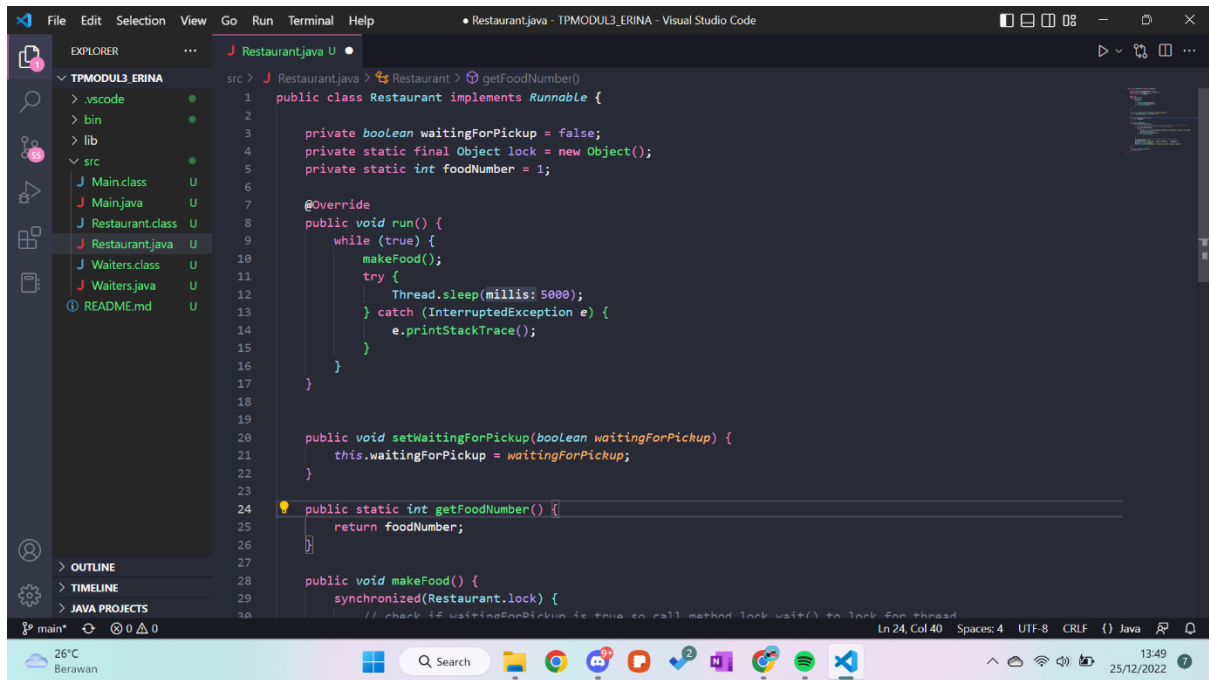


Nama : Erina Zahira Nurhasan

NIM : 1202210357

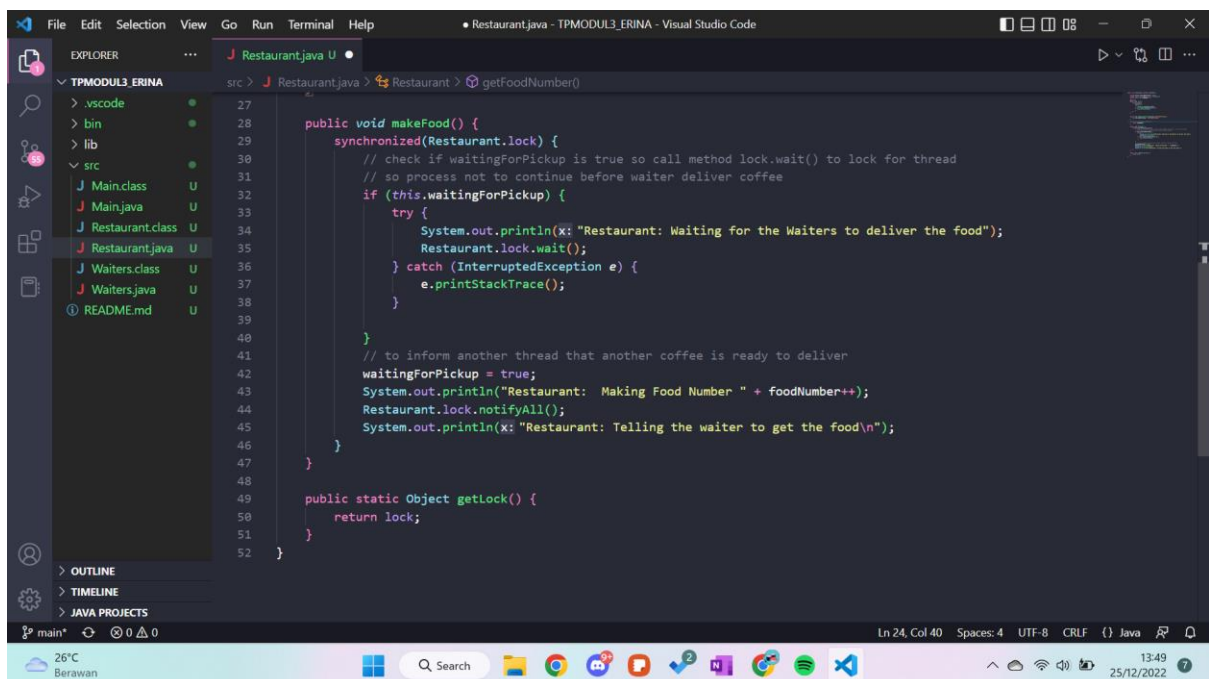
Kelas : SI-45-03

## 1. Restaurant.java



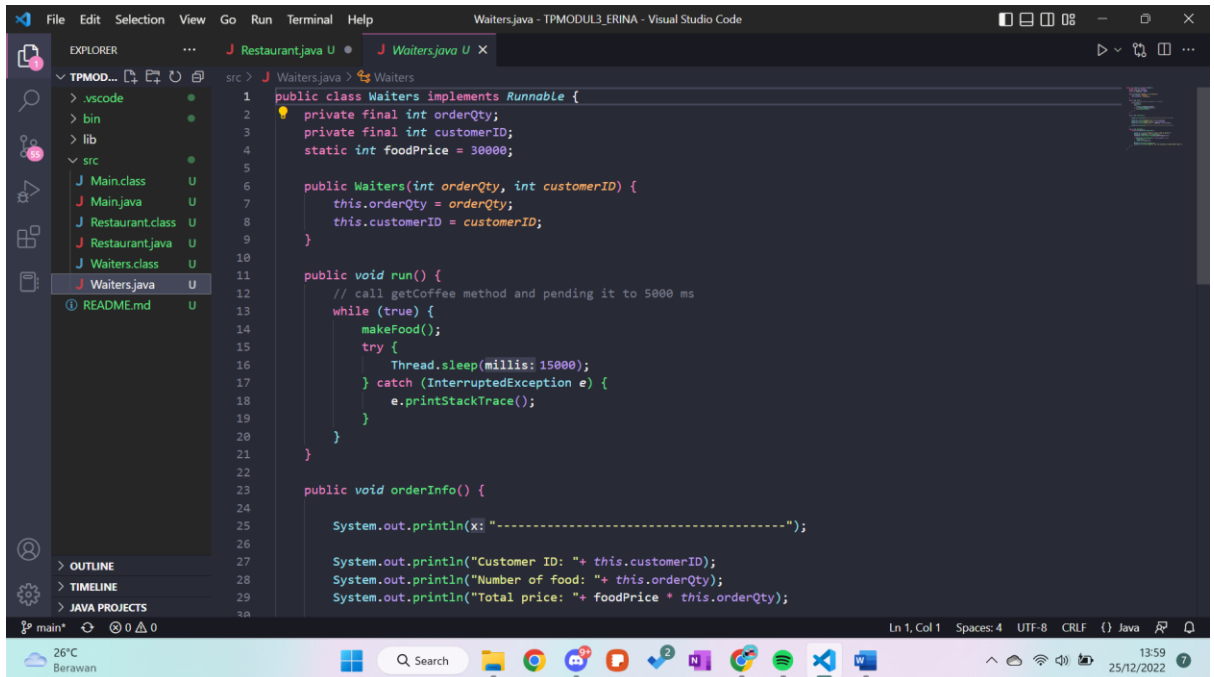
```
File Edit Selection View Go Run Terminal Help • Restaurant.java - TPMODUL3_ERINA - Visual Studio Code
EXPLORER
TPMODUL3_ERINA
  .vscode
  bin
  lib
  src
    J Main.class U
    J Main.java U
    J Restaurant.class U
    J Restaurant.java U
    J Waiters.class U
    J Waiters.java U
    README.md U
  OUTLINE
  TIMELINE
  JAVA PROJECTS
main* 26°C Berawan

src > J Restaurant.java > Restaurant > getFoodNumber()
1 public class Restaurant implements Runnable {
2
3
4 private boolean waitingForPickup = false;
5 private static final Object lock = new Object();
6 private static int foodNumber = 1;
7
8 @Override
9 public void run() {
10     while (true) {
11         makeFood();
12         try {
13             Thread.sleep(5000);
14         } catch (InterruptedException e) {
15             e.printStackTrace();
16         }
17     }
18
19 public void setWaitingForPickup(boolean waitingForPickup) {
20     this.waitingForPickup = waitingForPickup;
21 }
22
23
24 public static int getFoodNumber() {
25     return foodNumber;
26 }
27
28 public void makeFood() {
29     synchronized(Restaurant.lock) {
30         // check if waitingForPickup is true so call method lock wait() to lock for thread
31     }
32 }
```

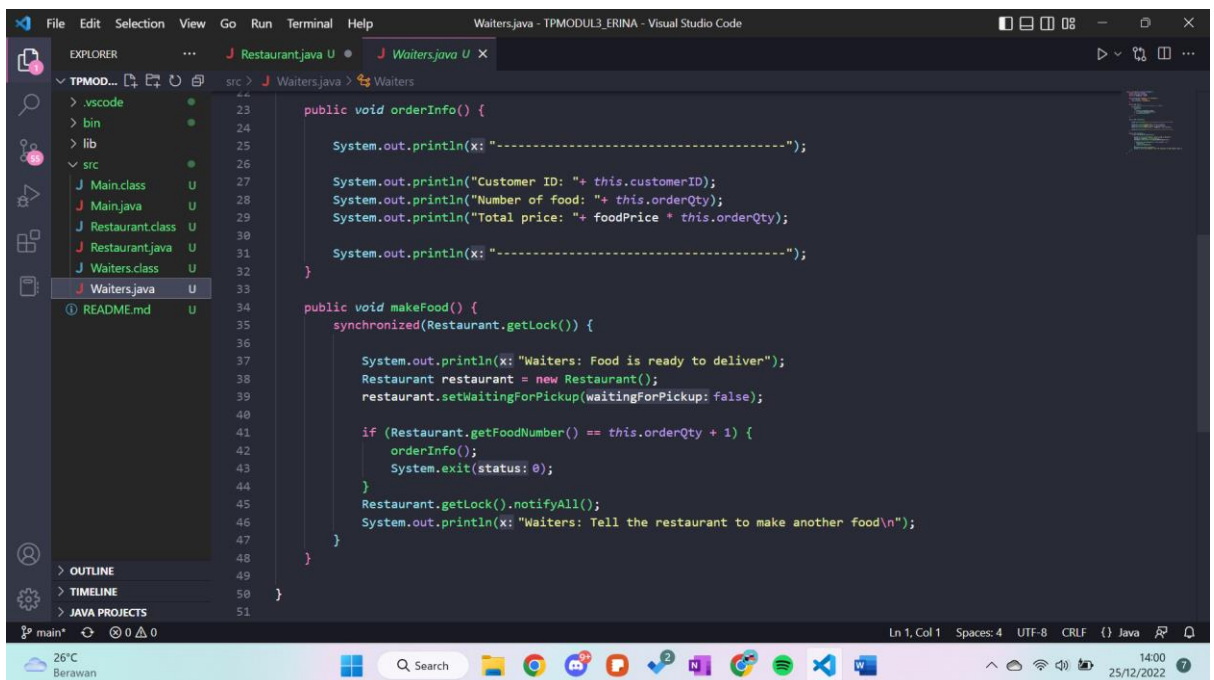


```
27
28 public void makeFood() {
29     synchronized(Restaurant.lock) {
30         // check if waitingForPickup is true so call method lock wait() to lock for thread
31         // so process not to continue before waiter deliver coffee
32         if (this.waitingForPickup) {
33             try {
34                 System.out.println("Restaurant: Waiting for the Waiters to deliver the food");
35                 Restaurant.lock.wait();
36             } catch (InterruptedException e) {
37                 e.printStackTrace();
38             }
39         }
40         // to inform another thread that another coffee is ready to deliver
41         waitingForPickup = true;
42         System.out.println("Restaurant: Making Food Number " + foodNumber++);
43         Restaurant.lock.notifyAll();
44         System.out.println("Restaurant: Telling the waiter to get the food\n");
45     }
46 }
47
48 public static Object getLock() {
49     return lock;
50 }
51
52 }
```

## 2. Waiters.java

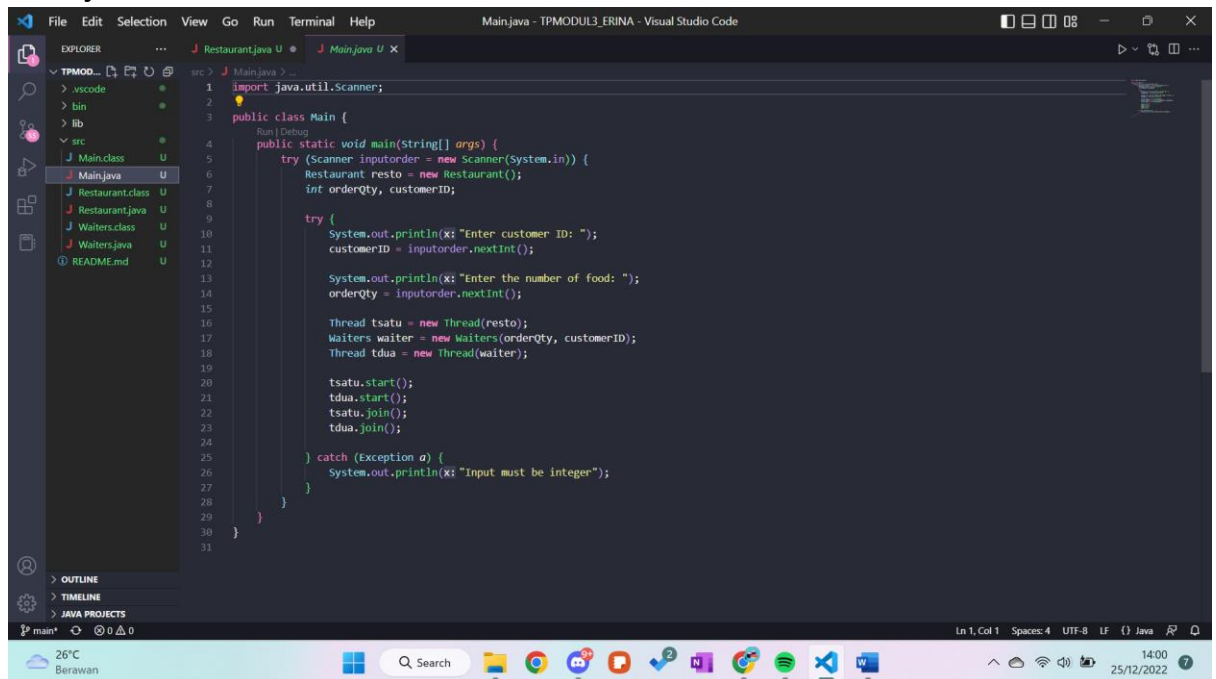


```
1 public class Waiters implements Runnable {
2     private final int orderQty;
3     private final int customerID;
4     static int foodPrice = 30000;
5
6     public Waiters(int orderQty, int customerID) {
7         this.orderQty = orderQty;
8         this.customerID = customerID;
9     }
10
11     public void run() {
12         // call getCoffee method and pending it to 5000 ms
13         while (true) {
14             makeFood();
15             try {
16                 Thread.sleep(millis: 15000);
17             } catch (InterruptedException e) {
18                 e.printStackTrace();
19             }
20         }
21     }
22
23     public void orderInfo() {
24
25         System.out.println(x: "-----");
26
27         System.out.println("Customer ID: " + this.customerID);
28         System.out.println("Number of food: " + this.orderQty);
29         System.out.println("Total price: " + foodPrice * this.orderQty);
30     }
31 }
```



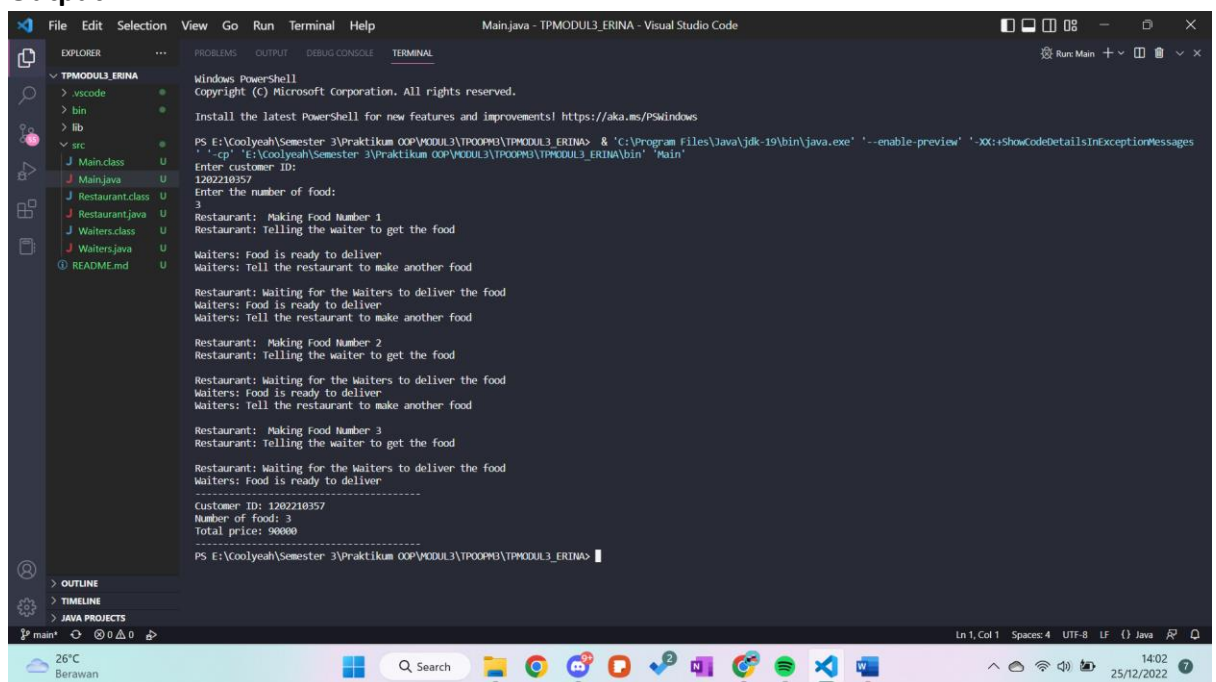
```
23     public void orderInfo() {
24
25         System.out.println(x: "-----");
26
27         System.out.println("Customer ID: " + this.customerID);
28         System.out.println("Number of food: " + this.orderQty);
29         System.out.println("Total price: " + foodPrice * this.orderQty);
30
31         System.out.println(x: "-----");
32     }
33
34     public void makeFood() {
35         synchronized(Restaurant.getLock()) {
36
37             System.out.println(x: "Waiters: Food is ready to deliver");
38             Restaurant restaurant = new Restaurant();
39             restaurant.setWaitingForPickup(waitingForPickup: false);
40
41             if (Restaurant.getFoodNumber() == this.orderQty + 1) {
42                 orderInfo();
43                 System.exit(status: 0);
44             }
45             Restaurant.getLock().notifyAll();
46             System.out.println(x: "Waiters: Tell the restaurant to make another food\n");
47         }
48     }
49
50 }
51 }
```

### 3. Main.java



```
1 import java.util.Scanner;
2
3 public class Main {
4     public static void main(String[] args) {
5         try (Scanner inputorder = new Scanner(System.in)) {
6             Restaurant resto = new Restaurant();
7             int orderQty, customerID;
8
9             try {
10                 System.out.println("Enter customer ID: ");
11                 customerID = inputorder.nextInt();
12
13                 System.out.println("Enter the number of food: ");
14                 orderQty = inputorder.nextInt();
15
16                 Thread tsatu = new Thread(resto);
17                 Waiters waiter = new Waiters(orderQty, customerID);
18                 Thread tdua = new Thread(waiter);
19
20                 tsatu.start();
21                 tdua.start();
22                 tsatu.join();
23                 tdua.join();
24
25             } catch (Exception o) {
26                 System.out.println("Input must be integer");
27             }
28         }
29     }
30 }
31
```

### 4. Output



```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Coolyeh\Semester 3\Praktikum OOP\MODUL3\TP00PM3\TPMODUL3_ERINA> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-enable-preview' '-XX:ShowCodeDetailsInExceptionMessages' '-cp' 'E:\Coolyeh\Semester 3\Praktikum OOP\MODUL3\TP00PM3\TPMODUL3_ERINA\bin' 'Main'
Enter customer ID:
1202210357
Enter the number of food:
3
Restaurant: Making Food Number 1
Restaurant: Telling the waiter to get the food
Waiters: Food is ready to deliver
Waiters: Tell the restaurant to make another food

Restaurant: Waiting for the Waiters to deliver the food
Waiters: Food is ready to deliver
Waiters: Tell the restaurant to make another food

Restaurant: Making Food Number 2
Restaurant: Telling the waiter to get the food

Restaurant: Waiting for the Waiters to deliver the food
Waiters: Food is ready to deliver
Waiters: Tell the restaurant to make another food

Restaurant: Making Food Number 3
Restaurant: Telling the waiter to get the food

Restaurant: Waiting for the Waiters to deliver the food
Waiters: Food is ready to deliver
-----
Customer ID: 1202210357
Number of food: 3
Total price: 90000
-----
PS E:\Coolyeh\Semester 3\Praktikum OOP\MODUL3\TP00PM3\TPMODUL3_ERINA>
```