

**LAPORAN PRAKTIKUM ALGORITMA DAN
PEMROGRAMAN**
PEKAN 7 STRING PADA JAVA



Disusun Oleh :
DIKA GIOWANDA
NIM 2311533025
Dosen Pengampu :
DR. WAHYUDI, S.T, M.T
Asisten Praktikum :
JOVANTRI IMMANUEL GULO

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
PADANG, NOVEMBER 2025

KATA PENGANTAR

Puji syukur penulis panjatkan atas ke hadirat Tuhan Yang Maha Esa, atas limpahan rahmat, hidayah, dan karunianya , sehingga laporan praktikum ini mengenai *String* bahasa pemrograman pada java dapat diselesaikan tepat waktu. Sholawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, keluarga, sahabat, dan seluruh pengikutnya hingga akhir zaman.

Laporan ini disusun sebagai bentuk dokumentasi dan refleksi dari kegiatan praktikum yang telah dilakukan, dengan tujuan untuk memahami konsep dasar *String* pada java. Dalam praktikum ini, penulis mempelajari berbagai operasi dasar dan lanjutan yang dapat dilakukan terhadap data bertipe *String*, seperti penggabungan, pemotongan, pencarian, konversi, serta penggunaan metode-metode bawaan seperti *length()*, *substring()*, *equals()*, dan lain-lain. Melalui kegiatan ini, penulis tidak hanya memperdalam pemahaman teoretis tentang *String*, tetapi juga meningkatkan kemampuan praktis dalam menulis kode Java yang efisien dan benar.

Penyusunan laporan ini dimaksudkan sebagai panduan pembelajaran bagi pemula yang ingin memahami bahasa pemrograman *String* pada java, sekaligus sebagai bahan refleksi bagi penulis sendiri dalam memperdalam ilmu pemrograman. Semoga materi ini dapat memberikan manfaat dan menjadi langkah awal yang baik dalam perjalanan belajar pemrograman.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Tak lupa, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta teman-teman yang telah memberikan bantuan dan dukungan selama proses penyusunan laporan ini.

Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya bagi rekan-rekan mahasiswa yang sedang mempelajari konsep *String* dalam Java.

Padang, 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	.ii
DAFTAR ISI.....	.iii
BAB 1 PENDAHULUAN.....	.1
1.1 Pengertian Praktikum.....	.1
1.2 Tujuan Praktikum.....	.1
1.3 Persyaratan Praktikum.....	.1
1.4 Tempat dan Waktu Praktikum.....	.2
1.5 Manfaat Praktikum.....	.2
BAB 2 PEMBAHASAN PRAKTIKUM.....	.3
2.1 Pengertian String.....	.3
2.3 Langkah Langkah Pengerjaan Praktikum Pekan 7.....	.4
BAB 3 PENUTUP.....	.23
3.1 Kesimpulan.....	.23
3.2 Saran.....	.23
DAFTAR PUSTAKA.....	.25

BAB 1

PENDAHULUAN

1.1 Pengertian Praktikum

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori,tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini antara lain sebagai berikut :

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik.
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.
6. Mengetahui dan mengaplikasikan if, else, if else, dan switch.

1.3 Persyaratan Praktikum

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

1. Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.

2. Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang *Java Development Kit* (JDK) dan *Integrated Development Environment* (IDE) yang direkomendasikan.
3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

1.4 Waktu dan Tempat Praktikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

1.5 Manfaat Praktikum

Manfaat praktikum bahasa Java meliputi kemampuan memahami dan menerapkan konsep dasar pemrograman seperti tipe data dan kontrol alur, melatih penggunaan kode yang modular dan dapat digunakan kembali, serta membangun pemahaman tentang sintaksis dan struktur *object oriented programming* (OOP) yang lebih mendalam. Selain itu, praktikum ini juga membantu membangun fondasi untuk mengembangkan aplikasi lintas platform yang lebih kompleks di berbagai bidang teknologi.

BAB 2

PEMBAHASAN PRAKTIKUM

2.1 Pengertian Perulangan While

String pada Java adalah objek yang mewakili serangkaian karakter, seperti kata, kalimat, atau teks lainnya. *String* di Java merupakan objek dari kelas *String* yang menyediakan berbagai metode untuk memanipulasi teks, seperti menggabungkan, memotong, atau mengganti karakter. Setiap string bersifat *immutable*, yang berarti setelah dibuat, nilainya tidak dapat diubah; operasi yang memodifikasinya akan menghasilkan objek *String* baru.

Karakteristik utama *String* di Java:

- Objek karakter: *String* adalah objek yang terdiri dari satu atau lebih karakter tunggal, yang direpresentasikan oleh tipe data *char*.
- *Immutable*: Setelah objek *String* dibuat, nilainya tidak dapat diubah. Setiap perubahan akan membuat objek *String* baru.
- Termasuk dalam paket `java.lang`: Kelas *String* sudah tersedia secara otomatis dan tidak perlu di impor.
- Manipulasi: Kelas *String* menyediakan berbagai metode untuk memproses teks, seperti *concat()* untuk menggabungkan, *substring()* untuk memotong, *toUpperCase()* dan *toLowerCase()* untuk mengubah huruf besar/kecil, dan *replace()* untuk mengganti karakter atau *substring*.
- Dapat dibuat dengan berbagai cara: *String* dapat dibuat menggunakan literal (contoh: "Halo, dunia!") atau menggunakan konstruktor `new String("Halo, dunia!")`.

Menggunakan metode *String*:

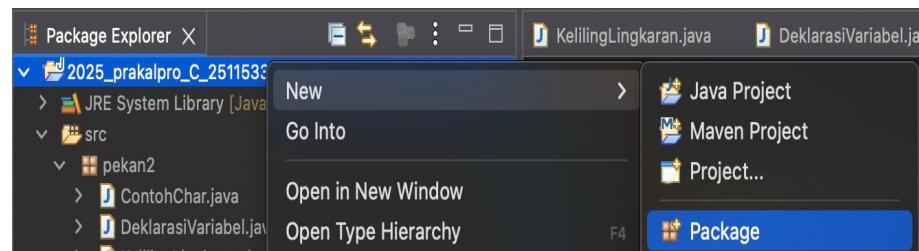
- Objek *String* memiliki banyak metode yang bisa digunakan untuk melakukan operasi, misalnya:
- *length()*: Untuk mendapatkan panjang string (jumlah karakter).

- *toUpperCase()*: Mengubah seluruh string menjadi huruf besar.
- *toLowerCase()*: Mengubah seluruh string menjadi huruf kecil.
- *indexOf()*: Mencari posisi indeks dari karakter pertama yang muncul dalam *String*.

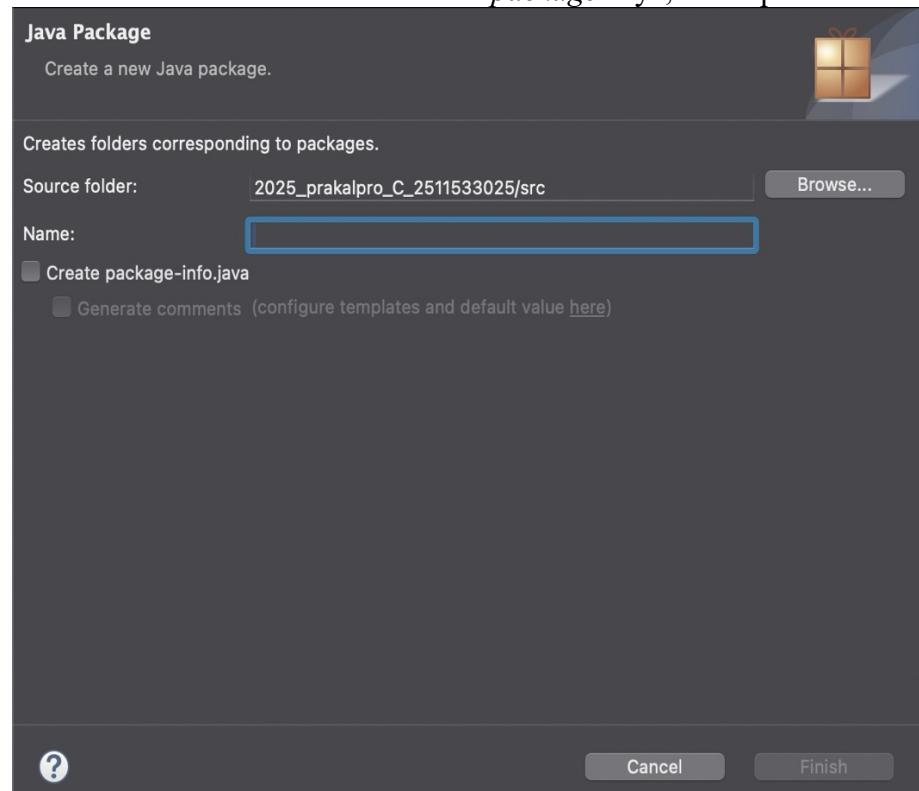
2.2 Langkah Langkah Pengerjaan Praktikum Pekan 7

Sebelum ke langkah Langkah pengerjaan contoh projek, saya akan menjelaskan terlebih dahulu gimana caranya menambahkan projek nya terlebih dahulu :

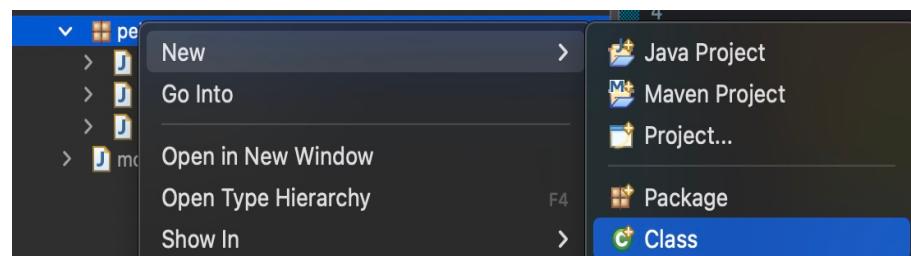
1. Setelah kalian masuk ke eclipse java dan membuat folder file projek java, click kanan pada mouse kalian , kemudian pilih *new* terus pilih *package*.



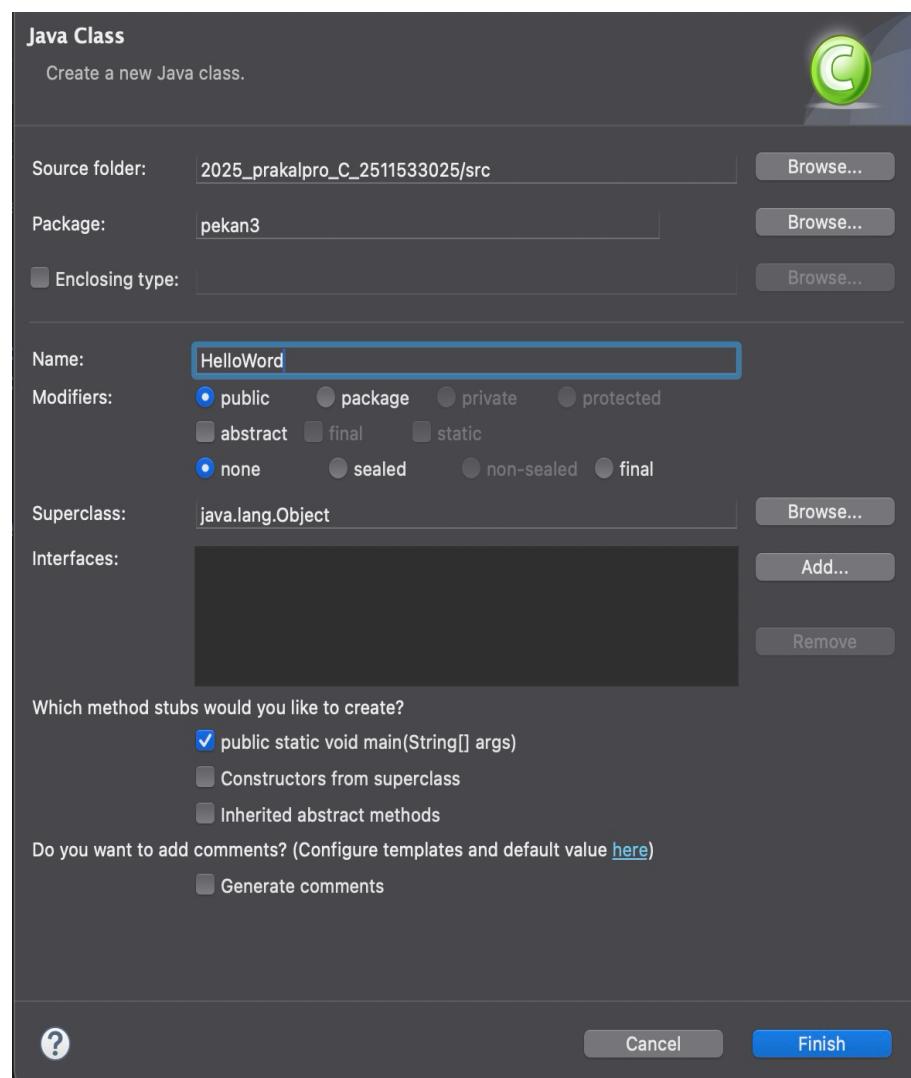
Setelah di click akan muncul halaman java *package*, dimana kita disuruh bikin nama untuk folder *package* nya, lalu pilih finish.



2. Jika folder *packagenya* sudah jadi, click kanan mouse pada folder *package* nya, lalu akan muncul pilihan, kalian pilih *new* kemudian pilih *class*.



Setelah di click akan muncul halaman java *class*, dimana kita disuruh untuk membuat nama *classnya* dibagian yang bertuliskan nama dan jangan lupa untuk centang pada bagian *public static void main*, lalu *click* finis.



Setelah selesai pembuatan folder *class* nya maka baru kita bisa membuat projek yang kita inginkan.

Catatan :

Jika kita ingin membuat projek baru lagi, maka cukup buat folder *class* yang baru.

- A. Baiklah karena proses menambahkan projek sudah dijelaskan, selanjutnya langkah langkah penggerjaan projek :

1. Membuat String 1

1. Tahap awal

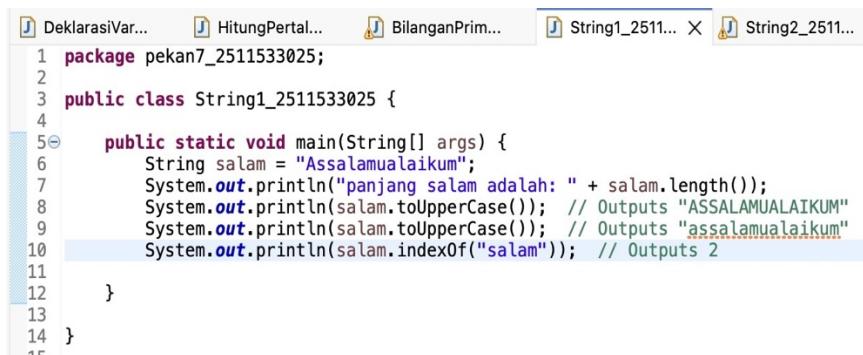
Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai public classnya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas.

2. Tahap inti

- Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri.
- Masukan kode *String* salam = "ASSALAMUALAIKUM" untuk mendeklarasikan variabel *String* bernama salam dengan nilai Assalamualaikaum.
- Masukan kode *System.out.println("panjang salam adalah: + salam.length())* untuk mencetak panjang *String* salam ke *konsol*.
- Masukan kode *System.out.println(salam.toUpperCase()); // Outputs "ASSALAMUALAIKUM"* untuk mencetak nilai dari variabel 'salam' dalam huruf kapital, kode // menunjukkan bahwa jika variabel salam awalnya berisi "Assalamualaikum" (atau variasi huruf kecil lainnya), *output* yang dihasilkan adalah "ASSALAMUALAIKUM"
- Masukan kode *System.out.println(salam.toUpperCase()); // Outputs "assalamualaikum"* Untuk mengubah teks menjadi huruf besar.
- Masukan kode *System.out.println(salam.indexOf("salam")); // Outputs 2* untuk menemukan indeks kemunculan pertama dari *substring* "salam" di dalam string yang direferensikan oleh variabel salam itu sendiri, kode // mengkonfirmasi bahwa output yang dihasilkan oleh eksekusi kode tersebut adalah angka 2.

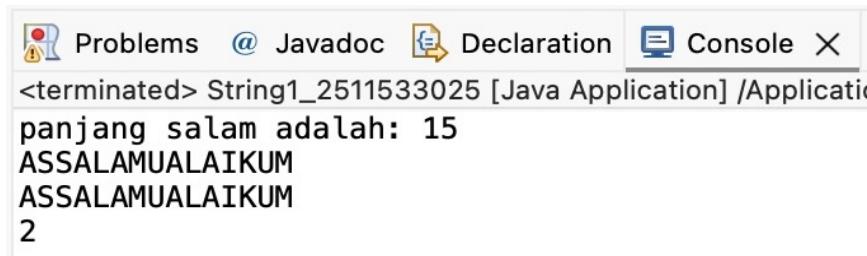
3. Tahap akhir

Masukan kode `System.out.println` untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Jika kita ingin *outputnya* menyamping maka kita gunakan `print`. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar *String 1*:



```
1 package pekan7_2511533025;
2
3 public class String1_2511533025 {
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.toUpperCase()); // Outputs "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); // Outputs 2
11    }
12
13
14 }
```

4. Hasil projek String 1.



```
panjang salam adalah: 15
ASSALAMUALAIKUM
ASSALAMUALAIKUM
2
```

2. Membuat Projek String 2

1. Tahap awal

Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai public *class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa pembuka sebagai pembuka perintah. Tapi pada projek kali ini kita perlu menambahkan *import java.util.(.....)* yang titik itu isinya sesuai kode program yang kita gunakan. Misalnya jika kode program *random* maka kita masukan kode *random* di titik titik tersebut dan jika menggunakan *scanner* maka kita masukan kode *scanner* pada titik titik tersebut.

2. Tahap inti

- Biasanya kode `public static void main(String[] args)` akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri.
- Masukan kode `Scanner input= new Scanner(System.in);` untuk membuat sebuah objek bernama `input` dari kelas `Scanner`.
- Masukan kode `System.out.print("Nama Depan: ")`; untuk menampilkan teks "Nama Depan: " ke *konsol* atau layar tanpa menambahkan baris baru setelahnya.
- Masukan kode `String firstName= input.nextLine();` untuk membaca satu baris teks masukan dari pengguna dan menyimpannya dalam variabel `String` bernama `firstName`.
- Masukan kode `System.out.print("Nama Belakang: ")` Untuk menampilkan teks "Nama Belakang: " ke layar *konsol*.
- Masukan kode `String lastName= input.nextLine();` untuk membaca satu baris teks masukan dari pengguna dan menyimpannya dalam variabel `String` bernama `lastName`.
- Masukan kode `String txt1 = "Dosen\"intelektual\" kampus";` mendeklarasikan sebuah variabel `String` bernama `txt1` dan menetapkan nilai teks "Dosen\"intelektual\" kampus" kepadanya. Fungsi utamanya adalah untuk menyimpan data teks tersebut dalam memori program untuk digunakan nanti.
- Masukan kode `System.out.println("Nama Lengkap: "+firstName + lastName);` untuk mencetak teks ke *konsol* atau *output* standar. Hasil akhirnya adalah mencetak "Nama Lengkap: " diikuti oleh nilai dari variabel nama depan dan nama belakang, dipisahkan oleh spasi.
- Masukan kode `System.out.println("Nama Lengkap: "+firstName.concat(lastName))` untuk mencetak gabungan nama depan dan nama belakang ke *konsol*.
- Masukan kode `System.out.println(txt1);` untuk menampilkan atau mencetak nilai yang tersimpan dalam variabel bernama `txt1` ke standard *output*, yang biasanya adalah *konsol*.
- Masukan kode `int x = 10` untuk membuat sebuah "wadah" bernama `x` yang dapat menyimpan bilangan bulat, dan memasukkan nilai 10 ke dalamnya.
- Masukan kode `int y = 20` untuk membuat sebuah "wadah" bernama `x` yang dapat menyimpan bilangan bulat, dan memasukkan nilai 20 ke dalamnya.
- Masukan kode `int z = x + y` untuk menjumlahkan nilai dari dua variabel, yaitu `x` dan `y`, dan menyimpan hasil penjumlahan tersebut ke dalam variabel baru bernama `z`, yang bertipe data integer (bilangan bulat).

- Masukan kode `System.out.println("x+ y= "+z)` untuk mencetak teks ke *konsol*, yang akan menampilkan "x + y = " diikuti dengan nilai variabel z.
- Masukan kode `String a = "10"` untuk mendeklarasikan sebuah variabel dengan nama a dan tipe data *String* (teks). Variabel ini kemudian diinisialisasi atau diberi nilai awal berupa *String* "10".
- Masukan kode `String b = "20"` untuk mendeklarasikan sebuah variabel dengan nama a dan tipe data *String* (teks). Variabel ini kemudian diinisialisasi atau diberi nilai awal berupa *String* "20".
- Masukan kode `String c = a + b` untuk menggabungkan nilai dari dua variabel, yaitu a dan b, yang diasumsikan bertipe data *String* atau tipe data lain yang dapat dikonversi menjadi *String*. Hasil penggabungan tersebut kemudian disimpan ke dalam variabel baru bernama c yang juga bertipe data *String*.
- Masukan kode `System.out.println("String a + string b =" +c)` untuk mencetak teks dan nilai variabel ke *konsol*.
- Masukan kode `String v = a + y` untuk menggabungkan dua variabel a dan y (yang diasumsikan bertipe data *String*) dan menyimpan hasilnya dalam variabel v yang juga bertipe *String*.
- Masukan kode `System.out.println("String a + integer y =" +v)` untuk mencetak teks dan nilai variabel ke *konsol*.

3. Tahap akhir

Masukan kode `system.out.println` untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Jika kita ingin *outputnya* menyamping maka kita gunakan *print*. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar *String 2* :

```

1 package pekan7_2511533025;
2 import java.util.Scanner;
3 public class String2_2511533025 {
4
5     public static void main(String[] args) {
6         Scanner input= new Scanner(System.in);
7         System.out.print("Nama Depan: ");
8         String firstName= input.nextLine();
9         System.out.print("Nama Belakang: ");
10        String lastName= input.nextLine();
11        String txt1 = "Dosen\"intelektual\" kampus";
12        System.out.println("Nama Lengkap: "+firstName + " " + lastName);
13        System.out.println("Nama Lengkap: "+firstName.concat(lastName));
14        System.out.println(txt1);
15        int x = 10;
16        int y = 20;
17        int z = x + y;
18        System.out.println("x+ y= "+z);
19        String a = "10";
20        String b = "20";
21        String c = a + b;
22        System.out.println("String a + string b =" +c);
23        String v = a + y;
24        System.out.println("String a + integer y =" +v);
25
26    }
27
28 }

```

4. Hasil projek String 2.

Tapi ada perbedaan pada projek sekarang, karena *output* pada projek ini kita yang atur. Pada *output* ini kita perlu memasukan nama sesuai keinginan kita sendiri.

```

Problems @ Javadoc Declaration Console X
<terminated> String2_2511533025 [Java Application] /Applicat
Nama Depan: Dika
Nama Belakang: Giowanda
Nama Lengkap: Dika Giowanda
Nama Lengkap: DikaGiowanda
Dosen"intelektual" kampus
x+ y= 30
String a + string b =1020
String a + integer y =1020

```

3. Membuat Projek Mahasiswa dan Panggil Mahasiswa yang Terhubung tetapi beda Class namun Outputnya hanya Satu

1. Tahap awal

Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas

dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Sedikit tambahan pada projek ini kita harus buat *importnya* sendiri.

2. Tahap inti

- Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat class baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Tapi pada projek kali ini kode ini tidak kita masukan di projek 1, namun kita masukan ke projek ke 2 lebih tepatnya sambungan dari projek 1.
- Masukan kode // variabel global merupakan variabel yang dideklarasikan di luar fungsi atau blok kode apa pun.
- Masukan kode *private int nim* untuk mendeklarasikan sebuah variabel dalam pemrograman berorientasi objek.
- Masukan kode *private String nama, nim2* untuk mendeklarasikan dua variabel, yaitu nama dan nim2, di dalam sebuah kelas.
- Masukan kode // membuat *mutator (setter)* untuk memodifikasi atau mengubah nilai dari anggota data *privat* dalam pemrograman berorientasi objek.
- Masukan kode *public void setNim (int nim) {* untuk mendefinisikan metode yang memungkinkan bagian lain dari program untuk mengubah nilai internal variabel nim dari objek saat ini, tanpa mengembalikan informasi apa pun.
- Masukan kode *this.nim=nim* untuk menetapkan nilai dari parameter atau variabel lokal nim ke variabel *instance* (anggota kelas) *this.nim*.
- Masukan kode *public void setNim2 (String nim2) {* untuk menetapkan (mengatur) nilai dari variabel anggota bernama nim2 di dalam sebuah kelas.
- Masukan kode *this.nim2=nim2* untuk menetapkan nilai dari parameter atau variabel lokal nim2 ke variabel *instance* (anggota kelas) *this.nim2*.
- Masukan kode *public void setNama (String nama) {* untuk menetapkan atau mengubah nilai dari atribut nama pada sebuah objek.
- Masukan kode *this.nama=nama* untuk menetapkan nilai dari parameter atau variabel lokal nama ke variabel *instance* (anggota kelas) *this.nama*.
- Masukan kode // membuat *accessor (getter)* Untuk mengambil (mendapatkan) nilai dari suatu properti atau variabel *privat* dalam sebuah kelas.

- Masukan kode `public int getNim() {` pengambil nilai untuk sebuah variabel yang kemungkinan bernama nim.
- Masukan kode `return nim;` untuk menghentikan eksekusi fungsi tersebut dan mengembalikan nilai yang disimpan dalam variabel bernama nim ke bagian kode yang memanggil fungsi tersebut.
- Masukan kode `public String getNim2() {` pengambil nilai untuk sebuah variabel yang kemungkinan bernama nim2.
- Masukan kode `return nim2;` untuk menghentikan eksekusi fungsi tersebut dan mengembalikan nilai yang disimpan dalam variabel bernama nim2 ke bagian kode yang memanggil fungsi tersebut.
- Masukan kode `public String getNama() {` untuk mengambil atau mendapatkan nilai dari sebuah variabel internal (kemungkinan bernama nama) dalam suatu objek.
- Masukan kode `return nama;` untuk mengembalikan nilai dari variabel nama.
- Masukan kode `//metode lain untuk memberikan penjelasan atau catatan dalam kode sumber yang tidak akan dieksekusi oleh program saat dijalankan.`
- Masukan kode `public void Cetak() {` untuk melakukan operasi pencetakan, seperti menampilkan teks ke *konsol* atau mengirim data ke *printer*.
- Masukan kode `System.out.println("Nim: "+nim);` Untuk menampilkan teks "Nim: " diikuti dengan nilai dari variabel nim ke *konsol*.
- Masukan kode `System.out.println("Nama : "+nama)` untuk mencetak teks "Nama : " diikuti dengan nilai dari variabel nama ke *konsol*, dan kemudian berpindah ke baris baru.
- Masukan kode `public void Cetak2() {` untuk melakukan operasi pencetakan, seperti menampilkan teks ke *konsol* atau mengirim data ke *printer*.
- Masukan kode `System.out.println("Nim: "+nim2)` untuk mencetak nilai dari variabel nim2 ke *konsol* atau layar *output*.
- Masukan kode `System.out.println("Nama : "+nama);` untuk mencetak *output* ke *konsol*.
- Lalu untuk lanjutan kodennya kita buat *class* baru.
- Selanjutnya masukan kode `Mahasiswa_2511533025 a= new Mahasiswa_2511533025()` untuk membuat sebuah objek baru dari kelas `Mahasiswa_2511533025`.
- Masukan kode `a.setNim(23532);` Untuk mengatur nilai atribut 'nim' dari objek 'a'.
- Masukan kode `a.setNama("Rahmat")` untuk mengatur (set) nilai atribut nama dari objek 'a' menjadi "Rahmat".

- Masukan kode `System.out.println(a.getNim());` untuk menampilkan nilai yang dikembalikan oleh metode `getNim()` dari objek `a` ke *konsol* atau layar, diikuti dengan karakter baris baru.
- Masukan kode `System.out.println(a.getNama());` mencetak nilai yang dikembalikan oleh metode `getNama()` dari objek `a` ke *konsol*.
- Masukan kode `a.Cetak();` untuk menjalankan instruksi yang terkandung di dalam metode `Cetak` untuk objek spesifik yang bernama.

3. Tahap akhir

Masukan kode `system.out.println` untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Jika kita ingin *outputnya* menyamping maka kita gunakan `print`. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar Mahasiswa dan Panggil Mahasiswa :

```

1 package pekan7_2511533025;
2
3 public class Mahasiswa_2511533025 {
4
5     // variabel global
6     private int nim;
7     private String nama,nim2;
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    // membuat accessor (getter)
19    public int getNim() {
20        return nim;
21    }
22    public String getNim2() {
23        return nim2;
24    }
25    public String getNama() {
26        return nama;
27    }
28    // metode lain
29    public void Cetak() {
30        System.out.println("Nim : "+nim);
31        System.out.println("Nama : "+nama);
32    }
33    public void Cetak2() {
34        System.out.println("Nim : "+nim2);
35        System.out.println("Nama : "+nama);
36    }
37
38
39
40
41 }

```

Dan ini sambungannya yang beda *class*

```

1 package pekan7_2511533025;
2
3 public class PanggilMahasiswa_2511533025 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533025 a= new Mahasiswa_2511533025();
7         a.setNim(23532);
8         a.setNama("Rahmat");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12
13    }
14
15 }

```

4. Hasil projek gabungan Mahasiswa dan Panggil Mahasiswa.

```
Problems @ Javadoc Declaration Console X
<terminated> PanggilMahasiswa_2511533025 [Java Application]
23532
Rahmat
Nim : 23532
Nama : Rahmat
```

d. Membuat projek Panggil Mahasiswa 2

1. Tahap awal

Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setingginya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai public classnya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Sedikit tambahan pada projek ini kita harus buat *importnya* sendiri.

2. Tahap inti

- Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri.
- Masukan kode *Scanner input= new Scanner(System.in);* untuk membuat sebuah objek *Scanner* baru yang bernama *input*.
- Masukan kode *System.out.print("NIM: ");* untuk mencetak string "NIM: " ke standard *output* (biasanya *konsol* atau layar terminal) tanpa menambahkan karakter baris baru setelah teks tersebut.
- Masukan kode *String x= input.nextLine();* untuk membaca satu baris teks dari input pengguna (misalnya, dari keyboard) dan menyimpannya ke dalam variabel bertipe data *String* bernama *x*.
- Masukan kode *System.out.print("Nama: ");* Untuk mencetak teks ke *konsol*.
- Masukan kode *String y= input.nextLine()* Untuk membaca satu baris input dari pengguna.
- Masukan kode *Mahasiswa_2511533025 a= new Mahasiswa_2511533025();* mendeklarasikan sebuah variabel bernama *a* dengan tipe *Mahasiswa_2511533025* dan menginisialisasinya dengan sebuah objek baru dari kelas tersebut.

- Masukan kode `a.setNim2(x);` untuk memanggil sebuah metode bernama `setNim2` pada objek yang disebut `a`, dan meneruskan nilai dari variabel `x` sebagai argumen.
- Masukan kode `a.setNama(y);` Baris kode tersebut digunakan untuk mengatur nilai properti 'nama' dari objek 'a' dengan nilai yang terkandung dalam variabel 'y'.
- Masukan kode `if(x.startsWith("25")) {` untuk memeriksa apakah nilai dari variabel `x` diawali dengan string "25".
- Masukan kode `System.out.println(y + anda angkatan 2025");` untuk mencetak teks ke *konsol*.
- Masukan kode `if(x.contains("1153")) {` sebagai gerbang jika teks "1153" ada di dalam variabel `x`, maka instruksi selanjutnya dalam blok akan dieksekusi. Jika tidak ada, instruksi tersebut akan dilewati dan memeriksa apakah variabel 'x' berisi *String* "1153".
- Masukan kode `System.out.println("Anda Mahasiswa Informatika");` untuk mencetak teks ke *konsol*.
- Masukan kode `a.Cetak2();` untuk memanggil suatu fungsi atau metode.
- Masukan kode `input.close();` untuk menutup aliran masukan atau sumber daya terkait.

3. Tahap akhir

Masukan kode `system.out.println` untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Jika kita ingin *outputnya* menyamping maka kita gunakan *print*. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar projek Panggil Mahasiswa 2 :

```

1 package pekan7_2511533025;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511533025 {
4
5     public static void main(String[] args) {
6         Scanner input= new Scanner(System.in);
7         System.out.print("NIM: ");
8         String x= input.nextLine();
9         System.out.print("Nama: ");
10        String y= input.nextLine();
11        Mahasiswa_2511533025 a= new Mahasiswa_2511533025();
12        a.setNim2(x);
13        a.setNama(y);
14        if(x.startsWith("25")) {
15            System.out.println(y + " anda angkatan 2025");
16        }
17        if(x.contains("1153")) {
18            System.out.println("Anda Mahasiswa Informatika");
19        }
20        a.Cetak2();
21        input.close();
22
23    }
24
25}
26

```

4. Hasil projek Panggil Mahasiswa 2.

Tapi ada perbedaan pada projek sekarang, karena *output* pada projek ini kita yang atur. Pada *output* ini kita perlu memasukan nama dan nim sesuai keinginan kita sendiri.

```

Problems @ Javadoc Declaration Console <terminated> PanggilMahasiswa2_2511533025 [Java Application]
NIM: 2511533025
Nama: Dika Giowanda
Dika Giowanda anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511533025
Nama : Dika Giowanda

```

e. Membuat projek Bilangan Prima

1. Tahap awal

Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas

dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Sedikit tambahan pada projek ini kita harus buat *importnya* sendiri.

2. Tahap inti

- Biasanya kode `public static void main(String[] args)` akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Tetapi pada projek ini, kode tersebut terletak di tengah-tengah. Terus apa yang pertama? nah lihat langkah selanjutnya.
- Masukan kode `public static boolean isPrime(int n) {` untuk memeriksa apakah suatu bilangan bulat merupakan bilangan prima. Fungsi ini dirancang untuk mengembalikan nilai *boolean*, yaitu *true* jika bilangan tersebut prima dan *false* jika sebaliknya.
- Masukan kode `int factors = 0` Untuk mendeklarasikan variabel bernama *factors* dengan tipe data *integer* dan menginisialisasi nilainya menjadi 0.
- Masukan kode `for (int i = 1; i <= n; i++) {` untuk menjalankan blok kode yang ada di dalam tanda kurung kurawal secara berulang kali selama kondisi tertentu terpenuhi dan untuk mengulangi eksekusi blok kode sebanyak n kali, dimulai dari i bernilai 1 hingga i mencapai n.
- Masukan kode `if (n % i == 0) {` merupakan kondisi percabangan untuk memeriksa sisa bagi.
- Masukan kode `factors++;` untuk menambahkan 1 ke variabel '*factors*'.
- Masukan kode `return (factors == 2);` untuk mengembalikan nilai *boolean* (benar atau salah) yang menunjukkan apakah jumlah faktor sama dengan 2.
- Masukan kode `public static void main(String[] args) {` sebagai titik masuk (*entry point*) untuk eksekusi program Java.
- Masukan kode `Scanner input = new Scanner(System.in);` untuk membuat objek masukan (*input*) dari pengguna melalui *konsol* atau keyboard.
- Masukan kode `System.out.print("input nilai n = ");` untuk mencetak teks ke *konsol* tanpa menambahkan baris baru setelahnya.
- Masukan kode `int a= input.nextInt();` meminta pengguna memasukkan sebuah angka bulat dan menyimpannya dalam variabel a.
- Masukan kode `if (isPrime(a)) {` untuk memeriksa apakah variabel a berisi bilangan prima.
- Masukan kode `System.out.println(a+" bilangan prima")` untuk menginformasikan pengguna bahwa nilai yang

disimpan dalam variabel a adalah bilangan prima dengan menampilkannya secara jelas di *konsol*.

- Masukan kode } *else* { untuk menentukan blok kode alternatif yang akan dieksekusi jika kondisi dalam pernyataan *if* sebelumnya bernilai salah (*false*).
- Masukan kode *System.out.println(a+" bukan bilangan prima")*; untuk mencetak nilai variabel 'a' diikuti dengan teks " bukan bilangan prima" ke *konsol*.

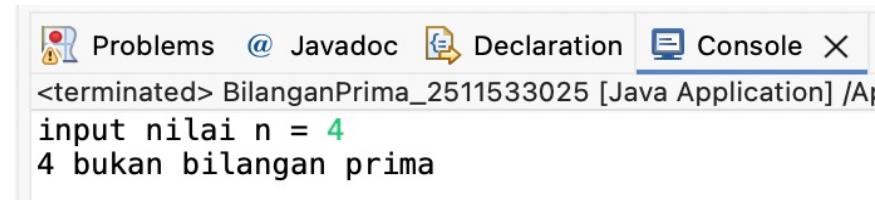
3. Tahap akhir

Masukan kode *system.out.println* untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Jika kita ingin *outputnya* menyamping maka kita gunakan *print*. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar projek Bilangan Prima :

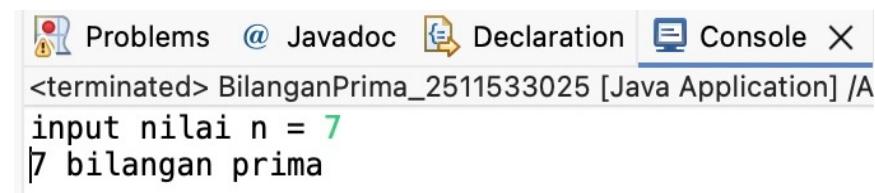
```
1 package pekan7_2511533025;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511533025 {
6
7     public static boolean isPrime(int n) {
8         int factors = 0;
9         for (int i = 1; i <= n; i++) {
10            if (n % i == 0) {
11                factors++;
12            }
13        }
14        return (factors == 2);
15    }
16
17    public static void main(String[] args) {
18        Scanner input = new Scanner(System.in);
19        System.out.print("input nilai n = ");
20        int a= input.nextInt();
21        if (isPrime(a)) {
22            System.out.println(a+" bilangan prima");
23        } else {
24            System.out.println(a+" bukan bilangan prima");
25        }
26    }
27
28 }
```

5. Hasil projek Bilangan Prima.

Tapi ada perbedaan pada projek sekarang, karena *output* pada projek ini kita yang atur, apabila kita memasukan angka bilangan genap maka *output* yang keluar bukan bilangan prima dan apabila kita memasukan angka bilangan ganjil yang hanya bisa dibagi dengan 1 atau bilangan dirinya sendiri maka *output* yang keluar adalah bilangan prima.



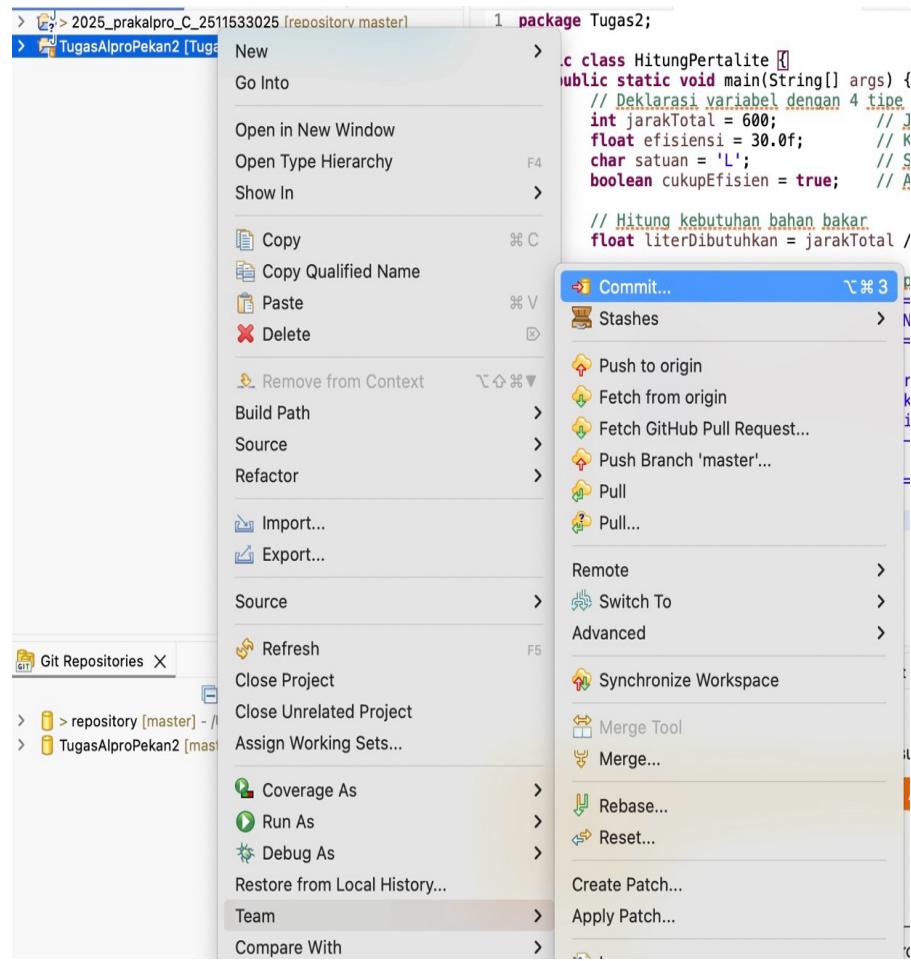
```
Problems @ Javadoc Declaration Console X
<terminated> BilanganPrima_2511533025 [Java Application] /A
input nilai n = 4
4 bukan bilangan prima
```



```
Problems @ Javadoc Declaration Console X
<terminated> BilanganPrima_2511533025 [Java Application] /A
input nilai n = 7
7 bilangan prima
```

B. Langkah Penyimpanan

1. Sebelum kita masuk ke langkah, kita harus buat akun github dulu. Jika sudah buat akun github, baru bisa kita simpan di githubnya. Selanjutnya ikutin langkah ini.Tekan kanan mouse pada folder projek, setelah itu pilih team, terus ke *commit*.



2. Lalu tekan tombol plus 2/double plus warna hijau.



3. Lalu masukan pesan pada kolom *commit message*.

Commit Message

i Unborn branch: this commit will create the branch 'master'.

4. Setelah itu tekan *commit* and *pus*, lalu ikutin arahan umtuk memasukan nama dan pw akun git hub kalian, maka projek kalian sudah tersimpan di akun github kalian.



5. Projek yang sudah disimpan di akun projek.

DikaSiNpc/2025_prakalpro_C_251153
3025

6. Gambar projek apabila sudah masuk di akun github.

The screenshot shows a GitHub repository page. On the left, there's a sidebar with navigation links like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below that is a 'Files' section showing the directory structure of the 'master' branch. The main area displays a table of commits:

Name	Last commit message	Last commit date
...		
pekan1	praktikum 1 kelas C	2 months ago
pekan2	praktikum pekan2	2 months ago
pekan3	pekan3 kelasC	2 months ago
pekan4	prakAlproPekan4	2 months ago
pekan5_2511533025	Praktikum pekan5_2511533025	2 weeks ago
pekan6_2511533025	PraktikumPekan6_2511533025	last week
pekan7_2511533025	PrakAlproPekan7_2511533025	3 days ago

Sedikit tambahan jika ingin cek hasil program nya tekan tombol run warna hijau.



Tapi pada program yang sekarang berbeda dari sebelumnya karena pada program ini ada yang inputnya kita masukan angka atau huruf sesuai keinginan kita pada kolom *console*, tapi ingat harus sesusi perintah kode yang kita masukan. Dan satu lagi pada projek pekan 7 ini terdapat kode program yang membutuhkan 2 *class* agar program dapat dijalankan dan keluar *outputnya*.

BAB 3

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa *String* merupakan salah satu tipe data dasar yang sangat penting dalam pemrograman Java. Meskipun bukan tipe primitif, *String* dalam Java bersifat *immutable*, artinya isi dari sebuah objek *String* tidak dapat diubah setelah dibuat, setiap operasi modifikasi akan menghasilkan *String* baru. Praktikum ini memperkenalkan berbagai metode bawaan yang tersedia dalam kelas *String*, seperti *length()*, *charAt()*, *substring()*, *concat()*, *equals()*, *equalsIgnoreCase()*, *toUpperCase()*, *toLowerCase()*, dan *indexOf()*. Dengan menerapkannya dalam berbagai contoh kode, penulis memahami cara kerja masing-masing metode serta pentingnya memilih metode yang tepat sesuai kebutuhan. Selain itu, penulis juga belajar perbedaan mendasar antara penggunaan operator metode *equals()* dalam membandingkan dua *String*. Praktikum ini tidak hanya meningkatkan keterampilan menulis kode, tetapi juga membentuk pola pikir yang lebih teliti dalam menangani data teks, yang sering menjadi bagian utama dalam aplikasi nyata. Dengan demikian, penguasaan terhadap *String* dan metode-metodenya merupakan fondasi penting bagi pengembangan aplikasi Java yang efektif dan bebas kesalahan. Dan saya sangat senang bisa mengikuti praktikum ini karena menambah wawasan dan ilmu pengetahuan tentang dunia pemrograman.

3.2 Saran

Untuk meningkatkan kualitas praktikum, disarankan untuk memperdalam pemahaman tentang sifat *immutable String* serta menambahkan studi kasus nyata seperti validasi input atau parsing data agar konsep lebih relevan dengan aplikasi dunia nyata. Selain itu, latihan yang sengaja memicu kesalahan umum-seperti penggunaan `==` dapat diperkuat untuk memvisualisasikan perubahan nilai secara

langsung, sehingga mahasiswa tidak hanya menguasai sintaks, tetapi juga memahami dampak teknis dan efisiensi dari setiap penerapannya.

DAFTAR PUSTAKA

- Sumber daring (website):

[1] Study.com, "Java String Data Type," 2025. [Daring].

Tersedia pada: <https://study.com/academy/lesson/java-string-data-type.html#>. [Diakses: 12-Nov-2025].

[2] Coding Studio, "String Adalah: Pengertian, Jenis, Fungsi & Contoh Penggunaan," 2023. [Daring]. Tersedia pada: <https://codingstudio.id/blog/string-adalah/>. [Diakses: 12-Nov-2025].