

JOBSHEET 11

Nama : Muhammad Ibnu Zauzi

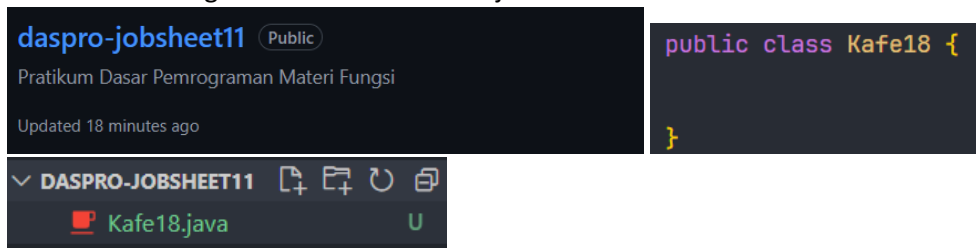
NIM : 24410720016

ABSEN : 18

KELAS : 1D

Percobaan 1: Membuat Fungsi Tanpa Parameter

1. Buat repository baru dengan nama daspro-jobsheet11. Selanjutnya buat class baru, dan simpan file tersebut dengan nama KafeNoAbsen.java.



2. Buat fungsi Menu di dalam class tersebut.

```
public static void Menu() {  
    System.out.println(x:"==== MENU RESTO KAFE ====");  
    System.out.println(x:"1. Kopi Hitam - Rp 15,000");  
    System.out.println(x:"2. Cappuccino - Rp 20,000");  
    System.out.println(x:"3. Latte - Rp 22,000");  
    System.out.println(x:"4. Teh Tarik - Rp 12,000");  
    System.out.println(x:"5. Roti Bakar - Rp 10,000");  
    System.out.println(x:"6. Mie Goreng - Rp 18,000");  
    System.out.println(x:"=====");  
    System.out.println(x:"Silakan pilih menu yang Anda inginkan.");  
}
```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi Menu dari dalam fungsi main

```
public static void main(String[] args) {  
    Menu();  
}
```

4. Push dan commit ke git!

```
$ git commit -m "percobaan 1"  
[main (root-commit) fa49366] percobaan 1  
1 file changed, 17 insertions(+)  
create mode 100644 Kafe18.java  
  
HP@LAPTOP-VICTUS MINGW64 /c/MatKu1/DasarF  
main)  
$ git push  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.
```

Pertanyaan

1. Tidak, fungsi tanpa parameter tidak selalu harus menjadi fungsi void, fungsi tanpa parameter juga bisa mengembalikan nilai tipe lain.
2. Bisa menampilkan daftar menu tanpa menggunakan fungsi Menu()

```
public static void main(String[] args) {  
    System.out.println(x:"==== MENU RESTO KAFE ====");  
    System.out.println(x:"1. Kopi Hitam - Rp 15,000");  
    System.out.println(x:"2. Cappuccino - Rp 20,000");  
    System.out.println(x:"3. Latte - Rp 22,000");  
    System.out.println(x:"4. Teh Tarik - Rp 12,000");  
    System.out.println(x:"5. Roti Bakar - Rp 10,000");  
    System.out.println(x:"6. Mie Goreng - Rp 18,000");  
    System.out.println(x:"=====");  
    System.out.println(x:"Silakan pilih menu yang Anda inginkan.");  
}
```

3. - Modularitas dan Struktur Kode yang Lebih Baik
- Mengurangi Pengulangan Kode
- Memudahkan Pemeliharaan Kode

Percobaan 2: Membuat Fungsi Dengan Parameter

1. Ubah fungsi Menu dengan dua buah parameter bertipe String dan boolean di dalam class KafeNoAbsen.java.

```
public static void Menu(String namaPelanggan, boolean isMember) {  
    System.out.println("Selamat datang, " + namaPelanggan + "!");  
  
    if (isMember) {  
        System.out.println(x:"Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");  
    }  
  
    System.out.println(x:"==== MENU RESTO KAFE ====");  
    System.out.println(x:"1. Kopi Hitam - Rp 15,000");  
    System.out.println(x:"2. Cappuccino - Rp 20,000");  
    System.out.println(x:"3. Latte - Rp 22,000");  
    System.out.println(x:"4. Teh Tarik - Rp 12,000");  
    System.out.println(x:"5. Roti Bakar - Rp 10,000");  
    System.out.println(x:"6. Mie Goreng - Rp 18,000");  
    System.out.println(x:"=====");  
    System.out.println(x:"Silakan pilih menu yang Anda inginkan.");  
}
```

2. Eksekusi fungsi UcapanTambahan dari dalam fungsi main.

```
public static void main(String[] args) {  
    Menu(namaPelanggan:"Andi", isMember:true);  
}
```

Pertanyaan

1. Parameter dalam fungsi digunakan untuk menerima input dari luar fungsi yang kemudian bisa diproses di dalam fungsi tersebut
2. Parameter mirip dengan variabel, namun ada perbedaan. Parameter adalah variabel khusus yang dideklarasikan di dalam tanda kurung pada definisi fungsi. Parameter menerima nilai yang diberikan saat fungsi dipanggil.
3. Parameter `isMember` pada method `Menu` berfungsi sebagai apakah pelanggan adalah anggota (member) atau bukan. Jika `isMember` bernilai `true`, maka program akan menampilkan pesan, Jika `isMember` bernilai `false`, maka pesan diskon ini tidak akan ditampilkan.
4. maka akan terjadi **error**. Java mengharuskan jumlah dan tipe parameter saat pemanggilan fungsi sesuai dengan yang dideklarasikan pada fungsi tersebut. Karena `Menu` membutuhkan dua parameter (`String` dan `boolean`), pemanggilan tanpa parameter tidak akan sesuai dengan definisi fungsi, sehingga tidak akan bisa dikompilasi.

```
public static void Menu(String namaPelanggan, boolean isMember, String kodePromo) {
    System.out.println("Selamat datang, " + namaPelanggan + "!");

    if (isMember) {
        System.out.println(x:"Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
    }

    if (kodePromo.equals(anObject:"DISKON50")) {
        System.out.println(x:"Anda mendapatkan diskon 50%!");
    } else if (kodePromo.equals(anObject:"DISKON30")) {
        System.out.println(x:"Anda mendapatkan diskon 30%!");
    } else {
        System.out.println(x:"Kode promo tidak valid.");
    }

    System.out.println(x:"===== MENU RESTO KAFE =====");
    System.out.println(x:"1. Kopi Hitam - Rp 15,000");
    System.out.println(x:"2. Cappuccino - Rp 20,000");
    System.out.println(x:"3. Latte - Rp 22,000");
    System.out.println(x:"4. Teh Tarik - Rp 12,000");
    System.out.println(x:"5. Roti Bakar - Rp 10,000");
    System.out.println(x:"6. Mie Goreng - Rp 18,000");
    System.out.println(x:"=====");
    System.out.println(x:"Silakan pilih menu yang Anda inginkan.");
}

Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
public static void main(String[] args) {
    Menu(namaPelanggan:"Andi", isMember:true, kodePromo:"DISKON50");
}
```

5.

Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

1. Pada Class yang sama buat fungsi `hitungTotalHargaNoAbsen` di dalam class tersebut yang mengembalikan nilai total harga dan parameter masukan pilihan Menu dan banyaknya pesanan

```
public static int HitungTotalHarga(int pilihanMenu, int banyakItem){
    int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};

    int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;
    return hargaTotal;
}
```

2. Modifikasi fungsi main di dalam class tersebut, dan eksekusi fungsi `PenerimaUcapan` dari dalam fungsi main

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s:"\nMasukkan nomor menu yang ingin anda pesan: ");
    int pilihanMenu = sc.nextInt();
    System.out.print(s:"Masukkan jumlah item yang ingin dipesan: ");
    int banyakItem = sc.nextInt();

    int totalHarga = HitungTotalHarga(pilihanMenu, banyakItem);

    System.out.println("Total harga untuk pesanan Anda: Rp" + totalHarga);
}
```

Pertanyaan

1. Suatu fungsi membutuhkan nilai kembalian (menggunakan `return`) ketika fungsi tersebut perlu memberikan hasil atau output kepada pemanggil fungsi. Dengan `return`, hasil dari fungsi bisa digunakan untuk melakukan proses lebih lanjut, menyimpan hasil, atau menampilkan data.

```
public static int HitungTotalHarga(int pilihanMenu, int banyakItem, String kodePromo){
    double diskon = 0;
    int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};
    int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;

    if (kodePromo.equals(anObject:"DISKON50")) {
        diskon = hargaTotal * 0.5;
        hargaTotal -= diskon;
        System.out.println(x:"Anda mendapatkan diskon 50%!");
    } else if (kodePromo.equals(anObject:"DISKON30")) {
        diskon = hargaTotal * 0.3;
        hargaTotal -= diskon;
        System.out.println(x:"Anda mendapatkan diskon 30%!");
    } else {
        System.out.println(x:"Kode promo tidak valid.");
    }

    return hargaTotal;
}

Run | Debug | Tabnine | Edit | Test | Fix | Explain | Document | Ask
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s:"\nMasukkan nomor menu yang ingin anda pesan: ");
    int pilihanMenu = sc.nextInt();
    System.out.print(s:"Masukkan jumlah item yang ingin dipesan: ");
    int banyakItem = sc.nextInt();
    System.out.print(s:"Masukkan kode promo: ");
    String kodePromo = sc.next();

    int totalHarga = HitungTotalHarga(pilihanMenu, banyakItem, kodePromo);
}
```

- 2.

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int totalHargaKeseluruhan = 0;

    while (true) {
        System.out.print(s:"\nMasukkan nomor menu yang ingin Anda pesan (1-6) atau 0 untuk selesai: ");
        int pilihanMenu = sc.nextInt();
        if (pilihanMenu == 0) {
            break;
        }

        System.out.print(s:"Masukkan jumlah item yang ingin dipesan: ");
        int banyakItem = sc.nextInt();
        sc.nextLine();

        System.out.print(s:"Masukkan kode promo: ");
        String kodePromo = sc.nextLine();

        int totalHarga = HitungTotalHarga(pilihanMenu, banyakItem, kodePromo);
        totalHargaKeseluruhan += totalHarga;

        System.out.println("Total harga untuk pesanan ini: Rp " + totalHarga);
    }

    System.out.println("\nTotal harga keseluruhan pesanan Anda: Rp " + totalHargaKeseluruhan);
}

```

3.

Percobaan 4: Fungsi Varargs

1. Buat class baru, simpan file tersebut dengan nama PengunjungCafeNoAbsen.java.
2. Buatlah fungsi daftarPengunjung (bertipe void) di dalam class tersebut dengan menggunakan parameter varArgs

```

public static void daftarPengunjung(String... namaPengunjung) {
    System.out.println(x:"Daftar Nama Pengunjung:");
    for (int i = 0; i < namaPengunjung.length; i++) {
        System.out.println("- " + namaPengunjung[i]);
    }
}

```

3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi namaPengunjung dari dalam fungsi main.

```

public static void main(String[] args) {
    daftarPengunjung(...namaPengunjung:"Ali", "Budi", "Citra");
}

```

Pertanyaan

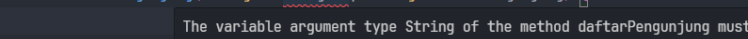
1. Penulisan parameter dengan String... namaPengunjung menunjukkan penggunaan **varargs** (variable arguments) dalam Java. Varargs memungkinkan kita untuk mengirimkan sejumlah argumen bertipe String tanpa harus menentukan jumlahnya terlebih dahulu.

```

public static void daftarPengunjung(String... namaPengunjung) {
    System.out.println(x:"Daftar Nama Pengunjung:");
    for (String pengunjung : namaPengunjung) {
        System.out.println("- " + pengunjung);
    }
}

```

2.

- 
- ```
public static void daftarPengenjung(String... kategori, String... namaPengenjung) {
 // ...
}

Run | Debug | Tabnine | Edit | Test | Fix | Explain | Docs
public static void main(String[] args) {
 daftarPengenjung(kategori:"Ali", ...namaPengenjung:"Budi", "Citra");
}
```
- The variable argument type String of the method daftarPengenjung must be the last parameter Java(67189279)
- `String[] kategori` - `PengenjungCafe18.daftarPengenjung(String[], String...)`
- [View Problem \(Alt+F8\)](#) [Quick Fix... \(Ctrl+.\)](#)

1. Buat class baru, simpan file tersebut dengan nama Percobaan6NoAbsen.java.

- ☕ Percobaan618.java 1, U

- ```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
  
    int p,l,t,L,vol;  
  
    System.out.println(x:"Masukkan panjang");  
    p = input.nextInt();  
    System.out.println(x:"Masukkan lebar");  
    l = input.nextInt();  
    System.out.println(x:"Masukkan tinggi");  
    t = input.nextInt();  
  
    L = p*l;  
    System.out.println("Luas Persegi panjang adalah " + L);  
  
    vol = p*l*t;  
    System.out.println("Volume balok adalah " + vol);  
}
```

- ```
static int hitungLuas(int pjg, int lb){
 int Luas = pjg*lb;
 return Luas;
}

Tabnine | Edit | Test | Explain | Document | Ask

static int hitungVolume(int tinggi, int a, int b){
 int volume = hitungLuas(a,b)*tinggi;
 return volume;
}
```

```

public static void main(String[] args) {
 Scanner input = new Scanner(System.in);

 int p,l,t,L,vol;

 System.out.println("Masukkan panjang");
 p = input.nextInt();
 System.out.println("Masukkan lebar");
 l = input.nextInt();
 System.out.println("Masukkan tinggi");
 t = input.nextInt();

 L = hitungLuas(p, l);
 System.out.println("Luas Persegi panjang adalah " + L);

 vol = hitungVolume(t, p, l);
 System.out.println("Volume balok adalah " + vol);
}

```

4. Lakukan validasi hasil dengan memberikan contoh input pada program tersebut! Tampilkan hasilnya!

```

Masukkan panjang
4
Masukkan lebar
5
Masukkan tinggi
3
Luas Persegi panjang adalah 20
Volume balok adalah 60

```

## Pertanyaan

1.
  - Program mulai dari method main.
  - Membuat objek Scanner untuk membaca input dari pengguna.
  - Mendeklarasikan variabel p, l, t, L, dan vol sebagai tipe int.
  - Menampilkan pesan "Masukkan panjang", membaca nilai panjang (p) dari pengguna.
  - Menampilkan pesan "Masukkan lebar", membaca nilai lebar (l) dari pengguna.
  - Menampilkan pesan "Masukkan tinggi", membaca nilai tinggi (t) dari pengguna.
  - Memanggil method hitungLuas dengan parameter p dan l untuk menghitung luas persegi panjang.
  - Di dalam hitungLuas, hasil perkalian  $p * l$  dihitung dan dikembalikan.
  - Menyimpan hasil dari hitungLuas ke variabel L, lalu mencetak hasilnya.
  - Memanggil method hitungVolume dengan parameter t, p, dan l untuk menghitung volume balok.
  - Di dalam hitungVolume, method hitungLuas dipanggil untuk menghitung luas alas, lalu dikalikan dengan t untuk menghitung volume.
  - Menyimpan hasil dari hitungVolume ke variabel vol, lalu mencetak hasilnya.
  - Program selesai.

```

1234567

```

- 2.

- Program dimulai dari method main.
  - Method Jumlah dipanggil dengan parameter 1 dan 1.
  - Mengembalikan hasil penjumlahan  $1 + 1 = 2$ , yang disimpan ke dalam variabel temp.
  - Method TampilJumlah dipanggil dengan parameter temp = 2 dan 5.
  - Di dalam TampilJumlah, method Jumlah dipanggil kembali dengan parameter bil1 = 2 dan bil2 = 5.
  - Mengembalikan hasil penjumlahan  $2 + 5 = 7$ .
  - Method TampilHinggaKei dipanggil dengan parameter i = 7.
  - Melakukan iterasi dari j = 1 hingga j = 7, mencetak angka secara berurutan.
  - Program selesai.
3. Kapan sebuah fungsi harus menggunakan parameter?
- **Gunakan parameter** jika fungsi membutuhkan **data dari luar** supaya bisa dijalankan.
  - **Tidak perlu parameter** jika fungsi tidak butuh data dari luar. Semua data yang dibutuhkan sudah tersedia di dalam fungsi itu sendiri.
- Kapan sebuah fungsi harus memiliki nilai kembalian?
- **Gunakan nilai kembalian** jika fungsi menghasilkan sesuatu yang perlu digunakan kembali di bagian program lain.
  - **Tidak perlu nilai kembalian** jika fungsi hanya melakukan tugas tertentu tanpa perlu memberikan hasil balik.

## Tugas

```
public class Kubus18 {
 Tabnine | Edit | Test | Explain | Document | Ask
 public static int hitungVolume(int sisi) {
 return sisi * sisi * sisi;
 }

 Tabnine | Edit | Test | Explain | Document | Ask
 public static int hitungLuasPermukaan(int sisi) {
 return 6 * (sisi * sisi);
 }

 Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
 public static void main(String[] args) {
 int sisi = 5;

 int volume = hitungVolume(sisi);
 int luasPermukaan = hitungLuasPermukaan(sisi);

 System.out.println("Panjang sisi kubus: " + sisi);
 System.out.println("Volume kubus: " + volume);
 System.out.println("Luas permukaan kubus: " + luasPermukaan);
 }
}
```

1.



```

import java.util.Scanner;

public class PenjualanCafe {

 public static void tampilkanPenjualan(String[] menu, int[][] penjualan) {
 System.out.println("\nData Penjualan:");
 for (int i = 0; i < menu.length; i++) {
 System.out.print(menu[i] + ": ");
 for (int j = 0; j < penjualan[i].length; j++) {
 System.out.print(penjualan[i][j] + " ");
 }
 System.out.println();
 }
 }

 public static void menuPenjualanTertinggi(String[] menu, int[][] penjualan) {
 int maxPenjualan = 0;
 String menuTerlaris = "";
 for (int i = 0; i < menu.length; i++) {
 int totalPenjualan = 0;
 for (int j = 0; j < penjualan[i].length; j++) {
 totalPenjualan += penjualan[i][j];
 }
 if (totalPenjualan > maxPenjualan) {
 maxPenjualan = totalPenjualan;
 menuTerlaris = menu[i];
 }
 }
 System.out.println("\nMenu dengan penjualan tertinggi: " + menuTerlaris + " (Total: " + maxPenjualan + ")");
 }
}

```

2.

```

 public static void rataRataPenjualan(String[] menu, int[][] penjualan) {
 System.out.println("\nRata-rata penjualan untuk setiap menu:");
 for (int i = 0; i < menu.length; i++) {
 int totalPenjualan = 0;
 for (int j = 0; j < penjualan[i].length; j++) {
 totalPenjualan += penjualan[i][j];
 }
 double rataRata = (double) totalPenjualan / penjualan[i].length;
 System.out.println(menu[i] + ": " + rataRata);
 }
 }

 public static void inputDataPenjualan(String[] menu, int[][] penjualan) {
 Scanner scanner = new Scanner(System.in);

 System.out.println("\nMasukkan data penjualan baru:");
 for (int i = 0; i < menu.length; i++) {
 System.out.println("Masukkan data penjualan untuk " + menu[i] + " (7 hari):");
 for (int j = 0; j < penjualan[i].length; j++) {
 System.out.print("Hari ke- " + (j + 1) + ": ");
 penjualan[i][j] = scanner.nextInt();
 }
 }
 System.out.println("\nData penjualan berhasil diupdate.");
 }
}

```

```

public static void main(String[] args) {
 String[] menu = {"Kopi", "Teh", "Es Degan", "Roti Bakar", "Gorengan"};
 int[][] penjualan = {
 {20, 20, 25, 20, 10, 60, 10},
 {30, 80, 40, 10, 15, 20, 25},
 {5, 9, 20, 25, 10, 50, 45},
 {50, 8, 17, 18, 10, 30, 6},
 {15, 10, 16, 15, 10, 10, 55}
 };

 Scanner scanner = new Scanner(System.in);
 int pilihan;

 do {
 System.out.println("\n--- Menu Program Penjualan Cafe ---");
 System.out.println("1. Tampilkan seluruh data penjualan");
 System.out.println("2. Input atau ubah data penjualan");
 System.out.println("3. Tampilkan menu dengan penjualan tertinggi");
 System.out.println("4. Tampilkan rata-rata penjualan setiap menu");
 System.out.println("5. Keluar");
 System.out.print("Masukkan pilihan: ");
 pilihan = scanner.nextInt();
 } while (pilihan != 5);
}

```

```

switch (pilihan) {
 case 1:
 tampilkanPenjualan(menu, penjualan);
 break;
 case 2:
 inputDataPenjualan(menu, penjualan);
 break;
 case 3:
 menuPenjualanTertinggi(menu, penjualan);
 break;
 case 4:
 rataRataPenjualan(menu, penjualan);
 break;
 case 5:
 System.out.println("Keluar dari program.");
 break;
 default:
 System.out.println("Pilihan tidak valid. Coba lagi.");
}
}

```