjadi saat menerima input pengguna, seperti masukan yang tidak valid.

* 1. Pengulangan:

Konsep pengulangan (loop) digunakan untuk mengatur alur permainan, termasuk bergantian pemain dan mengulangi permainan jika pengguna ingin bermain lagi.

**HASIL PRAKTIKUM**

1. **Perancangan Game**

Nama Game : Tic-Tac-Toe

Tujuan : Tujuan dari permainan ini adalah untuk membuat permainan Tic-Tac-Toe yang dapat dimainkan oleh dua pemain secara bergantian. Permainan ini bertujuan untuk memberikan pengalaman yang menyenangkan dan kompetitif kepada pemain sambil menguji kecerdasan mereka dalam merencanakan strategi untuk memenangkan permainan.

Konsep Dasar :

Konsep dasar permainan ini adalah permainan papan 3x3 di mana dua pemain bertujuan untuk mendapatkan tiga simbol mereka (X atau O) dalam satu baris, kolom, atau diagonal. Permainan ini akan dimainkan secara bergantian, dengan pemain memasukkan langkah mereka ke dalam papan permainan hingga ada pemenang atau hasil seri.

Fitur Utama :

1. Input Pemain: Memungkinkan pemain untuk memasukkan nama mereka sebagai Player 1 dan Player 2.
2. Pemilihan Simbol: Memungkinkan pemain untuk memilih simbol (X atau O) untuk bermain.
3. Papan Permainan: Menampilkan papan permainan 3x3 untuk pemain.
4. Validasi Langkah: Memastikan setiap langkah pemain adalah langkah yang sah.
5. Pengecekan Pemenang: Menentukan pemenang jika salah satu pemain berhasil mendapatkan tiga simbol mereka dalam satu baris, kolom, atau diagonal.
6. Pengecekan Seri: Menentukan permainan seri jika papan permainan terisi penuh tanpa ada pemenang.
7. Permainan Berulang: Memungkinkan pemain untuk memulai kembali permainan setelah selesai.

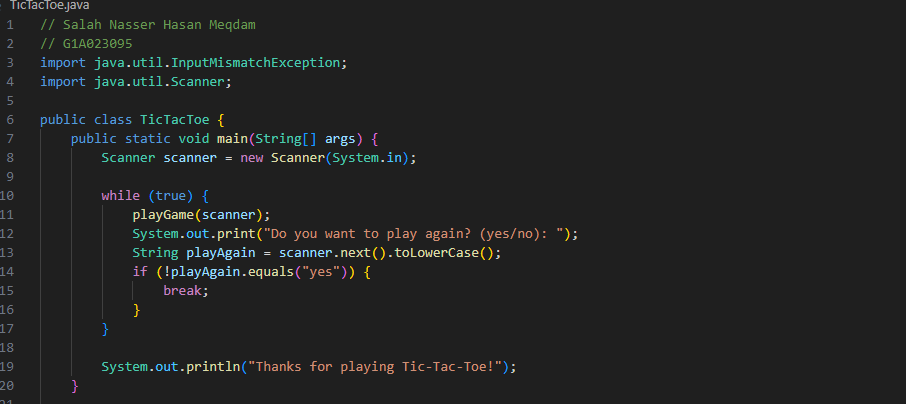
Alur Permainan :

1. Permainan dimulai dengan menyambut pemain dan memberikan instruksi tentang cara memainkan permainan.
2. Pemain diminta untuk memasukkan nama mereka sebagai Player 1 dan Player 2.
3. Pemain juga diminta untuk memilih simbol (X atau O) untuk bermain.
4. Papan permainan kosong akan ditampilkan.
5. Pemain akan bergantian untuk memasukkan langkah mereka ke dalam papan permainan.
6. Setiap langkah akan divalidasi untuk memastikan kevalidannya.
7. Setelah setiap langkah, permainan akan memeriksa apakah ada pemenang atau hasil seri.
8. Jika ada pemenang, pemenangnya akan diumumkan, dan permainan akan berakhir.
9. Jika hasilnya seri, permainan juga akan berakhir.
10. Pemain akan ditanya apakah mereka ingin memulai permainan lagi.
11. Jika mereka ingin bermain lagi, permainan akan dimulai kembali dari awal.
12. Jika tidak, permainan akan berakhir.

Tampilan Grafis :

Permainan akan menggunakan tampilan teks sederhana dengan tampilan papan permainan berbasis teks. Tampilan akan sederhana dan mudah dimengerti agar dapat diakses oleh berbagai pemain.

1. **Kode dan Penjelasan**
   1. **Inisialisasi dan Pengaturan Awal**



**Gambar 2.1 Source code inisialisasi dan pengaturan awal**

Penjelasan :

import java.util.InputMismatchException;

import java.util.Scanner;

Kode ini mengimpor dua kelas Java, yaitu InputMismatchException dan Scanner. InputMismatchException digunakan untuk menangani kesalahan dalam bentuk input yang tidak sesuai, sementara Scanner digunakan untuk membaca input dari pengguna.

public class TicTacToe {

Ini adalah awal deklarasi kelas utama dalam program.

public static void main(String[] args) {

Ini adalah method utama yang akan dijalankan pertama kali saat program dimulai.

Scanner scanner = new Scanner(System.in);

Kode ini membuat sebuah objek Scanner dengan nama scanner yang digunakan untuk membaca input dari pengguna melalui keyboard.

while (true) {

Ini adalah loop utama program yang akan terus berjalan selama kondisinya benar (dalam hal ini, selamanya). Ini berarti program akan terus berjalan hingga kondisi di dalam loop berubah menjadi false.

playGame(scanner);

Kode ini memanggil method playGame(scanner) untuk memulai permainan Tic-Tac-Toe. scanner adalah objek Scanner yang digunakan untuk membaca input dalam permainan.

System.out.print("Do you want to play again? (yes/no): ");

String playAgain = scanner.next().toLowerCase();

Kode ini mencetak pesan kepada pengguna dan meminta mereka untuk memasukkan jawaban apakah ingin bermain lagi atau tidak. Jawaban tersebut disimpan dalam variabel playAgain setelah diubah menjadi huruf kecil dengan toLowerCase().

if (!playAgain.equals("yes")) {

break;

}

Kode ini memeriksa apakah jawaban yang dimasukkan pengguna tidak sama dengan "yes". Jika tidak sama, maka program keluar dari loop utama dengan menggunakan pernyataan break.

System.out.println("Thanks for playing Tic-Tac-Toe!");

Setelah keluar dari loop utama, program mencetak pesan penutup yang menampilkan terima kasih kepada pemain yang telah bermain Tic-Tac-Toe.

* 1. **Permainan**



**Gambar 2.2 Source Code permainan utama**

Penjelasan :

System.out.println("Welcome to Tic-Tac-Toe!");

System.out.println("Instructions:");

System.out.println("1. Enter the names of the players.");

System.out.println("2. Choose 'X' or 'O' for each player.");

System.out.println("3. Players take turns to make a move.");

System.out.println("4. The first player to get three in a row wins.");

System.out.println("5. Have fun!\n");

Kode ini mencetak pesan selamat datang dan instruksi permainan ke layar. Instruksi ini menjelaskan aturan dasar permainan kepada pemain.

String player1Name = "";

String player2Name = "";

while (player1Name.trim().isEmpty()) {

System.out.print("Enter the name of Player 1: ");

player1Name = scanner.nextLine();

}

while (player2Name.trim().isEmpty()) {

System.out.print("Enter the name of Player 2: ");

player2Name = scanner.nextLine();

}

Kode ini mengatur pemain untuk memasukkan nama mereka sebagai "Player 1" dan "Player 2." Loop while digunakan untuk memastikan nama yang dimasukkan tidak kosong atau hanya berisi spasi.

char player1Symbol, player2Symbol;

while (true) {

System.out.println(player1Name + ", choose your symbol (X or O): ");

player1Symbol = scanner.next().toUpperCase().charAt(0);

if (player1Symbol == 'X' || player1Symbol == 'O') {

player2Symbol = (player1Symbol == 'X') ? 'O' : 'X';

break;

} else {

System.out.println("Invalid symbol choice. Please choose 'X' or 'O'.");

}

}

Kode ini memungkinkan Pemain 1 untuk memilih simbol mereka, yang bisa berupa 'X' atau 'O'. Loop while digunakan untuk memvalidasi pilihan simbol pemain dan memastikan pemilihan yang valid.

char[][] board = new char[3][3];

initializeBoard(board);

boolean gameOver = false;

char currentPlayer = player1Symbol;

Kode ini menginisialisasi papan permainan sebagai array 2D dengan ukuran 3x3 dan mengatur variabel lain seperti gameOver (untuk menentukan apakah permainan sudah selesai) dan currentPlayer (untuk menentukan giliran pemain saat ini).

while (!gameOver) {

Ini adalah loop utama permainan yang akan berjalan sampai permainan selesai (ketika gameOver menjadi true).

printBoard(board);

Kode ini mencetak papan permainan ke layar, menampilkan posisi langkah-langkah pemain saat ini.

int[] move = getPlayerMove(board, currentPlayer, player1Name, player2Name);

try {

int row = move[0] - 1;

int col = move[1] - 1;

} catch (InputMismatchException e) {

} catch (ArrayIndexOutOfBoundsException e) {

}

Kode ini memanggil method getPlayerMove untuk meminta langkah dari pemain yang sedang bermain. Menggunakan try-catch untuk menangani exception, seperti InputMismatchException (ketika pemain memasukkan input yang tidak sesuai) dan ArrayIndexOutOfBoundsException (ketika pemain memasukkan nomor baris atau kolom yang tidak valid).

if (isValidMove(board, row, col)) {

board[row][col] = currentPlayer;

// ...

} else {

System.out.println("Invalid move. Try again.");

}

Kode ini memeriksa apakah langkah pemain yang dimasukkan valid. Jika valid, langkah tersebut akan ditempatkan di papan. Selanjutnya, kode ini memeriksa apakah pemain tersebut telah memenangkan permainan atau jika permainan berakhir seri.

} catch (InputMismatchException e) {

System.out.println("Invalid input. Please enter valid row and column numbers.");

scanner.nextLine(); // Clear input buffer

} catch (ArrayIndexOutOfBoundsException e) {

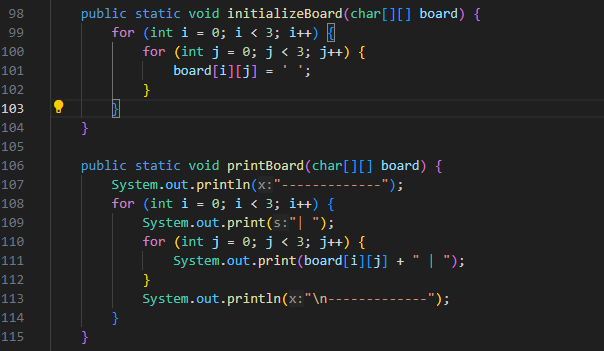
System.out.println("Invalid input. Please enter row and column numbers between 1 and 3.");

scanner.nextLine(); // Clear input buffer

}

Bagian ini menangani exception yang dapat terjadi jika pemain memasukkan input yang tidak valid. Pesan kesalahan akan dicetak, dan buffer input akan dihapus agar pemain dapat memasukkan input yang benar.

* 1. **Fungsi Tambahan**



**Gambar 2.3 Source code fungsi tambahan 1**

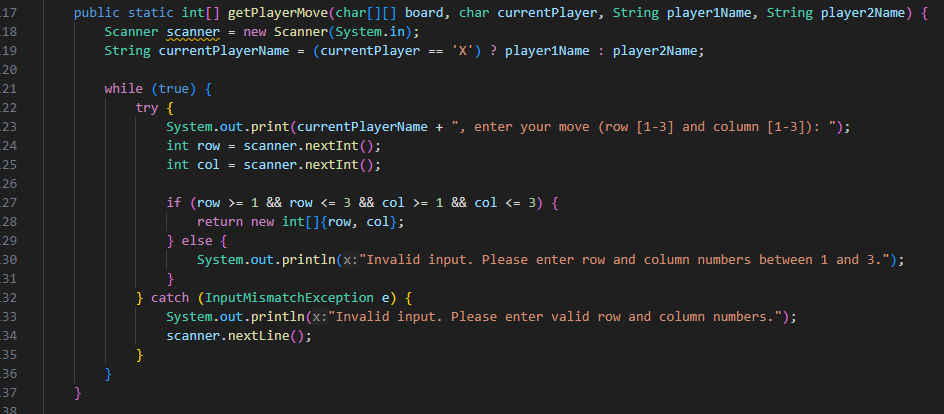
Penjelasan :

public static void initializeBoard(char[][] board)

Fungsi ini bertanggung jawab untuk menginisialisasi papan permainan dengan mengisi semua sel atau kotak pada papan dengan karakter spasi (' '). Papan permainan adalah matriks dua dimensi yang diwakili oleh parameter char[][] board.

public static void printBoard(char[][] board)

Fungsi ini bertanggung jawab untuk mencetak papan permainan ke layar sehingga pemain dapat melihat kondisi papan saat bermain.



**Gambar 2.4 Source code fungsi tambahan 2**

Pejelasan :

public static int[] getPlayerMove(char[][] board, char currentPlayer, String player1Name, String player2Name) {

Ini adalah deklarasi sebuah metode bernama getPlayerMove yang mengambil beberapa parameter, yaitu papan permainan (board), simbol pemain saat ini (currentPlayer), nama pemain 1 (player1Name), dan nama pemain 2 (player2Name).

String currentPlayerName = (currentPlayer == 'X') ? player1Name : player2Name;

Ini adalah pernyataan yang digunakan untuk menentukan nama pemain yang sedang giliran berdasarkan simbol pemain saat ini (currentPlayer). Jika currentPlayer adalah 'X', maka currentPlayerName akan diisi dengan player1Name, dan jika bukan, maka diisi dengan player2Name.

while (true) { ... }

Ini adalah loop tak terbatas yang digunakan untuk terus meminta pemain untuk memasukkan langkah mereka sampai langkah yang valid dimasukkan.

try { ... } catch (InputMismatchException e) { ... }

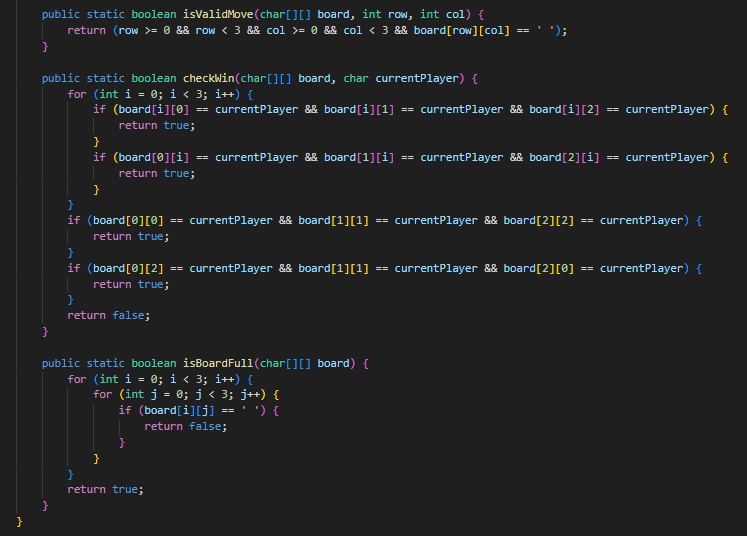
Ini adalah blok try-catch yang digunakan untuk menangani kemungkinan kesalahan saat pemain memasukkan input.

Di dalam blok try, program mencoba untuk melakukan hal berikut:

* Mencetak pesan kepada pemain yang menginstruksikan mereka untuk memasukkan langkah mereka (baris dan kolom).
* Membaca input baris dan kolom yang dimasukkan oleh pemain menggunakan scanner.nextInt().
* Memeriksa apakah input baris dan kolom berada dalam rentang 1 hingga 3. Jika iya, maka langkah yang valid telah dimasukkan, dan program mengembalikan array yang berisi baris dan kolom tersebut.
* Jika input tidak valid, program mencetak pesan kesalahan dan meminta pemain untuk memasukkan langkah lagi.

Di dalam blok catch (InputMismatchException e), program menangani exception InputMismatchException, yang terjadi jika pemain memasukkan input yang tidak valid (misalnya, huruf atau karakter selain angka) :

* Program mencetak pesan kesalahan dan menggunakan scanner.nextLine() untuk membersihkan buffer input sehingga pemain dapat mencoba lagi.



**Gambar 2.5 Source code fungsi tambahan 3**

Penjelasan :

public static boolean isValidMove(char[][] board, int row, int col):

Fungsi ini digunakan untuk memvalidasi apakah langkah yang dimasukkan oleh pemain (baris dan kolom) adalah langkah yang sah. Penjelasan lebih rinci:

* char[][] board: Parameter ini adalah representasi papan permainan Tic-Tac-Toe dalam bentuk matriks dua dimensi.
* int row, int col: Parameter ini adalah baris dan kolom yang ingin diperiksa untuk validitas langkah.
* return (row >= 0 && row < 3 && col >= 0 && col < 3 && board[row][col] == ' ': Fungsi ini mengembalikan true jika langkah yang dimasukkan berada dalam batas papan (baris dan kolom antara 0 hingga 2) dan jika sel di papan tersebut masih kosong (' ').

public static boolean checkWin(char[][] board, char currentPlayer):

Fungsi ini digunakan untuk memeriksa apakah pemain yang sedang bermain saat ini (currentPlayer) telah memenangkan permainan. Penjelasan lebih rinci:

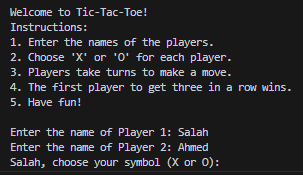
* char[][] board: Parameter ini adalah representasi papan permainan Tic-Tac-Toe dalam bentuk matriks dua dimensi.
* char currentPlayer: Parameter ini adalah simbol (X atau O) dari pemain yang sedang bermain saat ini.
* Fungsi ini melakukan serangkaian pemeriksaan untuk menentukan apakah pemain yang sedang bermain saat ini telah memenangkan permainan. Ini termasuk memeriksa baris, kolom, dan diagonal papan permainan. Jika ada tiga simbol pemain yang sama secara berurutan dalam baris, kolom, atau diagonal, maka pemain tersebut telah memenangkan permainan, dan fungsi ini mengembalikan true. Jika tidak ada pemenang, fungsi ini mengembalikan false.

public static boolean isBoardFull(char[][] board):

Fungsi ini digunakan untuk memeriksa apakah papan permainan telah penuh (tidak ada sel kosong) dan hasil permainan adalah seri (draw). Penjelasan lebih rinci:

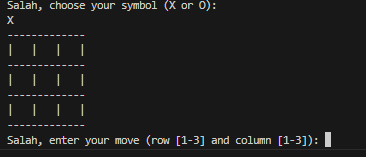
* char[][] board: Parameter ini adalah representasi papan permainan Tic-Tac-Toe dalam bentuk matriks dua dimensi.
* Fungsi ini melakukan iterasi melalui setiap sel di papan permainan dan memeriksa apakah ada setidaknya satu sel yang masih kosong (' '). Jika tidak ada sel kosong, itu berarti papan penuh, dan fungsi ini mengembalikan true. Jika masih ada setidaknya satu sel kosong, itu berarti permainan masih berlanjut, dan fungsi ini mengembalikan false.

**3. Demo Permainan**

****

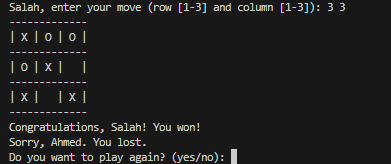
**Gambar 2.6 Demo awal**

Ketika program dijalankan, pemain akan disambut dengan pesan selamat datang dan instruksi permainan. Pemain diminta untuk memasukkan nama Player 1 dan Player 2, serta memilih simbol 'X' atau 'O' untuk masing-masing pemain.



**Gambar 2.7 Demo Memasukkan langkah**

Setelah pemain memasukkan nama dan memilih simbol, permainan dimulai. Papan permainan akan ditampilkan, dan pemain yang giliran akan diminta untuk memasukkan langkah mereka dengan menentukan baris dan kolom tempat mereka ingin menempatkan simbol mereka.



**Gambar 2.8 Demo Akhir permainan**

Setiap langkah pemain akan divalidasi. Jika pemain berhasil mencapai tiga simbol berturut-turut dalam baris, kolom, atau diagonal, program akan mengumumkan pemenangnya. Setelah permainan selesai, pemain akan ditanya apakah mereka ingin bermain lagi. Jika pemain memilih "yes," permainan akan dimulai dari awal dengan pemain yang sama atau pemain yang berbeda.

**KESIMPULAN**

Dalam laporan praktikum ini, penulis telah berhasil merancang dan mengimplementasikan permainan Tic-Tac-Toe menggunakan bahasa pemrograman Java. Permainan ini melibatkan dua pemain yang saling bergantian memasukkan langkah mereka ke dalam papan permainan 3x3. Tujuan utama dari permainan ini adalah untuk mencapai tiga simbol berturut-turut dalam baris, kolom, atau diagonal.

Beberapa konsep pemrograman Java telah diterapkan dalam pengembangan permainan ini, termasuk pemrograman berorientasi objek (OOP), input pengguna, exception handling, dan pengulangan. Papan permainan direpresentasikan sebagai matriks dua dimensi, dan ada fungsi-fungsi penting seperti inisialisasi papan, validasi langkah pemain, pemeriksaan kemenangan, dan pengecekan permainan seri.

Dalam proses pembuatan permainan, penulis juga memastikan agar program dapat menangani berbagai kesalahan input pengguna dengan memberikan pesan kesalahan yang sesuai dan memungkinkan pemain untuk mencoba lagi.

**SARAN**

Dalam pengembangan permainan ini, ada beberapa perbaikan dan pengembangan yang dapat dilakukan seperti pembuatan Tampilan Grafis yang Lebih Menarik, dan Peningkatan Kecerdasan Buatan. Saat ini, permainan menggunakan tampilan teks sederhana. Ini seharusnya dapat dipertimbangkan untuk pengembangan tampilan grafis yang lebih menarik untuk meningkatkan pengalaman pemain. Penulis juga berharap dapat menambahkan mode permainan melawan komputer dengan tingkat kesulitan yang berbeda. Ini akan meningkatkan kompleksitas permainan dan memberikan tantangan tambahan kepada pemain.

**DAFTAR PUSTAKA**

Robby, M. (2019). *Rancang Bangun Aplikasi Point of Sales Berbasis Java pada Toko Barokah* https://repository.um-surabaya.ac.id/548/3/BAB\_II.pdf

Irving, P. (2006). Konsep Permainan Tic-Tac-Toe Menggunakan Algoritma Genetika https://journal.uii.ac.id/Snati/article/view/1603