

Nama : Nadia Khoirunisah

Nim : 2222105156

Kelas : 2TI03

Matkul: Pemrograman Berorientasi Objek

Codingan game TicTacToe:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class TicTacToe {
    int boardWidth = 600;
    int boardHeight = 650; //50px for the text panel on top

    JFrame frame = new JFrame("Tic-Tac-Toe");
    JLabel textLabel = new JLabel();
    JPanel textPanel = new JPanel();
    JPanel boardPanel = new JPanel();

    JButton[][] board = new JButton[3][3];
    String playerX = "X";
    String playerO = "O";
    String currentPlayer = playerX;

    boolean gameOver = false;
    int turns = 0;

    TicTacToe() {
        frame.setVisible(true);
        frame.setSize(boardWidth, boardHeight);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
    }
}
```

```
textLabel.setBackground(Color.darkGray);
textLabel.setForeground(Color.white);
textLabel.setFont(new Font(name:"Arial", Font.BOLD, size:50));
textLabel.setHorizontalAlignment(JLabel.CENTER);
textLabel.setText(text:"Tic-Tac-Toe");
textLabel.setOpaque(isOpaque:true);

textPanel.setLayout(new BorderLayout());
textPanel.add(textLabel);
frame.add(textPanel, BorderLayout.NORTH);

boardPanel.setLayout(new GridLayout(rows:3, cols:3));
boardPanel.setBackground(Color.darkGray);
frame.add(boardPanel);

for (int r = 0; r < 3; r++) {
    for (int c = 0; c < 3; c++) {
        JButton tile = new JButton();
        board[r][c] = tile;
        boardPanel.add(tile);

        tile.setBackground(Color.darkGray);
        tile.setForeground(Color.white);
        tile.setFont(new Font(name:"Arial", Font.BOLD, size:120));
        tile.setFocusable(false);
        //tile.setText(currentPlayer);
    }
}
```

```

        tile.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (gameOver) return;
                JButton tile = (JButton) e.getSource();
                if (tile.getText() == "") {
                    tile.setText(currentPlayer);
                    turns++;
                    checkWinner();
                    if (!gameOver) {
                        currentPlayer = currentPlayer == playerX ? playerO : playerX;
                        textLabel.setText(currentPlayer + "'s turn.");
                    }
                }
            }
        });
    }
}

void checkWinner() {
    //horizontal
    for (int r = 0; r < 3; r++) {
        if (board[r][0].getText() == "") continue;

```

```

            if (board[r][0].getText() == board[r][1].getText() &&
                board[r][1].getText() == board[r][2].getText()) {
                for (int i = 0; i < 3; i++) {
                    setWinner(board[r][i]);
                }
                gameOver = true;
                return;
            }

        //vertical
        for (int c = 0; c < 3; c++){
            if (board[0][c].getText() == "") continue;

            if (board[0][c].getText() == board[1][c].getText() &&
                board[1][c].getText() == board[2][c].getText()) {
                for (int i = 0; i < 3; i++) {
                    setWinner(board[i][c]);
                }
                gameOver = true;
                return;
            }
        }

        //diagonally

```

```

if (board[0][0].getText() == board[1][1].getText() &&
    board[1][1].getText() == board[2][2].getText() &&
    board[0][0].getText() != "") {
    for (int i = 0; i < 3; i++) {
        setWinner(board[i][i]);
    }
    gameOver = true;
    return;
}

//anti-diagonally
if (board[0][2].getText() == board[1][1].getText() &&
    board[1][1].getText() == board[2][0].getText() &&
    board[0][2].getText() != "") {
    setWinner(board[0][2]);
    setWinner(board[1][1]);
    setWinner(board[2][0]);
    gameOver = true;
    return;
}

if (turns == 9) {
    for (int r = 0; r < 3; r++) {
        for (int c = 0; c < 3; c++) {
            setTie(board[r][c]);
        }
    }
}

```

```

    }
    gameOver = true;
}

void setWinner(JButton tile) {
    tile.setForeground(Color.green);
    tile.setBackground(Color.gray);
    textLabel.setText(currentPlayer + " is the winner!");
}

void setTie(JButton tile) {
    tile.setForeground(Color.orange);
    tile.setBackground(Color.gray);
    textLabel.setText(text: "Tie!");
}
}

```

Program di atas adalah implementasi permainan Tic-Tac-Toe menggunakan Java Swing. Berikut adalah penjelasan untuk setiap bagian utama dari program ini:

Deklarasi Variabel dan Objek:

- BoardWidth dan boardHeight: Ukuran panel permainan.
- Frame: JFrame sebagai window utama permainan.
- TextLabel: JLabel untuk menampilkan teks permainan dan informasi pemenang.
- TextPanel dan boardPanel: JPanel untuk teks dan panel permainan.
- Board: Array dua dimensi dari JButton untuk tile pada papan permainan.
- Variabel playerX, playerO, currentPlayer: String untuk merepresentasikan pemain X, O, dan giliran pemain saat ini.
- GameOver: Boolean untuk menandakan apakah permainan telah selesai.
- Turns: Integer untuk menghitung jumlah giliran yang telah dilakukan.

Konstruktor TicTacToe():

- Inisialisasi frame dan setting propertinya seperti ukuran, judul, dan layout.
- Inisialisasi textLabel dan menambahkannya ke textPanel.
- Inisialisasi boardPanel dengan layout GridLayout 3x3 untuk menampilkan tile permainan.
- Membuat 9 JButton (tile) dan menambahkannya ke boardPanel. Setiap tile diberi properti seperti background, font, dan action listener untuk menangani klik pemain.

Action Listener untuk Tile:

- Di dalam action listener untuk setiap tile, dicek apakah permainan sudah selesai (gameOver).
- Jika tile belum diisi (getText() kosong), maka tile tersebut diisi dengan simbol pemain saat ini (currentPlayer), jumlah giliran (turns) ditambah, dan kemudian dipanggil checkWinner() untuk memeriksa apakah ada pemenang.
- Jika permainan belum selesai setelah mengisi tile, giliran pemain diubah dan teks label diperbarui dengan giliran pemain berikutnya.

Metode checkWinner():

- Memeriksa kondisi kemenangan secara horizontal, vertikal, diagonal, dan anti-diagonal.

- Jika ada pemain yang memenuhi kondisi untuk menang, pemain tersebut diumumkan sebagai pemenang dengan memanggil `setWinner()`.
- Jika semua tile terisi tanpa ada pemenang (turns mencapai 9), permainan dianggap seri dan diumumkan dengan memanggil `setTie()`.

Metode `setWinner()` dan `setTie()`:

- `setWinner(JButton tile)`: Mengubah warna dan teks tile yang membuat pemain menang, serta mengupdate teks label.
- `setTie(JButton tile)`: Mengubah warna dan teks tile untuk kondisi seri, serta mengupdate teks label.

Program ini menyediakan antarmuka grafis sederhana untuk permainan Tic-Tac-Toe, dengan fitur menangkap input pemain dan memeriksa kondisi kemenangan atau seri secara otomatis.

Hasil codingan game TicTacToe:

