**LAPORAN UJUAN TENGAH SEMESTER**

**COMPUTER AND PROGRAMMING**



**DISUSUN OLEH :**

Nama : LIO KUSNATA

NPM : G1A023013

Kelas : A

**Asisten Dosen**Randi Julian Saputra (G1A019066)

# **Dosen Pengampu**

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

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**

**Tahun ajaran 2023/2024**

# **Landasan Teori**

Sejarah terciptanya Java pertama kali yaitu yang dipelopori oleh James Gosling, Patrick Naughton, Chris Warth, Ed Frank, dan Mike Sheridan dari Sun Microsystem pada tahun 1991. Sebagai bagian dari suatu proyek penelitian untuk mengembangkan perangkat lunak bagi consumer barang-barang elektronik seperti televisi, VCR, toaster dan mesin- mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan portable di segala jenis hardware. Java merupakan suatu bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain.

Bahasa Pemrograman Java ini berorientasi objek Object Oriented Programming (OOP), dan dapat dijalankan pada berbagai platform sistem operasi, Menurut Nofriadi (2015:1) ”Bahasa Pemrograman Java Merupakan Salah satu dari sekian banyak bahasa pemograman yang dapat dijalankan di berbagai sistem operasi termasuk telepon genggam”, sedangkan menurut Supardi (2010:1) “java merupakan Bahasa Pemrograman yang dikembangan dari bahasa pemrograman C++, Sehingga bahasa pemrograman ini seperti bahasa pemrograman C++”.

Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program-program yang dapat dijalankan melalui www dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program-program yang dapat digunakan dengan mudah dan portable di berbagai platform yang berbeda. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi (Maya, 2015).

Dapat di simpulkan bahwa bahasa pemograman java adalah perkembangan dari bahasa pemrograman C++ yang dimana bahasa pemograman java jauh lebih dapat diandalkan sehingga java bisa membuat seluruh bentuk pemograman dari aplikasi desktop hingga ke aplikasi mobile, sehingga java sangat cocok digunakan sebagai bahasa pemograman untuk kalangan programmer yang menggunakannya. Java dapat didefinisikan sebagai bahasa pemrograman yang berorientasi objek dan multi platform yang digunakan untuk mengembangkan perangkat lunak. Terpisah secara geografis adalah masalah integritas, kepemilikan, dan kemacetan (deadlock) ketika banyak processor mengupdate data yang sama. Java adalah bahasa pemrograman yang sangat mudah ditransfer yang digunakan di seluruh platform dan berbagai tipe perangkat, mulai dari smartphone hingga TV pintar.

Java digunakan untuk membuat aplikasi seluler dan web, perangkat lunak perusahaan, perangkat Internet of Things (IoT), game, big data, dan aplikasi terdistribusi berbasis cloud serta berbagai aplikasi lainnya. Berikut adalah beberapa contoh aplikasi di dunia nyata tertentu yang diprogram dengan Java.Java adalah bahasa pemrograman multi platform. Ini berarti bahwa Java dapat ditulis untuk satu OS dan berjalan di OS lain. Kode Java pertama kali ditulis dalam Java Development Kit, yang tersedia untuk Windows, Linux, dan macOS.

Programmer menulis dalam bahasa pemrograman Java, yang kemudian diterjemahkan kit menjadi kode komputer yang dapat dibaca oleh perangkat apa pun yang memiliki perangkat lunak yang tepat. Ini dicapai dengan perangkat lunak yang disebut pengompilasi. Pengompilasi mengambil kode komputer tingkat tinggi seperti Java dan menerjemahkannya ke dalam bahasa yang dipahami sistem operasi yang disebut bytecode. Bytecode kemudian diproses oleh interpreter yang disebut mesin virtual Java (JVM). JVM tersedia untuk sebagian besar platform perangkat lunak dan perangkat keras, dan inilah yang memungkinkan kode Java ditransfer dari satu perangkat ke perangkat lain. Untuk menjalankan Java, JVM memuat kode, memverifikasi, dan menyediakan lingkungan runtime.Java dikenal dengan moto “Write Once, Run Anywhere” yang memiliki arti bahwa Java adalah bahasa pemrograman yang mampu dijalankan di berbagai platform tanpa perlu penyesuaian ulang di platformnya. Contohnya, dapat dijalankan di Android, Linux, Windows, dan lain-lain. Alasan kenapa Java dapat dijalankan di mana saja, karena bahasa pemrograman ini memiliki sistem syntax atau kode pemrograman level tertinggi.

Syntax akan disusun dengan Java Virtual Machine (JVM) menjadi kode numeric (bytescode) platform, sehingga aplikasi Java dapat dijalankan di berbagai perangkat. Singkatnya, Java adalah sebuah bahasa pemrograman. Biasanya digunakan untuk pengembangan bagian back-end dari software, aplikasi Android, dan juga website. Java merupakan perubahan luas dari pemrograman yang terdistribusi dan pengembangan aplikasi yang mengambil tempat di industri komputer saat ini. Bahasa seperti java secara radikal menggeser beban komputasi dari komputer desktop lokal ke server yang memberikan konten yang dieksekusi untuk penggunanya.

Ketika bahasa pemrograman alami pertama kali dikembangkan, bahasa-bahasa tersebut terbagi menjadi dua kategori, bergantung pada caranya berkomunikasi dengan perangkat keras yang mendasarinya.

* Kompilator: Program yang lengkap ditulis dalam sintaks yang mirip Bahasa Inggris alami dengan kompilator, lalu bahasa akan mengompilasikan (atau menerjemahkan) seluruh kode ke dalam kode mesin. Kode yang telah dikompilasikan kemudian dijalankan di perangkat keras.
* Interpreter: Dengan interpreter, setiap pernyataan kode tingkat tinggi diinterpretasikan ke dalam kode mesin ketika proses berjalan. Pernyataan tertulis dijalankan langsung oleh perangkat keras sebelum melihat pernyataan berikutnya.

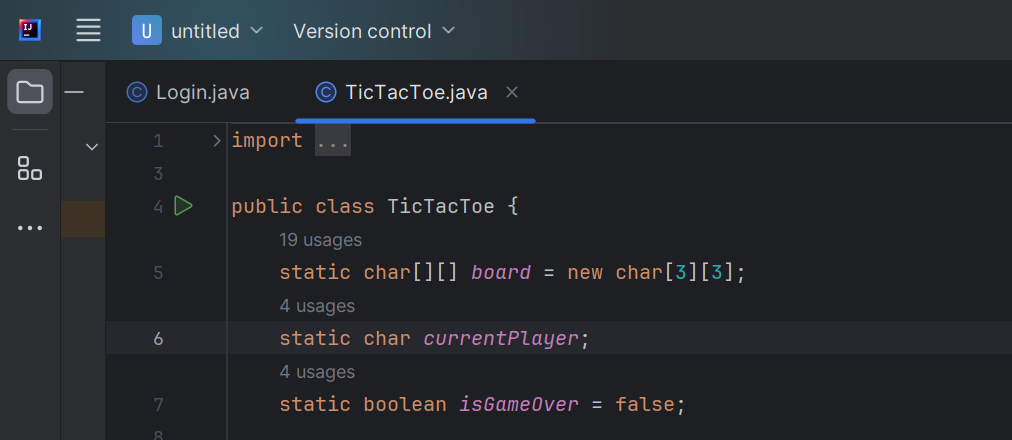
Adapun fungsi Java adalah sebagai berikut:

1. Mendukung Native Method merupakan salah satu fungsi dari Java. Dengan adanya dukungan ini, maka Java memungkinkan programmer dapat menulis berbagai fungsi dengan cepat dibandingkan yang lain.
2. Bahasa yang digunakan sederhana, Dikarenakan menggunakan sintaks yang bisa dibilang mirip dengan C++, maka hal tersebut membuat Java lebih sederhana dibandingkan dengan bahasa pemrograman lainnya.
3. Bahasa pemrograman Java berfokus pada objek. Artinya, program komputer dapat saling melakukan komunikasi dalam satu kelompok objek.
4. Pengamanan yang cukup ketat di mana Java memiliki pengaman tiga lapis. Hal ini dikarenakan untuk melindungi sistem dari orang-orang yang tidak bertanggung jawab.
5. Java dapat digunakan pada sistem operasi manapun. Hal ini dikarenakan bahasa Java termasuk ke dalam Platform Independence, sehingga saat dibuat sebuah file maka file tersebut dapat dijalankan di mana saja.

**Soal dan Pembahasan**

**Game TicTacToe menggunakan java :**

1. Halaman Utama



G1A023013

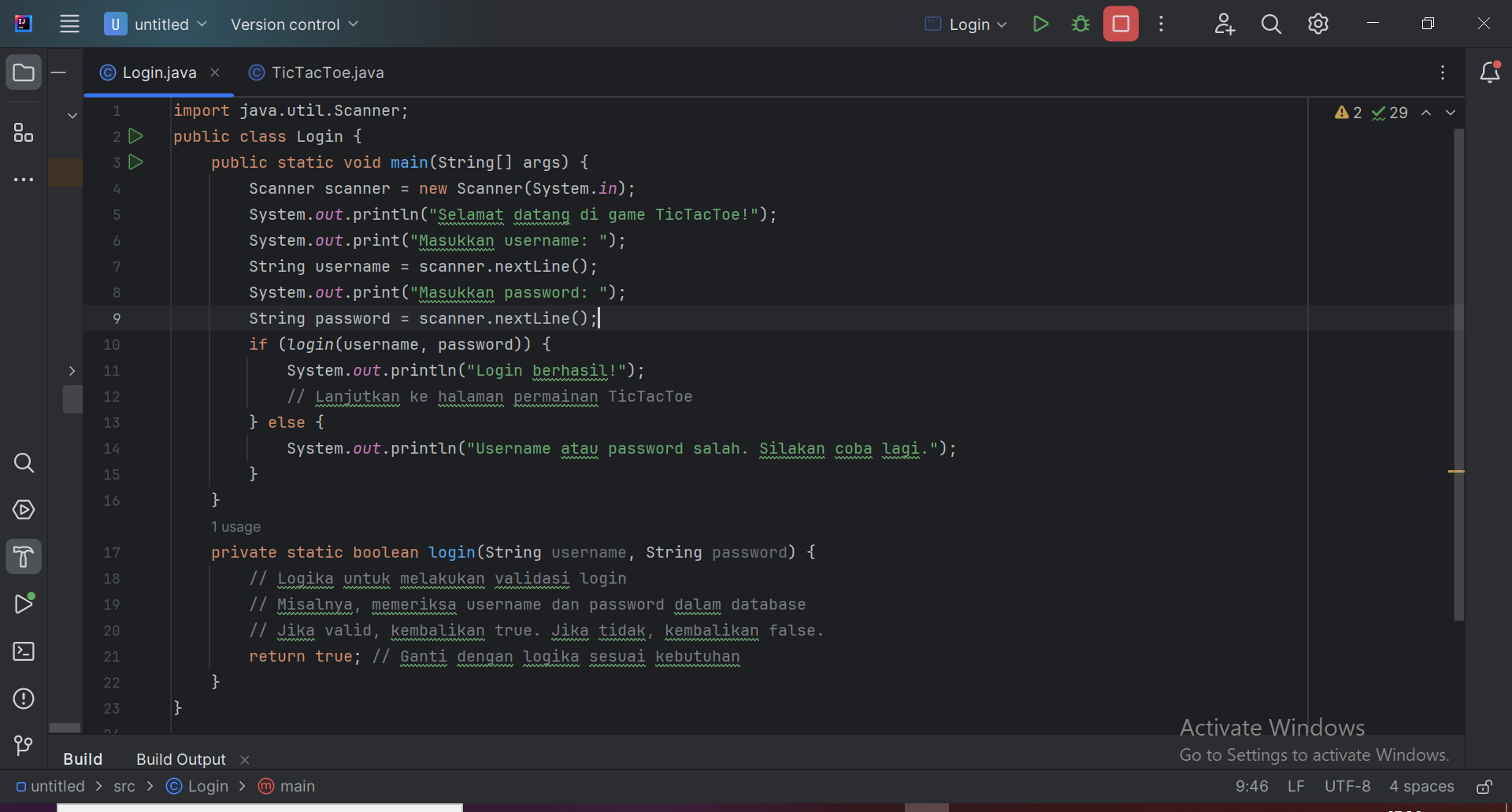
Gambar 1 halaman utama

Penjelasan:

import java.util.Scanner;  
import java.util.Random;  
  
public class TicTacToe {  
 static char[][] *board* = new char[3][3];  
 static char *currentPlayer*;  
 static boolean *isGameOver* = false;

Pada kode ini saya menggunakan kelas Scanner dan Random untuk membaca input dari pengguna dan menghasilkan angka acak. Kelas TicTacToe memiliki beberapa variabel statistik seperti board, currentPlayer, dan isGameOver. Variabel board adalah array 2 dimensi yang merepresentasikan papan permainan TicTacToe. Variabel currentPlayer menyimpan karakter pemain saat ini yang sedang bermain, yaitu 'X' atau 'O'. Variabel isGameOver digunakan untuk menandakan apakah permainan sudah selesai atau belum. Kode ini juga memiliki beberapa metode yang digunakan untuk mengatur permainan TicTacToe.

1. Halaman Login



G1A023013

Gambar 2 halaman login

Penjelasan :

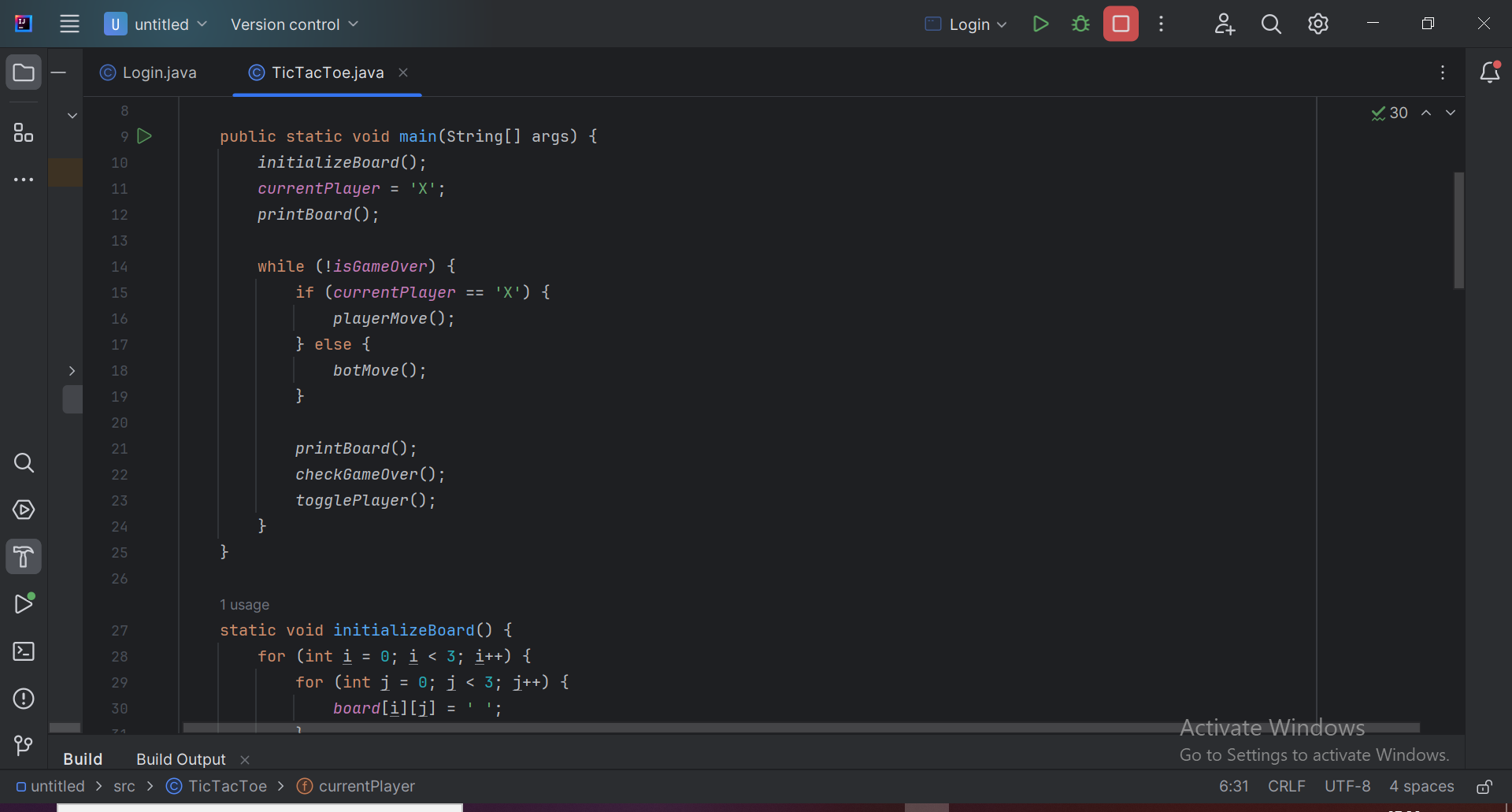
import java.util.Scanner;  
public class Login {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Selamat datang di game TicTacToe!");  
 System.*out*.print("Masukkan username: ");  
 String username = scanner.nextLine();  
 System.*out*.print("Masukkan password: ");  
 String password = scanner.nextLine();  
 if (*login*(username, password)) {  
 System.*out*.println("Login berhasil!");  
 // Lanjutkan ke halaman permainan TicTacToe  
 } else {  
 System.*out*.println("Username atau password salah. Silakan coba lagi.");  
 }  
 }  
 private static boolean login(String username, String password) {  
 // Logika untuk melakukan validasi login  
 // Misalnya, memeriksa username dan password dalam database  
 // Jika valid, kembalikan true. Jika tidak, kembalikan false.  
 return true; // Ganti dengan logika sesuai kebutuhan  
 }  
}

pada halaman login ini terdapat code “public static” pada bagian **Scanner scanner = new Scanner(System.*in*);** berfungsi memberikan pertannyaan username dan password pada halaman output nya nanti. Dan Code ini meminta pengguna untuk memasukkan username dan password, kemudian memvalidasi input tersebut menggunakan method login.

Pada awal program, sebuah objek Scanner (import java.util.Scanner;)dibuat untuk membaca input dari pengguna. Kemudian, pesan selamat datang dan permintaan untuk memasukkan username dan password ditampilkan. Input username dan password kemudian disimpan dalam variabel username dan password.

Setelah itu, program memanggil method login dengan parameter username dan password yang dimasukkan. Method login ini bertugas untuk melakukan validasi login, seperti memeriksa username dan password dalam database. Jika validasi berhasil, method login akan mengembalikan nilai true, dan program akan mencetak pesan System.*out*.println("Login berhasil!"); "Login berhasil!" dan melanjutkan ke halaman permainan TicTacToe. Jika validasi gagal, method login akan mengembalikan nilai false, dan program akan mencetak pesan System.*out*.println("Username atau password salah. Silakan coba lagi."); "Username atau password salah. Silakan coba lagi."

1. Halaman main



G1A023013

Gambar 3 halaman main

Penjelasan :

public static void main(String[] args) {  
 *initializeBoard*();  
 *currentPlayer* = 'X';  
 *printBoard*();  
  
 while (!*isGameOver*) {  
 if (*currentPlayer* == 'X') {  
 *playerMove*();  
 } else {  
 *botMove*();  
 }  
  
 *printBoard*();  
 *checkGameOver*();  
 *togglePlayer*();  
 }  
}

pada code ini berfungsi untuk menjalankan game , yang dimana di atas dinyatakan kita akan memainkan X sebagai kita , maka 0 akan di mainkan computer/Bot. Pada code ini lah akan ada aturan giliran pemain dan memeriksa apakah pemainnan sudah selesai atau belum.

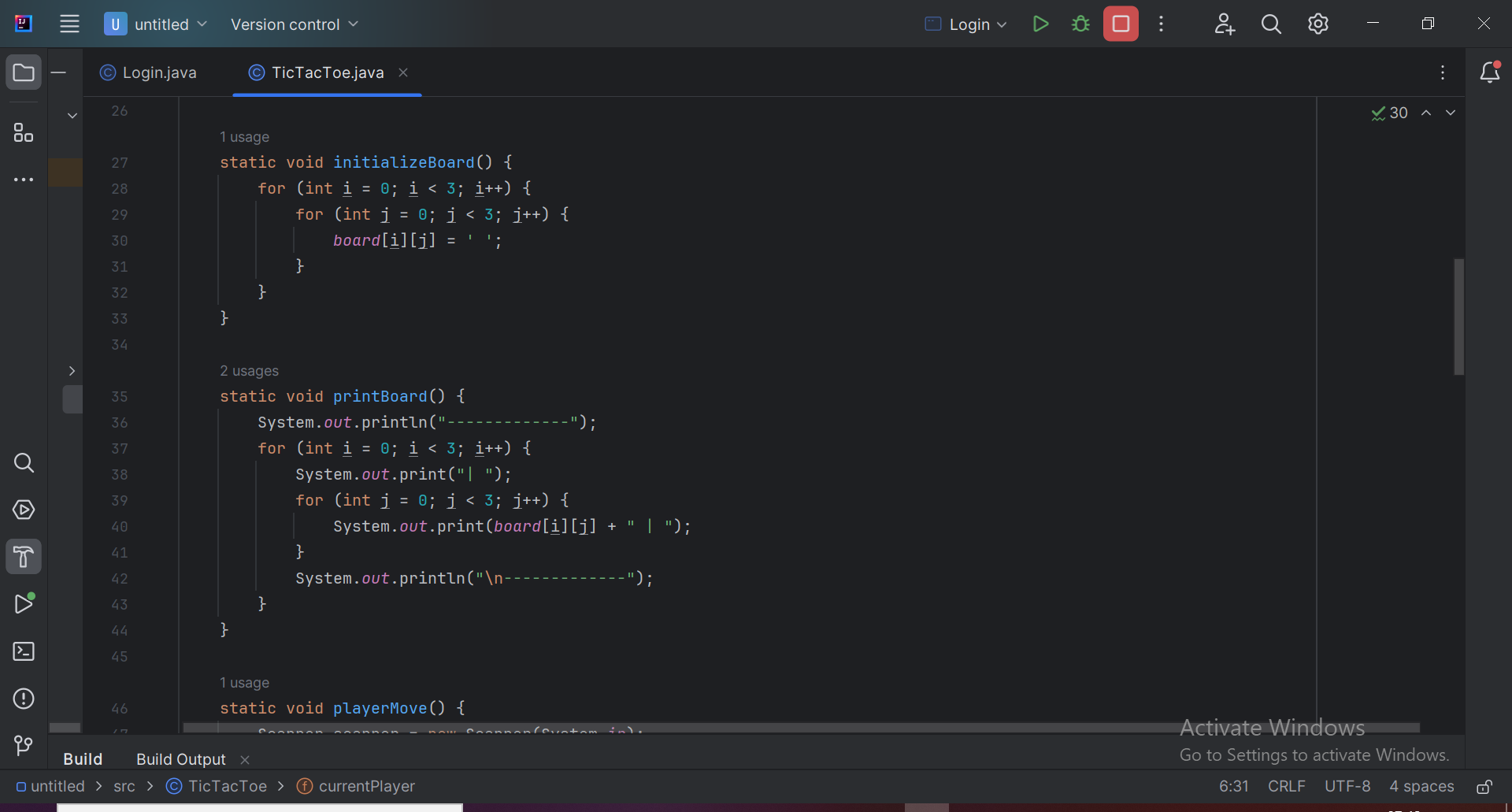
Pertama, metode inisialisasiBoard()*initializeBoard*(); dipanggil untuk menginisialisasi papan permainan dengan karakter kosong. Kemudian, variabel currentPlayer diatur menjadi 'X' yang menandakan pergantian pemain pertama *currentPlayer* = 'X'. Setelah itu, metode printBoard() dipanggil untuk mencetak papan permainan ke layar.

Selanjutnya, program akan masuk ke dalam loop while yang akan berjalan selama isGameOver bernilai false. Di dalam loop, program akan memeriksa nilai currentPlayer. Jika currentPlayer adalah 'X' if (*currentPlayer* == 'X') , maka method playerMove() akan dipanggil untuk meminta input dari pemain. Jika currentPlayer bukan 'X', maka method botMove() akan dipanggil untuk menghasilkan langkah bot.

Setelah pemain atau bot melakukan langkah, method printBoard() dipanggil untuk mencetak papan permainan yang telah diperbarui ke layar. Kemudian, method checkGameOver() dipanggil untuk memeriksa apakah permainan sudah selesai atau belum. Jika permainan sudah selesai, isGameOver akan diubah menjadi true dan loop while akan berhenti. Jika permainan belum selesai, giliran pemain akan diubah menggunakan method togglePlayer().

Secara keseluruhan, code ini mengatur jalannya permainan TicTacToe dengan mengatur giliran pemain, memeriksa apakah permainan sudah selesai, dan mencetak papan permainan ke layar setiap setelah pemain atau bot melakukan langkah.

1. Halaman inisialisasi papan permainan



G1A023013

Gambar 4 meminilisasi papan permainan

Penjelasan :

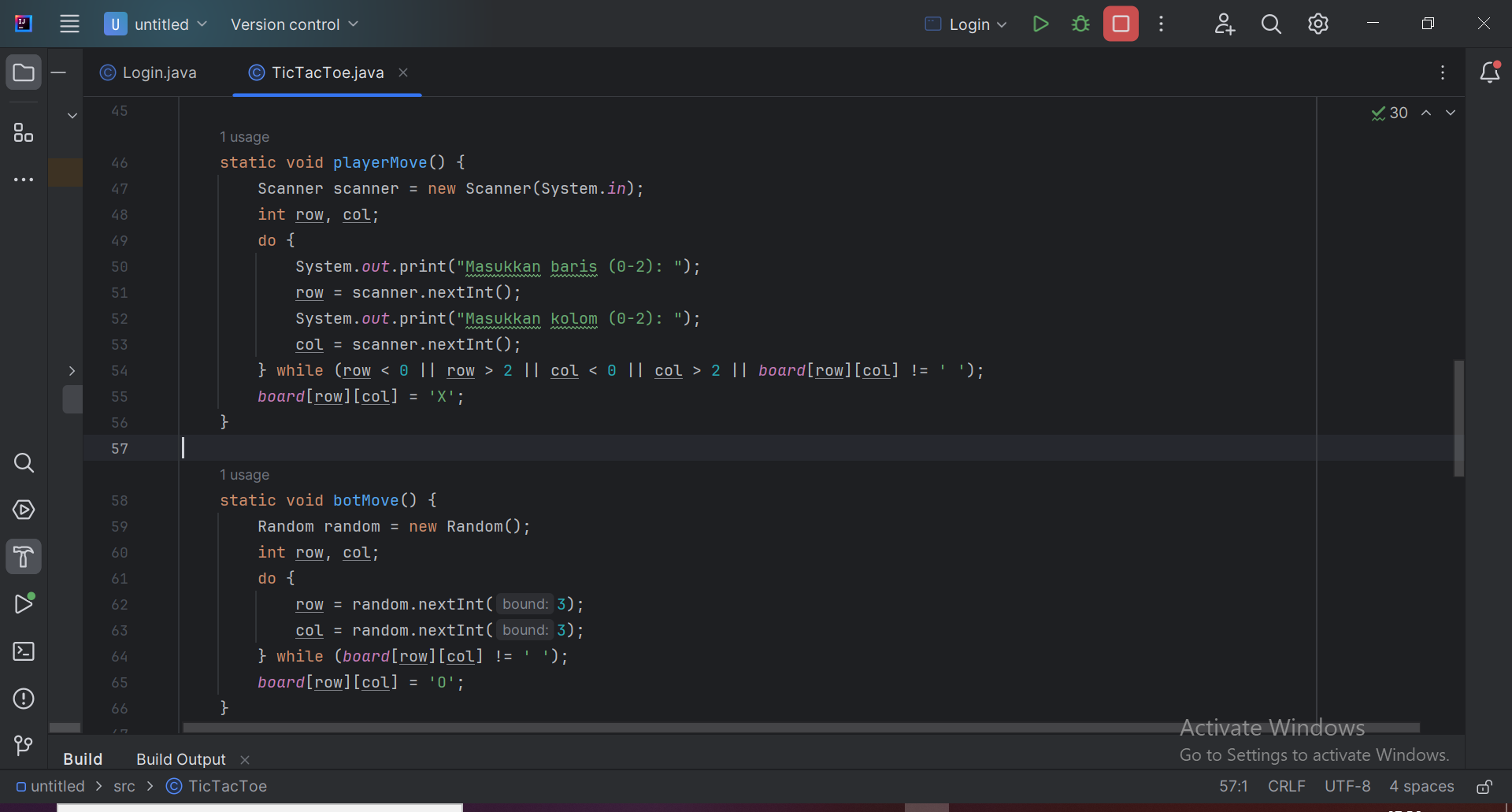
static void initializeBoard() {  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 *board*[i][j] = ' ';  
 }  
 }  
}  
  
static void printBoard() {  
 System.*out*.println("-------------");  
 for (int i = 0; i < 3; i++) {  
 System.*out*.print("| ");  
 for (int j = 0; j < 3; j++) {  
 System.*out*.print(*board*[i][j] + " | ");  
 }  
 System.*out*.println("\n-------------");  
 }  
}

Pada code ini terdiri dari dua method, yaitu `initializeBoard()` dan `printBoard()`,.Method `initializeBoard()` digunakan untuk menginisialisasi papan permainan dengan karakter kosong (' '). Pada code ini, terdapat nested loop `for` yang berjalan sebanyak 3 kali untuk mengiterasi baris dan kolom pada array `board`. Setiap elemen pada array `board` diisi dengan karakter kosong.

Method `printBoard()` digunakan untuk mencetak papan permainan ke layar. Pada code ini, terdapat nested loop `for` yang berjalan sebanyak 3 kali untuk mengiterasi baris dan kolom pada array `board`. Setiap elemen pada array `board` dicetak dengan format "| [elemen] |". Setelah mencetak setiap baris, ditambahkan garis horizontal sebagai pemisah antar baris.

Secara keseluruhan, code ini digunakan untuk menginisialisasi papan permainan dengan karakter kosong dan mencetak papan permainan ke layar dalam format yang terstruktur.

1. playerMove



G1A023013

Gambar 5 static void playerMove

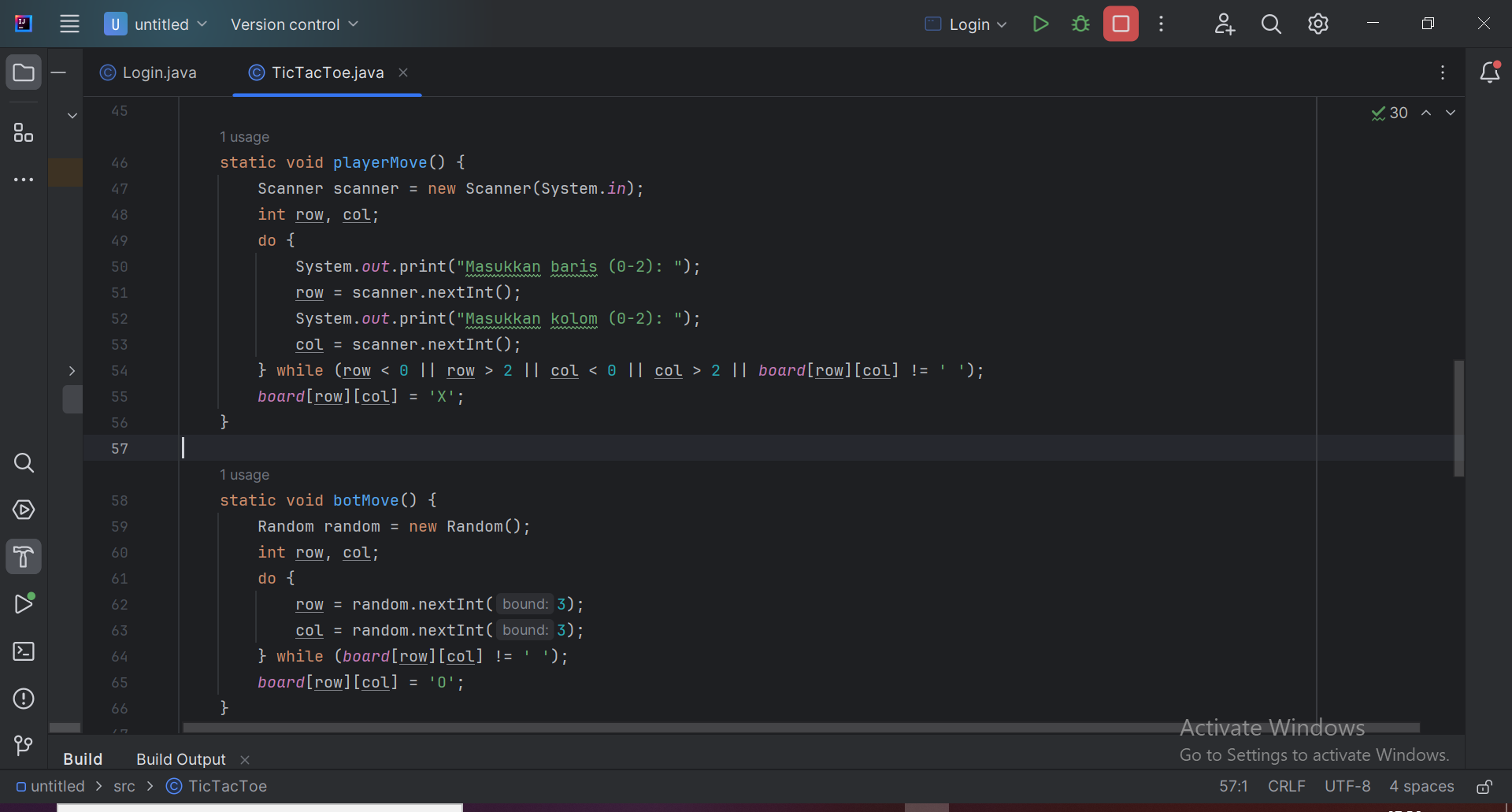
Penjelasan:

static void playerMove() {  
 Scanner scanner = new Scanner(System.*in*);  
 int row, col;  
 do {  
 System.*out*.print("Masukkan baris (0-2): ");  
 row = scanner.nextInt();  
 System.*out*.print("Masukkan kolom (0-2): ");  
 col = scanner.nextInt();  
 } while (row < 0 || row > 2 || col < 0 || col > 2 || *board*[row][col] != ' ');  
 *board*[row][col] = 'X';  
}

Kode ini adalah sebuah metode yang disebut "playerMove". Metode ini digunakan untuk mengambil input dari pemain dalam permainan tertentu.

Pertama, metode ini membuat objek Scanner Scanner scanner = new Scanner(System.*in*); untuk membaca input dari pengguna melalui konsol. Kemudian, menggunakan pernyataan do- while, metode ini akan terus meminta pemain untuk memasukkan baris dan kolom (dalam rentang 0-2) sampai input yang valid diberikan. Setelah itu, metode ini memeriksa apakah posisi yang dimasukkan oleh pemain sudah terisi atau tidak di papan permainan (board). Jika posisi tersebut sudah terisi, maka pemain akan diminta untuk memasukkan posisi yang valid lagi.

1. Halaman botMove



G1A023013

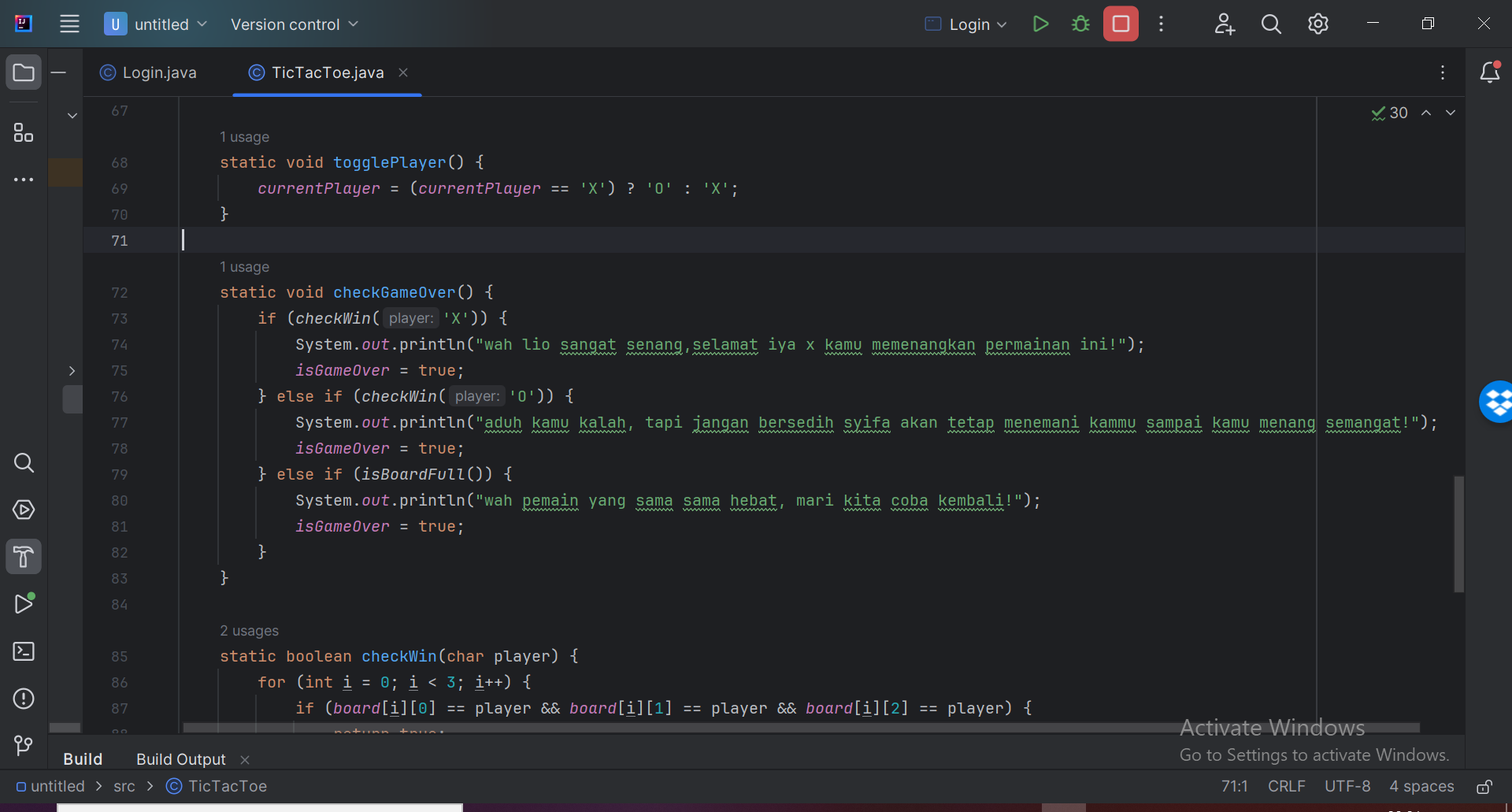
Gambar 6 static void botMove

Penjelasan:

static void botMove() {  
 Random random = new Random();  
 int row, col;  
 do {  
 row = random.nextInt(3);  
 col = random.nextInt(3);  
 } while (*board*[row][col] != ' ');  
 *board*[row][col] = 'O';  
}

Kode ini adalah sebuah metode yang disebut "botMove". Metode ini digunakan untuk menggerakkan bot dalam permainan tertentu. Pertama, metode ini membuat objek Random untuk menghasilkan angka acak. Kemudian, dengan menggunakan pernyataan do- while, metode ini akan terus menghasilkan angka acak untuk baris dan kolom (dalam rentang 0-2) sampai posisi yang valid ditemukan. Setelah itu, metode ini memeriksa apakah posisi yang dihasilkan oleh angka acak sudah terisi atau tidak di papan permainan (board). Jika posisi tersebut sudah terisi, maka metode akan menghasilkan angka acak baru untuk mencari posisi yang valid. Jika posisi yang dihasilkan oleh angka acak valid dan belum terisi, maka metode ini akan menandai posisi tersebut dengan karakter 'O' di papan permainan (board).

1. Toggleplayer



G1A023013

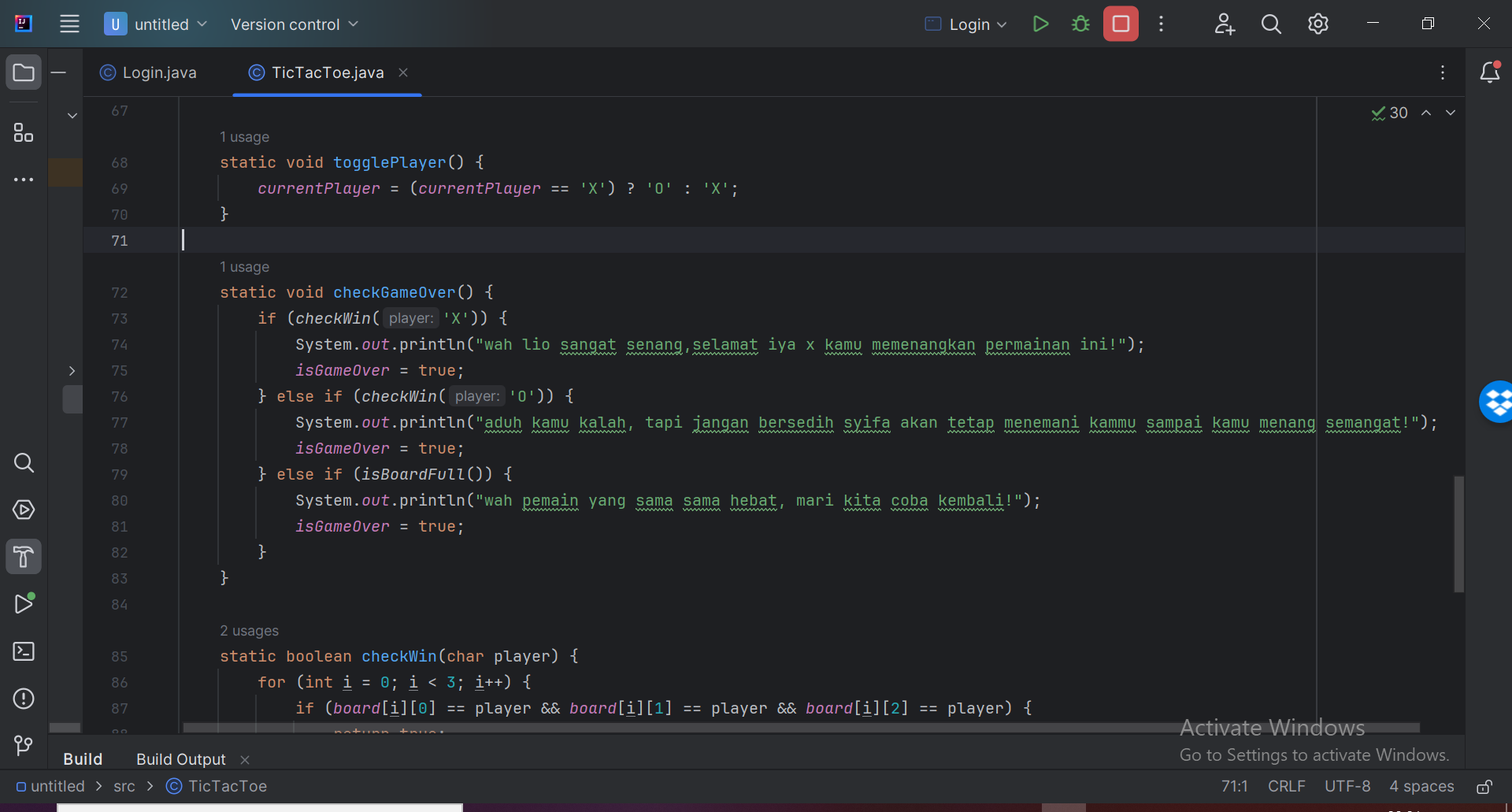
Gambar 7 static void toggleplayer

Penjelasan:

static void togglePlayer() {  
 *currentPlayer* = (*currentPlayer* == 'X') ? 'O' : 'X';  
}

Code ini adalah sebuah metode yang disebut "togglePlayer". Metode ini digunakan untuk mengganti pemain saat giliran bermain dalam permainan tertentu Pertama, metode ini menggunakan operator ternary untuk memeriksa nilai dari variabel currentPlayer. Jika nilai currentPlayer adalah 'X', maka nilai currentPlayer akan diubah menjadi 'O'. Jika nilai currentPlayer bukan 'X', maka nilai currentPlayer akan diubah menjadi 'X' Dengan menggunakan operator ternary, metode ini secara efisien mengganti pemain saat giliran bermain. Jika currentPlayer adalah 'X', maka setelah metode ini dijalankan, currentPlayer akan menjadi 'O'. Jika currentPlayer bukan 'X', maka setelah metode ini dijalankan, currentPlayer akan menjadi 'X'.

1. Halaman checkGame0ver



G1A023013

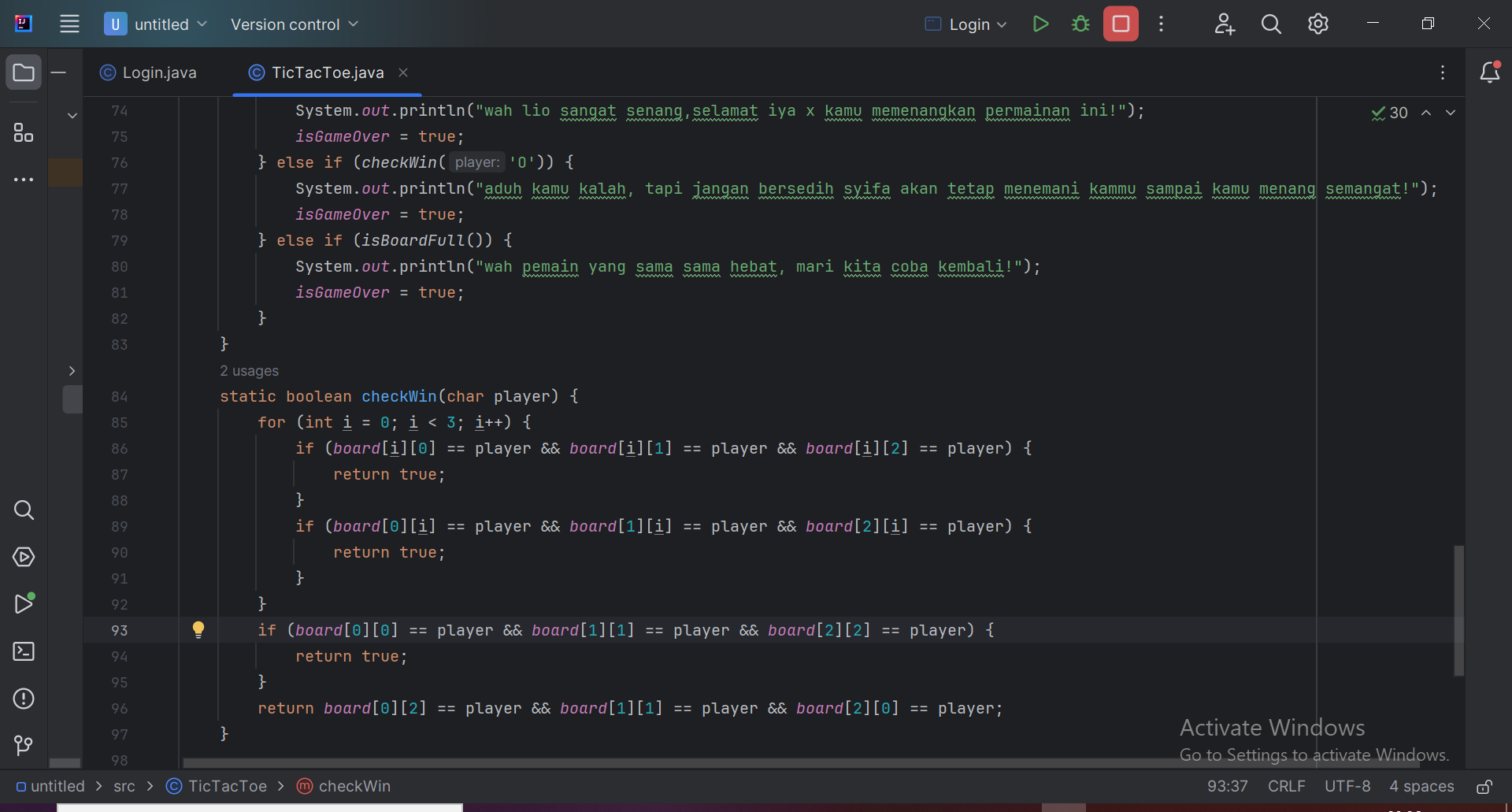
Gambar 8 static void chekGameOver

Penjelasan:

static void checkGameOver() {  
 if (*checkWin*('X')) {  
 System.*out*.println("wah lio sangat senang,selamat iya x kamu memenangkan permainan ini!");  
 *isGameOver* = true;  
 } else if (*checkWin*('O')) {  
 System.*out*.println("aduh kamu kalah, tapi jangan bersedih syifa akan tetap menemani kammu sampai kamu menang semangat!");  
 *isGameOver* = true;  
 } else if (*isBoardFull*()) {  
 System.*out*.println("wah pemain yang sama sama hebat, mari kita coba kembali!");  
 *isGameOver* = true;  
 }  
}

Kode ini adalah sebuah metode yang disebut "checkGameOver". Metode ini digunakan untuk memeriksa apakah permainan telah berakhir. Pertama, metode ini memeriksa apakah pemain dengan karakter 'X' telah memenangkan permainan dengan memanggil metode "checkWin" dan memberikan karakter 'X' sebagai argumen if (*checkWin*('X')) {  
 System.*out*.println("wah lio sangat senang,selamat iya x kamu memenangkan permainan ini!");  
. Jika pemain 'X' memenangkan permainan, maka akan dicetak pesan "wah lio sangat senang,selamat iya x kamu memenangkan permainan ini!" dan variabel isGameOver akan diatur menjadi true. Selanjutnya, metode ini memeriksa apakah pemain dengan karakter 'O' telah memenangkan permainan dengan memanggil metode "checkWin" dan memberikan karakter 'O' sebagai argumen. Jika pemain 'O' memenangkan permainan, maka akan dicetak pesan "aduh kamu kalah, tapi jangan bersedih syifa akan tetap menemani kammu sampai kamu menang semangat!" dan variabel isGameOver akan diatur menjadi true.Selain itu, metode ini juga memeriksa apakah papan permainan (board) sudah penuh dengan memanggil metode "isBoardFull". Jika papan permainan sudah penuh, maka akan dicetak pesan "wah pemain yang sama-sama hebat, mari kita coba kembali! Dengan demikian, kode ini bertujuan untuk memeriksa kondisi akhir permainan, seperti pemain memenangkan permainan, pemain kalah, atau papan permainan sudah penuh.

1. Halaman boolean



G1A023013

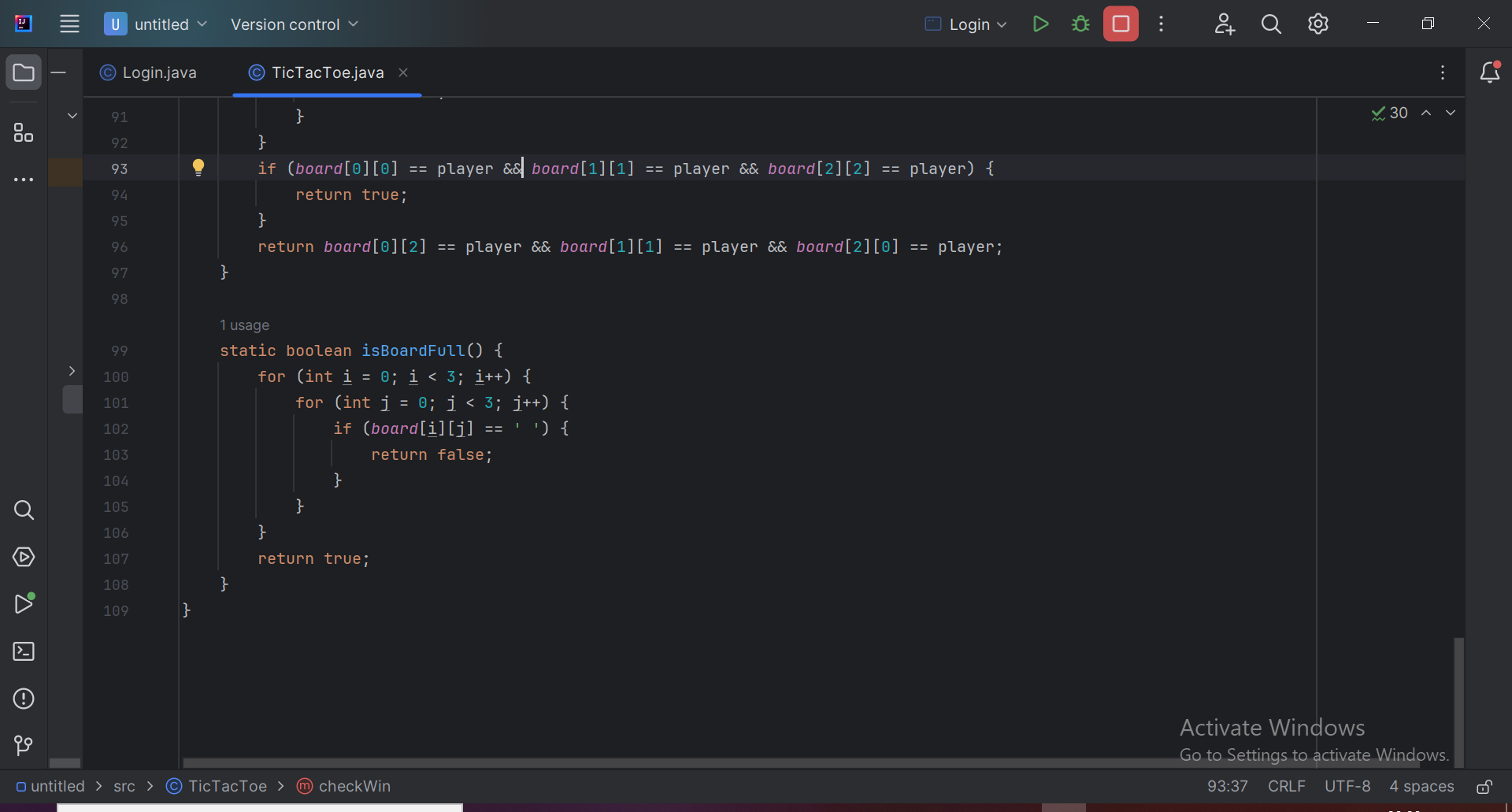
Gambar 9 static boolean

Penjelasan:

static boolean checkWin(char player) {  
 for (int i = 0; i < 3; i++) {  
 if (*board*[i][0] == player && *board*[i][1] == player && *board*[i][2] == player) {  
 return true;  
 }  
 if (*board*[0][i] == player && *board*[1][i] == player && *board*[2][i] == player) {  
 return true;  
 }  
 }  
 if (*board*[0][0] == player && *board*[1][1] == player && *board*[2][2] == player) {  
 return true;  
 }  
 return *board*[0][2] == player && *board*[1][1] == player && *board*[2][0] == player;  
}

Kode ini adalah sebuah metode yang disebut "checkWin". Metode ini digunakan untuk memeriksa apakah pemain (player) telah memenangkan permainan dalam suatu kondisi tertentu. Metode ini menggunakan perulangan untuk memeriksa setiap baris dan kolom pada papan permainan (board). Pada setiap iterasi, metode ini memeriksa apakah semua elemen pada baris atau kolom tersebut sama dengan karakter pemain yang diberikan. Jika ya, maka metode ini akan mengembalikan nilai true, menandakan bahwa pemain telah memenangkan permainan. Selain itu, metode ini juga memeriksa diagonal utama (board[0][0], board[1][1], board[2][2]) dan diagonal kedua (board[0][2], board[1] [1], papan[2][0]). Jika semua elemen pada salah satu diagonal tersebut sama dengan karakter pemain yang diberikan, maka metode ini juga akan mengembalikan nilai true. Jika tidak ada kondisi yang memenuhi untuk memenangkan permainan, metode ini akan mengembalikan nilai false. Dengan demikian, kode ini bertujuan untuk memeriksa apakah pemain telah memenangkan permainan berdasarkan kondisi pada papan permainan (board).

1. Halaman akhir



G1A023013

Gambar 10 boolean isBoardFull

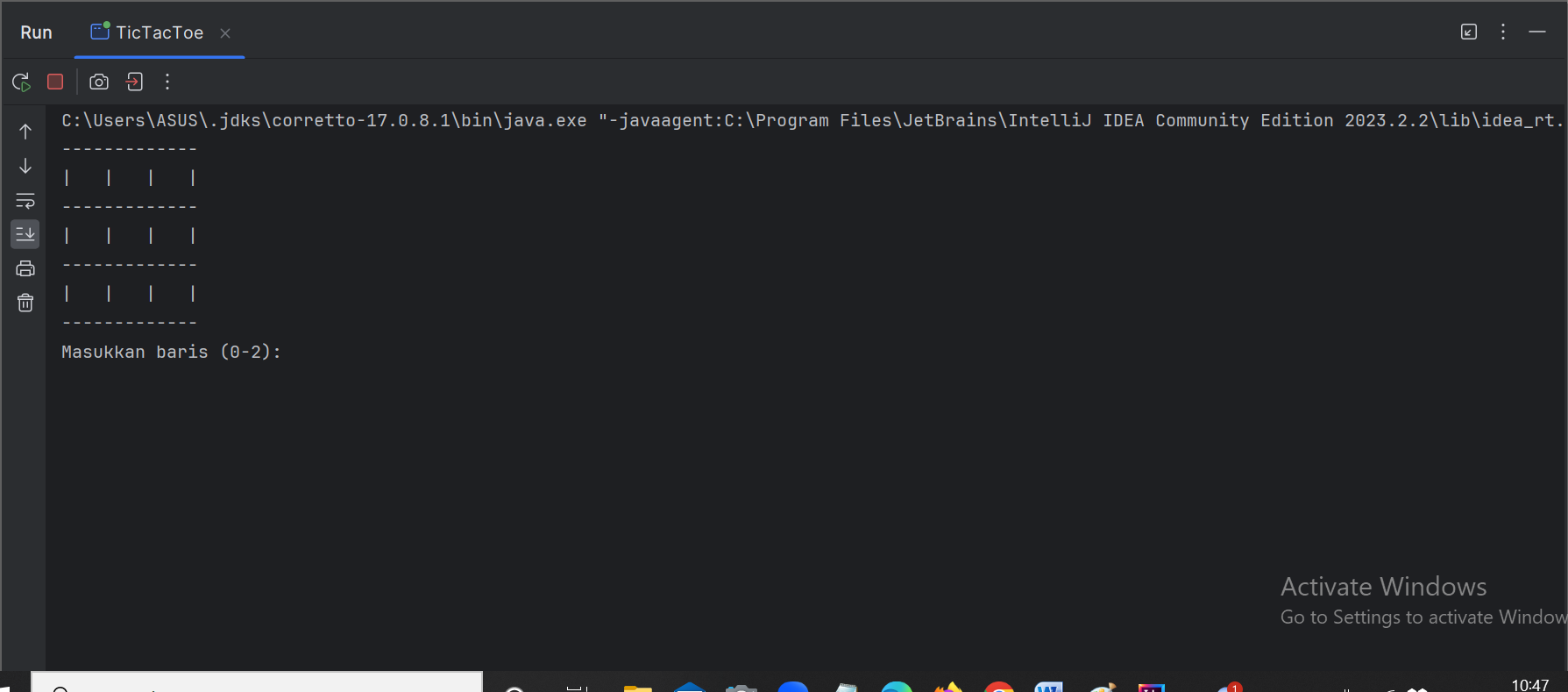
Penjelasn:

static boolean isBoardFull() {  
 for (int i = 0; i < 3; i++) {  
 for (int j = 0; j < 3; j++) {  
 if (*board*[i][j] == ' ') {  
 return false;  
 }  
 }  
 }  
 return true;  
 }  
}

pada bagian trakhir ini adalah bagian yang menyatakakan static boolean isBoardFull()

**Berikut adalah output atau hasil dari program yang telah dibuat:**

1. Output Halaman Utama



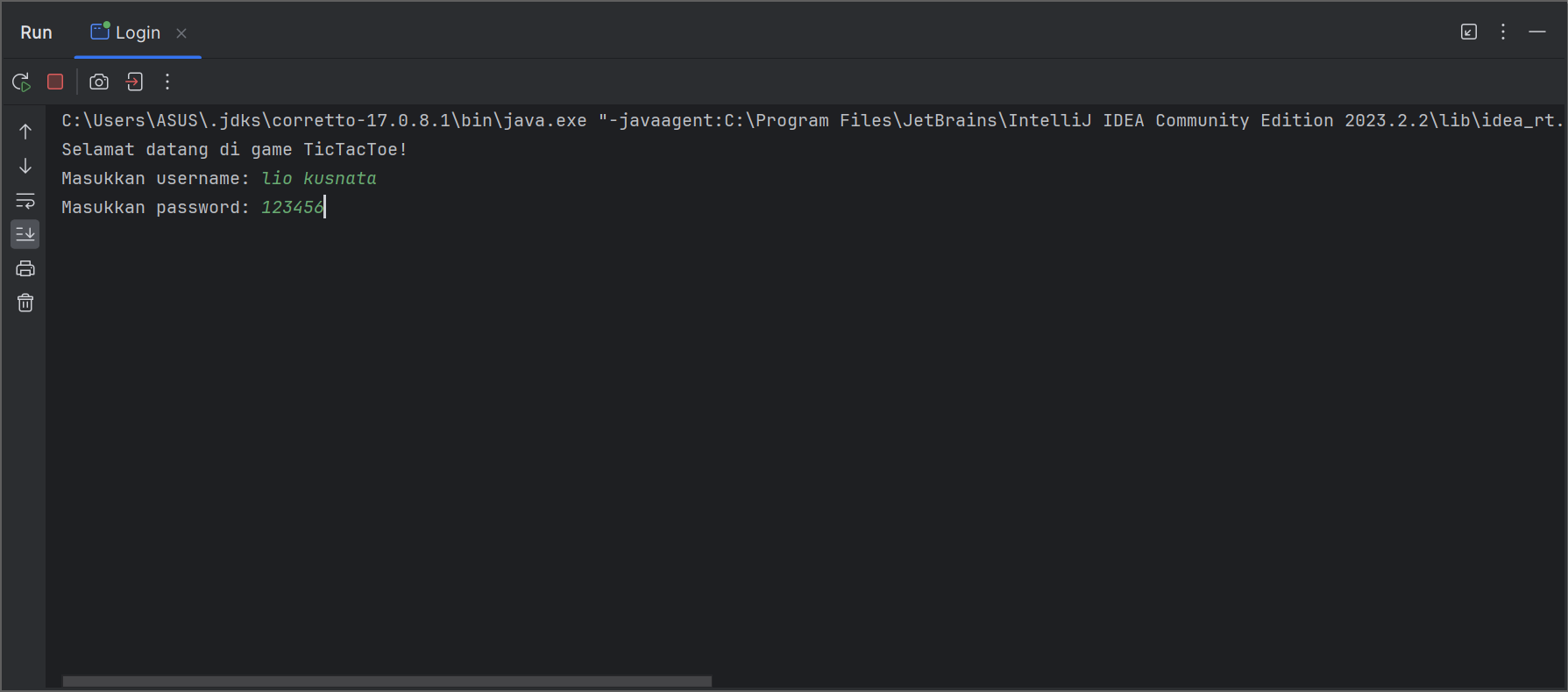
G1A023013

Gambar 1.1 output Halaman Utama

Penjelasan :

Pada output halaman pertama akan menghasilkan kotak untuk melakukan permainan tictactoe namun code ini tetap harus berjalan bersama dengan class.

1. Output Halaman Login



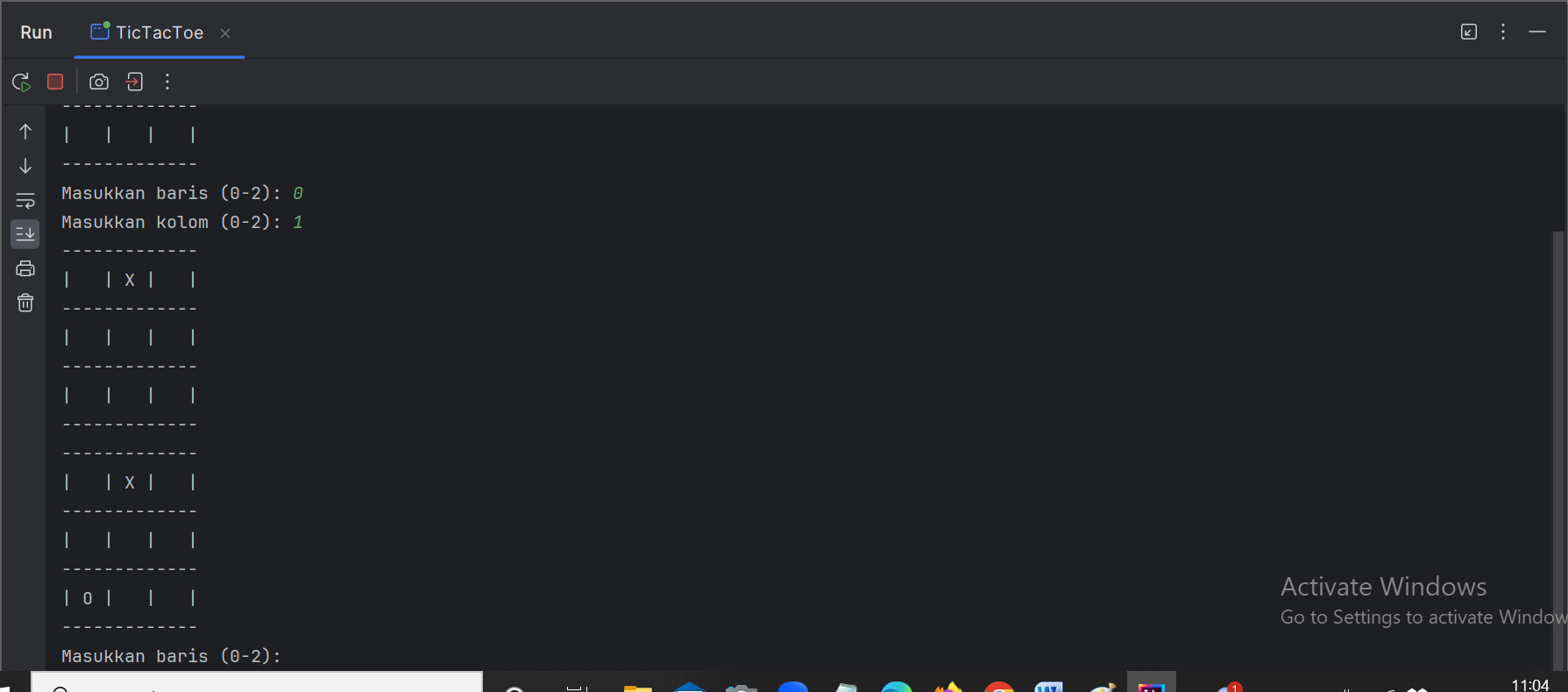
G1A023013

Gambar 1.2 Halaman Login

Penjelasan :

Pada gambar ini menampakan hasil dari code halaman login yang telah di jelaskan di atas, dan kita dapat mengisi username dan password.

1. Memasukan 0-2

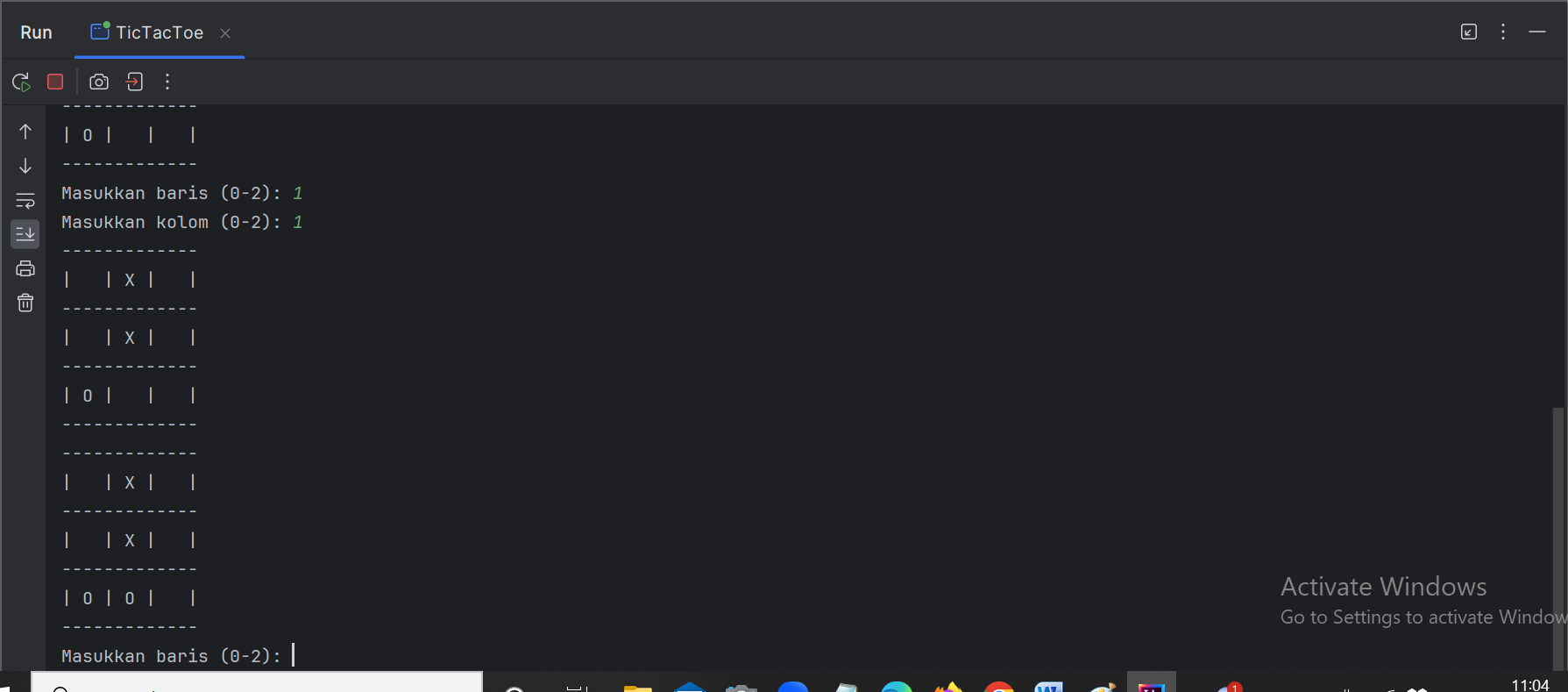


G1A023013

Gambar 1.3 mamasukan baris dan kolom

Penjelasan : pada bagian ini adalah hasil output dari playerMove

1. Bot bermain

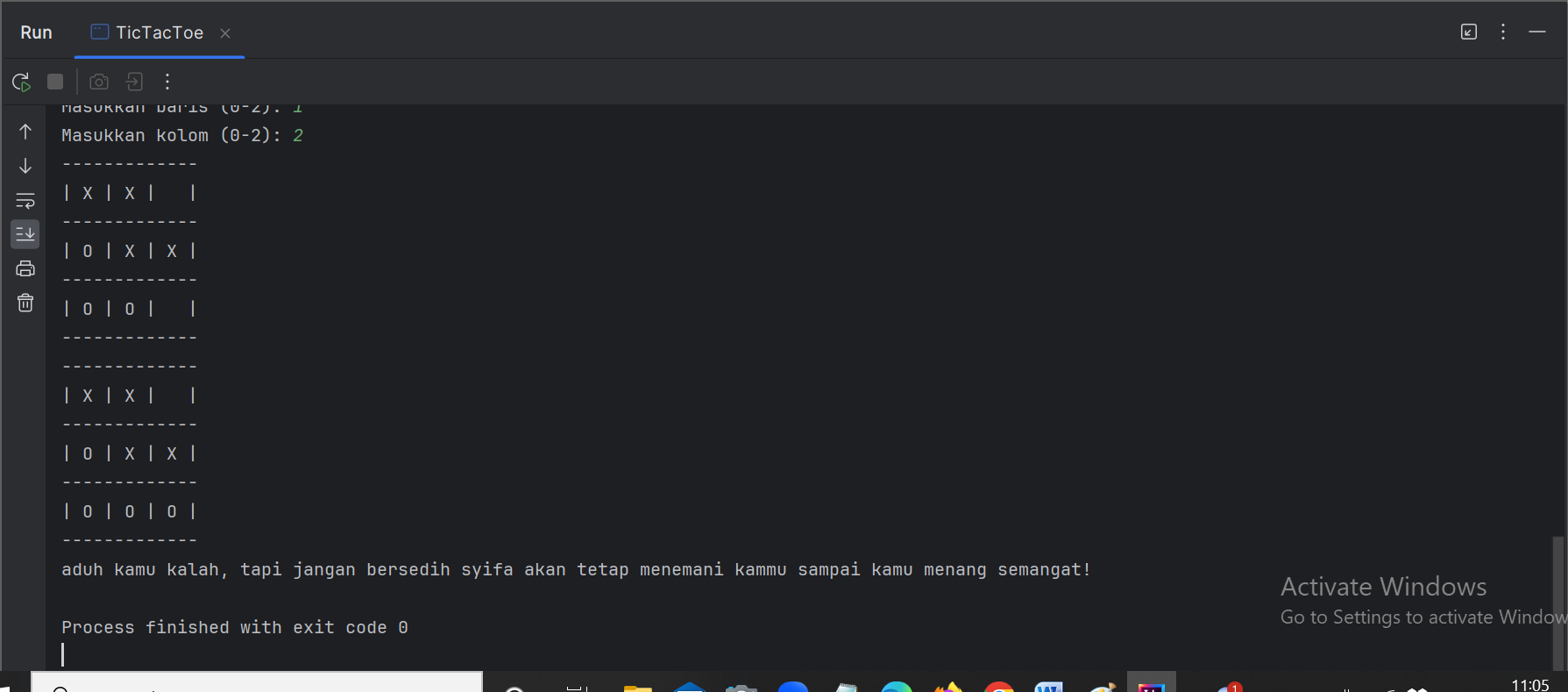


G1A023013

Gambar 1.4 bot bermain

Penjelasan : output ini hasil dari botMove

1. Ucapan akhir

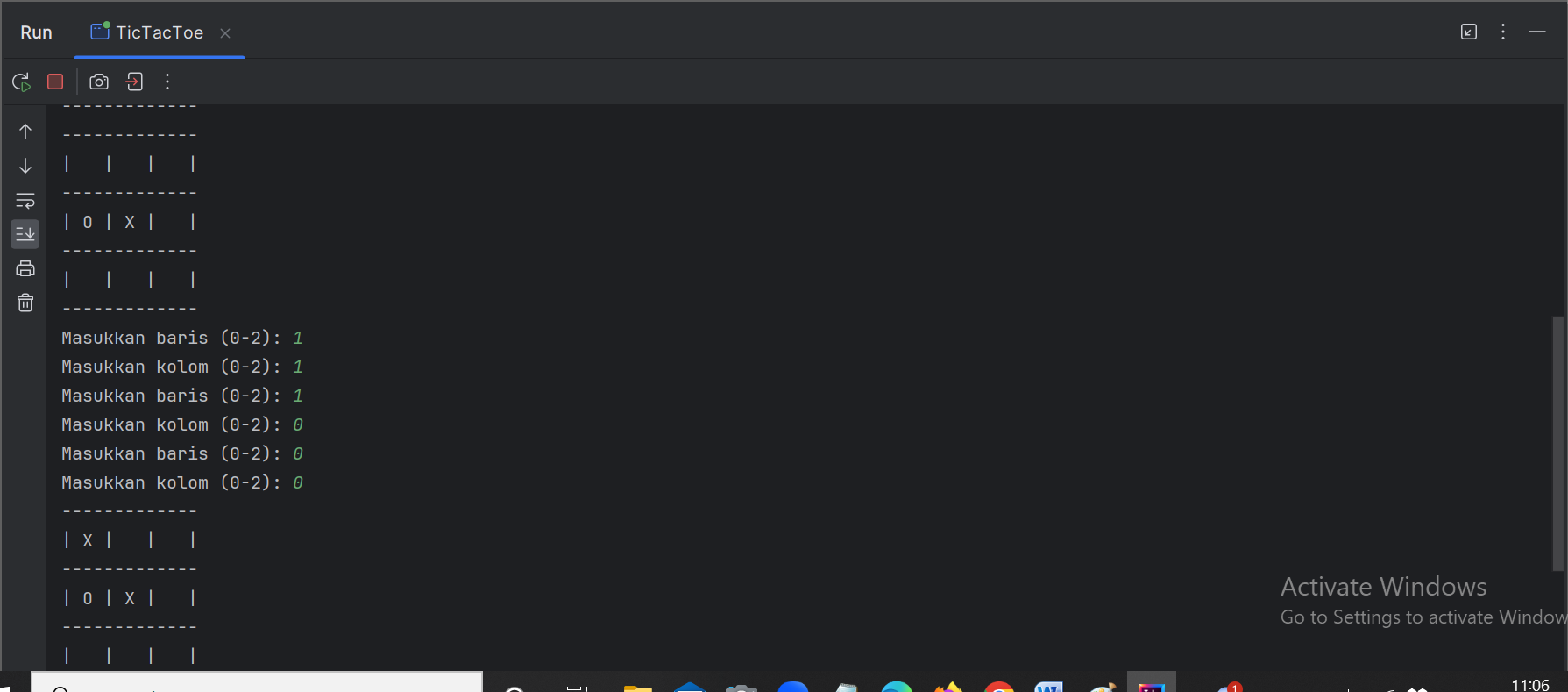


G1A023013

Gambar 1.5 kata-kata di akhir

Penjelasan : output ini hasil dari checkGameOver

1. Mengulang pengisian baris dan kolom



G1A023013

Gambar 1.6 pengisian 0-2 ulang

Penjelasan : output ini hasil dari playerMove

# **Kesimpulan dan Saran**

1. **Kesimpulan**

**Java** merupakan bahasa pemrograman yang biasa digunakan untuk mengembangkan bagian back-end pada software, aplikasi Android, dan juga website. Java dapat dijalankan di berbagai platform tanpa perlu disusun ulang yang dicocokan dengan platformnya. Contohnya, Java dapat berjalan di Android, Windows, Linux, dan platform lainnya. Hal ini dapat terjadi karena Java mempunyai sistem syntax atau kode pemrograman level tinggi. Jadi ketika program dijalankan, syntax akan di-compile dengan menggunakan bantuan dari **Java Virtual Machine**(JVM) yang menjadikan kode numeric (bytescode) platform.

Sehingga aplikasi Java ini dapat dijalankan di berbagai perangkat.  java dikenal memiliki moto “Write Once, Run Anywhere”. Hal ini menunjukkan bahwa setiap kode program yang ditulis menggunakan Java bisa langsung dijalankan pada platform berbeda-beda tanpa perlu menyusun ulang. Yang menarik adalah bahwa Java bisa digunakan untuk membuat program yang dapat berjalan di atas HP, PDA, dan peralatan yang dilengkapi dengan Java Virtual Machine (JVM). Java juga lebih mudah dan Java sangat membantu terutama jika anda mengembangkan sofware berskala besar, multiplafform, dan mulitiinteerface. Ini dikarenakan Java memiliki standar yang komperhensif, integral dan jauh lebih terorganisasi dibanding bahasa pemrograman lain.

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi tidak hanya desktop dan web namun juga bisa membuat aplikasi mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (general purpose). Bahasa Pemrograman Java berorientasi object (OOP-Object Oriented Programming), dan dapat dijalankan pada berbagai platform sistem operasi. Pada OOP, program komputer sebagai kelompok object yang saling berinteraksi.

1. Saran

Dikarenakan cukup rumitnya bahasa pemrograman Java dan semakin meningkatnya perkembangan teknologi, semakin banyak pula peningkatan pesaing yang berkompetisi di bidang pemrograman Java, maka disarankan kepada para pengguna (khususnya pemula), untuk sejak dini mulai mempelajari dan bisa mendalami bahasa pemrograman java tersebut.

Program tersebut juga memiliki penyimpanan yang cukup besar sehingga memerlukan memori yang cukup dalam menyimpan aplikasi java tersebut. Dan bisa memilah aplikasi yang terkandung dalam Java yang dapat memberikan manfaat yang baik.

# **Daftar Pustaka**

Jean Scenny (2013, April 04),

<https://www.scribd.com/doc/134141676/Makalah-Java-Tugas-PBO-I>

Putri\_Mufidah (2015, Januari)

https://www.scribd.com/doc/251811807/Laporan-Pratikum-Bahasa-Pemrograman-

Java-Menggunakan-Netbeans

Rachmat Selamet (2016)

[PEMBANGUNAN SISTEM ONLINE TEST (likmi.ac.id)](https://jurnal.likmi.ac.id/Jurnal/11_2016/112016_03_Rachmat.pdf)

Rickyanto (2003),

https://repository.dinamika.ac.id/id/eprint/1835/4/BAB\_II.pdf

Rizki Fauzan (2017),

https://www.academia.edu/36795910/makalah\_pengenalan\_javaserta\_netbeans\_

setting\_path\_pada\_java?source=swp\_share

# Utia Fauziah Yahya (2013, Oktober 31),

# <https://www.slideshare.net/utiafauziahyahya/makalah-java>