

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

"Method, Object, dan Operasi String dalam Pemrograman Java

PEKAN 7



OLEH:

LEXI MULIA YUNASPI

(2511531006)

DOSEN PENGAMPU:

DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Segala puji dan syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan kesempatan yang diberikan kepada saya sehingga dapat menyelesaikan laporan mengenai beberapa kode program yang dibuat pada pekan 7 ini dengan baik dan tepat waktu. Laporan ini disusun sebagai bagian dari proses pembelajaran untuk mengembangkan kemampuan dalam merancang dan mengimplementasikan suatu program secara sistematis dan terstruktur. Saya menyadari bahwa dalam proses penyusunan laporan ini masih jauh dari sempurna, oleh karena itu saya sangat mengharapkan kritik dan saran yang membangun guna penyempurnaan karya dan pengetahuan kami di masa mendatang.

Saya juga mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bimbingan, dorongan, dan dukungan, khususnya kepada dosen pengampu dan asisten praktikum yang memotivasi saya dalam menyelesaikan tugas ini. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna tidak hanya bagi saya sebagai penyusun, tetapi juga bagi pembaca yang ingin mempelajari lebih dalam mengenai proses pembuatan program secara terstruktur dan sistematis. Harapan saya, pengetahuan yang diperoleh melalui laporan ini dapat digunakan sebagai bekal dalam mengembangkan kemampuan di bidang pemrograman dan teknologi informasi.

Padang, 14 November 2025

Lexi Mulia Yunaspi

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum.....	1
1.2 Manfaat Praktikum.....	1
1.3 Tujuan Praktikum.....	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Praktikum.....	30
BAB III PENUTUP.....	31
3.1 Kesimpulan.....	31
3.2 Saran.....	31
DAFTAR PUSTAKA.....	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Praktikum pemrograman Java ini dilakukan supaya mahasiswa bisa lebih paham cara kerja dasar dari sebuah program, bukan cuma lihat teori di kelas. Lewat kegiatan ini, mahasiswa belajar gimana sebuah program dibangun mulai dari penggunaan method, class, dan object sampai manipulasi data seperti operasi pada String. Praktikum ini juga jadi kesempatan untuk mencoba langsung konsep OOP yang sering dianggap sulit kalau cuma dibaca. Dengan praktik langsung, mahasiswa jadi lebih mudah memahami logika program, alur eksekusi kode, dan hubungan antarbagian di dalam program.

1.2 Tujuan Praktikum

1. Memahami cara membuat dan memakai method dalam program Java.
2. , seperti atribut, setter, getter, dan cara memanggil method melalui objek. Belajar konsep dasar OOP
3. mempraktikkan operasi dasar pada String, seperti menggabungkan teks, mencari karakter, mengubah huruf, dan memanipulasi isi string lainnya.
4. Meningkatkan pemahaman logika pemrograman melalui percobaan langsung pada kode.
5. Memberikan pengalaman mengembangkan program sederhana tapi terstruktur.

1.3 Manfaat Praktikum

1. Mahasiswa jadi lebih terbiasa menulis kode dan memahami alurnya.
- 2, Membantu mahasiswa mengenal konsep OOP lebih cepat karena langsung praktik, bukan hanya teori.

3. Memudahkan mahasiswa mengerjakan tugas kuliah yang berkaitan dengan method, class, dan objek.
4. Mahasiswa bisa mengembangkan logika pemrograman dan lebih percaya diri saat menulis program sendiri.
5. Jadi dasar kuat untuk materi pemrograman berikutnya yang lebih kompleks.

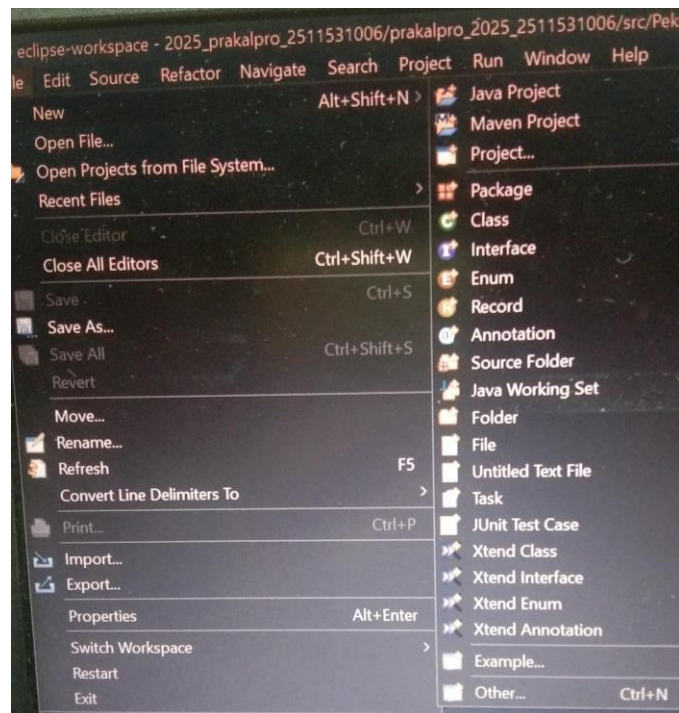
BAB II

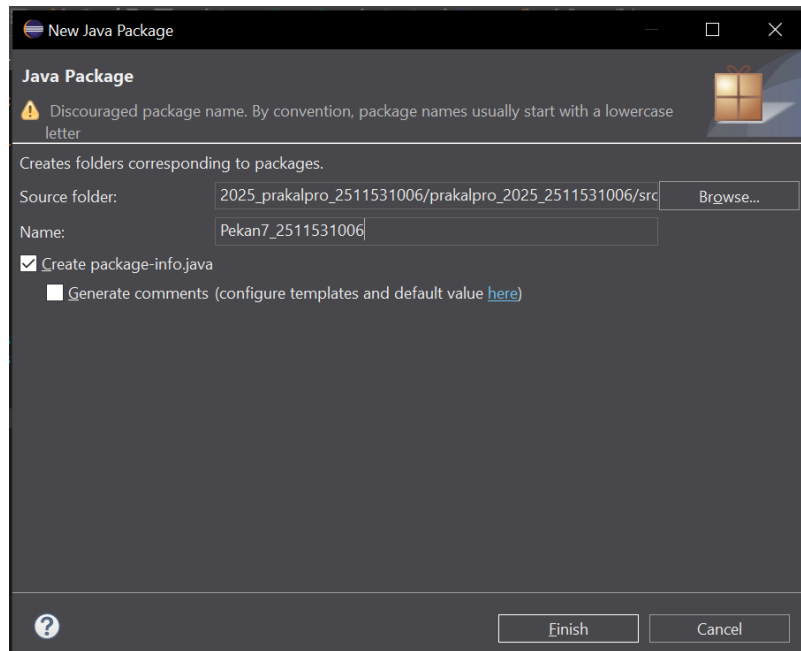
PEMBAHASAN

2.1 Langkah Praktikum

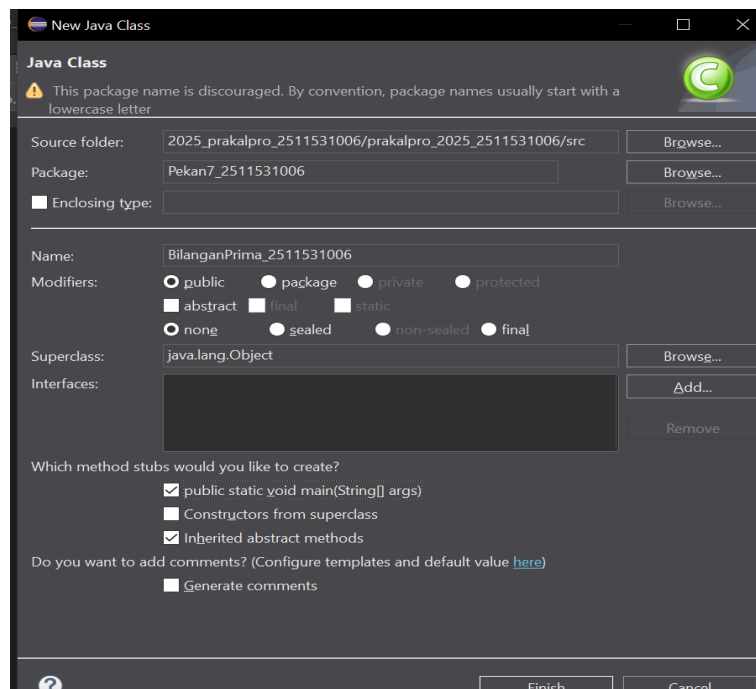
1. Program Pertama

- a. Buatlah package terlebih dahulu dengan mengklik kanan di folder src. Setelah itu beri nama pada package tanpa huruf kapital, karakter khusus serta tanpa “space.

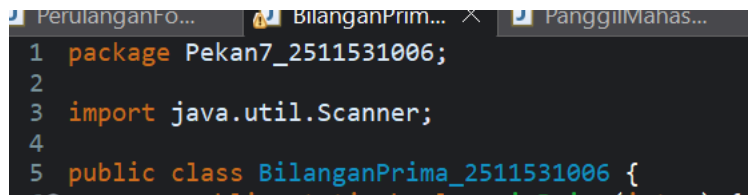




- b. Setelah itu pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



- c. Potongan kode Java pada gambar tersebut berfungsi untuk mendefinisikan struktur awal dari sebuah program Java yang berada dalam sebuah paket dan menggunakan library bawaan Java. Baris pertama `package Pekan7_2511531006;` menunjukkan bahwa file program ini berada di dalam paket (folder) bernama `Pekan7_2511531006`, yang biasanya digunakan untuk mengelompokkan file atau program berdasarkan minggu, tugas, atau proyek tertentu agar lebih terorganisir. Baris kedua `import java.util.Scanner;` berfungsi untuk mengimpor kelas `Scanner` dari pustaka Java, yang digunakan untuk membaca input dari pengguna melalui keyboard. Selanjutnya, baris `public class BilanganPrima_2511531006 {` mendefinisikan sebuah kelas publik dengan nama `BilanganPrima_2511531006`, yang kemungkinan besar berisi logika utama program untuk memeriksa atau menampilkan bilangan prima. Secara keseluruhan, kode ini adalah bagian awal dari struktur program Java yang menunjukkan identitas paket, kebutuhan pustaka input, dan deklarasi kelas utama tempat logika program akan dijalankan.



```
1 package Pekan7_2511531006;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511531006 {
```

- d. Potongan kode pada gambar tersebut merupakan bagian dari sebuah fungsi atau method di dalam program Java yang digunakan untuk mengecek apakah suatu bilangan termasuk bilangan prima atau tidak. Method tersebut bernama `isPrime` dan memiliki tipe data kembalian boolean, yang berarti akan mengembalikan nilai `true` jika bilangan tersebut prima dan `false` jika bukan. Parameter `int n` menunjukkan bahwa fungsi ini menerima sebuah bilangan bulat sebagai input. Di dalamnya, terdapat variabel `factors` yang diinisialisasi dengan nilai 0, berfungsi untuk menghitung jumlah faktor dari bilangan `n`. Kemudian, terdapat loop `for` yang berjalan dari `i = 1` hingga `i <= n`, artinya setiap angka dari 1 sampai `n` akan diuji. Di dalam perulangan, terdapat kondisi `if (n % i == 0)` yang digunakan untuk

memeriksa apakah n habis dibagi oleh i . Jika kondisi ini benar, berarti i adalah salah satu faktor dari n . Dengan logika ini, program nantinya akan menentukan apakah n memiliki tepat dua faktor (1 dan dirinya sendiri), yang merupakan ciri khas bilangan prima.

```
6 public static boolean isPrime(int n) {  
7     int factors = 0;  
8     for (int i = 1; i <= n; i++) {  
9         if (n % i == 0) {
```

- e. Potongan kode ini merupakan lanjutan dari method `isPrime` yang berfungsi untuk menentukan apakah sebuah bilangan merupakan bilangan prima. Pada bagian yang terlihat, terdapat perintah `factors++`; yang berarti menambah nilai variabel `factors` sebanyak satu setiap kali ditemukan angka i yang dapat membagi habis nilai n (yakni ketika $n \% i == 0$). Dengan kata lain, setiap kali ditemukan faktor dari n , penghitung faktor akan bertambah satu. Setelah seluruh perulangan selesai, baris `return (factors == 2)`; digunakan untuk mengembalikan nilai `true` atau `false`. Jika jumlah faktor dari bilangan n adalah tepat dua (yakni 1 dan bilangan itu sendiri), maka ekspresi `factors == 2` akan bernilai `true`, artinya bilangan tersebut adalah bilangan prima. Namun, jika faktor lebih dari dua, maka hasilnya `false` karena bilangan tersebut dapat dibagi oleh angka lain selain 1 dan dirinya sendiri. Jadi, bagian kode ini adalah inti logika yang menentukan status keprimaan suatu bilangan berdasarkan jumlah faktornya.

```
10         factors++;  
11     }  
12 }  
13 }  
14 return (factors == 2);  
15 }  
16 }
```

- f. Potongan kode Java pada gambar tersebut merupakan bagian dari metode `main`, yaitu titik awal eksekusi program Java. Pada baris pertama, `Scanner input = new`

Scanner(System.in); digunakan untuk membuat objek Scanner bernama input, yang berfungsi membaca data yang diketikkan oleh pengguna melalui keyboard (System.in). Dengan kata lain, baris ini memungkinkan program menerima input dari pengguna. Selanjutnya, baris System.out.print("Input nilai n = "); digunakan untuk menampilkan teks "Input nilai n = " ke layar tanpa berpindah ke baris baru, sebagai petunjuk agar pengguna memasukkan suatu nilai. Jadi, dua baris kode ini bekerja sama untuk meminta dan menerima input dari pengguna yang akan digunakan dalam proses selanjutnya di dalam program.

```
17 public static void main(String[] args) {  
18     Scanner input = new Scanner(System.in);  
19     System.out.print("Input nilai n = ");
```

- g. Potongan kode tersebut membaca input angka dari pengguna menggunakan `int a = input.nextInt();`, lalu memeriksa apakah angka tersebut bilangan prima dengan `if (isPrime(a))`. Jika `isPrime(a)` bernilai true, maka program menjalankan perintah dalam blok `if`.

```
20     int a = input.nextInt();  
21     if (isPrime(a)) {
```

- h. Potongan kode tersebut menampilkan hasil dari pengecekan bilangan prima. Jika kondisi `if (isPrime(a))` bernilai benar, maka program menampilkan teks `<angka>` bilangan prima. Namun jika tidak, blok `else` dijalankan dan menampilkan `<angka>` bukan bilangan prima. Jadi bagian ini berfungsi memberi tahu pengguna apakah angka yang dimasukkan termasuk bilangan prima atau bukan.

```

22         System.out.println(a+"bilangan prima");
23     } else {
24         System.out.println(a+ "bukan bilangan prima");
25     }
26 }
27 }
28 }
29 }

```

- i. Tampilan semua kode program dari awal sampai akhir.

```

1 package Pekan7_2511531006;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511531006 {
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
16     public static void main(String[] args) {
17         Scanner input = new Scanner(System.in);
18         System.out.print("Input nilai n = ");
19         int a = input.nextInt();
20         if (isPrime(a)) {
21             System.out.println(a+"bilangan prima");
22         } else {
23             System.out.println(a+ "bukan bilangan prima");
24         }
25     }
26 }
27 }
28 }
29 }

```

- j. Hasil output program

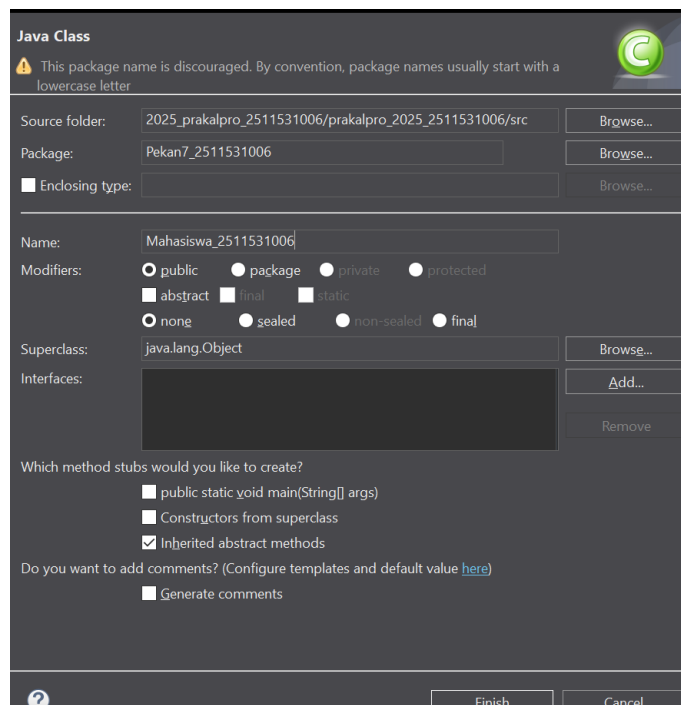
```

Console X
<terminated> BilanganPrima_2511531006 [Java Application] C
Input nilai n = 2
2bilangan prima

```

2.Kode Program Kedua

- a. pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



- b. Potongan kode tersebut mendefinisikan struktur awal program Java. Baris package Pekan7_2511531006; menunjukkan bahwa file ini berada dalam paket bernama Pekan7_2511531006, yang digunakan untuk mengelompokkan kelas agar lebih terorganisasi. Sementara baris public class Mahasiswa_2511531006 { mendefinisikan sebuah kelas publik bernama Mahasiswa_2511531006, yang menjadi wadah utama untuk mendeklarasikan variabel, metode, dan logika program yang berkaitan dengan objek mahasiswa.

```

1 package Pekan7_2511531006;
2
3 public class Mahasiswa_2511531006 {

```

- c. Potongan kode tersebut mendeklarasikan dua variabel global dalam kelas, yaitu nim dan nama, keduanya bertipe String. Kata kunci private berarti variabel ini hanya dapat diakses dari dalam kelas yang sama, sehingga data mahasiswa lebih terlindungi dari perubahan langsung oleh kelas lain. Biasanya variabel seperti ini akan diakses atau diubah melalui *method* getter dan setter.

```

4 // Variabel global
5 private String nim;
6 private String nama;

```

- d. Potongan kode pada gambar tersebut merupakan sebuah method setter dalam bahasa pemrograman Java, yang berfungsi untuk memberikan atau mengubah nilai dari atribut nim di dalam suatu objek. Method ini bernama setNim dan memiliki satu parameter bertipe String bernama nim. Di dalam tubuh method, baris `this.nim = nim;` digunakan untuk membedakan antara variabel instance (`this.nim`) dan parameter lokal (`nim`) yang memiliki nama sama. Kata kunci `this` mengacu pada atribut nim milik objek saat ini, sehingga nilai yang diterima dari parameter akan disimpan ke atribut tersebut. Dengan cara ini, method `setNim` memungkinkan pengguna program untuk memperbarui nilai NIM mahasiswa secara aman melalui mekanisme enkapsulasi.

```

7 // membuat mutator (setter)
8 public void setNim(String nim) {
9     this.nim = nim;
10 }

```

- e. Potongan kode tersebut merupakan method setter bernama `setNama` dalam bahasa Java yang digunakan untuk menetapkan atau mengubah nilai atribut nama pada

suatu objek. Method ini memiliki parameter bertipe String dengan nama nama, yang menampung nilai baru yang ingin diberikan. Di dalam tubuh method, perintah `this.nama = nama;` berfungsi untuk menyimpan nilai parameter ke dalam atribut objek. Kata kunci `this` digunakan agar program dapat membedakan antara variabel instance nama milik objek dan parameter lokal nama.

```
11 public void setName(String nama) {  
12     this.nama = nama;  
13 }
```

- f. Potongan kode tersebut merupakan method getter dalam bahasa Java dengan nama `getNim`. Method ini berfungsi untuk mengambil atau mengembalikan nilai dari atribut `nim` yang terdapat pada suatu objek. Method ini memiliki tipe pengembalian String, karena nilai `nim` sendiri bertipe String. Di dalam tubuh method, terdapat perintah `return nim;` yang berarti method ini akan mengembalikan nilai dari atribut `nim` ketika dipanggil.

```
14 public String getNim() {  
15     return nim;  
16 }
```

- g. Potongan kode tersebut merupakan sebuah method getter dalam bahasa pemrograman Java yang digunakan untuk mengambil atau mengakses nilai dari variabel nama yang biasanya bersifat *private* di dalam suatu kelas. Method ini bernama `getNama()` dan bertipe String, artinya nilai yang dikembalikan oleh method ini berupa teks. Di dalam tubuh method, terdapat perintah `return nama;` yang berarti method ini akan mengembalikan nilai dari variabel nama ketika dipanggil. Dengan adanya method ini, data nama tetap terlindungi (karena tidak diakses langsung), tetapi tetap dapat diketahui nilainya oleh kelas lain melalui pemanggilan `objek.getNama()`.

```

17 public String getNama() {
18     return nama;
19 }

```

- h. Potongan kode tersebut adalah sebuah method bernama Cetak() dalam bahasa pemrograman Java yang digunakan untuk menampilkan data mahasiswa ke layar (output console). Method ini memiliki tipe void, artinya tidak mengembalikan nilai apa pun. Di dalam method, terdapat dua baris perintah System.out.println() yang digunakan untuk mencetak teks ke layar. Baris pertama menampilkan NIM, sedangkan baris kedua menampilkan Nama. Penggunaan kata kunci this di depan nim dan nama menunjukkan bahwa nilai yang diakses berasal dari variabel milik objek saat ini, bukan dari variabel lain di luar kelas. Dengan demikian, method ini berfungsi untuk menampilkan informasi identitas mahasiswa (NIM dan Nama) dari objek yang memanggilnya.

```

20 public void Cetak() {
21     System.out.println("NIM : " + this.nim);
22     System.out.println("Nama : " + this.nama);
23 }
24 }

```

- i. Tampilan semua kode program ke 3

```

1 package Pekan7_2511531006;
2
3 public class Mahasiswa_2511531006 {
4     // Variabel global
5     private String nim;
6     private String nama;
7     // membuat mutator (setter)
8     public void setNim(String nim) {
9         this.nim = nim;
10    }
11    public void setName(String nama) {
12        this.nama = nama;
13    }
14    public String getNim() {
15        return nim;
16    }
17    public String getNama() {
18        return nama;
19    }
20    public void Cetak() {
21        System.out.println("NIM : " + this.nim);
22        System.out.println("Nama : " + this.nama);
23    }
24 }

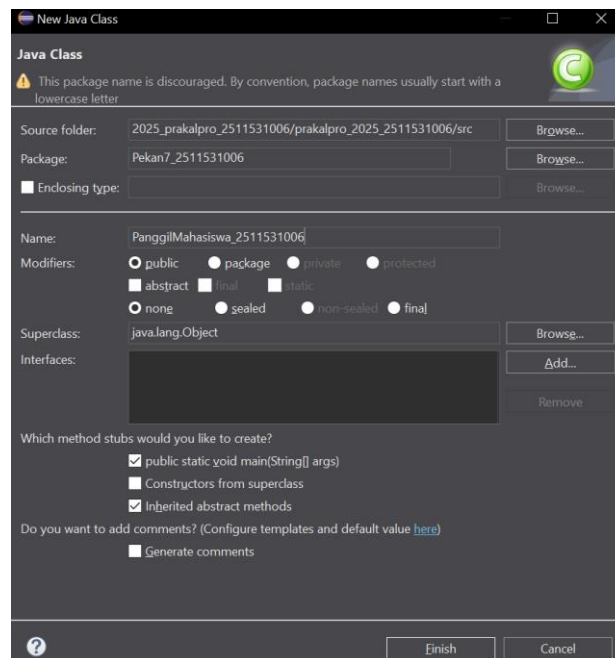
```

j. Output program

```
1  
5  
7  
NIM : 2511531006  
Nama : Lexi
```

3.Kode Program Ketiga

- a. pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



- b. Jika sudah ditekan finish, maka akan muncul seperti ini

```
1 package Pekan7_2511531006;  
2  
3 public class PanggilMahasiswa_2511531006 {  
4     public static void main(String[] args) {
```

- c. Potongan kode pada gambar tersebut menunjukkan proses pembuatan sebuah objek dari kelas Mahasiswa_2511531006 dan kemudian mengisi data pada

objek tersebut melalui method setter. Pada baris pertama, objek bernama a dibuat menggunakan `new Mahasiswa_2511531006()`, yang berarti program sedang menciptakan satu instance baru dari kelas mahasiswa tersebut. Setelah objek berhasil dibuat, baris `a.setNim("2511531006");` digunakan untuk memberikan nilai NIM kepada objek a, di mana nilai NIM dimasukkan sebagai string. Kemudian baris `a.setNama("Lexi Mulia Yunaspi");` berfungsi untuk mengisi atribut nama pada objek yang sama. Dengan demikian, ketiga baris kode ini bekerja bersama untuk membuat objek mahasiswa serta mengatur data identitasnya melalui method setter agar dapat digunakan pada proses selanjutnya dalam program.

```
5      Mahasiswa_2511531006 a = new Mahasiswa_2511531006();
6      a.setNim("2511531006");
7      a.setNama("Lexi Mulia Yunaspi");
```

- d. Potongan kode tersebut berfungsi untuk menampilkan nilai yang sudah disimpan dalam objek melalui method getter. Baris `System.out.println(a.getNim());` memanggil method `getNim()` pada objek a, yang berarti program meminta nilai NIM yang sebelumnya telah diset melalui setter, lalu mencetaknya ke layar. Begitu juga dengan baris `System.out.println(a.getNama());`, yang memanggil method `getNama()` untuk mengambil nilai nama dari objek tersebut dan menampilkannya. Dengan demikian, kedua baris ini digunakan untuk memastikan bahwa nilai yang disimpan di dalam objek mahasiswa sudah benar serta menampilkan data tersebut sebagai output ke pengguna.

```
8      System.out.println(a.getNim());
9      System.out.println(a.getNama());
```

- e. Baris kode `a.Cetak();` digunakan untuk memanggil sebuah method bernama `Cetak()` yang berada di dalam kelas `Mahasiswa_2511531006`. Method ini

biasanya dibuat untuk menampilkan seluruh data mahasiswa secara lengkap dalam satu tampilan, misalnya menampilkan NIM, nama, atau atribut lain yang dimiliki objek tersebut. Jadi, daripada memanggil `println()` satu per satu, method `Cetak()` memberikan cara yang lebih praktis untuk mencetak data objek sekaligus. Dengan memanggil `a.Cetak();`, program meminta objek `a` untuk menjalankan fungsinya sendiri, yaitu mencetak atau menampilkan informasi yang sudah tersimpan di dalamnya.

```
10         a.Cetak();
11     }
12 }
```

f. Tampilan semua kode program

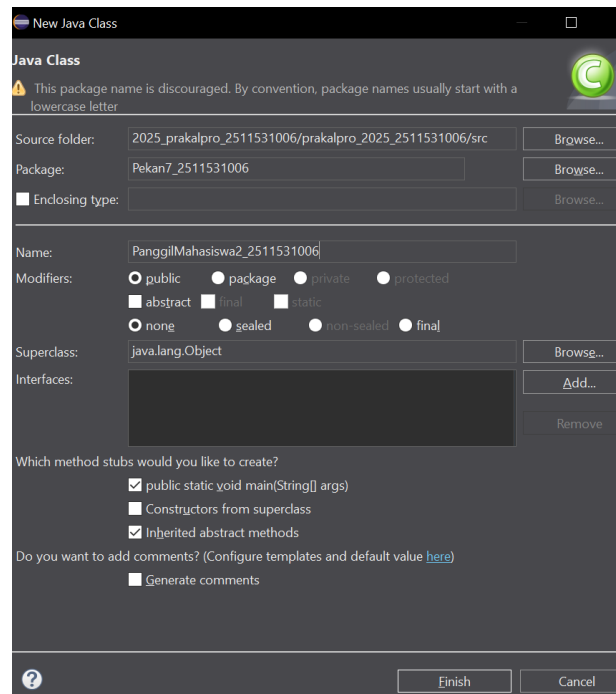
```
1 package Pekan7_2511531006;
2
3 public class PanggilMahasiswa_2511531006 {
4     public static void main(String[] args) {
5         Mahasiswa_2511531006 a = new Mahasiswa_2511531006();
6         a.setNim("2511531006");
7         a.setNama("Lexi Mulia Yunaspi");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12 }
```

g. Hasil output program

```
Console X
<terminated> PanggilMahasiswa_2511531006 [Java A
2511531006
Lexi Mulia Yunaspi
NIM    : 2511531006
Nama   : Lexi Mulia Yunaspi
```

4.Kode Program keempat

- a. pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



- b. Jika sudah ditekan finish, maka akan muncul seperti ini

```
1 package Pekan7_2511531006;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531006 {
6     public static void main(String[] args) {
```

- c. Potongan kode tersebut menunjukkan proses pembuatan objek Scanner dan pendefinisian dua variabel string. Baris Scanner input = new Scanner(System.in); berfungsi untuk membuat alat pembaca input dari keyboard, sehingga program sebenarnya siap menerima masukan dari pengguna jika dibutuhkan. Namun, pada bagian berikutnya justru digunakan dua variabel string yang sudah diisi secara langsung, yaitu

String x = "2511531006"; untuk menyimpan NIM dan String y = "Lexi Mulia Yunaspi"; untuk menyimpan nama. Artinya, data yang akan digunakan pada program tidak diambil dari input pengguna, tetapi diberikan secara langsung (hard-coded). Bagian ini biasanya digunakan untuk keperluan testing atau contoh sederhana agar program bisa berjalan tanpa memerlukan input dari luar.

```
7      Scanner input = new Scanner(System.in);
8
9      String x = "2511531006";
10     String y = "Lexi Mulia Yunaspi";
11
```

- d. Dua baris kode tersebut digunakan untuk menampilkan isi variabel x dan y ke layar. Pada baris `System.out.println(x);`, program mencetak nilai yang tersimpan di dalam variabel x, yaitu NIM yang sebelumnya sudah di-set secara langsung. Baris berikutnya, `System.out.println(y);`, mencetak isi variabel y, yaitu nama mahasiswa. Kedua baris ini memastikan bahwa nilai yang telah disimpan dalam kedua variabel tersebut benar dan dapat ditampilkan sebagai output. Dengan kata lain, bagian ini berfungsi untuk menunjukkan hasil data yang sudah diberikan ke variabel tanpa melalui proses setter atau objek.

```
11
12     System.out.println(x);
13     System.out.println(y);
14
```

- e. Potongan kode ini menunjukkan proses pembuatan objek baru serta pengisian data menggunakan method setter. Pada baris `Mahasiswa_2511531006 a = new Mahasiswa_2511531006();`, program membuat sebuah objek mahasiswa bernama **a** dari kelas

Mahasiswa_2511531006. Setelah objek berhasil dibuat, nilai yang sebelumnya sudah disimpan di variabel **x** dan **y** digunakan sebagai input untuk mengisi data objek tersebut. Baris `a.setNim(x);` memberikan nilai NIM ke dalam objek menggunakan nilai dari variabel **x**, sedangkan baris `a.setNama(y);` memberikan nilai nama menggunakan isi variabel **y**. Dengan cara ini, data yang awalnya disimpan dalam variabel biasa dipindahkan ke dalam atribut objek mahasiswa, sehingga objek tersebut memiliki data lengkap yang bisa digunakan pada proses berikutnya seperti pencetakan atau pengolahan data lain.

```
15 Mahasiswa_2511531006 a = new Mahasiswa_2511531006();  
16 a.setNim(x);  
17 a.setNama(y);
```

- f. Potongan kode ini berfungsi untuk menganalisis isi NIM yang tersimpan dalam variabel **x**, kemudian menampilkan informasi tambahan berdasarkan pola tertentu di dalam NIM tersebut. Pada blok pertama, `if (x.startsWith("25"))`, program memeriksa apakah NIM dimulai dengan angka “25”. Jika benar, maka program menampilkan pesan `y + " anda angkatan 2025"`, yang berarti nama mahasiswa akan digabungkan dengan informasi bahwa ia termasuk angkatan 2025. Pada blok kedua, `if (x.contains("1153"))`, program memeriksa apakah di dalam NIM terdapat urutan angka “1153”. Jika kondisi ini terpenuhi, maka ditampilkan pesan “Anda Mahasiswa Informatika”, karena biasanya kode bagian program studi disisipkan di tengah NIM. Dengan demikian, dua kondisi ini digunakan untuk mengidentifikasi tahun angkatan dan program studi mahasiswa berdasarkan struktur NIM yang diberikan.

```

18
19     if (x.startsWith("25")) {
20         System.out.println(y + " anda angkatan 2025");
21     }
22
23     if (x.contains("1153")) {
24         System.out.println("Anda Mahasiswa Informatika");
25     }
26

```

- g. Potongan kode pada gambar tersebut menunjukkan dua pemanggilan method yang terjadi menjelang akhir program. Baris `a.Cetak();` digunakan untuk memanggil method `Cetak()` dari objek `a`, yang biasanya berfungsi untuk menampilkan atau mencetak data yang sebelumnya telah disimpan atau diolah dalam objek tersebut. Dengan kata lain, perintah ini menginstruksikan program untuk menampilkan output akhir kepada pengguna. Setelah itu, baris `input.close();` digunakan untuk menutup objek `Scanner` bernama `input` yang sebelumnya dipakai untuk membaca data dari keyboard.

```

27         a.Cetak();
28         input.close();
29     }
30 }

```

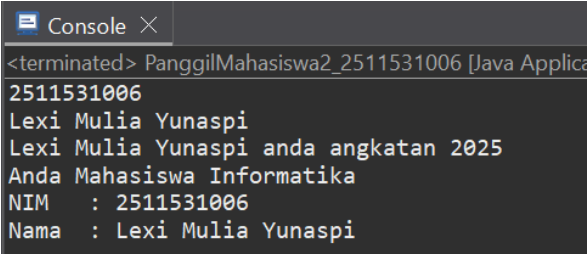
- h. Tampilan semua kode program

```

1 package Pekan7_2511531006;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531006 {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8
9         String x = "2511531006";
10        String y = "Lexi Mulia Yunaspi";
11
12        System.out.println(x);
13        System.out.println(y);
14
15        Mahasiswa_2511531006 a = new Mahasiswa_2511531006();
16        a.setNim(x);
17        a.setNama(y);
18
19        if (x.startsWith("25")) {
20            System.out.println(y + " anda angkatan 2025");
21        }
22
23        if (x.contains("2511531006")) {
24            System.out.println("Anda Mahasiswa Informatika");
25        }
26
27        a.Cetak();
28        input.close();
29    }
30 }

```

i. Output kode program



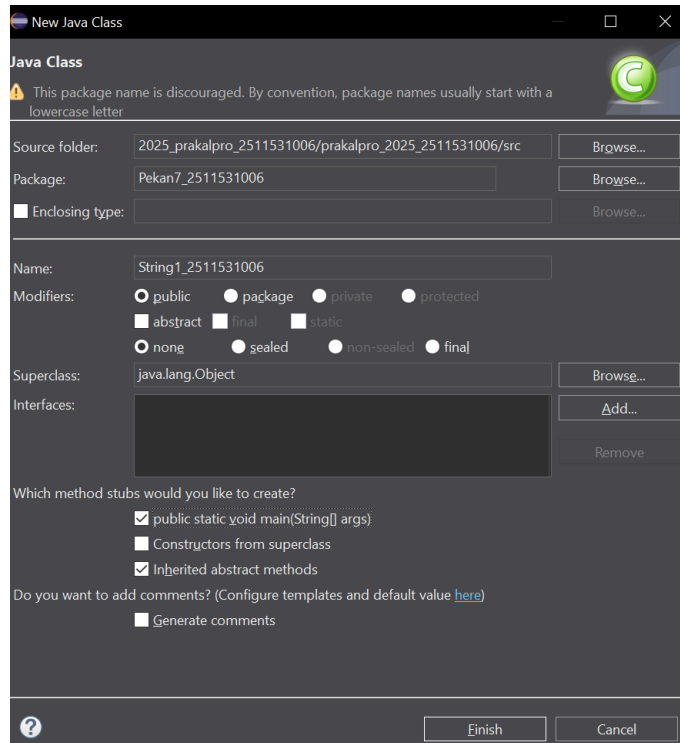
```

Console X
<terminated> PanggilMahasiswa2_2511531006 [Java Applica
2511531006
Lexi Mulia Yunaspi
Lexi Mulia Yunaspi anda angkatan 2025
Anda Mahasiswa Informatika
NIM : 2511531006
Nama : Lexi Mulia Yunaspi

```

5.Kode Program Kelima

- a. pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



b. Jika sudah ditekan finish, maka akan muncul seperti ini

```
1 package Pekan7_2511531006;  
2  
3 public class String1_2511531006 {  
4  
5     public static void main(String[] args) {
```

c. Baris kode tersebut mendefinisikan sebuah variabel bertipe String dengan nama `salam` dan langsung memberinya nilai berupa teks "Assalamualaikum". Artinya, program menyimpan sebuah kalimat salam ke dalam variabel tersebut agar dapat digunakan kembali pada bagian lain dari kode, misalnya untuk ditampilkan kepada pengguna atau dikombinasikan dengan teks lain. Variabel string seperti ini berfungsi sebagai tempat menyimpan data berupa teks, sehingga baris tersebut merupakan inisialisasi awal yang mempersiapkan nilai

salam agar siap dipanggil kapan pun dibutuhkan selama program berjalan.

```
6      String salam = "Assalamualaikum";
```

- d. Potongan kode tersebut berfungsi untuk menampilkan berbagai informasi dan manipulasi terhadap sebuah variabel string bernama `salam`. Pada baris pertama, program mencetak panjang teks menggunakan `salam.length()`, yang menghitung jumlah karakter dalam string tersebut. Dengan ini, pengguna bisa mengetahui seberapa banyak karakter yang terkandung di dalam teks. Kemudian, `salam.toUpperCase()` digunakan untuk mengubah seluruh huruf di dalam string menjadi huruf kapital, sedangkan `salam.toLowerCase()` mengubahnya menjadi huruf kecil. Kedua metode ini sering digunakan untuk standarisasi teks agar mudah dibaca atau dibandingkan. Terakhir, `salam.indexOf("salam")` digunakan untuk mencari posisi awal kemunculan kata "*salam*" di dalam string.

```
6      String salam = "Assalamualaikum";
7      System.out.println("panjang salam adalah: " + salam.length());
8      System.out.println(salam.toUpperCase());
9      System.out.println(salam.toLowerCase());
10     System.out.println(salam.indexOf("salam"));
11
12
13
14
15     }
16
17 }
```

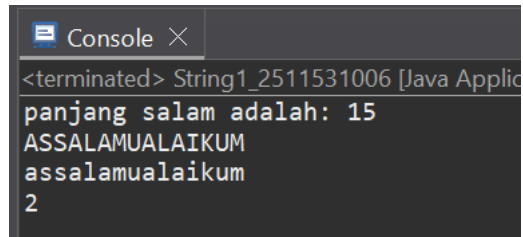
- e. Tampilan semua kode program

```

1 package Pekan7_2511531006;
2
3 public class String1_2511531006 {
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());
9         System.out.println(salam.toLowerCase());
10        System.out.println(salam.indexOf("salam"));
11
12
13
14
15    }
16
17 }

```

f. Output kode program



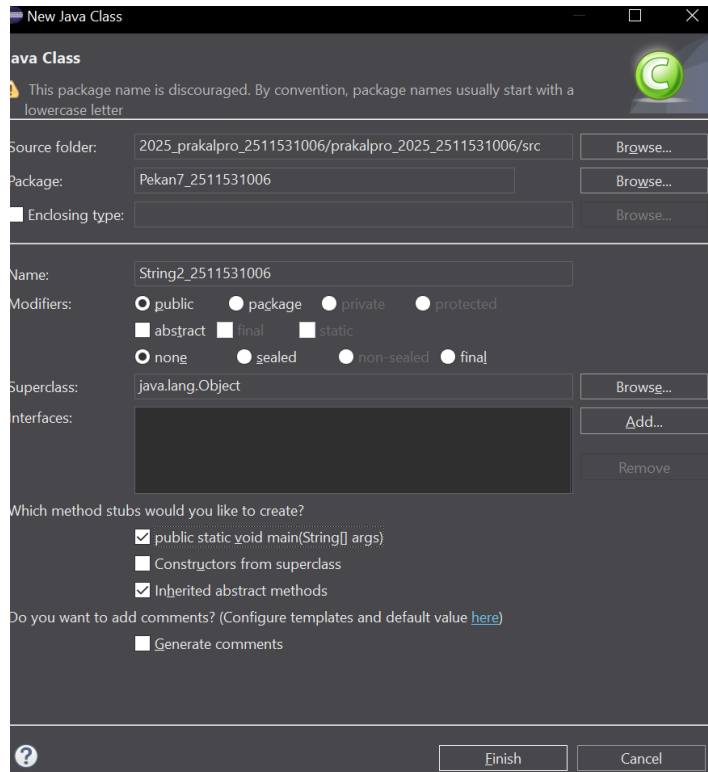
```

Console X
<terminated> String1_2511531006 [Java Applic
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2

```

6.Kode Program Keenam

- a. pilih “New”, lalu pilih class. Buat nama dengan ketentuan nama harus Uppercase pada awal kalimat dan tanpa “space”, lalu centang tanda “public static void main (string[] args)”.



- b. Jika sudah ditekan finish, maka akan muncul seperti ini

```
1 package Pekan7_2511531006;  
2  
3 public class String2_2511531006 {  
4  
5     public static void main(String[] args) {
```

- c. Potongan kode ini mendefinisikan tiga variabel string yang akan digunakan dalam program. Variabel `firstName` menyimpan nama depan, yaitu "Syifa", sedangkan `lastName` menyimpan nama belakang, yaitu "Muhassanah". Kode ini juga memiliki variabel `txt1` yang berisi kalimat "Dosen \"intelektual\" kampus". Tanda backslash (`\`) digunakan untuk melakukan *escaping* sehingga tanda kutip ganda dapat muncul di dalam teks tanpa mengakhiri string. Dengan cara ini, program dapat menyimpan teks yang mengandung kutipan secara aman dan tetap terbaca dengan benar. Kode tersebut pada dasarnya menunjukkan bagaimana Java menangani

penyimpanan nama dan teks yang memiliki karakter khusus di dalamnya.

```
6 String firstName = "Syifa";
7 String lastName = "Muhassanah";
8 String txt1 = "Dosen\\intelektual1\\\" kampus";
```

- d. Potongan kode ini menampilkan cara menggabungkan string dan mencetak hasilnya ke layar. Pada baris pertama, program menampilkan nama lengkap dengan cara menggabungkan firstName, spasi, dan lastName menggunakan operator +. Baris berikutnya melakukan hal yang sama, tetapi menggunakan metode concat(), yaitu firstName.concat(lastName), sehingga hasilnya berupa dua string yang disatukan tanpa spasi kecuali jika ditambahkan secara manual. Terakhir, System.out.println(txt1); mencetak isi variabel txt1 yang berisi kalimat dengan tanda kutip yang sudah di-escape. Secara keseluruhan, bagian kode ini menunjukkan dua cara penggabungan teks serta bagaimana string ditampilkan di konsol.

```
9 System.out.println("Nama Lengkap: "+firstName + " " + lastName);
10 System.out.println("Nama Lengkap: "+firstName.concat(lastName));
11 System.out.println(txt1);
```

- e. Potongan kode ini mendefinisikan tiga variabel bertipe int untuk melakukan operasi aritmatika sederhana. Variabel x diberi nilai 10 dan y diberi nilai 20. Selanjutnya, variabel z digunakan untuk menyimpan hasil penjumlahan dari kedua variabel tersebut, yaitu x + y. Dengan demikian, nilai yang tersimpan dalam z adalah 30. Bagian kode ini menunjukkan cara dasar mendeklarasikan variabel bilangan bulat dan melakukan operasi matematika di dalam program Java.

```
12 int x = 10;
13 int y = 20;
14 int z = x+y;
```

- f. Kode ini menunjukkan perbedaan antara penjumlahan angka dan penggabungan string. Baris pertama mencetak hasil penjumlahan variabel `z`, sehingga program menampilkan nilai 30. Setelah itu, dibuat tiga variabel string: `a` berisi "10", `b` berisi "20", dan `c` yang merupakan hasil penggabungan kedua string tersebut menggunakan operator `+`. Karena keduanya berupa teks, operasi ini tidak menjumlahkan nilai numeriknya, tetapi hanya menggabungkan menjadi "1020". Contoh ini memperlihatkan bahwa operator `+` pada string berfungsi sebagai *concatenation*, bukan sebagai operasi aritmatika.

```
15      System.out.println("x+ y= "+z);
16      String a = "10";
17      String b = "20";
18      String c = a + b;
```

- g. Potongan kode pada gambar tersebut menunjukkan proses operasi penggabungan (*concatenation*) antara variabel String dan variabel lainnya, lalu hasilnya ditampilkan ke layar. Pada baris `System.out.println("String a + string =" +c);`, program mencetak hasil penggabungan dua buah string yang sebelumnya disimpan dalam variabel `c`. Selanjutnya, pada baris `String v = a + y;`, variabel `v` dibentuk dengan cara menggabungkan string `a` dengan nilai integer `y`. Karena ada operasi dengan string, maka integer `y` otomatis diubah menjadi string terlebih dahulu. Terakhir, baris `System.out.println("String a + integer y=" +v);` menampilkan hasil penggabungan tersebut. Secara keseluruhan, bagian ini menunjukkan bahwa ketika sebuah string digabungkan dengan tipe lain seperti integer, Java akan mengonversi tipe tersebut menjadi string sehingga hasil akhirnya tetap berupa teks.

```

19     System.out.println("String a + string =" + c);
20     String v = a + y;
21     System.out.println("String a + integer y=" + v);
22
23 }
24 }

```

h. Tampilan semua kode program

```

1 package Pekan7_2511531006;
2
3 public class String2_2511531006 {
4
5     public static void main(String[] args) {
6         String firstName = "Syifa";
7         String lastName = "Muhassanah";
8         String txt1 = "Dosen\intelektual1\ 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 =" + c);
20        String v = a + y;
21        System.out.println("String a + integer y=" + v);
22
23    }
24 }

```

i. Output kode program

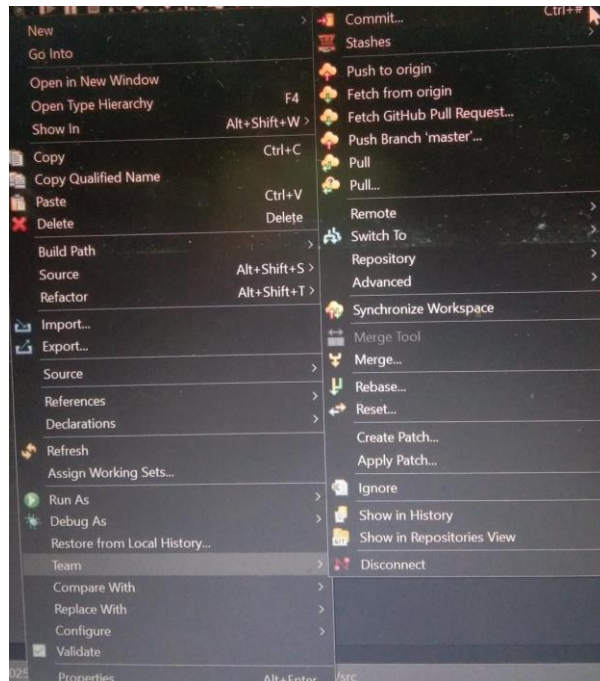
```

Console X
<terminated> String2_2511531006 [Java Application]
Nama Lengkap: Syifa Muhassanah
Nama Lengkap: SyifaMuhassanah
Dosen"intelektual1" kampus
x+ y= 30
String a + string =1020
String a + integer y=1020

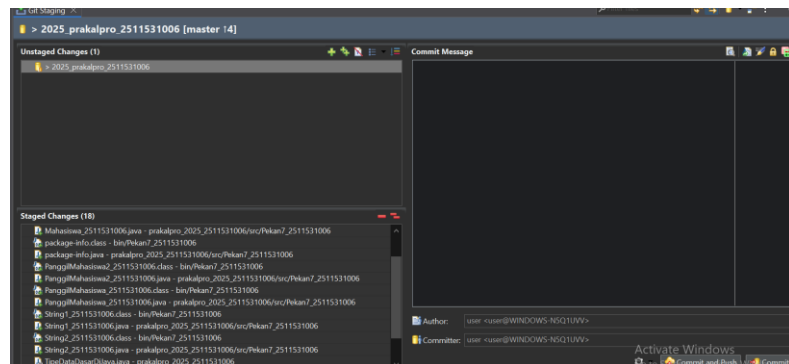
```

7. Menghubungkan Eclipse ke Github

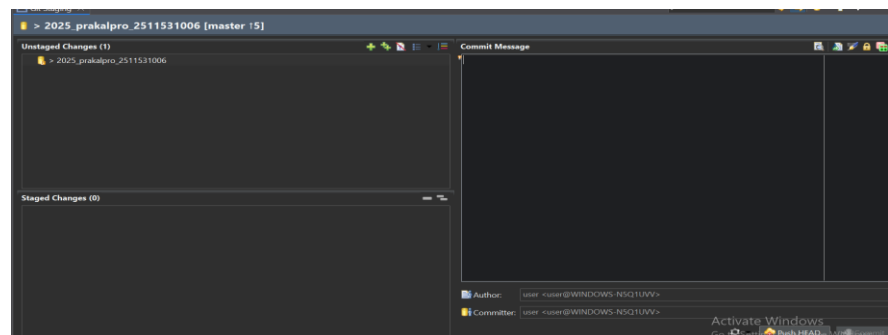
- a. Klik kanan project yang telah dibuat, lalu pilih “Team”, lalu pilih “Commit”.



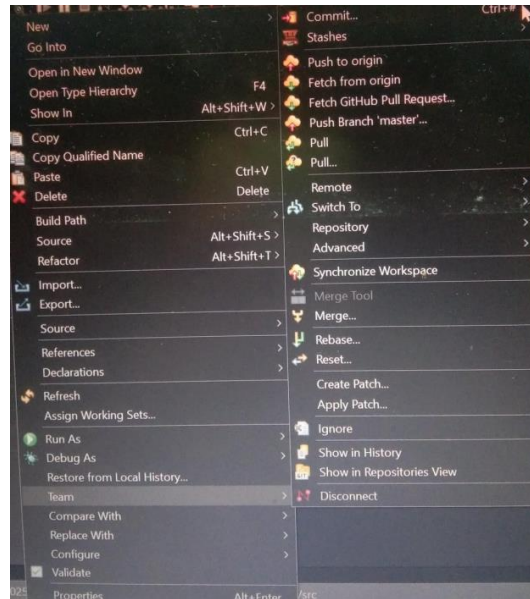
- b. Block semua yang berada di atas dengan , lalu tekan logo tambah dan Setelah ditambah, buat pesan commitnya, lalu klik “Commit”.



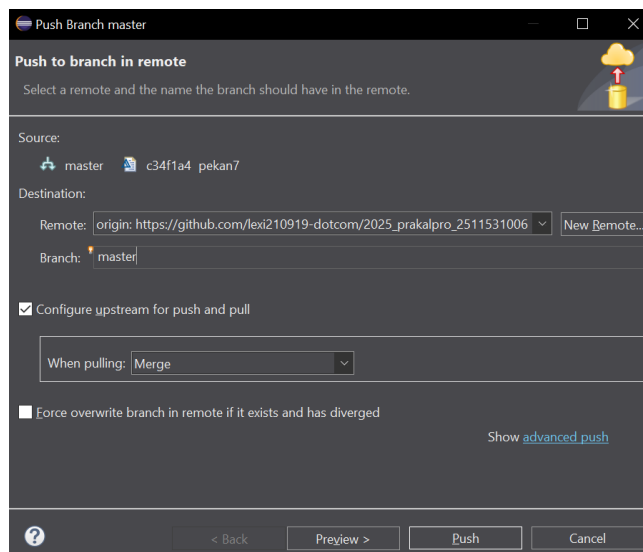
- c. Jika tampilan sudah hilang, maka item berhasil di commit.



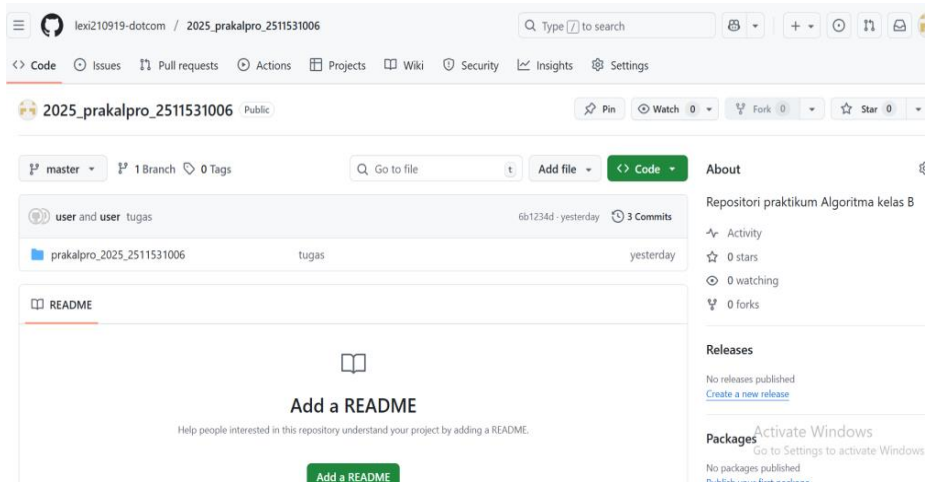
- d. Kembali ke menu awal dengan cara mengklik proyek yang telah dibuat tadi, lalu klik kanan proyek, lalu pilih team, lalu pilih push branch master.



- e. Masukkan link URL yang anda simpan tadi berdasarkan GitHub Repository yang telah dibuat di awal, untuk bagian user diisikan dengan nama usernam profile, dan untuk password diisikan dengan token yang telah dibuat di awal, lalu klik “Preview”.



- f. . Maka file yang berada di Eclipse berhasil disalin di Repository GitHub. Anda bisa mengecek apakah sudah masuk di repository GitHub di akun anda.



BAB III

PENUTUP

3.1.Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa pemrograman Java menyediakan struktur yang jelas dalam membangun program melalui penggunaan method, class, dan object. Melalui latihan yang diberikan, mahasiswa dapat memahami cara membuat dan menggunakan method seperti fungsi pengecekan bilangan prima, serta mempelajari konsep dasar Object-Oriented Programming (OOP) seperti atribut, setter, getter, dan pemanggilan method melalui objek. Selain itu, praktikum ini juga memperkenalkan berbagai operasi pada tipe data String, seperti penggabungan, pencarian karakter, perubahan huruf, serta manipulasi nilai teks lainnya. Pemahaman terhadap ketiga materi ini menjadi dasar penting untuk mengembangkan program yang lebih terstruktur, modular, dan mudah dikelola di tahap pembelajaran selanjutnya.

3.2 Saran

Untuk meningkatkan kualitas pembelajaran, disarankan agar saya lebih sering melakukan latihan mandiri agar terbiasa memahami alur program yang terdiri dari beberapa method dan objek. Selain itu, perlu dilakukan eksplorasi lebih jauh terhadap fitur-fitur OOP lainnya. Agar pemahaman mengenai OOP semakin kuat. Pada materi String, mahasiswa juga dianjurkan mempelajari method-method lain yang tersedia di kelas String sehingga dapat memanfaatkan fungsi yang tepat sesuai kebutuhan pemrograman. Praktikum selanjutnya juga diharapkan menyediakan variasi studi kasus yang lebih kompleks sehingga mahasiswa dapat mengaplikasikan konsep yang telah dipelajari secara lebih menyeluruh.

DAFTAR PUSTAKA

Deitel, P. J., & Deitel, H. M. (2019). *Java: How to Program, Early Objects*. Pearson Education.

Eckel, B. (2006). *Thinking in Java* (4th Edition). Prentice Hall.

Oracle. (2024). *Java Documentation*. Diakses dari: <https://docs.oracle.com/javase/>

Wahana Komputer. (2018). *Pemrograman Java untuk Pemula*. Andi Offset.