

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PENERAPAN *STRING* DAN *METHOD* DALAM
PEMROGRAMAN JAVA**



Disusun Oleh:

Haliya Isma Husna Putri Ahmadi
2511532002

Dosen Pengampu:
Wahyudi. Dr. S.T.M.T

Asisten Praktikum:
Muhammad Zaky Al Hafiz

**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas Rahmat dan karunia-Nya, penulis dapat menyelesaikan lapora praktikum Algoritma dan Pemrograman pekan 7 dengan judul “Penerapan *String* dan *Method* dalam Pemrograman Java” tepat pada waktunya.

Penyusunan laporan ini bertujuan untuk memenuhi salah satu tugas dalam mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran bagi penulis dalam memahami penerapan konsep *string* dan *method* pada bahasa pemrograman *Java*. Melalui praktikum ini, penulis dapat mempelajari bagaimana cara memanipulasi data teks menggunakan berbagai *method* bawaan *Java*, serta memahami fungsi *method* dalam menyusun program yang terstruktur dan efisien.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi perbaikan di masa yang akan datang.

Ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada dosen pengampu, asisten praktikum, serta semua pihak yang terlibat dalam pelaksanaan praktikum dan penyusunan laporan ini. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 10 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat	4
BAB II PEMBAHASAN	5
2.1 Dasar Teori.....	5
2.1.1 <i>String</i>	5
2.1.2 <i>Methode</i>	5
2.2 Langkah Kerja	7
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i>	7
2.2.2 Program Pertama: <i>String1_2511532002</i>	9
2.2.3 Program Kedua: <i>String2_2511532002</i>	10
2.2.4 Program Ketiga: <i>BilanganPrima_2511532002</i>	11
2.2.5 Program Keempat: <i>Mahasiswa_2511532002</i>	13
2.2.6 Program Kelima: <i>PanggilMahasiswa_2511532002</i>	14
2.2.7 Program Keenam: <i>PanggilMahasiswa2_2511532002</i>	15
2.2.8 <i>Upload</i> ke <i>GitHub</i>	17
BAB III KESIMPULAN.....	20
3.1 Kesimpulan	20
3.2 Saran	20
DAFTAR PUSTAKA.....	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, kemampuan untuk mengolah dan memanipulasi data merupakan hal yang sangat penting. Salah satu jenis data yang paling sering digunakan adalah *string* dan *methode*, dimana *string* memungkinkan programmer untuk mengelola informasi berbasis karakter yang membentuk teks. Sedangkan, konsep *method* berperan besar dalam membuat program yang terstruktur dan efisien. Dengan *method*, kode dapat dibagi menjadi bagian-bagian kecil yang memiliki fungsi spesifik, sehingga memudahkan proses pengembangan dan penggunaan ulang kode (*reusability*).

Melalui praktikum ini, mahasiswa diharapkan mampu memahami penerapan *string* dan *methode* dalam pemrograman *Java*, sehingga dapat menulis program yang tidak hanya benar secara sintaks, tetapi juga baik secara logika dan struktur.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

- 1.2.1 Memahami konsep dasar, logika dan struktur *syntax string* dalam bahasa pemrograman *Java*.
- 1.2.2 Mampu mengimplementasikan *String* dan *method* memanipulasi data teks dalam pembuatan program di *Java*.
- 1.2.3 Membangun logika pemrograman dalam menyusun program dan algoritma dengan pengolahan data teks.

1.3 Manfaat

Manfaat yang diharapkan dari pelaksanaan praktikum ini antara lain:

- 1.3.1 Memberikan pemahaman mengenai konsep dasar *string* dan *method*, khususnya dalam pengolahan data teks di *Java*.
- 1.3.2 Memberikan pengalaman langsung dalam mengimplementasikan konsep pengolahan data dan struktur program yang kompleks.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Dalam bahasa pemrograman *Java*, terdapat dua konsep penting yang digunakan untuk membangun program yang efisien dan terstruktur dalam pengolahan atau memanipulasi data, yaitu *String* dan *Method*. Kedua konsep ini saling berkaitan dalam pengolahan data dan penyusunan logika program.

String merupakan salah satu tipe data yang paling sering digunakan dalam *Java* untuk menyimpan dan mengelola teks. *String* bersifat *immutable*, artinya nilai *string* tidak dapat diubah setelah dibuat. Untuk memudahkan pengolahan teks, *Java* menyediakan berbagai *method* pada *class String*, sehingga proses manipulasi data menjadi lebih efisien dan terstruktur.

2.1.1 *String*

String adalah tipe data yang digunakan untuk menyimpan kumpulan karakter atau teks. Variabel *String* berisi kumpulan karakter yang berada dalam kutip dua (" "). Salah satu sifat utama *String* adalah *immutable*, artinya nilai dari objek *string* tidak dapat diubah setelah dibuat. Jika dilakukan operasi seperti penggabungan atau pemotongan, maka *Java* akan membuat objek *String* baru. Konsep ini penting agar data tetap konsisten dan aman saat digunakan dalam program.

2.1.2 *Method*

Method merupakan sekumpulan perintah atau blok kode yang digunakan untuk menjalankan tugas tertentu di dalam program. *Method* membantu memecah program menjadi bagian-bagian kecil yang lebih mudah dibaca, dikelola, dan digunakan ulang (*reusable*). Dengan menggunakan *method*, programmer dapat menulis kode yang lebih efisien dan terstruktur.

Berikut adalah beberapa *method* bawaan dari *class String* untuk memanipulasi data :

1. *Length()*

Digunakan untuk mengetahui jumlah karakter dalam string, termasuk spasi.

2. *toUpperCase()* dan *toLowerCase()*

Digunakan untuk mengubah bentuk huruf menjadi kapital (*UpperCase*) atau huruf kecil (*LowerCase*) Seluruhnya.

3. *indexOf(String str):*

Mengembalikan *indeks* (posisi) pertama dari kemunculan *str* di dalam *string*.

4. *equals(String anotherString):*

Membandingkan dua *string* secara *case-sensitive* (memperhatikan besar-kecil huruf).

5. *equalsIgnoreCase(String anotherString):*

Membandingkan dua *string* tanpa memperhatikan besar-kecil huruf.

6. *contains(CharSequence s):*

Memeriksa apakah *string* berisi urutan karakter yang ditentukan.

Contoh : `"hello world".contains("world")` akan menghasilkan *true*.

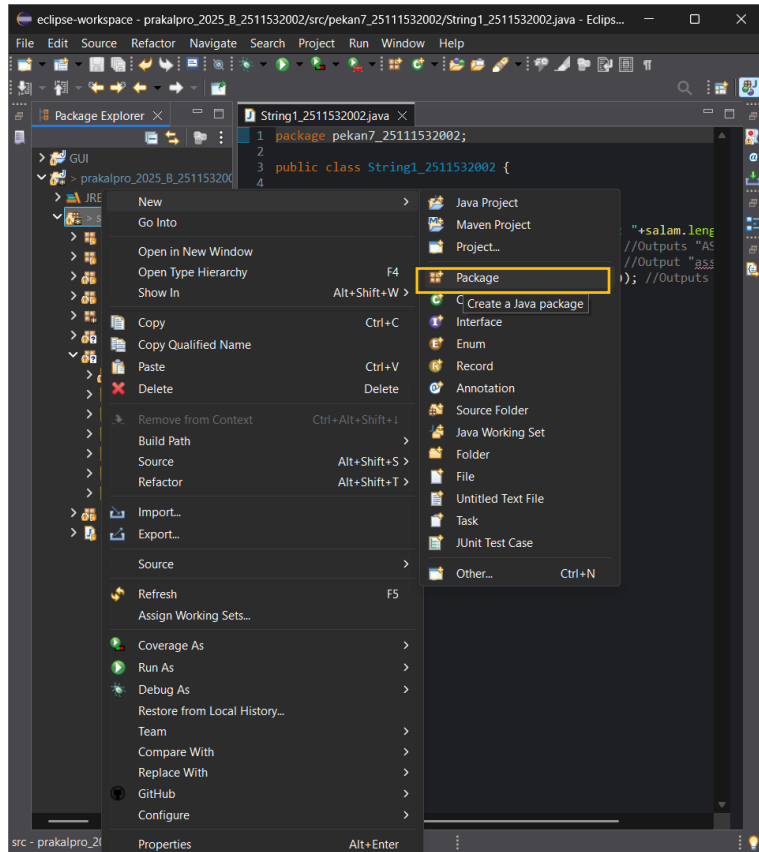
7. *replace(char oldChar, char newChar):*

Mengganti semua kemunculan *oldChar* dengan *newChar*.

2.2 Langkah Kerja

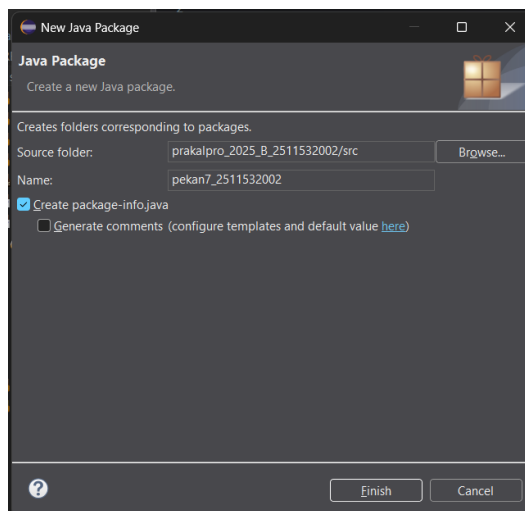
2.2.1 Pembuatan *Package* dan *Class*

1. Sebelum membuat suatu program *user* perlu untuk membuat *package* terlebih dahulu. Caranya dengan klik kanan pada *src* lalu pilih “New” dan klik “Package”.



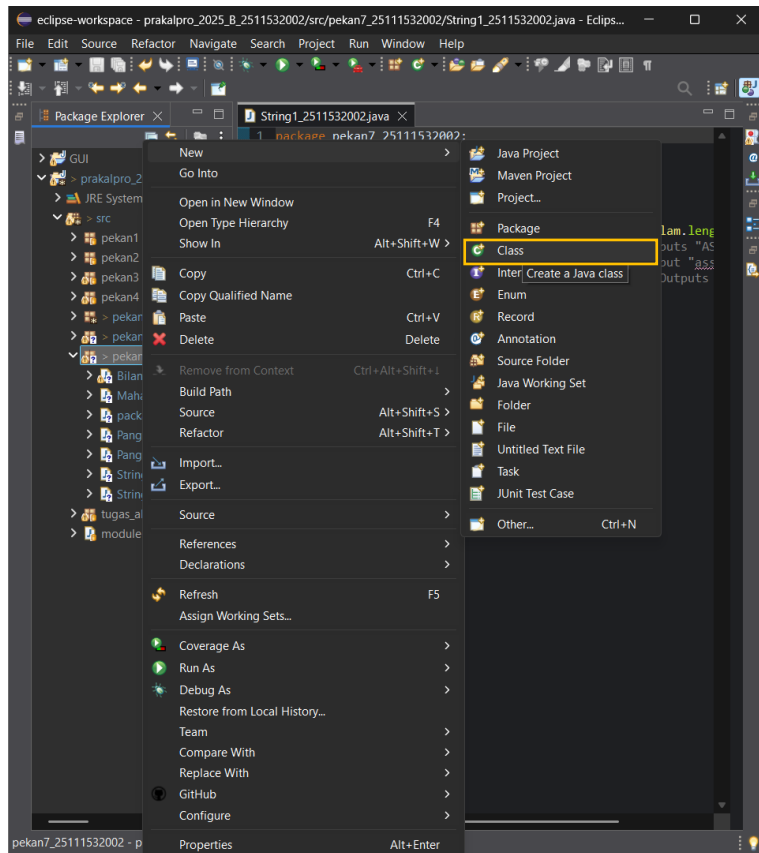
2.2.1 Pembuatan *Package*

2. Setelah itu akan muncul menu “Java Package”, buat nama *package* dengan ketentuan tanpa *capslock*, *space*, dan karakter khusus lainnya. Lalu klik “finish”.



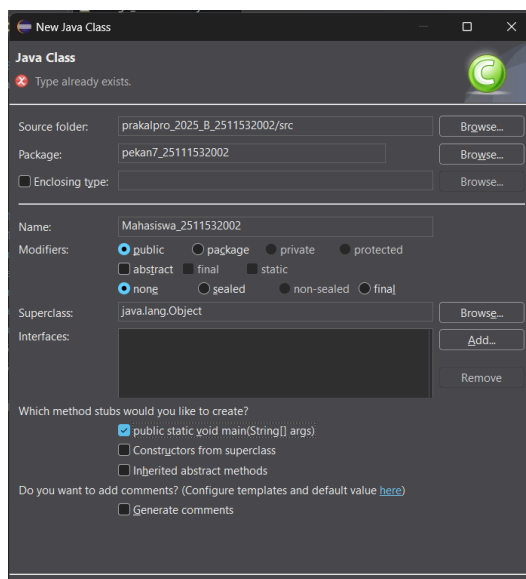
2.2.2 Pembuatan *Package*

3. Selajutnya adalah membuat Membuat *class*, pada *package* yang telah dibuat tapi klik kanan lalu pilih “New” dan pilih “Class”.



2.2 3 Pembuatan Class

4. Setelah itu pada menu “Java Class” buat nama *class*, dengan ketentuan nama harus *capslock* diawal kalimat dan tanpa spasi.



2.2 4 Pembuatan Class

Jika pada program yang akan dibuat menggunakan *class main*, maka *user* dapat mengaktifkan “*Public static void main (String[] grs)*” agar tidak perlu membuat *methode main()* secara manual nantinya. Selanjutnya “*finish*”.

2.2.2 Program Pertama: String1_2511532002

Pada *class* atau program ini, digunakan beberapa *method* bawaan dari *class String* untuk memanipulasi teks, seperti *length()*, *toUpperCase()* dan *toLowerCase()*, dan *indexOf()*.

1. Pada *class* yang sudah dibuat, masukan *syntax* seperti berikut.



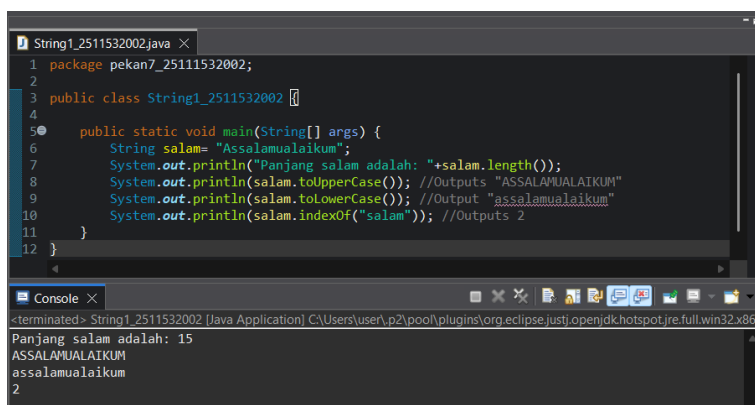
```

1 package pekan7_2511532002;
2
3 public class String1_2511532002 {
4
5     public static void main(String[] args) {
6         String salam= "Assalamualaikum";
7         System.out.println("Panjang salam adalah: "+salam.length());
8         System.out.println(salam.toUpperCase()); //Outputs "ASSALAMUALAIKUM"
9         System.out.println(salam.toLowerCase()); //Output "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); //Outputs 2
11    }
12 }
13

```

2.2 5 Syntax Program String1_2511532002

2. Setelah itu jalan kan program dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka pada menu *console* akan muncul hasil program seperti berikut.



```

String1_2511532002.java
1 package pekan7_2511532002;
2
3 public class String1_2511532002 {
4
5     public static void main(String[] args) {
6         String salam= "Assalamualaikum";
7         System.out.println("Panjang salam adalah: "+salam.length());
8         System.out.println(salam.toUpperCase()); //Outputs "ASSALAMUALAIKUM"
9         System.out.println(salam.toLowerCase()); //Output "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); //Outputs 2
11    }
12 }
13

Console
String1_2511532002 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justi.openjdkhotspot.jre.full.win32.x86_64.jdk\bin\java.exe
Panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2

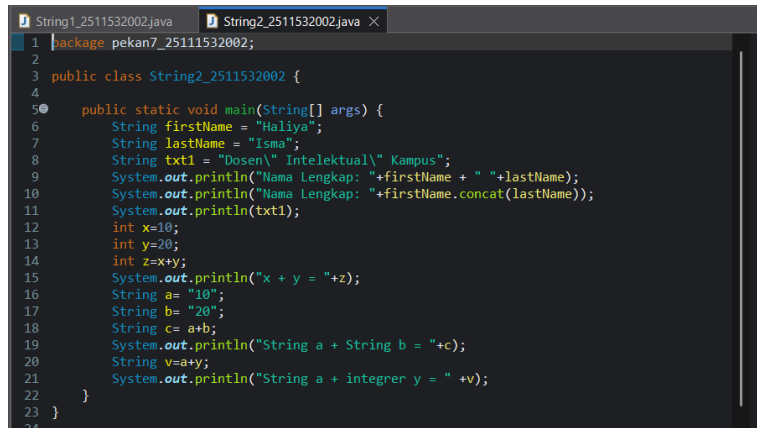
```

2.2 6 Hasil Prohgram String1_2511532002

- a) *length()* untuk mengetahui jumlah char dalam *string salam* (*Assalamualaikum*) =15
- b) *toUpperCase()* dan *toLowerCase()* digunakan untuk mengubah huruf *string salam* (*Assalamualaikum*).
- c) *indexOf("salam")* mencari posisi awal substring "salam" di dalam string utama, dan hasilnya berupa nilai indeks (dimulai dari 0).

2.2.3 Program Kedua: String2_2511532002

1. Buat kembali *class* baru pada *package* `pekan7_2511532002` dengan nama kelas "`String2_2511532002`". Selanjutnya masukan *syntax* seperti berikut.



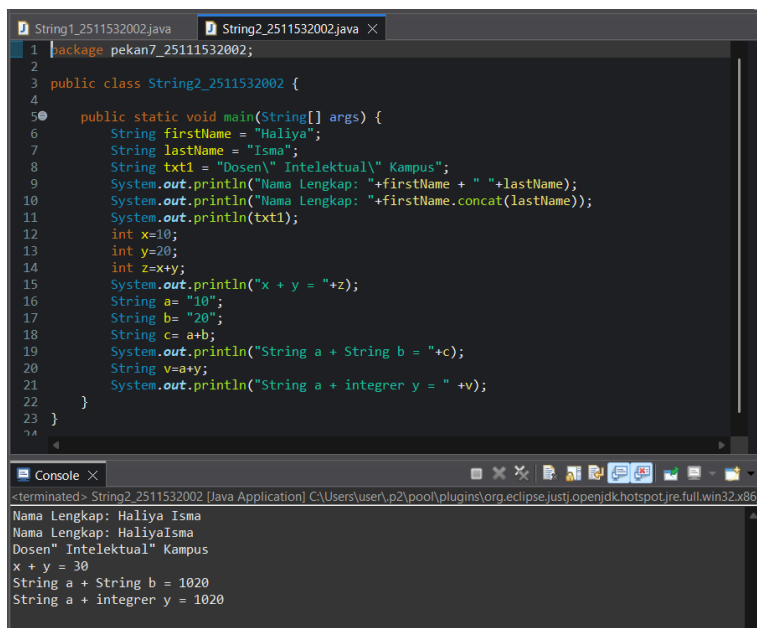
```

1 package pekan7_2511532002;
2
3 public class String2_2511532002 {
4
5     public static void main(String[] args) {
6         String firstName = "Haliya";
7         String lastName = "Isma";
8         String txt1 = "Dosen\ Intelktual\ Kampus";
9         System.out.println("Nama Lengkap: "+firstName + " "+lastName);
10        System.out.println("Nama Lengkap: "+firstName.concat(lastName));
11        System.out.println(txt1);
12        int x=10;
13        int y=20;
14        int z=x+y;
15        System.out.println("x + y = "+z);
16        String a= "10";
17        String b= "20";
18        String c= a+b;
19        System.out.println("String a + String b = "+c);
20        String v=a+y;
21        System.out.println("String a + integrer y = " +v);
22    }
23 }

```

2.2 7 Syntax Program String2_2511532002

3. Lalu jalankan program dengan klik *Run* di pada *toolbar*. Maka pada menu *console* akan muncul hasil program, dimana akan menlihatkan perbedaan operasi aritmatika pada *string*.



```

1 package pekan7_2511532002;
2
3 public class String2_2511532002 {
4
5     public static void main(String[] args) {
6         String firstName = "Haliya";
7         String lastName = "Isma";
8         String txt1 = "Dosen\ Intelktual\ Kampus";
9         System.out.println("Nama Lengkap: "+firstName + " "+lastName);
10        System.out.println("Nama Lengkap: "+firstName.concat(lastName));
11        System.out.println(txt1);
12        int x=10;
13        int y=20;
14        int z=x+y;
15        System.out.println("x + y = "+z);
16        String a= "10";
17        String b= "20";
18        String c= a+b;
19        System.out.println("String a + String b = "+c);
20        String v=a+y;
21        System.out.println("String a + integrer y = " +v);
22    }
23 }

```

```

terminated> String2_2511532002 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Nama Lengkap: Haliya Isma
Nama Lengkap: HaliyaIsma
Dosen\ Intelktual\ Kampus
x + y = 30
String a + String b = 1020
String a + integrer y = 1020

```

2.2 8 Hasil Program String2_2511532002

- a) Operator `+` pada *string* berfungsi sebagai penggabung teks, bukan penjumlahan. Method `.concat()` memiliki fungsi yang sama, namun lebih efisien jika banyak *string*.

b) `String txt1 = "Dosen\" Intelektual\" Kampus";`

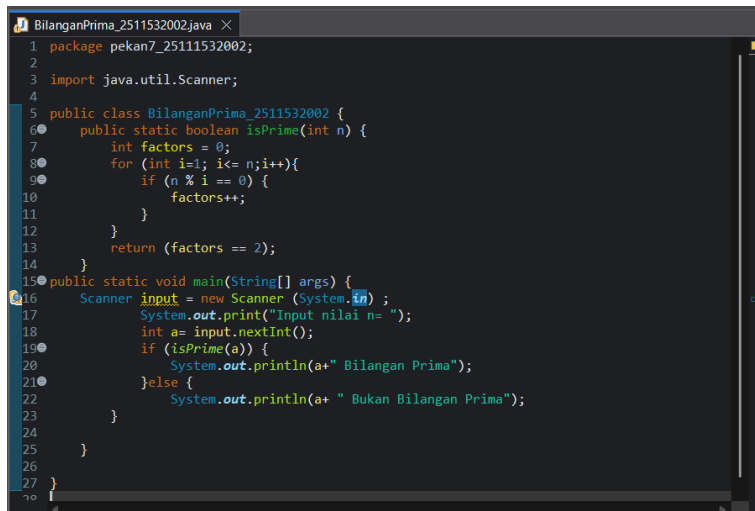
Tanda `\` digunakan agar karakter kutip ganda dapat ditampilkan.

c) `String c = a + b;`

d) a dan b bertipe *String*, maka `+` berfungsi sebagai penggabung teks, menghasilkan `"1020"`.

2.2.4 Program Ketiga: BilanganPrima_2511532002

1. Buat kembali *class* baru dengan nama kelas `"BilanganPrima_2511532002"`. Selanjutnya masukan *syntax* seperti berikut.



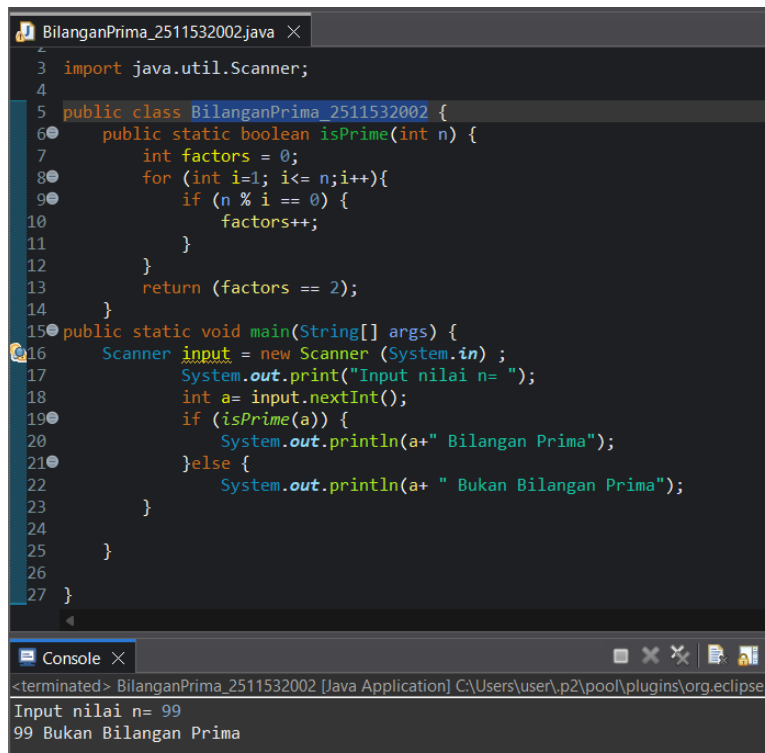
```

1 package pekan7_2511532002;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511532002 {
6     public static boolean isPrime(int n) {
7         int factors = 0;
8         for (int i=1; i<= n;i++){
9             if (n % i == 0) {
10                 factors++;
11             }
12         }
13         return (factors == 2);
14     }
15     public static void main(String[] args) {
16         Scanner input = new Scanner (System.in);
17         System.out.print("Input nilai n= ");
18         int a= input.nextInt();
19         if (isPrime(a)) {
20             System.out.println(a+ " Bilangan Prima");
21         }else {
22             System.out.println(a+ " Bukan Bilangan Prima");
23         }
24     }
25 }
26
27
28

```

2.2 9 Syntax Program BilanganPrima_2511532002

2. Lalu jalankan program, dengan klik tombol *Run* di pojok kiri atas pada *toolbar*.



```

3  import java.util.Scanner;
4
5  public class BilanganPrima_2511532002 {
6      public static boolean isPrime(int n) {
7          int factors = 0;
8          for (int i=1; i<= n;i++){
9              if (n % i == 0) {
10                 factors++;
11             }
12         }
13         return (factors == 2);
14     }
15     public static void main(String[] args) {
16         Scanner input = new Scanner (System.in) ;
17         System.out.print("Input nilai n= ");
18         int a= input.nextInt();
19         if (isPrime(a)) {
20             System.out.println(a+" Bilangan Prima");
21         }else {
22             System.out.println(a+ " Bukan Bilangan Prima");
23         }
24     }
25 }
26
27 }

```

Console

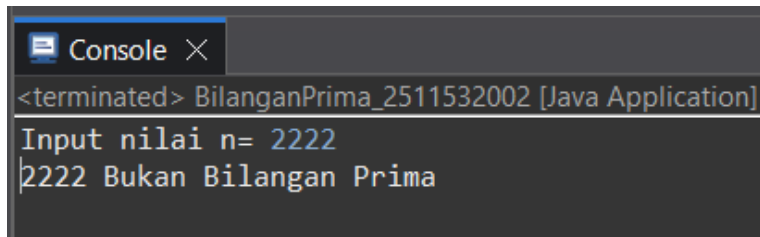
```

<terminated> BilanganPrima_2511532002 [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse
Input nilai n= 99
99 Bukan Bilangan Prima

```

2.2 10 Hasil Uji Program BilanganPrima_2511532002

Pada menu *console* akan muncul hasil program, dimana *user* dapat coba menginputkan bilangan acak dan program akan menentukan apakah bilangan tersebut prima atau tidak.

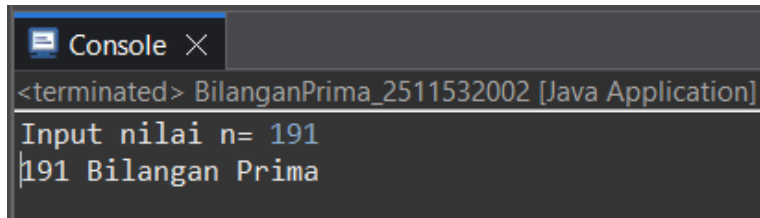


Console

```

<terminated> BilanganPrima_2511532002 [Java Application]
Input nilai n= 2222
2222 Bukan Bilangan Prima

```

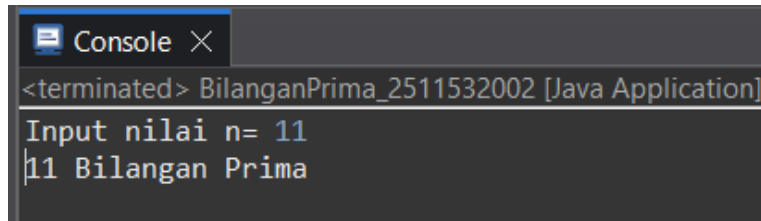


Console

```

<terminated> BilanganPrima_2511532002 [Java Application]
Input nilai n= 191
191 Bilangan Prima

```



Console

```

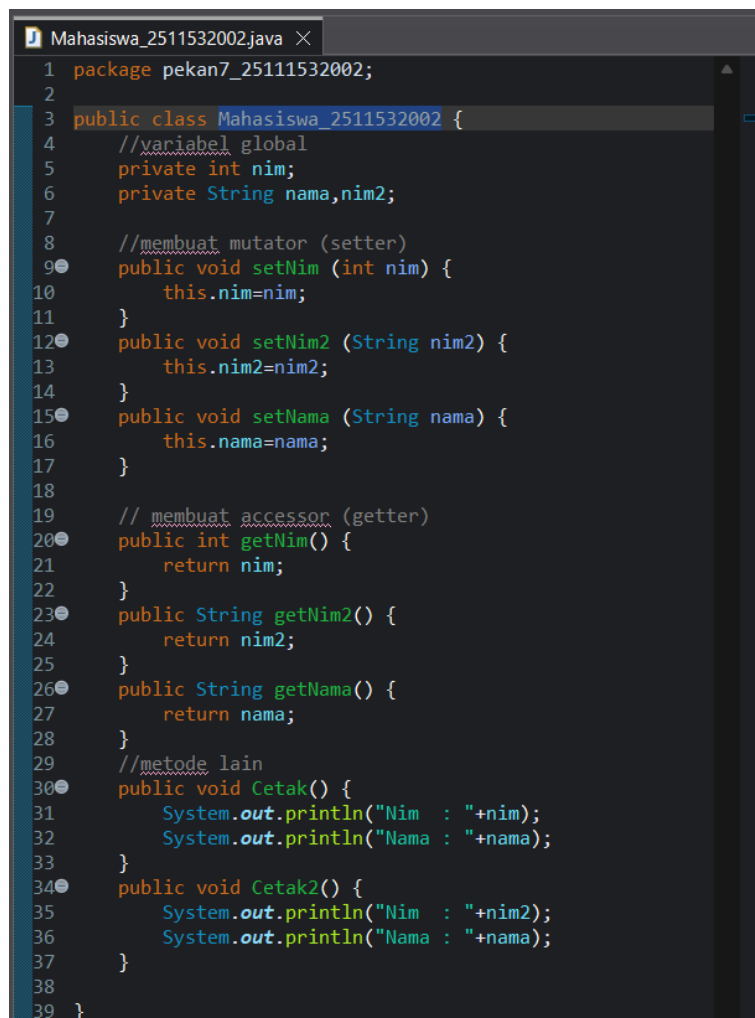
<terminated> BilanganPrima_2511532002 [Java Application]
Input nilai n= 11
11 Bilangan Prima

```

2.2.5 Program Keempat: Mahasiswa_2511532002

Untuk program selanjutnya yaitu program *Mahasiswa_2511532002*, *PanggilMahasiswa_2511532002*, dan *PanggilMahasiswa2_2511532002* saling berkaitan. Dimana *Mahasiswa_2511532002* berfungsi sebagai *class* utama (*blueprint*) yang mendefinisikan atribut dan *method* milik objek “*Mahasiswa*”.

1. Buat kembali *class* baru dengan nama kelas “*Mahasiswa_2511532002*”. Selanjutnya masukan *syntax* seperti berikut.



```

1 package pekan7_2511532002;
2
3 public class Mahasiswa_2511532002 {
4     //variabel global
5     private int nim;
6     private String nama,nim2;
7
8     //membuat mutator (setter)
9     public void setNim (int nim) {
10         this.nim=nim;
11     }
12     public void setNim2 (String nim2) {
13         this.nim2=nim2;
14     }
15     public void setNama (String nama) {
16         this.nama=nama;
17     }
18
19     // membuat accessor (getter)
20     public int getNim() {
21         return nim;
22     }
23     public String getNim2() {
24         return nim2;
25     }
26     public String getNama() {
27         return nama;
28     }
29     //metode lain
30     public void Cetak() {
31         System.out.println("Nim : "+nim);
32         System.out.println("Nama : "+nama);
33     }
34     public void Cetak2() {
35         System.out.println("Nim : "+nim2);
36         System.out.println("Nama : "+nama);
37     }
38 }
39 }
  
```

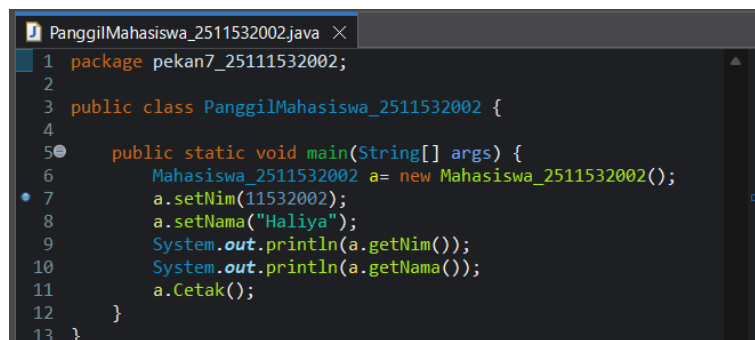
2.2 11 Syntax Program Mahasiswa_2511532002

Deklarasi variabel global (atribut) *nim*, *nim2*, dan *nama* dengan *modifier private*, agar tidak bisa diakses langsung dari luar *class* kecuali melalui *setter* dan *getter*.

2.2.6 Program Kelima: PanggilMahasiswa_2511532002

Program berikut berfungsi sebagai *class* pemanggil (*main class*) yang membuat objek dari *class* “Mahasiswa_2511532002” dan menggunakan *method* pemanggilan *setter* dan *getter*.

2. Buat kembali *class* baru dengan nama kelas “PanggilMahasiswa_2511532002”. Selanjutnya masukan *syntax* seperti berikut.



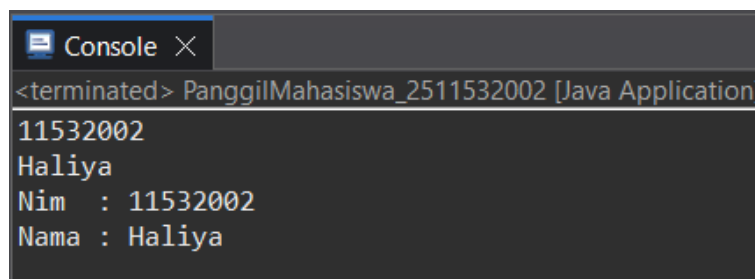
```

1 package pekan7_2511532002;
2
3 public class PanggilMahasiswa_2511532002 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511532002 a= new Mahasiswa_2511532002();
7         a.setNim(11532002);
8         a.setNama("Haliya");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }

```

2.2 12 Syntax Program PanggilMahasiswa_2511532002

- a) Buat objek baru dari *class* Mahasiswa_2511532002
 - b) *Method* setNim() dan setNama() untuk memberikan nilai.
 - c) GetNim() dan getNama() untuk menampilkan hasilnya.
 - d) *method* Cetak() untuk menampilkan data mahasiswa di *console*.
3. Lalu jalankan program, dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka pada menu *console* akan muncul hasil program.



```

<terminated> PanggilMahasiswa_2511532002 [Java Application]
11532002
Haliya
Nim : 11532002
Nama : Haliya

```

2.2 13 Hasil Program PanggilMahasiswa_2511532002

Program memperlihatkan pemanggilan *method setter* dan *getter* dari *class* lain. Setelah objek *a* dibuat, *method* dipanggil untuk menampilkan data “Mahasiswa_2511532002”.

```

1 package pekan7_2511532002;
2
3 public class Mahasiswa_2511532002 {
4     //variabel global
5     private int nim;
6     private String nama,nim2;
7
8     //membuat mutator (setter)
9     public void setNim (int nim) {
10         this.nim=nim;
11     }
12     public void setNim2 (String nim2) {
13         this.nim2=nim2;
14     }
15     public void setNama (String nama) {
16         this.nama=nama;
17     }
18
19     // membuat accessor (getter)
20     public int getNim() {
21         return nim;
22     }
23     public String getNim2() {
24         return nim2;
25     }
26     public String getNama() {
27         return nama;
28     }
29     //metode lain
30     public void cetak() {
31         System.out.println("Nim : "+nim);
32         System.out.println("Nama : "+nama);
33     }
34     public void cetak2() {
35         System.out.println("Nim : "+nim2);
36         System.out.println("Nama : "+nama);
37     }
38 }

```

```

1 package pekan7_2511532002;
2
3 public class PanggilMahasiswa_2511532002 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511532002 a= new Mahasiswa_2511532002();
7         a.setNim(11532002);
8         a.setNama("Haliya");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11    }
12 }
13
14

```

```

<terminated> PanggilMahasiswa_2511532002 [Java Application] C:\Users\user1\p2\pool\
11532002
Haliya
Nim : 11532002
Nama : Haliya

```

2.2 14 Gambaran Hasil Program PanggilMahasiswa_2511532002

2.2.7 Program Keenam: PanggilMahasiswa2_2511532002

Program berikut berfungsi sebagai *class* pemanggil (*main class*) yang membuat objek dari *class* “Mahasiswa_2511532002” dan menggunakan *method* pemanggilan *setter* dan *getter*.

4. Buat kembali *class* baru dengan nama kelas “PanggilMahasiswa2_2511532002”. Selanjutnya masukan *syntax* seperti berikut.

```

1 package pekan7_2511532002;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511532002 {
6
7     public static void main(String[] args) {
8         Scanner input= new Scanner (System.in);
9         System.out.print("NIM : ");
10        String x= input.nextLine();
11        System.out.print("Nama : ");
12        String y= input.nextLine();
13        Mahasiswa_2511532002 a= new Mahasiswa_2511532002();
14        a.setNim2(x);
15        a.setNama(y);
16        if (x.startsWith("25")) {
17            System.out.println(y+ " Anda Angkatan 2025");
18        }
19        if (x.contains("1153")) {
20            System.out.println("Anda Mahasiswa Informatika!");
21        }
22        a.Cetak2();
23        input.close();
24    }
25 }

```

2.2 15 Syntax Program PanggilMahasiswa2_2511532002

Program ini menunjukkan penggunaan *method string* dalam logika program, untuk mengolah/ mengecek data yang :

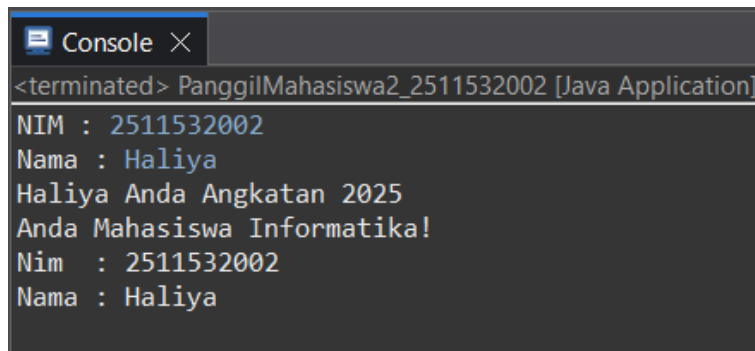
- a) *startsWith()* untuk memeriksa awalan teks.

Seperti pada *syntax x.startsWith("25")* mengecek apakah NIM diawali “25”(angkatan 2025).

- b) *contains()* untuk mencari substring di dalam string.

Seperti pada *x.contains("I153")*, mengecek apakah NIM berisi “I153”(Informatika).

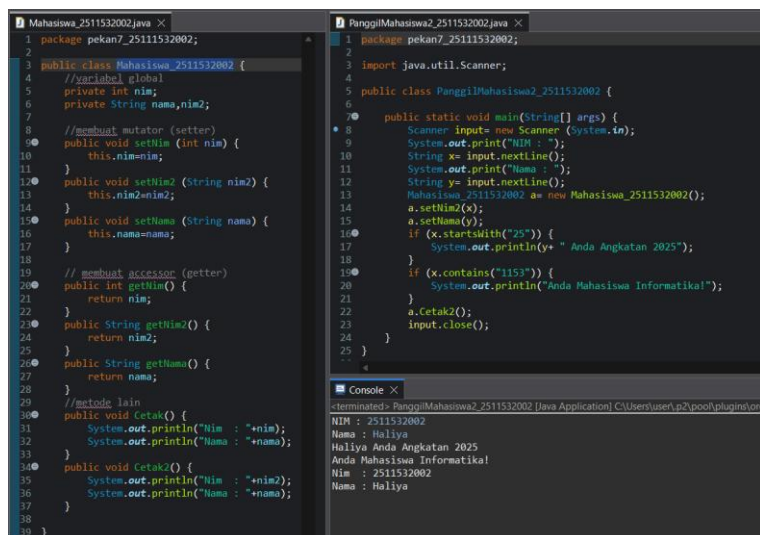
5. Lalu jalankan program, dengan klik tombol *Run* di pojok kiri atas pada *toolbar*. Maka pada menu *console* akan muncul hasil program.



```

<terminated> PanggilMahasiswa2_2511532002 [Java Application]
NIM : 2511532002
Nama : Haliya
Haliya Anda Angkatan 2025
Anda Mahasiswa Informatika!
Nim : 2511532002
Nama : Haliya
  
```

2.2 17 Hasil Program PanggilMahasiswa2_2511532002



```

1 package pekan7_2511532002;
2
3 public class Mahasiswa_2511532002 {
4     //variabel global
5     private int nim;
6     private String nama,nim2;
7
8     //membuat mutator (setter)
9     public void setNim (int nim) {
10         this.nim=nim;
11     }
12     public void setNim2 (String nim2) {
13         this.nim2=nim2;
14     }
15     public void setName (String nama) {
16         this.nama=nama;
17     }
18
19     // membuat accessor (getter)
20     public int getNim() {
21         return nim;
22     }
23     public String getNim2() {
24         return nim2;
25     }
26     public String getNama() {
27         return nama;
28     }
29
30     //metode lain
31     public void cetak() {
32         System.out.println("Nim : "+nim);
33         System.out.println("Nama : "+nama);
34     }
35     public void cetak2() {
36         System.out.println("Nim : "+nim2);
37         System.out.println("Nama : "+nama);
38     }
39 }
  
```

```

1 package pekan7_2511532002;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511532002 {
6
7     public static void main(String[] args) {
8         Scanner input= new Scanner (System.in);
9         System.out.print("NIM : ");
10        String x= input.nextLine();
11        System.out.print("Nama : ");
12        String y= input.nextLine();
13        Mahasiswa_2511532002 a= new Mahasiswa_2511532002();
14        a.setNim2(x);
15        a.setName(y);
16        if (x.startsWith("25")) {
17            System.out.println("y= " + " Anda Angkatan 2025");
18        }
19        if (x.contains("I153")) {
20            System.out.println("Anda Mahasiswa Informatika!");
21        }
22        a.cetak2();
23        input.close();
24    }
25 }
  
```

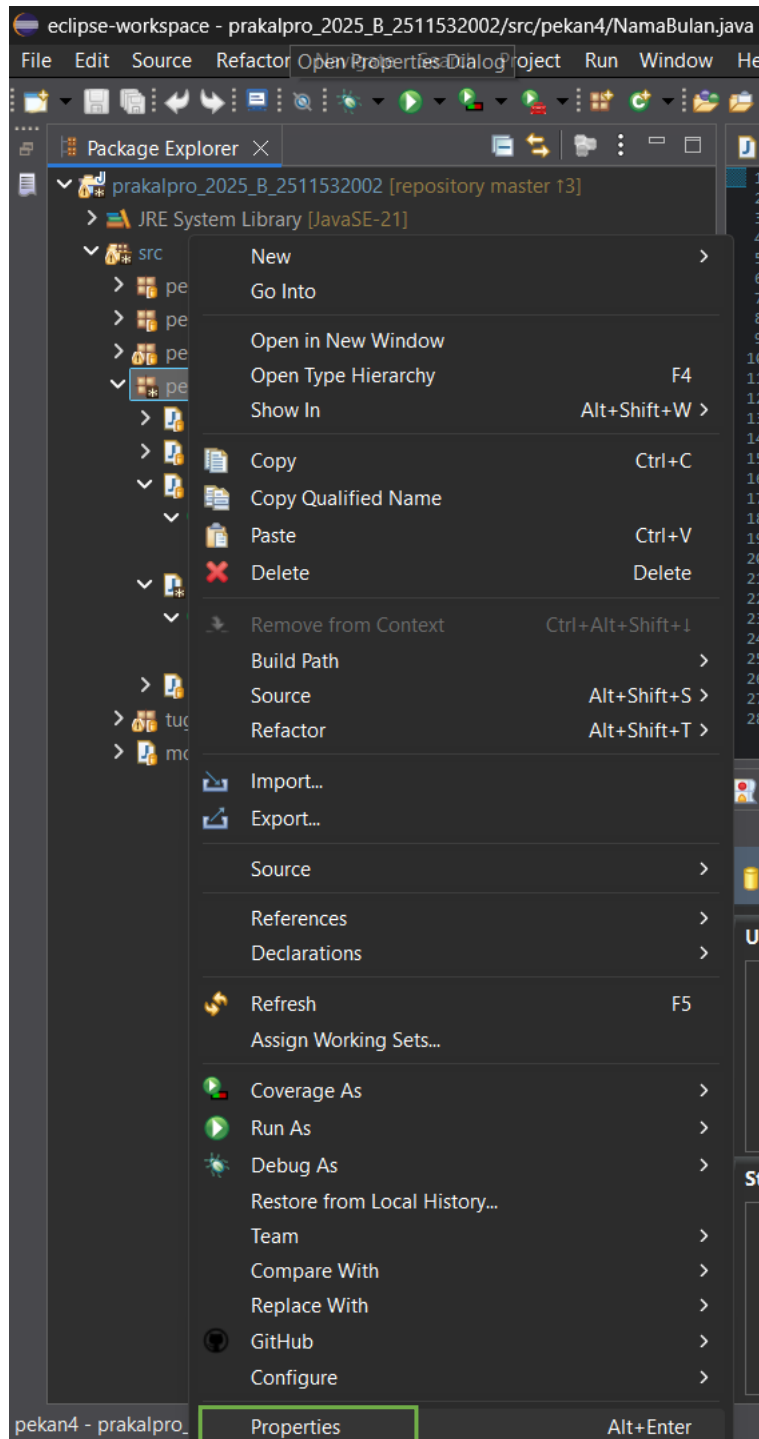
```

<terminated> PanggilMahasiswa2_2511532002 [Java Application] C:\Users\user\p2\pool\plugins\org
NIM : 2511532002
Nama : Haliya
Haliya Anda Angkatan 2025
Anda Mahasiswa Informatika!
Nim : 2511532002
Nama : Haliya
  
```

2.2 16 Gambaran Hasil Program PanggilMahasiswa2_2511532002

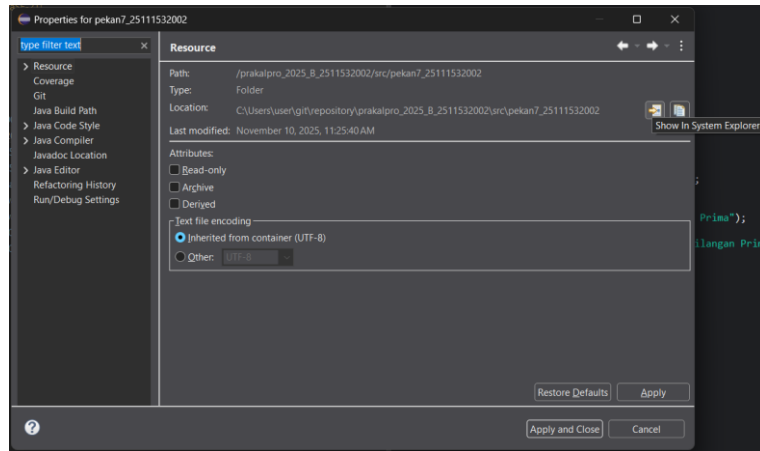
2.2.8 Upload ke GitHub

1. Setelah semua program selesai dibuat, selanjutnya adalah *user* perlu untuk memasukkan program yang dibuat di Eclipse ke *repository* GitHub. Selain dengan cara “*Team*” atau “*Commit and Push*”. Bisa juga dengan cara *upload* manual. Caranya klik kanan pada *package*, lalu pilih *properties*.



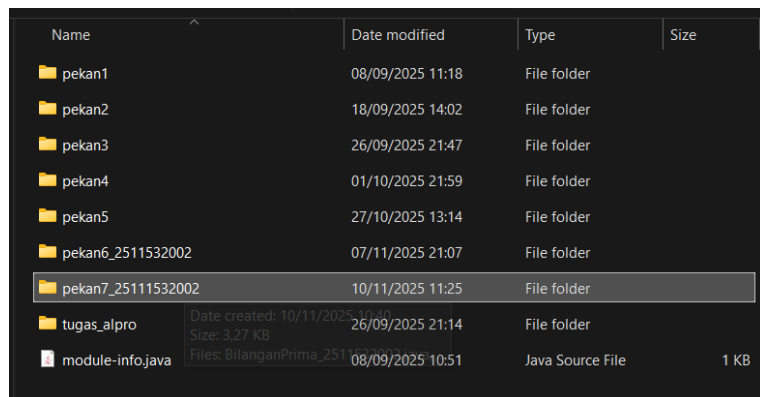
2.2 18 Package Properties of pekan7_2511532002

2. Maka akan muncul *pop up menu* “*Properteis for pekan7_2511532002*”. Lalu klik *icon* “*Show in System Folder*”.



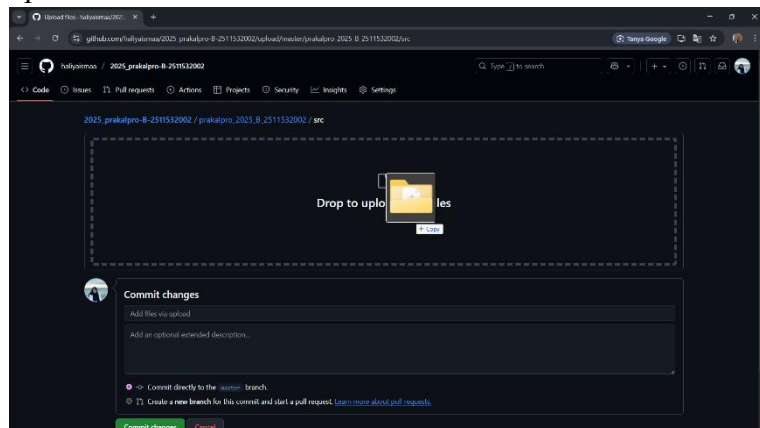
2.2 19 Properteis for pekan7_2511532002

3. Maka akan menunjukan lokasi *file* “*pekan7_2511532002*” pada *File Explorer computer*.



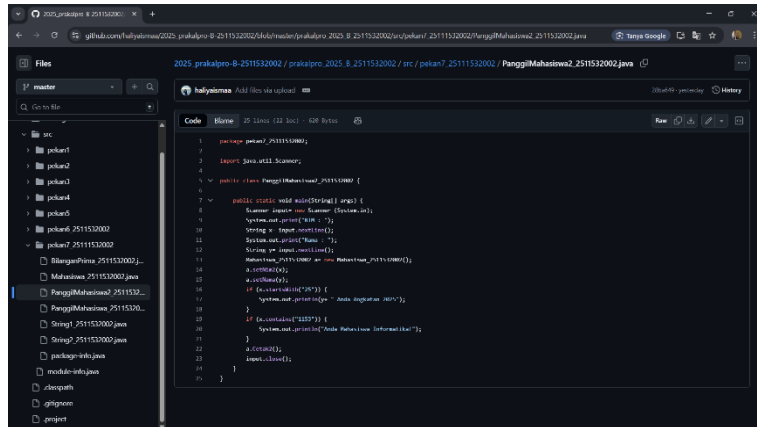
2.2 20 File Location of pekan7_2511532002

4. Setelah itu *drag and drop* file “*pekan7_2511532002*” ke GitHub. Lalu tambahkan keterangan jika perlu. Setelah itu klik *upload*.



2.2 21 Drag and Drop file pekan7_2511532002 ke GitHub

5. Maka dapat dilihat pada *repository GitHub* bahwa file “*peka7_2511532002*” telah terunggah via *Upload*.



2.2 22 Hasil Upload Package pekan7_2511532002 ke GitHub

Dapat kita lihat disini jika kita buka, maka akan muncul *file package* dan *class* yang telah di buat di Eclipse.

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan dengan topik *String dan Method* pada pemrograman *Java*. Dapat disimpulkan bahwa dapat disimpulkan bahwa *class String* memiliki banyak *method* bawaan yang memudahkan pengolahan teks tanpa perlu membuat algoritma tambahan secara manual. *String* pada *Java* bersifat *immutable*, artinya nilai *string* tidak dapat diubah setelah dibuat. Dengan memahami berbagai *method* seperti *.length()*, *.toUpperCase()*, *.toLowerCase()*, dan *.indexOf()*, dapat membantu *user* lebih mudah mengelola dan memanipulasi data teks dalam program

3.2 Saran

Saran untuk pelaksanaan praktikum kedepannya adalah diharapkan agar mahasiswa lebih teliti dalam memahami *syntax* agar meminimalisir kesalahan penulisan kode. Selain itu, disarankan menggunakan *method concat()* saat melakukan penggabungan *string* agar program lebih efisien. Mahasiswa juga perlu lebih teliti dalam memahami perbedaan antara tipe data *String* dan *numerik*, karena kesalahan logika pada penggabungan keduanya dapat menyebabkan *output* yang tidak sesuai dengan yang diharapkan.

DAFTAR PUSTAKA

- [1] W3Schools, “Java String Methods,” [Daring]. Tersedia pada: https://www.w3schools.com/java/java_ref_string.asp [Diakses: 11-Nov-2025].
- [2] GeeksforGeeks, “Java String Methods,” [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/java/java-string-methods/> [Diakses: 11-Nov-2025].
- [3] Nextgen, “String pada Java,” [Daring]. Tersedia pada: <https://nextgen.co.id/string-pada-java/> [Diakses: 11-Nov-2025].