

Tugas 1 PBO

Makassar, 19 Februari 2024

TUGAS PEMROGRAMAN BEROIENTASI OBJEK



Nama : Muh. Reza
Stambuk : 13020220070
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2024**

PRAKTIKUM 1:

1. `public class Praktikum1 { \`

- Mendefinisikan kelas Java dengan nama `Praktikum1`.
- Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- Setiap program Java harus memiliki setidaknya satu kelas yang memiliki metode `main()` sebagai titik masuk (entry point) untuk menjalankan program.

2. `public static void main(String[] args) { \`

- Mendefinisikan metode `main()`, yang merupakan titik masuk (entry point) untuk menjalankan program Java.
- Kata kunci `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini adalah metode kelas, bukan metode instance, sehingga dapat dipanggil tanpa membuat objek kelas.
- `void` adalah tipe pengembalian metode, yang berarti metode ini tidak mengembalikan nilai apapun.
- `String[] args` adalah parameter dari metode `main()`, yang digunakan untuk menerima argumen baris perintah saat program dijalankan.

3. `float f = 20.0f; \`

- Mendeklarasikan dan menginisialisasi variabel `f` dengan tipe data `float`.
- Nilai inisialisasi adalah `20.0f`, yang merupakan literal float.
- Tanda `f` pada akhir nilai literal menunjukkan bahwa nilai tersebut adalah tipe data `float`.

4. `double fl; \`

- Mendeklarasikan variabel `fl` dengan tipe data `double`.
- Variabel ini hanya dideklarasikan tanpa diinisialisasi, sehingga nilainya default akan menjadi `0.0`.

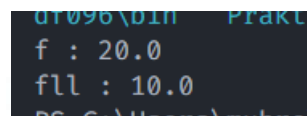
5. `fl = 10.0f; \`

- Menginisialisasi variabel `fl` dengan nilai `10.0f`.
- Meskipun `fl` adalah tipe data `double`, nilai `float` (`10.0f`) secara otomatis dikonversi menjadi `double` karena `double` memiliki rentang yang lebih besar daripada `float`.

6. `System.out.println("f : " + f + "\nfl : " + fl);`

- Mencetak nilai variabel `f` dan `fl` ke konsol.
- `System.out.println()` digunakan untuk mencetak teks diikuti oleh baris baru.
- `+` digunakan untuk menggabungkan teks dan nilai variabel.
- `\n` adalah karakter escape untuk membuat baris baru dalam output.

OUTPUT:



```
01096\bin - Prakt
f : 20.0
fl : 10.0
PS C:\Users\mubra
```

PRAKTIKUM 2 :

1. ``public class Praktikum2 {``

- Mendefinisikan kelas Java dengan nama ``Praktikum2``.
- Kata kunci ``public`` menunjukkan bahwa kelas ini dapat diakses dari luar paket.

2. ``public static void main(String[] args) {``

- Mendefinisikan metode ``main()``, yang merupakan titik masuk (entry point) untuk menjalankan program Java.
- Kata kunci ``public`` menunjukkan bahwa metode ini dapat diakses dari luar kelas.
- Kata kunci ``static`` menunjukkan bahwa metode ini adalah metode kelas, bukan metode instance, sehingga dapat dipanggil tanpa membuat objek kelas.
- ``void`` adalah tipe pengembalian metode, yang berarti metode ini tidak mengembalikan nilai apapun.
- ``String[] args`` adalah parameter dari metode ``main()``, yang digunakan untuk menerima argumen baris perintah saat program dijalankan.

3. ``int i;``

- Mendeklarasikan variabel ``i`` dengan tipe data ``int``.
- Variabel ini hanya dideklarasikan tanpa diinisialisasi, sehingga nilainya default akan menjadi ``0``.

4. ``i = 5;``

- Menginisialisasi variabel ``i`` dengan nilai ``5``.

5. ``System.out.println("Hello World");``

- Mencetak teks `"Hello World"` ke konsol.
- ``System.out.println()`` digunakan untuk mencetak teks diikuti oleh baris baru.

6. `System.out.println("ini nilai i = " + i);\`

- Mencetak nilai variabel `i` ke konsol bersama dengan teks "ini nilai i = ".
- `+` digunakan untuk menggabungkan teks dan nilai variabel. Dalam hal ini, nilai variabel `i` diubah menjadi string dan digabungkan dengan teks yang lain sebelum dicetak.

OUTPUT:

```
Hello World  
ini nilai i = 5
```

PRAKTIKUM 3:

1. `public class Praktikum3 {`

- Mendefinisikan kelas Java dengan nama `Praktikum3`.
- Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.

2. `public static void main(String[] args) {`

- Mendefinisikan metode `main()`, yang merupakan titik masuk (entry point) untuk menjalankan program Java.
- Kata kunci `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini adalah metode kelas, bukan metode instance, sehingga dapat dipanggil tanpa membuat objek kelas.
- `void` adalah tipe pengembalian metode, yang berarti metode ini tidak mengembalikan nilai apapun.
- `String[] args` adalah parameter dari metode `main()`, yang digunakan untuk menerima argumen baris perintah saat program dijalankan.

3. `short ks = 1;`

- Mendeklarasikan variabel `ks` dengan tipe data `short` dan menginisiasinya dengan nilai `1`.
- Tipe data `short` merupakan tipe data bilangan bulat yang memiliki rentang nilai lebih kecil daripada `int`.

4. `int ki = 1;`

- Mendeklarasikan variabel `ki` dengan tipe data `int` dan menginisiasinya dengan nilai `1`.
- Tipe data `int` merupakan tipe data bilangan bulat yang memiliki rentang nilai yang lebih besar daripada `short`.

5. `long kl = 10000;`

- Mendeklarasikan variabel `kl` dengan tipe data `long` dan menginisialisasinya dengan nilai `10000`.

- Tipe data `long` merupakan tipe data bilangan bulat yang memiliki rentang nilai yang lebih besar daripada `int`.

6. `char c = 65;`

- Mendeklarasikan variabel `c` dengan tipe data `char` dan menginisialisasinya dengan nilai karakter yang sesuai dengan kode Unicode `65`, yang merepresentasikan karakter `'A'`.

7. `char c1 = 'z';`

- Mendeklarasikan variabel `c1` dengan tipe data `char` dan menginisialisasinya dengan karakter `'z'`.

8. `double x = 50.2f;`

- Mendeklarasikan variabel `x` dengan tipe data `double` dan menginisialisasinya dengan nilai `50.2f`.

- Karena nilai literal `50.2f` adalah tipe data `float`, Java secara otomatis mengonversinya menjadi `double`.

9. `float y = 50.2f;`

- Mendeklarasikan variabel `y` dengan tipe data `float` dan menginisialisasinya dengan nilai `50.2f`.

10. `System.out.println("Karakter = " + c);`

- Mencetak nilai variabel `c` ke konsol bersama dengan teks `"Karakter = "`.

- `System.out.println()` digunakan untuk mencetak teks diikuti oleh baris baru.

11. `System.out.println("Bilangan integer (short) = " + ks);`

- Mencetak nilai variabel `ks` ke konsol bersama dengan teks `"Bilangan integer (short) = "`.

12. `System.out.println("\t(int) = " + ki);`

- Mencetak nilai variabel `ki` ke konsol dengan indentasi menggunakan karakter `\t` dan teks `"(int) = "`.

13. ``System.out.println("\t(long)= " + kl);``

- Mencetak nilai variabel ``kl`` ke konsol dengan indentasi menggunakan karakter ``\t`` dan teks `"(long)= "`.

14. ``System.out.println("Bilangan Real x = " + x);``

- Mencetak nilai variabel ``x`` ke konsol bersama dengan teks "Bilangan Real x =".

15. ``System.out.println("Bilangan Real y = " + y);``

- Mencetak nilai variabel ``y`` ke konsol bersama dengan teks "Bilangan Real y =".

OUTPUT:

```
Karakter = A
Karakter = z
Karakter = A
Karakter = z
Bilangan integer (short) = 1
                (int) = 1
                (long)= 10000
Bilangan Real x = 50.20000076293945
Bilangan Real y = 50.2
```


PRAKTIKUM 4:

1. `import java.util.Scanner;`

- Ini adalah pernyataan import yang mengimpor kelas `Scanner` dari paket `java.util`.
- Kelas `Scanner` digunakan untuk memindai (membaca) input dari pengguna melalui konsol.

2. `public class Praktikum4 {`

- Mendefinisikan kelas Java dengan nama `Praktikum4`.
- Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.

3. `public static void main(String[] args) {`

- Mendefinisikan metode `main()`, yang merupakan titik masuk (entry point) untuk menjalankan program Java.
- Kata kunci `public` menunjukkan bahwa metode ini dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini adalah metode kelas, bukan metode instance, sehingga dapat dipanggil tanpa membuat objek kelas.
- `void` adalah tipe pengembalian metode, yang berarti metode ini tidak mengembalikan nilai apapun.
- `String[] args` adalah parameter dari metode `main()`, yang digunakan untuk menerima argumen baris perintah saat program dijalankan.

4. `int a;`

- Mendeklarasikan variabel `a` dengan tipe data `int`.
- Variabel ini akan digunakan untuk menyimpan nilai integer yang akan dimasukkan oleh pengguna.

5. `Scanner masukan;`

- Mendeklarasikan variabel `masukan` dengan tipe data `Scanner`.
- Variabel ini akan digunakan untuk membaca input dari pengguna.

6. `System.out.print("Contoh membaca dan Menulis, ketik nilai integer: ");`

- Mencetak teks "Contoh membaca dan Menulis, ketik nilai integer: " ke konsol.
- `System.out.print()` digunakan untuk mencetak teks tanpa karakter baru diakhirinya.

7. ``masukan = new Scanner(System.in);``

- Membuat objek ``Scanner`` baru dengan menggunakan konstruktor ``Scanner(System.in)``.

- Ini memungkinkan program untuk membaca input dari ``System.in``, yang merupakan input dari konsol.

8. ``a = masukan.nextInt();``

- Memanggil metode ``nextInt()`` dari objek ``masukan`` untuk membaca sebuah integer dari pengguna dan menyimpannya ke dalam variabel ``a``.

9. ``System.out.print("Nilai yang dibaca : " + a);``

- Mencetak nilai variabel ``a`` ke konsol bersama dengan teks "Nilai yang dibaca : ".

- ``System.out.print()`` digunakan untuk mencetak teks tanpa karakter baru diakhirinya.

10. ``masukan.close();``

- Memanggil metode ``close()`` dari objek ``masukan`` untuk menutup scanner setelah selesai digunakan.

- Ini merupakan praktik yang baik untuk menutup scanner untuk mencegah kebocoran sumber daya dan menghindari memori yang tidak perlu terpakai.

OUTPUT:

```
Contoh membaca dan Menulis, ketik nilai integer: 12
Nilai yang dibaca : 12
```

PRAKTIKUM 5:

1. `import java.io.BufferedReader;`

`import java.io.InputStreamReader;`

- Mengimpor kelas ``BufferedReader`` dan ``InputStreamReader`` dari paket ``java.io``.

- ``BufferedReader`` digunakan untuk membaca teks dari input karakter dengan buffering efisien.

- ``InputStreamReader`` digunakan untuk membaca byte dari ``System.in`` dan menerjemahkannya menjadi karakter menggunakan set encoding yang diberikan atau default encoding.

2. **import javax.swing.JOptionPane;**

- Mengimpor kelas `JOptionPane` dari paket `javax.swing`.
- `JOptionPane` digunakan untuk menampilkan dialog box dengan berbagai jenis pilihan, termasuk masukan dari pengguna.

3. **import java.io.IOException;**

- Mengimpor kelas `IOException` dari paket `java.io`.
- Kelas ini digunakan untuk menangani kesalahan input/output.

4. **public class Praktikum5 {**

- Mendefinisikan kelas Java dengan nama `Praktikum5`.

5. **public static void main(String[] args) throws IOException {**

- Mendefinisikan metode `main()`, yang merupakan titik masuk (entry point) untuk menjalankan program Java.
- `throws IOException` menunjukkan bahwa metode ini dapat melempar kesalahan input/output (`IOException`), dan akan ditangani oleh pemanggil metode ini.

6. **char cc;**

int bil;

- Mendeklarasikan variabel `cc` dengan tipe data `char` dan `bil` dengan tipe data `int`.

7. **InputStreamReader isr = new InputStreamReader(System.in);**

BufferedReader dataIn = new BufferedReader(isr);

- Membuat objek `InputStreamReader` dengan menginisialisasinya dengan `System.in`.
- Membuat objek `BufferedReader` dengan menggunakan objek `InputStreamReader` sebagai input.

8. **System.out.print("hello\n");**

- Mencetak teks "hello" ke konsol diikuti dengan karakter baris baru (`\n`).

9. **System.out.print("baca 1 karakter : ");**

- Mencetak teks "baca 1 karakter : " ke konsol.

10. **cc = dataIn.readLine().charAt(0);**

- Membaca satu baris input dari pengguna menggunakan objek `BufferedReader`.
- Mengambil karakter pertama dari baris yang dibaca dan menyimpannya dalam variabel `cc`.

11. **System.out.print("baca 1 bilangan : ");**

- Mencetak teks "baca 1 bilangan : " ke konsol.

12. **bil = Integer.parseInt(dataIn.readLine());**

- Membaca satu baris input dari pengguna menggunakan objek `BufferedReader`.
- Mengonversi baris tersebut menjadi tipe data `int` dan menyimpannya dalam variabel `bil`.

13. **String kar = JOptionPane.showInputDialog("Karakter 1 : ");**

- Menampilkan dialog input menggunakan `JOptionPane` dengan teks "Karakter 1 : ".
- Mengambil masukan dari pengguna dalam bentuk string dan menyimpannya dalam variabel `kar`.

14. **System.out.println(kar);**

- Mencetak nilai variabel `kar` ke konsol diikuti dengan karakter baris baru.

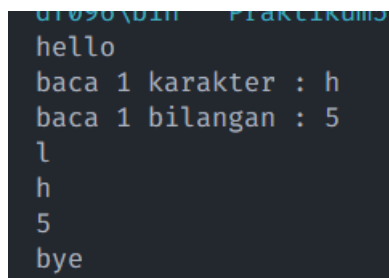
15. `System.out.print(cc + "\n" + bil + "\n");`

- Mencetak nilai variabel `cc` dan `bil` ke konsol, masing-masing diikuti dengan karakter baris baru.

16. `System.out.print("bye \n");`

- Mencetak teks "bye" ke konsol diikuti dengan karakter baris baru.

OUTPUT:



```
hello
baca 1 karakter : h
baca 1 bilangan : 5
l
h
5
bye
```

PRAKTIKUM 6:

1. `public class Praktikum6 {`

- Mendefinisikan kelas Java dengan nama `Praktikum6`.

2. `public static void main(String[] args) {`

- Mendefinisikan metode `main()`, yang merupakan titik masuk (entry point) untuk menjalankan program Java.

3. `int a = 5, b = 6;`

- Mendeklarasikan dan menginisialisasi variabel `a` dengan nilai `5` dan variabel `b` dengan nilai `6`. Keduanya memiliki tipe data `int`.

4. `float d = 2.f, e = 3.2f;`

- Mendeklarasikan dan menginisialisasi variabel `d` dengan nilai `2.f` (tipe data `float`) dan variabel `e` dengan nilai `3.2f` (tipe data `float`).

5. **char g = '5';**

- Mendeklarasikan dan menginisialisasi variabel `g` dengan nilai `'5'` (tipe data `char`). Meskipun karakter `'5'` direpresentasikan sebagai angka, dalam konteks karakter, itu masih mewakili karakter `'5'`, bukan bilangan 5.

6. **double k = 3.14;**

- Mendeklarasikan dan menginisialisasi variabel `k` dengan nilai `3.14` (tipe data `double`).

7. **System.out.println((float) a);**

- Mencetak nilai variabel `a` ke konsol setelah mengonversinya menjadi tipe data `float`.

8. **System.out.println((double) b);**

- Mencetak nilai variabel `b` ke konsol setelah mengonversinya menjadi tipe data `double`.

9. **System.out.println((int) d);**

- Mencetak nilai variabel `d` ke konsol setelah mengonversinya menjadi tipe data `int`.

10. **System.out.println((double) e);**

- Mencetak nilai variabel `e` ke konsol setelah mengonversinya menjadi tipe data `double`.

11. **System.out.println((int) g);**

- Mencetak nilai variabel `g` ke konsol setelah mengonversinya menjadi tipe data `int`. Ini akan mengonversi karakter `'5'` menjadi nilai **ASCII** yang sesuai, yaitu **53**.

12. **System.out.println((float) g);**

- Mencetak nilai variabel `g` ke konsol setelah mengonversinya menjadi tipe data `float`. Ini juga akan mengonversi karakter `'5'` menjadi nilai **ASCII** yang sesuai, yaitu **53.0f**.

13. **System.out.println((double) g);**

- Mencetak nilai variabel `g` ke konsol setelah mengonversinya menjadi tipe data `double`. Ini juga akan mengonversi karakter `'5'` menjadi nilai **ASCII** yang sesuai, yaitu **53.0**.

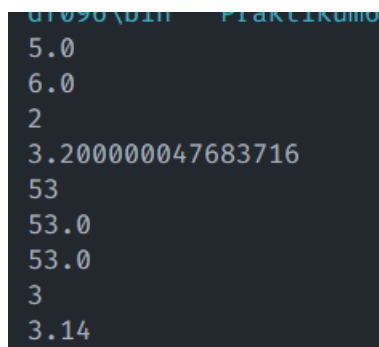
14. **System.out.println((int) k);**

- Mencetak nilai variabel `k` ke konsol setelah mengonversinya menjadi tipe data `int`. Ini akan membuang bagian desimal dari nilai `double k`, sehingga hasilnya adalah **3**.

15. **System.out.println((float) k);**

- Mencetak nilai variabel `k` ke konsol setelah mengonversinya menjadi tipe data `float`. Variabel `k` yang berisi nilai `double` akan dikonversi ke dalam tipe data `float`, tetapi dengan potensi kehilangan presisi karena `float` memiliki rentang yang lebih kecil daripada `double`.

OUTPUT:



```
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
```

PRAKTIKUM 7:

1. `public class Praktikum7 { ... }`:

Ini adalah deklarasi kelas Java dengan nama "**Praktikum7**". Setiap program Java harus memiliki setidaknya satu kelas dengan nama yang sama dengan nama file Java-nya.

2. `public static void main(String[] args) { ... }`:

Ini adalah metode utama (main method) yang akan dieksekusi pertama kali ketika program dijalankan. Metode ini memiliki aksesibilitas publik (public), bersifat statis (static), tidak mengembalikan nilai (**void**), dan menerima argumen dalam bentuk array string (**String[] args**).

3. `int a = 8, b = 9;`:

Mendeklarasikan dua variabel integer, **a** dan **b**, dan memberi nilai awal masing-masing 8 dan 9.

4. `float d = 2.f, e = 3.2f;`: Mendeklarasikan dua variabel float, **d** dan **e**, dan memberi nilai awal masing-masing 2.0 dan 3.2.

5. `char g = '5';`:

Mendeklarasikan sebuah variabel karakter (**char**) dengan nama **g** dan memberi nilai awal '5'.

6. `double k = 3.14;`:

Mendeklarasikan sebuah variabel double dengan nama **k** dan memberi nilai awal 3.14.

7. `String n = "67", m = "45", l = "100";`:

Mendeklarasikan tiga variabel string, **n**, **m**, dan **l**, dan memberi nilai awal masing-masing "67", "45", dan "100".

8. **``a = Integer.parseInt(n);``**:

Mengubah nilai variabel ``a`` menjadi hasil dari parsing (mengubah) string ``n`` menjadi integer.

9. **``k = Double.parseDouble(m);``**:

Mengubah nilai variabel ``k`` menjadi hasil dari parsing string ``m`` menjadi double.

10. **``d = Float.parseFloat(l);``**:

Mengubah nilai variabel ``d`` menjadi hasil dari parsing string ``l`` menjadi float.

11. **``System.out.println("a : " + a + "\nk : " + k + "\nd : " + d);``**:

Mencetak nilai variabel ``a``, ``k``, dan ``d`` ke konsol dengan pesan yang sesuai.

12. **``n = String.valueOf(b);``**:

Mengubah nilai variabel ``n`` menjadi representasi string dari nilai variabel ``b``.

13. **``m = String.valueOf(g);``**:

Mengubah nilai variabel ``m`` menjadi representasi string dari nilai variabel ``g``.

14. **``l = String.valueOf(e);``**: Mengubah nilai variabel ``l`` menjadi representasi string dari nilai variabel ``e``.

15. **``k = Double.valueOf(a).intValue();``**:

Mengubah nilai variabel ``k`` menjadi representasi integer dari nilai variabel ``a``.

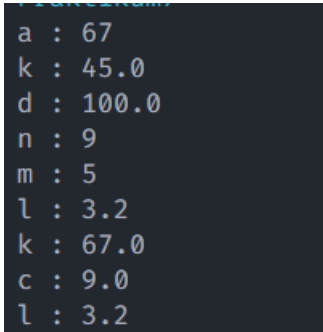
16. **``double c = Integer.valueOf(b).doubleValue();``**:

Mengubah nilai variabel ``c`` menjadi representasi double dari nilai variabel ``b``.

17. `System.out.println("k : " + k + "\nc : " + c + "\nl : " + l);`:

Mencetak nilai variabel `k`, `c`, dan `l` ke konsol dengan pesan yang sesuai.

OUTPUT:



```
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
```

PRAKTIKUM 8:

1. `public class Praktikum8 { ... }`:

- Ini adalah definisi kelas Java dengan nama `Praktikum8`.
- Kata kunci `public` menunjukkan bahwa kelas ini dapat diakses dari luar paket.
- Program Java selalu dimulai dengan definisi kelas.

2. `public static void main(String[] args) { ... }`:

- Ini adalah metode utama (main method) yang merupakan titik awal dari eksekusi program.
- Metode ini didefinisikan sebagai `public`, sehingga dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini bisa dipanggil tanpa membuat objek dari kelasnya.
- `void` menunjukkan bahwa metode ini tidak mengembalikan nilai apa pun.
- `String[] args` adalah parameter yang diterima metode main, yang berisi argumen baris perintah saat program dijalankan.

3. `int x = 1;` dan `int y = 2;`:

- Mendeklarasikan dua variabel bertipe `int` yang diberi nama `x` dan `y` dan menginisiasinya dengan nilai `1` dan `2` secara berturut-turut.

4. ``System.out.print("x = " + x + "\n");`` dan ``System.out.print("y = " + y + "\n");``:

- Menampilkan nilai variabel ``x`` dan ``y`` ke output standar (biasanya konsol).
- ``System.out.print`` digunakan untuk mencetak teks ke konsol.
- ``+`` digunakan untuk menggabungkan teks dengan nilai variabel.
- ``\n`` digunakan untuk membuat baris baru setelah mencetak nilai variabel.

5. ``((x < y) ? x : y)``:

- Ini adalah ekspresi kondisional (ternary) yang membandingkan nilai ``x`` dan ``y``.
- Jika kondisi ``(x < y)`` benar (true), maka ekspresi mengembalikan nilai ``x``, jika tidak, ekspresi mengembalikan nilai ``y``.

6. ``System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));``:

- Menampilkan hasil dari ekspresi kondisional ``(x < y) ? x : y`` ke output standar.
- Ini juga menggunakan ``System.out.print`` untuk mencetak hasilnya.

OUTPUT:

```
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
```

PRAKTIKUM 9:

Berikut adalah penjelasan untuk masing-masing sintaks dalam kode yang Anda berikan:

1. ``public class Praktikum9 { ... }``:

- Ini adalah definisi kelas Java dengan nama ``Praktikum9``.
- Program Java selalu dimulai dengan definisi kelas.

2. `public static void main(String[] args) { ... }`:

- Ini adalah metode utama (main method) yang merupakan titik awal dari eksekusi program.
- Metode ini didefinisikan sebagai `public`, sehingga dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini bisa dipanggil tanpa membuat objek dari kelasnya.
- `void` menunjukkan bahwa metode ini tidak mengembalikan nilai apa pun.
- `String[] args` adalah parameter yang diterima metode main, yang berisi argumen baris perintah saat program dijalankan.

3. `int x = 1; dan int y = 2;`:

- Mendeklarasikan dua variabel bertipe `int` yang diberi nama `x` dan `y` dan menginisialisasinya dengan nilai `1` dan `2` secara berturut-turut.

4. `float fx; dan float fy;`:

- Mendeklarasikan dua variabel bertipe `float` yang diberi nama `fx` dan `fy` tanpa memberikan nilai awal.

5. `System.out.print("x/y (format integer) = " + x / y);`:

- Mencetak hasil dari pembagian `x` dengan `y` (format integer) ke output standar.
- Pada kasus ini, hasilnya adalah `0` karena pembagian dua bilangan bulat akan menghasilkan bilangan bulat.

6. `System.out.print("\nx/y (format float) = " + x / y);`:

- Mencetak hasil dari pembagian `x` dengan `y` (format float) ke output standar.
- Pada kasus ini, hasilnya juga `0` karena pembagian dua bilangan bulat akan menghasilkan bilangan bulat.

7. ``fx = x;` dan `fy = y;`:`

- Mengkonversi variabel ``x`` dan ``y`` (yang bertipe ``int``) ke dalam variabel ``fx`` dan ``fy`` (yang bertipe ``float``).

8. ``System.out.print("\nx/y (format integer) = " + fx / fy);` dan
`System.out.print("\nx/y (format float) = " + fx / fy);`:`

- Mencetak hasil dari pembagian ``fx`` dengan ``fy`` (format integer dan float) ke output standar.

- Pada kasus ini, pembagian dilakukan dengan tipe data float sehingga hasilnya adalah nilai yang benar, tidak seperti sebelumnya yang menggunakan tipe data integer.

9. ``System.out.print("\nfloat(x)/float(y) (format integer)= " + (float) x / (float) y);` dan
`System.out.print("\nfloat(x)/float(y) (format float) = " + (float) x / (float) y);`:`

- Mencetak hasil dari pembagian dua variabel bertipe ``float`` ``(float) x / (float) y`` (format integer dan float) ke output standar.

- Melakukan konversi terlebih dahulu dari tipe data ``int`` ke ``float`` menggunakan ``(float)``.

10. ``x = 10;` dan `y = 3;`:`

- Mengubah nilai variabel ``x`` menjadi ``10`` dan variabel ``y`` menjadi ``3``.

11. ``System.out.print("\nx/y (format integer) = " + x / y);` dan
`System.out.print("\nx/y (format float) = " + x / y);`:`

- Mencetak hasil dari pembagian ``x`` dengan ``y`` (format integer dan float) ke output standar.

- Seperti sebelumnya, pembagian dengan tipe data integer akan menghasilkan hasil integer, sedangkan pembagian dengan tipe data float akan menghasilkan hasil float.

OUTPUT:

```
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer)= 0.5
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float) = 3
```

PRAKTIKUM 10:

1. `public class Praktikum10 { ... }`:

- Ini adalah definisi kelas Java dengan nama `Praktikum10`.
- Program Java selalu dimulai dengan definisi kelas.

2. `public static void main(String[] args) { ... }`:

- Ini adalah metode utama (main method) yang merupakan titik awal dari eksekusi program.
- Metode ini didefinisikan sebagai `public`, sehingga dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini bisa dipanggil tanpa membuat objek dari kelasnya.
- `void` menunjukkan bahwa metode ini tidak mengembalikan nilai apa pun.
- `String[] args` adalah parameter yang diterima metode main, yang berisi argumen baris perintah saat program dijalankan.

3. `System.out.print("Hello");`:

- Mencetak teks "Hello" ke output standar (biasanya konsol) tanpa membuat baris baru.
- `System.out.print` digunakan untuk mencetak teks ke konsol.

4. ``System.out.print("\nHello ");``:

- Mencetak teks "Hello " ke output standar (biasanya konsol) dengan membuat baris baru sebelum mencetak.
- ``\n`` digunakan untuk membuat baris baru sebelum mencetak teks "Hello".

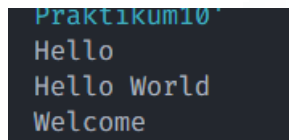
5. ``System.out.println("World");``:

- Mencetak teks "World" ke output standar (biasanya konsol) dengan membuat baris baru setelah mencetak.
- ``System.out.println`` digunakan untuk mencetak teks ke konsol dan membuat baris baru setelah mencetak.

6. ``System.out.println("Welcome");``:

- Mencetak teks "Welcome" ke output standar (biasanya konsol) dengan membuat baris baru setelah mencetak.
- Seperti sebelumnya, ``System.out.println`` digunakan untuk mencetak teks ke konsol dan membuat baris baru setelah mencetak.

OUTPUT:



```
Praktikum10
Hello
Hello World
Welcome
```

PRAKTIKUM 11:

Pada kode yang Anda berikan, berikut adalah penjelasan untuk masing-masing sintaks:

1. ``public class Praktikum11 { ... }``:

- Ini adalah definisi kelas Java dengan nama ``Praktikum11``.
- Program Java selalu dimulai dengan definisi kelas.

2. ``public static void main(String[] args) { ... }``:

- Ini adalah metode utama (main method) yang merupakan titik awal dari eksekusi program.

- Metode ini didefinisikan sebagai `public`, sehingga dapat diakses dari luar kelas.
- Kata kunci `static` menunjukkan bahwa metode ini bisa dipanggil tanpa membuat objek dari kelasnya.
- `void` menunjukkan bahwa metode ini tidak mengembalikan nilai apa pun.
- `String[] args` adalah parameter yang diterima metode main, yang berisi argumen baris perintah saat program dijalankan.

3. `int i, j;`

- Mendeklarasikan dua variabel bertipe `int` yang diberi nama `i` dan `j` tanpa memberikan nilai awal.

4. `i = 3;`

- Memberikan nilai `3` ke variabel `i`.

5. `j = i++;`

- Memasukkan nilai `i` ke dalam `j`, kemudian menambahkan nilai `1` ke `i`.
- Ini adalah contoh operator postfix increment, yang artinya nilai `i` akan ditingkatkan setelah nilai telah dimasukkan ke dalam `j`.

6. `System.out.println("Nilai i : " + (++i) + "\nNilai j : " + j);`

- Mencetak nilai `i` dan `j` ke output standar (biasanya konsol).
- `++i` adalah contoh operator prefix increment, yang artinya nilai `i` akan ditingkatkan sebelum digunakan dalam ekspresi.
- Dua nilai tersebut dicetak bersamaan dengan pesan "Nilai i : " dan "Nilai j : ".

OUTPUT:

```

14:03:10
Nilai i : 5
Nilai j : 3

```


PRAKTIKUM 12:

1. `public class Praktikum12 { ... }`:

Ini adalah deklarasi kelas Java dengan nama `Praktikum12`. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari luar paket. Kelas ini berisi method `main` yang akan dieksekusi saat program dijalankan.

2. `public static void main(String[] args) { ... }`:

Ini adalah method utama yang dieksekusi saat program dijalankan. Method ini memiliki kata kunci `public`, sehingga dapat diakses dari luar kelas. Kata kunci `static` menandakan bahwa method ini terkait dengan kelas itu sendiri, bukan dengan instance kelas. `void` menunjukkan bahwa method ini tidak mengembalikan nilai. Parameter `String[] args` adalah argumen yang dapat diterima oleh program saat dijalankan.

3. `int n = 10;`, `int x = 1;`, `int y = 2;`:

Ini adalah deklarasi dan inisialisasi variabel bertipe integer `n`, `x`, dan `y` dengan nilai masing-masing 10, 1, dan 2.

4. `System.out.println("n = " + n);`:

Ini mencetak nilai variabel `n` ke konsol dengan menggunakan `System.out.println()`. Tanda `+` digunakan untuk menggabungkan teks string dengan nilai variabel.

5. `System.out.println("x = " + x);`, `System.out.println("y = " + y);`:

Ini mencetak nilai variabel `x` dan `y` ke konsol dengan cara yang sama seperti pada langkah sebelumnya.

6. `System.out.println("n & 8 = " + (n & 8));`:

Ini melakukan operasi bitwise AND antara `n` dan 8, dan mencetak hasilnya ke konsol. Operator bitwise AND (`&`) membandingkan setiap bit dari kedua operand dan menghasilkan 1 hanya jika kedua bit yang bersesuaian adalah 1.

7. `System.out.println("x & ~ 8 = " + (x & ~8));`:

Ini melakukan operasi bitwise AND antara `x` dan komplement dari 8 (`~8`), kemudian mencetak hasilnya ke konsol. Operator bitwise NOT (`~`) akan mengubah setiap bit menjadi kebalikannya.

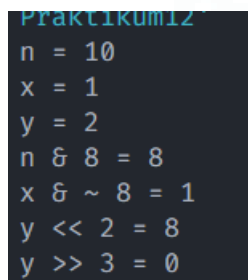
8. `System.out.println("y << 2 = " + (y << 2));`:

Ini melakukan operasi bitwise shift kiri pada `y` sebanyak 2 bit, kemudian mencetak hasilnya ke konsol. Operator shift kiri (`<<`) menggeser semua bit nilai operand ke kiri sebanyak jumlah bit yang ditentukan.

9. `System.out.println("y >> 3 = " + (y >> 3));`:

Ini melakukan operasi bitwise shift kanan pada `y` sebanyak 3 bit, kemudian mencetak hasilnya ke konsol. Operator shift kanan (`>>`) menggeser semua bit nilai operand ke kanan sebanyak jumlah bit yang ditentukan.

OUTPUT:



```
Praktikum12
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
```

PRAKTIKUM 13:

1. `public class Praktikum13 { ... }`:

Ini adalah deklarasi kelas Java dengan nama `Praktikum13`. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari luar paket. Kelas ini berisi method `main` yang akan dieksekusi saat program dijalankan.

2. `public static void main(String[] args) { ... }`:

Ini adalah method utama yang dieksekusi saat program dijalankan. Method ini memiliki kata kunci `public`, sehingga dapat diakses dari luar kelas. Kata kunci `static` menandakan bahwa method ini terkait dengan kelas itu sendiri, bukan dengan instance kelas. `void` menunjukkan bahwa method ini tidak mengembalikan nilai. Parameter `String[] args` adalah argumen yang dapat diterima oleh program saat dijalankan.

3. **``char i, j;``:**

Ini adalah deklarasi variabel ``i`` dan ``j`` bertipe karakter.

4. **``i = 3; j = 4;``:**

Ini adalah inisialisasi variabel ``i`` dengan nilai 3 dan variabel ``j`` dengan nilai 4.

5. **``System.out.println("i = " + (int) i);``:**

Ini mencetak nilai variabel ``i`` ke konsol setelah dikonversi menjadi tipe data integer dengan menggunakan ``(int) i``. ``(int)`` adalah operasi casting untuk mengubah tipe data.

6. **``System.out.println("j = " + j);``:**

Ini mencetak nilai variabel ``j`` ke konsol.

7. **``System.out.println("i & j = " + (i & j));``:**

Ini melakukan operasi bitwise AND antara ``i`` dan ``j``, kemudian mencetak hasilnya ke konsol. Operator bitwise AND (``&``) membandingkan setiap bit dari kedua operand dan menghasilkan 1 hanya jika kedua bit yang bersesuaian adalah 1.

8. **``System.out.println("i | j = " + (i | j));``:**

Ini melakukan operasi bitwise OR antara ``i`` dan ``j``, kemudian mencetak hasilnya ke konsol. Operator bitwise OR (``|``) membandingkan setiap bit dari kedua operand dan menghasilkan 1 jika salah satu atau kedua bit yang bersesuaian adalah 1.

9. **``System.out.println("i ^ j = " + (i ^ j));``:**

Ini melakukan operasi bitwise XOR antara ``i`` dan ``j``, kemudian mencetak hasilnya ke konsol. Operator bitwise XOR (``^``) membandingkan setiap bit dari kedua operand dan menghasilkan 1 hanya jika hanya satu dari kedua bit yang bersesuaian adalah 1.

10. `System.out.println(Math.pow(i, j));`:

Ini mencetak hasil dari pangkat `i` pangkat `j` menggunakan fungsi `Math.pow()`. Namun, perlu dicatat bahwa dalam konteks operasi bitwise, fungsi pangkat tidak relevan.

11. `System.out.println(~i = " + ~i);`:

Ini melakukan operasi bitwise NOT pada `i`, kemudian mencetak hasilnya ke konsol. Operator bitwise NOT (`~`) akan mengubah setiap bit menjadi kebalikannya.

OUTPUT:

```
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
```

PRAKTIKUM 14:

1. `public class Praktikum14 { ... }`:

Ini adalah deklarasi kelas Java dengan nama `Praktikum14`. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari luar paket. Kelas ini berisi method `main` yang akan dieksekusi saat program dijalankan.

2. `public static void main(String[] args) { ... }`:

Ini adalah method utama yang dieksekusi saat program dijalankan. Method ini memiliki kata kunci `public`, sehingga dapat diakses dari luar kelas. Kata kunci `static` menandakan bahwa method ini terkait dengan kelas itu sendiri, bukan dengan instance kelas. `void` menunjukkan bahwa method ini tidak mengembalikan nilai. Parameter `String[] args` adalah argumen yang dapat diterima oleh program saat dijalankan.

3. `if (true && true) { ... }`:

Ini adalah pernyataan kondisional `if` yang memeriksa apakah kedua ekspresi di dalamnya bernilai `true`. Karena keduanya benar, maka blok kode di dalamnya dieksekusi. Operator `&&` adalah operator logika AND yang menghasilkan `true` jika kedua operannya benar.

4. ``System.out.println(true && true);``:

Ini mencetak hasil dari ekspresi logika ``true && true`` ke konsol. Karena keduanya benar, hasilnya adalah ``true``.

5. ``if (true & true) { ... }``:

Ini adalah pernyataan kondisional ``if`` yang menggunakan operator bitwise AND (``&``) untuk memeriksa apakah kedua ekspresi di dalamnya benar. Meskipun hasil dari operasi bitwise AND adalah ``true``, namun karena ini adalah operator bitwise, maka semua kondisi yang diuji akan dievaluasi terlepas dari hasilnya. Oleh karena itu, blok kode di dalamnya dieksekusi.

6. ``System.out.println(true & false);``:

Ini mencetak hasil dari ekspresi bitwise ``true & false`` ke konsol. Karena operasi bitwise AND membandingkan bit-bit dari operandnya, hasilnya adalah ``false``.

7. ``if (true) { ... }``:

Ini adalah pernyataan kondisional ``if`` yang memeriksa apakah ekspresinya benar. Karena ekspresinya adalah ``true``, maka blok kode di dalamnya dieksekusi.

8. ``System.out.println(true);``:

Ini mencetak nilai ``true`` ke konsol.

9. ``if (true || true) { ... }``:

Ini adalah pernyataan kondisional ``if`` yang menggunakan operator logika OR (``||``) untuk memeriksa apakah salah satu dari kedua ekspresi di dalamnya benar. Karena minimal salah satunya benar, blok kode di dalamnya dieksekusi.

10. ``if (true | false) { ... }``:

Ini adalah pernyataan kondisional ``if`` yang menggunakan operator bitwise OR (``|``) untuk memeriksa apakah salah satu dari kedua ekspresi di dalamnya benar. Karena hasil dari operasi bitwise OR adalah ``true``, maka blok kode di dalamnya dieksekusi.

OUTPUT:

```
Praktikum15
true
false
true
true
true
```

PRAKTIKUM 15:

1. `public class Praktikum15 { ... }`:

Ini adalah deklarasi kelas Java dengan nama `Praktikum15`. Kata kunci `public` menandakan bahwa kelas ini dapat diakses dari luar paket. Kelas ini berisi method `main` yang akan dieksekusi saat program dijalankan.

2. `public static void main(String[] args) { ... }`:

Ini adalah method utama yang dieksekusi saat program dijalankan. Method ini memiliki kata kunci `public`, sehingga dapat diakses dari luar kelas. Kata kunci `static` menandakan bahwa method ini terkait dengan kelas itu sendiri, bukan dengan instance kelas. `void` menunjukkan bahwa method ini tidak mengembalikan nilai. Parameter `String[] args` adalah argumen yang dapat diterima oleh program saat dijalankan.

3. `int i = 0; int j = 0; char c = 8; char d = 10;`:

Ini adalah deklarasi dan inisialisasi variabel bertipe integer `i` dan `j`, serta variabel bertipe karakter `c` dan `d`.

4. `int e = (((int) c > (int) d) ? c : d);`:

Ini adalah ekspresi ternary yang menghasilkan nilai variabel `e` berdasarkan perbandingan nilai variabel `c` dan `d`. Jika nilai `c` lebih besar dari nilai `d`, maka nilai `e` akan sama dengan nilai `c`, jika tidak, maka nilai `e` akan sama dengan nilai `d`.

5. `int k = ((i > j) ? i : j);`:

Ini juga merupakan ekspresi ternary yang menghasilkan nilai variabel `k` berdasarkan perbandingan nilai variabel `i` dan `j`. Jika nilai `i` lebih besar dari nilai `j`, maka nilai `k` akan sama dengan nilai `i`, jika tidak, maka nilai `k` akan sama dengan nilai `j`.

6. `System.out.print("Nilai e = " + e);`:

Ini mencetak nilai variabel `e` ke konsol dengan menggunakan `System.out.print()`.

7. `System.out.print("\nNilai k = " + k);`:

Ini mencetak nilai variabel `k` ke konsol dengan menggunakan `System.out.print()`, dengan menambahkan karakter baris baru (`\n`) sebelum mencetak nilai tersebut.

8. `k = ((i++ > j++) ? i : j);`:

Ini adalah ekspresi ternary yang menghasilkan nilai variabel `k` berdasarkan perbandingan nilai variabel `i` dan `j`, setelah itu nilai `i` dan `j` akan ditambah 1 (`i++` dan `j++`). Jika nilai `i` sebelum ditambah lebih besar dari nilai `j` sebelum ditambah, maka nilai `k` akan sama dengan nilai `i` setelah penambahan, jika tidak, maka nilai `k` akan sama dengan nilai `j` setelah penambahan.

OUTPUT:

```
Praktikum15
Nilai e = 10
Nilai k = 0
Nilai k = 4
```

PRAKTIKUM 16:

1. `public class Praktikum16 { ... }`:

Mendefinisikan kelas Java dengan nama `Praktikum16`.

2. **``public static void main(String[] args) { ... }``:**

Method utama yang akan dieksekusi saat program dijalankan. Method ini memiliki kata kunci ``public``, sehingga bisa diakses dari luar kelas. ``static`` menandakan bahwa method ini terkait dengan kelas itu sendiri, bukan instance kelas. ``void`` menunjukkan bahwa method ini tidak mengembalikan nilai. Parameter ``String[] args`` adalah argumen yang bisa diterima oleh program saat dijalankan.

3. **``boolean Bool1, Bool2, TF, res;``:**

Mendeklarasikan empat variabel bertipe boolean yaitu ``Bool1``, ``Bool2``, ``TF``, dan ``res``. Variabel-variabel ini digunakan untuk menyimpan nilai boolean.

4. **``int i = 5, j = 2, hsl;``:**

Mendeklarasikan tiga variabel bertipe integer yaitu ``i``, ``j``, dan ``hsl``, dan menginisialisasi ``i`` dengan nilai 5 dan ``j`` dengan nilai 2. Variabel ``hsl`` digunakan untuk menyimpan hasil operasi aritmatika.

5. **``float x = 5, y = 5;``:**

Mendeklarasikan dua variabel bertipe float yaitu ``x`` dan ``y``, dan menginisialisasi keduanya dengan nilai 5. Variabel-variabel ini digunakan untuk menyimpan nilai float.

6. **``System.out.println("Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program");``:**

Mencetak teks instruksi ke layar.

7. **``Bool1 = true; Bool2 = false;``:**

Menginisialisasi variabel ``Bool1`` dengan nilai ``true`` dan variabel ``Bool2`` dengan nilai ``false``.

9. Langkah-langkah 8 hingga 14 adalah operasi-operasi boolean seperti AND, OR, NOT, dan XOR yang kemudian hasilnya dicetak ke layar.

10. Langkah-langkah 15 hingga 33 adalah operasi-operasi aritmatika dan relasional pada tipe data integer dan float yang kemudian hasilnya dicetak ke layar.

OUTPUT:

```
Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program
Bool1 && Bool2 = false
Bool1 || Bool2 = true
!Bool1 = false
Bool1 ^ Bool2 = true
i + j = 7
i - j = 3
i / j = 2
i * j = 10
i / j (pembagian bulat) = 2
i % j (sisa modulo) = 1
i == j : false
i != j : true
i < j : false
i > j : true
i <= j : false
i >= j : true
x != y : false
x < y : false
x > y : false
x <= y : true
x >= y : true
```

