**LAPORAN PRAKTIKUM DASAR PEMROGRAMAN**

**JOBSHEET 13: FUNGSI 1**

****

**Nama: Afifah Khoirunnisa**

**NIM: 2341720250**

**Kelas: 1B**

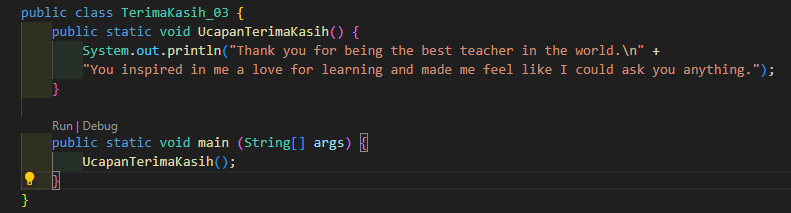
**Prodi : D-IV Teknik Informatika**

2. Praktikum 2.1

Percobaan 1: Membuat Fungsi Tanpa Parameter

Waktu Percobaan: 40 menit

Source code:

****

Output:



Pertanyaan!

1. Apakah fungsi tanpa parameter selalu digunakan untuk fungsi “void” saja?

**Jawaban:**

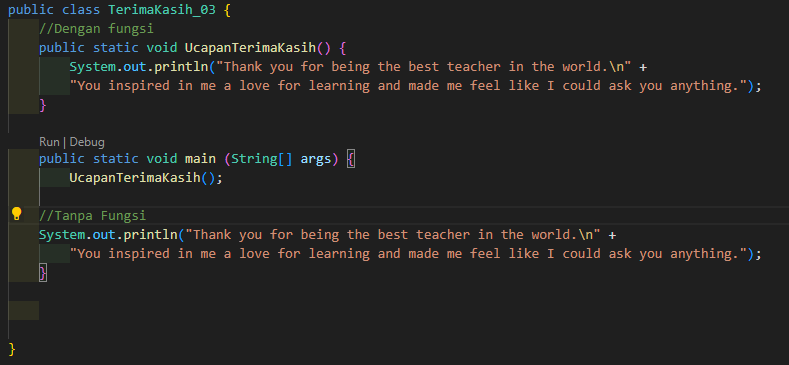
Tidak, fungsi tanpa parameter tidak selalu digunakan untuk fungsi “void” saja. Fungsi tanpa parameter juga dapat digunakan untuk fungsi yang memiliki nilai balik.

1. 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!

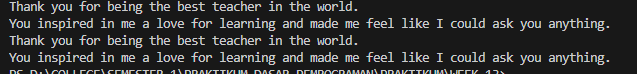
**Jawaban:**

Kalimat “Thank you for…dst” dapat dituliskan tanpa fungsi UcapanTerimaKasih.

Source code:



Output:



1. Apakah keuntungan menggunakan fungsi di dalam program?

**Jawaban:**

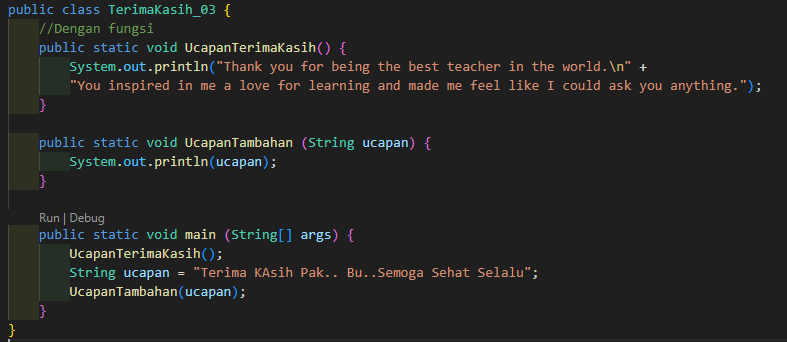
Ada beberapa keuntungan menggunakan fungsi di dalam program, antara lain:

* Kerja program menjadi lebih terstruktur. Fungsi dapat digunakan untuk membagi program menjadi beberapa bagian yang lebih kecil dan lebih mudah dipahami.
* Kode program menjadi lebih mudah diuji. Fungsi dapat diuji secara independen, sehingga memudahkan dalam menemukan dan memperbaiki kesalahan (bug) pada program.
* Kode program menjadi lebih mudah untuk dimaintain. Fungsi dapat diubah atau ditambahkan tanpa mempengaruhi bagian lain dari program.
* Kode program menjadi lebih efisien. Fungsi dapat digunakan untuk melakukan tugas yang berulang-ulang, sehingga mengurangi jumlah kode yang perlu ditulis.

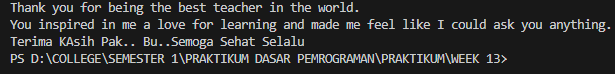
**2.2 Percobaan 2: Membuat Fungsi Dengan Parameter**

**Waktu Percobaan: 40 menit**

Source code:



Output:



Pertanyaan!

1. Apakah kegunaan suatu parameter di dalam fungsi?

**Jawaban:**

Parameter adalah variable yang digunakan untuk menerima data dari luar fungsi. Data ini dapat berupa nilai data, objek, dtau fungsi lain. Parameter digunakan untuk membuat fungsi menjadi lebih fleksibel dan mudah digunakan.

Berikut adalah beberapa kegunaan parameter di dalam fungsi:

* Untuk menerima data input dari pemanggil fungsi. Misalnya, fungsi cetak\_nama() menerima parameter nama untuk menerima nama yang akan dicetak.
* Untuk mengontrol aliran program. Misalnya, fungsi periksa\_usia() menerima parameter usia untuk menentukan apakah seseorang sudah cukup umur.
* Untuk berbagi data antar fungsi. Misalnya, fungsi simpan\_data() dan baca\_data() dapat menggunakan parameter yang sama untuk menyimpan dan membaca data dari file.

1. Apakah parameter sama dengan variabel? jelaskan!

**Jawaban:**

Parameter tidak sama dengan variabel. Parameter adalah variabel yang hanya didefinisikan di dalam fungsi. Variabel adalah variabel yang dapat didefinisikan di mana saja dalam program.

1. Apakah parameter di dalam Bahasa pemrograman Java hanya digunakan untuk melewatkan data input saja? Bagaimana dengan data output?

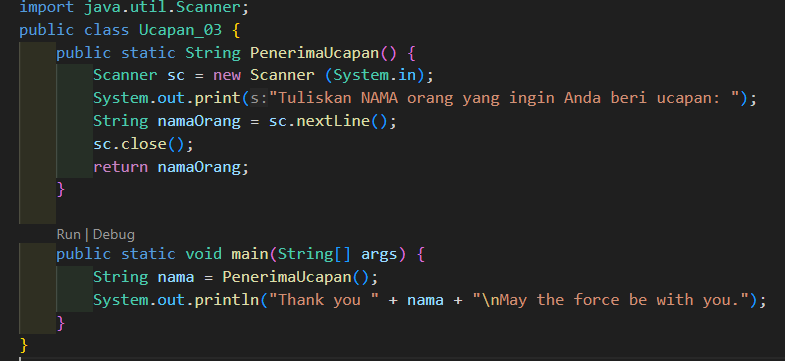
**Jawaban:**

Parameter di dalam Bahasa pemrograman Java dapat digunakan untuk melewatkan data input maupun data output. Untuk melewatkan data output, fungsi dapat menggunakan nilai balik (return value). Nilai balik adalah nilai yang dikembalikan oleh fungsi kepada pemanggilnya. Nilai balik ini dapat berupa nilai data, objek, atau fungsi lain.

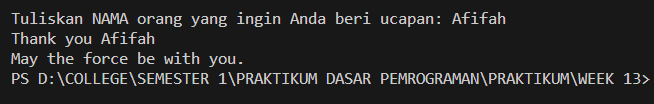
**2.3 Percobaan 3: Membuat Fungsi dengan Nilai Kembalian**

**Waktu Percobaan: 40 menit**

Source code:



Output:



Pertanyaan!

1. Jelaskan mengenai, kapan suatu fungsi membutuhkan nilai kembalian (return)!

**Jawaban:**

Suatu fungsi membutuhkan nilai kembalian (return) jika fungsi tersebut menghasilkan nilai yang perlu dikembalikan kepada pemanggil fungsi. Nilai balik ini dapat berupa nilai data, objek, atau fungsi lain. Jika suatu fungsi tidak menghasilkan nilai yang perlu dikembalikan, maka fungsi tersebut dapat menggunakan tipe void.

1. Apakah bisa ditambahkan statement System.out.println di dalam fungsi dengan nilai kembalian? Apa pengaruh nya?

**Jawaban:**

System.out.println dapat ditambahkan di dalam fungsi dengan nilai kembalian. Statement ini akan mencetak output ke layar.

Pengaruh dari menambahkan statement System.out.println di dalam fungsi dengan nilai kembalian adalah:

* Output dari statement System.out.println akan ditampilkan ke layar.
* Nilai balik dari fungsi tetap akan dikembalikan kepada pemanggil fungsi.

1. Apakah bisa fungsi tanpa nilai kembalian di dalam fungsi main dipanggil tanpa dilewatkan ke variabel? seperti pada percobaan 1? Jelaskan!

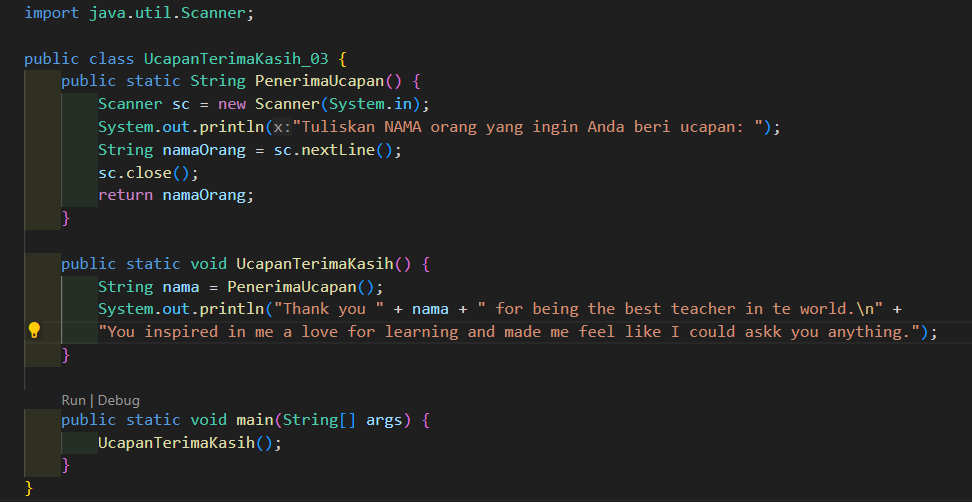
**Jawaban:**

Ya, fungsi tanpa nilai kembalian di dalam fungsi main dapat dipanggil tanpa dilewati ke variabel. Hal ini karena fungsi tanpa nilai kembalian tidak menghasilkan nilai yang perlu dikembalikan kepada pemanggil fungsi.

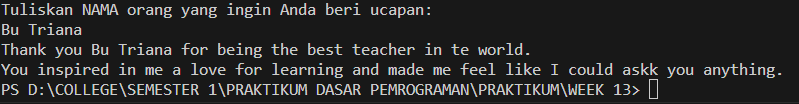
**2.4 Percobaan 4: Fungsi yang dapat meng-Calling Fungsi Lain**

**Waktu Percobaan: 50 menit**

Source code:

****

Output:



Pertanyaan!

1. Berdasarkan uji coba pada percobaan 4, manakah fungsi yang akan eksekusi pertama kali? Jelaskan!

Jawaban:

Berdasarkan uji coba pada percobaan 4, fungsi yang akan dieksekusi pertama kali adalah fungsi main(). Hal ini karena fungsi main() adalah fungsi utama dari program. Fungsi main() akan dieksekusi pertama kali ketika program dijalankan.

Fungsi PenerimaUcapan() akan dieksekusi setelah fungsi main() dieksekusi. Hal ini karena fungsi PenerimaUcapan() dipanggil oleh fungsi main().

Berikut adalah urutan eksekusi fungsi pada percobaan 4:

* Fungsi main()
* Fungsi PenerimaUcapan()

1. Manakah tata cara penulisan fungsi benar di dalam class? Di atas fungsi main ataukah dibawah fungsi main? Jelaskan!

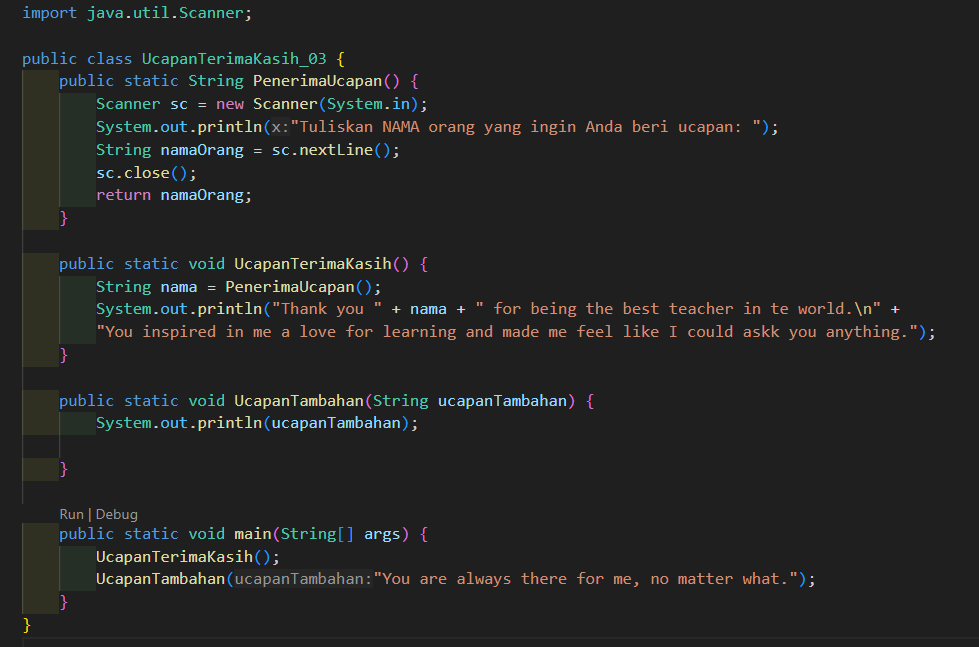
**Jawaban:**

Tata cara penulisan fungsi yang benar di dalam class adalah di atas fungsi main. Hal ini karena fungsi main() adalah fungsi utama dari program. Fungsi harus ditulis terlebih dahulu agar dapat memanggil fungsi-fungsi lain yang ada di dalam class.

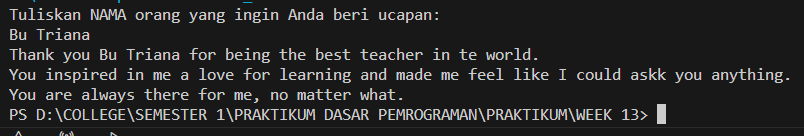
1. Modifikasi program di atas dengan menambahkan fungsi UcapanTambahan() dengan input parameter String. Fungsi UcapanTambahan() berisi ucapan tambahan yang ingin disampaikan kepada penerima ucapan.

**Jawaban:**

Source code:



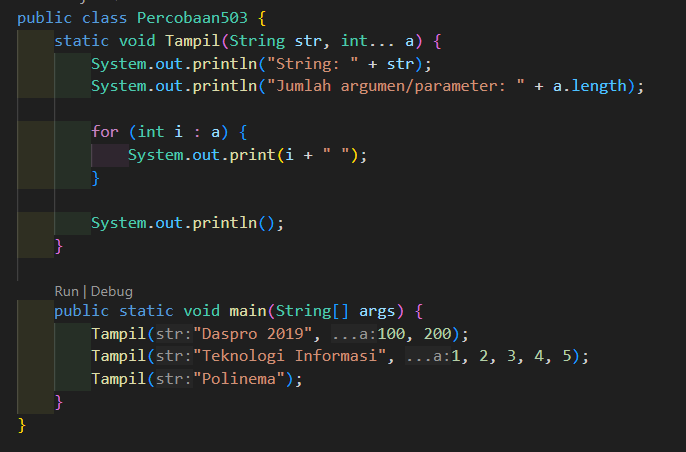
Output:



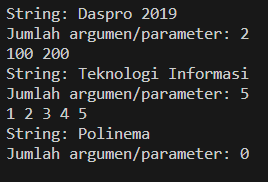
**2.5 Percobaan 5: Fungsi Varargs**

**Waktu Percobaan: 40 menit**

Source code:



Output:



Pertanyaan!

1. Jelaskan mengapa penulisan parameter di praktikum 5 di tulis dengan int... a!

**Jawaban:**

Penulisan parameter di praktikum 5 di tulis dengan int... a karena parameter tersebut adalah varargs. Varargs adalah parameter yang dapat menerima jumlah argumen yang tidak ditentukan.

Pada praktikum 5, fungsi Tampil() menerima dua parameter. Parameter pertama adalah String yang digunakan untuk menerima string. Parameter kedua adalah int... yang digunakan untuk menerima angka.

1. Sebutkan kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalah dalam dunia nyata! (minimal 3)

**Jawaban:**

Berikut adalah beberapa kegunaan varargs dalam implementasi kode program untuk menyelesaikan permasalahan dalam dunia nyata:

* Untuk menerima jumlah argumen yang tidak ditentukan. Misalnya, fungsi Tampil() pada praktikum 5 dapat digunakan untuk menampilkan string dan angka apa pun, tanpa perlu dibatasi jumlah argumennya.
* Untuk membuat kode program lebih ringkas. Misalnya, fungsi Tampil() pada praktikum 5 dapat digunakan untuk menampilkan string dan angka dalam satu fungsi, tanpa perlu membuat fungsi terpisah untuk setiap jumlah argumen.
* Untuk membuat kode program lebih fleksibel. Misalnya, fungsi Tampil() pada praktikum 5 dapat digunakan untuk menampilkan string dan angka dengan jumlah argumen yang berbeda-beda, sesuai kebutuhan.

1. Bisakah kita menggunakan dua tipe data varaargs dalam satu fungsi? Berikan contohnya!

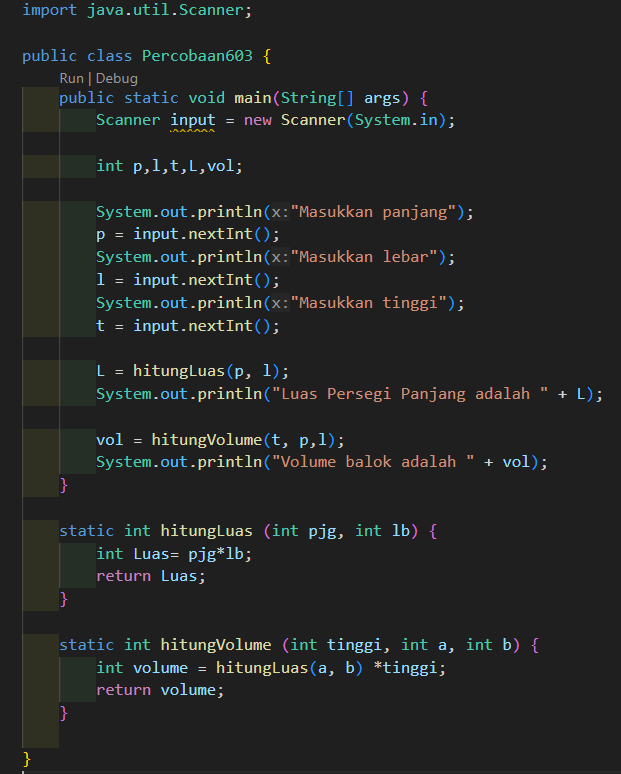
**Jawaban:**

Dalam Java, dapat menggunakan varargs (variadic arguments) untuk memungkinkan fungsi menerima jumlah argumen yang tidak terbatas dari jenis data tertentu. Namun, secara langsung, tidak dapat memiliki dua varargs parameter dalam satu fungsi. Varargs sendiri sudah mewakili sejumlah argumen yang dapat berubah dinamis, sehingga tidak perlu lebih dari satu.

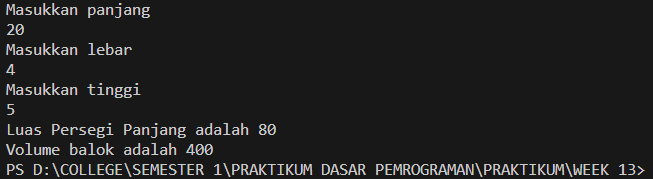


**2.6 Percobaan 6: Pembuatan Kode Program, Dengan Fungsi versus Tanpa Fungsi Waktu Percobaan: 50 menit**

Source code:



Output:



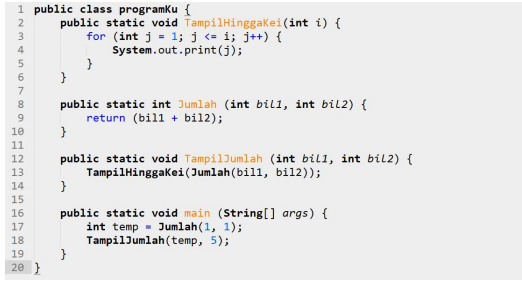
Pertanyaan!

1. Sebutkan tahapan dan urutan ekskusi pada percobaan 6 di atas!

**Jawaban:**

* 1. Deklarasi Kelas dan Fungsi:
* Deklarasi kelas Percobaan603.
* Deklarasi fungsi main, hitungLuas, dan hitungVolume.
  1. Deklarasi Variabel:
* Scanner input dan variabel p, l, t, L, dan vol dideklarasikan.
  1. Inisialisasi Scanner:
* Objek Scanner dibuat untuk membaca input dari pengguna.
  1. Input Panjang, Lebar, dan Tinggi:
* Mencetak pesan untuk meminta input panjang, lebar, dan tinggi.
* Membaca input panjang, lebar, dan tinggi dari pengguna.
  1. Panggil Fungsi hitungLuas:
* Memanggil fungsi hitungLuas dengan parameter p dan l untuk menghitung luas persegi panjang.
* Hasilnya disimpan dalam variabel L.
  1. Output Luas Persegi Panjang:
* Mencetak hasil perhitungan luas persegi panjang.
  1. Panggil Fungsi hitungVolume:
  2. Memanggil fungsi hitungVolume dengan parameter t, p, dan l untuk menghitung volume balok.
* Hasilnya disimpan dalam variabel vol.
  1. Output Volume Balok:
* Mencetak hasil perhitungan volume balok.

1. Apakah output dari program dibawah ini kemudian jelaskan alur jalannya program tersebut!



**Jawaban:**

Output:

****

Program tersebut adalah program sederhana dalam bahasa pemrograman Java yang mendefinisikan sebuah kelas bernama `ProgramKu`. Program ini memiliki tiga metode statis: `TampilHinggaKei`, `Jumlah`, dan `TampilJumlah`. Metode `TampilHinggaKei` digunakan untuk mencetak angka dari 1 hingga nilai parameter yang diberikan tanpa spasi di antaranya. Metode `Jumlah` mengembalikan hasil penjumlahan dari dua bilangan yang diberikan. Metode `TampilJumlah` memanggil `Jumlah` untuk mendapatkan hasil penjumlahan dan kemudian memanggil `TampilHinggaKei` untuk mencetak angka dari 1 hingga hasil penjumlahan tersebut. Metode `main` dijalankan saat program dijalankan, di mana sebuah variabel `temp` diinisialisasi dengan hasil penjumlahan 1 dan 1, dan kemudian metode `TampilJumlah` dipanggil dengan parameter `temp` dan 5. Akhirnya, output program adalah angka 12, yang merupakan hasil dari penjumlahan 1+1, diikuti dengan angka dari 1 hingga 2 (hasil dari pemanggilan `TampilHinggaKei`).

1. 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!

**Jawaban:**

Keputusan untuk menggunakan parameter atau nilai kembalian dalam fungsi tergantung pada sifat dan kebutuhan spesifik fungsi dalam konteks program. Parameter digunakan ketika fungsi memerlukan input eksternal atau data dari luar untuk diolah di dalamnya, meningkatkan fleksibilitas dan modularitas fungsi. Dengan menggunakan parameter, fungsi dapat diimplementasikan untuk beroperasi dengan nilai yang berbeda, memungkinkan penggunaannya dalam berbagai konteks.

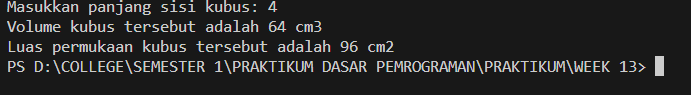
Sebaliknya, nilai kembalian diperlukan ketika fungsi dimaksudkan untuk menghasilkan nilai atau informasi yang dapat digunakan di tempat lain dalam program. Hal ini relevan ketika suatu fungsi melakukan proses dan hasilnya perlu diketahui atau digunakan oleh bagian lain dari program. Fungsi void tanpa nilai kembalian biasanya cocok untuk tugas yang tidak menghasilkan output yang harus digunakan di tempat lain dalam program, seperti mencetak ke layar. Pilihan untuk menggunakan parameter, nilai kembalian, atau kombinasi keduanya didesain untuk mendukung prinsip modularitas, reusabilitas, dan pemahaman yang baik terhadap struktur dan tujuan program secara keseluruhan.

**Tugas 1**

**Source code:**

****

**Output:**

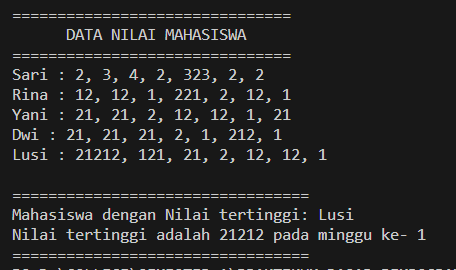
****

**Tugas 2**

**Source code:**



**Output:**

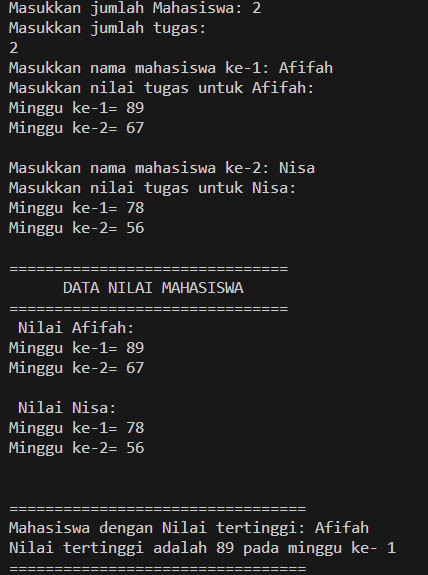
****

**Tugas 3:**

**Source code**



**Output:**

****

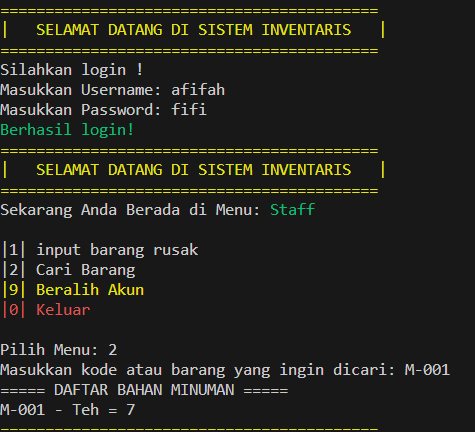
**Tugas kelompok:**

**1. Fungsi pencarian Barang**

**Source code:**

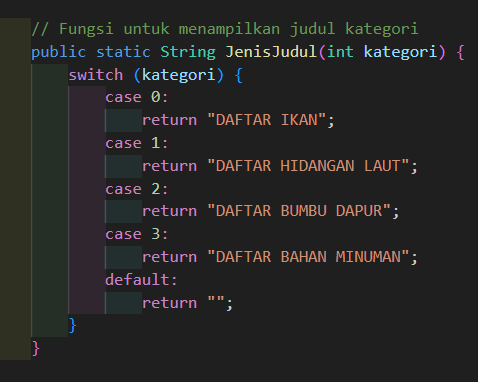
****

**Output:**

****

**Fungsi pada pemilihan judul kategori**

**Source code;**

****

**Output:**

****