Latihan Dasar Pemrograman Pertemuan 13

Nama :Rasyiq Satrio Musthafa

Kelas :TI 1G

No Absen : 25

Pertanyaan 1.

```
public class kafe25 {
    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: "Silahkan pilih menu yang anda inginkan.");

    }
    Run | Debug
    public static void main(String[] args) {
        menu();
    }
}
```

1. Apakah fungsi tanpa parameter selalu harus bertipe void?
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!
3. Jelaskan keuntungan menggunakan fungsi Menu() dibandingkan menulis semua perintah penampilan menu langsung di dalam fungsi main
4. Uraikan secara singkat alur eksekusi program ketika fungsi Menu() dipanggil dari main (mulai dari program dijalankan sampai daftar menu tampil di layar).

Jawab

1. Fungsi tanpa parameter tidak harus bertipe void. Parameter itu tanda agar fungsi itu berjalan di luar cabang. Tipe fungsi itu soal *output*.

2. Daftar menu tidak dapat ditampilkan jika fungsi menu() tidak diperintah.

```
public class kafe25 {

    Run | Debug
    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: "Silahkan pilih menu yang anda inginkan.");
    }
}
```

3. Keuntungan memakai fungsi Menu() adalah kode menjadi lebih terstruktur. Bagian main tetap ringkas karena tidak dipenuhi perintah tampilan menu. Perubahan pada isi menu juga lebih mudah dilakukan karena cukup dilakukan di satu tempat. Selain itu, penggunaan fungsi terpisah membantu menjaga keteraturan program dan mengurangi pengulangan kode.
4. Saat program dijalankan, eksekusinya dimulai dari fungsi main. Di dalam main, terdapat satu perintah yaitu menu();. Pada saat baris tersebut dieksekusi, alur program berpindah ke fungsi menu(). Fungsi menu() kemudian menjalankan seluruh perintah System.out.println() yang ada di dalamnya secara berurutan. Setelah semua baris tersebut dicetak, daftar menu tampil sepenuhnya di layar, dan eksekusi kembali ke main. Setelah itu, program selesai dijalankan.

Pertanyaan 2.

```java
public class AlterKafe25 {
    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% 
        }

        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: "Silahkan pilih menu yang anda inginkan.");
    }

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

1. Apakah kegunaan parameter di dalam fungsi?
2. Jelaskan mengapa pada percobaan ini fungsi ***Menu()*** menggunakan parameter ***namaPelanggan*** dan ***isMember?***
3. Apakah parameter sama dengan variabel? Jelaskan.
4. Jelaskan bagaimana cara kerja parameter ***isMember*** pada fungsi ***Menu()***. Apa perbedaan output ketika ***isMember*** bernilai true dan ketika false?
5. Apa yang akan terjadi jika memanggil fungsi Menu() tanpa menyertakan parameter ***namaPelanggan*** dan ***isMember***?
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.
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.
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.

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

Jawab

1. Parameter dalam fungsi digunakan untuk menerima nilai dari luar fungsi sehingga fungsi dapat bekerja berdasarkan data yang diberikan.
2. Karena data yang ada pada fungsi main() akan diekspor ke fungsi menu, sehingga data yang diimpor mirip dengan yang di ekspor.
3. Parameter dan variabel tidak sepenuhnya sama, meskipun keduanya sama-sama menyimpan nilai. Parameter adalah variabel **khusus** yang digunakan untuk menerima nilai ketika suatu fungsi dipanggil. Nilainya berasal dari luar fungsi. Sementara itu, variabel biasa dapat dibuat di mana saja dalam program dan digunakan untuk menyimpan data sesuai kebutuhan.
4. Parameter isMember berperan sebagai penanda apakah pelanggan termasuk anggota atau tidak. Nilai ini dikirim dari main ke fungsi menu() saat fungsi dipanggil. Di dalam fungsi, isMember digunakan pada bagian if (isMember):
    a. Jika nilainya **true**, kondisi if terpenuhi dan program menampilkan pesan khusus untuk member, misalnya pemberitahuan diskon.
    b. Jika nilainya **false**, kondisi if tidak dijalankan sehingga pesan mengenai keanggotaan tidak muncul. Menu tetap ditampilkan, tetapi tanpa informasi khusus tersebut.
5. Akan muncul error((The method menu(String, boolean) in the type AlterKafe25 is not applicable for the arguments ()") karena fungsi menu() **mewajibkan dua parameter**, yaitu String namaPelanggan dan boolean isMember.
6.

```
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 == "DISKON50") {
        System.out.println(x: "anda diberikan diskon 50% karena sudah menggunakan kode");
    } else if (kodePromo == "DISKON30") {
        System.out.println(x: "anda diberikan diskon 30% karena sudah menggunakan kode");
    } else {
        System.out.println(x: "Invalid.");
    }
```

```java
public class modAlterKafe25 {
    public static void menu(String namaPelanggan, boolean isMember, String kodePromo) {
            System.out.print("halo Budi!!! ");
            isMember = true;
            kodePromo = "DISKON30";
        } else {
            System.out.println("Selamat datang, " + namaPelanggan + "!");
        }
        if (isMember) {
            System.out.println("Anda adalah member, dapatkan diskon 10% untuk setiap pembelian");
        }

        if (kodePromo == "DISKON50") {
            System.out.println("anda diberikan diskon 50% karena sudah menggunakan kode");
        } else if (kodePromo == "DISKON30") {
            System.out.println("anda diberikan diskon 30% karena sudah menggunakan kode");
        } else {
            System.out.println("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: "Budi", isMember: false, kodePromo: "");
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
...
ee11.modAlterKafe25'
halo Budi!!! Anda adalah member, dapatkan diskon 10% untuk setiap pembelian
anda diberikan diskon 30% karena sudah menggunakan kode
==== MENU RESTO KAFE ====
1. Kopi Hitam - Rp. 15,000
2. Cappuccino - Rp 20,000
3, Latte - Rp 22,000
4. Teh Tarik - Rp 12,000
5. Roti Bakar - Rp 10,000
6. Mie Goreng - Rp 18,000
```

7.
8. Pemakaian parameter namaPelanggan dan isMember pada fungsi menu() memang menjadikan program lebih mudah dibaca dan lebih nyaman dikembangkan. Ketika suatu fungsi menerima data dari luar melalui parameter, fungsi tersebut menjadi lebih fleksibel. Program tidak terkunci pada satu nama pelanggan atau satu status member saja. Nilai-nilai itu dapat diganti kapan saja dari bagian program lain tanpa perlu mengubah isi fungsi. Hal ini membuat struktur kode lebih bersih dan lebih mudah dipahami oleh siapa pun yang membaca.
9. Sudah.

Pertanyaan 3

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

Jawab

1. Fungsi butuh nilai kembalian ketika program memerlukan *hasil perhitungan* atau *data baru* dari fungsi tersebut untuk dipakai lagi di bagian lain program. Kalau fungsinya cuma menjalankan tindakan—seperti menampilkan teks ke layar—maka tidak perlu mengembalikan nilai apa pun.
2. Data yang dikembalikan harus sesuai dengan parameter fungsinya, yaitu integer.
    a. **pilihanMenu**

        Parameter ini berisi nomor menu yang dipilih pelanggan. Karena harga disimpan dalam array, angka ini berfungsi sebagai penunjuk posisi (index) menu mana yang diambil. Minus satu diperlukan karena array mulai dari 0, sedangkan nomor menu biasanya ditampilkan mulai dari 1
    b. **banyakItem**

        Parameter ini menunjukkan jumlah porsi atau jumlah item yang dipesan untuk menu tersebut. Angka ini akan dikalikan dengan harga satuannya, sehingga total harga bisa dihitung dengan tepat.

3.
```
int[] hargaItems = {15000, 20000, 22000, 12000, 10000, 18000};
double[] diskonKode = {0.3, 0.5};

double i = 0;

if (kodePromo == "DISKON50") {
    i = diskonKode[1];
} else if (kodePromo == "DISKON30") {
    i = diskonKode[0];
}

int hargaTotalKotor = hargaItems[pilihanMenu - 1] * banyakItem;
double hargaTotal = hargaTotalKotor * i;

return hargaTotal;


blic 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 == "DISKON50") {
    System.out.println(x: "anda diberikan diskon 50% karena sudah menggunakan kode");
} else if (kodePromo == "DISKON30") {
    System.out.println(x: "anda diberikan diskon 30% karena sudah menggunakan kode");
} else {
    System.out.println(x: "Invalid.");
}
```

4.
```
do {
    System.out.print(s: "\nMasukan nomor menu yang ingin anda pesan: ");
    pilihanMenu = sc.nextInt();

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

    System.out.print(s: "Apakah anda ingin menambah pesanan? (ya/tidak): ");
    String jawab = sc.next();
    lanjut = !jawab.equalsIgnoreCase(anotherString: "tidak");

} while (lanjut);
```

5. Sudah.

Pertanyaan 4

1. Jelaskan mengapa penulisan parameter di praktikum 4 ditulis dengan ***String...namaPengunjung***!
2. Modifikasi method ***daftarPengunjung*** menggunakan ***for-each loop***
3. Bisakah menggunakan dua tipe data varaargs dalam satu fungsi? Jelaskan jawaban Anda berdasarkan aturan varargs di Java, dan berikan contohnya!
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?
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 4"

Jawab

1. karena fungsi tersebut perlu menerima jumlah argumen yang tidak tetap. Tiga titik (...) adalah varargs, yang membuat parameter otomatis diperlakukan sebagai array String.
2.
```
static void daftarPengunjung(String... namaPengunjung) {
    System.out.println(x: "Daftar Nama Pengunjung");
    for (String nama : namaPengunjung) {
        System.out.println("- " + nama);
    }
```
3. Java hanya mengizinkan satu varargs per metode, dan harus berada di akhir parameter. Pembatasan ini dibuat supaya compiler tidak bingung ketika mencocokkan argumen yang diberikan pemanggil fungsi. Contohnya: **void cetakData(String... nama, int... nilai)** Adalah aturan yang dilarang
4. Jika fungsi daftarPengunjung dipanggil tanpa argumen, program tetap berjalan dengan normal. Tidak ada error saat kompilasi maupun saat dijalankan. Parameter String... namaPengunjung adalah varargs. Varargs berarti parameter tersebut boleh menerima 0 argumen, 1 argumen, atau lebih. Jika tidak ada argumen yang diberikan, Java otomatis membuatkannya sebagai array kosong (String[0]).
5. sudah