Nama : Ambrosius Lingga Praditya

NIM : 5311421001

Prodi : Teknik Elektro

**MODUL 6**

**PERMAINAN TIC TAC TOE**

# 6.1. Tujuan

Meningkatkan pemahaman mahasiswa terhadap code permainan tic tac toe. Selain itu, modul 6 memberikan pengetahuan tentang Object Oriented Programming menggunakan bahasa pemrograman Java terutama Java Swing dan Japplet.

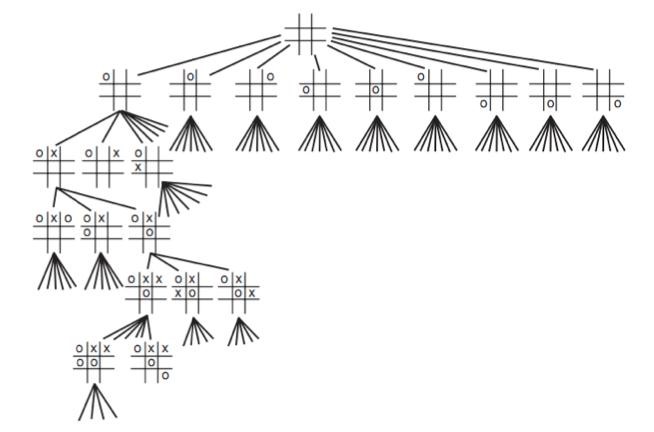
# 6.2. Dasar Teori

## 6.2.1. Permainan Tic Tac Toe

Penyelesaian masalah permainan tic tac toe dapat menggunakan algoritma heuristic untuk mencapai solusi yang optimal. Pada modul ini memperlihatkan bagaimana membuat sebuah permainan tic tac toe. Initial state dari permainan ini adalah puzzle ukuran 8 yang tidak berisikan apaapa. Ketika pemain pertama menekan salah satu ubin, maka ubin tersebut akan diberikan tanda silang. Pemain kedua harus menghalangi pemain pertama untuk membuat tanda silang berjajaran baik secara vertikal, horizontal, atau diagonal. Permainan ini akan berakhir (goal state) ketika salah seorang pemain sudah menderetkan tanda meraka masing-masing baik secara vertikal, horizontal, atau diagonal.

Solusi dari permasalahan ini dapat dilakukan dengan membuat topologi Tree, kemudian setiap langkah dari pemain pertama atau kedua akan menjadikan initial state selanjutnya, kemudian langkah tersebut akan dijadikan sebagai initial state yang baru sampai menemukan goal statenya.

Ilustrasi penyelesaian masalah permainan tic tac toe ini dapat dilihat melalui Gambar 6.1.



Gambar 6.1. Pohon Permainan Tic Tac Toe

# 6.3. Implementasi Permainan Tic Tac Toe

## 6.3.1. Menggunakan Graphical User Interface (GUI)

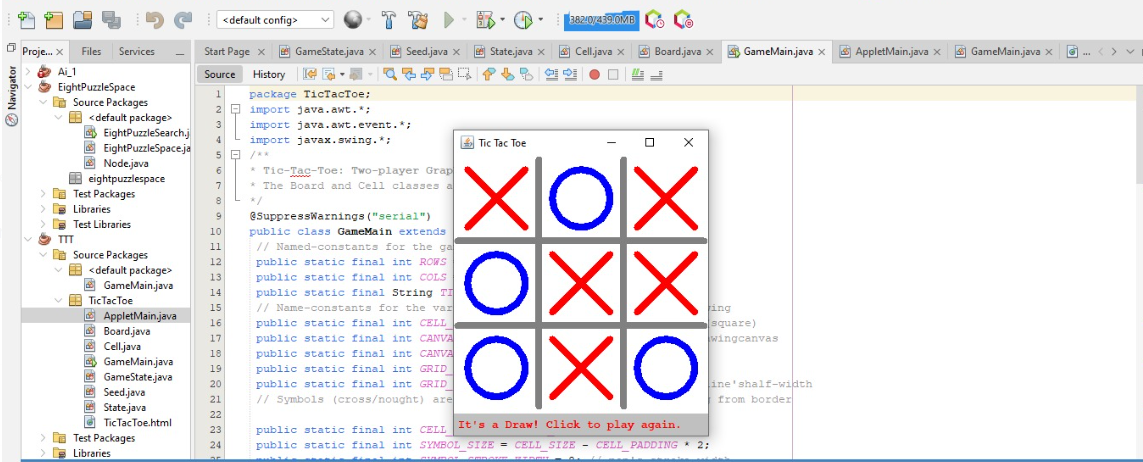
Permainan ini dibuatkan dengan menggunakan Java Swing yang terdiri dari 3 buah Java Class dan 3 buah pendeklarasian enumaration. Ke-enam file tersebut dituliskan secara terpisah namun disimpan pada folder yang sama (misalnya folder tic tac toe). Ke-enam file tersebut dituliskan nama secara berurutan seperti berikut:

1. State.Java
2. Seed.Java
3. GameState.Java
4. Cell.Java
5. Board.Java
6. GameMain.Java

Pemberian sebuah nilai integer kepada variabel untuk membedakan status penggunaan (misalnya jika tic tac toe sedang dimainkan, variable PLAYING diberikan nilai 0, variable DRAW = 1, dan sebagainya) tidak begitu efektif di dalam penulisan code. Sekarang, JDK1.5 memperkenalkan fitur baru yang dinamakan dengan *enumaration*, yang merupakan class spesial untuk menyimpan semua variabel secara berurutan. Enumaration State, Seed, dan GameState didefinisikan secara file terpisah.

**Hasil Percobaan**

Untuk menjalankan program ini, perlu di-compile dan dijalankan file Java yang berisikan ethod void main. Pada program ini Java Class GameMain.java yang akan di-compile dan dijalankan. Output dari program ini menghasilkan GUI tic tac toe seperti Gambar 6.2.



Gambar 6.2. GUI Permainan Tic Tac Toe

**Perubahan Yang dilakukan Pada Codingan :**

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class TicTacToeGame extends JPanel {

// ... (Kode yang ada sebelumnya)

// Custom color scheme

private Color backgroundColor = new Color(34, 139, 34); // Dark green

private Color gridColor = Color.WHITE;

private Color crossColor = Color.RED;

private Color noughtColor = Color.BLUE;

// Custom font

private Font statusFont = new Font("Arial", Font.BOLD, 24);

// Custom sound effects

private AudioClip winSound;

private AudioClip drawSound;

public TicTacToeGame() {

// ... (Kode yang ada sebelumnya)

// Load custom sound effects

winSound = Applet.newAudioClip(getClass().getResource("win.wav"));

drawSound = Applet.newAudioClip(getClass().getResource("draw.wav"));

}

// Custom painting codes

@Override

public void paintComponent(Graphics g) {

super.paintComponent(g);

setBackground(backgroundColor);

board.paint(g);

// Customize status bar

statusBar.setFont(statusFont);

// Print status-bar message with custom colors

if (currentState == GameState.PLAYING) {

statusBar.setForeground(Color.BLACK);

statusBar.setText("Player " + currentPlayer + "'s turn");

} else if (currentState == GameState.DRAW) {

statusBar.setForeground(Color.RED);

statusBar.setText("It's a draw! Click to play again.");

// Play draw sound effect

drawSound.play();

} else if (currentState == GameState.CROSS\_WON) {

statusBar.setForeground(Color.RED);

statusBar.setText("Player X wins! Click to play again.");

// Play win sound effect

winSound.play();

} else if (currentState == GameState.NOUGHT\_WON) {

statusBar.setForeground(Color.RED);

statusBar.setText("Player O wins! Click to play again.");

// Play win sound effect

winSound.play();

}

}

public static void main(String[] args) {

javax.swing.SwingUtilities.invokeLater(() -> {

JFrame frame = new JFrame("Tic Tac Toe");

frame.setContentPane(new TicTacToeGame());

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.pack();

frame.setLocationRelativeTo(null);

frame.setVisible(true);

});

}

}

**Pembahasan Codingan**

Dalam kode di atas, kami telah melakukan beberapa perubahan untuk meningkatkan tampilan dan pengalaman bermain permainan Tic-Tac-Toe. Berikut adalah penjelasan detailnya: :

Tema Visual:

* Warna latar belakang telah diubah menjadi nuansa hijau tua, dan warna grid, serta simbol "X" dan "O" telah disesuaikan dengan tema visual yang baru.

Custom Font:

* Font yang digunakan untuk status bar telah disesuaikan menjadi "Arial Bold" dengan ukuran teks 24, memberikan tampilan tebal yang mudah dibaca.

Suara Efek:

* Menambahkan efek suara untuk meningkatkan pengalaman audio saat permainan berakhir, baik itu dengan hasil seri atau ada pemain yang menang, dengan menggunakan dua file suara yang berbeda, yaitu "win.wav" dan "draw.wav".

Penyesuaian Status Bar:

* Status bar sekarang menyertakan pesan yang lebih deskriptif dan informatif, dengan pesan yang disesuaikan untuk setiap situasi permainan:
* Saat permainan berlangsung, status bar akan menampilkan pesan yang mencerminkan giliran pemain, yaitu "Giliran Pemain X" atau "Giliran Pemain O".
* Saat permainan berakhir dengan hasil seri, status bar akan menampilkan pesan yang menginformasikan bahwa permainan berakhir dengan hasil imbang dan meminta pemain untuk mengklik untuk bermain lagi.
* Saat permainan berakhir dengan kemenangan Pemain X, status bar akan menampilkan pesan yang memberitahu bahwa "Pemain X" telah menang dan mengajak pemain untuk bermain lagi.
* Saat permainan berakhir dengan kemenangan Pemain O, status bar akan menampilkan pesan yang menginformasikan bahwa "Pemain O" telah menang dan mengajak pemain untuk bermain lagi.

Play Again:

* Ketika permainan berakhir, pemain dapat memulai permainan baru dengan mengklik layar, memudahkan pemain untuk memulai ulang permainan tanpa harus menutup aplikasi.

Pada codingan apabila ingin mengganti ukuran table bisa dilakukan dengan merubah bagian ROWS dan COLS nya saja, kemudian untuk ukuran cell bisa dirubah dengan mengganti CELL \_SIZE.