**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 09 – Nested Class, Inner Class, Nested Anonymous Class”

****

Oleh:

Nama : Muhammad Zaidan Ahbab

NPM : 4523210081

Class : A

Dosen:

Adi Wahyu Pribadi , S.Si., M.Kom

**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2024**

**Sistem Manajemen Buku**

Source Code Library.java

public class Library {

    static String LibraryName;

    public static String getLibraryName() {

        return LibraryName;

    }

    static class Book {

        private String title;

        private String author;

        private String isbn;

        public Book(String title, String author, String isbn) {

            this.title = title;

            this.author = author;

            this.isbn = isbn;

        }

        public void displayBookInfo() {

            System.out.println("Nama Perpustakaan : " + Library.getLibraryName());

            System.out.println("Judul Buku        : " + title);

            System.out.println("Penulis           : " + author);

            System.out.println("ISBN              : " + isbn);

            System.out.println("-----------------------------------");

        }

    }

}

Source Code Main.java

public class Main {

    public static void main(String[] args) {

        Library.LibraryName = "Perpustakaan Kota";

        Library.Book book1 = new Library.Book("Pemrograman Java", "Budi", "1234567890");

        Library.Book book2 = new Library.Book("Algoritma dan Struktur Data", "Siti", "0987654321");

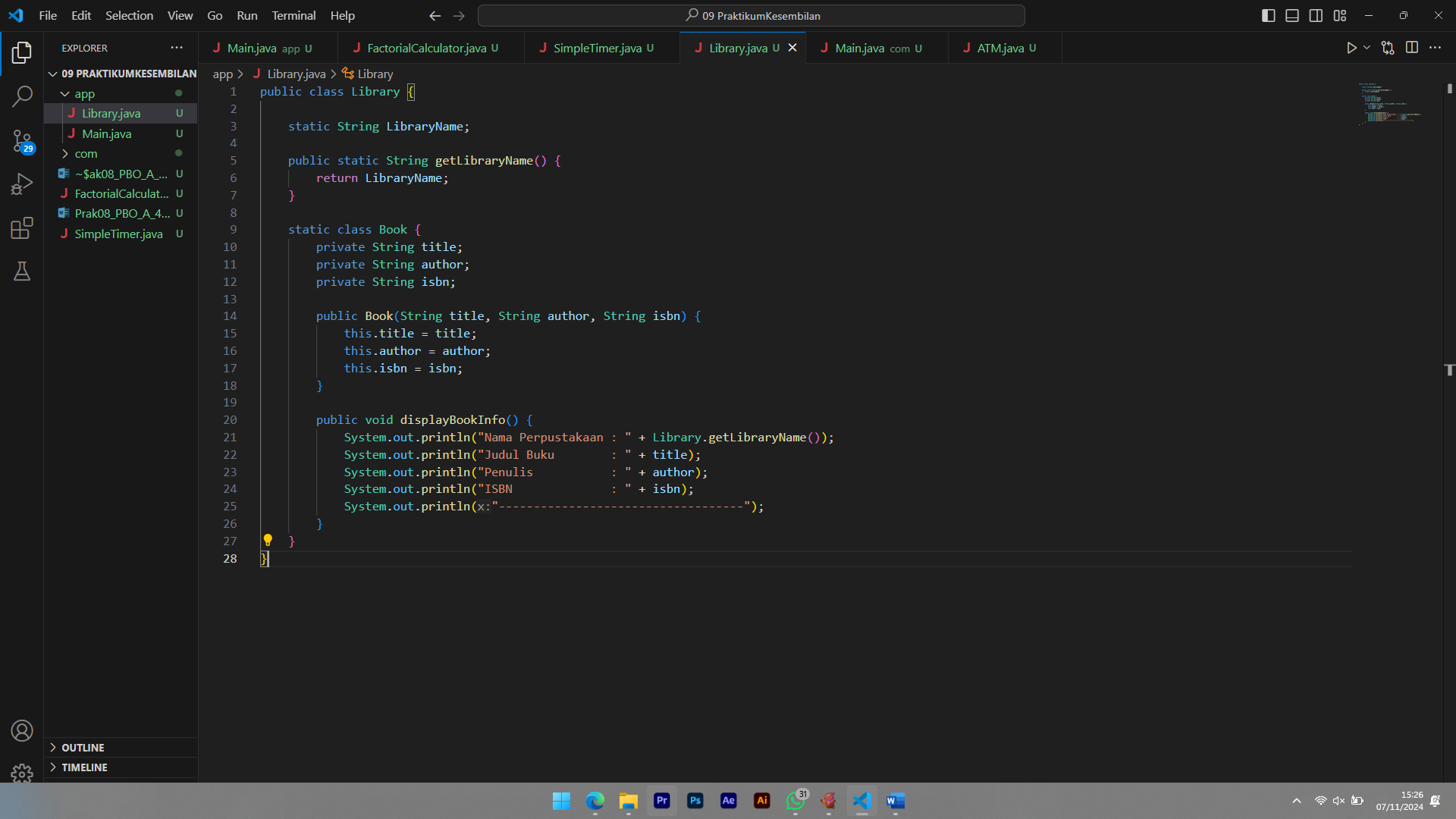
        book1.displayBookInfo();

        book2.displayBookInfo();

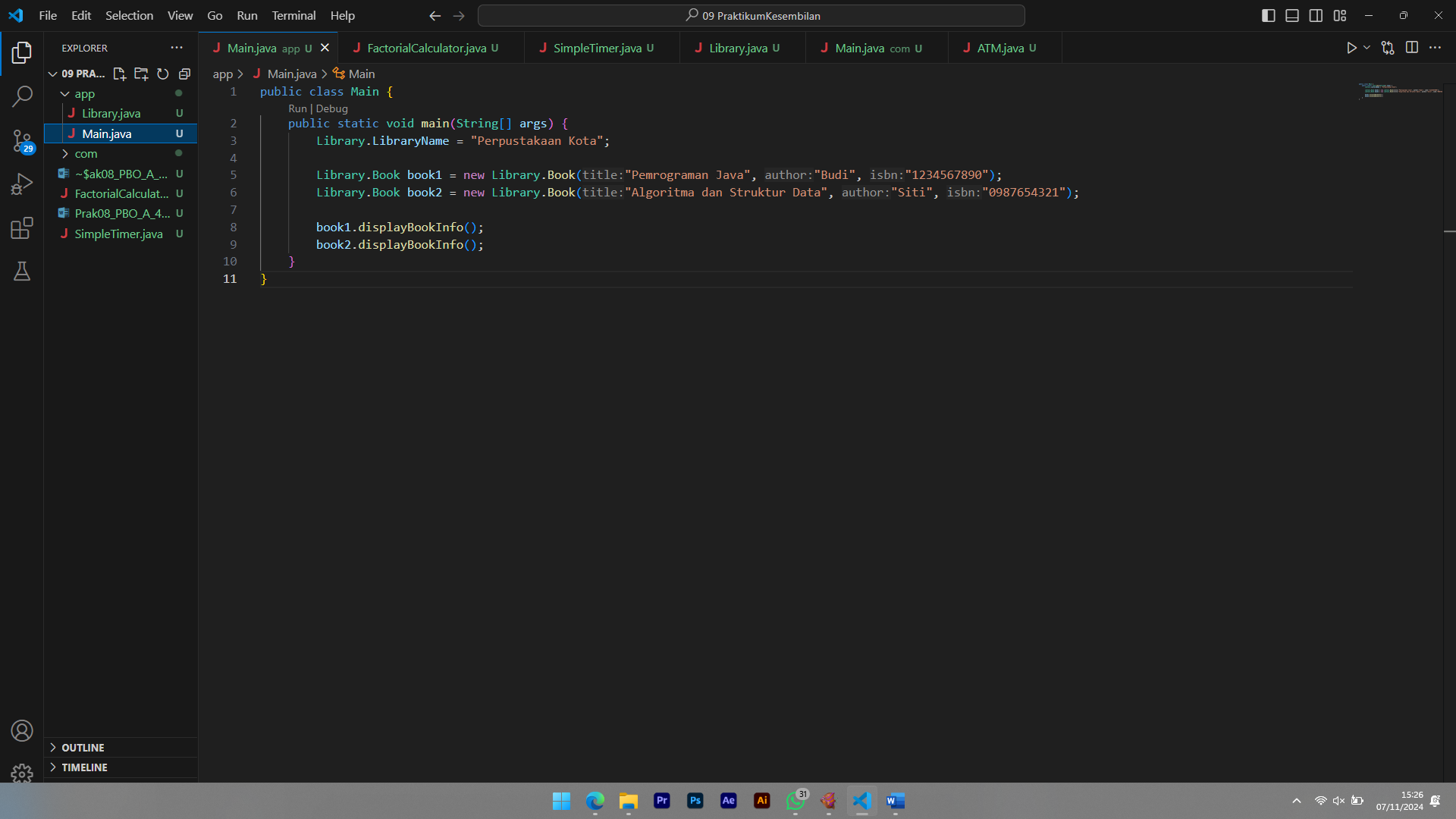
    }

}

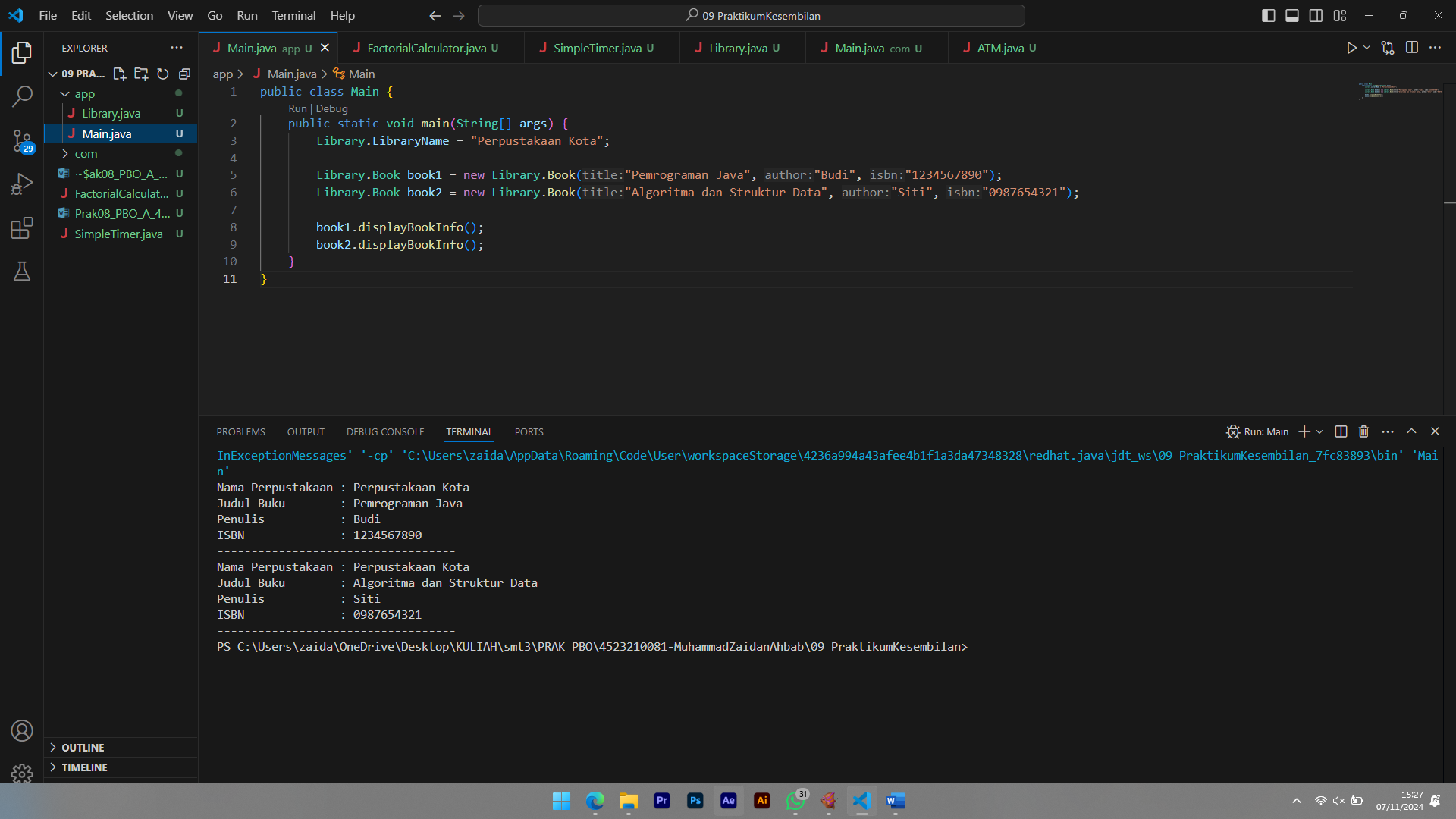
Tampilan Library.java



Tampilan Main.java



Hasil Running



**Sistem ATM**

Source Code ATM.java

package com;

public class ATM {

    private String location;

    public ATM(String location) {

        this.location = location;

    }

    class BankAccount {

        private String accountNumber;

        private double balance;

        public BankAccount(String accountNumber) {

            this.accountNumber = accountNumber;

            this.balance = 0.0;

        }

        public void deposit(double amount) {

            if (amount > 0) {

                balance += amount;

                System.out.println("Deposit        : RP." + String.format("%,.2f", amount));

            } else {

                System.out.println("Jumlah dposit harus positif.");

            }

        }

        public void withdraw(double amount) {

            if (amount > 0 && amount <= balance) {

                balance -= amount;

                System.out.println("Withdraw       : Rp." + String.format("%,.2f", amount));

            } else {

                System.out.println("Penarikan tidak valid.");

            }

        }

        public void displayAccountInfo() {

            System.out.println("Lokasi ATM     : " + ATM.this.location);

            System.out.println("Nomor Akun     : " + accountNumber);

            System.out.println("Saldo Sekarang : Rp." + String.format("%,.2f", balance));

            System.out.println("-----------------------------------");

        }

    }

}

Source Code Main.java

package com;

public class Main {

    public static void main(String[] args) {

        ATM atm = new ATM("Jakarta");

        ATM.BankAccount account = atm.new BankAccount("0011223344");

        account.displayAccountInfo();

        account.deposit(1000000);

        account.displayAccountInfo();

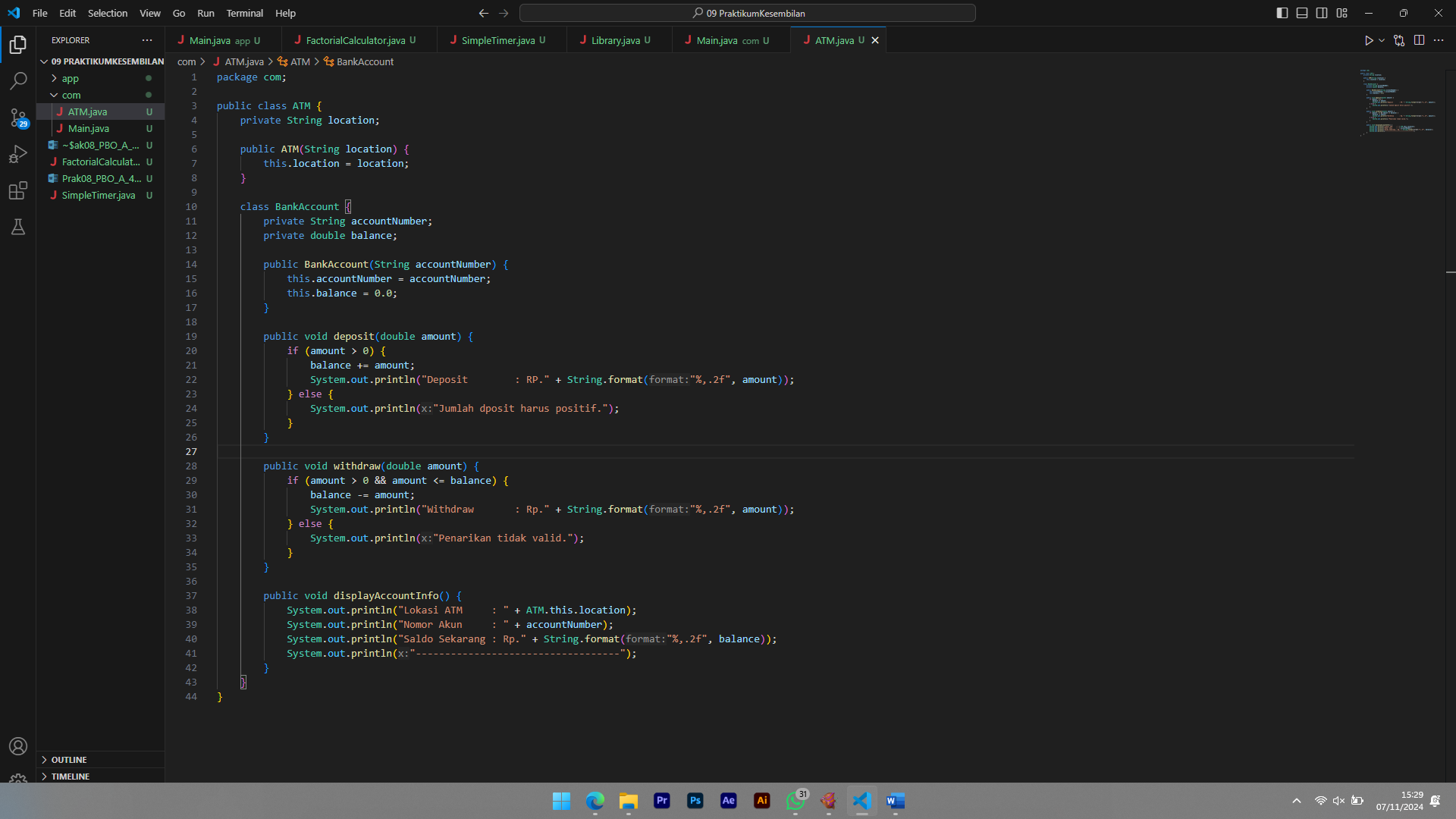
        account.withdraw(500000);

        account.displayAccountInfo();

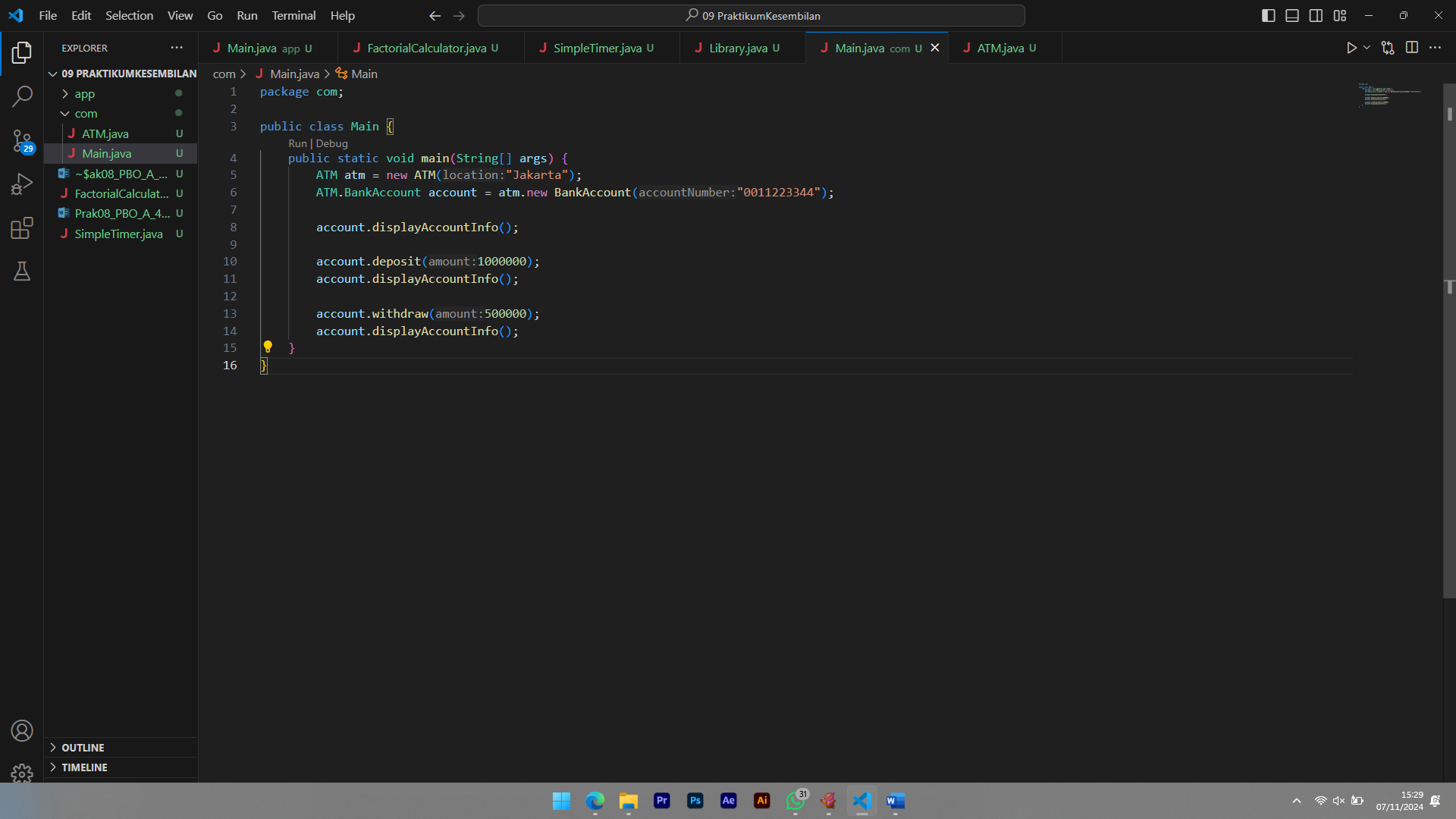
    }

}

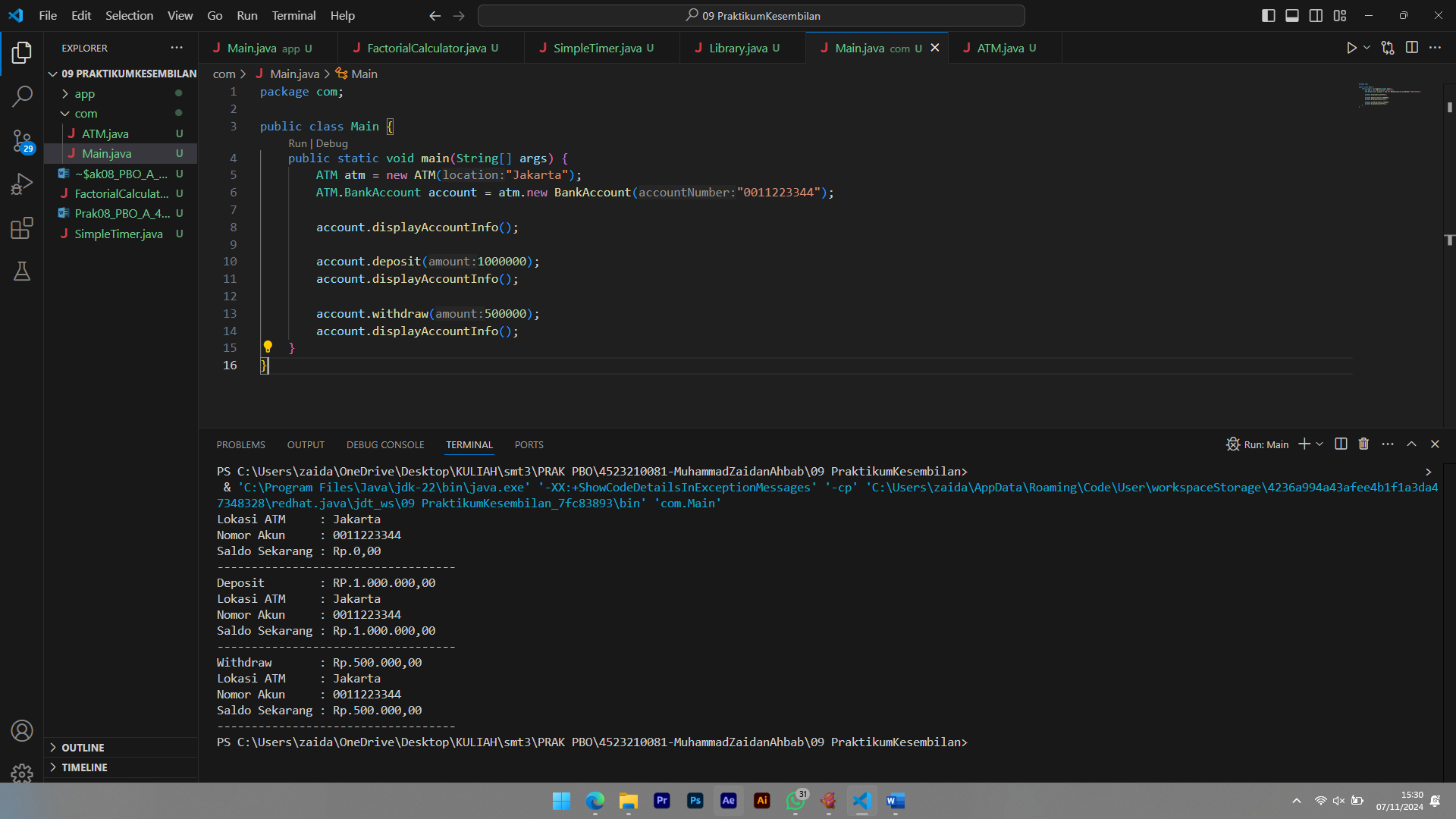
Tampilan ATM.java



Tampilan Main.java



Hasil Running



**Sistem Factorial Calculator**

Source Code FactorialCalculator.java

import java.util.Scanner;

public class FactorialCalculator {

    public void calculate(int number) {

        class Factorial {

            private int n;

            public Factorial(int n) {

                this.n = n;

            }

            public int getResult() {

                int result = 1;

                for (int i = 2; i <= n; i++) {

                    result \*= i;

                }

                return result;

            }

        }

        Factorial factorial = new Factorial(number);

        int result = factorial.getResult();

        System.out.println("Faktorialdari" + number + "adalah" + result);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkanangka:");

        int num = scanner.nextInt();

        FactorialCalculator calculator = new FactorialCalculator();

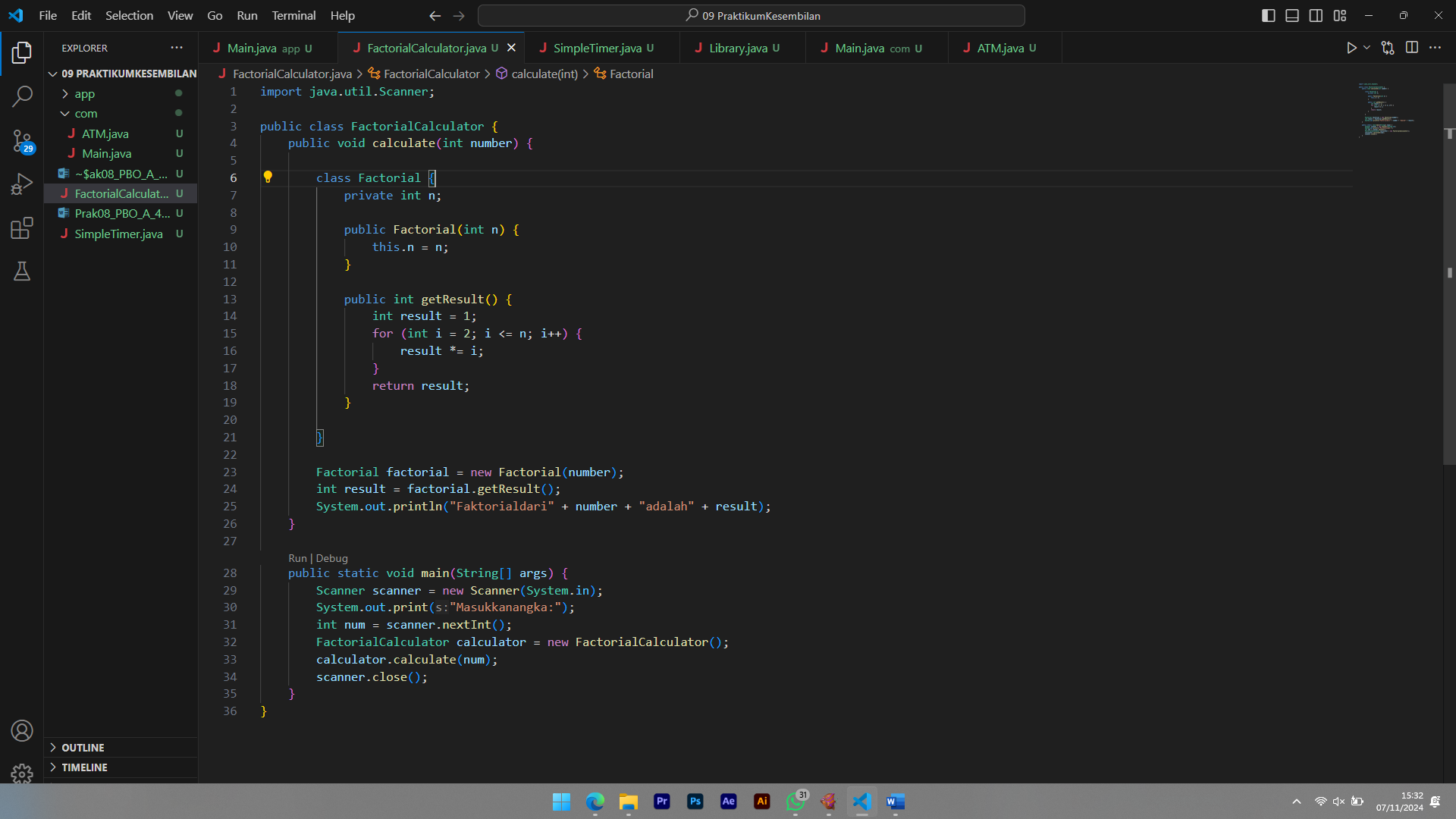
        calculator.calculate(num);

        scanner.close();

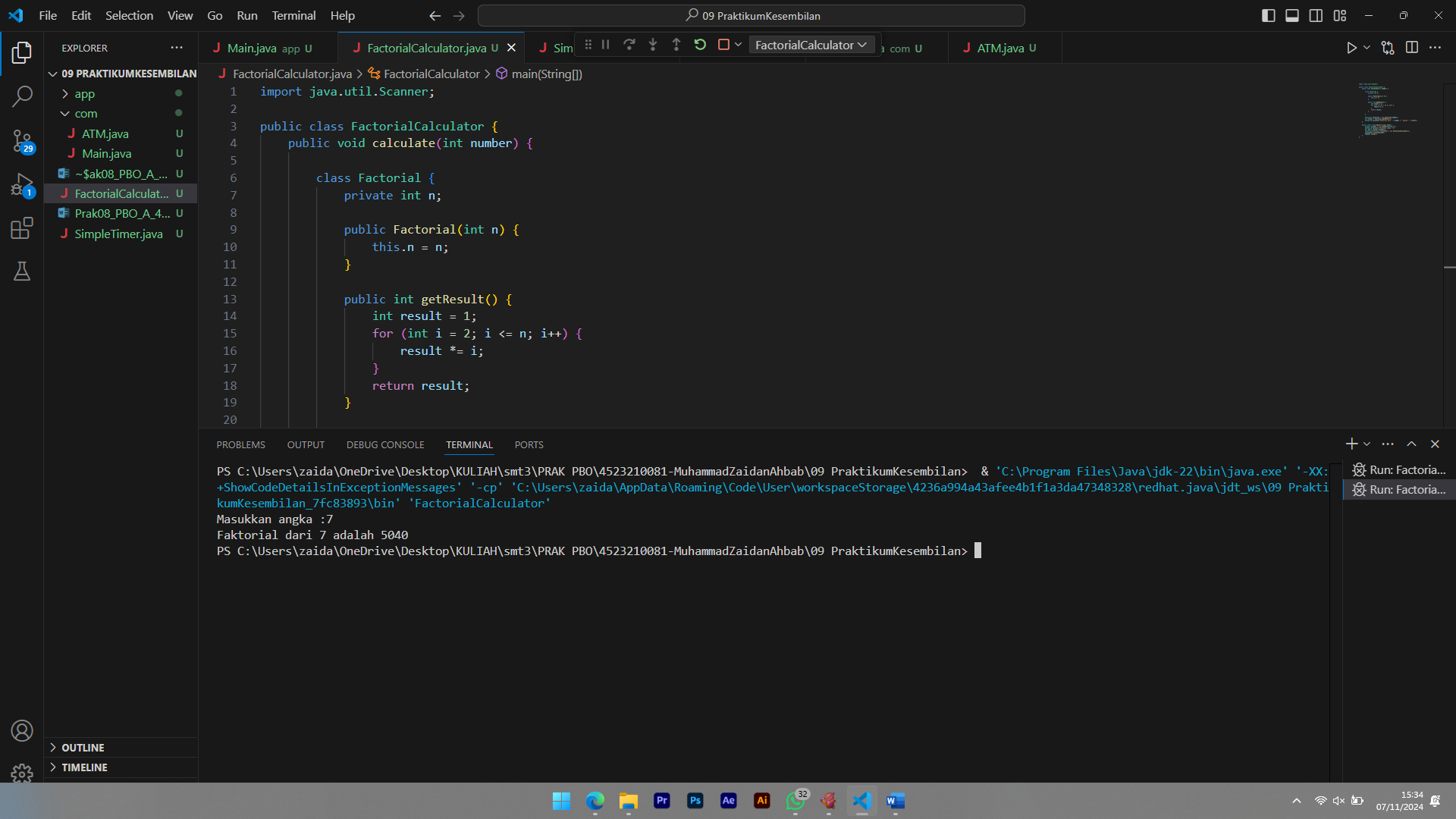
    }

}

Tampilan FactorialCalculator.java



Hasil Running



Program Java ini menghitung faktorial dari angka yang dimasukkan user.

Cara Kerja Program

1. Program meminta user untuk memasukkan angka.
2. Angka tersebut dikirim ke metode calculate(int number) dari kelas FactorialCalculator.
3. Di dalam metode calculate, terdapat kelas lokal bernama Factorial, yang memiliki metode getResult() untuk menghitung faktorial dari angka yang diberikan.
4. Setelah kalkulasi selesai, hasil faktorial ditampilkan di layar.

Misalkan pengguna memasukkan angka 7, berikut adalah langkah-langkah kalkulasinya:

* Faktorial dari 7 adalah 7! = 7 \* 6 \* 5 \* 4 \* 3 \* 2 \* 1 = 5040.

**Sistem AnonymousInnerClass-TimerSederhana**

Source Code SimpleTimer.java

import java.util.Timer;

import java.util.TimerTask;

public class SimpleTimer {

    public static void main(String[] args) {

        System.out.println("Timer dimulai ");

        Timer timer = new Timer();

        TimerTask task = new TimerTask() {

            int counter = 1;

            @Override

            public void run() {

                if (counter <= 5) {

                    System.out.println("Detik ke" + counter);

                    counter++;

                } else {

                    System.out.println("Timer selesai.");

                    timer.cancel();

                }

            }

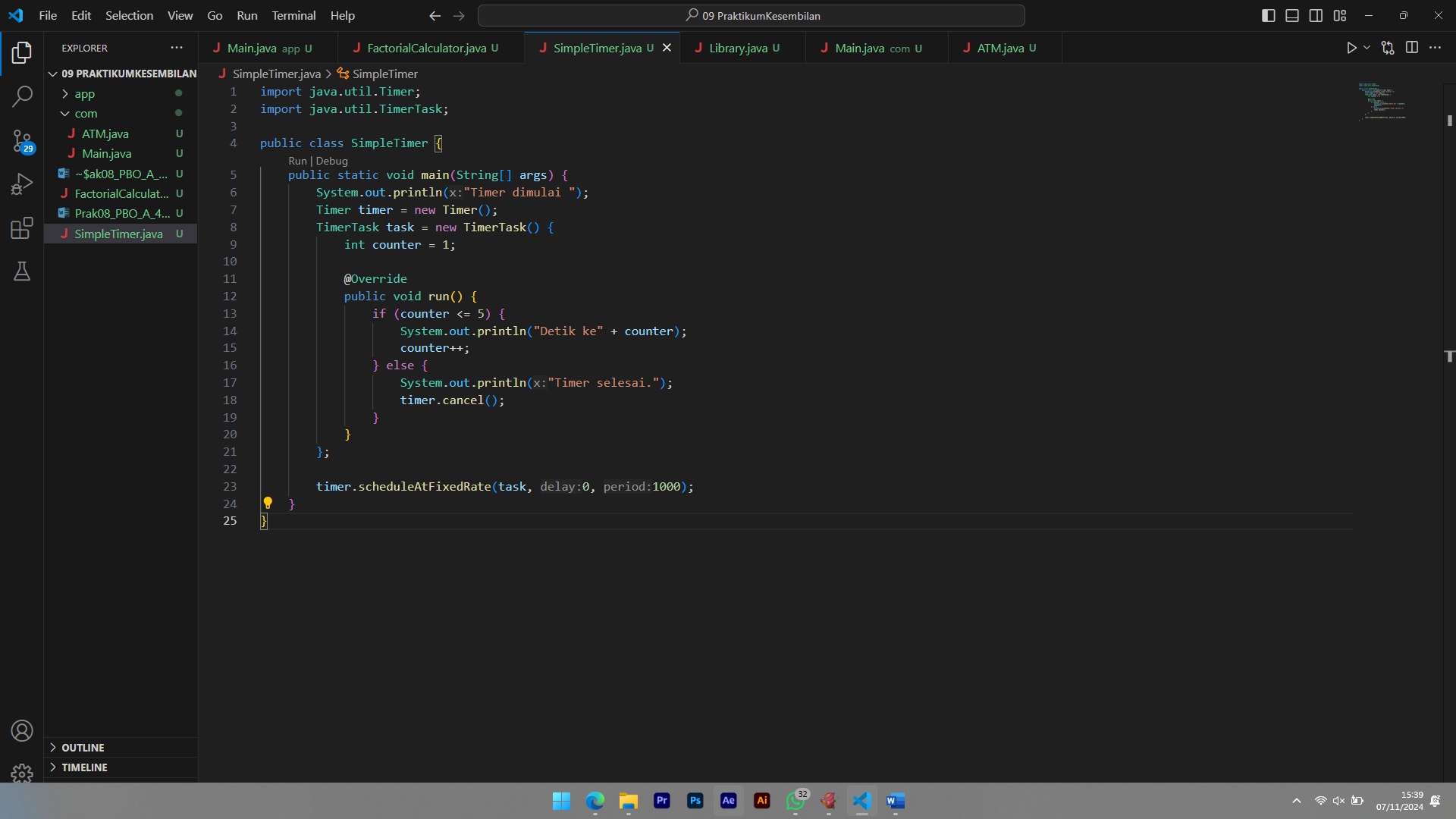
        };

        timer.scheduleAtFixedRate(task, 0, 1000);

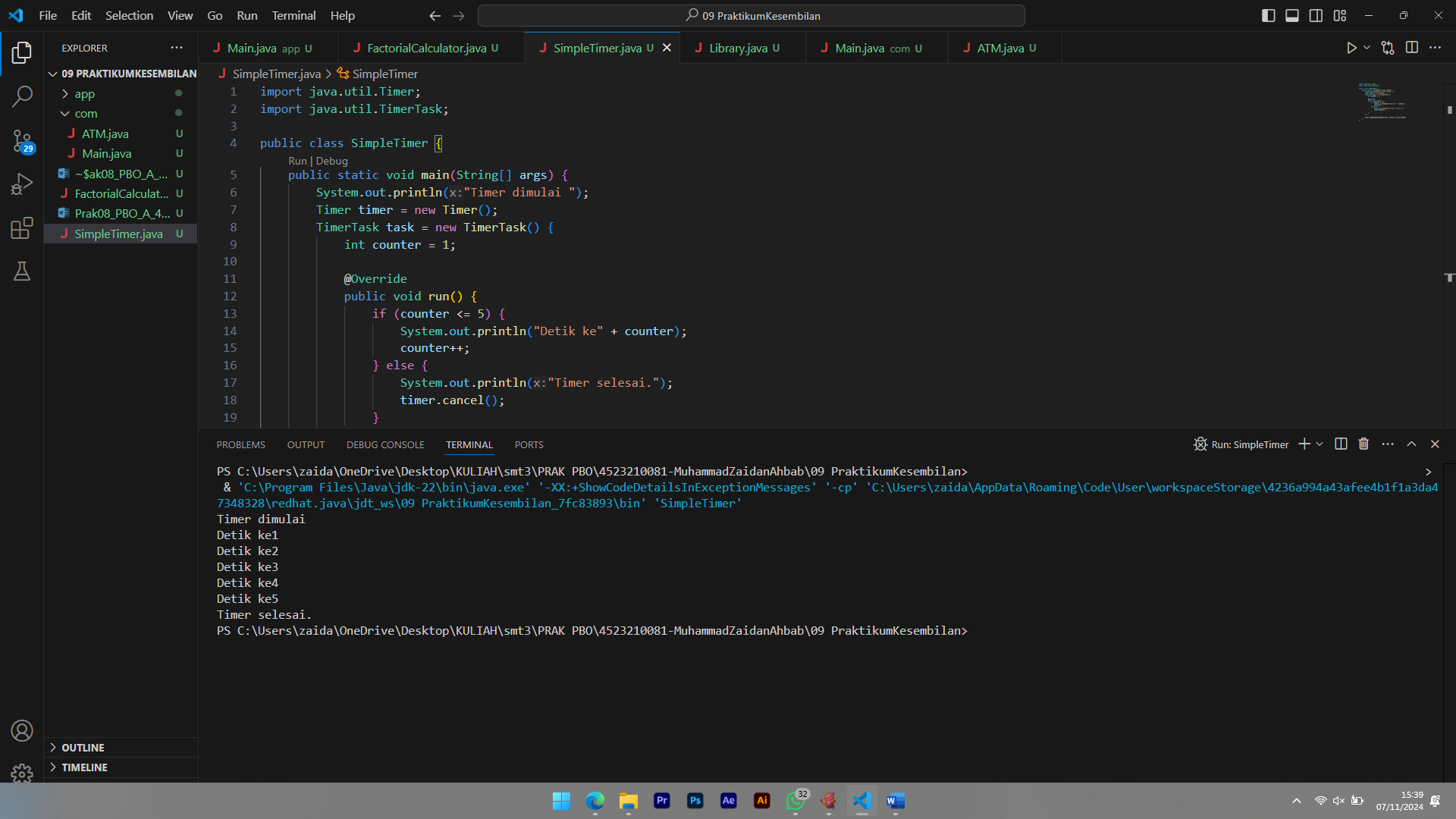
    }

}

Tampilan FactorialCalculator.java



Hasil Running



Program Java ini membuat timer sederhana yang mencetak pesan setiap detik hingga lima detik.

### Cara Kerja Program

1. Program memulai dengan mencetak pesan "Timer dimulai".
2. Objek Timer dan TimerTask dibuat. TimerTask berisi metode run() yang dieksekusi setiap detik.
3. Variabel counter diinisialisasi ke 1. Setiap detik, counter bertambah satu, dan program mencetak "Detik ke X", dengan X adalah nilai counter.
4. Ketika counter mencapai angka 6, program mencetak "Timer selesai." dan menghentikan timer dengan timer.cancel().