**DASAR PEMROGRAMAN JOBSHEET 14**

****

**HANIEF MOCHSIN**

**KELAS 1B (13)**

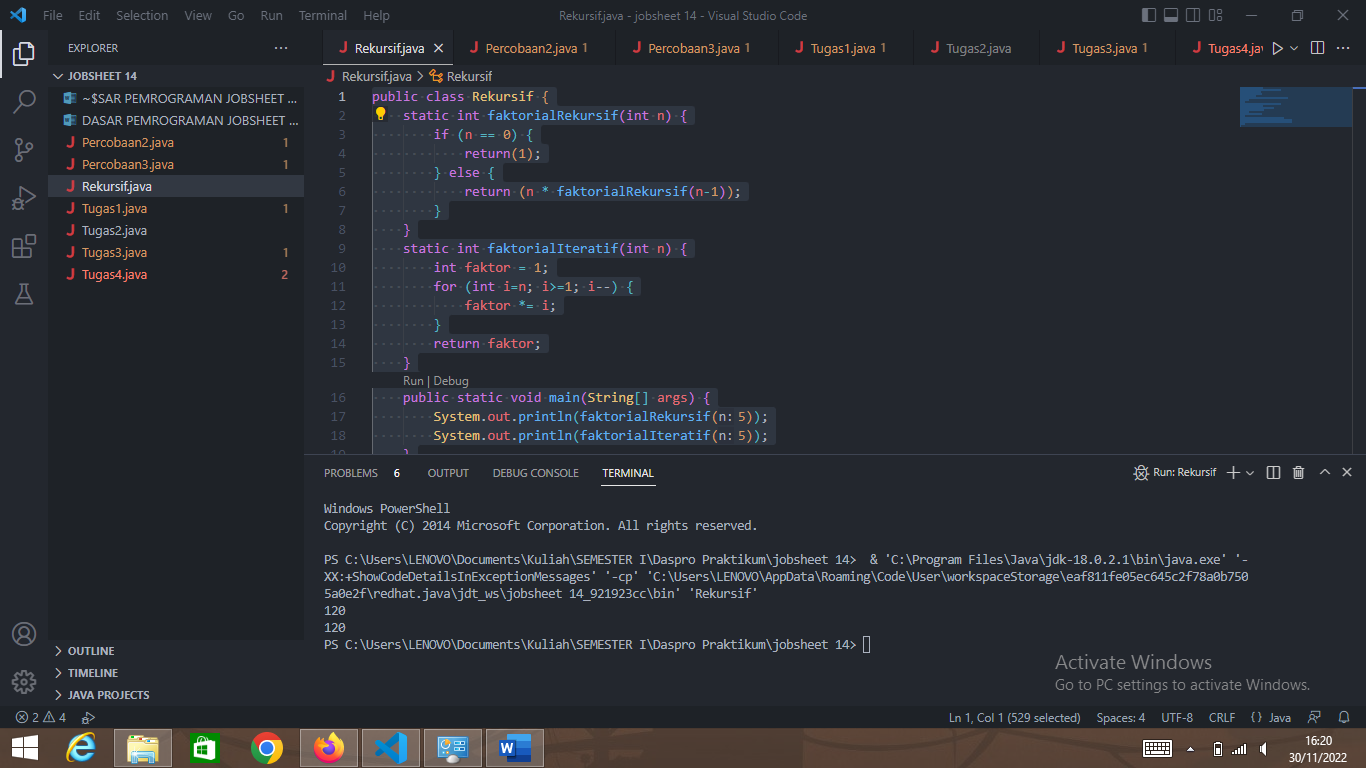
**2241720181**

**TEKNIK INFORMATIKA**

**TEKNOLOGI INFORMASI**

**PERCOBAAN 1**

1. public class Rekursif {
2. static int faktorialRekursif(int n) {
3. if (n == 0) {
4. return(1);
5. } else {
6. return (n \* faktorialRekursif(n-1));
7. }
8. }
9. static int faktorialIteratif(int n) {
10. int faktor = 1;
11. for (int i=n; i>=1; i--) {
12. faktor \*= i;
13. }
14. return faktor;
15. }
16. public static void main(String[] args) {
17. System.out.println(faktorialRekursif(5));
18. System.out.println(faktorialIteratif(5));
19. }
20. }



**PERTANYAAN**

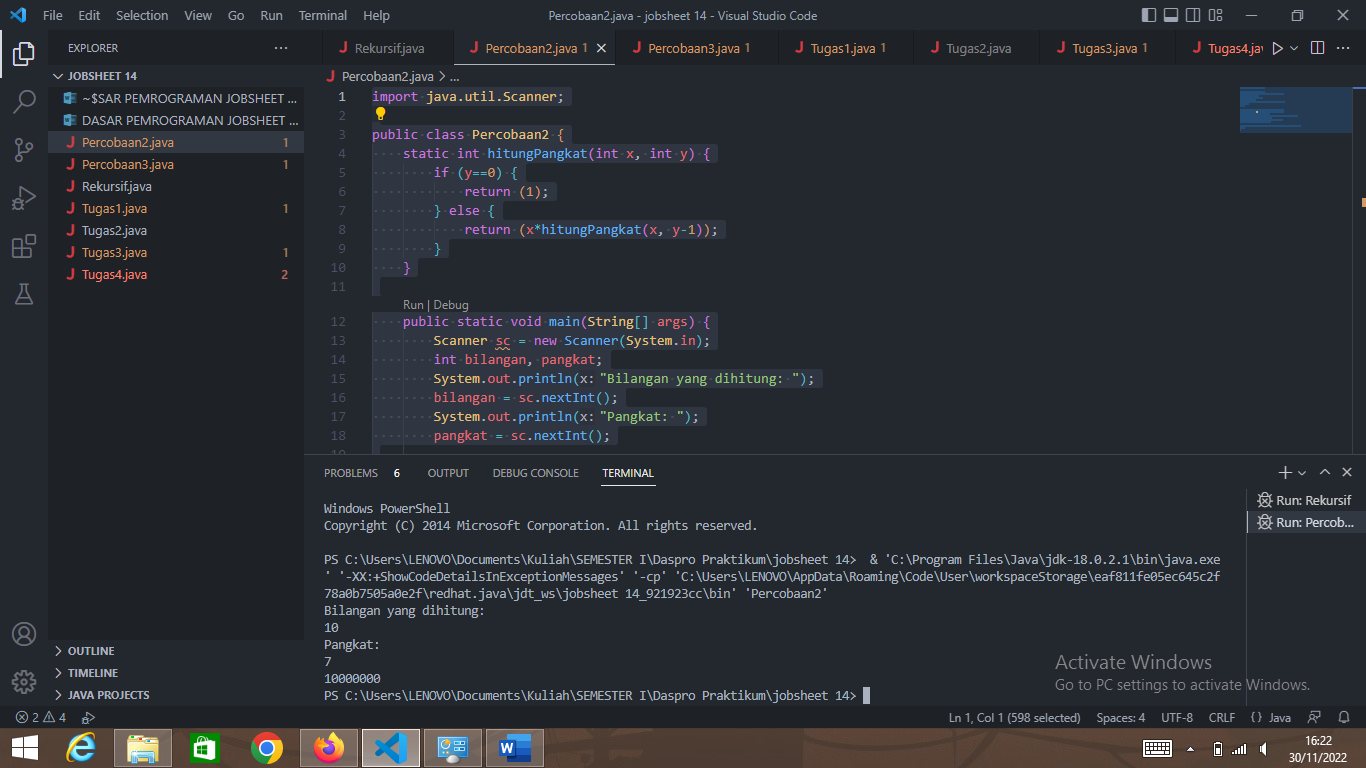
1. Apa yang dimaksud dengan fungsi rekursif?
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
3. Pada Percobaan1, apakah hasil yang diberikan fungsi faktorialRekursif() dan fungsi faktorialIteratif() sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

**JAWAB**

1. Fungsi yang memanggil dirinya sendiri
2. Membuat deret fibonacci, menghitung pangkat, menghitung faktorial, menghitung maksimum dan minimum.
3. Jika iteratif menggunakan perulangan for, sedangkan rekursif menggunakan if else.

**PERCOBAAN 2**

1. import java.util.Scanner;
2. public class Percobaan2 {
3. static int hitungPangkat(int x, int y) {
4. if (y==0) {
5. return (1);
6. } else {
7. return (x\*hitungPangkat(x, y-1));
8. }
9. }
10. public static void main(String[] args) {
11. Scanner sc = new Scanner(System.in);
12. int bilangan, pangkat;
13. System.out.println("Bilangan yang dihitung: ");
14. bilangan = sc.nextInt();
15. System.out.println("Pangkat: ");
16. pangkat = sc.nextInt();
17. System.out.println(hitungPangkat(bilangan, pangkat));
18. }
19. }



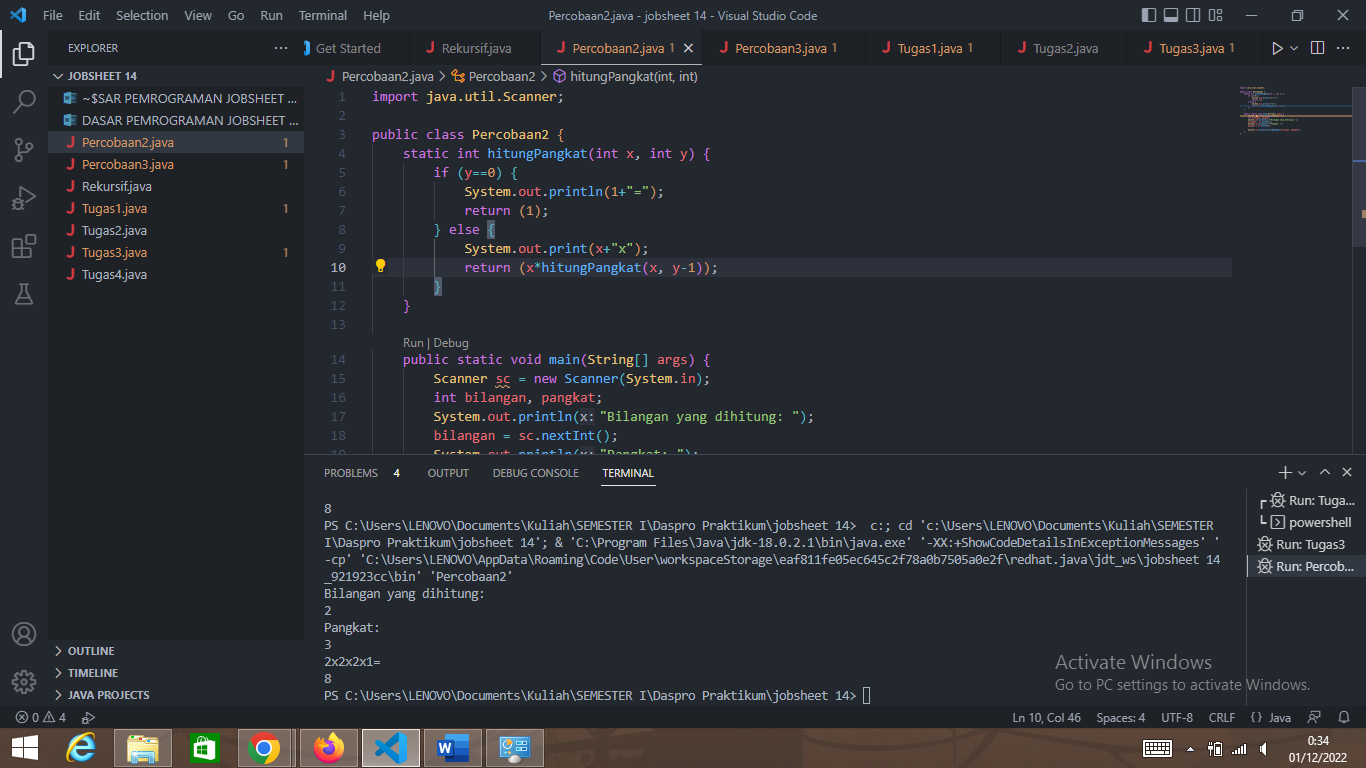
**PERTANYAAN**

1. Pada Percobaan2, terdapat pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) pada fungsi main, kemudian dilakukan pemanggilan fungsi hitungPangkat() secara berulangkali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : hitungPangkat(2,5) dicetak 2x2x2x2x2x1 = 32

**JAWAB**

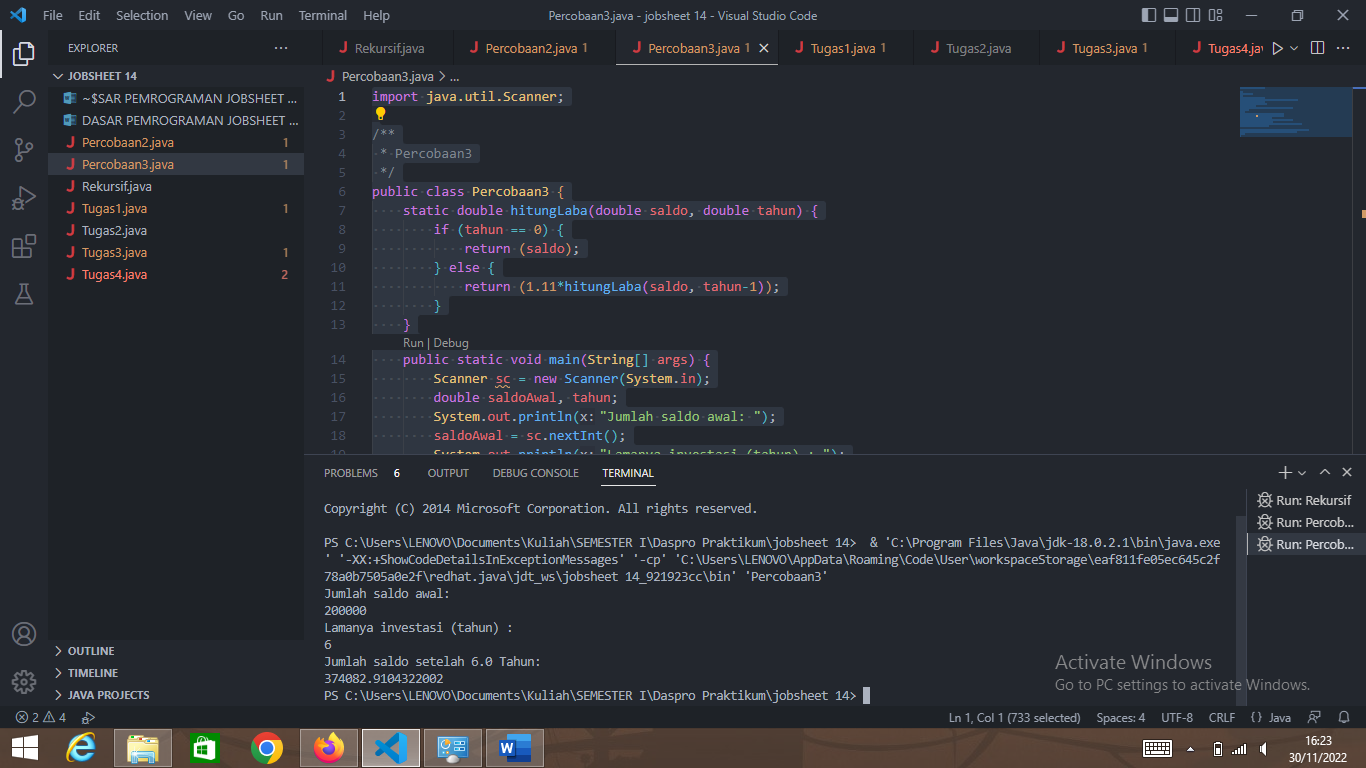
1. Sampai base case dari fungsi tersebut atau pangkat yang di inputkan itu mencapai syarat.
2. Code =
3. import java.util.Scanner;
4. public class Percobaan2 {
5. static int hitungPangkat(int x, int y) {
6. if (y==0) {
7. System.out.println(1+"=");
8. return (1);
9. } else {
10. System.out.print(x+"x");
11. return (x\*hitungPangkat(x, y-1));
12. }
13. }
14. public static void main(String[] args) {
15. Scanner sc = new Scanner(System.in);
16. int bilangan, pangkat;
17. System.out.println("Bilangan yang dihitung: ");
18. bilangan = sc.nextInt();
19. System.out.println("Pangkat: ");
20. pangkat = sc.nextInt();
21. System.out.println(hitungPangkat(bilangan, pangkat));
22. }
23. }

Result =



**PERCOBAAN 3**

1. import java.util.Scanner;
2. /\*\*
3. \* Percobaan3
4. \*/
5. public class Percobaan3 {
6. static double hitungLaba(double saldo, double tahun) {
7. if (tahun == 0) {
8. return (saldo);
9. } else {
10. return (1.11\*hitungLaba(saldo, tahun-1));
11. }
12. }
13. public static void main(String[] args) {
14. Scanner sc = new Scanner(System.in);
15. double saldoAwal, tahun;
16. System.out.println("Jumlah saldo awal: ");
17. saldoAwal = sc.nextInt();
18. System.out.println("Lamanya investasi (tahun) : ");
19. tahun = sc.nextInt();
20. System.out.println("Jumlah saldo setelah "+tahun+" Tahun: ");
21. System.out.println(hitungLaba(saldoAwal, tahun));
22. }
23. }



**PERTANYAAN**

1. Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!
2. Jabarkan trace fase ekspansi dan fase subtitusi algoritma perhitungan laba di atas jika diberikan nilai hitungLaba(100000,3)

**JAWAB**

1. Pada fungsi hitungLaba didalam if terdapat parameter tahun == 0 merupakan base casenya. Sedangkan pada else nya terdapat pemanggilan fungsi didalam return maka itu adalah rekursion callnya.
2. Ekspansi=

1.11\*hitungLaba(100000,2)

1.11\*(1.11\*hitungLaba(100000,1))

1.11\*(1.11(1.11\*hitungLaba(100000,0)))

Subtitusi=

1.11\*(1.11(1.11\*100000))

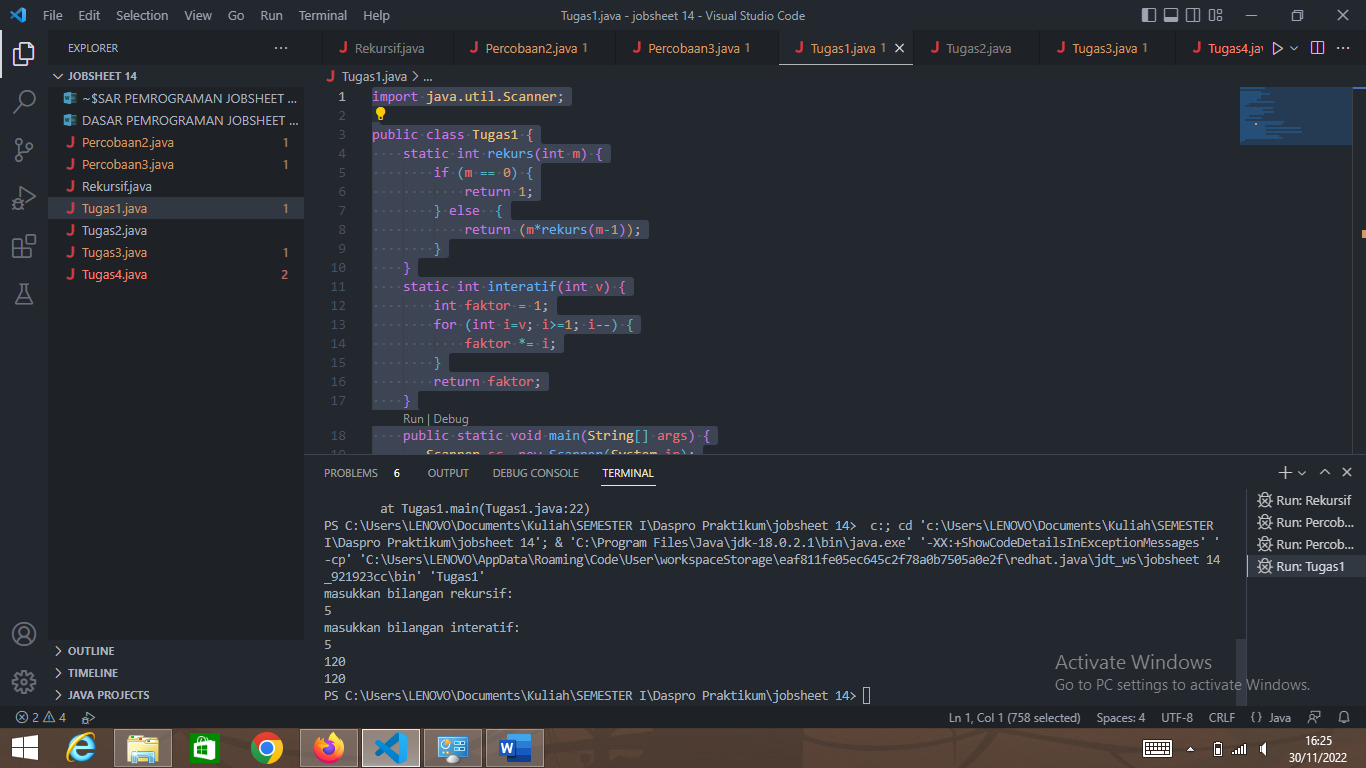
1.11\*(1.11\*111000.00000000001)

1.11\*123210.00000000003

136763.10000000003

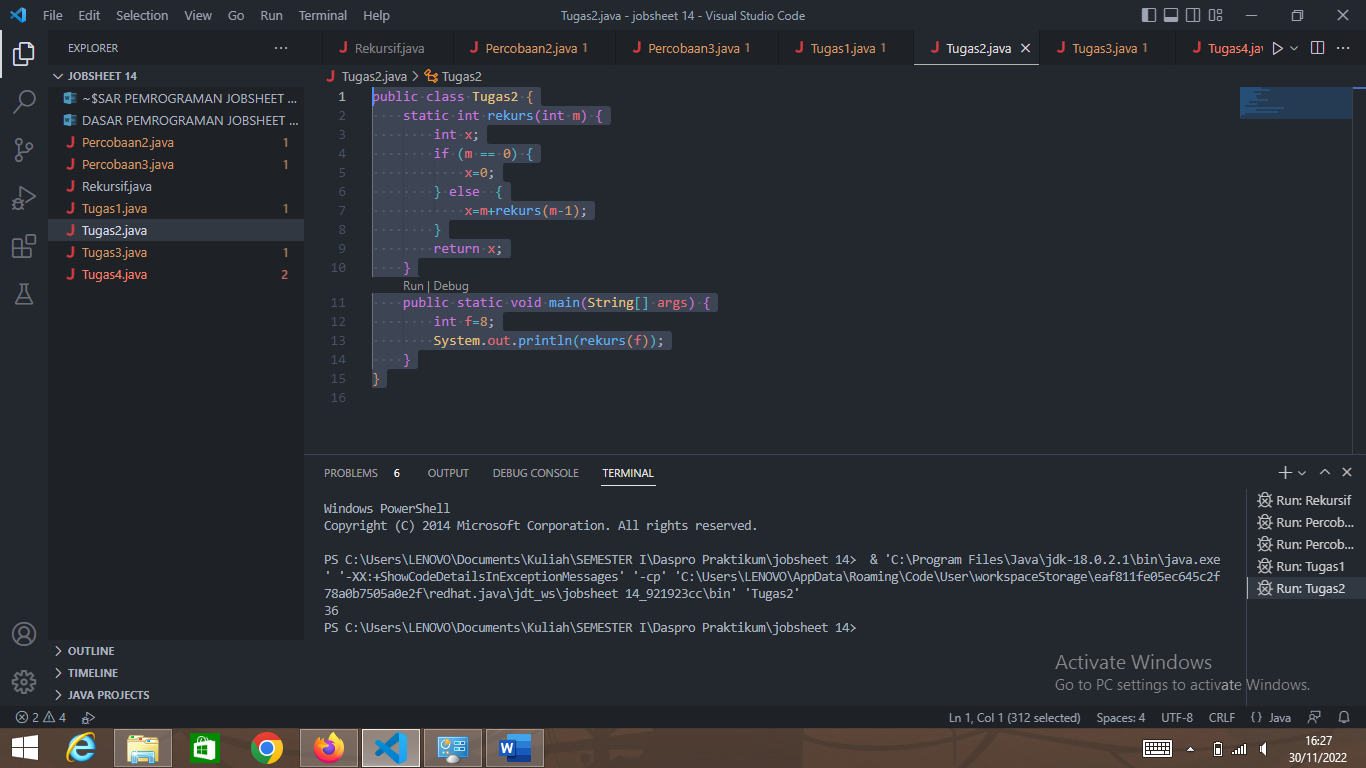
**TUGAS 1**

1. import java.util.Scanner;
2. public class Tugas1 {
3. static int rekurs(int m) {
4. if (m == 0) {
5. return 1;
6. } else  {
7. return (m\*rekurs(m-1));
8. }
9. }
10. static int interatif(int v) {
11. int faktor = 1;
12. for (int i=v; i>=1; i--) {
13. faktor \*= i;
14. }
15. return faktor;
16. }
17. public static void main(String[] args) {
18. Scanner sc =new Scanner(System.in);
19. int bil=0, bil1=0;
20. System.out.println("masukkan bilangan rekursif: ");
21. bil = sc.nextInt();
22. System.out.println("masukkan bilangan interatif: ");
23. bil1= sc.nextInt();
24. System.out.println(rekurs(bil));
25. System.out.println(interatif(bil1));
26. }
27. }



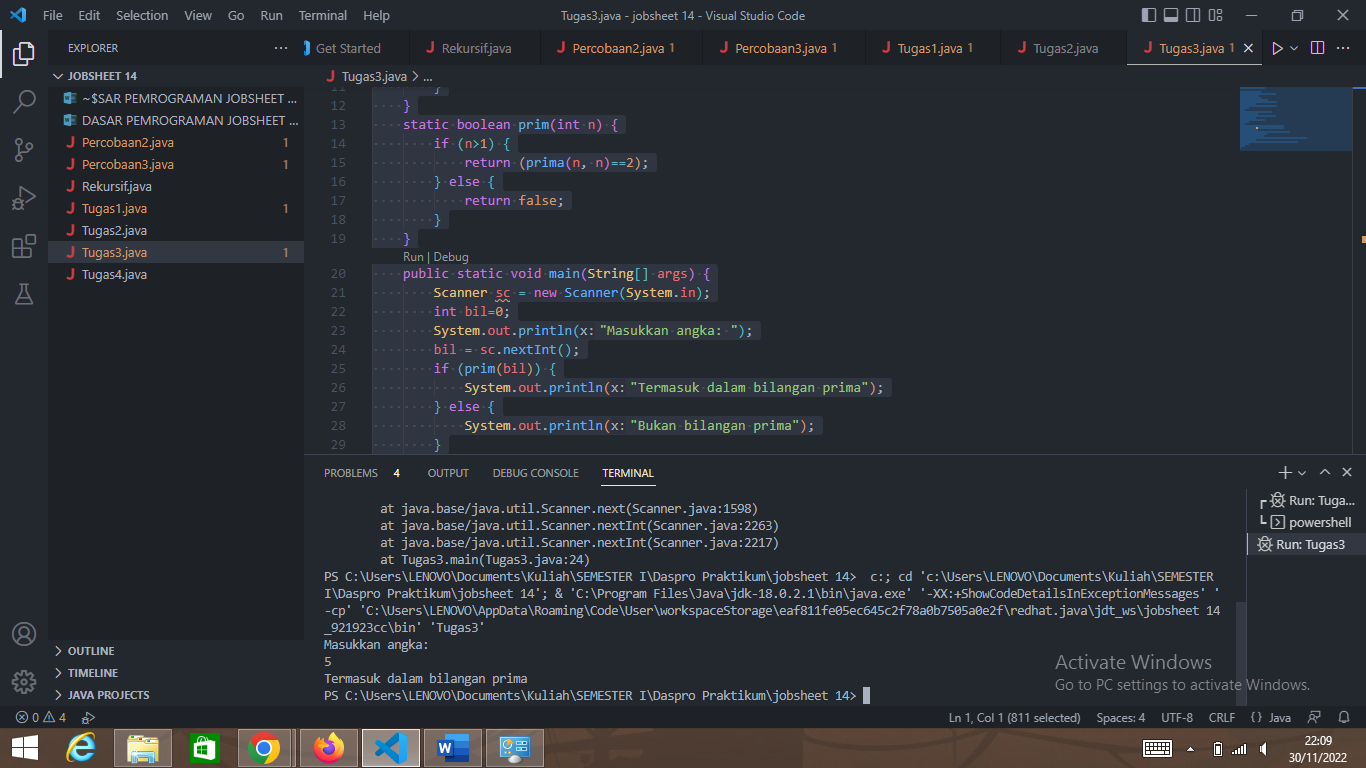
**TUGAS 2**

1. public class Tugas2 {
2. static int rekurs(int m) {
3. int x;
4. if (m == 0) {
5. x=0;
6. } else  {
7. x=m+rekurs(m-1);
8. }
9. return x;
10. }
11. public static void main(String[] args) {
12. int f=8;
13. System.out.println(rekurs(f));
14. }
15. }



**TUGAS 3**

1. import java.util.Scanner;
2. public class Tugas3 {
3. static int prima(int o, int p) {
4. if (p==1) {
5. return 1;
6. } else if (o%p==0) {
7. return 1 + prima(o, --p);
8. } else {
9. return 0 + prima(o, --p);
10. }
11. }
12. static boolean prim(int n) {
13. if (n>1) {
14. return (prima(n, n)==2);
15. } else {
16. return false;
17. }
18. }
19. public static void main(String[] args) {
20. Scanner sc = new Scanner(System.in);
21. int bil=0;
22. System.out.println("Masukkan angka: ");
23. bil = sc.nextInt();
24. if (prim(bil)) {
25. System.out.println("Termasuk dalam bilangan prima");
26. } else {
27. System.out.println("Bukan bilangan prima");
28. }
29. }
30. }



**TUGAS 4**

import java.util.Scanner;

public class Tugas4 {

    static Scanner sc = new Scanner(System.in);

    static int angka=0, bilangan=1, nomor=0;

    static int marmut(int first, int second) {

        if (first == 0) {

            return 0;

        } else if (first > 0 ) {

            nomor = angka + bilangan;

            pasangan(angka, bilangan, nomor, first, second);

            angka = bilangan;

            bilangan = nomor;

            marmut(first - 1, ++second);

        }

        return 0;

    }

    static void pasangan(int q, int w, int e, int limit, int t) {

        System.out.print("bulan ke : "+t+" | "+q+" | "+w+" | "+e+" |");

        System.out.println();

    }

    public static void main(String[] args) {

        int p = 1;

        System.out.print("Masukkan Bulan: ");

        int limit = sc.nextInt();

        marmut(limit, p);

    }

}

