

PEMROGRAMAN BERORIENTASI OBJEK

(Alun Sujjada, S.Kom, M.T)

Nama : Phina Putri Amalia

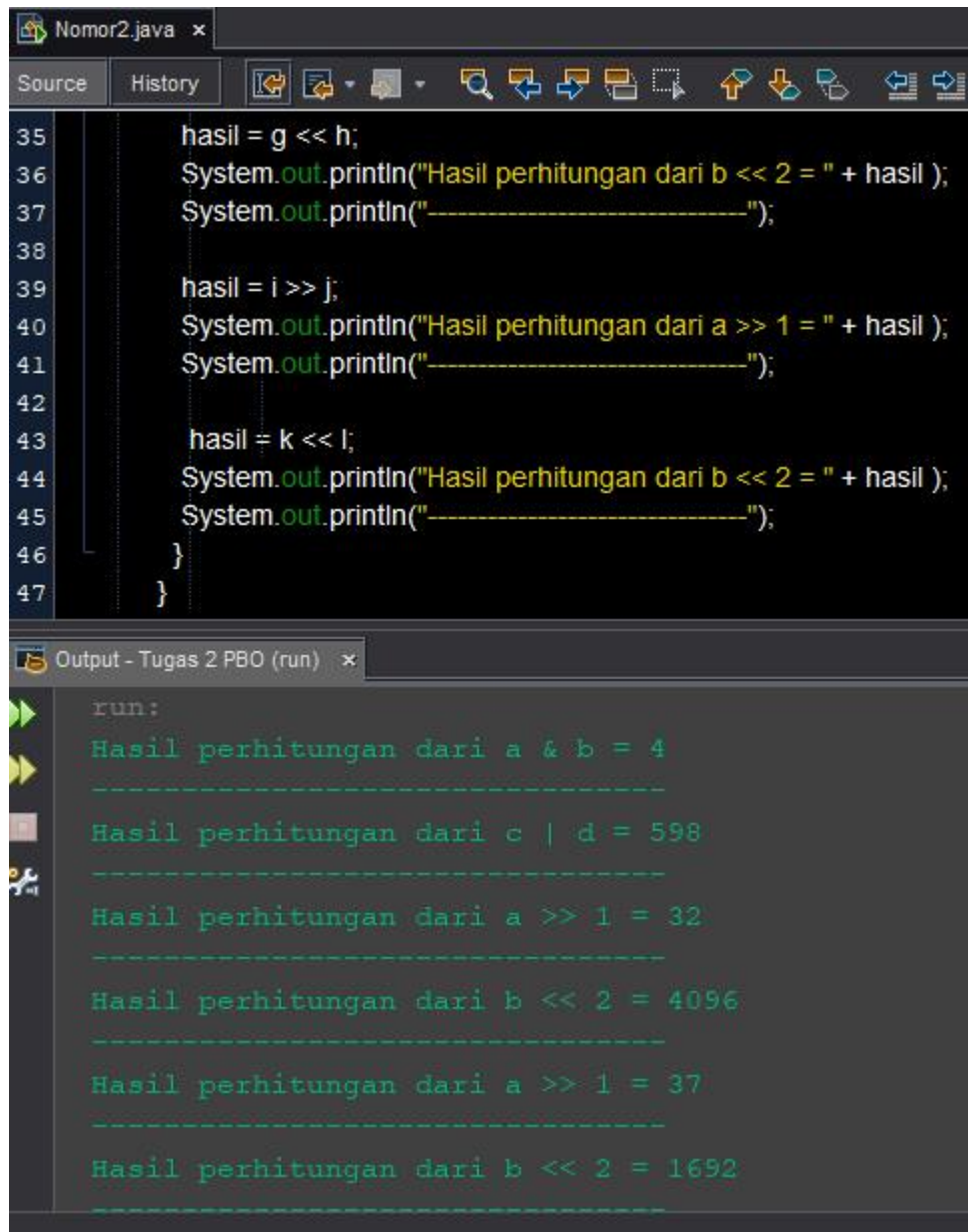
Kelas : Reg TI20A

NIM : 20200040063

Tugas 2 : PBO

A. Source Code dan Output

```
Nomor2.java x
Source History
6 public class Nomor2 {
7     public static void main(String args[]){
8
9         int a = 100;
10        int b = 20;
11        int c = 528;
12        int d = 70;
13        int e = 256;
14        int f = 3;
15        int g = 128;
16        int h = 5;
17        int i = 600;
18        int j = 4;
19        int k = 423;
20        int l = 2;
21        int hasil;
22
23        hasil = a & b;
24        System.out.println("Hasil perhitungan dari a & b = " + hasil );
25        System.out.println("-----");
26
27        hasil = c | d;
28        System.out.println("Hasil perhitungan dari c | d = " + hasil );
29        System.out.println("-----");
30
31        hasil = e >> f;
32        System.out.println("Hasil perhitungan dari a >> 1 = " + hasil );
33        System.out.println("-----");
34    }
```



The screenshot shows a Java IDE with a file named 'Nomor2.java'. The source code is as follows:

```
35     hasil = g << h;
36     System.out.println("Hasil perhitungan dari b << 2 = " + hasil );
37     System.out.println("-----");
38
39     hasil = i >> j;
40     System.out.println("Hasil perhitungan dari a >> 1 = " + hasil );
41     System.out.println("-----");
42
43     hasil = k << l;
44     System.out.println("Hasil perhitungan dari b << 2 = " + hasil );
45     System.out.println("-----");
46 }
47 }
```

The output window, titled 'Output - Tugas 2 PBO (run)', displays the following results:

```
run:
Hasil perhitungan dari a & b = 4
-----
Hasil perhitungan dari c | d = 598
-----
Hasil perhitungan dari a >> 1 = 32
-----
Hasil perhitungan dari b << 2 = 4096
-----
Hasil perhitungan dari a >> 1 = 37
-----
Hasil perhitungan dari b << 2 = 1692
-----
```

Penjelasan :

1. Operasi bitwise “and” akan memproses bit per bit dari kedua variabel, jika kedua bit sama-sama 1, maka hasilnya juga 1, selain kondisi tersebut, nilai akhirnya adalah 0.

2. Operasi $|$ (OR), hasilnya akan bernilai 0 jika kedua bit bernilai 0, selain itu nilai bit akan di set menjadi 1.
3. Operator **shift right** “>>” dimana bahasa Java akan menggeser posisi bit dalam variabel a ke kanan sebanyak 1 tempat. Selain itu Operator shift right menggeser nilai biner variabel a ke arah kanan, sehingga digit paling kanan akan dihapus. Operator shift right ini akan menghasilkan nilai asal / 2.
4. Operator **shift left** “<<” dimana nilai variabel b akan digeser sebanyak 2 digit ke kiri. Dan selain itu Ketika hasil pergeseran ke kiri, digit paling kanan akan diisi angka 0. Setiap penggeseran 1 tempat ke kiri akan mengkali 2 nilai asal.

B. Ilustrasi operator bitwise nomor 2

a. $100 \& 20$

Perbandingan akan berlangsung dari bit ke bit (angka ke angka), membandingkan berdasarkan sifat dari AND yaitu

untuk menghasilkan nilai 1 (true) kedua operand harus bernilai 1(true) jika tidak akan menghasilkan nilai 0 (false). Dan setelah perbandingan selesai dilakukan, hasil dari perbandingan akan diubah ke bentuk bilangan desimal.

$$01100100 = 100$$

$$00010100 = 20$$

----- AND

$$00000100 = 4$$

b. $528 \mid 70$

Perbandingan akan berlangsung dari bit ke bit (angka ke angka), membandingkan berdasarkan sifat dari OR (\mid) yaitu untuk mendapatkan nilai 1 (true) maka salah satu atau semua operand harus bernilai 1 (true), jika semua operand bernilai 0 (false) maka akan mendapatkan nilai 0 (false).

$$1000010000 = 528$$

$$0001000110 = 70$$

----- OR

$$1001010110 = 598$$

c. $256 \gg 3$

Operator Bitwise Shift Right (\gg) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kanan. Operator ini akan mengubah bilangan desimal menjadi bilangan biner setelah itu menggeser angka dari bilangan biner tersebut ke kanan dan pada akhirnya akan mengubahnya kembali ke bilangan desimal.

$100000000 = 256$

$\gg 3$

----- SHIFT RIGHT

$000100000 = 32$

d. $128 \ll 5$

Operator Bitwise Shift Left (\ll) adalah operator yang akan menggeser nilai dalam bentuk bilangan biner ke kiri. Operator ini akan mengubah bilangan desimal menjadi bilangan biner lalu menggeser angka dari bilangan biner tersebut ke kiri dan setelah itu akan mengubahnya kembali ke bilangan desimal.

$100000000 = 128$

$\ll 5$

----- SHIFT LEFT

$10000000000000 = 4096$

e. $600 \gg 4$

$1001011000 = 600$

$\gg 4$

----- SHIFT RIGHT

$0000100101 = 37$

f. $423 \ll 2$

$110100111 = 423$

$\ll 2$

----- SHIFT LEFT

$11010011100 = 1692$