Evaluasi Quiz & Tugas selama NTS

Evaluasi quiz(1)

Berikut ini saya jabarkan beberapa hal mengenai kesalahan umum dan fatal yang perlu Anda PERHATIKAN. Semoga menjadi bahan evaluasi menjelang UTS.

Perhatikan Perintah soal. Di soal quiz diWAJIBKAN **membuat kelas** bernama Pulsa. Membuat kelas dengan cara Klik Kanan > New > JavaClass.
Kesalahan banyak terjadi adalah:
- mengganti nama mainclass anda dengan PULSA, atau
- membuat kelas baru dan tidak digunakan, atau
- tidak membuat kelas sama sekali

```java
public class Pulsa {

    public int _pulsa;
    public int _kuota;

    public Pulsa(int _pulsa, int _kuota) {
        this._pulsa = _pulsa;
        this._kuota = _kuota;
    }

    public static void main(String[] args) throws IOException{
        ServerSocket sc = null;
        try {
            sc = new ServerSocket(1234);
        } catch (IOException ex) {
            Logger.getLogger(Pulsa.class.getName()).log(Level.SEVERE, null, ex);
        }

        while (true)
        {
            Socket incoming = null;
            try {
                incoming = sc.accept();
```

CONTOH **SALAH**

Mengganti/membuat main kelas dengan nama PULSA. Implementasi benar namun tidak sesuai dengan maksud dari pembuat soal.

Pulsa.java | Main.java

```java
                System.out.println(ex);
            }
        }
    }
    public static void handleSocket(Socket incoming)
        BufferedReader input1 = new BufferedReader(ne
        PrintStream output1 = new PrintStream(incomin
        output1.println("Hallo. Enter END to Exit");

        int pulsa = 0;
        int kuota = 0;
        int isiPulsa50 = 0;
        int totpulsa = pulsa + isiPulsa50;
        boolean done = false;
        while(!done){
            String str1 = input1.readLine();
            if(str1==null){
                done = true;
                System.out.println("Null Recieved");
            }
```

CONTOH **SALAH**

Sudah membuat kelas Pulsa dan constructornya, namun pada nyata tidak dipakai pada Main Class. Buat apa?! Entahlah...

```java
public class Pulsa {
    public int Pulsa;
    public int Kuota;


    public Pulsa (int pulsa,int kuota)
    {
        Pulsa = pulsa;
        Kuota = kuota;
    }
}
```

Implementasi yang benar untuk kelas **PULSA**

```java
private static void handleSocket(Socket inc

    BufferedReader reader = new BufferedRea

    PrintStream out = new PrintStream(incom
    out.println("Hi, Enter END to exit");

    Pulsa p = new Pulsa(0, 0);
```

```java
if (str.trim().equals("CEK_PULSA"))
{
    out.println("Sisa pulsa anda sebesar Rp. "+ p._pulsa +" dan kuota internet tersisa "+ p._kuota +" Mb");
}
else if (str.trim().equals("BELI_PULSA"))
```

```java
else if("BELI_PULSA".equals(pesan))
{
    pesan = reader.readLine();
    pulsa.Pulsa += Integer.parseInt(pesan);
    out.println("Pembelian pulsa Rp. " + pesan + " berhasil. Pulsa saat ini Rp. " + pulsa.Pulsa);
}
```

Implementasi yang benar untuk pengambilan dan penggunaan kelas PULSA di Server

Evaluasi quiz (2)

Method BELI PULSA sebagian besar salah implementasi. Coba kalian putar otak dan bayangkan Anda beli pulsa di Ind*m**** atau sejenisnya.

1. Silahkan masukkan command Anda. Anda memasukkan “BELIPULSA”

2. Lalu sistem meminta kembali nominal Pulsa yang Anda masukkan. Anda menekan digit nilai nominal Pulsa.

Dari skenario di atas, berarti untuk membeli pulsa harusnya mesin server meminta ulang digitnya. **Tidak bisa dimasukkan ke dalam command saat Anda mau beli PULSA**

Commandnya aneh g?!

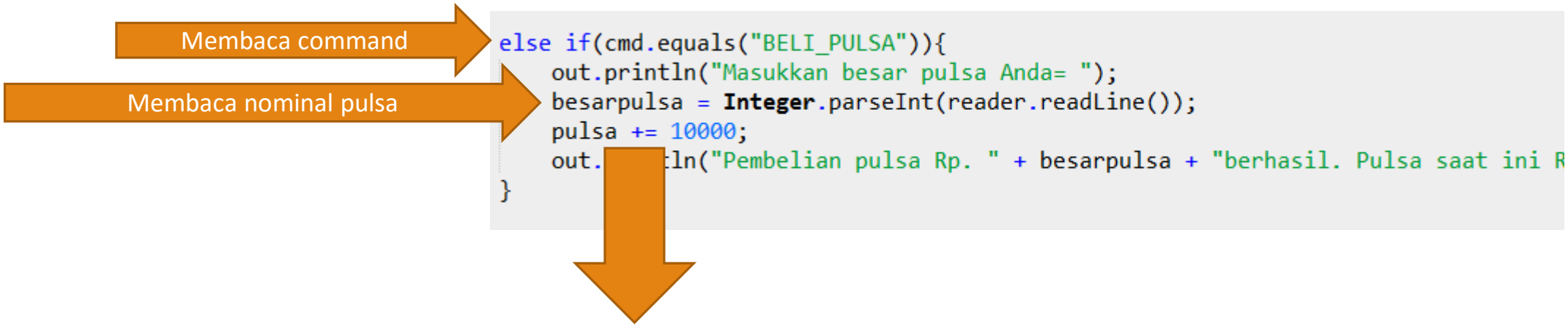
```
else if (inputs.equals("BELI_PULSA" + inputi)){
        pulsa += inputi;
        out.println("Pembelian pulsa Rp. " + inputs + " berhasil. Pulsa saat ini Rp. " + pulsa);

}
```

Bila nilai inputi ada isikan 0 (inputi = 0). Berarti command tersebut harus dibuat/ dipanggil “BELI_PULSA0”, bukan beli PULSA. Andai nilai inputi = 100. Berarti commandnya “BELI_PULSA100”.

Kalau begitu if-nya bakal banyak dong, apabila nominal pulsanya tidak dibatasi apapun. Wahhh...

Oleh karena itu solusinya menggunakan 2 kali pemanggilan / input USER.
Lihat slide berikutnya.
---- *Gitu aja kok repot :D*



Ngapain ada pulsa += 10000; ??!
Berarti Anda buat mesin yang menipu donk?! Masak orang sudah input'in 50.000, dapatnya hanya 10 ribu dong? Wah.... Ckckckckck

Solusinya
Anda Casting tu besarpulsa. Mula-mula reader.readline() adalah string. Lalu convert menuju Integer dengan fungsi Integer.parseInt(.....). Nanti pulsa +=besarpulsa;

Selesai :D

COMMAND “BELI_PULSA”

MEMBACA NOMINAL PULSA

```
while (true)
{
    pesan =reader.readLine();
    if("CEK_PULSA".equals(pesan))
    {
        out.println("Sisa pulsa anda sebesar Rp. " + pulsa.Pulsa + " dan kuota
    }
    else if("BELI_PULSA".equals(pesan))
    {
        pesan = reader.readLine();
        pulsa.Pulsa += Integer.parseInt(pesan);
        out.println("Pembelian pulsa Rp. " + pesan + " berhasil. Pulsa saat in
    }
```

Implementasi yang benar

Evaluasi quiz (3)

Banyak kesalahan gara-gara salah meletakkan coding. Walaupun Anda sudah membawa buku setebal 2000 lembar, namun kalau g pernah koding, ya kaya gini deh

1. Salah meletakkan coding di try-catch
2. Salah sintaks gara-gara lupa huruf besar atau huruf kecilnya
3. Salah sintaks untuk perbandingan STRING
4. Salah penempatan HandlingSocket
5. Ilmu PBO yang kurang dewa-sa

1

Catch begin

```
public static void main(String[] args)
{
    ServerSocket sc = null;
    try {
        // TODO code application logic here

        sc = new ServerSocket(1234);
    } catch (IOException ex) {
        Logger.getLogger(SERVER.class.getName()).log(Level.SEVERE, null, ex);

     while(true){
            try {
                Socket incoming = null;
                incoming = sc.accept();

                handleSocket(incoming);

                incoming.close();;
            } catch (IOException ex1) {
                Logger.getLogger(SERVER.class.getName()).log(Level.SEVERE, null, ex1);
            }
        }
    }
```

Catch end

Ada yang aneh?! Kira-kira program ini jalan atau tidak?

Tentu tidak berjalan.. Karena lihat tutup kurung dari try-catch. Catch begin dengan statement Logger.getLogger (....). Nah mestinya langsung ditutup kurung dong, baru statement While(true) dll. Artinya kalau implementasi seperti ini, berarti saat server berhasil jalan, maka while(true) tidak dijalankan. Bila server gagal jalan, maka while(true) dijalankan. Logikanya terbalik-balik.

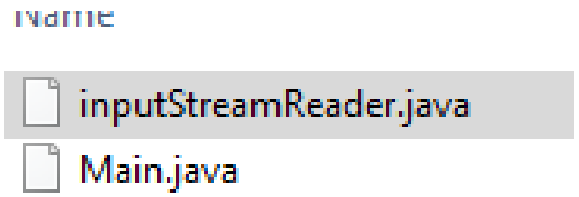
2

```
private static void handleSocket(Socket incoming) throws IOException
{
    BufferedReader reader = new BufferedReader (new inputStreamReader (incoming.getInputStream()));
    PrintStream out = new PrintStream(incoming.getOutputStream());
```

Ada yang aneh dari tulisan yang diberi tanda kuning tersebut?!

Ya tentu, itu adalah fungsi untuk inputStreamReader. Namun ada kesalahan yang fatal. *inputStreamReader* ditulis dengan ***InputStreamReader***.

Lah berarti salah dong? Merah dong? Iya.. Sampai ada yang implementasi kelas inputStreamReader yang baru.. Daebak.. Suangar.... Yahhhh.. Ternyata kelasnya kosong ☹. #sedih




```
public class inputStreamReader {

}
```

3

```
else if (str =="CEK_PULSA")
{
    output1.println("Pulsa anda =Rp. "+p.pulsa);
    output1.println("Kuota anda sebesar "+p.kuota +"Mb");
    done = true;
}
```

Bila ingin membandingkan string dengan string, HARAM hukumnya menggunakan ==. Karena bila dijalankan, nanti jadi HORROR. Tidak akan masuk ke **if** ini.

Public static Void Main

4a

Private static void

```
public static void main(String[] args) {
    // TODO code application logic here
    ServerSocket ss = null;

        ss = new ServerSocket(1234);

    while (true)
    {
       .....
    }

    private static void handleSocket(Socket incoming) throws IOException {
        BufferedReader baca = new BufferedReader(new InputStreamReader(incoming
        PrintStream out = new PrintStream(incoming.getOutputStream());
        String cmd,hitung = "";
        int pulsa=0 , kuota =0;
        while(true)
        {
            cmd = baca.readLine();
```

Ada yang aneh?!
Tentu.. handleSocket adalah sebuah fungsi dari kelas. Sehingga fungsi di dalam kelas harusnya ditulis di luar void main(). Kenapa? Karena void main sendiri adalah fungsi dari sebuah kelas

Public static Void Main

4b

Private static void

```
public class serverOperator {

    public static void main(String[] args) {
        // TODO code application logic here
        ServerSocket ss = null;

            ss = new ServerSocket(1234);

        while (true)
        {
            .....
        }
    }

    private static void handleSocket(Socket incoming) throws IOException {
            BufferedReader baca = new BufferedReader(new InputStreamReader(
            PrintStream out = new PrintStream(incoming.getOutputStream());
            String cmd,hitung = "";
            int pulsa=0 , kuota =0;
            while(true)
```

Ini implementasi handleSocket yang benar...

Hayo.. Jangan asal jiplak. **Ingat Anda bukan juru ketik, tapi insinyur pembuat sistem.**

```java
public class Pulsa {

    static Pulsa parsePulsa(int nominal) {
        throw new UnsupportedOperationException("Not supported yet.");
    }

    public int pulsa;
    public int kuota;

    public Pulsa(int saldo, int kuotaa)
    {
        pulsa = saldo;
        kuota = kuotaa;
    }


}
```

```java
Pulsa myPulsa = new Pulsa(pulsaAwal, kuotaAwal);
int Pulsaku = Integer.parseInt(myPulsa.toString());
```

Awalnya sih sudah benar. Dengan constructor dan memasukkan nilainya benar.
Namun tujuan si programmer menggambil nilai pulsa dari kelas PULSA yang diimplementasi dalam variable myPulsa.

Kelas PULSA memiliki 2 atribut pulsa dan kuota.
Semestinya int Pulsaku = myPulsa.pulsa; sudah cukup.

Ngapain kelas dicasting segala ke INT dll. Malah jadi HORROR

Inilah pentingnya belajar PBO

Evaluasi quiz (4)

Gunakan pola komunikasi dengan COMMAND. Setelah menerima command, nanti server memilah-milah dengan fungsi IF.

Jangan lupa sintaks “END”, incoming.close, while(done)

Bila pakai while(true) jangan lupa pas END, di-break.

```
boolean done = false;
while(!done){
    String clientCommand = input.readLine();

    pulsa.kuota-= 5;
    if(pulsa.kuota < 0){
        pulsa.kuota = 0;
    }

    if("CEK_PULSA".equals(clientCommand)){
            ...
    }
    else if("BELI_PULSA".equals(clientCommand)){
        ....
    }
    else if("KONVERSI_KUOTA".equals(clientCommand)){
        ....
    }
    else if("END".equals(clientCommand)){
        done = true;
        output.println("Disconnected");
    }
}

incoming.close();
```

Struktur Program di Server yang benar

Benar berarti sesuai dengan maksud soal.

```
while(true){
    msgToServer = reader.readLine();
    output.println(msgToServer);
    if("BELI_PULSA".equals(msgToServer)){
        msgToServer = reader.readLine();
        output.println(msgToServer);
    }
    msgReceive = input.readLine();
    System.out.println(msgReceive);
}
```

Struktur Program di Client yang benar

Benar berarti sesuai dengan maksud soal dan menyesuaikan dengan program yang ada di SERVER.

Dilihat dari sini, Client diimplementasi dalam bentuk while(true) dan awalnya meminta perinta msgToServer yang isinya adalah COMMAND.
Baru dibawahnya, bila BELI_PULSA, maka di-IF dan USER diminta kembali untuk memasukkan nominal pulsa.

Setelah itu baru RECEIVE Messsage dari Server.