

LAPORAN PRAKTIKUM TEKNIK PEMROGRAMAN : PRAKTIKUM REKURSIF

Mata Kuliah : Teknik Pemrograman

Dosen Pengampu : Dwi Rolliawaati, M.T.



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun Oleh:

Nama M.Khotibul Umam

NIM : 09020620031

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA

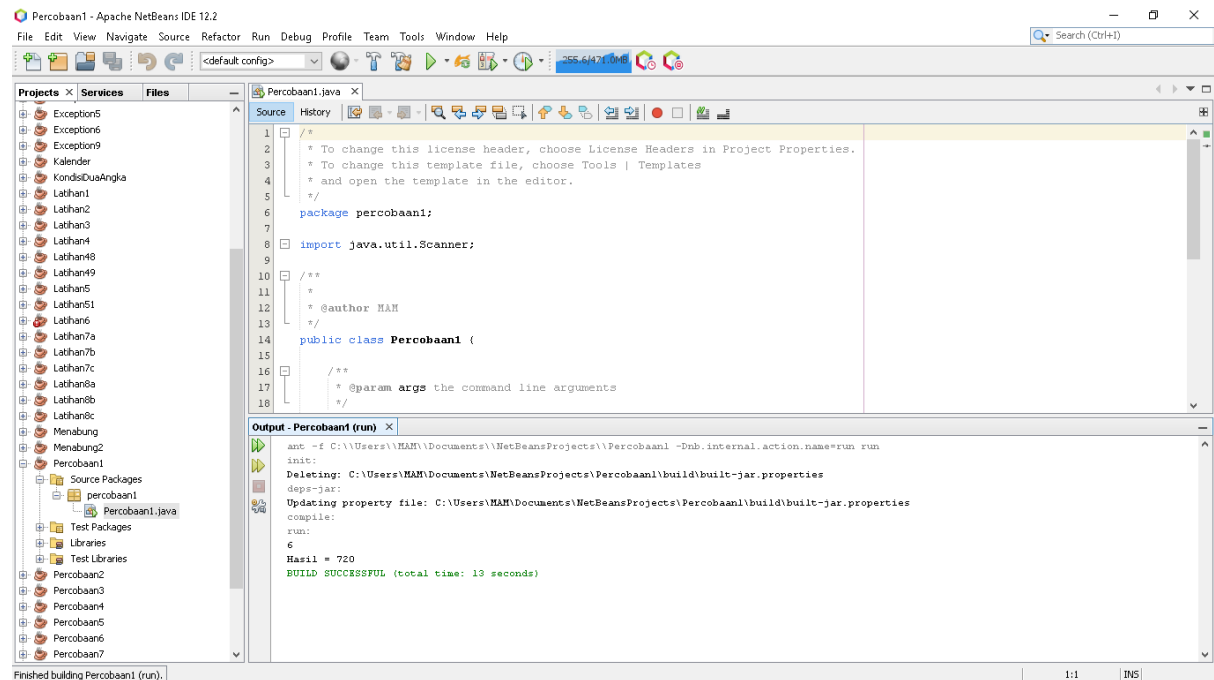
FAKULTAS SAINS DAN TEKNOLOGI

SISTEM INFORMASI 2021

DAFTAR ISI

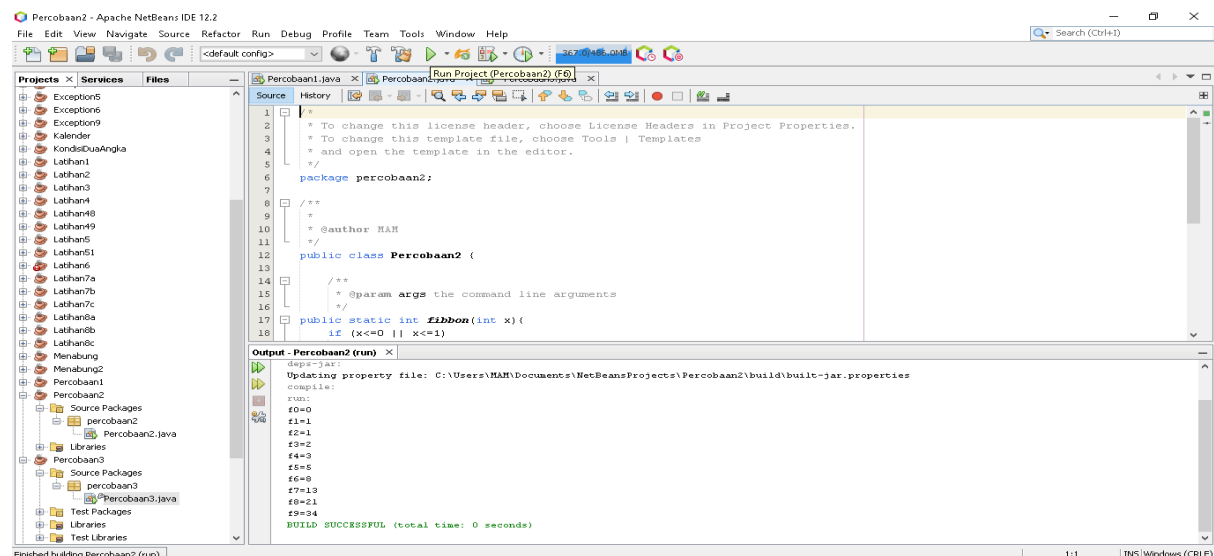
Latihan Percobaan 1	3
Latihan Percobaan 2	3
Latihan Percobaan 3	4
Latihan Percobaan 4	5
Latihan Percobaan 5	5
Latihan Percobaan 6	6
Latihan Percobaan 7	7
Latihan Percobaan 8	8
Latihan Percobaan 9	9
Latihan Percobaan 10	10
Latihan Percobaan 11	10
Latihan Percobaan 12	11
Latihan Percobaan 13	11

Percobaan 1



Pada proses pertama ini kita akan mencari nilai faktorial dari nilai yang dimasukan oleh pengguna. Diawal kita membuat fungsi rekursif dimana jika nilai yang dimasukan adalah 1 maka nilai balik (return value) adalah nilai itu sendiri. Sementara jika tidak maka akan dihitung menggunakan rumus faktorial yaitu $(x * \text{faktorial}(x-1))$. terkait *input/output* pada program, Karena jenis program yang saya buat maka saya perlu inisialisasi fungsi rekursifnya maka di awal saya menggunakan x. Pada fungsi return pengguna akan memberikan nilai dan juga bisa mengakhiri eksekusi dari fungsi tersebut.

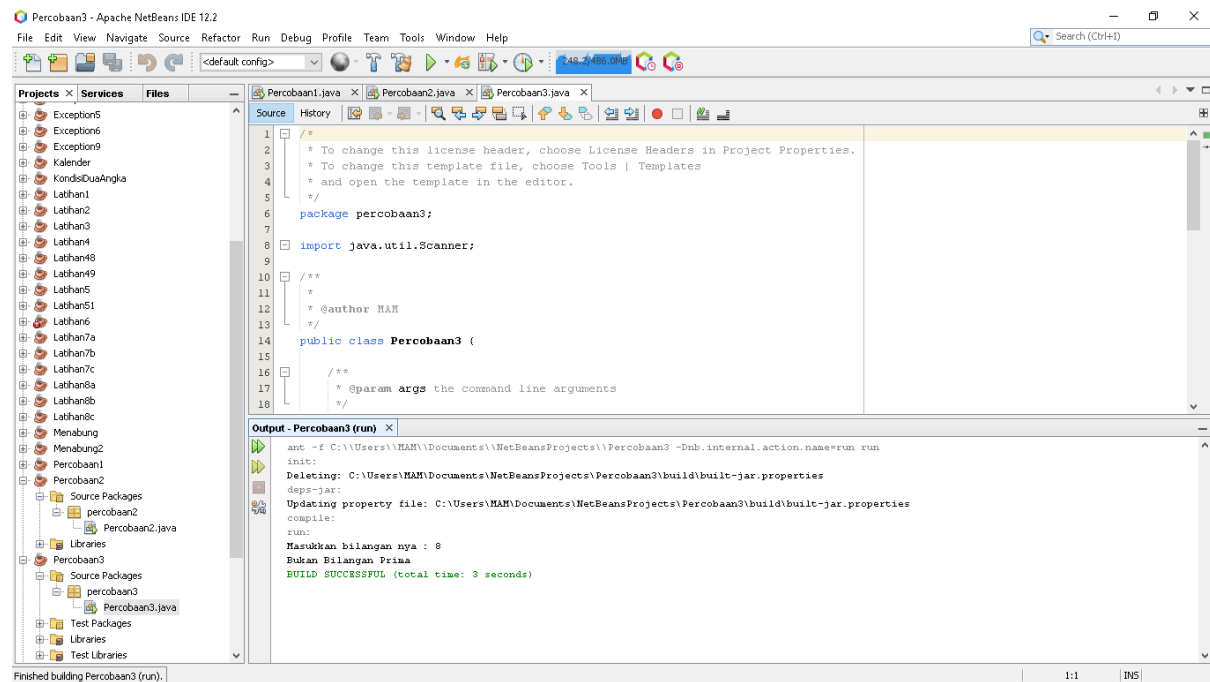
Percobaan 2



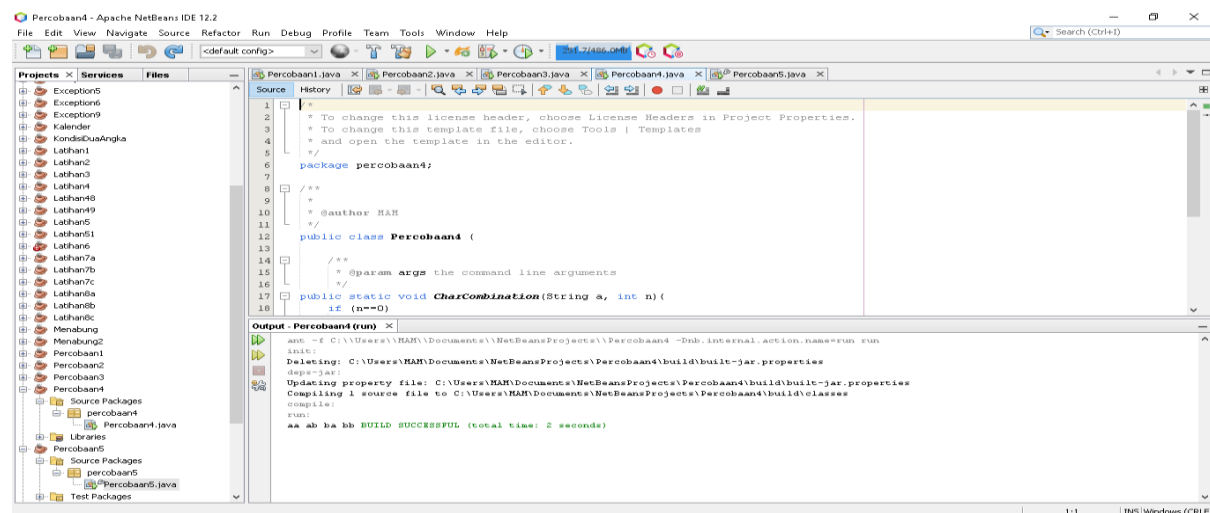
Program di atas hanya mencari suku ke-n dari deret Fibonacci. Bagaimana jika kita ingin mencetak deret Fibonacci mulai suku pertama hingga suku ke-n?

Untuk mencetak deret fibonacci, maka kita harus memasukkan nilai $n = 10$ agar nanti outputnya munculnya mulai dari 0 hingga suku ke-n. Untuk mengirim nilai n mulai dari 0 hingga bil, maka kita menggunakan perulangan for. Dengan perulangan for, fungsi fibo akan dipanggil ketika nilai berubah dan hasil dikirim kembali ke fungsi main dan langsung dicetak.

Percobaan 3

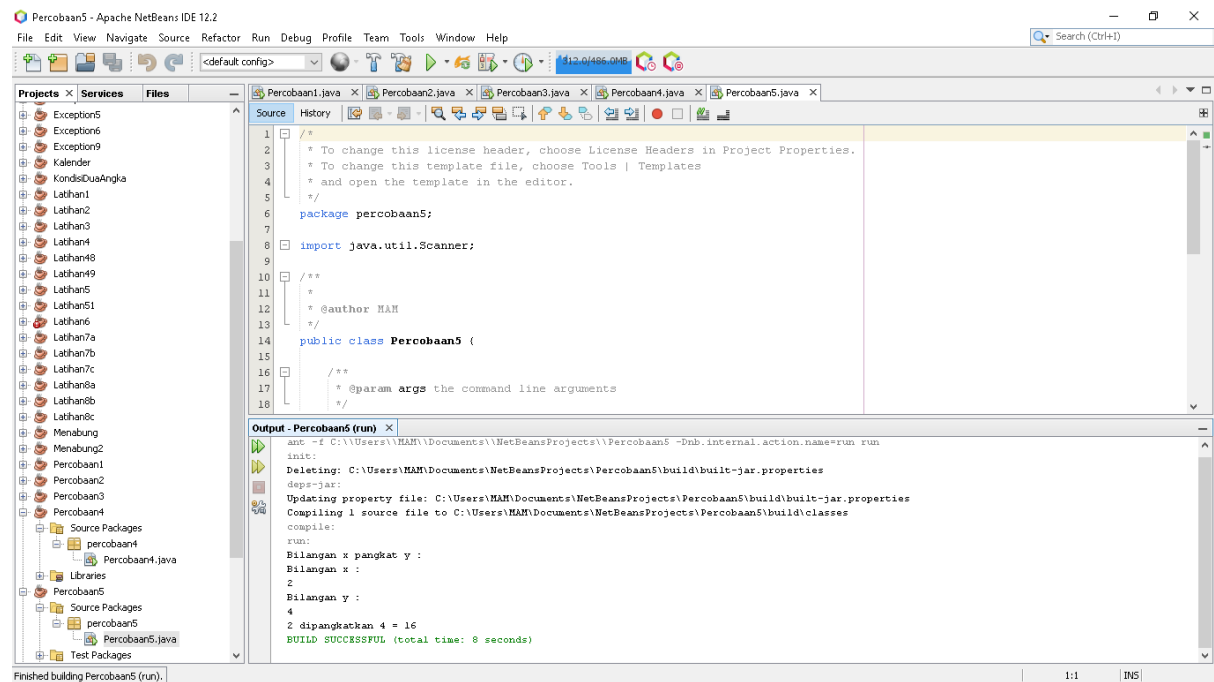


Percobaan 4 :



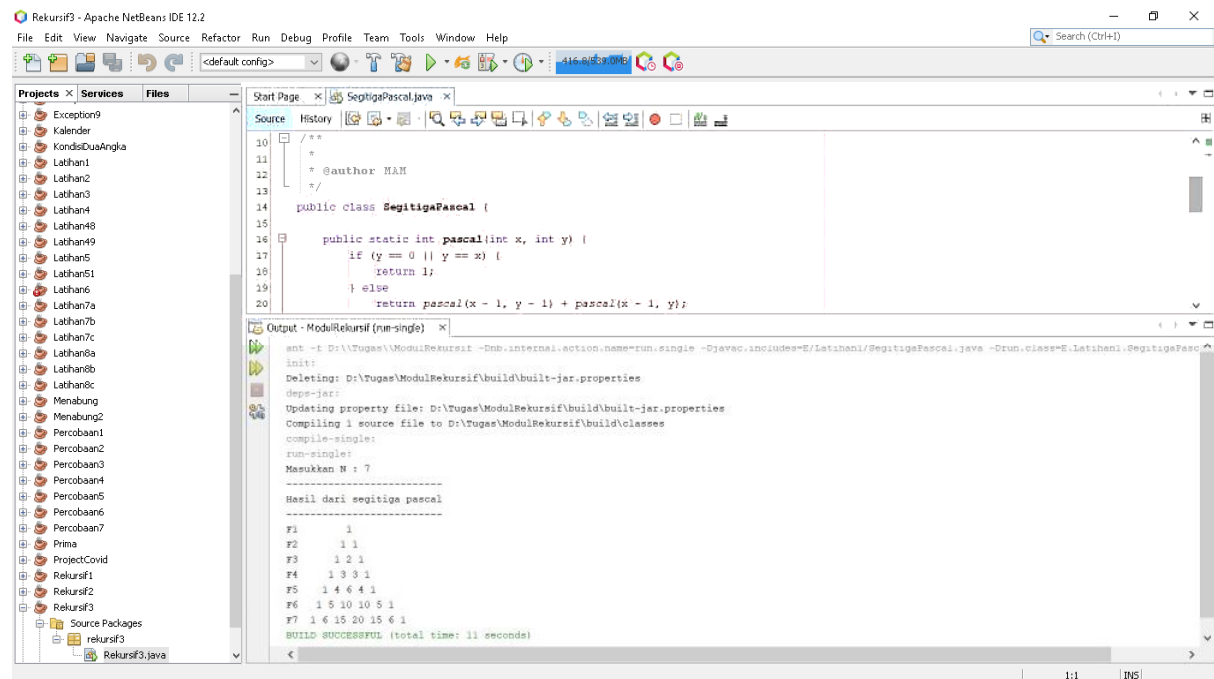
int 'i' disitu menunjukkan angka 97 karena char 'a' dimulai dari 97 dan dibatasi hingga 99 yakni dengan char $(n-1) = 2$. Pada line ke 17 untuk menentukan bagian depan dari sebuah kombinasi, jika bagian awal ditentukan oleh user maka bagian batas juga akan otomatis ditentukan user. Jika index 1 maka akan sampai kombinasi a, jika index 2 maka akan sampai kombinasi b, jika index 3 maka akan sampai kombinasi c, dan seterusnya.

Percobaan 5:

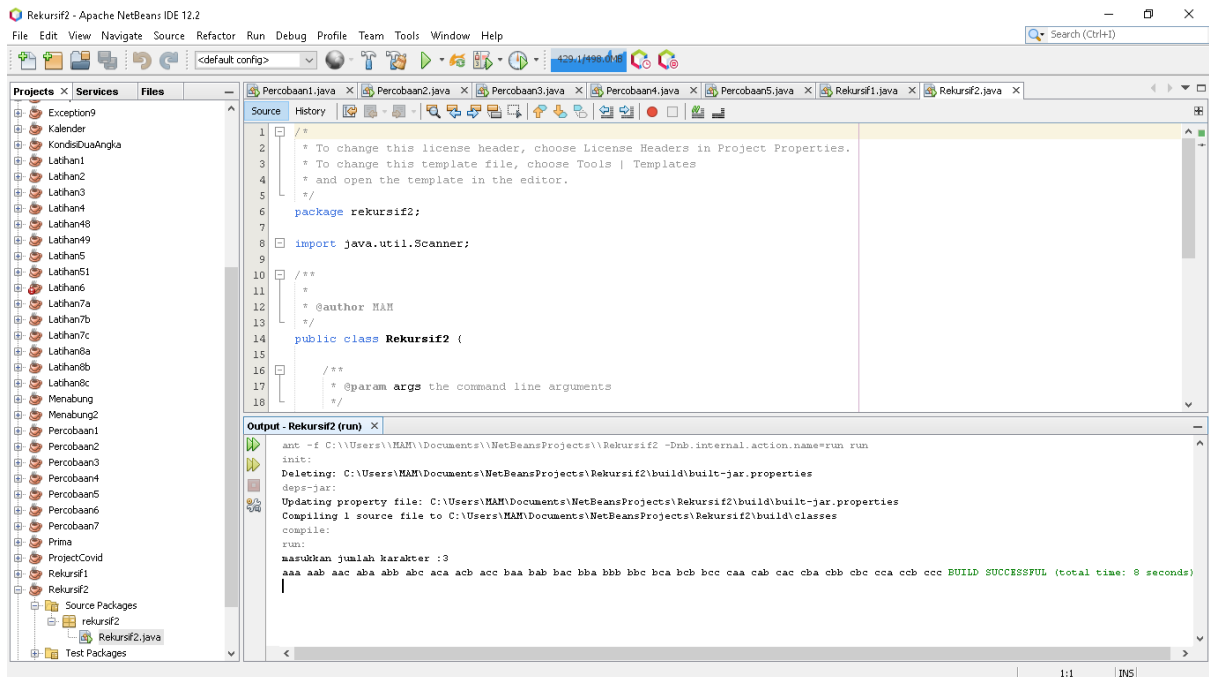


Pada proses ini berfungsi untuk menghitung pangkat. If($y=0$) jika y bernilai 0 maka akan mengeksekusi kondisi yang pertama yaitu return 1 yang mempunyai hasil 1. Kemudian else return $x * \text{pangkat rekursif}(x, y-1)$ berfungsi untuk menghitung pangkatnya. Rekursif disini dimulai pada saat penghenti terjadi Ketika $y=0$, digunakan untuk memecahkan permasalahan secara sederhana.

Latihan 1



Latihan 2

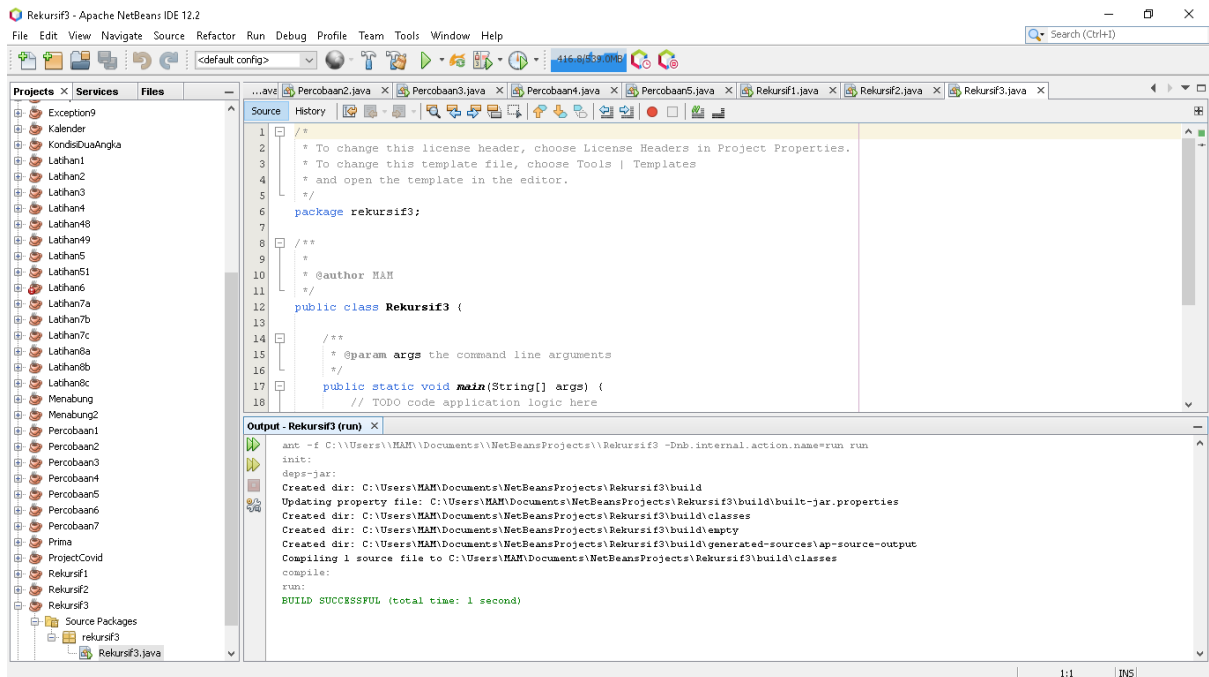


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package rekursif2;
7
8  import java.util.Scanner;
9
10 /**
11  *
12  * @author HAM
13  */
14 public class Rekursif2 {
15
16     /**
17     * @param args the command line arguments
18     */
19
20     public static void main(String[] args) {
21         Scanner scanner = new Scanner(System.in);
22         System.out.println("Masukkan jumlah karakter :");
23         int n = scanner.nextInt();
24         scanner.close();
25         rekursif2(n);
26     }
27
28     public static void rekursif2(int n) {
29         if (n > 0) {
30             rekursif2(n - 1);
31             System.out.print(n + " ");
32         }
33     }
34 }
```

Output - Rekursif2 (run)

```
ant -f C:\Users\HAM\Documents\NetBeansProjects\Rekursif2 -Dnb.internal.action.name=run
init:
Deleting: C:\Users\HAM\Documents\NetBeansProjects\Rekursif2\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Rekursif2\build\build-jar.properties
Compiling 1 source file to C:\Users\HAM\Documents\NetBeansProjects\Rekursif2\build\classes
compile:
run:
masukkan jumlah karakter :3
aaa aab aac aba abb abc acc acb acc baa bab bac bba bbb bbc bca bcb bcc caa cab cac cba cbb cbc cca ccb ccc BUILD SUCCESSFUL (total time: 9 seconds)
```

Latihan 3

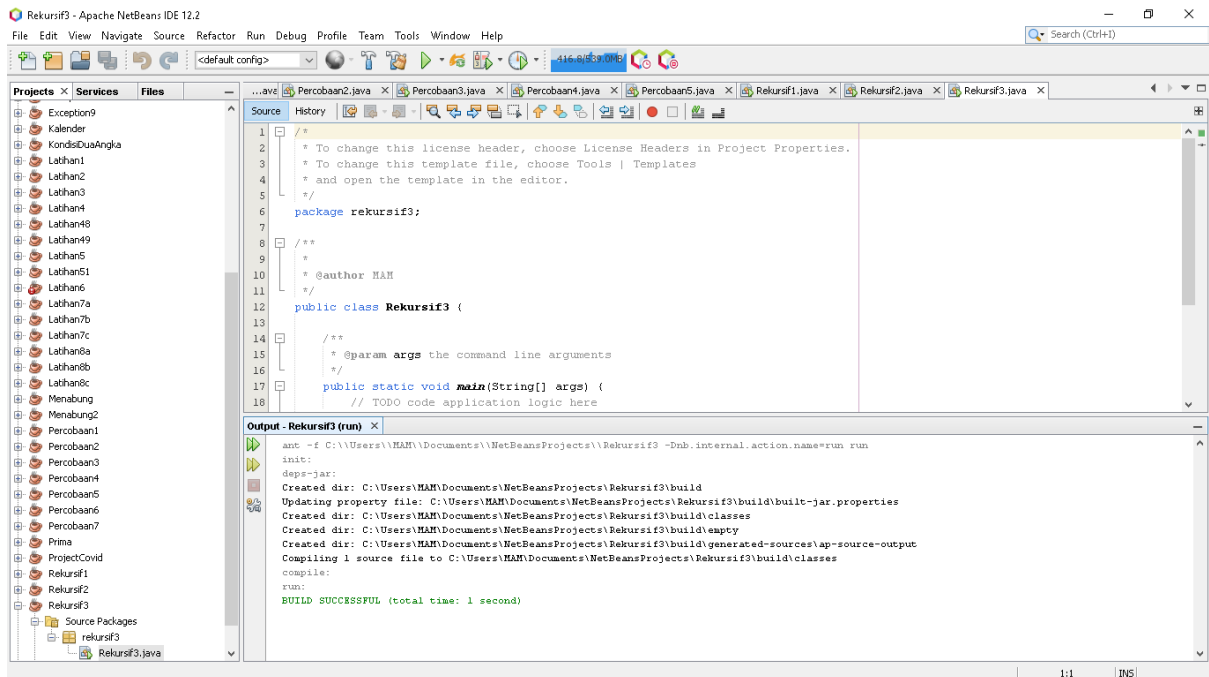


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package rekursif3;
7
8  /**
9  *
10  * @author HAM
11  */
12 public class Rekursif3 {
13
14     /**
15     * @param args the command line arguments
16     */
17
18     public static void main(String[] args) {
19         // TODO code application logic here
20     }
21 }
```

Output - Rekursif3 (run)

```
ant -f C:\Users\HAM\Documents\NetBeansProjects\Rekursif3 -Dnb.internal.action.name=run
init:
deps-jar:
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\build-jar.properties
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\empty
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
compile:
run:
BUILD SUCCESSFUL (total time: 1 second)
```

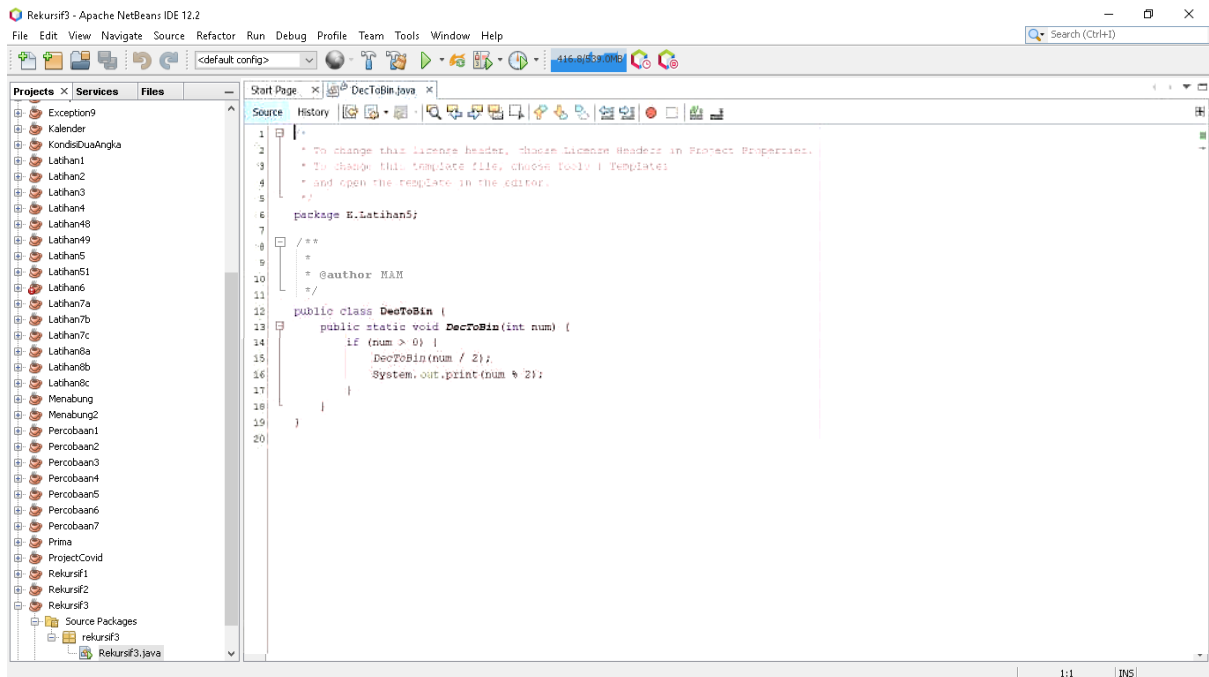
Latihan 4



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package rekursif3;
7
8  /**
9   *
10   * @author HAM
11   */
12  public class Rekursif3 {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20  }
```

```
ant -f C:\Users\HAM\Documents\NetBeansProjects\Rekursif3 -Dnb.internal.action.name=run run
init:
setup-jar:
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\build-jar.properties
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\empty
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
compile:
BUILD SUCCESSFUL (total time: 1 second)
```

Latihan 5:

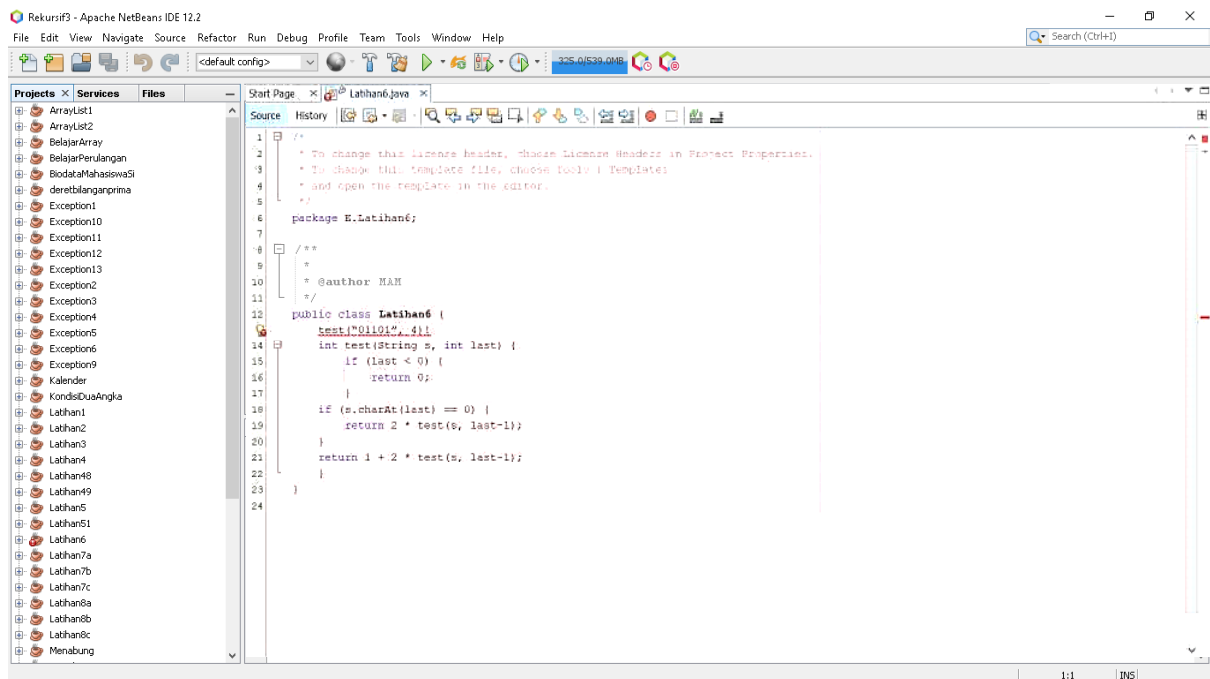


```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package E.Latihan5;
7
8  /**
9   *
10   * @author HAM
11   */
12  public class DecToBin {
13      public static void DecToBin(int num) {
14          if (num > 0) {
15              DecToBin(num / 2);
16              System.out.print(num % 2);
17          }
18      }
19  }
```

```
ant -f C:\Users\HAM\Documents\NetBeansProjects\Rekursif3 -Dnb.internal.action.name=run run
init:
setup-jar:
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build
Updating property file: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\build-jar.properties
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\empty
Created dir: C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\generated-sources\ap-source-output
Compiling 1 source file to C:\Users\HAM\Documents\NetBeansProjects\Rekursif3\build\classes
compile:
BUILD SUCCESSFUL (total time: 1 second)
```

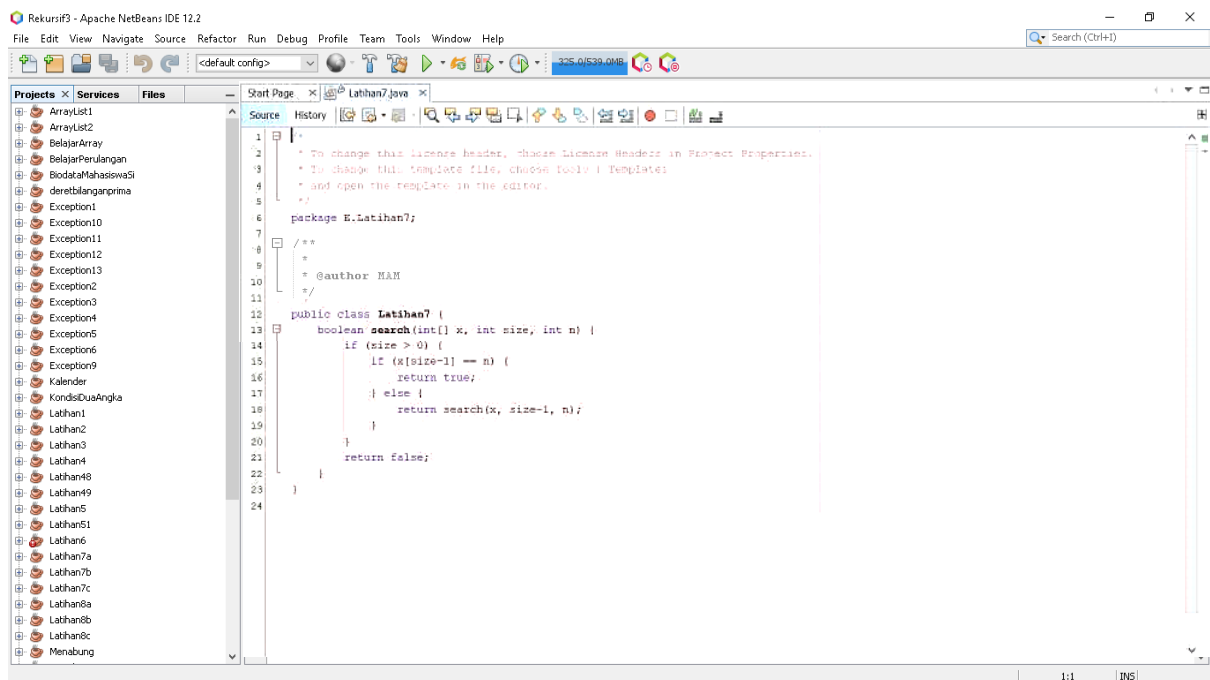
Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialisn tipe data integer dengan variabel num. Selanjutnya dilakukan proses percabangan, apabila variabel num nilainya > 0 maka harus dibagi 2. Setelah itu menampilkan nilai sisa bagi 2 variabel num.

Latihan 6



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package E.Latihan6;
7
8  /**
9   *
10   * @author MAM
11   */
12  public class Latihan6 {
13      int test(String s, int last) {
14          if (last < 0) {
15              return 0;
16          }
17          if (s.charAt(last) == 0) {
18              return 2 * test(s, last-1);
19          }
20          return 1 + 2 * test(s, last-1);
21      }
22  }
23
24  
```

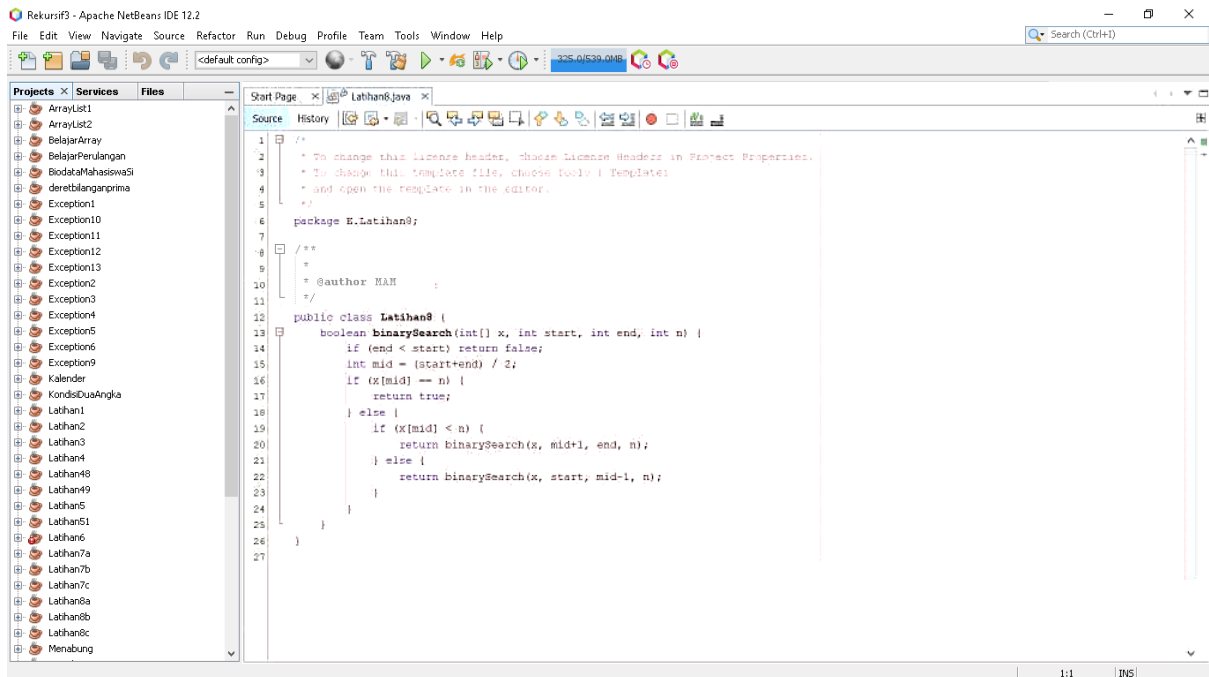
Latihan 7:



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package E.Latihan7;
7
8  /**
9   *
10   * @author MAM
11   */
12  public class Latihan7 {
13      boolean search(int[] x, int size, int n) {
14          if (size > 0) {
15              if (x[size-1] == n) {
16                  return true;
17              } else {
18                  return search(x, size-1, n);
19              }
20          }
21          return false;
22      }
23  }
24  
```

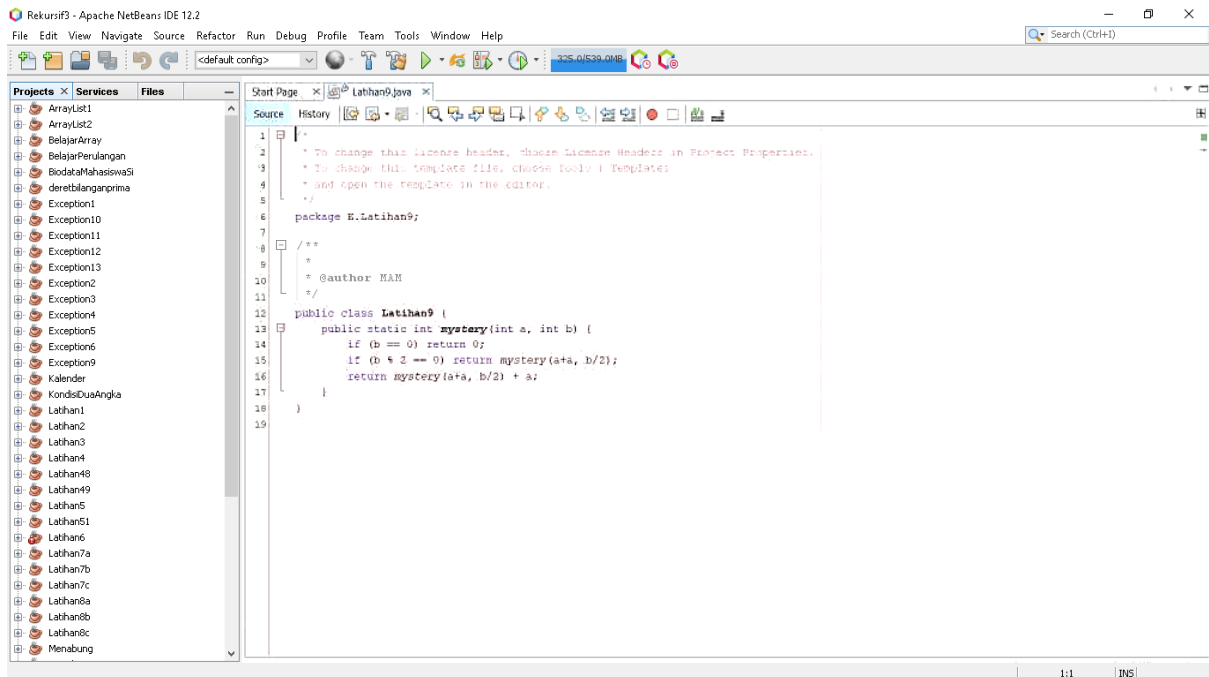
Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialisn tipe data integer dengan variabel x, size dan n. Apabila nilai dari variabel size > 0 maka selanjunya dilakukan proses percabangan dan apabila x dengan size – 1 sama dengan nilai n maka perlu melakukan proses boolean dan jawabannya benar (TRUE). Jika tidak maka melakukan proses dengan menampilkan nilai variabel x, size dan n. Kemudian jika tidak termasuk percabangan pertama maka proses selanjutnya yaitu dengan proses boolean dengan jawaban yang salah (FALSE)

Latihan 8 :



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package E.Latihan8;
7
8  /**
9   *
10   * @author MAM
11   */
12  public class Latihan8 {
13      boolean binarySearch(int[] x, int start, int end, int n) {
14          if (end < start) return false;
15          int mid = (start+end) / 2;
16          if (x[mid] == n) {
17              return true;
18          } else {
19              if (x[mid] < n) {
20                  return binarySearch(x, mid+1, end, n);
21              } else {
22                  return binarySearch(x, start, mid-1, n);
23              }
24          }
25      }
26  }
27  }
```

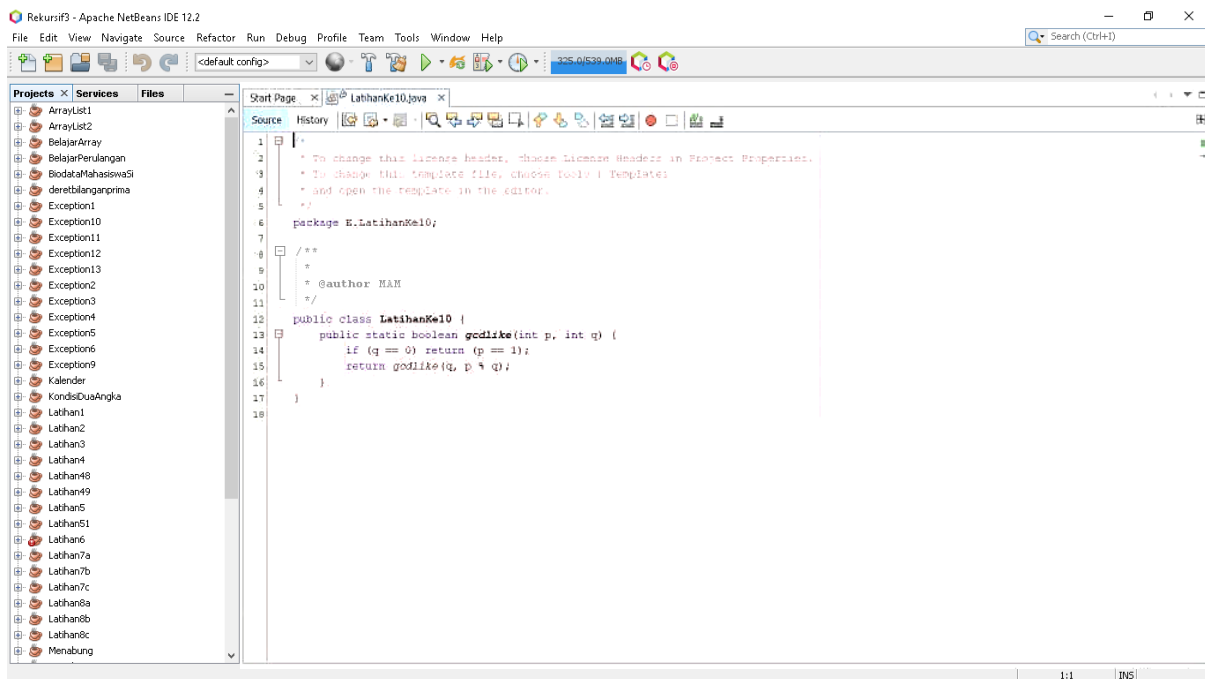
Latihan 9:



```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package E.Latihan9;
7
8  /**
9   *
10   * @author MAM
11   */
12  public class Latihan9 {
13      public static int mystery(int a, int b) {
14          if (b == 0) return 0;
15          if (b % 2 == 0) return mystery(a+a, b/2);
16          return mystery(a+a, b/2) + a;
17      }
18  }
19  }
```

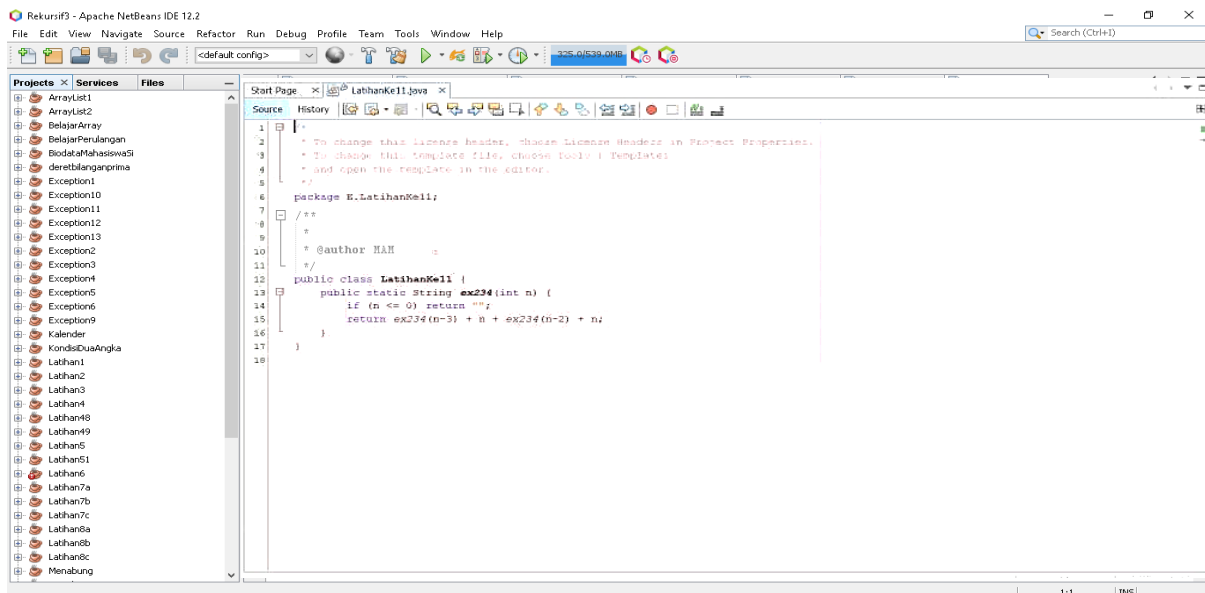
Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data integer dengan variabel mystery yang nantinya tipe publicnya dapat dipanggil, serta variabel a, b. Melakukan percabangan apabila nila variabel $b = 0$, sebaliknya apabila variabel b sisa bagi 2 = 0 maka akan memanggil variabel mystery diikuti dengan penjumlahan pada varibel a + a dan variabel b di bagi 2. Jika tidak mau dilakkan uji coba pada percabangan maka langsung lakukan proses dengan memanggil variabel mystery diikuti dengan penjumlahan variabel a + a dan variabel b dibagi 2 kemudian dijumlahkan dengan variabel a.

Latihan 10 :



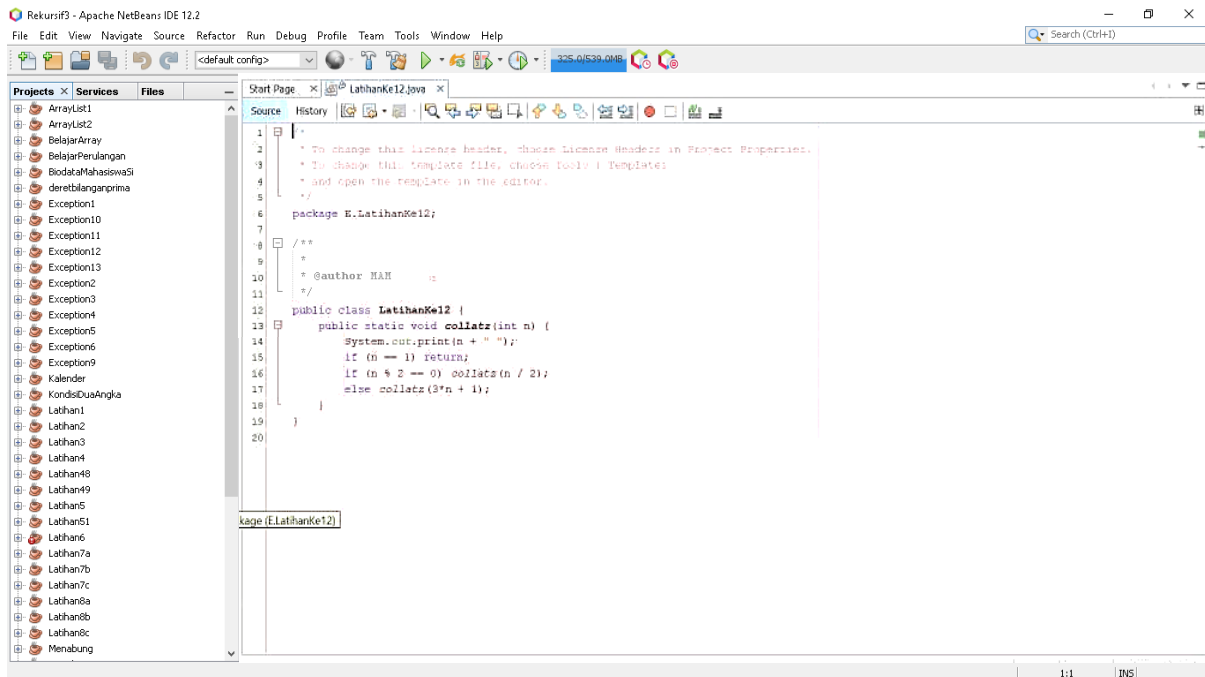
Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data boolean dengan variabel gcdlike yang nantinya tipe publicnya dapat dipanggil, serta variabel p dan q. Proses selanjutnya dengan melakukan percabangan apabila nilai variabel q = 0 maka p nilainya 1. Jika tidak perlu melakukan percabangan maka menggunakan proses boolean variabel gcdlike, serta variabel p dan q, dan nilai sisa bagi p dan q.

Latihan 11:



Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data string dengan variabel ex234, serta variabel n dengan tipe data integer. Melakukan proses percabangan apabila nilai variabel n kurang dari atau sama dengan 0 maka proses akan selesai atau diakhiri. Jika tidak maka akan melakukan proses pemanggilan variabel ex234 kemudian dengan menambahkan rumus sesuai kode program tersebut.

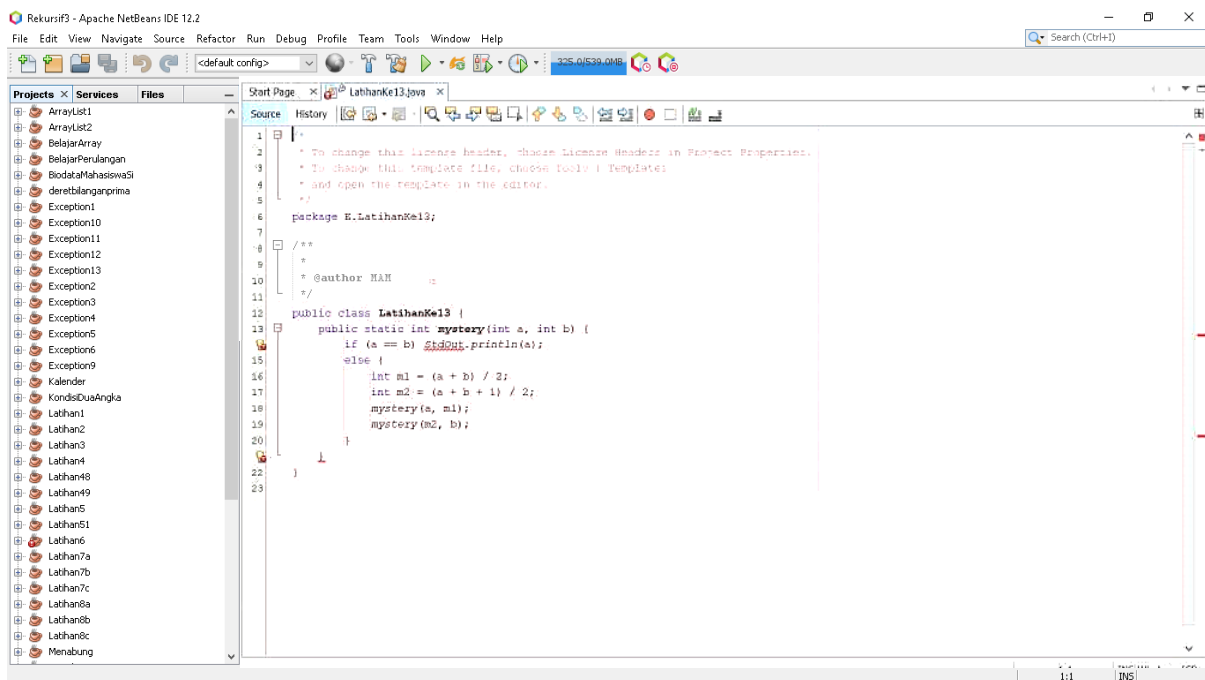
Latihan 12:



```
1 2
3  * To change this license header, choose License Headers in Project Properties.
4  * To change this template file, choose Tools | Templates
5  * and open the template in the editor.
6  */
7
8  package E.LatihanKe12;
9
10 /**
11  *
12  * @author HAM
13  */
14 public class LatihanKe12 {
15     public static void collatz(int n) {
16         System.out.print(n + " ");
17         if (n == 1) return;
18         if (n % 2 == 0) collatz(n / 2);
19         else collatz(3 * n + 1);
20     }
21 }
```

Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data integer dengan variabel n yang nantinya tipe public static void bernama collatz digunakan pada saat pemanggilannya. Melakukan proses percabangan untuk menampilkan variabel n, apabila nilainya 1 maka program tidak bisa merespon prosesnya. Jika sisa bagi 2 nilai variabel $n = 0$ maka akan dilakukan pemanggilan pada tipe public static void dan melakukan proses menghitung nilai variabel n dibagi 2. Jika tidak maka sama aja akan memanggilnya dan melakukan proses dengan dikalikan 3 pada variabel n kemudian di tambah 1.

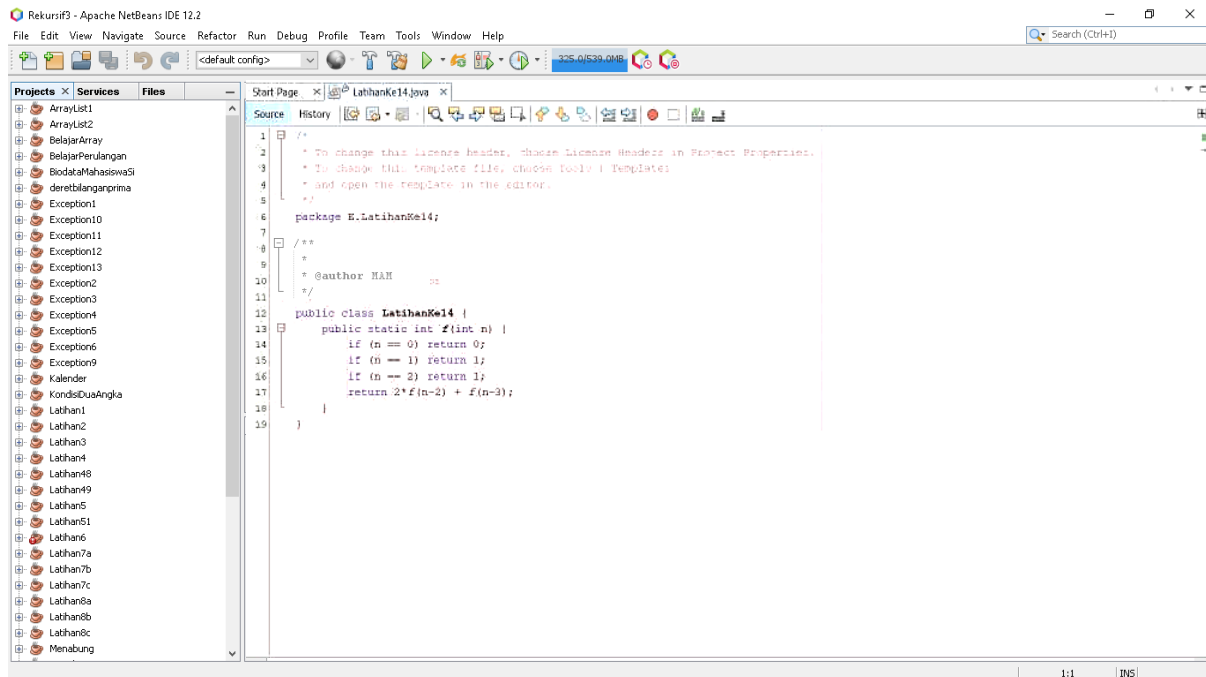
Latihan 13



```
1 2
3  * To change this license header, choose License Headers in Project Properties.
4  * To change this template file, choose Tools | Templates
5  * and open the template in the editor.
6  */
7
8  package E.LatihanKe13;
9
10 /**
11  *
12  * @author HAM
13  */
14 public class LatihanKe13 {
15     public static int mystery(int a, int b) {
16         if (a == b) System.out.println(a);
17         else {
18             int m1 = (a + b) / 2;
19             int m2 = (a + b + 1) / 2;
20             mystery(a, m1);
21             mystery(m2, b);
22         }
23     }
24 }
```

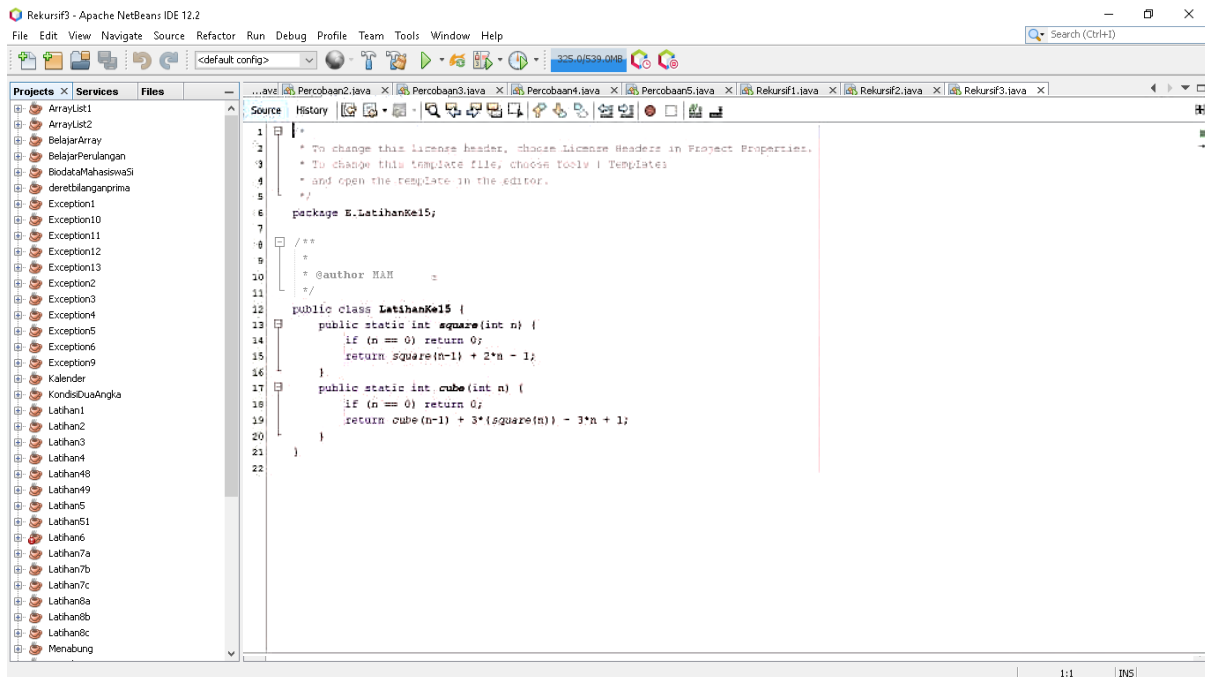
Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data integer dengan variabel mystery, serta variabel a dan b. Melakukan proses percabangan apabila variabel $a = b$ maka akan menampilkan nilai a. Jika tidak maka akan melakukan proses dengan inialisasi m1 dan m2 b dan masing-masing memiliki rumus yang berbeda, kemudian memanggil variabel mystery dan menampilkan variabel a, variabel m1 dan variabel m2, variabel b.

Latihan 14:



Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data integer dengan variabel f dan variabel n dengan tipe data n. Melakukan proses percabangan apabila nilai variabel $n = 0$ maka langsung selesai. Jika variabel $n = 1$ atau 2 bahkan lebih maka melakukan proses return 1 dan jika tidak maka menghitung dengan rumus yang ada.

Latihan 15:



Pada program Rekursif diatas menjelaskan sebuah proses dengan menginisialkan tipe data integer dengan variabel square yang nantinya tipe publicnya dapat dipanggil, serta variabel n. Melakukan percabangan apabila nilai variabel $n = 0$ maka langsung selesai. Jika tidak maka melakukan proses pemanggilan variabel square. Selanjutnya menginisialisasikan variabel cube serta variabel n juga. Melakukan percabangan apabila nilai variabel $n = 0$ maka langsung selesai. Jika tidak maka melakukan proses pemanggilan pada variabel cube.