**LAPORAN UJUAN TENGAH SEMESTER**

**PEMROGRAMAN INTERNET & E-COMMERCE**



**DISUSUN OLEH :**

Nama : 1. Bavio Robia Rahmadan

2. Muhammad Ripal Rabbani   
NPM : 1. G1A023002

2. G1A023064

Kelas : B

# **DOSEN PENGAMPU :**

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

**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”.

* 1. **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.

* 1. **Arsetiktur 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 memungkinkan pemakai untuk menjalankan program Java, hanya menjalankan, tidak untuk membuat kode baru lagi. JRE berisi JVM dan pustaka Java yang digunakan.

* 1. **Java 2**

Sun Microsystems telah mendefinisikan tiga buah edisi dari Java 2, yaitu sebagai berikut :

**1. Java 2 Standard Edition (J2SE)**, adalah inti dari bahasa pemrograman Java. JDK merupakan salah satu perangkat (tool) dari J2SE untuk mengkompilasi dan menjalankan program Java. Di dalamnya terdapat 10 perangkat untuk mengkompilasi program Java dan JRE. J2SE ini digunakan pada perangkat keras seperti layar komputer (desktop).

**2. Java 2 Enterprise Edition (J2EE),** merupakan kumpulan tertinggi (superset) dari J2SE yang memperbolehkan kita untuk mengembangkan aplikasi-aplikasi berskala besar (enterprise) karena dijalankan pada jaringan komputer.

**3. Java 2 Micro Edition (J2ME)**, merupakan kumpulan bagian (subset) dari J2SE yang digunakan untuk menangani pemrograman di dalam perangkat perangkat kecil, yang tidak memungkinkan untuk mendukung implementasi J2SE secara penuh. Paket J2ME digunakan pada perangkat yang memiliki kapasitas memori kecil seperti telepon selular, pager atau PDA.

* 1. **Java 2 Micro Edition (J2ME)**

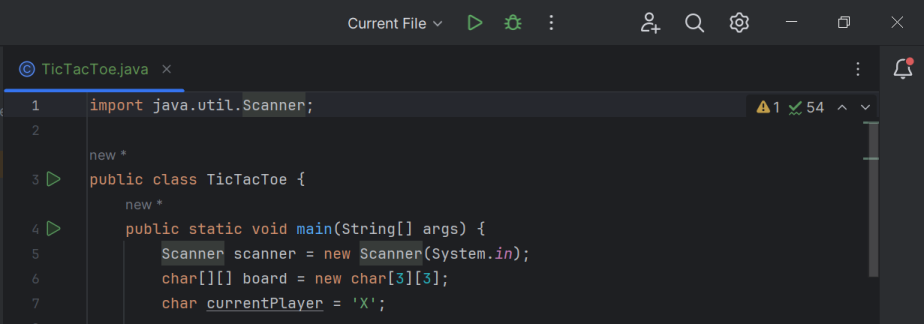
**A.Pengenalan Java 2 Micro Edition (J2ME)**

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada perangkat lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer dekstop yang biasanya lebih kecil dibandingkan perangkat komputer dekstop. J2ME biasa digunakan pada telepon seluler, pager, personal digital assistants (PDA‟s) dan sejenisnya.

Soal dan Pembahasan

Game Tictactoe atau Turtle Maze menggunakan java :

1. Halaman Utama



Gambar 1 Source code

Penjelasan Source Code:

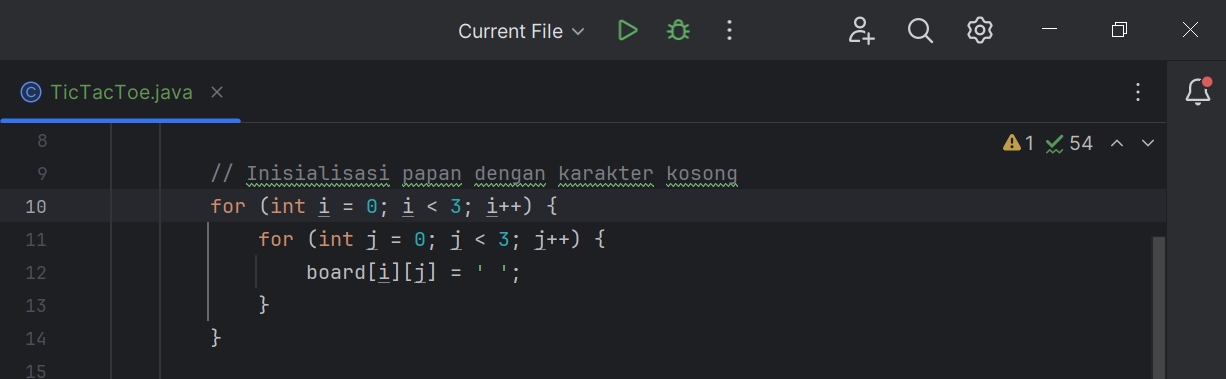
Kode di atas merupakan contoh kode dalam pembuatan halaman utama dari game TicTacToe. Kode `import java.util.Scanner;` digunakan untuk mengimpor kelas `Scanner` dari pustaka Java `java.util`. Ini memungkinkan kita untuk menggunakan kelas `Scanner` dalam kode Java kita. Kelas `Scanner` digunakan untuk membaca masukan dari pengguna melalui keyboard atau dari sumber lain, seperti berkas teks.

Baris kode Scanner scanner = new Scanner(System.in); dalam bahasa pemrograman Java digunakan untuk membuat sebuah objek dari kelas Scanner yang akan digunakan untuk membaca masukan dari pengguna melalui keyboard atau sumber lain yang terhubung dengan System.in

Baris kode char[][] board = new char[3][3]; digunakan untuk membuat sebuah array dua dimensi (matriks) dari tipe data karakter (char) dengan ukuran 3x3. Ini adalah contoh awal dari pembuatan papan permainan (misalnya, papan permainan Tic-Tac-Toe) atau matriks untuk menyimpan data dalam bentuk tabel dua dimensi.

Char currentPlayer = 'X'; digunakan untuk mendefinisikan dan menginisialisasi variabel currentPlayer dengan karakter 'X'. Ini adalah cara Anda menentukan pemain saat ini dalam permainan yang menggunakan karakter 'X' dan 'O', seperti permainan Tic-Tac-Toe..

2. Inisialisasi papan dengan karakter kosong



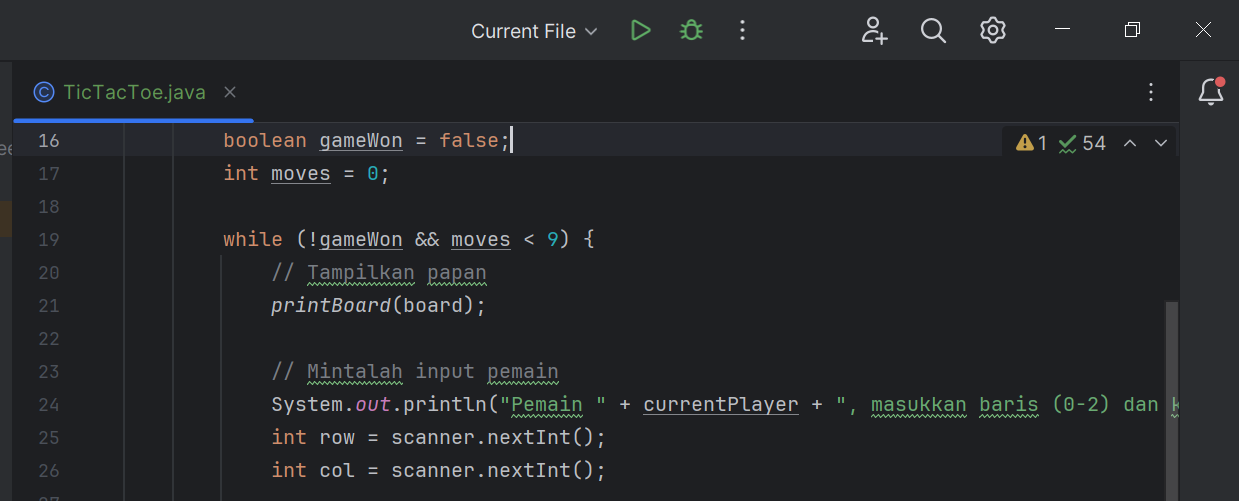
Gambar 2 Source code

Penjelasan :

Kode for (int i = 0; i < 3; i++) { adalah awal dari sebuah perulangan (loop) dalam bahasa pemrograman Java. Perulangan ini menggunakan struktur for untuk mengulangi blok kode yang ada di dalamnya sebanyak tiga kali dengan menggunakan variabel i sebagai penghitung. pada awal perulangan ini, int i = 0; digunakan untuk menginisialisasi variabel i dengan nilai awal 0. Kemudian, i < 3; adalah kondisi yang harus terpenuhi agar perulangan terus berjalan. Artinya, perulangan akan terus berlanjut selama nilai i kurang dari 3. Setiap kali perulangan selesai, nilai i akan bertambah satu. Jadi, perulangan ini akan menjalankan blok kode di dalamnya sebanyak tiga kali, di mana i akan memiliki nilai 0, 1, dan 2 pada setiap iterasi .

Kode for (int j = 0; j < 3; j++) { adalah bagian dari perulangan bersarang (nested loop) dalam bahasa pemrograman Java. Perulangan ini digunakan untuk mengulangi blok kode yang ada di dalamnya sebanyak tiga kali dengan menggunakan variabel j sebagai penghitung.Pada awal perulangan ini, int j = 0; digunakan untuk menginisialisasi variabel j dengan nilai awal 0. Kemudian, j < 3; adalah kondisi yang harus terpenuhi agar perulangan terus berjalan. Jadi, perulangan ini akan menjalankan blok kode di dalamnya sebanyak tiga kali, di mana j akan memiliki nilai 0, 1, dan 2 pada setiap iterasi.

3.Tampilan papan dan minta input pemain



Gambar 3. Source Code

Penjelasan :

Kode boolean gameWon = false; digunakan untuk mendefinisikan dan menginisialisasi variabel gameWon dengan nilai false. Ini berarti awalnya permainan dianggap belum dimenangkan (atau belum ada pemenang) karena gameWon memiliki nilai false.

Kode int moves = 0; digunakan untuk mendefinisikan dan menginisialisasi variabel moves dengan nilai 0. Ini adalah cara untuk melacak jumlah langkah atau tindakan yang telah terjadi dalam permainan. Dalam konteks permainan, gameWon biasanya akan digunakan untuk menentukan apakah ada pemenang dalam permainan (misalnya, jika kondisi kemenangan tertentu terpenuhi), dan moves akan digunakan untuk melacak berapa banyak langkah atau tindakan yang telah dilakukan dalam permainan tersebut.

Kode while (!gameWon && moves < 9) { berfungsi untuk menjalankan blok kode di dalamnya selama permainan belum dimenangkan (variabel gameWon adalah false) dan jumlah langkah belum mencapai batas maksimum (9 langkah). Ini adalah cara umum untuk mengendalikan jalannya permainan hingga ada pemenang atau permainan berakhir dalam situasi seri.

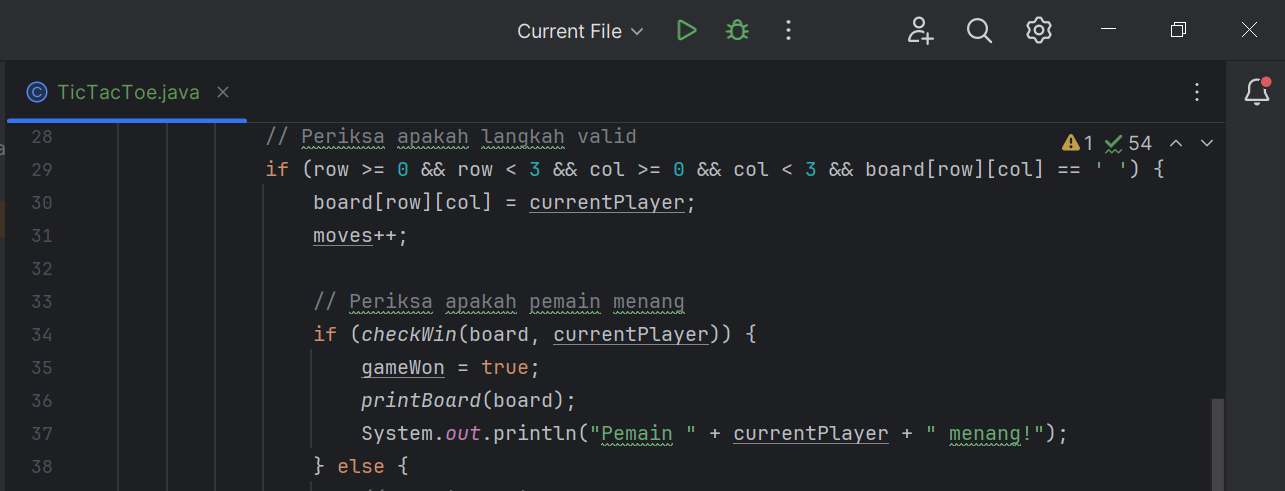
Kode printBoard(board); adalah pemanggilan fungsi atau metode yang disebut printBoard dengan argumen board. Fungsi ini bertujuan untuk mencetak atau menampilkan papan permainan (atau papan lainnya) ke layar atau keluaran yang sesuai.

Umumnya, printBoard akan berisi logika untuk mencetak matriks atau struktur data lainnya ke layar dalam format yang mudah dibaca. Dalam konteks permainan seperti Tic-Tac-Toe, ini akan digunakan untuk mencetak status papan permainan saat ini sehingga pemain atau pengguna dapat melihatnya.

Kode System.out.println("Pemain " + currentPlayer + ", masukkan baris (0-2) dan kolom (0-2) yang ingin Anda isi:"); digunakan untuk mencetak pesan ke layar yang meminta pemain saat ini untuk memasukkan baris dan kolom di mana mereka ingin menempatkan tanda mereka di papan permainan.

Kode int row = scanner.nextInt(); Dan int col = scanner.nextInt(); ini digunakan untuk membaca masukan dari pengguna (pemain saat ini) menggunakan objek scanner yang telah dibuat sebelumnya dengan menggunakan kelas Scanner. Lebih khusus, dua variabel int, yaitu row dan col, digunakan untuk menyimpan masukan baris dan kolom yang dimasukkan oleh pemain.

4. Periksa apakah langkah valid dan Pemain menang



Gambar 4. Source Code

Penjelasan :

Kode row >= 0 dan row < 3: Ini memeriksa apakah input pemain untuk baris (row) berada dalam rentang 0 hingga 2, yang sesuai dengan indeks yang valid untuk papan 3x3. Jika input pemain berada di luar rentang ini, langkah tidak valid.

Kode col >= 0 dan col < 3: Ini mirip dengan yang sebelumnya, tetapi memeriksa input pemain untuk kolom (col).

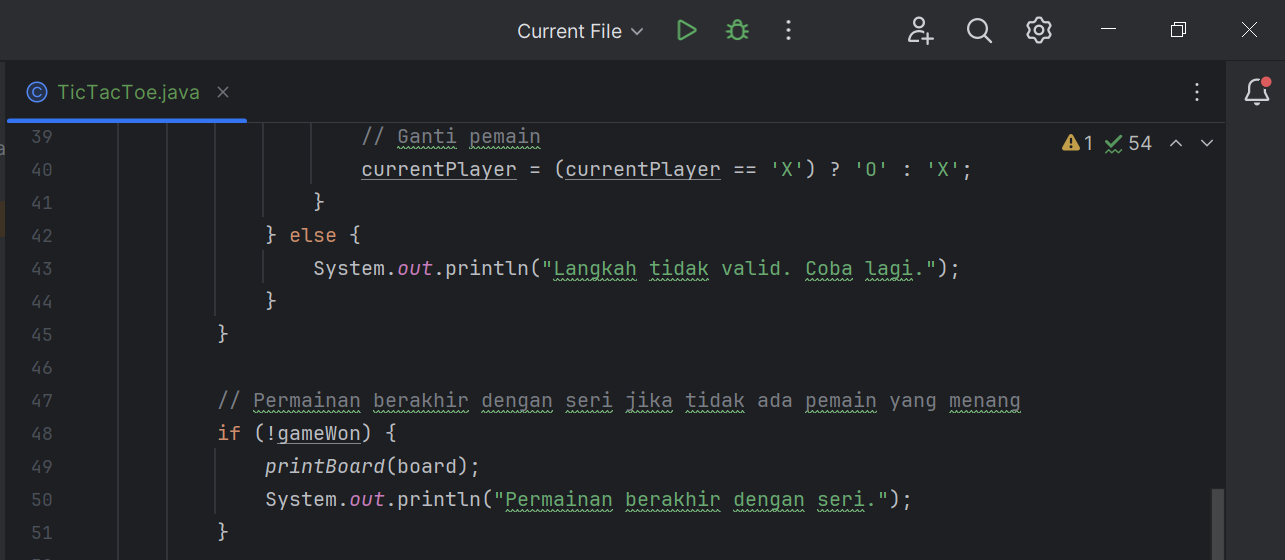
Kode board[row][col] == ‘ ‘: Ini memeriksa apakah sel di baris dan kolom yang dimasukkan oleh pemain masih kosong (dalam kondisi awal). Jika sel tersebut sudah diisi oleh pemain sebelumnya, langkah tidak valid.

Kode checkWin(board, currentPlayer): Ini adalah panggilan fungsi checkWin yang memeriksa apakah pemain saat ini telah mencapai kondisi kemenangan. Isi dari fungsi checkWin akan bervariasi tergantung pada implementasi Anda. Biasanya, Anda akan memeriksa semua kondisi kemenangan yang mungkin, seperti baris, kolom, dan diagonal yang berisi tiga simbol pemain yang sama. Jika checkWin mengembalikan true, artinya pemain saat ini telah menang, maka variable gameWon diatur ke true untuk menandakan bahwa permainan telah dimenangkan.

Kode printBoard(board) untuk menampilkan papan saat permainan berakhir.

(currentPlayer) sebuah pesan dicetak yang memberi tahu pemain mana yang telah menang dengan menyertakan ymbol pemain.

5. Ganti pemain dan Berakhir dengan seri



Gambar 5. Source Code

Penjelasan :

Kode currentPlayer = (currentPlayer == 'X') ? 'O' : 'X'; adalah contoh penggunaan operator ternary (ternary operator atau conditional operator) untuk mengganti giliran pemain dalam permainan Tic-Tac-Toe.

Jika currentPlayer saat ini adalah 'X' (currentPlayer == 'X' adalah true), maka operator ternary akan mengganti currentPlayer menjadi 'O' ('O').

Jika currentPlayer saat ini bukan 'X' (artinya 'O'), maka operator ternary akan mengganti currentPlayer menjadi 'X'.

Dengan cara ini, setelah setiap langkah, permainan akan menggantikan giliran pemain secara otomatis. Ini memungkinkan pemain bergantian melakukan langkah mereka dalam permainan Tic-Tac-Toe.

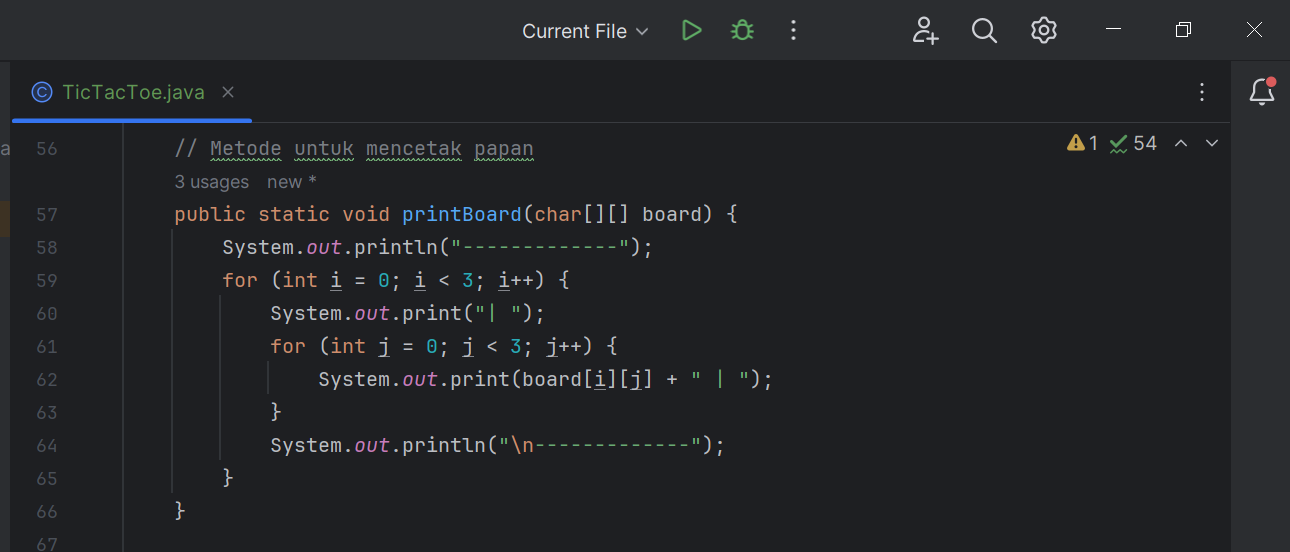
Kode System.out.println("Langkah tidak valid. Coba lagi."); berfungsi untuk memberikan umpan balik kepada pemain saat mereka mencoba melakukan langkah yang tidak valid dalam permainan Tic-Tac-Toe. Umpan balik ini penting untuk menginformasikan pemain bahwa langkah yang mereka coba tidak dapat diterima, dan mereka harus mencoba lagi.

Kode !gameWon: Ini adalah kondisi yang memeriksa apakah variabel gameWon tidak diatur ke true. Dalam konteks permainan Tic-Tac-Toe, gameWon akan diatur ke true jika salah satu pemain telah memenangkan permainan. Dengan kata lain, ini memeriksa apakah permainan belum dimenangkan oleh salah satu pemain. Jika gameWon adalah false, artinya permainan belum dimenangkan oleh salah satu pemain, maka kode di dalam blok if akan dieksekusi.

Kode printBoard(board): Ini mencetak papan permainan saat permainan berakhir. Ini memungkinkan pemain melihat papan permainan terakhir sebelum permainan berakhir dengan seri.

Kode System.out.println("Permainan berakhir dengan seri.");: Ini mencetak pesan yang memberi tahu pemain bahwa permainan berakhir dengan hasil seri.

6. Metode untuk mencetak papan



Gambar 6. Source Code

Penjelasan Source Code :

Kode printBoard adalah bagian dari kode untuk mencetak papan permainan Tic-Tac-Toe ke layar konsol.

Kode for (int i = 0; i < 3; i++) {

System.out.print("| "); berfungsi untuk mencetak garis pembatas vertikal di antara sel-sel papan permainan dalam bentuk garis vertikal "|" pada layar konsol. Ini digunakan untuk memisahkan sel-sel dalam baris papan permainan.

Loop for ini berjalan tiga kali karena i diberi nilai awal 0 dan berhenti ketika i tidak lagi kurang dari 3.

Setiap kali loop berjalan, System.out.print("| ") digunakan untuk mencetak karakter "|" (garis vertikal) diikuti oleh spasi kosong ke layar konsol. Ini menciptakan efek garis vertikal yang memisahkan sel-sel dalam baris papan permainan.

Akibatnya, ketika loop ini selesai, Anda akan memiliki tiga garis vertikal "|" yang tercetak berdampingan, seperti ini: | | |

Kode ini digunakan dalam metode printBoard untuk membantu mencetak garis vertikal yang memisahkan setiap sel dalam baris papan permainan, memberikan tampilan yang jelas dan terstruktur pada layar konsol.

Kode for (int j = 0; j < 3; j++) {

System.out.print(board[i][j] + " | "); berfungsi untuk mencetak isi sel-sel papan permainan dalam satu baris ke layar konsol. Ini digunakan untuk menampilkan konten papan permainan, yaitu tanda 'X', 'O', atau spasi kosong (' '), serta memisahkannya dengan karakter "|" (garis vertikal) untuk memberikan tampilan papan permainan yang terstruktur.

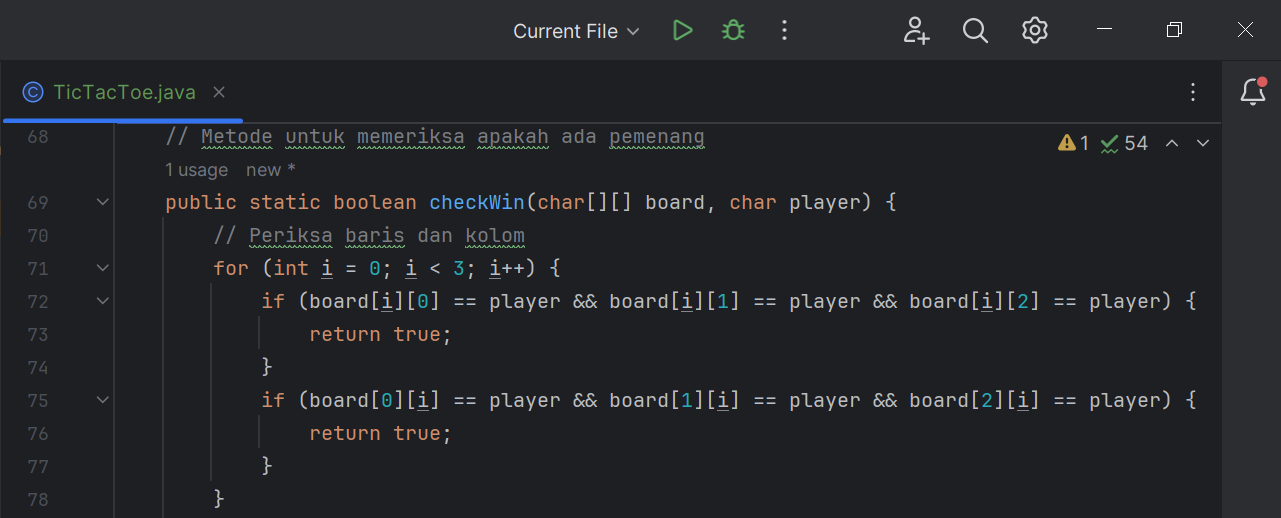
Loop for ini berjalan tiga kali karena j diberi nilai awal 0 dan berhenti ketika j tidak lagi kurang dari 3.

Setiap kali loop berjalan, board[i][j] digunakan untuk mengakses isi sel papan permainan pada baris i dan kolom j.

Kode System.out.print(board[i][j] + " | ") digunakan untuk mencetak isi sel papan permainan ke layar konsol, diikuti oleh karakter "|" (garis vertikal) dan spasi kosong. Ini menciptakan tampilan sel yang terpisah dalam satu baris.

Akibatnya, setelah loop ini selesai, Anda akan memiliki tampilan baris papan permainan dengan sel-sel yang memiliki isi dan dipisahkan oleh karakter "|"

7. Periksa apakah ada pemenang dan Periksa baris kolom



Gambar 7. Source Code

Penjeladan :

Metode checkWin memiliki fungsi untuk memeriksa apakah pemain yang ditentukan (dalam parameter player) telah memenangkan permainan Tic-Tac-Toe dengan melihat papan permainan (board). Ini adalah bagian penting dari logika permainan yang memutuskan pemenang atau hasil imbang.

Kode for (int i = 0; i < 3; i++) { berfungsi untuk melakukan iterasi (pengulangan) sejumlah tiga kali (dalam hal ini) dengan menggunakan variabel penghitung i yang dimulai dari 0 dan berakhir ketika i tidak lagi kurang dari 3.

int i = 0; Ini adalah inisialisasi variabel penghitung i dengan nilai awal 0.

i < 3; Ini adalah kondisi yang harus dipenuhi agar loop terus berjalan. Loop akan berlanjut selama nilai i kurang dari 3.

i++ Ini adalah langkah iterasi yang akan dijalankan setiap kali iterasi selesai. Dengan i++, nilai i akan meningkat sebanyak satu setiap kali iterasi berlangsung.

Kode if (board[i][0] == player && board[i][1] == player && board[i][2] == player) {

return true; dan if (board[0][i] == player && board[1][i] == player && board[2][i] == player) { return true; adalah bagian dari metode checkWin dalam permainan Tic-Tac-Toe yang berfungsi untuk memeriksa apakah pemain yang ditentukan (dalam parameter player) telah memenangkan permainan dalam baris tertentu pada papan permainan (board).

board[i][0], board[i][1], dan board[i][2] adalah sel dalam baris tertentu pada papan permainan. Misalnya, jika i adalah 0, ini akan mengacu pada baris pertama (indeks 0).

board[i][0] == player memeriksa apakah sel pertama dalam baris tersebut berisi simbol pemain yang sedang diperiksa (dalam parameter player).

board[i][1] == player memeriksa apakah sel kedua dalam baris tersebut juga berisi simbol pemain yang sama.

board[i][2] == player memeriksa apakah sel ketiga dalam baris tersebut juga berisi simbol pemain yang sama.

board[0][i], board[1][i], dan board[2][i] adalah sel dalam kolom tertentu pada papan permainan. Misalnya, jika i adalah 0, ini akan mengacu pada kolom pertama (indeks 0).

board[0][i] == player memeriksa apakah sel pertama dalam kolom tersebut berisi simbol pemain yang sedang diperiksa (dalam parameter player).

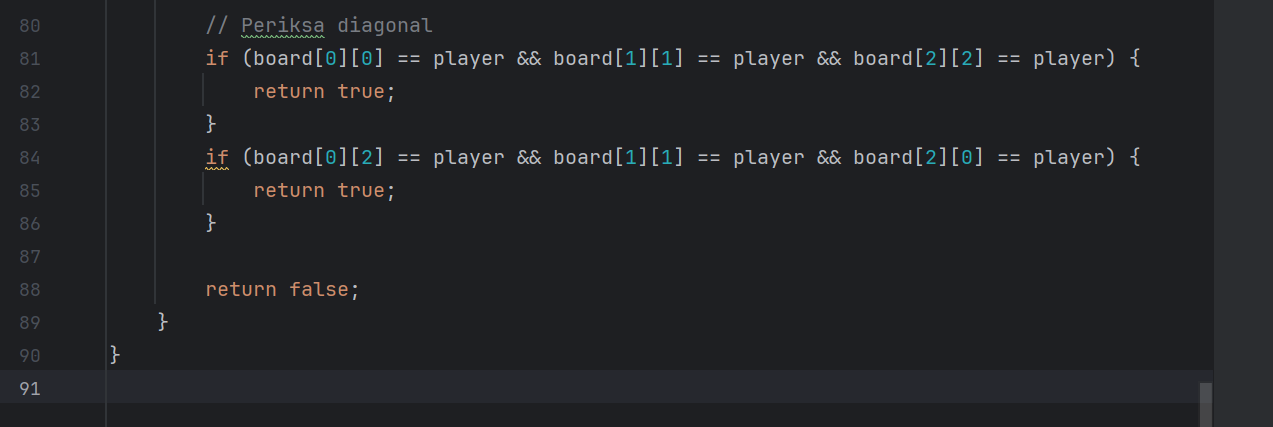
board[1][i] == player memeriksa apakah sel kedua dalam kolom tersebut juga berisi simbol pemain yang sama.

board[2][i] == player memeriksa apakah sel ketiga dalam kolom tersebut juga berisi simbol pemain yang sama.

Jika ketiga kondisi tersebut terpenuhi, maka ini berarti semua tiga sel dalam baris tersebut memiliki simbol pemain yang sama (semua 'X' atau semua 'O'). Dalam konteks Tic-Tac-Toe, ini merupakan salah satu kondisi kemenangan karena pemain telah berhasil menempatkan tanda mereka dalam satu baris yang sama.

Ketika kondisi ini terpenuhi, return true; digunakan untuk mengakhiri pemanggilan metode checkWin dan mengembalikan true, yang akan menunjukkan bahwa pemain telah memenangkan permainan dalam baris tertentu. kode ini adalah contoh cara memeriksa kemenangan dalam satu baris pada papan permainan dalam TicTacToe.

8. Periksa diagonal



Gambar 8. Source Code

Penjelasan :

Kode board[0][0] == player && board[1][1] == player && board[2][2] == player dan board[0][2] == player && board[1][1] == player && board[2][0] == player berfungsi untuk memeriksa apakah pemain yang ditentukan (dalam parameter player) telah memenangkan permainan dalam diagonal utama (dari kiri atas ke kanan bawah) pada papan permainan Tic-Tac-Toe.

board[0][0] adalah sel di sudut kiri atas papan permainan.

board[1][1] adalah sel di tengah papan permainan.

board[2][2] adalah sel di sudut kanan bawah papan permainan.

board[0][2] adalah sel di sudut kanan atas papan permainan.

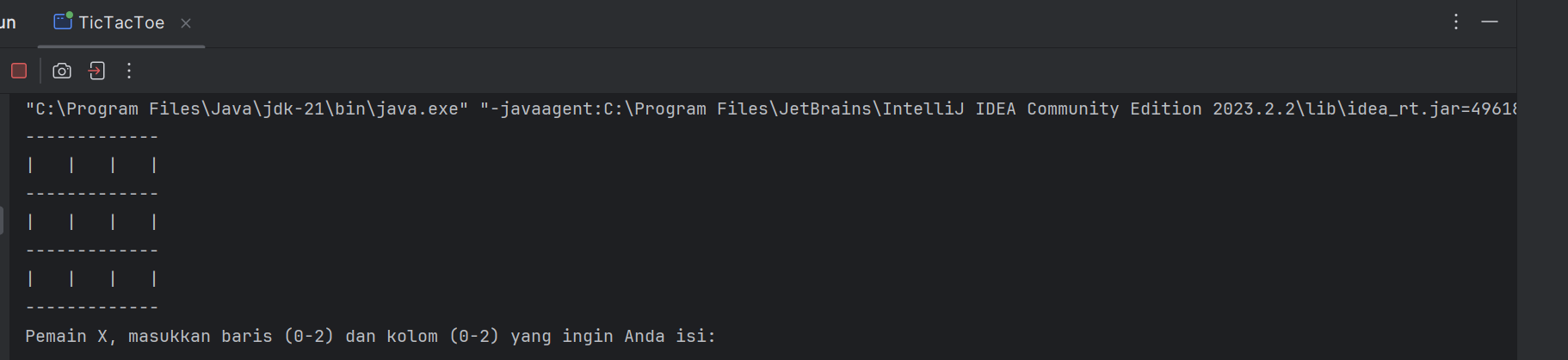
board[1][1] adalah sel di tengah papan permainan.

board[2][0] adalah sel di sudut kiri bawah papan permainan.

Kondisi board[0][0] == player && board[1][1] == player && board[2][2] == player dan board[0][2] == player && board[1][1] == player && board[2][0] == player memeriksa apakah ketiga sel ini memiliki simbol pemain yang sama (semua 'X' atau semua 'O'). Jika ketiga kondisi ini terpenuhi, ini berarti pemain telah berhasil menempatkan tanda mereka dalam diagonal utama, yang merupakan salah satu kondisi kemenangan dalam permainan Tic-Tac-Toe.

Ketika kondisi ini terpenuhi, return true; digunakan untuk mengakhiri pemanggilan metode checkWin dan mengembalikan true, yang menunjukkan bahwa pemain telah memenangkan permainan dengan menguasai diagonal utama papan permainan.

9. Output Papan

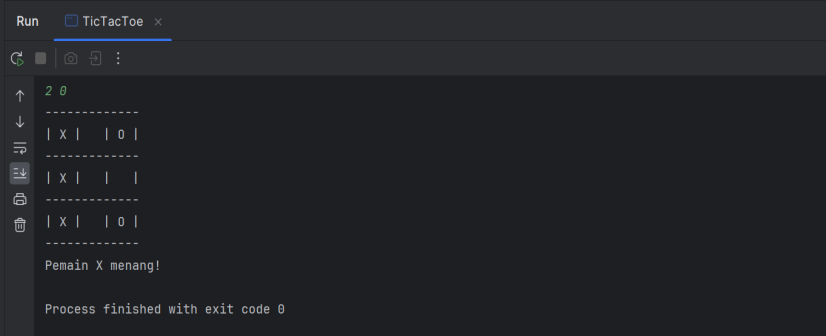


Gambar 9. Screen Output

Penjelasan :

Saat ini, pemain X memiliki giliran. Papan memiliki 3 baris (0, 1, dan 2) dan 3 kolom (0, 1, dan 2) yang diberi nomor. Pada saat ini, pemain X harus memilih di mana dia ingin menempatkan tanda X-nya dengan memasukkan nomor baris dan kolom yang sesuai (misalnya, "0 0" untuk menempatkan X di baris 0, kolom 0)

10. Output X Win

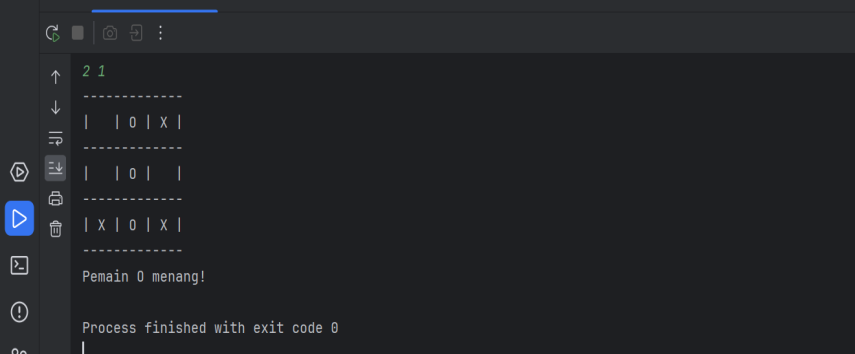


Gambar 10. Screen Output

Penjelasan :

Output tersebut menunjukkan bahwa pemain X telah menang dalam permainan Tic Tac Toe dengan menempatkan tanda X di semua kolom di baris pertama.

11. Output O Win



Gambar 11. Screen Output

Penjelassan :

Output tersebut menunjukkan bahwa pemain O telah menang dalam permainan Tic Tac Toe dengan menempatkan tanda O di semua kolom di baris Kedua.

# **Kesimpulan dan Saran**

1. **Kesimpulan**

Kesimpulan merupakan rangkuman dari seluruh hasil penelitian yang telah dilakukan oleh penulis, beberapa kesimpulan yang berguna jedepannya untuk pengembangan maupun masukan bagi pembaca dan penulis sendiri. Adapun kesimpulan yang didapat dari penelitian ini dapat dijelaskan sebagai berikut:

1. Pemograman java dapat diimplementasikan dalam berbagai bentuk dukungan sistem, penelitian ini menghasilkan aplikasi pengkodean pada database serta neatbeans sebagai bahasa pemograman.

2. Pada saat aplikasi dijalankan untuk mengenkripsi database, maka proses yang terjadi pada database yang telah dienkripsi akan mengalami perubahan.

3. Menggunakan algoritma knapsackmerupakan algoritma asimetris yaitu dalam penggunaan kunci enkripsi dan deskripsi berbeda. Kunci untuk enkripsi tidak sama dengan kunci deskripsi.

4. Aplikasi yang penulis rancang dengan Netbeans yang dapat memudahkan pengguna(user) dalam manjalankan aplikasi dan hanya bisa mengenkripsi data pada database.

**B. Saran**

Adapun saran yang penulis kemukakan adalah sebagai berikut:

1. Pada aplikasi pengenkripsian database dengan menggunakan algoritma knapsack ini masih perlunya pengembangan di kemudian hari, untuk pengembangan yang lebih lanjut dibutuhkan keamanan dengan algoritma-algoritma yang berbeda sebagai pilihan pengenkripsian database, seperti algoritma brute force, algoritma rsa, dan algoritma lainnya.
2. Desain interface dapat dimanfaatkan oleh pengembang untuk membuat desain interface yang menarik perhatian pengguna.
3. Pengguna harus mengingat nilai superincreasing dan nilai n dari program ini.
4. Aplikasi ini tidak belum berjalan dijalur HTTPS sehingga masih memungkinkan pencurian data terjadi. Diharapkan kedepannya aplikasi ini sudah support berjalan di jalur HTTPS sehingga data jauh lebih aman.

# **Daftar Pustaka**

Rickyanto, Isak. 2003. Dasar Pemrograman Berorientasi Objek Dengan Java. Yogyakarta: Andi.

Nofriadi. 2015. Java Fundamental Dengan Netbeans 8.0.2.

Hendri. 2008. Belajar Otodidak Java Dengan Netbeans. Jakarta: Elex Media Komputindo.

Nugroho, Adi. 2009. Rekayasa Perangkat Lunak Menggunakan UML dan Java. Jogjakarta: CV. Andi Mengimbangi.

Agung, Leo. 2012. Aplikasi Pemrograman Javascript untuk Halaman Web. Yogyakarta: Andi Mengimbangi.

Suyanto. 2015. Pemrogaman Java: Pengenalan Java. Jurnal Komputer.