JOBSHEET 11

Fungsi 1

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 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("===== MENU RESTO KAFE =====");

        System.out.println("1. Kopi Hitam - Rp 15,000");

        System.out.println("2. Cappuccino - Rp 20,000");

        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.");

    }

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!



Pertanyaan

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

2. Apakah bisa perintah menampilkan menu yang ada dituliskan tanpa fungsi Menu?

modifikasi kode program tersebut untuk dapat menampilkan daftar menu tanpa

menggunakan fungsi!

3. Apakah keuntungan menggunakan fungsi di dalam program?

Jawaban Pertanyaan

1. tidak hanya void saja, asal fungsi tersebut memiliki “return”

2.

import java.util.Scanner;

public class Kafe14 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Nama pelanggan : ");

        String namaPelanggan = input.nextLine();

        System.out.print("Apakah member? [y/n]");

        String isMember = input.nextLine();

        System.out.println("Selamat datang, " + namaPelanggan + "!");

        if (isMember.equalsIgnoreCase("y")) {

            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("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.");

    }

}

3. • Modular: sekelompok statement yang berfungsi untuk menjalankan

tugas tertentu, dikelompokkan sendiri dan dipisah, dengan diberikan

nama tertentu, sehingga jika diperlukan dalam suatu program untuk

menjalankan tugas tersebut, maka dapat memanggil nama statement

tersebut.

• Efektif: Jika tugas tersebut dilakukan dalam program berulang-ulang,

maka tidak perlu dituliskan berulang-ulang, tapi yang dilakukan hanya

cukup memanggil fungsi tersebut

2.2 Percobaan 2: Membuat Fungsi Dengan Parameter

Waktu Percobaan: 40 menit

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("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("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.");

    }

2. Eksekusi fungsi UcapanTambahan dari dalam fungsi main.

 public static void main(String[] args) {

        Menu("Andi", true);

    }

}

Pertanyaan!

1. Apakah kegunaan parameter di dalam fungsi?

2. Apakah parameter sama dengan variabel? jelaskan!

3. Bagaimana cara kerja parameter isMember pada method Menu

4. Apa yang akan terjadi jika memanggil Menu tanpa menyertakan parameter

namaPelanggan dan isMember?

5. 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

Jawaban Pertanyaan

1. untuk menerima input dari luar fungsi tersebut

2. sama, karena parameter sendiri adalah variable yang menampung nilai untuk diproses di

dalam fungsi

3. parameter isMember tersebut bertipe data Boolean, jika true maka akan mengeksekusi if

Yang ada pada function menu

4. error

5.

public class Kafe14 {

    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.equalsIgnoreCase("DISKON50")) {

            System.out.println("Diskon 50%");

        } else if (kodePromo.equalsIgnoreCase("DISKON30")) {

            System.out.println("Diskon 30%");

        } else {

            System.out.println("Kode 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("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.");

    }

    public static void main(String[] args) {

        Menu("Andi", true,"DISKON50");

    }

}

2.3 Percobaan 3: Membuat Fungsi dengan Nilai Kembalian

Waktu Percobaan: 40 menit

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 class hitungTotalHarga14 {

    public static int hitungTotalHarga14(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) {

        System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");

        int pilihanMenu = sc.nextInt();

        System.out.print("Masukkan jumlah item yang ingin dipesan: ");

        int banyakItem = sc.nextInt();

        int totalHarga = hitungTotalHarga14(pilihanMenu, banyakItem);

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

    }

Pertanyaan!

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

2. Modifikasi kode diatas sehingga fungsi hitungTotalHarga dapat menerima kodePromo.

Jika kodePromo adalah "DISKON50", maka mendapat diskon 50% dari totalHarga. Jika

kodePromo adalah "DISKON30", maka mendapat diskon 50% dari totalHarga

tampilkan berikan diskon 30%. Jika tidak ada kode promo yang berlaku, tampilkan

kode invalid dan tidak ada pengurangan total harga totalHarga.

3. Modifikasi kode diatas sehingga bisa memilih beberapa jenis menu berbeda serta

menampilkan total keseluruhan pesanan.

Jawaban Pertanyaan

1. Ketika fungsi tersebut memerlukan inputan dari luar

2.

import java.util.Scanner;

public class HitungTotalHarga14 {

    public static int hitungTotalHarga14(int pilihanMenu, int banyakItem, String kodePromo) {

        int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};

        if (pilihanMenu < 1 || pilihanMenu > hargaItems.length) {

            System.out.println("Menu tidak valid!");

        }

        int hargaTotal = hargaItems[pilihanMenu - 1] \* banyakItem;

        double diskon;

        if (kodePromo.equalsIgnoreCase("DISKON50")) {

            diskon = 0.5;

            System.out.println("Diskon 50%");

        } else if (kodePromo.equalsIgnoreCase("DISKON30")) {

            diskon = 0.3;

            System.out.println("Diskon 30%");

        } else {

            diskon = 1.0;

            System.out.println("Kode invalid dan tidak ada pengurangan harga");

        }

        return (int) (hargaTotal \* diskon);

    }

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");

        int pilihanMenu = input.nextInt();

        System.out.print("Masukkan jumlah item yang ingin dipesan: ");

        int banyakItem = input.nextInt();

        input.nextLine();

        System.out.print("Masukkan kode promo: ");

        String kodePromo = input.nextLine();

        int totalHarga = hitungTotalHarga14(pilihanMenu, banyakItem, kodePromo);

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

    }

}

3.

import java.util.Scanner;

public class HitungTotalHarga14 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int totalKeseluruhan = 0;

        String[] namaMenu = {"Nasi Goreng", "Mie Ayam", "Bakso", "Sate Ayam", "Es Teh", "Jus Jeruk"};

        int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};

        System.out.println("Daftar Menu:");

        for (int i = 0; i < namaMenu.length; i++) {

            System.out.println((i + 1) + ". " + namaMenu[i] + " - Rp" + hargaItems[i]);

        }

        char tambahPesanan;

        do {

            System.out.print("\nMasukkan nomor menu yang ingin Anda pesan: ");

            int pilihanMenu = input.nextInt();

            if (pilihanMenu < 1 || pilihanMenu > namaMenu.length) {

                System.out.println("tidak valid, Silakan pilih ulang");

                continue;

            }

            System.out.print("Masukkan jumlah: ");

            int banyakItem = input.nextInt();

            input.nextLine();

            System.out.print(" kode promo: ");

            String kodePromo = input.nextLine();

            int totalHarga = hitungTotalHarga(pilihanMenu, banyakItem, kodePromo, hargaItems);

            totalKeseluruhan += totalHarga;

            System.out.println("Subtotal : " + totalHarga);

            System.out.print("apakah beli lagi? (y/n): ");

            tambahPesanan = input.next().charAt(0);

            input.nextLine();

        } while (tambahPesanan == 'y' || tambahPesanan == 'Y');

        System.out.println("Total: " + totalKeseluruhan);

    }

    public static int hitungTotalHarga(int pilihanMenu, int banyakItem, String kodePromo, int[] hargaItems) {

        int hargaTotal = hargaItems[pilihanMenu - 1] \* banyakItem;

        double diskon;

        if (kodePromo.equalsIgnoreCase("DISKON50")) {

            diskon = 0.5;

            System.out.println("Diskon 50% diterapkan.");

        } else if (kodePromo.equalsIgnoreCase("DISKON30")) {

            diskon = 0.3;

            System.out.println("Diskon 30% diterapkan.");

        } else {

            diskon = 1.0;

            System.out.println("Kode promo tidak valid. Tidak ada diskon.");

        }

        return (int) (hargaTotal \* diskon);

    }

}

2.4 Percobaan 4: Fungsi Varargs

Waktu Percobaan: 40 menit

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

    }

Pertanyaan!

1. Jelaskan mengapa penulisan parameter di praktikum 4 di tulis dengan String...

namaPengunjung!

2. Modifikasi method daftarPengunjung menggunakan for-each loop!

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

Jawaban Pertanyaan

1. … berarti menjadikan variable namaPengunjung jadi banyak variable tanpa harus deklarasi banyak

2.

public class PengunjungCafe14 {

    public static void daftarPengunjung(String... namaPengunjung) {

        System.out.println("Daftar Nama Pengunjung:");

        for (String nama : namaPengunjung) {

            System.out.println("- " + nama);

        }

    }

    public static void main(String[] args) {

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

    }

}

2.5 Percobaan 5: Pembuatan Kode Program, Dengan Fungsi versus Tanpa

Fungsi

Waktu Percobaan: 50 menit

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



2. Buatlah program untuk menghitung luas persegi panjang dan volume balok tanpa

menggunakan fungsi

public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int p,i,t,L,vol;

        System.out.println("Masukkan panjang");

        p=input.nextInt();

        System.out.println("Masukkan lebar");

        i=input.nextInt();

        System.out.println("Masukkan tinggi");

        t=input.nextInt();

        L=p\*l\*t;

        System.out.println("Luas Persegi panjang adalah " +L);

        vol=p\*l\*t;

        System.out.println("Volume balok adalah " + vol);

    }

3. Program menghitung luas persegi dan volume balok di atas jika dibuatkan fungsi maka

terdapat 3 fungsi yaitu hitungLuas, hitungVolume dan fungsi main, seperti di bawah

ini:

Fungsi hitungLuas

static int hitungLuas (int pjg, int lb) {

        int Luas=pjg\*lb;

        return Luas;

    }

Fungsi hitungVolume

static int hitungVolume (int tinggi, int a, int b) {

        int volume= hitungLuas(a,b)\*tinggi;

        return volume;

    }

Fungsi main

 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!

Pertanyaan!

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

2. Apakah output dari program di bawah ini kemudian jelaskan alur jalannya program

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!

Jawaban Pertanyaan

1. pertama program akan menjalankan fungsi main terlebih dahulu, di fungsi main user akan meng input variable p,l,t(Panjang lebar dan tinggi), lalu variable L menghubungkan variable di fungsi hitungLuas (pjg dengan p,lb dengan l) dan Variabel vol menghubungkan variable di fungsi hitungVolume (tinggi dengan t,a dengan p,b dengan l),

-setelah variable di fungsi hitungLuas menerima input dari fungsi main maka luas akan dihitung dan hasilnya akan di return(agar bisa dibaca di luar fungsi tsb)

- setelah variable di fungsi hitungVolume menerima input dari fungsi main maka volume akan dihitung dan hasilnya akan di return(agar bisa dibaca di luar fungsi tsb)

Setelah di hitung dalam fungsi tsb, hasil akan ditampilkan di fungsi main

2.

public class programKu14 {

    public static void TampilHinggaKei (int i) {

        for (int j = 1; j <= i; j++) {

            System.out.print(j);

        }

    }

    public static int Jumlah (int bil1, int bil2) {

        return (bil1 +  bil2);

    }

    public static void TampilJumlah (int bil1, int bil2) {

        TampilHinggaKei(Jumlah(bil1, bil2));

    }

    public static void main(String[] args) {

        int temp = Jumlah(1, 1);

        TampilJumlah(temp, 5);

    }

}



ALUR :

Jumlah menambahkan dua bilangan.

TampilHinggaKei menampilkan angka dari 1 hingga nilai tertentu.

TampilJumlah menggabungkan kedua fungsi tersebut untuk menampilkan hasil akhir.

Di main, angka 1-7 ditampilkan berdasarkan penjumlahan 2 + 5.

### 3. Kapan Fungsi Tidak Memerlukan Parameter:

**1. Data Sudah Tersedia di Dalam Fungsi atau Objek**

2. Pemrosesan Data yang Hasilnya Dibutuhkan untuk Langkah Selanjutnya

### ****Kapan Fungsi Tidak Memiliki Nilai Kembalian (Void):****

1. **Tugas Hanya untuk Menjalankan Proses:**

2. Memodifikasi Data Internal Objek (Mutator):