

PRAKTIKUM DASPRO WEEK 13

FUNGSI 1



STEVAN ZAKY SETYANTO

2341720101

D-IV TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023

1. Tujuan

1. Mahasiswa mampu memahami penggunaan fungsi static pada Java dengan parameter dan mengembalikan nilai.
2. Mahasiswa mampu membuat program menggunakan fungsi static dan mengeksekusi fungsi tersebut.

2. Praktikum

2.1 Percobaan 1: Membuat Fungsi Tanpa Parameter

Waktu Percobaan: 40 menit

1. Buat class baru, dan simpan file tersebut dengan nama TerimaKasih_NoAbsen.java.
2. Buat fungsi UcapanTerimaKasih di dalam class tersebut.
3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi UcapanTerimaKasih dari dalam fungsi main.

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > TerimaKasih_29.java > ...
1 public class TerimaKasih_29{
2     public static void UcapanTerimaKasih(){
3         System.out.println("Thank you for being the best teacher in the world.\n"+
4             "You inspired in me a lover for learning and made me feel like I could ask you anything. ");
5     }
6     Run | Debug
7     public static void main(String[] args){
8         UcapanTerimaKasih();
9     }
10 }
11
12
13
14
```

Pertanyaan!

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi "void" saja?
2. Apakah bisa kalimat "Thank you for.....dst" dituliskan tanpa fungsi UcapanTerimaKasih? modifikasi kode program tersebut untuk dapat menampilkan suatu kalimat tanpa dan dengan menggunakan fungsi!
3. Apakah keuntungan menggunakan fungsi di dalam program?

Jawaban

1. Tidak, fungsi tanpa parameter tidak selalu harus berjenis "void". Fungsi tanpa parameter dapat memiliki tipe data pengembalian (return type) selain "void".
2. Bisa (tanpa fungsi)

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > TerimaKasih_29.java > TerimaKasih_29
1 public class TerimaKasih_29{
2
3     Run | Debug
4     public static void main(String[] args){
5         System.out.println("Thank you for being the best teacher in the world.\n"+
6             "You inspired in me a lover for learning and made me feel like I could ask you anything. ");
7
8     }
9
10
11
12
```

Dengan fungsi

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > TerimaKasih_29.java > TerimaKasih_29 > main(String[])
1 public class TerimaKasih_29{
2     public static void UcapanTerimaKasih(){
3         System.out.println("Thank you for being the best teacher in the world.\n"+
4             "You inspired in me a lover for learning and made me feel like I could ask you anything. ");
5     }
6     Run | Debug
7     public static void main(String[] args){
8         UcapanTerimaKasih();
9     }
10 }
11
12
13
14
```

3. Memecah program menjadi bagian-bagian untuk kejelasan dan pemeliharaan, kemampuan untuk menggunakan kembali blok kode di tempat lain, menyediakan tingkat abstraksi untuk memahami tugas yang dilakukan, mempermudah perbaikan dan penyesuaian kode di masa depan.

2.2 Percobaan 2: Membuat Fungsi Dengan Parameter

Waktu Percobaan: 40 menit

1. Buat fungsi UcapanTambahkan dengan sebuah parameter bertipe String di dalam class TerimaKasih_NoAbsen!
2. Eksekusi fungsi UcapanTambahkan dari dalam fungsi main.

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > TerimaKasih_29.java > TerimaKasih_29 > main(String[])
1 public class TerimaKasih_29{
2     public static void UcapanTerimaKasih(){
3         System.out.println("Thank you for being the best teacher in the world.\n"+
4             "You inspired in me a lover for learning and made me feel like I could ask you anything. ");
5     }
6
7     public static void UcapanTambahkan(String ucapan){
8         System.out.println(ucapan);
9     }
10     Run | Debug
11     public static void main(String[] args){
12         UcapanTerimaKasih();
13         String ucapan = "Terima kasih Pak.. Bu.. Semoga Sehat Sejalalu";
14         UcapanTambahkan(ucapan);
15     }
16 }
17
18
19
20
```

Pertanyaan!

1. Apakah kegunaan suatu parameter di dalam fungsi?
2. Apakah parameter sama dengan variabel? jelaskan!
3. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewati data input saja? Bagaimana dengan data output?

Jawab

1. Parameter memungkinkan mengirim data ke dalam fungsi. Dengan menentukan parameter saat mendefinisikan fungsi (seperti `UcapanTambah(String ucapan)`), memberikan cara bagi fungsi untuk menerima dan menggunakan nilai yang dikirim ketika fungsi dipanggil.
2. Parameter: Variabel yang digunakan dalam deklarasi fungsi untuk menerima nilai dari pemanggil fungsi.
Variabel: Tempat penyimpanan nilai dalam suatu program, dapat berubah nilainya selama eksekusi program.
Dalam konteks fungsi, parameter adalah variabel khusus yang digunakan untuk menerima nilai dari luar fungsi saat dipanggil.
3. Dalam bahasa pemrograman Java, parameter digunakan untuk melewatkan data input ke dalam suatu fungsi atau metode. Namun, data output dapat dikembalikan dari fungsi menggunakan kata kunci `return`. Jadi, sementara parameter memungkinkan untuk mengirim data ke dalam fungsi, `return` memungkinkan fungsi untuk mengembalikan nilai sebagai hasil eksekusi.

2.3 Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

Waktu Percobaan: 40 menit

1. Buat class baru, dan menyimpan file tersebut dengan nama `Ucapan_NoAbsen.java`.
2. Buat fungsi `PenerimaUcapan` di dalam class tersebut yang mengembalikan nilai `namaOrang` (String) dan parameter masukan nama (String).
3. Buat fungsi `main` di dalam class tersebut, dan eksekusi fungsi `PenerimaUcapan` dari dalam fungsi `main`.

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J Ucapan_29.java > Ucapan_29
1  import java.util.Scanner;
2  public class Ucapan_29 {
3
4      public static String PenerimaUcapan(){
5          Scanner sc = new Scanner(System.in);
6          System.out.println("Tuliskan NAMA orang yang ingin Anda beri ucapan:");
7          String namaOrang = sc.nextLine();
8          sc.close();
9          return namaOrang;
10
11     }
12     Run | Debug
13     public static void main(String[] args) {
14         String nama = PenerimaUcapan();
15         System.out.println("Thank you "+nama+"\nMay the force be with you.");
16     }
17 }
```

Pertanyaan!

1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (`return`)!
2. Apakah bisa ditambahkan statement `System.out.println` di dalam fungsi dengan nilai kembalian? Apa pengaruh nya?
3. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi `main` dipanggil tanpa dilewatkan ke variabel? seperti pada percobaan 1? Jelaskan!

Jawab

1. Ketika tidak menggunakan `void`
2. Bisa , namun hanya akan mencetak nilai yang dimasukkan oleh pengguna ke konsol, tetapi nilai kembalian fungsi tetap berupa `namaOrang`.
3. Ya ,, karena fungsi tanpa nilai kembalian atau dengan `void` tidak perlu `return`

2.4 Percobaan 4: Fungsi yang dapat meng-Calling Fungsi Lain

Waktu Percobaan: 50 menit

1. Buat class baru, simpan file tersebut dengan nama nama UcapanTerimaKasih_NoAbsen.java. Pada class ini, kita akan mencoba menggabungkan fungsi-fungsi yang sudah kita buat sebelumnya pada class TerimaKasih dan Ucapan.
2. Buatlah fungsi PenerimaUcapan, dimana pada class tersebut akan mengembalikan nilai namaOrang (String) dan parameter masukan nama (String).
3. Buatlah fungsi UcapanTerimaKasih di dalam class tersebut yang memanggil fungsi PenerimaUcapan
4. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi UcapanTerimaKasih dari dalam fungsi main.

```
D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J UcapanTerimaKasih_29.java > U UcapanTerimaKasih_29 > U UcapanTerimaKasih()
1  import java.util.Scanner;
2  public class UcapanTerimaKasih_29{
3
4      public static String PenerimaUcapan(){
5          Scanner sc = new Scanner(System.in);
6          System.out.println(x:"Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
7          String namaOrang = sc.nextLine();
8          sc.close();
9          return namaOrang;
10     }
11
12     void UcapanTerimaKasih_29.UcapanTerimaKasih()
13
14     public static void UcapanTerimaKasih(){
15         String nama = PenerimaUcapan();
16         System.out.println("Thank you "+nama+" for being the best teacher in the world.\n"+
17             "You inspired in me a love for learning and made me feel like I could ask you anything.");
18     }
19
20     public static void main(String[] args) {
21         UcapanTerimaKasih();
22     }
23 }
```

Pertanyaan!

1. Berdasarkan uji coba pada percobaan 4, manakah fungsi yang akan eksekusi pertama kali? Jelaskan!
2. Manakah tata cara penulisan fungsi benar di dalam class? Di atas fungsi main ataukah dibawah fungsi main? Jelaskan!
3. Modifikasi program di atas dengan menambahkan fungsi UcapanTambahan() dengan input parameter String. Fungsi UcapanTambahan() berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan.

Jawab

1. Fungsi PenerimaUcapan karena fungsi tersebut berada pada awal program dan fungsi tersebut mengharuskan menginputkan nama yang akan dibutuhkan di fungsi ke 2
2. Tidak ada keharusan yang terpenting masih di dalam class dan di luar void main

```

D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J UcapanTerimaKasih_29.java > UcapanTerimaKasih_29
1  import java.util.Scanner;
2  public class UcapanTerimaKasih_29{
3
4      public static String PenerimaUcapan(){
5          Scanner sc = new Scanner(System.in);
6          System.out.println(x:"Tuliskan NAMA orang yang ingin Anda beri ucapan: ");
7          String namaOrang = sc.nextLine();
8          sc.close();
9          return namaOrang;
10     }
11
12     public static String UcapanTerimaKasih(){
13         String nama = PenerimaUcapan();
14
15         return "Thank you "+nama+" for being the best teacher in the world.\n"+
16             "You inspired in me a love for learning and made me feel like I could ask you anything.";
17     }
18     public static void UcapanTambahn(String ucapan){
19         String tambahan = UcapanTerimaKasih();
20
21         System.out.println(tambahan +"\n"+ucapan);
22     }
23
24     Run | Debug
25     public static void main(String[] args) {
26         String ucapan = "I love forever";
27         UcapanTambahn(ucapan);
28     }
29 }

```

3.

2.5 Percobaan 5: Fungsi Varargs

Waktu Percobaan: 40 menit

1. Buat class baru, simpan file tersebut dengan nama nama Percobaan5NoAbsen.java.
2. Buatlah fungsi Tampil (bertipe void) di dalam class tersebut dengan menggunakan dua jenis tipe data parameter yaitu string dan int
3. Buat fungsi main di dalam class tersebut, dan eksekusi fungsi Tampil dari dalam fungsi main.

```

D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J Percobaan5_29.java > Percobaan5_29 > main(String[])
1  public class Percobaan5_29 {
2
3      static void Tampil(String str, int... a) {
4          System.out.println("String: " + str);
5          System.out.println("Jumlah argumen/parameter: "+a.length);
6
7          for(int i : a){
8              System.out.println(i + " ");
9          }
10         System.out.println();
11     }
12
13     Run | Debug void Percobaan5_29.main(String[] args)
14     public static void main(String[] args) {
15         Tampil(str:"Daspro 2019", ...a:100, 200);
16         Tampil(str:"Teknologi Informasi",...a:1,2,3,4,5);
17         Tampil(str:"Polinema");
18     }
19 }

```

Pertanyaan!

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan int... a!
2. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata! (minimal 3)
3. Bisakah kita menggunakan dua tipe data varaargs dalam satu fungsi?Berikan contohnya!

Jawab

1. Dengan menggunakan varargs, metode bisa menangani situasi di mana jumlah argumen dapat berubah-ubah tanpa harus menulis ulang metode dengan parameter yang berbeda. Ini membuat kode lebih ringkas dan lebih mudah digunakan, terutama dalam kasus-kasus di mana kita tidak tahu pasti berapa banyak argumen yang akan kita berikan.
2.
 1. Pembuatan Daftar Belanja
 2. Pengelolaan Catatan Harian
 3. Penghitungan Pengeluaran Bulanan
3. Tidak, dalam satu fungsi atau metode, Anda tidak dapat menggunakan dua tipe data varargs yang berbeda. Varargs harus memiliki tipe data yang konsisten dalam satu varargs.

2.6 Percobaan 6: Pembuatan Kode Program, Dengan Fungsi versus Tanpa Fungsi

Waktu Percobaan: 50 menit

```

D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J Percobaan629.java > 🚀 Percobaan629
1  import java.util.Scanner;
2  public class Percobaan629 {
3
4      Run | Debug
5      public static void main(String[] args) {
6          Scanner input = new Scanner(System.in);
7          int p,l,t,l,vol;
8
9          System.out.println(x:"Masukkan panjang = ");
10         p = input.nextInt();
11         System.out.println(x:"Masukkan lebar = ");
12         l = input.nextInt();
13         System.out.println(x:"Masukkan tinggi = ");
14         t = input.nextInt();
15
16         L=hitungLuas(p, l);
17         System.out.println("Luas persegi panjang adalah = "+L);
18         vol = hitungVolume(t, p, l);
19         System.out.println("Volume balok adalah = "+vol);
20     }
21     static int hitungLuas (int pjg,int lb){
22         int Luas = pjg * lb;
23         return Luas;
24     }
25     static int hitungVolume (int tinggi, int a, int b){
26         int volume = hitungLuas(a,b)*tinggi;
27         return volume;
28     }
29 }

```

Pertanyaan!

1. Sebutkan tahapan dan urutan eksekusi pada percobaan 6 di atas!
2. Apakah output dari program dibawah ini kemudian jelaskan alur jalannya program

```

D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J programKu29.java > programKu29
1  public class programKu29 {
2      public static void TampilHinggaKei(int i){
3          for (int j = 1; j <= i; j++){
4              System.out.println(j);
5          }
6      }
7      public static int Jumlah(int bil1, int bil2){
8          return (bil1 + bil2);
9      }
10     }
11     public static void TampilJumlah (int bil1, int bil2){
12         TampilHinggaKei(Jumlah(bil1, bil2));
13     }
14     Run | Debug
15     public static void main(String[] args) {
16         int temp = Jumlah(bil1:1, bil2:1);
17         TampilJumlah(temp, bil2:5);
18     }
19
20
21 }

```

tersebut!

3. Pada saat apakah fungsi yang kita buat harus menggunakan parameter atau tidak? Pada saat apakah fungsi yang kita buat harus memiliki nilai kembalian atau tidak? Jelaskan!

Jawab

1. Deklarasi Variabel:

Variabel p, l, t, L, dan vol dideklarasikan untuk menyimpan panjang, lebar, tinggi, luas, dan volume.

Inisialisasi Scanner:

Scanner diinisialisasi untuk menerima input dari pengguna.

Input Panjang, Lebar, dan Tinggi:

Program meminta pengguna untuk memasukkan nilai panjang, lebar, dan tinggi menggunakan Scanner.

Pemanggilan hitungLuas:

Nilai panjang (p) dan lebar (l) yang dimasukkan oleh pengguna digunakan sebagai argumen untuk memanggil fungsi hitungLuas.

Fungsi hitungLuas mengembalikan nilai luas (L), yang kemudian disimpan dalam variabel L.

Output Luas Persegi Panjang:

Program mencetak nilai luas (L) ke layar.

Pemanggilan hitungVolume:

Nilai tinggi (t), panjang (p), dan lebar (l) yang dimasukkan oleh pengguna digunakan sebagai argumen untuk memanggil fungsi hitungVolume.

Fungsi hitungVolume mengembalikan nilai volume (vol), yang kemudian disimpan dalam variabel vol.

Output Volume Balok:

Program mencetak nilai volume (vol) ke layar.

Selesai:

Program selesai dieksekusi.


```

1
2
3
4
5
6
7

```

2. Program dimulai dengan eksekusi fungsi 'main()'. Di dalam fungsi 'main()', variable 'temp' diinisialisasi dengan hasil dari pemanggilan fungsi 'Jumlah(1,1)' yaitu 2. Selanjutnya, fungsi 'TampilJumlah(temp,5)' dipanggil. Di dalam fungsi 'TampilJumlah()', dilakukan pemanggilan fungsi 'Jumlah(bil1, bil2)' dengan parameter 'temp' dan '5' yang mengembalikan nilai 7. Hasil dari pemanggilan 'Jumlah (bil1, bil2)' digunakan sebagai parameter untuk memanggil fungsi 'TampilHinggakei()'. Di dalam fungsi 'TampilHinggakei()' dilakukan perulangan untuk menampilkan angka dari 1 hingga 7.

3. Kesimpulan alurnya adalah memanggil fungsi 'Jumlah()' untuk menjumlahkan dua bilangan, kemudian hasil penjumlahan tersebut digunakan sebagai parameter untuk menampilkan angka dari 1 hingga hasil penjumlahan tersebut.

4. Penggunaan Parameter:

Mengirim Informasi:

Gunakan parameter saat fungsi Anda membutuhkan informasi untuk menjalankan tugas tertentu. Parameter memungkinkan Anda mengirimkan nilai ke dalam fungsi, sehingga fungsi dapat bekerja dengan data yang berbeda setiap kali dipanggil.

Dapat Digunakan Kembali:

Fungsi dengan parameter seringkali lebih dapat digunakan kembali karena dapat diterapkan pada berbagai input.

Penggunaan Nilai Kembalian:

Keluaran:

Gunakan nilai kembalian saat fungsi Anda perlu memberikan hasil atau keluaran. Ini sangat penting ketika fungsi melakukan perhitungan atau transformasi.

Aliran Data:

Nilai kembalian memungkinkan aliran data antara bagian-bagian berbeda dari program Anda.

Hasil panggilan satu fungsi dapat digunakan sebagai input untuk fungsi lain.

Penanganan Kesalahan:

Fungsi dapat mengembalikan nilai khusus atau kode untuk menunjukkan keberhasilan atau kegagalan, memungkinkan penanganan kesalahan yang lebih baik.

Keterbacaan:

Fungsi dengan nilai kembalian sering menghasilkan kode yang lebih mudah dibaca dan dimengerti. Hasil panggilan fungsi dapat digunakan langsung dalam ekspresi.

3. Tugas

Waktu Pengerjaan: 100 menit

1. Buatlah sebuah class KubusNoAbsen yang di dalamnya terdapat fungsi untuk menghitung volume kubus dan luar permukaan kubus!

```

D: > SIAKAD LMS SEMESTER 1 > Praktik Dasar Pemrograman > Week 13 > J Kubus29.java > Kubus29 > hitungPermukaan(int)
1  import java.util.Scanner;
2  public class Kubus29 {
3      Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int sisi, v, per;
7          System.out.println(x:"Masukkan nilai sisi: ");
8          sisi = sc.nextInt();
9
10         v = hitungVolume(sisi);
11         per = hitungPermukaan(sisi);
12         System.out.println("Hasil volume: "+v);
13         System.out.println("Hasil permukaan: "+per);
14
15     }
16     static int hitungVolume(int a){
17         int vol = a * a * a;
18         return vol;
19     }
20     static int hitungPermukaan(int b){
21         int permukaan = 6 * (b*b);
22         return permukaan;
23     }
24 }
25
26
27
28 }

```

2. Buatlah program untuk mengisi array 2 dimensi nilai tugas lima mahasiswa sesuai dengan studi kasus pada tugas pada materi teori sebagai berikut :

Tambahkan fungsi untuk mengambil informasi dari data diatas dengan rincian sebagai berikut :

- Fungsi untuk meninputkan data nilai mahasiswa
- Fungsi untuk menampilkan seluruh nilai mahasiswa mulai dari minggu pertama sampai ketujuh
- Fungsi untuk mencari pada hari keberapakah terdapat nilai tertinggi dibanding hari lain dari keseluruhan mahasiswa
- Fungsi untuk menampilkan mahasiswa yang memiliki nilai tertinggi (tampilkan pula keterangan nilai dari minggu ke-)

jawab

```

static void nilaiMhs() {
    nilaiSiswa = new int[5][7];
    for (int i = 0; i < nilaiSiswa.length; i++) {
        for (int j = 0; j < nilaiSiswa[i].length; j++) {
            System.out.print("Masukkan Nilai " + namaSiswa[i] + " Di minggu ke- " + (j + 1) + ": ");
            nilaiSiswa[i][j] = sc.nextInt();
        }
    }
}

a.
System.out.println(x:"\nNama   | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7");
for (int i = 0; i < nilaiSiswa.length; i++) {
    System.out.printf(format:"%-5s | %6d | %6d | %6d | %6d | %6d | %6d | %6d\n",
        namaSiswa[i], nilaiSiswa[i][0], nilaiSiswa[i][1], nilaiSiswa[i][2], nilaiSiswa[i][3], nilaiSiswa[i][4], nilaiSiswa[i][5], nilaiSiswa[i][6]);
}

b.

```

```

static int cariHariMaks() {
    if (nilaiSiswa.length == 0 || nilaiSiswa[0].length == 0) {
        // Menghandle kasus dimana array kosong
        return -1; // atau nilai yang sesuai
    }

    int maxDay = 1; // Mulai dari minggu pertama
    int maxSum = 0;

    for (int j = 0; j < nilaiSiswa[0].length; j++) {
        int sum = 0;
        for (int i = 0; i < nilaiSiswa.length; i++) {
            sum += nilaiSiswa[i][j];
        }

        if (sum > maxSum) {
            maxSum = sum;
            maxDay = j + 1;
        }
    }

    return maxDay;
}

```

c.

```

1 static void tampilkanNilaiTertinggi() {
2     int MhsNilaiTertinggi = -1;
3     int totNilaiTertinggi = 0; // Inisialisasi dengan 0
4
5     for (int i = 0; i < nilaiSiswa.length; i++) {
6         int jmlNilai = 0;
7         System.out.print("\nNilai " + namaSiswa[i] + ":\n");
8
9         for (int j = 0; j < nilaiSiswa[i].length; j++) {
10             System.out.println("Week " + (j + 1) + ": " + nilaiSiswa[i][j]);
11
12             totNilaiTertinggi += nilaiSiswa[i][j];
13         }
14
15         if (jmlNilai > totNilaiTertinggi || totNilaiTertinggi == -1) {
16             totNilaiTertinggi = jmlNilai;
17             MhsNilaiTertinggi = i;
18         }
19         System.out.println("Total Nilai: " + jmlNilai + "\n");
20     }
21
22     if (MhsNilaiTertinggi != -1) {
23         System.out.println("Mahasiswa dengan Nilai Tertinggi: " + namaSiswa[MhsNilaiTertinggi]);
24         System.out.println("Total Nilai Tertinggi: " + totNilaiTertinggi);
25     }
26 }

```

d.

3. Modifikasi program tugas no 2 dengan memastikan terdapat input dari user untuk menentukan jumlah mahasiswa dan juga jumlah tugas!

```
1 static void namaMhs() {
2     System.out.print("Masukkan jumlah mahasiswa: ");
3     int jmlMhs = sc.nextInt();
4     namaSiswa = new String[jmlMhs];
5     for (int i = 0; i < namaSiswa.length; i++) {
6         System.out.print("Masukkan nama mahasiswa ke-" + (i + 1) + ": ");
7         namaSiswa[i] = sc.nextLine();
8     }
9 }
10
11 static void nilaiMhs() {
12     System.out.print("Masukkan jumlah tugas: ");
13     int jmlTugas = sc.nextInt();
14     nilaiSiswa = new int[jmlMhs][jmlTugas];
15     for (int i = 0; i < nilaiSiswa.length; i++) {
16         for (int j = 0; j < nilaiSiswa[i].length; j++) {
17             System.out.print("Masukkan nilai " + namaSiswa[i] + " di tugas ke-" + (j + 1) + ": ");
18             nilaiSiswa[i][j] = sc.nextInt();
19         }
20     }
21 }
```

Tugas Kelompok

Implementasikan fungsi ke dalam project kelompok Anda. Jangan lupa, semoga kode program harus di-push ke repository Anda



```
1 static int hitungWaktu(int jamAwal, int jamAkhir) {  
2     int waktuParkir = 0;  
3     if (jamAwal == jamAkhir) {  
4         waktuParkir = 1;  
5     } else {  
6         waktuParkir = jamAkhir - jamAwal;  
7     }  
8     if (jamAkhir != jamAwal && menitAkhir >= menitAwal) {  
9         waktuParkir++;  
10    }  
11    return waktuParkir;  
12 }
```



```
1 //Menentukan waktu Parkir  
2 waktuParkir = hitungWaktu(jamAwal, jamAkhir);
```



```
1 //Menentukan Tarif Parkir Perjam untuk motor dan mobil
2 tarifParkirPerjam = hitungTarifJam(jenisKendaraan, waktuParkir, tarifJam);
```



```
1 public static int hitungTarifJam (String jenisKendaraan, int jamParkir, int tarif[][]){
2     int tarifJam=0;
3     if (jenisKendaraan.equalsIgnoreCase("motor")) {
4         if (jamParkir==2) {
5             tarifJam = tarif[0][0]; //500
6         } else if (jamParkir==3) {
7             tarifJam = tarif[0][1]; //500+1000
8         } else if (jamParkir==4) {
9             tarifJam = tarif[0][2]; //500+1000+1500
10        } else if (jamParkir==5) {
11            tarifJam = tarif[0][3]; //500+1000+1500+2000
12        } else if (jamParkir>5) {
13            tarifJam = tarif[0][4]; //500+1000+1500+2000+(2000 perjam)
14        }
15    }
16
17    if (jenisKendaraan.equalsIgnoreCase("mobil")) {
18        if (jamParkir==2) {
19            tarifJam = tarif[1][0]; //500
20        } else if (jamParkir==3) {
21            tarifJam = tarif[1][1]; //500+1000
22        } else if (jamParkir==4) {
23            tarifJam = tarif[1][2]; //500+1000+1500
24        } else if (jamParkir==5) {
25            tarifJam = tarif[1][3]; //500+1000+1500+2000
26        } else if (jamParkir>5) {
27            tarifJam = tarif[1][4]; //500+1000+1500+2000+(2000 perjam)
28        }
29    }
30
31    return tarifJam;
32 }
```