

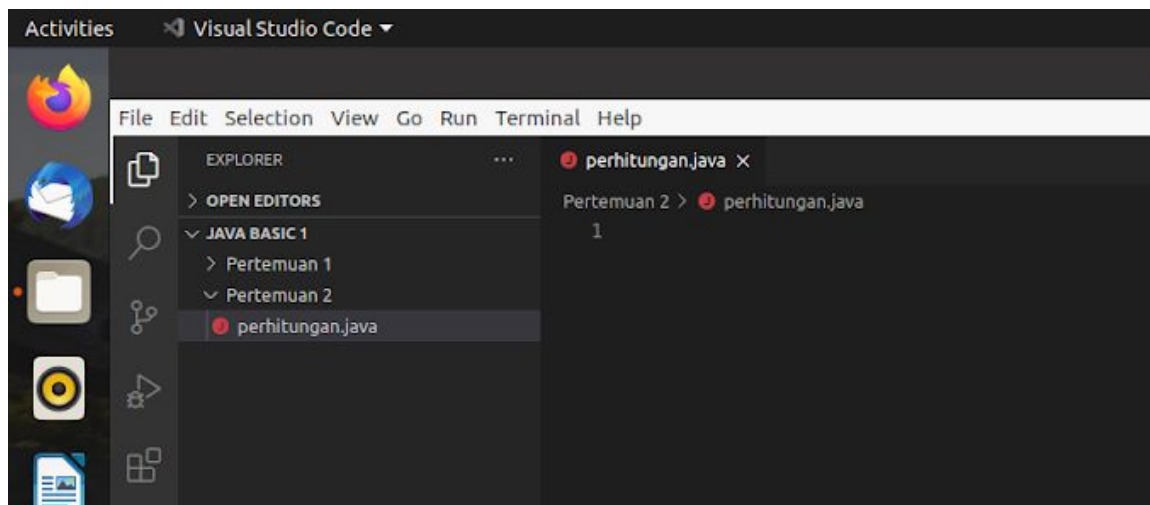
Modul Pemrograman Berbasis Objek (PBO)

Java Basic - Variabel, Operator, Tipe Data Part 1

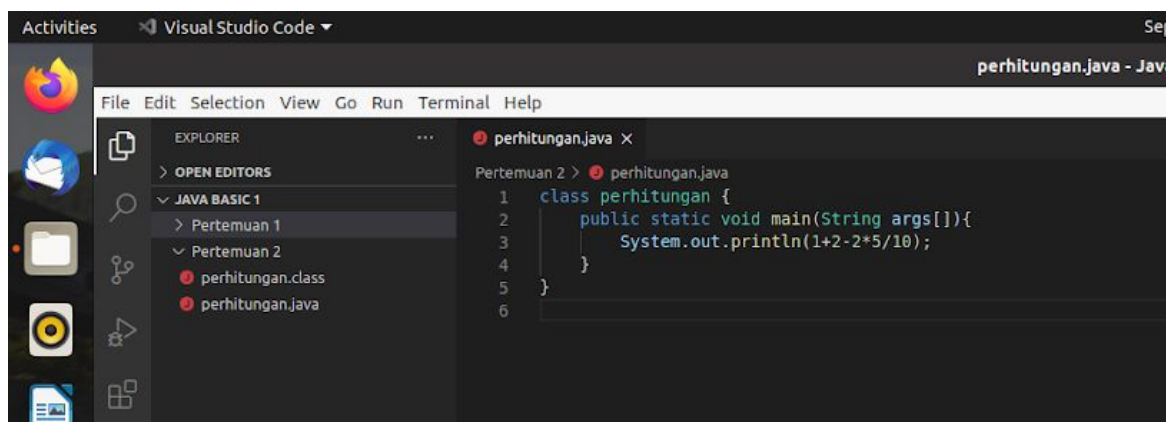
Pada modul kali ini kita akan belajar mengenai variable, operator, dan tipe data yang terdapat pada java

Sebelum kita membahas ke 3 hal tersebut, mari kita buat program penghitungan

1. Buka text editor teman teman, dan buat file bernama perhitungan.java



2. Tulis kode dibawah ini kedalam file perhitungan.java



3. Jalankan kode tersebut

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$ cd Pertemuan\ 2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java perhitungan
2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java perhitungan
2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

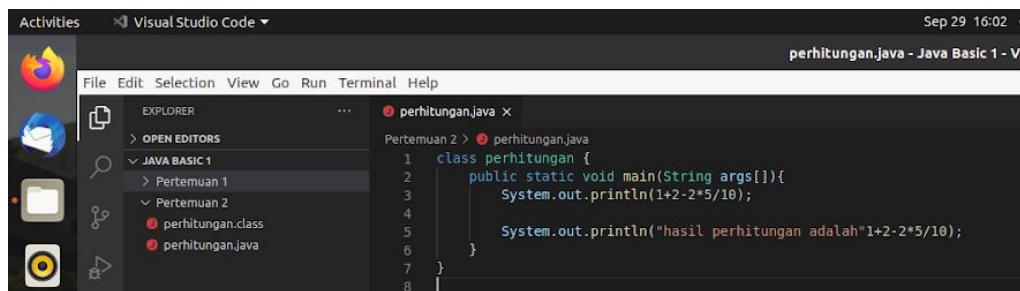
Penjelasan :

Seperti yang teman teman lihat pada kode diatas, kode yang kita buat digunakan untuk melakukan perhitungan. Kita dapat melakukan perhitungan langsung pada perintah :

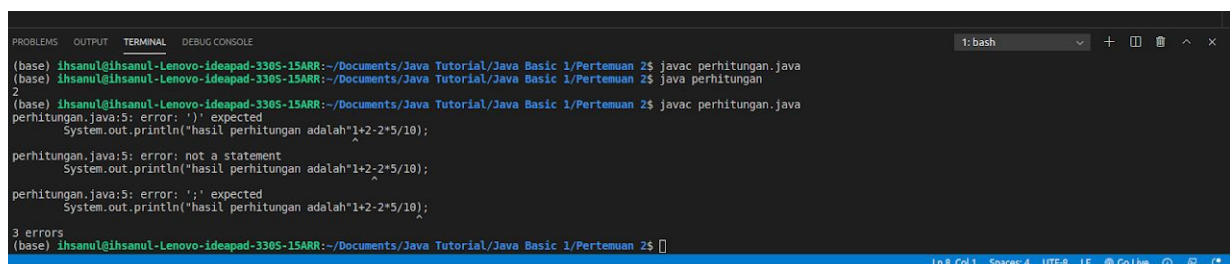
```
System.out.println(1+2-2*5/10);
```

Perintah diatas kita gunakan untuk menampilkan tulisan di terminal/monitor teman teman. Tetapi, perintah diatas juga dapat menampilkan perhitungan.

Nah sekarang, kita ingin menampilkan teks dan hasil perhitungan, tulis kode dibawah ini didalam file perhitungan.java yang telah teman teman buat dan jalan kan



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
  JAVA BASIC 1
    Pertemuan 1
    Pertemuan 2
      perhitungan.class
      perhitungan.java
  perhitungan.java x
  Pertemuan 2 > perhitungan.java
1 class perhitungan {
2   public static void main(String args[]){
3     System.out.println(1+2-2*5/10);
4
5     System.out.println("hasil perhitungan adalah"1+2-2*5/10);
6   }
7 }
8
```

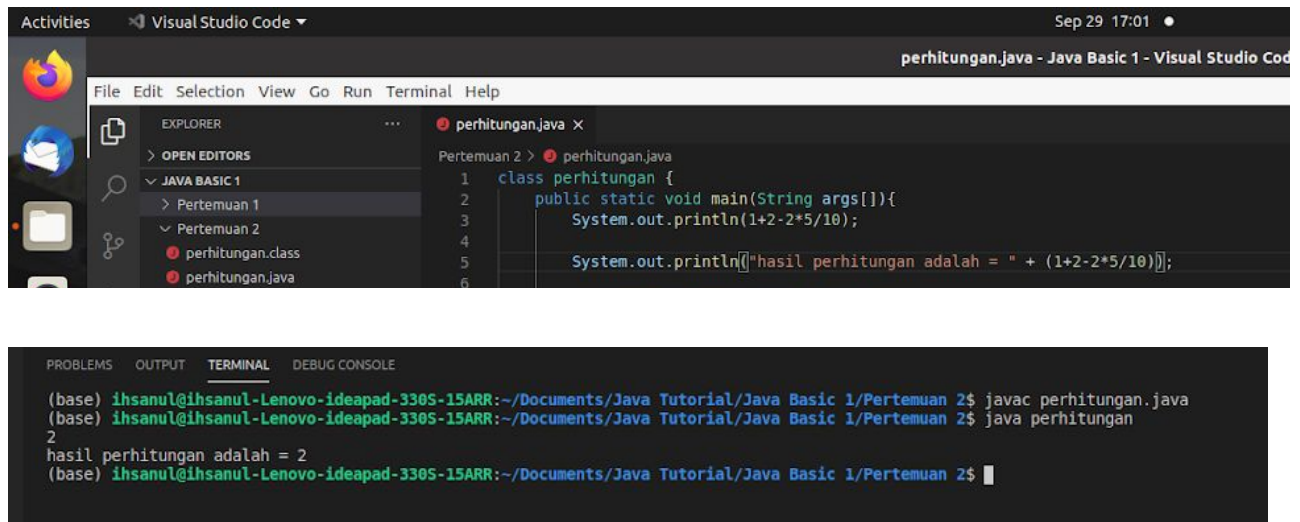


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java perhitungan
2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungan.java
perhitungan.java:5: error: ';' expected
    System.out.println("hasil perhitungan adalah"1+2-2*5/10);
                                           ^
perhitungan.java:5: error: not a statement
    System.out.println("hasil perhitungan adalah"1+2-2*5/10);
                                           ^
perhitungan.java:5: error: ';' expected
    System.out.println("hasil perhitungan adalah"1+2-2*5/10);
                                           ^
3 errors
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

Seperti yang teman teman lihat, program tidak bisa di compile. Kenapa? Karena kita tidak bisa menggabungkan 2 tipe data dalam 1 kalimat.

Tipe data sendiri adalah jenis data yang kita gunakan pada program yang kita buat. Ambil contoh ketika kita ingin menampilkan tulisan, maka tipe data yang kita gunakan adalah string, jika kita ingin menampilkan angka maka tipe data yang kita gunakan adalah integer atau double, dan masih terdapat tipe data lain nya.

Untuk mengatasi masalah tersebut, kita bisa menggunakan string concatenation (menyambungkan tipe data yang berbeda menjadi string)



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with 'JAVA BASIC 1' containing 'Pertemuan 1' and 'Pertemuan 2'. Under 'Pertemuan 2', there are files 'perhitungan.class' and 'perhitungan.java'. The Editor panel shows the code for 'perhitungan.java':

```
1 class perhitungan {
2     public static void main(String args[]){
3         System.out.println(1+2-2*5/10);
4
5         System.out.println("hasil perhitungan adalah = " + (1+2-2*5/10));
6     }
```

The TERMINAL panel at the bottom shows the execution of the program:

```
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungan.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java perhitungan
2
hasil perhitungan adalah = 2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

Loh, loh, loh, Kok bisa? Tanda + bukan nya untuk bikin pertambahan?

Jadi gini temen temen, tanda “+” disebelah tulisan “hasil perhitungan adalah” itu kita gunakan untuk menyambungkan 2 tipe data berbeda agar bisa ditampilkan bersama

Sedangkan tanda kurung digunakan untuk memblok angka angka tersebut agar melakukan perhitungan terlebih dahulu

Setelah perhitungan dilakukan, hasil dari angka tersebut diubah menjadi tipe data string dan disambungkan dengan tulisan yang telah kita buat, dan hasil nya pun ditampilkan di layar komputer temen temen

Sebentar, tadi bukan nya angka itu tipe data nya integer atau float? Kok bisa tiba tiba jadi string?

Untuk menjelaskan hal tersebut, kita akan menulis kode lagi masih di file perhitungan.java

Tulis kode dibawah ini pada file perhitungan.java dan jalankan

Penjelasan :

```
System.out.println("1+2+3+4+5+6") ;
```

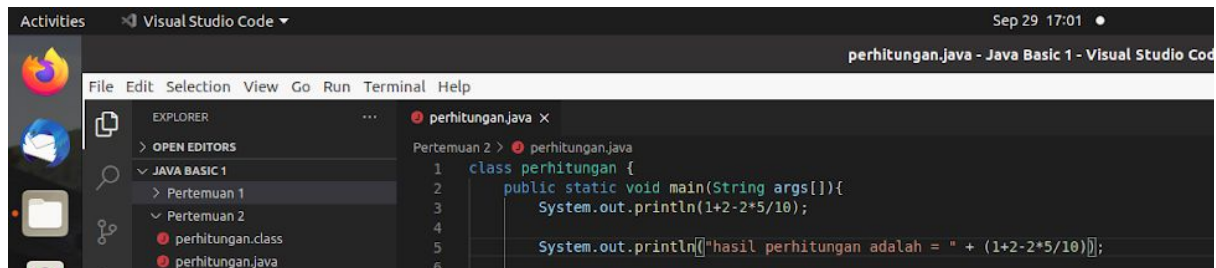
Pada perintah diatas, tanda “ “ menandakan bahwa yang ditulis didalam tanda tersebut adalah tipe data string, jadi walau tulisan nya berbentuk angka, tetapi kita memperlakukan angka tersebut sebagai teks bukan angka. Contoh lain nya, “Buku andi berjumlah 2”, pada kalimat tersebut angka 2 dianggap sebagai teks bukan angka

```
System.out.println(1+2+3+4+5+6)
```

Sedangkan pada perintah diatas, kita menganggap bahwa perhitungan tersebut adalah angka, makanya hasil dari penjumlahan tersebut adalah 21, tidak seperti pada perintah pertama

Notes: tanda // pada kode diatas, dianggap sebagai komentar, komentar tidak akan tampil pada terminal/layar komputer ketika program dijalankan. Fungsi komentar sendiri digunakan untuk menjelaskan maksud dari kode kalian

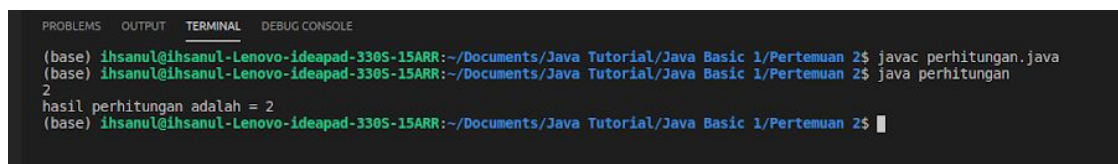
Kembali ke kode yang pertama kita buat



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  > OPEN EDITORS
    > JAVA BASIC 1
      > Pertemuan 1
      > Pertemuan 2
        perhitungannya.class
        perhitungannya.java
  perhitungannya.java x
Pertemuan 2 > perhitungannya.java
1 class perhitungannya {
2     public static void main(String args[]){
3         System.out.println(1+2-2*5/10);
4
5         System.out.println("hasil perhitungannya adalah = " + (1+2-2*5/10));
6     }
```

Perhatikan pada kode “hasil perhitungannya adalah” + (1+2+2*5/10)

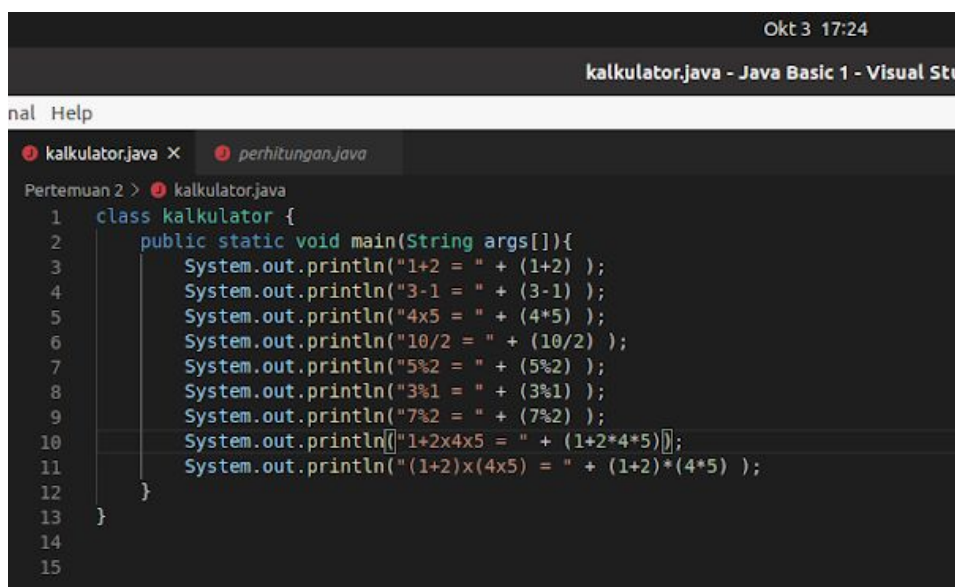
Sebelum hasil kode yang telah dieksekusi ditampilkan, perhitungan dilakukan terlebih dahulu di dalam tanda kurung tutup (). Lalu, hasil perhitungan tersebut digabungkan ke dalam string “hasil perhitungannya adalah”. Sebelum digabungkan, hasil perhitungan diubah secara otomatis ke dalam tipe data string dan setelah itu digabungkan. Lalu hasil penggabungan nya, ditampilkan ke terminal/layar komputer teman teman sebagai sebuah teks seperti dibawah ini



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac perhitungannya.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java perhitungannya
2
hasil perhitungannya adalah = 2
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

Mari kita buat beberapa perhitungan lagi, tetapi untuk kali ini kita akan menggunakan file baru bernama kalkulator.java

Tulis kode dibawah ini dan jalankan



```
Okt 3 17:24
kalkulator.java - Java Basic 1 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
kalkulator.java x perhitungannya.java
Pertemuan 2 > kalkulator.java
1 class kalkulator {
2     public static void main(String args[]){
3         System.out.println("1+2 = " + (1+2) );
4         System.out.println("3-1 = " + (3-1) );
5         System.out.println("4x5 = " + (4*5) );
6         System.out.println("10/2 = " + (10/2) );
7         System.out.println("5%2 = " + (5%2) );
8         System.out.println("3%1 = " + (3%1) );
9         System.out.println("7%2 = " + (7%2) );
10        System.out.println("1+2x4x5 = " + (1+2*4*5));
11        System.out.println("(1+2)x(4x5) = " + (1+2)*(4*5) );
12    }
13 }
14
15
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ javac kalkulator.java
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$ java kalkulator
1+2 = 3
3-1 = 2
4x5 = 20
10/2 = 5
5%2 = 1
3%1 = 0
7%2 = 1
1+2x4x5 = 41
(1+2)x(4x5) = 60
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

Pada kode diatas kita melakukan perhitungan dengan operator aritmatika. Operator aritmatika kita gunakan untuk melakukan operasi yang berhubungan dengan perhitungan aritmatika

Berikut adalah daftar operator aritmatika :

- + = Pertambahan
- = Pengurangan
- * = Perkalian
- / = Pembagian
- % = Sisa hasil bagi

Untuk pertambahan sampai pembagian tidak akan kita bahas karena itu merupakan perhitungan aritmatika yang sudah sering kita lakukan sedari sekolah. Yang akan kita bahas hanya sisa hasil bagi atau biasa disebut modulo.

Mari kita lakukan perhitungan modulo tanpa kode

$\begin{array}{r} 3 \\ 2 \overline{) 7} \\ \underline{6} \\ 1 \end{array}$ <p>→ Sisa Hasil Bagi / Modulo</p>	$\begin{array}{r} 2 \\ 3 \overline{) 8} \\ \underline{6} \\ 2 \end{array}$ <p>→ Sisa Hasil Bagi / Modulo</p>
$\begin{array}{r} 5 \\ 2 \overline{) 5} \\ \underline{10} \\ 1 \end{array}$ <p>→ Sisa Hasil Bagi / Modulo</p>	$\begin{array}{r} 4 \\ 2 \overline{) 8} \\ \underline{8} \\ 0 \end{array}$ <p>→ Sisa Hasil Bagi / Modulo</p>

Jika kamu melihat pada operasi pembagian di atas, kamu dapat melihat bahwa dari pembagian diatas mempunyai sisa hasil bagi. Kamu mungkin dapet melanjutkan pembagian tersebut hingga menghasilkan jawaban koma atau desimal. Tetapi, disini yang kita cari adalah sisa hasil bagi dimana hasil terakhir dari perhitungan diatas tidak bisa dibagi lagi.

Mari kita kembali ke kode kalkulator kembali dan fokus pada 2 baris kode terakhir dan hasil eksekusinya

```
System.out.println("1+2x4x5 = " + (1+2*4*5));  
System.out.println("(1+2)x(4x5) = " + (1+2)*(4*5) );
```

```
1+2x4x5 = 41  
(1+2)x(4x5) = 60  
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1/Pertemuan 2$
```

Perhitungannya sama, tapi hasilnya kok beda?

Jadi pada Java terdapat yang namanya operator precedence yang artinya ada prioritas pada operator perhitungan

Urutan operator precedence pada operator aritmatika adalah

1. () = Asosiasi operator dari kiri ke kanan
2. *, / , % = Asosiasi operator dari kiri ke kanan
3. + , - = Asosiasi operator dari kiri ke kanan

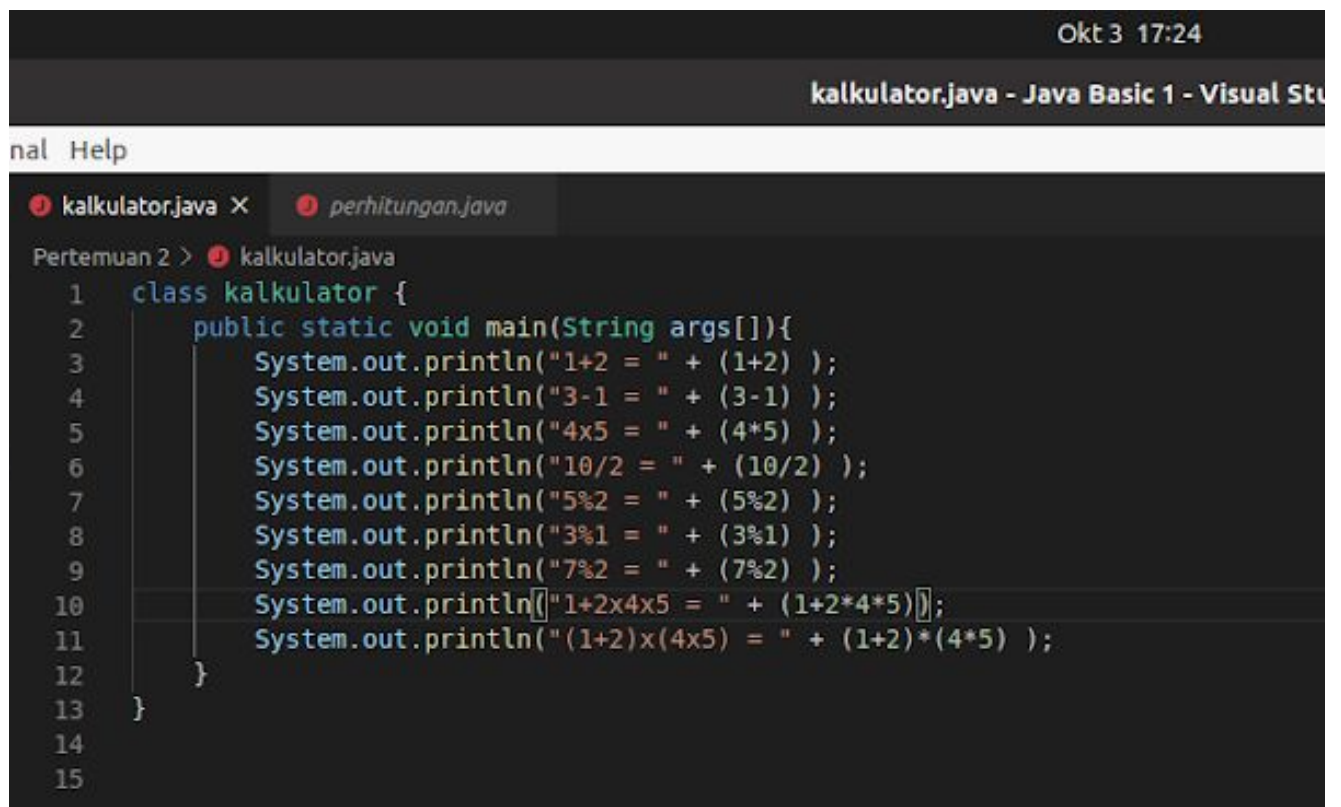
Jadi pada perhitungan $1+2 \times 4 \times 5 = 41$ maka perhitungan dilakukan seperti ini

1. Operator + berada pada urutan ke 3, dan operator *, / berada pada urutan ke 2, maka yang akan dieksekusi adalah * terlebih dahulu
2. Karena asosiasi * dari kiri ke kanan, maka yang akan dijalankan terlebih dahulu adalah * bagian kiri. Jadi 2×4 akan dieksekusi terlebih dahulu yang menghasilkan 8
3. Setelah dapat 8 maka operator yang selanjutnya dieksekusi adalah * yang selanjutnya, jadi 8×5 menghasilkan 40
4. Lalu operator yang dieksekusi terakhir adalah +. Hasil eksekusi nya adalah $40+1 = 41$

Sedangkan pada perhitungan $(1+2) \times (4 \times 5) = 60$ adalah sebagai berikut

1. Operator () berada pada urutan pertama, maka akan dieksekusi pertama kali
2. Terdapat 2 operator (), yaitu $(1+2)$ dan (4×5) . Yang akan dieksekusi terlebih dahulu adalah $(1+2)$ setelah itu (4×5) . Karena, operator () asosiasi nya berjalan dari kiri ke kanan. Jadi kiri dieksekusi terlebih dahulu, baru yang kanan
3. Hasil dari $(1+2)$ adalah 3, (4×5) adalah 20
4. Lalu operator yang dieksekusi terakhir adalah x. Hasil eksekusi nya adalah $3 \times 20 = 60$

Mari kembali ke kode kalkulator.java



```
Okt 3 17:24
kalkulator.java - Java Basic 1 - Visual Studio
nal Help
kalkulator.java x perhitungan.java
Pertemuan 2 > kalkulator.java
1 class kalkulator {
2     public static void main(String args[]){
3         System.out.println("1+2 = " + (1+2) );
4         System.out.println("3-1 = " + (3-1) );
5         System.out.println("4x5 = " + (4*5) );
6         System.out.println("10/2 = " + (10/2) );
7         System.out.println("5%2 = " + (5%2) );
8         System.out.println("3%1 = " + (3%1) );
9         System.out.println("7%2 = " + (7%2) );
10        System.out.println("1+2x4x5 = " + (1+2*4*5));
11        System.out.println("(1+2)x(4x5) = " + (1+2)*(4*5) );
12    }
13 }
14
15
```

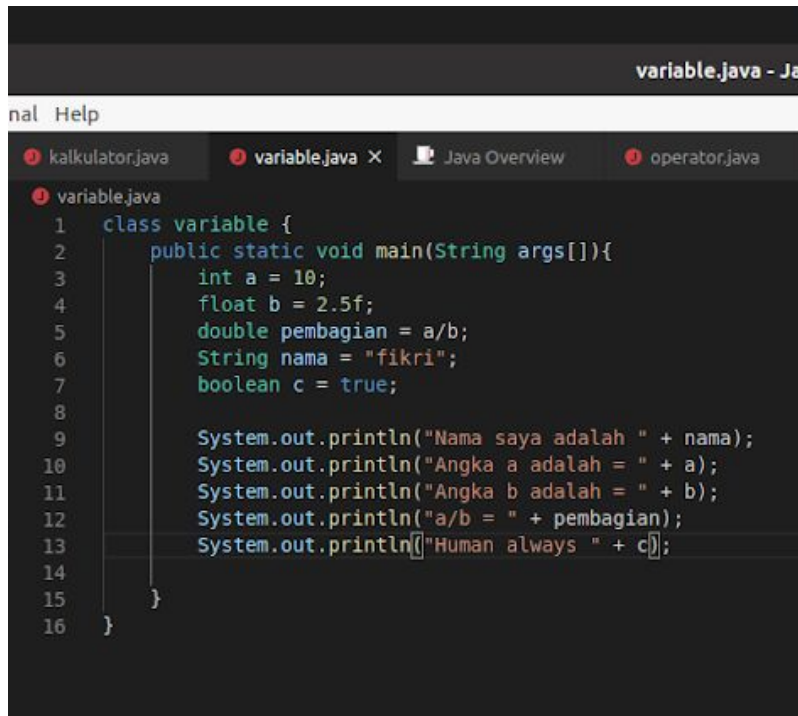
Pada kode diatas kita, kita membuat perhitungan 1+2, 3-1, 4+5, dan lain nya. Sekarang teman teman bayangkan jika seandainya teman teman ingin kembali melakukan perhitungan tersebut, jika menggunakan cara seperti diatas, maka teman teman harus menulis ulang kode tersebut. Contohnya, teman teman telah melakukan perhitungan 1+2 dan 3-1 seperti kode diatas. Pada baris kode berikutnya, teman teman ingin mengalikan hasil dari 1+2 dengan hasil 3-1. Bagaimana teman teman akan melakukan nya?

Mungkin sebagian teman teman akan menuliskan kodenya seperti ini

```
System.out.println("(1+2) x (3-1) = " + ((1+2)*(3-1)) );
```

Sekarang bayangkan, jika hasil dari angka yang teman teman dapatkan harus dikalikan kembali dengan nilai lain dan itu berulang sampai 10x. Maka, teman teman harus menulis kode yang sama berulang kali sampai 10x. Untuk mengatasi hal tersebut, mari kita berkenalan dengan variabel.

Untuk menjelaskan apa itu variable, mari kita tulis kode kembali didalam file baru bernama variable.java

A screenshot of an IDE window titled 'variable.java - Java'. The window shows a Java class named 'variable' with a 'main' method. The code declares variables of different types: 'int', 'float', 'double', 'String', and 'boolean'. It also includes several 'System.out.println' statements to print the values of these variables and a string concatenation.

```
1 class variable {  
2     public static void main(String args[]){  
3         int a = 10;  
4         float b = 2.5f;  
5         double pembagian = a/b;  
6         String nama = "fikri";  
7         boolean c = true;  
8  
9         System.out.println("Nama saya adalah " + nama);  
10        System.out.println("Angka a adalah = " + a);  
11        System.out.println("Angka b adalah = " + b);  
12        System.out.println("a/b = " + pembagian);  
13        System.out.println("Human always " + c);  
14    }  
15 }  
16 }
```

A screenshot of a terminal window showing the execution of the Java program. The user runs 'javac variable.java' and then 'java variable'. The output of the program is displayed, showing the printed values of the variables and the result of the division.

```
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$ javac variable.java  
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$ java variable  
Nama saya adalah fikri  
Angka a adalah = 10  
Angka b adalah = 2.5  
a/b = 4.0  
Human always true  
(base) ihsanul@ihsanul-Lenovo-ideapad-330S-15ARR:~/Documents/Java Tutorial/Java Basic 1$
```

Seperti yang teman teman lihat di atas, kita dapat menyimpan nilai pada sebuah wadah. Contohnya pada angka 10, kita menyimpan pada wadah bernama a, angka 2.5 kita simpan pada wadah b, dan hasil pembagian wadah a/b kita simpan pada wadah bernama pembagian. Wadah tersebut adalah sesuatu yang kita sebut sebagai variable.

Variable sendiri memiliki tipe data masing masing. Beberapa diantaranya adalah

int = tipe data berupa angka atau bilangan bulat. Contoh 24

float = tipe data berupa bilangan desimal. Contoh 2.5

Double = tipe data desimal seperti float, tetapi memiliki memori yang lebih besar

String = tipe berupa teks atau huruf, penulisan nilai pada tipe data string diawali dan diakhiri dengan “ “

boolean = tipe data berupa benar atau salah

Masih terdapat beberapa tipe data lain, tetapi sebagian besar yang akan kita pakai adalah ke 5 tipe data tersebut.

Untuk penulisan umum variable adalah sebagai berikut

(tipe data) (nama variable) = nilai variable

Semisal kita menginginkan untuk menyimpan nilai berupa angka, maka akan ditulis sebagai berikut

```
int angka = 10;
```

Penamaan pada variabel juga memiliki beberapa aturan, yaitu :

1. Nama variable boleh berupa huruf, angka, garis bawah, dan dollar.
2. Nama variable tidak boleh menggunakan kata kunci dijava. Seperti if, else, or, dan lain nya
3. Nama variable harus diawali huruf atau _ (underscore).
4. Jika nama variable memiliki 2 kata, disarankan menggunakan penulisan camelCase, yaitu kata pertama diawali huruf kecil, kata kedua diawali huruf besar. Contoh String namaSaya = "Fikri"
5. Disarankan nama variable memiliki arti sesuai nilai yang ingin dimasukan. Contoh misal ingin memiliki variable untuk menampung nama, maka bisa ditulis String nama = "fikri"

Itu saja untuk part 1, teman teman bisa lanjut ke modul part 2 pada file selanjutnya