

BAHASA PEMROGRAMAN JAVA

Pertemuan XI

Kelas Utilitas

Oleh
Achmad Arrosyidi



TUJUAN PEMBELAJARAN

Mahasiswa dapat membedakan penggunaan Kelas Utilitas pada Pemrograman (Math, Stack & Queue).



MATERI

Kelas Utilitas:

A. Kelas Math

1. Absolut
2. Pangkat
3. Pembulatan (ke bawah, ke atas, terdekat)
4. Sinus, Cosinus, Tangen
5. Phi
6. Random

B. Kelas Random

C. Kelas Stack

D. Kelas Queue

E. Kelas Date



A. KELAS - MATH

- Operator tambah, kurang, kali, dan bagi sudah merupakan syarat wajib sebuah bahasa pemrograman.
- Cukup dengan menggunakan keempat operator tersebut, hampir semua perhitungan matematika dapat diselesaikan.
- Tapi bagaimana jika butuh untuk membuat method sinus atau logaritma sendiri?

A. KELAS - MATH

- Java menyediakan class untuk menyelesaikan operasi-operasi matematika dasar lainnya.
- Sinus, cosinus, tangen, absolut, log, dan sebagainya sudah tersedia dan kita cukup menggunakannya, sehingga kita tidak perlu mendownload library-nya terlebih dahulu.



A.1. KELAS – MATH ABSOLUT

- Perhitungan nilai absolute:
- Jika angka bernilai positif, maka angka tersebut langsung ditampilkan.
- Namun jika angka bernilai negatif, maka dikalikan -1 (minus satu)
- Sehingga, jika membuat algoritma sendiri dari fungsi absolut dapat diterapkan dalam kode program sebagai berikut.



A.1. CONTOH MEMBUAT SENDIRI – KELAS ABSOLUT

```
1 //File: LatihanMathAbsolute
2 import java.io.*;
3 public class LatihanMathAbsolute {
4     public static int absolute(int angka) {
5         if (angka < 0) {
6             angka *= -1;
7         }
8         return angka;
9     }
10
11     public static void main (String[] arg) {
12         try {
13             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
14             System.out.print("Masukkan sebuah angka! ");
15             int a = Integer.parseInt(br.readLine());
16             int hasil = absolute(a);
17             System.out.println("Hasil nilai absolut dari angka yang anda masukkan adalah: " + hasil);
18         }
19         catch (Exception e){
20             System.out.println("Input-an harus berupa angka!");
21         }
22     }
23 }
```

Console

Masukkan sebuah angka! 2
Hasil nilai absolut dari angka yang anda masukkan adalah: 2
<<< Process finished (PID=10100). (Exit code 0)
===== READY =====

Console

Masukkan sebuah angka! -2
Hasil nilai absolut dari angka yang anda masukkan adalah: 2
<<< Process finished (PID=20396). (Exit code 0)
===== READY =====

A.1. KELAS – **MATH ABSOLUT**

- Kita tidak perlu bersusah payah membuatnya, termasuk untuk berbagai tipe data lainnya.
- Karena Java telah menyediakan fungsi tersebut dalam class Math-nya.
- Penggunaannya cukup dengan memanggil method static-nya bernama “abs” seperti berikut ini.



A.1. CONTOH MEMANGGIL - KELAS MATH ABSOLUT

```
1 //File: LatihanMathAbsolute2
2 import java.io.*;
3 public class LatihanMathAbsolute2 {
4     public static void main (String[] arg) {
5         try {
6             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7             System.out.print("Masukkan sebuah angka! ");
8             int a = Integer.parseInt(br.readLine());
9             System.out.println("Hasil nilai absolut dari angka yang anda masukkan adalah: " + Math.abs(a));
10        }
11        catch (Exception e){
12            System.out.println("Input-an harus berupa angka!");
13        }
14    }
15 }
```

Console

```
Masukkan sebuah angka! 2
Hasil nilai absolut dari angka yang anda masukkan adalah: 2
<<< Process finished (PID=9252). (Exit code 0)
===== READY =====
```

Console

```
Masukkan sebuah angka! -2
Hasil nilai absolut dari angka yang anda masukkan adalah: 2
<<< Process finished (PID=20084). (Exit code 0)
===== READY =====
```



A.2. KELAS – MATH PANGKAT

- Menghitung a pangkat b (a^b)
- Penghitungan a pangkat b memerlukan algoritma yang sedikit lebih rumit.
- Ada 3 kondisi untuk penghitungannya, yaitu apabila pangkatnya (b) lebih kecil dari nol, sama dengan nol, atau lebih besar dari nol.
- Penghitungan untuk pangkat lebih besar dari nol dapat dilakukan dengan algoritma berikut ini.



A.2. CONTOH MEMBUAT SENDIRI – KELAS PANGKAT

```
1 //File: LatihanMathPangkat.java
2 import java.io.*;
3 public class LatihanMathPangkat {
4     public static int pangkat(int angka, int pangkat) {
5         int hasil =1;
6         if (pangkat > 0) {
7             for (int i=0; i<pangkat; i++) {
8                 hasil *= angka;
9             }
10        }
11        return hasil;
12    }
13
14    public static void main (String[] arg) {
15        try {
16            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
17            System.out.print("Masukkan sebuah angka yang akan dipangkatkan! ");
18            int a = Integer.parseInt(br.readLine());
19            System.out.print("Masukkan sebuah angka sebagai pemangkat! ");
20            int b = Integer.parseInt(br.readLine());
21            int hasil = pangkat(a,b);
22            System.out.println("Hasil " + a + " pangkat " + b + " adalah: " + hasil);
23        }
24
25        catch (Exception e){
26            System.out.println("Input-an harus berupa angka!");
27        }
28    }
29 }
30
```

Console

Masukkan sebuah angka yang akan dipangkatkan! 2

Masukkan sebuah angka sebagai pemangkat! 3

Hasil 2 pangkat 3 adalah: 8

<<< Process finished (PID=17620). (Exit code 0)

===== READY =====



A.2. CONTOH MEMANGGIL - KELAS MATH PANGKAT

```
1 //File: LatihanMathPangkat2.java
2 import java.io.*;
3 public class LatihanMathPangkat2 {
4     public static void main (String[] arg) {
5         try {
6             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7             System.out.print("Masukkan sebuah angka yang akan dipangkatkan! ");
8             int a = Integer.parseInt(br.readLine());
9             System.out.print("Masukkan sebuah angka sebagai pemangkat! ");
10            int b = Integer.parseInt(br.readLine());
11            System.out.println("Hasil " + a + " pangkat " + b + " adalah: " + Math.pow(a,b));
12        }
13
14        catch (Exception e){
15            System.out.println("Input-an harus berupa angka!");
16        }
17    }
18 }
```

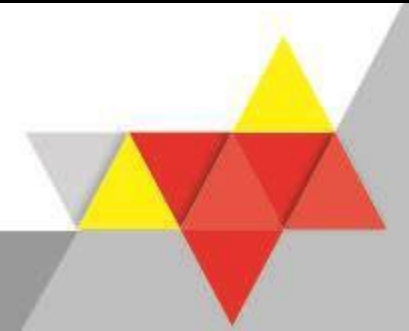
Console

Masukkan sebuah angka sebagai pemangkat! 3

Hasil 2 pangkat 3 adalah: 8.0

<<< Process finished (PID=17856). (Exit code 0)

===== READY =====



A.3. KELAS – MATH PEMBULATAN

- Algoritma pembulatan bilangan desimal juga cukup rumit jika kita dibuat sendiri.
- Pembulatan pun ada tiga macam yaitu:
 1. pembulatan ke atas (ceiling)
 2. pembulatan ke bawah (floor)
 3. pembulatan ke bilangan bulat terdekat.



A.3. KELAS – **MATH PEMBULATAN**

- Java telah menyediakan untuk:
 1. Pembulatan ke atas dengan method `Math.ceil()`
 2. Pembulatan ke bawah dengan method `Math.floor()`
 3. Pembulatan ke bilangan bulat terdekat dengan method `Math.round()`



A.3. CONTOH KELAS – MATH PEMBULATAN

```
1 //File: LatihanMathPembulatan.java
2 public class LatihanMathPembulatan {
3
4     public static void main (String[] args) {
5         double a = 3.71;
6         double b = 3.41;
7
8         System.out.println("a = "+a);
9         System.out.println("b = "+b);
10
11         System.out.println("Pembulatan ke atas a = "+ Math.ceil(a));
12         System.out.println("Pembulatan ke atas b = "+ Math.ceil(b));
13         System.out.println("Pembulatan ke bawah a = "+ Math.floor(a));
14         System.out.println("Pembulatan ke bawah b = "+ Math.floor(b));
15         System.out.println("Pembulatan ke terdekat a = "+ Math.round(a));
16         System.out.println("Pembulatan ke terdekat b = "+ Math.round(b));
17     }
18 }
```

Console

```
a = 3.71
b = 3.41
Pembulatan ke atas a = 4.0
Pembulatan ke atas b = 4.0
Pembulatan ke bawah a = 3.0
Pembulatan ke bawah b = 3.0
Pembulatan ke terdekat a = 4
Pembulatan ke terdekat b = 3
<<< Process finished (PID=12544). (Exit code 0)
===== READY =====
```


A.3. CONTOH KELAS – MATH PEMBULATAN

```
1 //File: LatihanMathPembulatan2.java
2 import java.io.*;
3 public class LatihanMathPembulatan2 {
4     public static void main (String[] args) {
5         try {
6             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7
8             System.out.print("Masukkan sebuah angka pertama (3.71) : ");
9             Double a=Double.parseDouble(br.readLine());
10            System.out.print("Masukkan sebuah angka kedua (3.41) : ");
11            Double b=Double.parseDouble(br.readLine());
12
13            System.out.println("Pembulatan ke atas a = "+ Math.ceil(a));
14            System.out.println("Pembulatan ke atas b = "+ Math.ceil(b));
15            System.out.println("Pembulatan ke bawah a = "+ Math.floor(a));
16            System.out.println("Pembulatan ke bawah b = "+ Math.floor(b));
17            System.out.println("Pembulatan ke terdekat a = "+ Math.round(a));
18            System.out.println("Pembulatan ke terdekat b = "+ Math.round(b));
19        }
20        catch(Exception e) {
21            System.out.println("Inputan harus berupa angka!!!");
22        }
23    }
24 }
```

Console

```
Masukkan sebuah angka pertama (3.71) : 3.71
Masukkan sebuah angka kedua (3.41) : 3.41
Pembulatan ke atas a = 4.0
Pembulatan ke atas b = 4.0
Pembulatan ke bawah a = 3.0
Pembulatan ke bawah b = 3.0
Pembulatan ke terdekat a = 4
Pembulatan ke terdekat b = 3
<<< Process finished (PID=14384). (Exit code 0)
===== READY =====
```



A.4. KELAS – MATH SINUS, COSINUS, DAN TANGEN

- Perhitungan sinus, cosinus, dan tangen dari sebuah sudut juga cukup rumit jika dibuat algoritmanya.
- Class Math menyediakan:
 1. perhitungan sinus dengan method `Math.sin()`
 2. perhitungan cosinus dengan method `Math.cos()`
 3. perhitungan tange dengan method `Math.tan()`
- Ketiga method tersebut menerima sebuah parameter, yaitu sudut dalam bentuk radian dengan tipe data double.
- Untuk mendapatkan nilai radian dari sebuah nilai derajat dari sudut, kita bisa menggunakan method `Math.toRadians()` dengan memasukkan derajat sudut sebagai parameter, seperti contoh berikut.

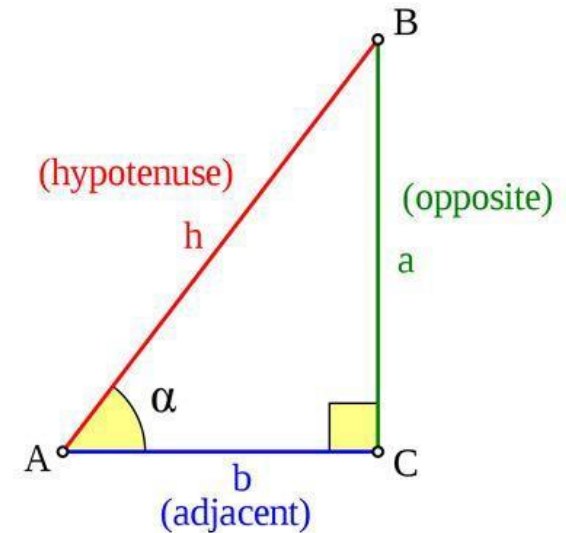
A.4. CONTOH KELAS – MATH SINUS, COSINUS, DAN TANGEN

```
1 //File: LatihanMathSinCosTangen.java
2 import java.io.*;
3
4 public class LatihanMathSinCosTangen {
5
6     public static void main (String[] args) {
7         try {
8             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
9
10            System.out.print("Masukkan sebuah angka sebagai besaran sudut pertama (30): ");
11            Double a=Double.parseDouble(br.readLine());
12            System.out.print("Masukkan sebuah angka sebagai besaran sudut kedua (40) : ");
13            Double b=Double.parseDouble(br.readLine());
14
15            System.out.println("Sin " + a + " = " + Math.sin(Math.toRadians(a)));
16            System.out.println("Cos " + a + " = " + Math.cos(Math.toRadians(a)));
17            System.out.println("Tan " + a + " = " + Math.tan(Math.toRadians(a)));
18            System.out.println("Sin " + b + " = " + Math.sin(Math.toRadians(b)));
19            System.out.println("Cos " + b + " = " + Math.cos(Math.toRadians(b)));
20            System.out.println("Tan " + b + " = " + Math.tan(Math.toRadians(b)));
21        }
22
23        catch(Exception e) {
24            System.out.println("Inputan harus berupa angka!!!");
25        }
26    }
27 }
```

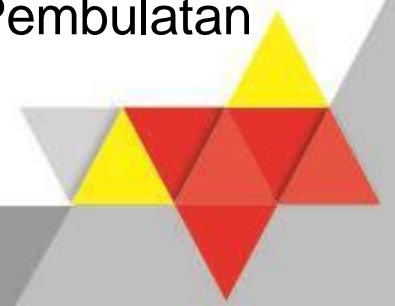
A.4. CONTOH KELAS – MATH SINUS, COSINUS, DAN TANGEN

Console

```
Masukkan sebuah angka sebagai besaran sudut pertama (30): 30
Masukkan sebuah angka sebagai besaran sudut kedua (40) : 40
Sin 30.0 = 0.49999999999999994
Cos 30.0 = 0.8660254037844387
Tan 30.0 = 0.5773502691896257
Sin 40.0 = 0.6427876096865393
Cos 40.0 = 0.766044443118978
Tan 40.0 = 0.8390996311772799
<<< Process finished (PID=13040). (Exit code 0)
===== READY =====
```

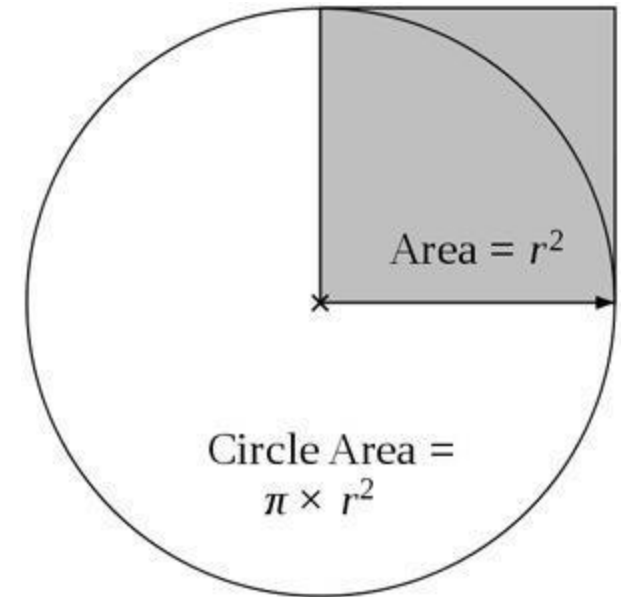


- Pada contoh di atas, kita tidak akan menemukan hasil nilai bulat $\frac{1}{2}$ dari $\sin 30$, karena tingkat presisi dari komputer cukup sulit untuk merepresentasikan bilangan desimal.
- Jika memerlukan pembulatan maka tinggal menggunakan kelas Math Pembulatan seperti pada contoh sebelumnya.



A.5. KELAS – MATH KONSTANTA PHI

- Perhitungan luas lingkaran membutuhkan nilai Phi, yaitu sebuah konstanta dengan nilai berkisar 3.14159.
- Angka dibelakang koma dari nilai ini sebenarnya masih sangat panjang.
- Class Math menyediakan konstanta ini untuk digunakan.
- Sebagai contoh, penghitungan luas lingkaran sesuai rumus standarnya menjadi sebagai berikut.



A.5. CONTOH KELAS – MATH KONSTANTA PHI

```
1 //File: LatihanMathPhi.java
2 import java.io.*;
3 public class LatihanMathPhi {
4     public static void main (String[] args) {
5         try {
6             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
7
8             System.out.print("Masukkan jari-jari pada sebuah lingkaran yang akan dihitung luasnya: ");
9             int r=Integer.parseInt(br.readLine());
10
11             System.out.println("Luas lingkaran dengan jari-jari " + r + " adalah: " +(Math.PI * Math.pow(r, 2)));
12         }
13
14         catch(Exception e) {
15             System.out.println("Inputan harus berupa angka!!!");
16         }
17     }
18 }
```

Console

```
CD: C:\java 19.2\M9
Current directory: C:\java 19.2\M9
java LatihanMathPhi
Process started (PID=18440) >>>
Masukkan jari-jari pada sebuah lingkaran yang akan dihitung luasnya: 5
Luas lingkaran dengan jari-jari 5 adalah: 78.53981633974483
<<< Process finished (PID=18440). (Exit code 0)
===== READY =====
```



A.6. KELAS – **MATH RANDOM**

- Class Math ini juga menyediakan method untuk membuat angka acak.
- Bedanya dengan cara mengacak biasa, class ini menghasilkan sebuah nilai desimal antara 0 sampai 1.
- Dari hasil pengacakan tersebut, kita bisa mengkombinasikan dengan method lain untuk menghasilkan bilangan acak dalam jangka nilai tertentu.
- Misalkan, jika membutuhkan bilangan acak antara 0-100 sebanyak 10 kali, maka kita dapat membuat program sebagai berikut.

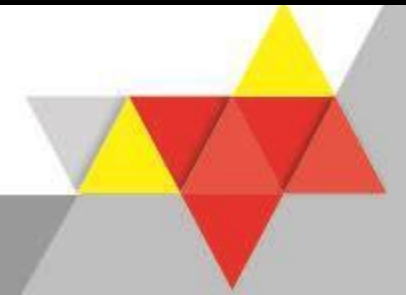


A.6. CONTOH KELAS – MATH RANDOM

```
1 //File: LatihanMathRandom.java
2
3 public class LatihanMathRandom {
4     public static void main(String[] args) {
5         double random;
6
7         // acak antara 0-100 sebanyak 10 kali
8         for(int i = 0; i < 10; i++) {
9             random = Math.random();
10            System.out.println("Angka acak 0-100 Ke-" + i + " = " + Math.floor((100 * random)));
11        }
12    }
13 }
```

Console

```
Angka acak 0-100 Ke-0 = 84.0
Angka acak 0-100 Ke-1 = 97.0
Angka acak 0-100 Ke-2 = 24.0
Angka acak 0-100 Ke-3 = 43.0
Angka acak 0-100 Ke-4 = 24.0
Angka acak 0-100 Ke-5 = 10.0
Angka acak 0-100 Ke-6 = 77.0
Angka acak 0-100 Ke-7 = 79.0
Angka acak 0-100 Ke-8 = 62.0
Angka acak 0-100 Ke-9 = 75.0
<<< Process finished (PID=7928). (Exit code 0)
===== READY =====
```



B. KELAS – **RANDOM**

- Kelas Random terdapat pada paket Java.util.
- Kelas ini merupakan kelas yang memungkinkan membangkitkan (men-generate) bilangan acak.
- Berbeda dengan fungsi random pada kelas Math, kelas Random memungkinkan memberikan biji yang digunakan untuk menentukan bilangan acak yang bersifat statis.
- Dalam artian hasilnya akan berupa nilai yang sama setiap kali program dijalankan.



B. CONTOH KELAS – RANDOM

```

1 //file: ContohRandom.java
2 import java.util.Random;
3 public class ContohRandom {
4     public static void main (String[] args){
5         byte acak[]=new byte[5];
6
7         //Membangkitkan bilangan acak menurut waktu (Dinamis)
8         Random rnd1 = new Random();
9         rnd1.nextBytes(acak);
10        System.out.println("Random menurut Waktu: ");
11        for (int i=0; i<acak.length; i++) {
12            System.out.println((i+1)+". "+acak[i]);
13        }
14
15        //Membangkitkan bilangan acak menurut nilai (Statis)
16        Random rnd2 = new Random(5);
17        rnd2.nextBytes(acak);
18        System.out.println("Random dengan nilai: ");
19        for (int i=0; i<acak.length; i++){
20            System.out.println((i+1)+". "+acak[i]);
21        }
22    }
23 }
24

```

Console

Random menurut Waktu:

```

1.24
2.-125
3.117
4.-100
5.-3

```

Random dengan nilai:

```

1.-65
2.89
3.3
4.-69
5.88

```

<<< Process finished (PID=15912). (Exit code 0)
===== READY =====

• Running #1

Console

Random menurut Waktu:

```

1.-76
2.-1
3.90
4.37
5.-26

```

Random dengan nilai:

```

1.-65
2.89
3.3
4.-69
5.88

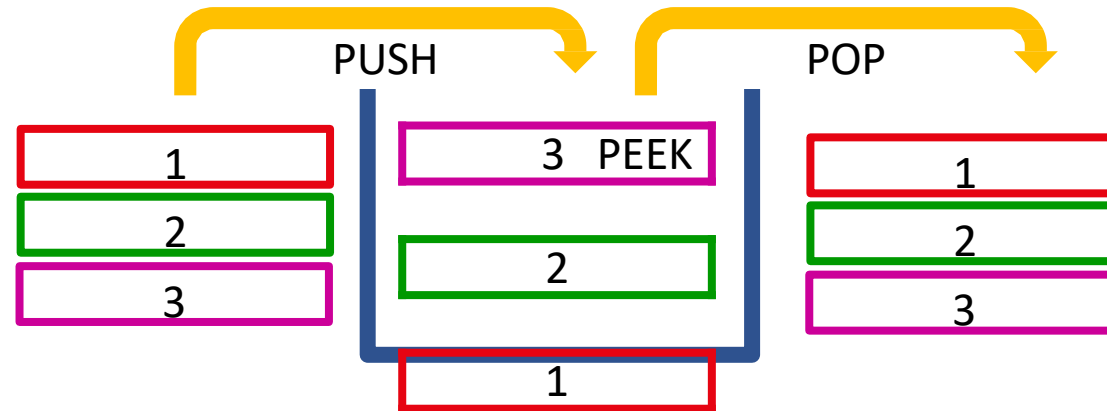
```

<<< Process finished (PID=2924). (Exit code 0)
===== READY =====

• Running #2



C. KELAS – **STACK**

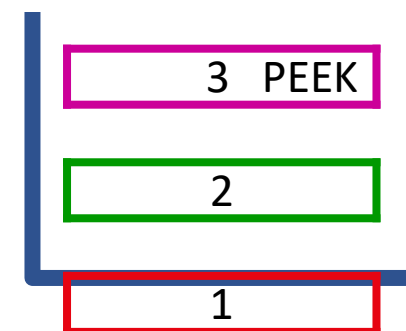
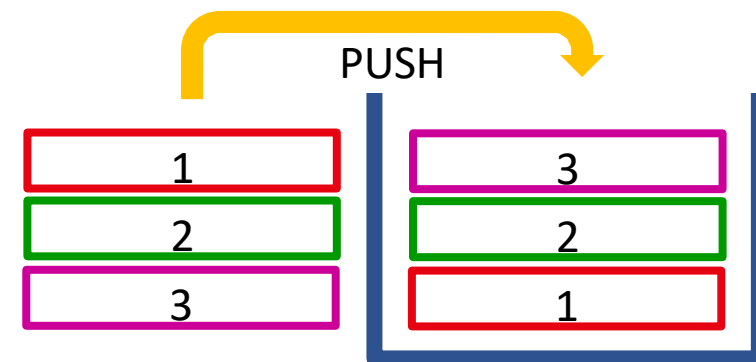


- STACK merupakan salah satu struktur data yang memiliki sistem kerja Last In First Out (LIFO), yang terakhir masuk pertama keluar.
- Stack dapat dibayangkan seperti tumpukan barang, hanya data terakhir yang dapat diperoleh (diakses) dengan satu langkah.
- Data-data yang terletak di bawahnya hanya bisa diambil (**POP**) setelah data data yang berada di atasnya diambil (dikeluarkan).

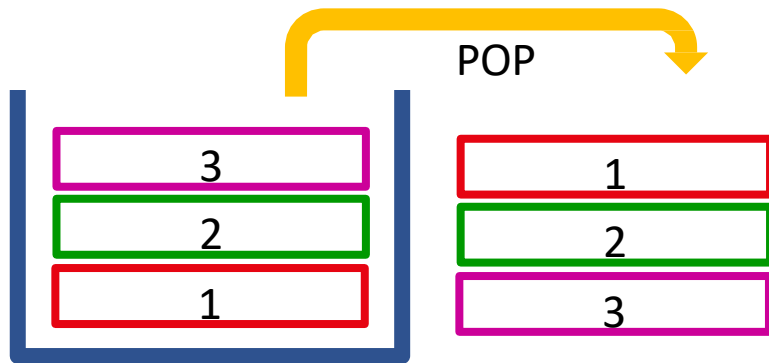


- Fungsi **PUSH**: digunakan untuk menambahkan data ke dalam stack. Penambahan data tidak bisa dilakukan apabila stack sudah penuh.
- Urutan perintahnya adalah: menambahkan nilai top dan menambahkan data pada posisi nilai top.
- Jika dalam Linked List menggunakan method addLast.
- Fungsi **PEEK**: digunakan untuk mengetahui data yang berada pada paling atas (puncak / peak) dari sebuah stack.

C. KELAS – **STACK**



C. KELAS – **STACK**



- Fungsi **POP**: digunakan untuk mengeluarkan data teratas stack dengan syarat bahwa stack tidak kosong.
- Urutan perintahnya adalah : menghapus data pada posisi nilai top dan menurunkan nilai top.



C. CONTOH KELAS – STACK

```
1 //File: LatihanStack.java
2 import java.util.*;
3 public class LatihanStack{
4     public static void main (String[] args)
5     {
6         Stack<String> s = new Stack<String>();
7
8         // menambah/push lima mahasiswa pada stack s
9         s.push("bagus");
10        s.push("Darla");
11        s.push("spanky");
12        s.pop();
13        s.push("Aji");
14        s.push("Rudi");
15        s.pop();
16        System.out.println("Nama mahasiswa yang terakhir (peak) adalah: " + s.peek());
17        System.out.println();
18        System.out.println("Nama-nama mahasiswa dengan urutan terbalik adalah: ");
19        while(!s.empty())
20            System.out.println(s.pop());
21        System.out.println();
22        System.out.println("Ukuran stack s saat ini adalah: " + s.size());
23    }
24 }
```

Console

Nama mahasiswa yang terakhir (peak) adalah: Aji

Nama-nama mahasiswa dengan urutan terbalik adalah:

Aji

Darla

bagus

Ukuran stack s saat ini adalah: 0

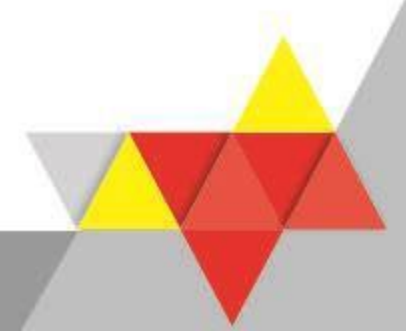
<<< Process finished (PID=12204). (Exit code 0)

===== READY =====



D. KELAS – **QUEUE**

- Queue (antrian) adalah barisan elemen yang apabila elemen ditambah maka penambahannya berada di posisi belakang (rear) dan jika dilakukan pengambilan elemen dilakukan di elemen paling depan (front). Oleh karena itu, queue bersifat FIFO (first in first out). Operasi-operasi dasar dari sebuah queue adalah :
 - **ENQUEUE** : proses penambahan elemen di posisi belakang
 - **DEQUEUE** : proses pengambilan elemen di posisi depan
- Selain operasi dasar di atas, ada pula operasi-operasi lain yang dapat dilakukan terhadap sebuah queue yaitu :
 - Operasi pemeriksaan **QUEUE KOSONG** (fungsi kosong)
 - Operasi pemeriksaan **QUEUE PENUH** (fungsi penuh).
 - Operasi **INISIALISASI** queue (fungsi inisialisasi)



D. CONTOH KELAS – QUEUE

```
1 //File: LatihanAntrian.java
2 import java.io.*;
3 import java.util.*;
4 public class LatihanAntrian {
5     private static int ukuran;
6     private static Queue<Integer> queue;
7     public static void main(String[] args) {
8         System.out.print("Berapa ukuran queue
9         di inginkan: ");
10        ukuran = inputData();
11        buatQueue();
12        bacaData();
13        tulisData();
14    }
15    private static void buatQueue() {
16        queue = new LinkedList<Integer>();
17    }
18    private static int inputData() {
19        BufferedReader bfr = new BufferedReader
20        (new InputStreamReader(System.in));
21        String angkaInput = null;
22        try{
23            angkaInput = bfr.readLine();
24        }
25    }
```

```
26 catch
27     (IOException e){e.printStackTrace();
28     }
29
30     int Data = Integer.valueOf(angkaInput).intValue();
31     return Data;
32 }
33
34 private static void tulisData(){
35     Integer data;
36     System.out.println("\nUkuran keluar elemen dari QUEUE
37     : ");
38     for (int i=0; i<ukuran; i++){
39         data = queue.remove();
40         System.out.println("Data ke-"+(i+1)+":"+data);
41     }
42     data = queue.size();
43     System.out.print("Ukuran Queue Sekarang adalah: " + data);
44 }
45
46 private static void bacaData(){
47     Integer data;
48     for(int i=0; i<ukuran; i++){
49         System.out.print("Data ke-"+(i + 1) + " : ");
50         data = inputData();
51         queue.add(data);
52     }
53     data = queue.size();
54     System.out.print("Ukuran QUEUE sekarang adalah: " + data);
55 }
56 }
```

D. CONTOH KELAS – QUEUE

Console

```
Berapa ukuran queue di inginkan: 4
Data ke-1 : 1
Data ke-2 : 2
Data ke-3 : 3
Data ke-4 : 4
Ukuran QUEUE sekarang adalah: 4
Ukuran keluar elemen dari QUEUE :
Data ke-1:1
Data ke-2:2
Data ke-3:3
Data ke-4:4
Ukuran Queue Sekarang adalah: 0 <<< Process finished (PID=20040). (Exit code 0)
===== READY =====
```



E. KELAS – DATE

```
1 //File: LatihanKelasDate.java
2 import java.util.Date; //Memanggil kelas Date yang terdapat pada paket java.util.Date
3
4 public class LatihanKelasDate {
5     public static void main (String[] args) {
6         Date WaktuSekarang = new Date(); //membuat objek bernama WaktuSekarang dari kelas Date
7
8         System.out.println("Waktu sekarang adalah: " + WaktuSekarang); //menampilkan isi dari objek WaktuSekarang
9     }
10 }
```

Console

Waktu sekarang adalah: Mon Jun 01 00:58:48 WIB 2020

<<< Process finished (PID=18056). (Exit code 0)

===== READY =====



**SELESAI
TERIMA KASIH**

