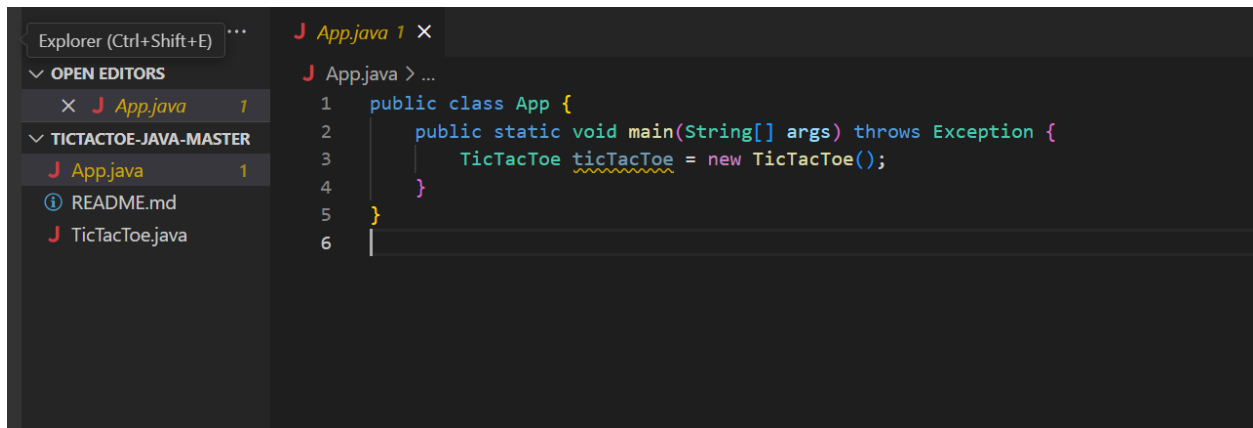


### File App.java

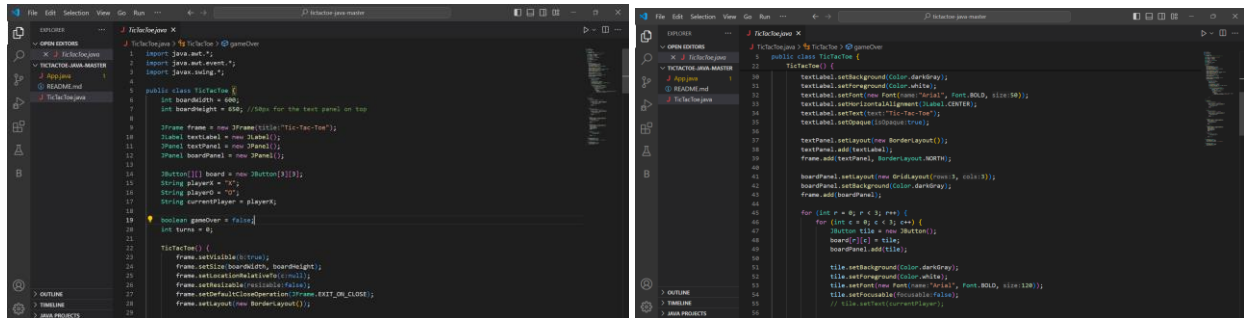


File ini berfungsi sebagai titik awal eksekusi program. Berikut adalah penjelasan singkat:

```
public class App {  
  
    public static void main(String[] args) throws Exception {  
  
        TicTacToe ticTacToe = new TicTacToe();  
  
    }  
  
}
```

- Class Declaration: `public class App` mendeklarasikan sebuah kelas bernama `App`.
- Main Method: `public static void main(String[] args) throws Exception` adalah metode utama yang akan dipanggil ketika program dijalankan. Metode ini adalah titik awal eksekusi program.
- Object Creation: `TicTacToe ticTacToe = new TicTacToe();` membuat instance dari kelas `TicTacToe` dan memulai permainan tic-tac-toe.
- Jadi, `App.java` bertugas untuk memulai aplikasi dengan membuat objek dari kelas `TicTacToe`.

### File `TicTacToe.java`



File ini berisi logika utama dan antarmuka pengguna (GUI) untuk permainan Tic-Tac-Toe. Berikut penjelasan rinci dari setiap bagian dalam file ini:

```
import java.awt.*;
```

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

```
import javax.swing.*;
```

- Mengimpor pustaka AWT (Abstract Window Toolkit) dan Swing untuk pembuatan GUI (Graphical User Interface) dan penanganan event.

```
public class TicTacToe {
```

```
    int boardWidth = 600;
```

```
    int boardHeight = 650; //50px for the text panel on top
```

- Mendeklarasikan kelas `TicTacToe`.
- Mendefinisikan lebar dan tinggi jendela (frame) untuk papan permainan.

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

```
JLabel textLabel = new JLabel();
```

```
JPanel textPanel = new JPanel();
```

```
JPanel boardPanel = new JPanel();
```

- Membuat objek `JFrame` dengan judul "Tic-Tac-Toe".
- Membuat label `JLabel` untuk menampilkan teks.
- Membuat panel `JPanel` untuk menampung label teks dan papan permainan.

```
JButton[][] board = new JButton[3][3];
```

```
String playerX = "X";
```

```
String playerO = "O";
```

```
String currentPlayer = playerX;
```

- Membuat array 2D `JButton` berukuran 3x3 untuk papan permainan.
- Mendefinisikan simbol untuk pemain X dan pemain O.
- Mendefinisikan pemain saat ini, dimulai dengan pemain X.

```
boolean gameOver = false;
```

```
int turns = 0;
```

- `gameOver` digunakan untuk mengecek apakah permainan sudah selesai.
- `turns` digunakan untuk menghitung jumlah giliran yang sudah dimainkan.

```
TicTacToe() {
```

```
frame.setVisible(true);

frame.setSize(boardWidth, boardHeight);

frame.setLocationRelativeTo(null);

frame.setResizable(false);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

frame.setLayout(new BorderLayout());

;
```

- Konstruktor `TicTacToe` untuk menginisialisasi GUI dan permainan.
- Menyetel jendela agar terlihat, menentukan ukurannya, mengatur lokasi di tengah layar, dan mencegah perubahan ukuran.
- Mengatur aksi default saat jendela ditutup dan menentukan tata letak `BorderLayout`.

```
textLabel.setBackground(Color.darkGray);

textLabel.setForeground(Color.white);

textLabel.setFont(new Font("Arial", Font.BOLD, 50));

textLabel.setHorizontalAlignment(JLabel.CENTER);

textLabel.setText("Tic-Tac-Toe");

textLabel.setOpaque(true);
```

- Mengatur latar belakang, warna teks, font, dan perataan label teks.
- Menyetel teks awal menjadi "Tic-Tac-Toe".
- Menetapkan label teks sebagai opaque untuk memastikan warna latar belakang terlihat.

```
textPanel.setLayout(new BorderLayout());
```

```
textPanel.add(textLabel);
```

```
frame.add(textPanel, BorderLayout.NORTH);
```

- Mengatur tata letak panel teks menjadi `BorderLayout`.
- Menambahkan label teks ke panel teks dan menambahkan panel teks ke bagian atas (utara) frame.

```
boardPanel.setLayout(new GridLayout(3, 3));
```

```
boardPanel.setBackground(Color.darkGray);
```

```
frame.add(boardPanel);
```

- Mengatur tata letak panel papan menjadi `GridLayout` berukuran 3x3.
- Menyetel latar belakang panel papan menjadi warna gelap dan menambahkannya ke frame.

```
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("Arial", Font.BOLD, 120));
```

```
tile.setFocusable(false);
```

- Loop bersarang untuk membuat dan menginisialisasi setiap tombol pada papan permainan.
- Menyetel latar belakang, warna teks, font, dan fokus tombol.

```
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.");  
  
            }  
  
        }  
  
    }  
  
});  
  
}  
  
}
```

```
}
```

- Menambahkan `ActionListener` ke setiap tombol untuk menangani klik.
- Jika permainan sudah berakhir, mengabaikan klik.
- Jika tombol belum diisi, mengisi dengan simbol pemain saat ini, menambah jumlah giliran, dan mengecek pemenang.
- Jika belum ada pemenang, mengganti pemain dan mengubah teks label untuk giliran pemain berikutnya.

```
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;
```

```
        }
```

```
    }
```



- Metode `checkWinner` untuk mengecek apakah ada pemenang.
- Mengecek kemenangan horizontal.

```
// 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;

    }

}
```

- Mengecek kemenangan vertikal.

```
// 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;

}

}

```

- Mengecek kemenangan diagonal.
- Mengecek kemenangan anti-diagonal.
- Jika giliran sudah mencapai 9 dan belum ada pemenang, mengatur permainan sebagai seri.

```

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("Tie!");

}

}

```

- Metode `setWinner` mengubah warna tombol pemenang dan menampilkan pesan pemenang.

- Metode `setTie` mengubah warna tombol untuk seri dan menampilkan pesan seri.

## **Ringkasan**

Program ini menciptakan permainan Tic-Tac-Toe berbasis GUI menggunakan Java Swing. App.java berfungsi sebagai titik awal aplikasi, sementara TicTacToe.java berisi logika permainan dan antarmuka pengguna. Program menangani inisialisasi GUI, interaksi pemain, pengecekan pemenang, dan penanganan hasil seri.