

# Dasar-Dasar Pemrograman 2

## Lab 10 Generics



FAKULTAS  
ILMU  
KOMPUTER

Sebenarnya, selama kalian membuat program di Java, kalian sudah menggunakan konsep generics secara tidak langsung. Masih ingat syntax di bawah ini?

```
import java.util.*;

class Main {
    public static void main(String[] args) {
        ArrayList<String> mahasiswa = new ArrayList<String>();
        ArrayList<Integer> tahunAjaran = new ArrayList<Integer>();
        mahasiswa.add("Burhan");
        tahunAjaran.add(2020);
    }
}
```

Tanda <String> di sebelah ArrayList menandakan kita memasukkan sebuah tipe class String sebagai parameter. Konsep ini disebut dengan “Type Parameterization” atau “Generics”.

Namun, apa itu generics? Mari kita lihat definisinya:



1. characteristic of or relating to a class or group of things; not specific.  
"chèvre is a generic term for all goat's milk cheese"

Dari yang kita lihat, generic itu berarti “Tidak spesifik”, menandakan sifatnya umum untuk tipe class manapun. Contohnya pada ArrayList itu sebuah class yang tidak spesifik untuk

menampung hanya String, hanya Integer, dll. Namun ArrayList itu bisa menampung tipe data apa saja yang diinginkan oleh pengguna struktur data ArrayList tersebut melalui parameter tipe class yang ditandai <> seperti tadi di contoh.

Tujuan dari generics adalah untuk mencegah runtime error dan menggantikannya dengan compile time error dan membatasi tipe data apa saja yang bisa masuk ke sebuah method. Mari langsung saja kita ke contoh pembuatan program dengan Generics :^)

Kita akan buat sebuah class Kepemilikan dimana nama pemilik akan disimpan dalam sebuah String, akan tetapi pemilik tersebut memiliki apa saja pastinya tidak dibatasi.

```
1 public class Main {
2     public static void main(String[] args) {
3         Laptop laptopApel = new Laptop("Mackbook Udara 2020");
4         Buku bukuDDP = new Buku("Introduction to Java Programming");
5
6         Kepemilikan<Laptop> k1 = new Kepemilikan<Laptop>("Burhan", laptopApel);
7         Kepemilikan<Buku> k2 = new Kepemilikan<Buku>("Depe", bukuDDP);
8
9         System.out.println(k1.toString());
10        System.out.println(k2.toString());
11    }
12 }
13
14 // NOTE: <T> bisa diubah menjadi apapun seperti <ABC> asal konsisten!
15 class Kepemilikan<T> {
16     String namaPemilik;
17     T benda;
18
19     Kepemilikan(String namaPemilik, T benda) {
20         this.namaPemilik = namaPemilik;
21         this.benda = benda;
22     }
23
24     public String toString() {
25         return this.benda.toString() + " milik " + this.namaPemilik;
26     }
27 }
28
29 class Laptop {
30     String jenis;
31
32     Laptop(String jenis) {
33         this.jenis = jenis;
```

```

34     }
35
36     public String toString() {
37         return "Laptop jenis " + this.jenis;
38     }
39 }
40 class Buku {
41     String judul;
42
43     Buku(String judul) {
44         this.judul = judul;
45     }
46
47     public String toString() {
48         return "Buku berjudul " + this.judul;
49     }
50 }

```

Contoh program panjang, namun cobalah membaca pelan-pelan yaa :D

**Poin-poin penting yang harus diperhatikan dalam contoh program:**

Mari kita lihat dari baris definisi class dahulu

- Baris 15 menyatakan bahwa dalam instansiasi class Kepemilikan kita harus memberikan parameter berupa nama class yang ingin dijadikan tipe T
- Baris 17 menyatakan variable benda dimana tipenya masih belum pasti (tergantung dari parameter T sehingga bisa saja String, Integer, dll.)
- Baris 19 juga mirip dengan 17, yaitu constructor yang menerima parameter berupa T dari yang kita telah definisikan

Lanjut ke penggunaan class yang telah didefinisikan

- Baris 6 dan 7 merupakan instansiasi dari class Kepemilikan dimana kita memberikan parameter berupa tipe class Laptop dan Buku (perhatikan tanda <> nya), **yang berarti tipe T masing-masing sudah dianggap sebagai Laptop dan Buku** dengan contohnya constructor masing-masing menerima Laptop untuk k1 dan Buku untuk k2.

Di kasus ini yang utama adalah **class Kepemilikan di mana kita bisa menerima tipe data apapun** untuk benda yang dimiliki, sekarang bayangkan apabila kita tidak menggunakan generic, maka kita harus menggunakan superclass lainnya dan menggunakan type-casting yang sendirinya tidak compile-safe.

Kita juga bisa **membatasi generic class apa saja yang bisa masuk sebagai parameter** dengan syarat seperti apakah class tersebut superclass atau subclass dari suatu class? Kita lihat syntaxnya sebagai berikut

```
1 class ClassOne<V extends Superclass> {  
2 }  
3  
4 class ClassTwo<V super Subclass> {  
5 }
```

Dimana ClassOne itu membatasi Generic type yang diterima hanya jika Generic type tersebut merupakan **instance class Superclass itu sendiri atau subclass dari Superclass**

Untuk ClassTwo itu membatasi Generic type yang diterima hanya jika Generic type tersebut merupakan **instance class Subclass itu sendiri atau merupakan class yang menjadi superclass dari Subclass**

Penjelasan program perbaris:

- Baris 1 kita mendefinisikan class bernama ClassOne namun kita menambahkan parameter tipe generic **dengan syarat** tipe generic yang dimasukkan harus merupakan instance yang merupakan subclass atau tipe Superclass itu sendiri.
- Baris 4 juga mirip dengan ClassOne namun ClassTwo mewajibkan kita untuk memasukkan parameter tipe yang merupakan instance dari superclass atau tipe Subclass itu sendiri.

# Soal Lab 10

---

## Mudik Lebaran

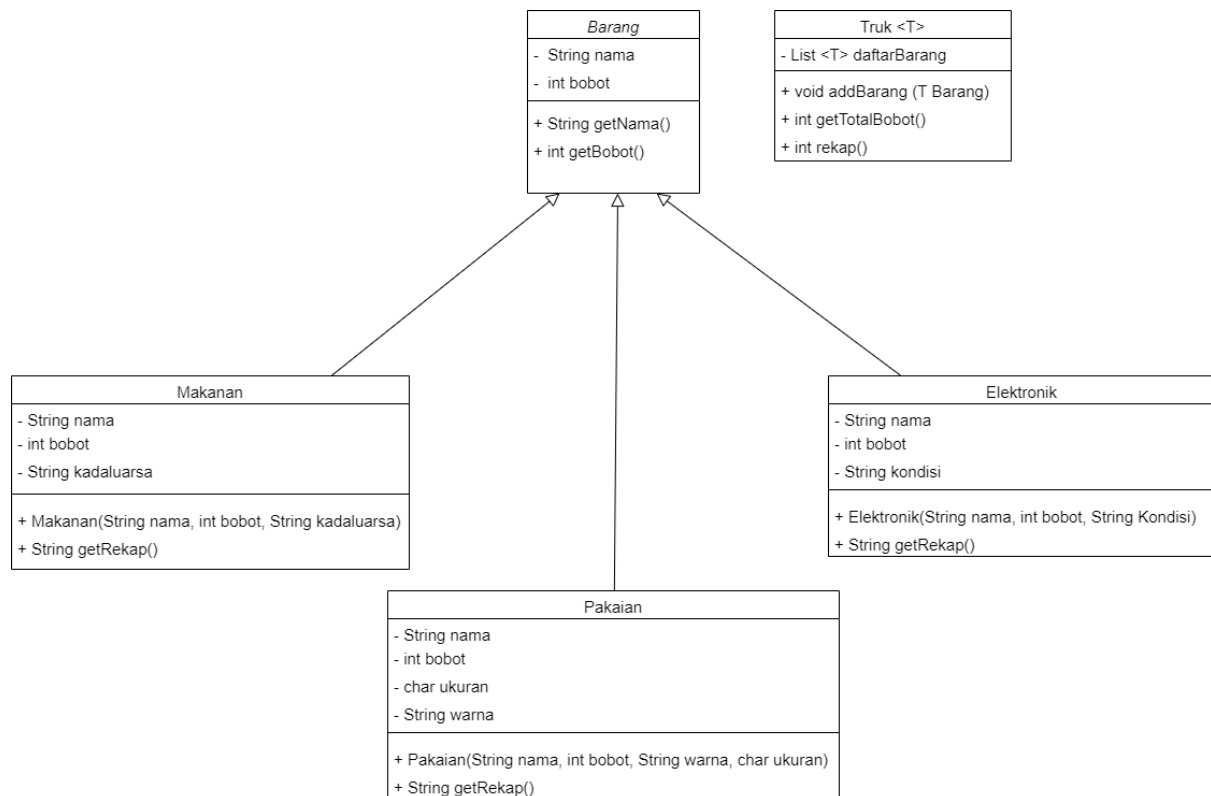


sumber: sasihsaum.com

Klik.... klik.... klik.... sambil memainkan **The Counter** sembari menunggu waktu, tak terasa kamu sudah sampai di penghujung perjalanan! Kamu terdiam sejenak sembari memikirkan apa saja yang sudah kamu lalui selama ini. Mulai dari memberi hadiah untuk **Sydney**, membantu Dek Depe bermain **Pacilmon Go**, mengunjungi Fasilkom secara virtual dengan **Pacilian Constructor**, berpetualang di **PacilGeon**, hingga bermain dengan **DogeGochi**. Ah, sungguh kenangan yang indah, dan kamu yakin bahwa semua itu akan bermanfaat untukmu di masa depan.

Tak ingin terlalu larut dalam nostalgia, kamu segera mengemas barang-barangmu untuk kembali ke kampung halaman. Namun, kamu baru ingat bahwa tahun ini diberlakukan larangan mudik oleh pemerintah. Oleh karena itu, kamu memutuskan untuk mengirim barang-barang parcel lebaran untuk keluargamu yang berada di kampung halaman. Oleh karena barang yang ingin kamu kirim jumlahnya sangat banyak, kamu membutuhkan truk untuk mengirim barang-barang tersebut. Untuk memudahkan kamu dalam manajemen barang-barang yang akan dikirim, kamu memutuskan untuk membuat sebuah sistem bernama Sistem Informasi Mudik atau **SIMudik**.

## Spesifikasi Program:



- **SIMudik.java**

Class ini merupakan class yang berperan menjadi simulator. Pada class ini disimpan 3 truk yaitu `trukMakanan` yang bertujuan untuk menyimpan objek Makanan, `trukPakaian` yang bertujuan untuk menyimpan objek Pakaian, dan `trukElektronik` yang bertujuan untuk menyimpan objek Elektronik.

- **Method `addMakanan()`**

Method ini bertujuan untuk menerima input data-data objek Makanan dan menambahkan objek tersebut ke `trukMakanan`.

- **Method `addPakaian()`**

Method ini bertujuan untuk menerima input data objek Pakaian dan menambahkan objek tersebut ke `trukPakaian`.

- **Method `addElektronik()`**

Method ini bertujuan untuk menerima input data objek Elektronik dan menambahkan objek tersebut ke `trukElektronik`.

- **Method `run()`**

Method ini sebagai method untuk berinteraksi dengan pengguna melalui perintah yang diberikan oleh pengguna. Perintah berupa sebuah bilangan bulat yaitu 1 (menambah barang), 2 (mencetak rekap truk), dan 3 (keluar).

- **Truk.java**

Truk adalah sebuah class generic yang berfungsi untuk menyimpan Barang. Ingat bahwa kamu harus memastikan **class Truk hanya menerima object yang berupa Elektronik, Makanan, atau Pakaian**. Dengan kata lain, kamu harus memastikan bahwa tidak bisa dibuat objek Truk untuk tipe selain Barang, seperti misalnya `Truk<String>`, `Truk<Integer>`, dan lain-lain. Berikut adalah atribut dan method yang dimiliki oleh class Truk.

Atribut:

- `ArrayList<T> daftarBarang`  
Atribut ini berupa sebuah ArrayList yang menampung objek-objek yang tipenya sesuai dengan tipe Truk saat objek Truk diinstansiasi. Dengan kata lain, sebuah `Truk<Pakaian>` memiliki atribut `daftarBarang` yang berupa ArrayList of Barang.
- `String nama`  
Atribut ini berupa sebuah String yang merepresentasikan nama dari sebuah objek Truk

Method:

- `addBarang()`  
Method ini merupakan sebuah method generic yang berfungsi untuk menambahkan objek Barang ke atribut `daftarBarang`
- `getTotalBobot()`  
Method ini merupakan sebuah method yang mengembalikan total bobot dari semua Barang yang ada pada atribut `daftarBarang`
- `rekap()`  
Method ini merupakan sebuah method untuk mencetak rekap dari sebuah truk. Untuk format keluaran bisa anda lihat di contoh keluaran 1 dan contoh keluaran 2

- **Barang.java**

Barang merupakan sebuah abstract class yang memiliki 3 buah subclass yaitu Makanan, Elektronik, dan Pakaian. Berikut adalah atribut dan method yang dimiliki oleh class Barang

Atribut:

- `bobot`  
Atribut ini merepresentasikan bobot dari objek barang yang bersangkutan dalam satuan gram
- `nama`  
Atribut ini merepresentasikan nama dari objek barang yang bersangkutan.

Method:

- `getBobot()`  
Method ini adalah method getter untuk atribut bobot.
- `getNama()`  
Method ini adalah method getter untuk atribut nama
- `getRekap()`  
Method ini mengembalikan informasi dari sebuah barang. Untuk formatnya dapat dilihat pada testcase.

- **Elektronik.java**

Elektronik merupakan salah satu concrete class yang meng-inherit class Barang. Pada class ini, terdapat atribut kondisi yang merepresentasikan kondisi dari sebuah objek elektronik.

- **Makanan.java**

Makanan merupakan salah satu concrete class yang meng-inherit class Barang. Pada class ini, terdapat atribut `expiredDate` yang merepresentasikan tanggal kedaluwarsa dari sebuah objek Makanan. Tidak ada format khusus untuk penulisan `expiredDate`.



- **Pakaian.java**

Pakaian merupakan salah satu concrete class yang meng-inherit class Barang. Pada class ini, terdapat atribut ukuran yang merepresentasikan ukuran dan atribut warna yang merepresentasikan warna dari sebuah objek Pakaian.

### **Petunjuk Pengerjaan:**

Disediakan sebuah template untuk memudahkan anda mengerjakan Lab 10.

1. Ubahlah class Truk agar menjadi sebuah class generic. Pastikan bahwa class generic Truk **hanya bisa menerima objek Barang dan objek yang meng-extends Barang** sebagai type parameter.
2. Tambahkan atribut ArrayList generic bernama daftarBarang pada class Truk
3. Tambahkan parameter generic dan implementasikan method addBarang() pada class Truk
4. Implementasikan method getTotalBobot() pada class Truk
5. Implementasikan method rekap() pada class Truk
6. Pada method addMakanan() di class SIMudik, buatlah objek Makanan berdasarkan data yang sudah diinput lalu tambahkan ke atribut trukMakanan
7. Pada method addPakaian di class SIMudik, buatlah objek Pakaian berdasarkan data yang sudah diinput lalu tambahkan ke atribut trukPakaian
8. Pada method addElektronik di class SIMudik, buatlah objek Elektronik berdasarkan data yang sudah diinput lalu tambahkan ke atribut trukElektronik

**Test Case** (Tulisan berwarna **biru** menandakan masukan dari *user*)

**Contoh Masukan 1:**

----- Selamat Datang di Sistem Informasi Mudik -----

Masukkan perintah 1 untuk menambah barang

Masukkan perintah 2 untuk melihat rekap dari sebuah truk

Masukkan perintah 3 untuk keluar dari sistem

Masukkan perintah: **1**

Masukkan jenis barang (Makanan / Pakaian / Elektronik): **Makanan**

Masukkan nama makanan: **Mie Sed00p**

Masukkan bobot (dalam gram): **120**

Masukkan tanggal kedaluwarsa: **31 Agustus 2021**

Makanan Mie Sed00p ditambahkan ke truk makanan!

Masukkan perintah: **1**

Masukkan jenis barang (Makanan / Pakaian / Elektronik): **Pakaian**

Masukkan nama pakaian: **Kaos Uniqque**

Masukkan bobot (dalam gram): **750**

Masukkan warna: **Merah**

Masukkan ukuran (S/M/L): **M**

Pakaian Kaos Uniqque ditambahkan ke truk pakaian!

Masukkan perintah: **1**

Masukkan jenis barang (Makanan / Pakaian / Elektronik): **Makanan**

Masukkan nama makanan: **Coklat Rokha**

Masukkan bobot (dalam gram): **1000**

Masukkan tanggal kedaluwarsa: **31 Desember 2021**

Makanan Coklat Rokha ditambahkan ke truk makanan!

Masukkan perintah: **2**

Masukkan jenis truk (Makanan / Pakaian / Elektronik): **Makanan**

Berikut rekap barang-barang dari Truk Makanan

Makanan Mie Sed00p

Bobot: 120

Kedaluwarsa: 31 Agustus 2021

Makanan Coklat Rokha

Bobot: 1000

Kedaluwarsa: 31 Desember 2021

Total bobot barang di truk ini: 1120 gram

Masukkan perintah: **2**

Masukkan jenis truk (Makanan / Pakaian / Elektronik): **Elektronik**

Belum ada barang di Truk Elektronik

Masukkan perintah: 3

Terima kasih telah menggunakan Sistem Informasi Mudik!

### Contoh Masukan 2:

----- Selamat Datang di Sistem Informasi Mudik -----

Masukkan perintah 1 untuk menambah barang

Masukkan perintah 2 untuk melihat rekap dari sebuah truk

Masukkan perintah 3 untuk keluar dari sistem

Masukkan perintah: 1

Masukkan jenis barang (Makanan / Pakaian / Elektronik): Elektronik

Masukkan nama barang elektronik: TV Toshiherba

Masukkan bobot (dalam gram): 7000

Deskripsikan kondisi saat ini: Masih baru dan bagus

Barang elektronik TV Toshiherba ditambahkan ke truk elektronik!

Masukkan perintah: 2

Masukkan jenis truk (Makanan / Pakaian / Elektronik): Elektronik

Berikut rekap barang-barang dari Truk Elektronik

Barang Elektronik TV Toshiherba

Bobot: 7000

Kondisi: Masih baru dan bagus

Total bobot barang di truk ini: 7000 gram

Masukkan perintah: 1

Masukkan jenis barang (Makanan / Pakaian / Elektronik): Elektronik

Masukkan nama barang elektronik: Mesin Cuci Sumsang

Masukkan bobot (dalam gram): 5000

Deskripsikan kondisi saat ini: Sering bergetar ketika dipakai

Barang elektronik Mesin Cuci Sumsang ditambahkan ke truk elektronik!

Masukkan perintah: 2

Masukkan jenis truk (Makanan / Pakaian / Elektronik): Elektronik

Berikut rekap barang-barang dari Truk Elektronik

Barang Elektronik TV Toshiherba

Bobot: 7000

Kondisi: Masih baru dan bagus

Barang Elektronik Mesin Cuci Sumsang

Bobot: 5000

Kondisi: Sering bergetar ketika dipakai

Total bobot barang di truk ini: 12000 gram

Masukkan perintah: 2

Masukkan jenis truk (Makanan / Pakaian / Elektronik): Pakaian

Belum ada barang di truk ini

Masukkan perintah: 1

Masukkan jenis barang (Makanan / Pakaian / Elektronik): Pakaian

Masukkan nama pakaian: Kaos Supremi

Masukkan bobot (dalam gram): 500

Masukkan warna: Putih

Masukkan ukuran (S/M/L): S

Pakaian Kaos Supremi ditambahkan ke truk pakaian!

Masukkan perintah: 2

Masukkan jenis truk (Makanan / Pakaian / Elektronik): Pakaian

Berikut rekap barang-barang dari Truk Pakaian

Barang Elektronik Kaos Supremi

Bobot: 500

Ukuran: S

Warna: Putih

Total bobot barang di truk ini: 500 gram

Masukkan perintah: 3

Terima kasih telah menggunakan Sistem Informasi Mudik!

## Komponen Penilaian

- 20% Implementasi penambahan barang pada class SIMudik
- 35% Implementasi generics pada class Truk
- 25% Implementasi method getTotalBobot dan rekap pada class Truk
- 10% Dokumentasi dan kerapian kode

## Revisi

Belum ada Revisi

Kumpulkan berkas .java yang telah di-zip dengan format penamaan seperti berikut.

Lab10\_[Kelas]\_[KodeAsdos]\_[NPM]\_[NamaLengkap].zip

Contoh:

Lab10\_A\_LN\_1234567890\_DekDepe.zip