

**LAPORAN UJUAN TENGAH SEMESTER  
KOMPUTER DAN PEMROGRAMAN**



**DISUSUN OLEH :**

**Kelompok 2**

Nama : 1. Anisa Julianti (G1A023052)  
2. Revialdi Rifqi Ramadhan P. (G1A023066)

Kelas : B / INFORMATIKA

**Dosen Pengampu**

Arie Vatesia, S.T. M.TI., P.hD

**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS BENGKULU**

**2023**

## Landasan Teori

Java adalah nama sebuah bahasa pemrograman yang sangat terkenal. Sebagai bahasa pemrograman, Java dapat digunakan untuk menulis program. Sebagaimana diketahui, program adalah kumpulan instruksi yang ditujukan untuk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh pemrogram (orang yang membuat program). Sebagai contoh, Anda mungkin terbiasa menggunakan pengolah kata seperti Word. Word adalah contoh aplikasi yang dibuat dengan menggunakan bahasa pemrograman. Java adalah bahasa yang sensitif terhadap huruf kapital dan huruf kecil atau dikenal dengan sebutan case sensitive. Artinya, Java membedakan huruf kapital dan huruf kecil di dalam kode. Sebagai contoh, huruf S kapital harus digunakan pada `System.out.println()`. Jika Anda menggunakan huruf s kecil, perintah tersebut menjadi tidak dikenal oleh kompiler Java.

Program adalah istilah yang biasa dipakai oleh pembuat program, sedang aplikasi adalah istilah dari sisi pemakai (user). Program adalah yang ditulis oleh pemrogram, sedangkan aplikasi adalah hasil terjemahan program, berupa kode yang dipahami oleh mesin. Pada sistem Windows, hasil terjemahan program disimpan di dalam file berekstensi .exe. File inilah yang bisa dijalankan oleh pemakai. Pada Java, hasil terjemahan berupa bytecode. Kode tersebut diproses oleh JRE dan selanjutnya dapat digunakan untuk mengontrol mesin agar melaksanakan tindakan yang telah diprogram oleh pemrogram. Selain kode yang dipahami oleh mesin (biasa disebut kode mesin) ataupun yang berbentuk bytecode, terdapat istilah kode sumber (source code). Kode sumber adalah kode asli yang ditulis oleh pemrogram. Kode dalam bahasa Java yang disimpan dengan ekstensi .java adalah contoh kode sumber.

Bahasa pemrograman berorientasi objek adalah bahasa yang mendukung pembuatan objek, dengan sifat yakni setiap objek mengandung data dan kode untuk mengakses data. Secara lebih khusus, data di objek dinamakan field dan kode untuk mengakses data disebut metode. Sebagai contoh, objek String mempunyai metode bernama `length()` yang berguna untuk mendapatkan jumlah karakter yang tersimpan. Selain field, data di objek sering dinamakan: Atribut dan Variabel instan.

Java menyediakan banyak tipe data. Namun, tipe dasar yang tersedia diperuntukkan untuk bilangan bulat, bilangan real, boolean, karakter, dan string. Bilangan bulat berhubungan dengan bilangan tanpa pecahan. Bilangan real untuk menangani bilangan yang bisa mengandung pecahan. Boolean terkait dengan keadaan benar (true) atau salah (false). Secara umum, boolean digunakan untuk menangani data yang hanya nilainya hanya ada dua keadaan. Karakter berkaitan dengan sebuah huruf, sebuah digit, atau sebuah simbol. Java sejak awal sudah didesain sebagai bahasa pemrograman yang bisa berjalan di berbagai komputer termasuk telepon genggam. Bahasa ini dikembangkan pertama kali oleh James Gosling saat masih bergabung di Sun Microsystems, saat ini Sun sudah diakuisisi oleh Oracle sehingga menjadi bagian dari Oracle.

Bahasa Java awal dirilis tahun 1995. Sintaksnya banyak meniru sintaks yang terdapat pada C dan C++, hanya saja model objek dibuat lebih sederhana serta dukungan rutin-rutin

level bawah yang minimal. Versi awal Java pada tahun 1996 sudah merupakan versi rilis dan bukan versi beta, sehingga dinamakan Java Versi 1.0. Java versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya. Paket-paket yang ada antara lain:

- Java lang: Peruntukan kelas elemen-elemen dasar.
- java.io: Peruntukan kelas input dan output, termasuk penggunaan berkas.
- Java.util: Peruntukan kelas pelengkap kelas kelas penanggalan. seperti kelas struktur data dan
- Java.net: Peruntukan kelas TCP/IP yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
- Java.awt: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
- Java applet: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.

Java adalah bahasa pemrograman yang dirancang tidak tergantung kepada platform alias platform independent, sehingga program yang ditulis di bahasa pemrograman java idealnya harus bisa dijalankan di semua sistem operasi. Karena itu ada JRE atau runtime environment java yang tersedia untuk banyak sistem operasi, sehingga JRE ini memberikan ruang hidup bagi program java. Dengan demikian, programmer tidak perlu bingung menjalankan program java, karena semua JRE sudah tersedia dan tinggal dieksekusi di sistem operasi yang bersangkutan. Jadi walaupun komputer Anda memakai Windows, Linux, atau Mac OS, maka ini bisa menjalankan satu program yang sama. Bahkan teorinya, program yang dikembangkan di Windows bisa dijalankan pula di sistem operasi yang lain, seperti Linux. Ini karena adanya Java Virtual Machine untuk tiap versi sistem operasi yang didukung oleh Java.

Virtual Machine adalah program yang memproses kode Anda sehingga Anda perlu menginstal program ini di tiap platform Anda, sebelum Anda bisa menjalankan kode Java. Jadi kalau mau menjalankan java di Windows, ya harus menginstal JRE untuk Windows, kalau mau menjalankan program java di Linux, harus menginstal JRE untuk Linux.

Program permainan (game) merupakan salah satu implementasi dari bidang ilmu komputer. Perkembangan permainan pada masa kini sudah sangat pesat dan telah menjadi mode tersendiri di dunia karena mayoritas pengguna komputer menghabiskan sebagian besar waktu mereka di depan komputer dalam program permainan. Salah satu algoritma yang digunakan untuk mengembangkan program permainan adalah algoritma berbasis pohon ruang pencarian (searching algorithm). Salah satu game yang menggunakan algoritma berbasis pohon ruang pencarian dalam menyelesaikan permainannya yaitu tic tac toe. Tic tac toe merupakan permainan menyusun potongan gambar dengan aturan sebuah potongan hanya dapat dipindahkan dengan menggesernya ke ruang kosong (blank tile).

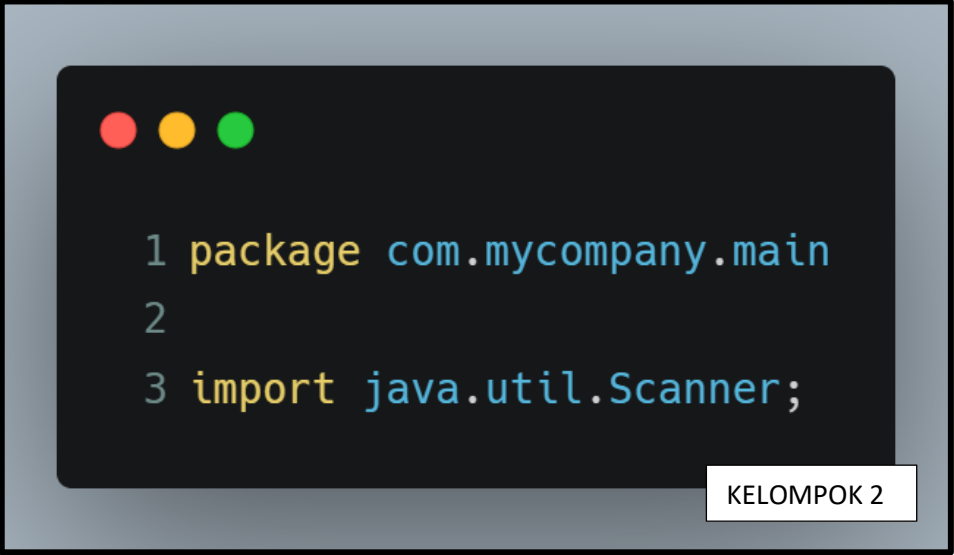
Tic tac toe ini merupakan jenis tic tac toe yang memiliki tingkat kesulitan dalam menyelesaikan masalahnya sangat tinggi dibandingkan jenis tic tac toe lain. Umumnya orang yang memainkan tic tac toe ini butuh waktu lama dalam menyelesaikan permainannya. Hal ini disebabkan karena pada tic tac toe tidak ada informasi tambahan yang dimiliki untuk membantu melakukan pencarian solusi, sehingga saat proses penyusunan potongan-potongan tic tac toe terjadi susunan tic tac toe semula. Untuk menyelesaikan persoalan pada permainan ini dibutuhkan suatu algoritma pencarian efektif yang dapat diterapkan.

## Soal dan Pembahasan

1. Buatlah Game Tictactoe atau Turtle Maze menggunakan java :

### ❖ Package

Printscreen

A screenshot of a code editor window with a dark background and light-colored text. The code consists of three lines: '1 package com.mycompany.main', '2', and '3 import java.util.Scanner;'. The code is highlighted with yellow and blue. In the top left corner of the editor, there are three colored circles (red, yellow, green). In the bottom right corner, there is a white box with the text 'KELOMPOK 2'.

```
1 package com.mycompany.main
2
3 import java.util.Scanner;
```

Gambar 1 Source code

Penjelasan Source Code:

Kode di atas merupakan deklarasi package untuk class MainJava. Biasanya, package digunakan untuk mengatur struktur direktori dalam pengembangan Java. Baris kedua mengimport kelas `Scanner` dari package `java.util`. Kelas ini diperlukan untuk membaca input dari pengguna.

### ❖ Menginisialisasi

Printscreen

A screenshot of a code editor window with a dark background and light-colored text. The code consists of four lines: '1 public class MainJava {', '2 private static char[][] board = new char[3][3];', '3 private static char currentPlayer = 'X';', and '4 }'. The code is highlighted with yellow and blue. In the top left corner of the editor, there are three colored circles (red, yellow, green). In the bottom right corner, there is a white box with the text 'KELOMPOK 2'.

```
1 public class MainJava {
2     private static char[][] board = new char[3][3];
3     private static char currentPlayer = 'X';
4 }
```

Gambar 2 Source code

Penjelasan Source Code:

Kode diatas baris pertama merupakan `MainJava` adalah nama class utama untuk permainan Tic-Tac-Toe. Di dalam class ini, berisi metode-metode dan variabel-variabel yang digunakan dalam permainan. Baris kedua `private static char[][] board = new char[3][3];` mendeklarasikan variabel `board` sebagai array 2D karakter dengan ukuran 3x3. Array ini digunakan untuk merepresentasikan papan permainan Tic-Tac-Toe. Baris ketiga `private static char currentPlayer = 'X';` mendeklarasikan variabel `currentPlayer` sebagai karakter dan menginisialisasinya dengan 'X'. Variabel ini digunakan untuk menyimpan simbol pemain saat ini, yang awalnya diatur sebagai 'X'.

#### ❖ Inisialisasi dan Display

Printscreen



```
1 public static void main(String[] args) {
2     initializeBoard();
3     displayBoard();
4
5     while (true) {
6         play();
7         displayBoard();
8         if (checkWin()) {
9             System.out.println("Player " + currentPlayer + " wins!");
10            break;
11        } else if (isBoardFull()) {
12            System.out.println("It's a draw!");
13            break;
14        }
15        switchPlayer();
16    }
17 }
```

KELOMPOK 2

Gambar 3 Source code

Penjelasan Source Code:

Kode dari baris pertama `main` adalah bagian utama dalam program ini. Pada bagian ini, logika permainan Tic-Tac-Toe diimplementasikan. Baris kedua memanggil metode `initializeBoard()`. Metode ini akan menginisialisasi papan permainan dengan karakter kosong (spasi). Baris ketiga memanggil metode `displayBoard()`. Metode ini akan menampilkan papan permainan secara visual di layar. Baris kelima memulai sebuah loop

`while` yang akan terus berjalan sampai ada break statement yang terpenuhi. Baris keenam memanggil metode `play()`. Metode ini akan meminta pemain saat ini untuk membuat langkahnya pada papan permainan. Baris ketujuh memanggil kembali metode `displayBoard()` untuk menampilkan perubahan papan permainan setelah langkah pemain. Baris selanjutnya pada `if (checkWin()) { ... } else if (isBoardFull()) { ... }` memeriksa apakah pemain saat ini memenangkan permainan atau apakah papan permainan telah penuh. Jika pemain menang, pesan akan ditampilkan dan permainan akan berakhir. Jika papan permainan penuh (tanpa pemenang), pesan draw akan ditampilkan dan permainan akan berakhir. Baris kelima belas memanggil metode `switchPlayer()`. Metode ini akan mengganti pemain saat ini dari 'X' ke 'O' (atau sebaliknya) sebelum memulai siklus loop baru.

#### ❖ Inisialisasi Board

Printscreen



```
1 private static void initializeBoard() {
2     for (int i = 0; i < 3; i++) {
3         for (int j = 0; j < 3; j++) {
4             board[i][j] = ' ';
5         }
6     }
7 }
```

KELOMPOK 2

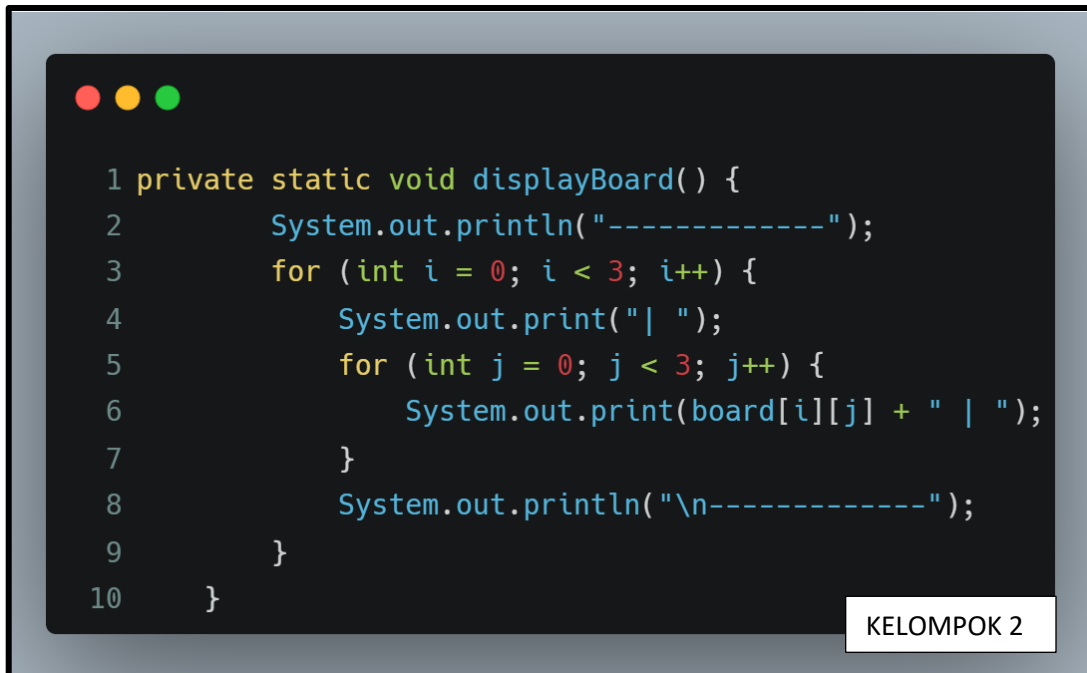
Gambar 4 Source code

#### Penjelasan Source Code:

Kode dari baris pertama `private static void initializeBoard() { ... }` merupakan metode `initializeBoard()` menggunakan nested loop (loop bersarang) untuk mengisi papan permainan dengan karakter kosong (spasi). Setiap sel pada papan diwakili oleh elemen dalam array `board[i][j]`, dan kode ini mengatur nilai setiap sel menjadi spasi kosong ( ' ') untuk mengosongkan papan saat permainan dimulai.

## ❖ Display Board

### Printscreen



```
1 private static void displayBoard() {
2     System.out.println("-----");
3     for (int i = 0; i < 3; i++) {
4         System.out.print("| ");
5         for (int j = 0; j < 3; j++) {
6             System.out.print(board[i][j] + " | ");
7         }
8         System.out.println("\n-----");
9     }
10 }
```

KELOMPOK 2


Gambar 5 Source code

### Penjelasan Source Code:

Kode pada baris pertama `private static void displayBoard() { ... }` merupakan metode `displayBoard()` digunakan untuk menampilkan papan permainan secara visual di layar. Di awal metode ini, garis pembatas horizontal (`"-----"`) ditampilkan. Kemudian, menggunakan nested loop, setiap elemen pada papan permainan ditampilkan di antara dua garis vertikal (`"|"`) dengan spasi di antara keduanya. Baris kedua `System.out.println("-----");` merupakan kode untuk mencetak garis pembatas horizontal pertama untuk papan permainan. Baris keempat `System.out.print("| ");` merupakan kode untuk mencetak garis vertikal pada awal setiap baris papan permainan. Baris keenam `System.out.print(board[i][j] + " | ");` mencetak simbol pada setiap sel papan permainan diikuti dengan garis vertikal. Simbol ini diambil dari array `board` yang mewakili papan permainan. Baris kedelapan `System.out.println("\n-----");` mencetak garis pembatas horizontal setelah setiap baris papan permainan.

❖ Penetapan pengisian pada permainan

Printscreen



```
1 private static void play() {
2     int row, col;
3     do {
4         System.out.println("Player " +
5             currentPlayer + ", enter your move (row and
6             column): ");
7         Scanner scanner = new
8             Scanner(System.in);
9         row = scanner.nextInt() - 1;
10        col = scanner.nextInt() - 1;
11        } while (row < 0 || row >= 3 || col
12        < 0 || col >= 3 || board[row][col] != ' ');
13        board[row][col] = currentPlayer;
14    }
```

KELOMPOK 2

Gambar 6 Source code

Penjelasan Source Code:

Kode baris pertama `private static void play() { ... }` merupakan metode `play()` digunakan untuk mendapatkan langkah pemain saat ini. Pertama, pemain diminta untuk memasukkan langkahnya berupa baris dan kolom pilihan dalam format "baris dan kolom". Kemudian, nilai baris dan kolom dimasukkan melalui input pengguna menggunakan objek `Scanner`. Loop `do-while` digunakan untuk memvalidasi langkah pemain, yaitu memastikan bahwa langkah masukan berada dalam rentang 0-2 dan sel pada papan permainan masih kosong. Jika langkah pemain valid, sel pada papan permainan akan diisi dengan simbol pemain saat ini. Baris kedua `int row, col;` mendeklarasikan variabel `row` dan `col` sebagai tipe data integer untuk menyimpan input baris dan kolom dari pemain. Baris keempat `System.out.println("Player " + currentPlayer + ", enter your move (row and column): ");` mencetak pesan yang meminta



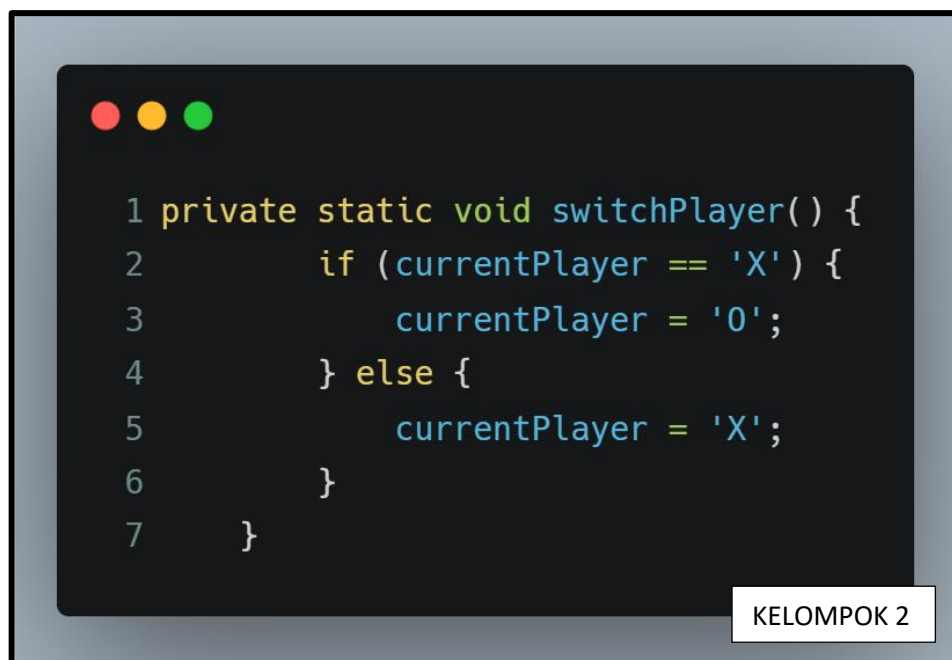
pemain saat ini untuk memasukkan langkahnya. Pesan ini menampilkan simbol pemain saat ini yang sedang melakukan langkah. Baris kelima `Scanner scanner = new Scanner(System.in);` membuat objek `Scanner` yang digunakan untuk membaca input dari pengguna. Baris keenam `row = scanner.nextInt() - 1;` membaca input pengguna untuk baris dan menyimpannya dalam variabel `row`. Pengurangan 1 dilakukan karena input dari pengguna dimulai dari 1, sedangkan indeks array dimulai dari 0. Baris ketujuh `col = scanner.nextInt() - 1;` membaca input pengguna untuk kolom dan menyimpannya dalam variabel `col`. Pengurangan 1 dilakukan karena input dari pengguna dimulai dari 1, sedangkan indeks array dimulai dari 0. Baris kedelapan `while (row < 0 || row >= 3 || col < 0 || col >= 3 || board[row][col] != ' ');` Loop `do-while` akan terus berjalan selama salah satu dari kondisi berikut terpenuhi:

- Baris atau kolom yang dimasukkan berada di luar rentang valid yaitu 0-2.
- Sel pada papan permainan yang dipilih oleh pemain tidak kosong.

Baris kesepuluh `board[row][col] = currentPlayer;` menetapkan simbol pemain saat ini pada sel papan permainan yang dipilih dengan baris dan kolom yang dimasukkan oleh pemain.

#### ❖ Switch Player

##### Printscreen



Gambar 7 Source code

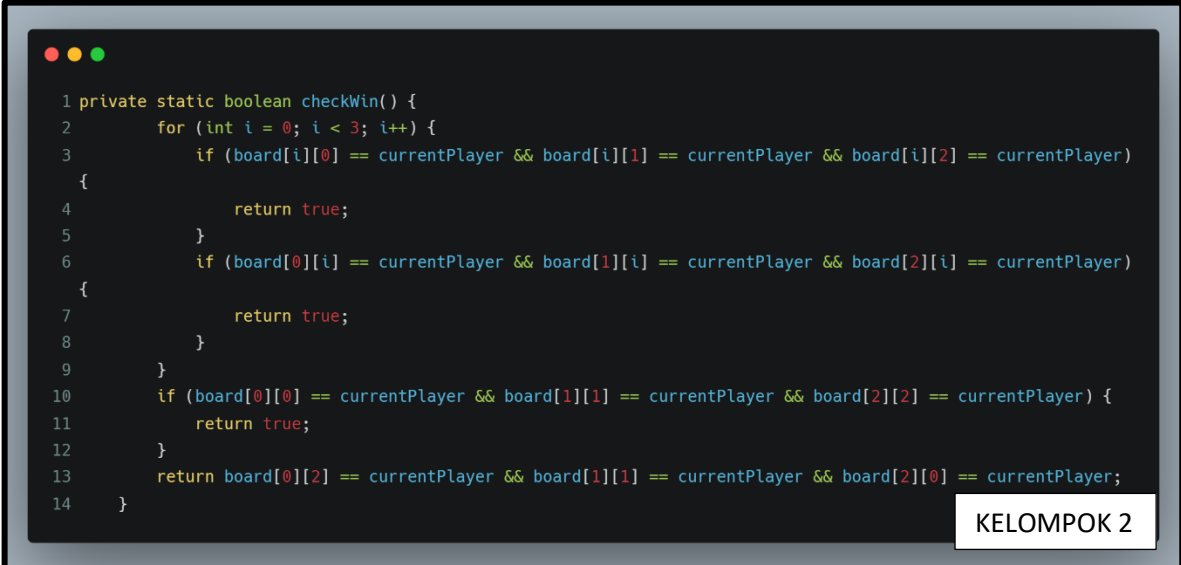
##### Penjelasan Source Code:

Kode pada baris pertama `private static void switchPlayer() { ... }` merupakan metode `switchPlayer()` digunakan untuk mengganti pemain saat ini setelah setiap putaran permainan.

Jika pemain saat ini adalah 'X', maka variabel `currentPlayer` akan diubah menjadi 'O', dan sebaliknya. Tujuan dari metode ini adalah untuk memastikan pergantian antara pemain 'X' dan 'O' setelah tiap putaran. Baris kedua `if (currentPlayer == 'X') { ... } else { ... }` merupakan kondisi yang memeriksa nilai dari variabel `currentPlayer`. Jika `currentPlayer` saat ini adalah 'X', maka blok kode dalam kurung kurawal pertama akan dieksekusi, yaitu mengubah nilai `currentPlayer` menjadi 'O'. Jika `currentPlayer` saat ini bukan 'X', maka blok kode dalam kurung kurawal kedua akan dieksekusi, yaitu mengubah nilai `currentPlayer` menjadi 'X'.

#### ❖ Check Win

##### Printscreen



```
1 private static boolean checkWin() {
2     for (int i = 0; i < 3; i++) {
3         if (board[i][0] == currentPlayer && board[i][1] == currentPlayer && board[i][2] == currentPlayer)
4         {
5             return true;
6         }
7         if (board[0][i] == currentPlayer && board[1][i] == currentPlayer && board[2][i] == currentPlayer)
8         {
9             return true;
10        }
11        if (board[0][0] == currentPlayer && board[1][1] == currentPlayer && board[2][2] == currentPlayer) {
12            return true;
13        }
14        return board[0][2] == currentPlayer && board[1][1] == currentPlayer && board[2][0] == currentPlayer;
15    }
16 }
```

KELOMPOK 2

Gambar 8 Source code

#### Penjelasan Source Code:

Baris pertama `private static boolean checkWin() { ... }` metode `checkWin()` digunakan untuk memeriksa apakah ada pemain yang sudah memenangkan permainan. Metode ini menggunakan loop `for` untuk melakukan pengecekan pada setiap baris dan kolom, serta diagonal pada papan permainan. Jika pemain saat ini ('X' atau 'O') memiliki tiga simbol berturut-turut di satu baris, satu kolom, atau salah satu diagonal, maka metode ini akan mengembalikan nilai `true`, yang menandakan bahwa ada pemain yang menang. Jika tidak ada pemain yang menang, maka metode ini akan mengembalikan nilai `false`. Loop `for` pada baris kedua digunakan untuk melakukan pengecekan pada baris dan kolom papan permainan. Variabel `i` digunakan sebagai indeks dari baris atau kolom yang sedang diperiksa. Kondisi pada baris ketiga `if (board[i][0] == currentPlayer && board[i][1] == currentPlayer &&

board[i][2] == currentPlayer) { ... }` memeriksa apakah pemain saat ini memiliki tiga simbol ('X' atau 'O') berturut-turut di satu baris. Jika kondisi ini terpenuhi, maka metode akan mengembalikan nilai `true`. Kondisi pada baris keenam `if (board[0][i] == currentPlayer && board[1][i] == currentPlayer && board[2][i] == currentPlayer) { ... }` memeriksa apakah pemain saat ini memiliki tiga simbol berturut-turut di satu kolom. Jika kondisi ini terpenuhi, maka metode akan mengembalikan nilai `true`. Kondisi baris kesepuluh `if (board[0][0] == currentPlayer && board[1][1] == currentPlayer && board[2][2] == currentPlayer) { ... }` memeriksa apakah pemain saat ini memiliki tiga simbol berturut-turut di diagonal dari pojok kiri atas ke pojok kanan bawah. Jika kondisi ini terpenuhi, maka metode akan mengembalikan nilai `true`. Baris ketiga belas `return board[0][2] == currentPlayer && board[1][1] == currentPlayer && board[2][0] == currentPlayer;` mengembalikan `true` jika pemain saat ini memiliki tiga simbol berturut-turut di diagonal dari pojok kanan atas ke pojok kiri bawah. Jika tidak terpenuhi, maka metode akan mengembalikan nilai `false`.

❖ Memeriksa papan / board

Printscreen



```
1 private static boolean isBoardFull() {
2     for (int i = 0; i < 3; i++) {
3         for (int j = 0; j < 3; j++) {
4             if (board[i][j] == ' ') {
5                 return false;
6             }
7         }
8     }
9     return true;
10 }
```

KELOMPOK 2

Gambar 9 Source code

Penjelasan Source Code:

Kode pada baris ini merupakan metode `isBoardFull()` digunakan untuk memeriksa apakah papan permainan sudah penuh atau belum. Metode ini melakukan pengecekan dengan mengiterasi semua sel pada papan permainan. Jika setidaknya ada satu sel yang masih kosong (tidak memiliki simbol 'X' atau 'O'), maka metode ini akan mengembalikan nilai `false`. Jika semua sel pada papan permainan sudah terisi, maka metode ini akan mengembalikan nilai `true`, yang menandakan bahwa permainan berakhir imbang karena tidak ada pemenang.

- Metode `isBoardFull()` digunakan untuk memeriksa apakah papan permainan sudah penuh atau belum. Metode ini menggunakan nested loop `for` untuk iterasi melalui setiap sel pada papan permainan. Jika ada setidaknya satu sel yang masih kosong (dinyatakan dengan karakter spasi ' '), artinya papan permainan belum penuh, dan metode ini akan mengembalikan `false`. Jika tidak ada sel yang kosong pada papan permainan, artinya papan permainan sudah penuh, dan metode ini akan mengembalikan `true`.

## **Kesimpulan dan Saran**

### **A. KESIMPULAN**

Dari penjelasan mengenai pemrograman membuat game Tictactoe diatas, dapat disimpulkan bahwa:

1. Java dikenal sebagai bahasa pemrograman yang bersifat strongly, yang berarti diharuskan mendeklarasikan tipe data dari semua variable yang apabila lupa atau salah dalam mengikuti aturan pendeklarasian variable, maka akan terjadi error pada saat proses kompilasi.
2. Bahasa pemrograman atau bahasa penataolahan adalah sistem notasi untuk menulis program komputer. Sebagian besar bahasa pemrograman adalah bahasa formal berdasarkan teks, tetapi mungkin juga berupa grafis.
3. Game Tic Tac Toe terdiri dari 3 baris dan 3 kolom. Pemenangnya ditentukan oleh pemain yang lebih dulu membentuk sebuah baris yang terdiri dari 3 kotak. Baris dapat berupa horisontal, vertikal, maupun diagonal. Selain disajikan dalam bentuk multiplayer, Tic Tac Toe juga disajikan dalam bentuk melawan komputer. Game Tic Tac Toe dibangun menggunakan PHP sebagai web server, Javascript, dan MySQL sebagai database.

### **B. SARAN**

Setelah mempelajari mengenai Java, Pemrograman Java, Pemrograman Game Tictactoe maka saran yang dapat diberikan adalah:

1. Melakukan analisis dan desain perangkat lunak: Sebelum memulai pengembangan game, penting untuk melakukan analisis kebutuhan, memahami fitur yang diharapkan, serta merancang struktur dan alur program dengan jelas. Hal ini akan membantu dalam mencapai tujuan yang diinginkan dan menghindari kesalahan perancangan yang dapat mempengaruhi kualitas akhir permainan.
2. Memiliki pengujian yang komprehensif: Lakukan pengujian secara menyeluruh untuk memastikan game Tictactoe dapat berjalan dengan baik dan sesuai dengan harapan. Ujilah berbagai skenario mulai dari input valid hingga kasus-kasus batas. Hal ini akan membantu dalam mengidentifikasi dan memperbaiki bug yang mungkin muncul.
3. Mempelajari pemrograman game yang lebih kompleks: Setelah mempelajari dan mengembangkan game Tictactoe, para mahasiswa dapat mencoba mengeksplorasi pemrograman game yang lebih kompleks. Misalnya, mengembangkan game dengan grafika yang lebih menarik, integrasi suara, dan fitur lainnya. Hal ini akan memperkaya pemahaman mereka tentang pemrograman game dan tantangan yang terlibat.

## **Daftar Pustaka**

Kadir, A. (2014). *Buku Pertama Belajar Pemrograman Java*. Media Pressindo.

Sianipar, R. H. (2015). *Pemrograman Java untuk Programmer* (Vol. 1). Penerbit ANDI.

Tim, E. M. S. (2015). *Pemrograman Java Dari Nol*. Elex Media Komputindo.