

JOBSHEET 11

Fungsi 1

1. Praktikum

1.1 Percobaan 1: Membuat Fungsi Tanpa Parameter

Waktu Percobaan: 40 menit

1. Buat folder baru pada repositori lokal Anda, beri nama **jobsheet11**. Selanjutnya buat *class* baru, dan simpan file tersebut dengan nama **KafeNoPresensi.java**.
2. Buat fungsi **Menu** di dalam *class* tersebut.

```
● ● ●

1 public static void Menu() {
2     System.out.println("===== MENU RESTO KAFE =====");
3     System.out.println("1. Kopi Hitam - Rp 15,000");
4     System.out.println("2. Cappuccino - Rp 20,000");
5     System.out.println("3. Latte - Rp 22,000");
6     System.out.println("4. Teh Tarik - Rp 12,000");
7     System.out.println("5. Roti Bakar - Rp 10,000");
8     System.out.println("6. Mie Goreng - Rp 18,000");
9     System.out.println("=====");
10    System.out.println("Silakan pilih menu yang Anda inginkan.");
11 }
```

```
public class Kafe01 {
    public static void Menu(){
        System.out.println("===== MENU RESTO KAFE =====");
        System.out.println("1. Kopi Hitam - Rp 15,000");
        System.out.println("2. Cappuccino - Rp 20,000");
        System.out.println("3. Latte - Rp 22,000");
        System.out.println("4. Teh Tarik - Rp 12,000");
        System.out.println("5. Roti Bakar - Rp 10,000");
        System.out.println("6. Mie Goreng - Rp 18,000");
        System.out.println("=====");
        System.out.println("Silahkan pilih menu yang Anda inginkan.");
    }
}
```

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

```
● ● ●

1 public static void main(String[] args) {
2     Menu();
3 }
```

```
Run | Debug
public static void main(String[] args) [
    Menu(); ]
]
```

4. Commit program Anda ke Github dengan pesan “Percobaan 1”

Pertanyaan

1. Apakah fungsi tanpa parameter selalu harus bertipe void?

Tidak, karena tipe void hanya digunakan jika fungsi tersebut tidak mengembalikan nilai

2. Apakah daftar menu pada program kafe dapat ditampilkan tanpa menggunakan fungsi **Menu()**? Modifikasi kode program tersebut untuk dapat menampilkan daftar menu tanpa menggunakan fungsi!

Daftar menu pada program kafe dapat ditampilkan tanpa fungsi Menu() dengan menuliskannya kembali pada fungsi main

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

3. Jelaskan keuntungan menggunakan fungsi Menu() dibandingkan menulis semua perintah penampilan menu langsung di dalam fungsi main.

Jika menggunakan fungsi Menu(), program dapat disusun lebih terstruktur dan lebih efektif. Jika program dijalankan berulang-ulang, maka tidak perlu menuliskan ulang kodennya, hanya cukup memanggil fungsi yang sudah ada.

4. Uraikan secara singkat alur eksekusi program ketika fungsi Menu() dipanggil dari main (mulai dari program dijalankan sampai daftar menu tampil di layar).

Program mulai mencari fungsi main, dan menjalankan fungsi tersebut yang didalamnya terdapat baris Menu(), sehingga program berpindah ke fungsi Menu() dan menjalankannya semua perintah di dalam fungsi Menu(), kemudian kembali lagi ke fungsi main. Karena tidak ada perintah lain di fungsi main, program akan berhenti.

1.2 Percobaan 2: Membuat Fungsi Dengan Parameter

Waktu Percobaan: 40 menit

1. Buka kembali class **KafeNoPresensi.java** yang sudah dibuat pada Percobaan 1.
2. Ubah fungsi **Menu** dengan dua buah parameter bertipe *String* dan boolean.

```

● ● ●

1  public static void Menu(String namaPelanggan, boolean isMember) {
2      System.out.println("Selamat datang, " + namaPelanggan + "!");
3
4      if (isMember) {
5          System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
6      }
7
8      System.out.println("===== MENU RESTO KAFE =====");
9      System.out.println("1. Kopi Hitam - Rp 15,000");
10     System.out.println("2. Cappuccino - Rp 20,000");
11     System.out.println("3. Latte - Rp 22,000");
12     System.out.println("4. Teh Tarik - Rp 12,000");
13     System.out.println("5. Roti Bakar - Rp 10,000");
14     System.out.println("6. Mie Goreng - Rp 18,000");
15     System.out.println("=====");
16     System.out.println("Silakan pilih menu yang Anda inginkan.");
17 }

```

3. Eksekusi fungsi **Menu()** dari dalam fungsi *main*.

```

● ● ●

1  public static void main(String[] args) {
2      Menu("Andi", true);
3
4 }

```

```

public class Kafe01 {
    public static void Menu(String namaPelanggan, boolean isMember){
        System.out.println("Selamat datang, "+namaPelanggan+"!");
        if (isMember) {
            System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
        }
        System.out.println("===== MENU RESTO KAFE =====");
        System.out.println("1. Kopi Hitam - Rp 15,000");
        System.out.println("2. Cappuccino - Rp 20,000");
        System.out.println("3. Latte - Rp 22,000");
        System.out.println("4. Teh Tarik - Rp 12,000");
        System.out.println("5. Roti Bakar - Rp 10,000");
        System.out.println("6. Mie Goreng - Rp 18,000");
        System.out.println("=====");
        System.out.println("Silakan pilih menu yang Anda inginkan.");
    }

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

```

4. Commit program Anda ke Github dengan pesan “Percobaan 2”

Pertanyaan

- Apakah kegunaan parameter di dalam fungsi?
Sebagai tempat untuk data masukan yang akan diolah dalam fungsi
- Jelaskan mengapa pada percobaan ini fungsi **Menu()** menggunakan parameter **namaPelanggan** dan **isMember**?
Fungsi Menu() menggunakan parameter namaPelanggan untuk memanggil nama

pelanggan dan isMember untuk mengecek apakah pelanggan member atau bukan, jika tidak menggunakan parameter, fungsi tidak dapat membedakan pelanggan member atau bukan

3. Apakah parameter sama dengan variabel? Jelaskan.

Parameter tidak sama dengan variabel, karena parameter digunakan untuk menerima nilai dari luar, kalau variabel digunakan untuk menyimpan nilai didalam program mana saja

4. Jelaskan bagaimana cara kerja parameter **isMember** pada fungsi **Menu()**. Apa perbedaan output ketika **isMember** bernilai true dan ketika false?

Parameter isMember menerima nilai dari fungsi pemanggil , dan menjalankan baris pemilihan di dalam fungsi Menu(), jika isMember bernilai true maka baris di dalam pemilihan if akan dieksekusi namun jika bernilai false maka baris tersebut tidak akan dieksekusi

5. Apa yang akan terjadi jika memanggil fungsi Menu() tanpa menyertakan parameter **namaPelanggan** dan **isMember**?

Program tidak akan dijalankan karena parameter membutuhkan nilai dari pemanggil

6. Modifikasi kode di atas dengan menambahkan parameter baru **kodePromo (String)**. Jika kodePromo adalah "DISKON50", tampilkan berikan diskon 50%. Jika kodePromo adalah "DISKON30", tampilkan berikan diskon 30%. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid.

```

public class Kafe01 {
    public static void Menu(String namaPelanggan, boolean isMember, String kodePromo){
        System.out.println("Selamat datang, "+namaPelanggan+"!");
        if (isMember) {
            System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian!");
        }
        if (kodePromo.equals("DISKON50")) {
            System.out.println("Anda mendapatkan diskon 50%");
        }else if (kodePromo.equals("DISKON30")) {
            System.out.println("Anda mendapatkan diskon 30%");
        }else {
            System.out.println("Kode promo invalid");
        }
        System.out.println("===== MENU RESTO KAFE =====");
        System.out.println("1. Kopi Hitam - Rp 15,000");
        System.out.println("2. Cappuccino - Rp 20,000");
        System.out.println("3. Latte - Rp 22,000");
        System.out.println("4. Teh Tarik - Rp 12,000");
        System.out.println("5. Roti Bakar - Rp 10,000");
        System.out.println("6. Mie Goreng - Rp 18,000");
        System.out.println("=====");
        System.out.println("Silahkan pilih menu yang Anda inginkan.");
    }

    Run | Debug
    public static void main(String[] args) {
        Menu(namaPelanggan: "Andi", isMember: true, kodePromo: "DISKON50");
    }
}

```

7. Berdasarkan fungsi Menu() di atas, jika nama pelanggan adalah "Budi", pelanggan tersebut member, dan menggunakan kode promo "DISKON30", tuliskan satu baris perintah pemanggilan fungsi menu yang benar.

```

public static void main(String[] args) {
    Menu(namaPelanggan: "Andi", isMember: true, kodePromo: "DISKON50");
    Menu(namaPelanggan: "Budi", isMember: true, kodePromo: "DISKON30");
}

```

8. Menurut Anda, apakah penggunaan parameter `namaPelanggan` dan `isMember` pada fungsi Menu() membuat program lebih mudah dibaca dan dikembangkan dibandingkan jika nilai-nilai tersebut ditulis langsung di dalam fungsi tanpa parameter? Jelaskan alasan Anda.

Penggunaan parameter pada fungsi Menu() membuat program lebih mudah dibaca dan dikembangkan karena kode lebih terstruktur dan pemanggilannya hanya perlu menuliskan fungsi menu tersebut

9. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”**

1.3 Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

Waktu Percobaan: 40 menit

- Buka kembali `class KafeNoPresensi.java` yang sudah digunakan pada Percobaan 1 dan 2.
- Pada `class` yang sama buat fungsi `hitungTotalHargaNoPresensi` di dalam `class` tersebut yang mengembalikan nilai total harga dan parameter masukan pilihan Menu

dan banyaknya pesanan. Tambahkan daftar harga untuk setiap menu kafe dalam variabel berupa Array.



```
1 public static int hitungTotalHarga(int pilihanMenu, int banyakItem) {  
2     int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};  
3  
4     int hargaTotal = hargaItems[pilihanMenu - 1] * banyakItem;  
5     return hargaTotal;  
6 }
```

```
public static int hitungTotalHarga01 (int pilihanMenu, int banyakItem){  
    int [] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};  
    int hargaTotal = hargaItems[pilihanMenu-1] * banyakItem;  
    return hargaTotal;  
}
```

3. Modifikasi fungsi **main** di dalam *class* tersebut, dan eksekusi fungsi **Menu()** dari dalam fungsi **main** untuk menampilkan daftar menu kafe.
4. Selanjutnya modifikasi fungsi main sehingga program dapat meminta input dari pengguna untuk **pilihanMenu**, **banyakItem** (jumlah porsi yang dipesan). Fungsi main juga memanggil fungsi **hitungTotalHargaNoPresensi(pilihanMenbanyakItem)** dan menyimpan nilai kembalian (return value) ke dalam variabel **totalBayar**, lalu program menampilkan **totalBayar** ke layar.



```
1  
2 System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");  
3 int pilihanMenu = sc.nextInt();  
4 System.out.print("Masukkan jumlah item yang ingin dipesan: ");  
5 int banyakItem = sc.nextInt();  
6  
7 int totalHarga = hitungTotalHarga(pilihanMenu, banyakItem);  
8  
9 System.out.println("Total harga untuk pesanan Anda: Rp" + totalHarga);
```

```

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    Menu(namaPelanggan: "Budi", isMember: true, kodePromo: "DISKON30");
    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 = hitungTotalHarga01(pilihanMenu, banyakItem);
    System.out.println("Total harga untuk pesanan anda: Rp"+totalHarga);
}

```

5. Jalankan program dan amati apakah nilai total yang ditampilkan sudah sesuai dengan harga dan jumlah pesanan.
6. Commit program Anda ke Github dengan pesan “Percobaan 3”

Pertanyaan

1. Jelaskan secara singkat kapan suatu fungsi membutuhkan nilai kembalian (*return value*) dan kapan fungsi tidak perlu mengembalikan nilai. Berikan minimal satu contoh dari program kafe pada Percobaan 3 untuk masing-masing kasus.

Fungsi membutuhkan nilai kembalian ketika fungsi tersebut menggunakan tipe data primitif, dan fungsi tersebut tidak perlu mengembalikan nilai ketika menggunakan tipe data referensi.

Contoh yang membutuhkan nilai kembalian yaitu pada fungsi `hitungTotalHarga01()` yang menggunakan tipe data int.

Dan contoh fungsi yang tidak perlu mengembalikan nilai yaitu pada fungsi `Menu()`.

2. Fungsi **hitungTotalHargaNoPresensi** saat ini mengembalikan total harga berdasarkan `pilihanMenu` dan `jumlahPesanan`. Sebutkan tipe data nilai kembalian dan dua buah parameter yang digunakan fungsi tersebut. Jelaskan arti masing-masing parameter dalam konteks program kafe.

Tipe data nilai kembalian yang digunakan dalam fungsi tersebut adalah tipe primitif yaitu int.

Parameter 1 = `pilihanMenu` : digunakan untuk menyimpan nomor menu yang dipilih pelanggan

Parameter 2 = `jumlahPesanan` : digunakan untuk menyimpan jumlah item yang dipesan pada `pilihanMenu`.

3. Modifikasi kode di atas sehingga fungsi **hitungTotalHargaNoPresensi** dapat menerima `kodePromo`. Jika `kodePromo` adalah "DISKON50", maka mendapat diskon 50% dari `totalHarga` dan tampilkan diskon. Jika `kodePromo` adalah "DISKON30", maka mendapat diskon 30% dari `totalHarga` dan tampilkan diskon. Jika tidak ada kode promo yang berlaku, tampilkan kode invalid dan tidak ada pengurangan total harga `totalHarga`.

```
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s: "Masukkan kode promo : ");
    String kodePromo=sc.nextLine();
    Menu(namaPelanggan: "Budi", isMember: true, kodePromo);
    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 = hitungTotalHarga01(pilihanMenu, banyakItem, kodePromo);
    System.out.println("Total harga untuk pesanan anda: Rp"+totalHarga);
}
}
```

4. Modifikasi kode di atas sehingga bisa memilih beberapa jenis menu berbeda serta menampilkan total keseluruhan pesanan. Bagaimana memodifikasi program sehingga pengguna dapat: memesan **lebih dari satu jenis menu** (misalnya menu 1 dan 3 sekaligus), dan menampilkan **total keseluruhan** pesanan (gabungan dari semua jenis menu)?

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print(s: "Masukkan kode promo : ");
    String kodePromo=sc.nextLine();
    Menu(namaPelanggan: "Budi", isMember: true, kodePromo);
    String pesanlagi;
    int totalKeseluruhan = 0;
    do {
        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();
        sc.nextLine();
        int totalHarga = hitungTotalHarga01(pilihanMenu, banyakItem, kodePromo);
        System.out.println("Total harga untuk pesanan anda: Rp"+totalHarga);
        System.out.print(s: "Apakah anda ingin memesan lagi? (Ya/Tidak) : ");
        pesanlagi=sc.nextLine();
        totalKeseluruhan += totalHarga;
    } while (pesanlagi.equalsIgnoreCase("Ya"));
    System.out.println("\nTotal keseluruhan untuk pesanan anda : Rp"+totalKeseluruhan);
}
```

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”

1.4 Percobaan 4: Fungsi Varargs

Waktu Percobaan: 40 menit

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

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

```
public class PengunjungCafe01 {
    static void daftarPengunjung(String... namaPengunjung){
        System.out.println("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*.

```
daftarPengunjung("Ali", "Budi", "Citra");
```

```
public class PengunjungCafe01 {
    Run | Debug
    public static void main(String[] args) {
        daftarPengunjung(...namaPengunjung: "Ali", "Budi", "Citra");
    }
}
```

4. Jalankan program dan amati hasilnya

```
Daftar nama pengunjung:
- Ali
- Budi
- Citra
PS C:\PENYIMPANAN\Documents\G\PraktikumDaspro>
```

5. Tambahkan dua baris berikut pada fungsi *main* di bawah kode pada nomor 3 di atas:

```
daftarPengunjung ("Andi");
daftarPengunjung ("Doni", "Eti", "Fahmi", "Galih");
```

```
public class PengunjungCafe01 {
    Run | Debug
    public static void main(String[] args) {
        daftarPengunjung(...namaPengunjung: "Ali", "Budi", "Citra");
        daftarPengunjung(...namaPengunjung: "Andi");
        daftarPengunjung(...namaPengunjung: "Doni", "Eti", "Fahmi", "Galih");
    }
}
```

6. Jalankan program. Amati perbedaan output ketika jumlah argumen yang dikirim berbeda-beda.

```
Daftar nama pengunjung:  
- Ali  
- Budi  
- Citra  
Daftar nama pengunjung:  
- Andi  
Daftar nama pengunjung:  
- Doni  
- Eti  
- Fahmi  
- Galih  
PS C:\ PENYIMPANAN\Documents\G\PraktikumDaspro>
```

7. Commit program Anda ke Github dengan pesan “Percobaan 4”

Pertanyaan

1. Jelaskan mengapa penulisan parameter di praktikum 4 ditulis dengan **String... namaPengunjung!**

Penulisan parameter ditulis dengan String... namaPengunjung, karena jumlah dalam parameter tersebut belum diketahui.

2. Modifikasi method **daftarPengunjung** menggunakan **for-each** loop.

```
static void daftarPengunjung(String... namaPengunjung){  
    System.out.println("Daftar nama pengunjung:");  
    for (String i : namaPengunjung) {  
        System.out.println("- "+i);  
    }  
}
```

3. Bisakah menggunakan dua tipe data varargs dalam satu fungsi? Jelaskan jawaban Anda berdasarkan aturan varargs di Java, dan berikan contohnya!

Penggunaan dua tipe data varargs dalam satu fungsi tidak dapat dilakukan karena setiap fungsi hanya diperbolehkan memiliki 1 variabel arguments dan variabel tersebut diletakkan di akhir jika memiliki lebih dari 1 parameter.

```
daftarPengunjung(String hari, String... namaPengunjung){
```

4. Jelaskan apa yang terjadi jika fungsi **daftarPengunjung** dipanggil tanpa argumen. Apakah program akan error saat kompilasi, error saat dijalankan, atau tetap berjalan? Jika tetap berjalan, bagaimana output yang dihasilkan?

Jika fungsi daftarPengunjung dipanggil tanpa argumen, maka program akan tetap berjalan namun hanya menampilkan output nama pengunjung

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 4”

2. Tugas

Waktu Pengerjaan: 100 menit

- Buatlah sebuah **class KubusNoPresensi** yang di dalamnya terdapat fungsi untuk menghitung volume kubus dan luar permukaan kubus! **Commit dan push program Anda ke Github dengan pesan “Tugas1”**

```
public class Kubus01 {  
    Run | Debug  
    public static void main(String[] args) {  
        hitungVolume(sisi: 3);  
        hitungLuas(sisi: 3);  
    }  
  
    public static int hitungVolume(int sisi){  
        int volume = sisi * sisi * sisi;  
        System.out.println("Volume kubus : "+volume);  
        return volume;  
    }  
  
    public static int hitungLuas(int sisi){  
        int luas = 6 * (sisi*sisi);  
        System.out.println("Luas kubus : "+luas);  
        return luas;  
    }  
}
```

- Di suatu kelas, terdapat **N** mahasiswa yang mengikuti satu mata kuliah. Nilai tiap mahasiswa untuk mata kuliah tersebut akan diinputkan oleh pengguna dan disimpan dalam sebuah array 1 dimensi berukuran **N**. Buatlah sebuah class **NilaiMahasiswaNoPresensi** yang di dalamnya terdapat beberapa fungsi berikut:
 - Fungsi **isianArray** untuk:
 - Mengisi elemen-elemen array dengan nilai yang diinputkan oleh user (jumlah elemen = N, dibaca dari input)
 - Fungsi **tampilArray** untuk:
 - Menerima parameter berupa array nilai mahasiswa
 - Menampilkan seluruh nilai yang telah diinputkan
 - Fungsi **hitTot** untuk:
 - Menerima parameter berupa array nilai mahasiswa
 - Menghitung dan mengembalikan total nilai seluruh mahasiswa (return tipe int atau double)
 - Fungsi **main** untuk:
 - Membaca nilai **N** (jumlah mahasiswa) dari input pengguna
 - Membuat array dengan ukuran **N**
 - Memanggil fungsi **isianArray**, **tampilArray**, dan **hitTot**
 - Menampilkan total nilai yang dikembalikan oleh fungsi **hitTot**

Commit dan push program Anda ke Github dengan pesan “Tugas2”

```

public class NilaiMahasiswa01 {

    static void isianArray(int[] nilaiMahasiswa){
        for (int i = 0; i < nilaiMahasiswa.length; i++) {
            System.out.print("Masukkan nilai mahasiswa ke-" +(i+1)+ " : ");
            nilaiMahasiswa[i] = sc.nextInt();
        }
    }

    static void tampilArray(int[] nilaiMahasiswa){
        System.out.println("Data nilai mahasiswa : ");
        for (int dataNilai : nilaiMahasiswa) {
            System.out.println(dataNilai);
        }
    }

    static int hitTot(int[] nilaiMahasiswa){
        int total = 0;
        for (int totalNilai : nilaiMahasiswa) {
            total += totalNilai;
        }
        return total;
    }

    Run | Debug
    public static void main(String[] args) {
        System.out.print("Masukkan jumlah mahasiswa : ");
        int N = sc.nextInt();
        int [] nilaiMahasiswa = new int[N];
        isianArray(nilaiMahasiswa);
        tampilArray(nilaiMahasiswa);
        int hitTot = hitTot(nilaiMahasiswa);
        System.out.println("Total nilai seluruh mahasiswa : "+hitTot);
    }
}

```

3. Ibu Mariana mengelola sebuah kafe. Berikut adalah rekap penjualan 5 menu dari hari pertama hingga ketujuh:

	Hari ke 1	Hari ke 2	Hari ke 3	Hari ke 4	Hari ke 5	Hari ke 6	Hari ke 7
Kopi	20	20	25	20	10	60	10
Teh	30	80	40	10	15	20	25
Es Kelapa Muda	5	9	20	25	10	5	45
Roti Bakar	50	8	17	18	10	30	6
Gorengan	15	10	16	15	10	10	55

- a. Buatlah sebuah class **RekapPenjualanCafeNoPresensi** yang digunakan untuk menyimpan dan mengolah data penjualan 5 menu kafe dari hari pertama hingga hari ketujuh. Di dalam class tersebut, buat dan gunakan beberapa fungsi (method) berikut:
- Fungsi untuk menginputkan data penjualan setiap menu untuk setiap hari (petunjuk: menggunakan array 2 dimensi).

- Fungsi untuk menampilkan seluruh data penjualan dari hari pertama hingga hari terakhir dalam bentuk tabel
- Fungsi untuk menampilkan menu yang memiliki total penjualan tertinggi selama tujuh hari (tampilkan nama menu dan total penjualannya).
- Fungsi untuk menampilkan rata-rata penjualan untuk setiap menu selama tujuh hari (tampilkan nama menu dan nilai rata-ratanya).

Gunakan fungsi main untuk:

- Memanggil fungsi input data penjualan,
- Menampilkan seluruh data penjualan,
- Menampilkan menu dengan penjualan tertinggi, dan
- Menampilkan rata-rata penjualan untuk setiap menu.

- b. Modifikasi program pada 3 (a) sehingga jumlah hari penjualan dan jumlah menu dibaca dari input pengguna, lalu sesuaikan ukuran array dan proses perhitungan agar tetap berjalan dengan benar.

Commit dan push program Anda ke Github dengan pesan “Tugas3”

```
 1 package jobsheet1;
 2 import java.util.Scanner;
 3 public class RekapPenjualanCafe01 {
 4     static Scanner sc = new Scanner(System.in);
 5
 6     static void inputData(int[][] cafe, String[] menu){
 7         System.out.println("Masukkan data penjual menu");
 8         for (int i = 0; i < cafe.length; i++) {
 9             System.out.print("Menu : "+menu[i]);
10             for (int j = 0; j < cafe[i].length; j++) {
11                 System.out.print("Hari ke "+(j+1)+" : ");
12                 cafe[i][j]=sc.nextInt();
13             }
14         }
15     }
16
17     static void tampilData(int[][] cafe,int jumlahHari, String[] menu){
18         System.out.println("\nRekap penjualan menu :");
19         System.out.print("\t");
20         for (int h = 0; h < jumlahHari; h++) {
21             System.out.print("\tHari ke "+(h+1));
22         }
23         for (int i = 0; i < cafe.length; i++) {
24             System.out.print("\n"+menu[i]+"\t");
25             if (menu[i].length() < 5 ) {
26                 System.out.print("\t");
27             }
28             for (int j = 0; j < cafe[i].length; j++) {
29                 System.out.print(cafe[i][j]+"\t\t");
30             }
31             System.out.println();
32         }
33     }
34
35     static int[] menuTerlaris(int[][] cafe, String[] menu){
36         int terlaris = 0;
37         String menuTerlaris = "";
38         int[] totalPenjualanMenu = new int[menu.length];
39         for (int i = 0; i < totalPenjualanMenu.length; i++) {
40             for (int j = 0; j < cafe[i].length; j++) {
41                 totalPenjualanMenu[i] += cafe[i][j];
42             }
43             if (totalPenjualanMenu[i] >= terlaris) {
44                 terlaris = totalPenjualanMenu[i];
45                 menuTerlaris = menu[i];
46             }
47         }
48         System.out.println("\nMenu "+menuTerlaris+ " memiliki total penjualan tertinggi : "+terlaris+ " menu terjual");
49         return totalPenjualanMenu;
50     }
51
52     static void hitungRata(int[] totalPenjualanMenu,int jumlahHari , String[] menu){
53         int[] rataPerMenu = new int[menu.length];
54         System.out.println("\nRata-rata Penjualan Tiap Menu");
55         System.out.println("Nama Menu\tRata-rata");
56         for (int i = 0; i < rataPerMenu.length; i++) {
57             rataPerMenu[i] = totalPenjualanMenu[i] / jumlahHari;
58             System.out.print((i+1)+". "+menu[i]+"\t");
59             if (menu[i].length() < 5 ) {
60                 System.out.print("\t\t");
61             }else if (menu[i].length() < 11) {
62                 System.out.print("\t");
63             }
64             System.out.println(rataPerMenu[i]);
65         }
66     }
67
68     public static void main(String[] args) {
69         System.out.print("Masukkan jumlah menu : ");
70         int jumlahMenu = sc.nextInt();
71         sc.nextLine();
72         System.out.print("Masukkan jumlah hari : ");
73         int jumlahHari = sc.nextInt();
74         sc.nextLine();
75         String[] menu = new String[jumlahMenu];
76         for (int i = 0; i < menu.length; i++) {
77             System.out.print("Masukkan nama menu ke- "+(i+1)+" : ");
78             menu[i]=sc.nextLine();
79         }
80         int[][] cafe = new int[jumlahMenu][jumlahHari];
81         inputData(cafe, menu);
82         tampilData(cafe, jumlahHari, menu);
83         int[] totalPenjualanMenu = menuTerlaris(cafe, menu);
84         hitungRata(totalPenjualanMenu, jumlahHari, menu);
85     }
86 }
```