

Tugas 1 smt1

Nama : Muhamad Wahyu Saputra

Kelas : XI RPL B

Absen : 35

1. Latihan string dan penggabungan

The screenshot shows an IDE with a Java file named `string.java` and its output. The code defines a `string` class with a `main` method that demonstrates various string operations, including printing, concatenation, and string immutability.

```
1 public class string {
2
3
4     public static void main(String[] args) {
5
6         System.out.println("=====STRING=====");
7
8         String iniString="selamat datang di latihan string oleh wahyu";
9         System.out.println("ini variabel string => " + iniString);
10
11         System.out.println("\n =====");
12
13         String stringKu;
14         stringKu="selamat datang di universitasku"+" \nsemoga kamu betah ya \n";
15         System.out.println(stringKu);
16
17         System.out.println("PENGABUNGAN STRING DIBAWAH\n");
18
19         stringKu += "jadi kalian yg maba harus membayar uang 100juta ke mas wahyu";
20         System.out.println(stringKu);
21
22     }
23 }
24
```

The output window shows the following text:

```
run:
=====STRING=====
ini variabel string =>selamat datang di latihan string oleh wahyu

=====
selamat datang di universitasku
semoga kamu betah ya

PENGABUNGAN STRING DIBAWAH

selamat datang di universitasku
semoga kamu betah ya
jadi kalian yg maba harus membayar uang 100juta ke mas wahyu
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Membandingkan string

```
1 public class membandingkanString {
2
3
4
5     public static void main(String[] args) {
6         String myString1 = "Saya";
7         String myString2 = "Wahyu";
8         String myString3 = "SayaWahyu";
9
10        System.out.println("STRING 1 " + myString1 );
11        System.out.println("STRING 2 " + myString2 );
12        System.out.println("STRING 3 " + myString3 );
13
14        System.out.println("STRING 1 + STRING 3 ==> " +myString1.equals(myString3));
15        System.out.println("STRING 2 + STRING 3 ==> " +myString1.equals(myString3));
16        System.out.println("STRING 1dan2 + STRING 3 ==> " +(myString1+myString2).equals(myString3));
17
18        System.out.println("equalsIgnoreCase=====");
19        String myString4 = "WAHYU ADALAH MANUSIA BUKAN ALIEN";
20        String myString5 = "Wahyu adalah manusia bukan alien";
21
22        System.out.println("STRING 4 => " + myString4 );
23        System.out.println("STRING 5 => " + myString5 );
24        System.out.println("perbandingan ini mengabaikan huruf upper dan lower=====");
25        System.out.println("STRING 4 + 5 => "+myString4.equalsIgnoreCase(myString5));
26
27        System.out.println("STRING 4 & 5 + 5 + 4 => "+(myString4+myString5).equalsIgnoreCase(myString5+myString4));
28
29
30        System.out.println("PERBANDINGAN '==' ");
31        String myString6 = "RPL";
32        String myString7 = new String (myString6);
33        String myString8 = "rpl";
34        String myString9 = "RPL";
35
36        System.out.println("STRING 6 + 8 => "+(myString6==myString8));
37        System.out.println("STRING 7 + 8 => "+(myString7==myString8));
38        System.out.println("STRING 6 + 9 => "+(myString6==myString9));
39
40    }
41
42 }
43
```

Output - STRING (run) X

```
run:
STRING 1 Saya
STRING 2 Wahyu
STRING 3 SayaWahyu
STRING 1 + STRING 3 ==> false
STRING 2 + STRING 3 ==> false
STRING 1dan2 + STRING 3 ==> true
equalsIgnoreCase=====
STRING 4 => WAHYU ADALAH MANUSIA BUKAN ALIEN
STRING 5 => Wahyu adalah manusia bukan alien
perbandingan ini mengabaikan huruf upper dan lower=====
STRING 4 + 5 => true
STRING 4 & 5 + 5 + 4 => true
PERBANDINGAN '=='
STRING 6 + 8 => false
STRING 7 + 8 => false
STRING 6 + 9 => true
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Awal akhir string

```
Source History
1
2 public class AwalAkhirString {
3
4
5 public static void main(String[] args) {
6     String myString = "Wahyu adalah manusia biasa";
7
8     System.out.println("STARTSWITH=====");
9
10    System.out.println(myString.startsWith("Wahyu"));
11    System.out.println(myString.startsWith("wahyu"));
12    System.out.println(myString.startsWith("Wah",1));
13    System.out.println(myString.startsWith("Wa",2));
14    System.out.println(myString.startsWith("W",3));
15
16    System.out.println("ENDSWITH=====");
17    System.out.println(myString.endsWith("bia"));
18    System.out.println(myString.endsWith("manusia"));
19    System.out.println(myString.endsWith("bias"));
20    System.out.println(myString.endsWith("biasa"));
21
22 }
23
24 }
25
```

```
Output - STRING (run) X
run:
STARTSWITH=====
true
false
false
false
false
ENDSWITH=====
false
false
false
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Urutan string

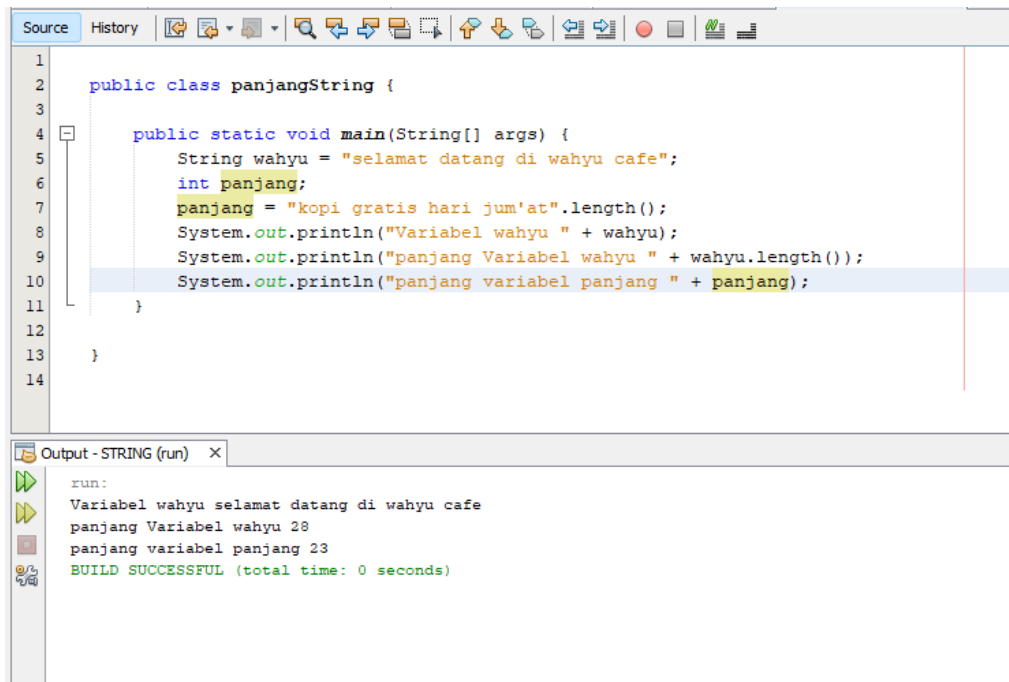
```
1
2 public class UrutanString {
3
4
5     public static void main(String[] args) {
6         String [] nama={"Ayam","Kucing","Sapi","Angsa","zebra"};
7         String temp;
8         System.out.println("Nama nama hewan sebelum diurutkan");
9         for (int i=0; i<nama.length; i++){
10             System.out.println(i+1+" "+nama[i]+" ");
11         }
12         //mengurutkan nama hewan
13
14         System.out.println("=====");
15         System.out.println("Nama nama hewan setelah diurutkan");
16         for (int i=0; i<(nama.length-1); i++){
17             for (int w=0; w<(nama.length-1); w++){
18                 if (nama[w].compareTo(nama[w+1])>0){
19                     temp=nama[w+1];
20                     nama[w+1]=nama[w];
21                     nama[w]=temp;
22                 }
23             }
24         }
25
26         for (int i=0; i<nama.length; i++){
27             System.out.println(i+1+" "+nama[i]);
28         }
29
30     }
31 }
```

Output - STRING (run) X

```
run:
Nama nama hewan sebelum diurutkan
1 Ayam
2 Kucing
3 Sapi
4 Angsa
5 zebra

=====
Nama nama hewan setelah diurutkan
1 Angsa
2 Ayam
3 Kucing
4 Sapi
5 zebra
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Panjangstring



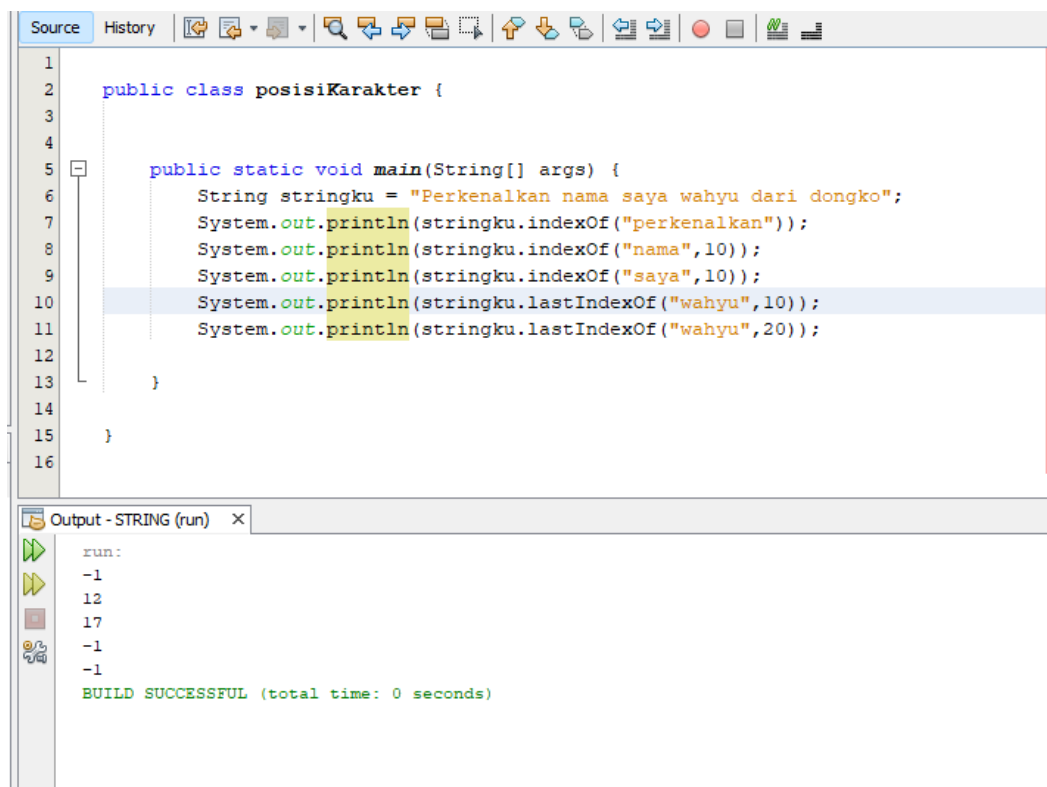
The screenshot shows an IDE with a Java source file. The code defines a class `panjangString` with a `main` method. It initializes a `String` variable `wahyu` with the value "selamat datang di wahyu cafe" and an `int` variable `panjang`. The `panjang` variable is assigned the length of the string "kopi gratis hari jum'at". The program then prints the length of `wahyu` and the value of `panjang`. The output window shows the results of the execution.

```
1
2 public class panjangString {
3
4     public static void main(String[] args) {
5         String wahyu = "selamat datang di wahyu cafe";
6         int panjang;
7         panjang = "kopi gratis hari jum'at".length();
8         System.out.println("Variabel wahyu " + wahyu);
9         System.out.println("panjang Variabel wahyu " + wahyu.length());
10        System.out.println("panjang variabel panjang " + panjang);
11    }
12
13 }
14
```

Output - STRING (run) X

```
run:
Variabel wahyu selamat datang di wahyu cafe
panjang Variabel wahyu 28
panjang variabel panjang 23
BUILD SUCCESSFUL (total time: 0 seconds)
```

6. Posisi Karakter



The screenshot shows an IDE with a Java source file. The code defines a class `posisiKarakter` with a `main` method. It initializes a `String` variable `stringku` with the value "Perkenalkan nama saya wahyu dari dongko". The program then prints the index of "perkenalkan", "nama", "saya", and "wahyu" in the string, and the last index of "wahyu". The output window shows the results of the execution.

```
1
2 public class posisiKarakter {
3
4     public static void main(String[] args) {
5         String stringku = "Perkenalkan nama saya wahyu dari dongko";
6         System.out.println(stringku.indexOf("perkenalkan"));
7         System.out.println(stringku.indexOf("nama", 10));
8         System.out.println(stringku.indexOf("saya", 10));
9         System.out.println(stringku.lastIndexOf("wahyu", 10));
10        System.out.println(stringku.lastIndexOf("wahyu", 20));
11    }
12
13 }
14
15 }
16
```

Output - STRING (run) X

```
run:
-1
12
17
-1
-1
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.arraykarakter

```
1
2 public class arrayKarakter {
3
4
5     public static void main(String[] args) {
6         String str="Ilmu komputer UNP";
7         char[] arraystr=str.toCharArray();
8         System.out.println("String : " + str);
9         System.out.println("String baru [toCharArray]: ");
10
11         for (int i=0; i<arraystr.length ; i++){
12             System.out.println(arraystr[i]);
13         }
14         System.out.println("String baru [getChars]: ");
15         char[] getstr= new char[10];
16         str.getChars(5,13,getstr,0);
17         for (int i=0; i<getstr.length ; i++){
18             System.out.println(getstr[i]);
19         }
20     }
21
22 }
23
```

Output - STRING (run) ×

```
>> String : Ilmu komputer UNP
>> String baru [toCharArray]:
I
l
m
u

k
o
m
p
u
t
e
r

U
N
P
String baru [getChars]:
k
o
m
p
u
t
e
r

BUILD SUCCESSFUL (total time: 0 seconds)
```

8.STRING ARRAY

The screenshot shows the NetBeans IDE 8.2 interface. The 'Source' window displays the code for the `stringArray` class. The code defines a `main` method that takes a `String[] args` parameter. It initializes a `char` array `arraystr` with the characters 'I', 'L', 'M', 'U', ' ', 'W', 'E', 'B'. It then prints the array, iterates through it to print each character on a new line, and finally prints the entire array as a single string using `String.valueOf(arraystr)`.

```
1 public class stringArray {
2
3
4     public static void main(String[] args) {
5         char[] arraystr={'I','L','M','U',' ','W','E','B'};
6
7         System.out.println("Array char");
8
9         for (int i=0; i<arraystr.length; i++){
10             System.out.println(arraystr[i]);
11         }
12         System.out.println("String Baru : ");
13
14         String str=String.valueOf(arraystr);
15         System.out.println(str);
16     }
17 }
```

The 'Output - STRING (run)' window shows the execution results:

```
run:
Array char
I
L
M
U

W
E
B
String Baru :
ILMU WEB
BUILD SUCCESSFUL (total time: 0 seconds)
```

9.STRING BUFFER

The screenshot shows the NetBeans IDE 8.2 interface. The 'Source' window displays the code for the `stringBufferku` class. The code defines a `main` method that takes a `String[] args` parameter. It initializes a `StringBuffer` object `sb`. It then appends the string 'IMMU WEB DESAIN' to `sb`, followed by a space and the string 'Juga boleh'. Finally, it prints the contents of `sb` using `sb.toString()`.

```
1 public class stringBufferku {
2
3
4     public static void main(String[] args) {
5         StringBuffer sb = new StringBuffer();
6
7         String kata= " WEB DESAIN";
8         sb.append(" IMMU ").append(kata).append(" Juga boleh");
9         System.out.println(sb.toString());
10        System.out.println(sb);
11    }
12
13
14 }
```

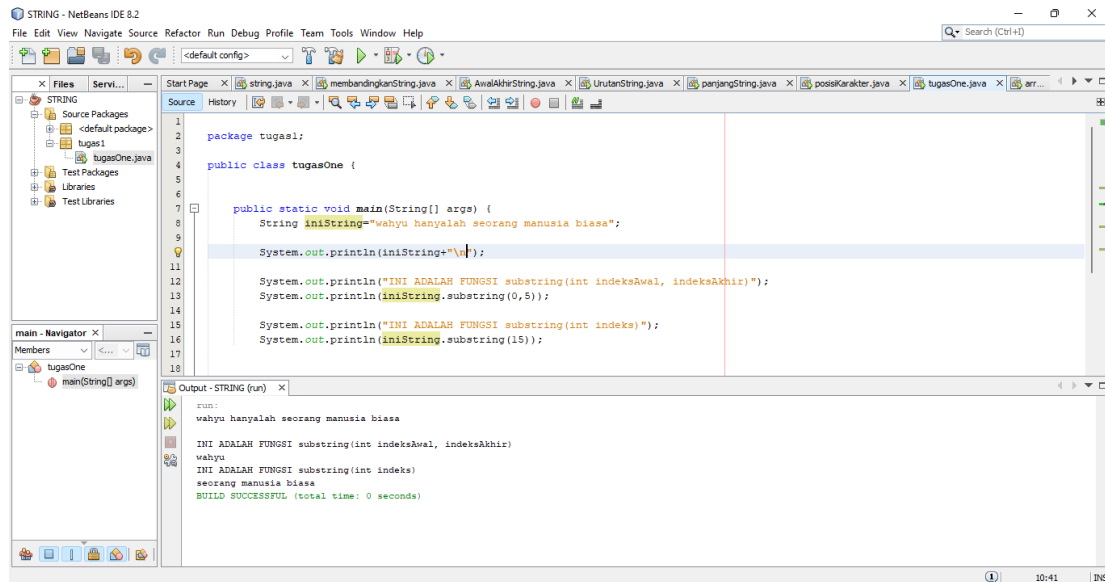
The 'Output - STRING (run)' window shows the execution results:

```
run:
IMMU WEB DESAIN Juga boleh
IMMU WEB DESAIN Juga boleh
BUILD SUCCESSFUL (total time: 0 seconds)
```

Finished building STRING (run).

TUGAS

TUGAS 1 => METHOD substring



TUGAS 2 => METHOD trim&replace

