

LAPORAN PRAKTIKUM ALGORITMA PEMOGRAMAN
PRINTLN PADA PEMROGRAMAN JAVA DAN
GITHUB REPOSITORY



Oleh:

Arya Pratama Hendri

NIM 2411533008

DOSEN PENGAMPU:

DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

A. Pendahuluan

Praktikum ini dilakukan untuk membuat program `println` sederhana dengan menggunakan bahasa pemrograman Java. Menginstal Java Development Kit (JDK) pada Eclipse IDE yang digunakan untuk membuat program Java. Membuat Github Repository sebagai tempat penyimpanan berbasis cloud.

1. Pemrograman Java

Program adalah himpunan atau kelompok instruksi tertulis pada komputer yang diciptakan oleh programmer yang dapat dieksekusi dari sebuah software yang akan dijalankan oleh komputer. Pemrograman berarti menciptakan program komputer. Eksekusi program adalah tindakan melaksanakan instruksi yang terkandung dalam suatu program. Pengertian Java adalah sebuah bahasa pemrograman umum tingkat yang berorientasi pada objek (*object-oriented programming*) yang dapat digunakan di berbagai platform (multi-platform) maupun sistem operasi. Programnya tersusun dari kelas-kelas atau bagian-bagian tertentu. Kelas ini sendiri terdiri atas metode-metode yang bertugas melakukan pekerjaan kemudian mengembalikan informasi setelah pekerjaan tersebut selesai.

2. Pengertian GitHub

GitHub adalah layanan hosting berbasis *cloud* yang memungkinkan pengembang untuk menyimpan proyek mereka menggunakan sistem kontrol versi Git, yang memungkinkan pengguna untuk melacak perubahan kode. Dengan GitHub, dapat mengelola repositori kode, melakukan review kode, dan berkolaborasi dengan tim atau komunitas secara efektif. Dengan GitHub, developer dapat membuat repository untuk proyek mereka, melakukan *commit* untuk menyimpan perubahan, dan menggunakan fitur *branching* untuk mengembangkan fitur baru tanpa mengganggu kode utama. Mendukung berbagai bahasa pemrograman dan dapat digunakan dalam pengembangan perangkat lunak, seperti proyek *open source*. Memiliki fitur seperti pelacakan bug, permintaan fitur, manajemen tugas dan wiki untuk setiap proyek.

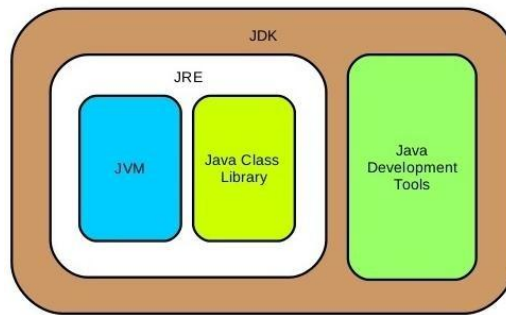
3. Eclipse IDE dan JDK

Eclipse adalah sebuah IDE (Integrated Development Environment) yang dikembangkan oleh IBM dan Eclipse Foundation untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*) yang bersifat *open source*. Kemampuannya yang membuatnya populer adalah untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

JDK (*Java Development Kit*) adalah sebuah paket aplikasi yang berisi JVM (*Java Virtual Machine*) + JRE (*Java Runtime Environment*) + berbagai aplikasi untuk proses pembuatan kode program Java.

JRE (*Java Runtime Environment*) adalah sebuah paket aplikasi yang berisi JVM (*Java Virtual Machine*) serta beberapa library kode program tambahan yang diperlukan untuk menjalankan aplikasi Java. Untuk bisa menjalankan sebuah aplikasi Java di sebuah komputer, harus terinstall aplikasi JRE.

Java Development Kit (JDK)



B. Tujuan

Tujuan dilakukannya praktikum ini adalah sebagai berikut:

1. Memahami cara menginstal Eclipse IDE dan JDK sebagai software dalam membuat program Java.
2. Memahami cara membuat program java di Eclipse dan menjalankannya
3. Membuat akun GitHub dan repository GitHub
4. Memahami cara mengoneksikan Eclipse ke repository GitHub

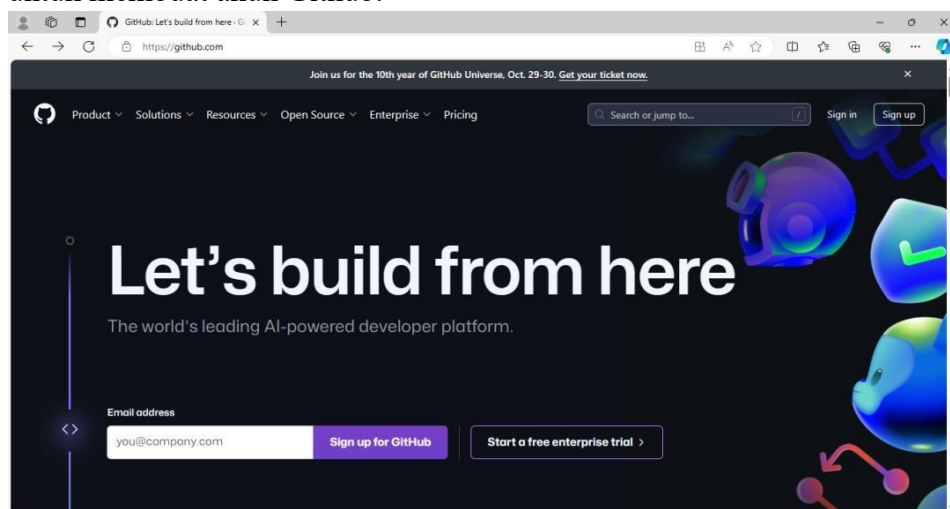
C. Langkah Kerja Praktikum

a. Alat dan Bahan

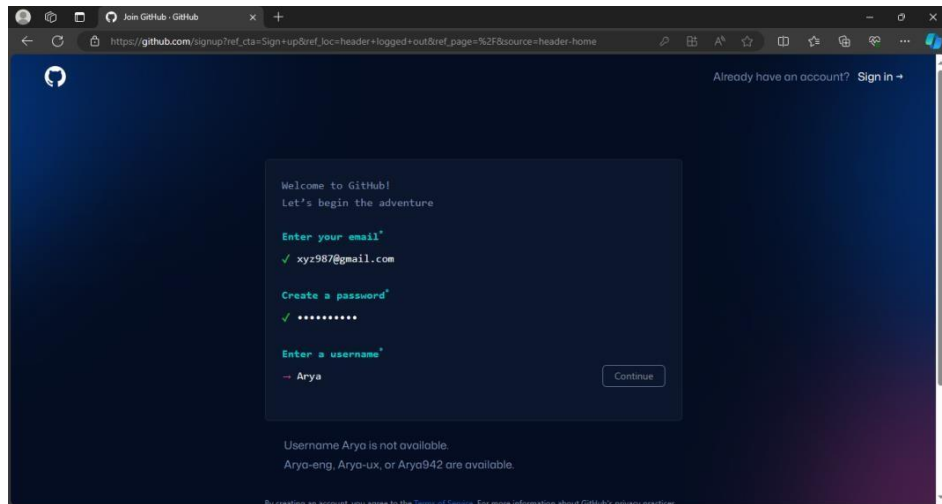
1. Perangkat computer atau laptop
2. IDE (Integrated Development Environment) Eclipse
3. Java Development Kit (JDK)
4. Jaringan Internet/wifi

b. Membuat akun GitHub dan membuat Repository

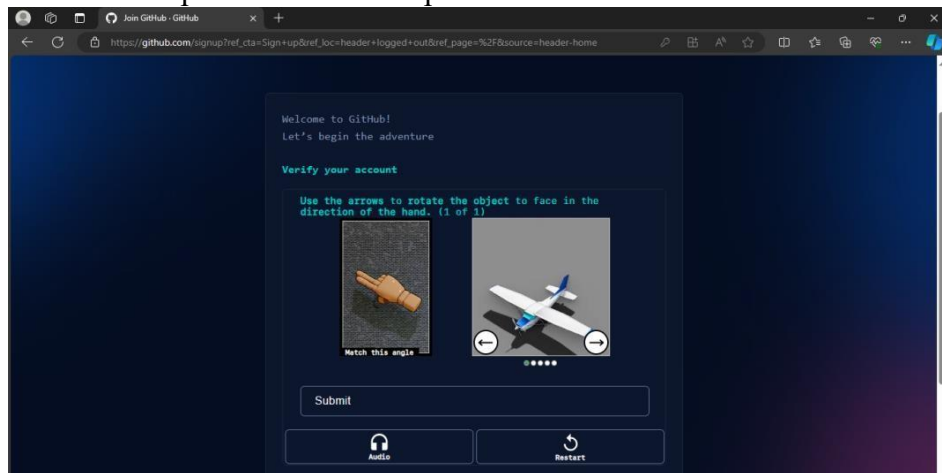
1. Di browser ketik „Github.com“. Setelah laman web terbuka, pilih “Sign Up” untuk membuat akun Github.



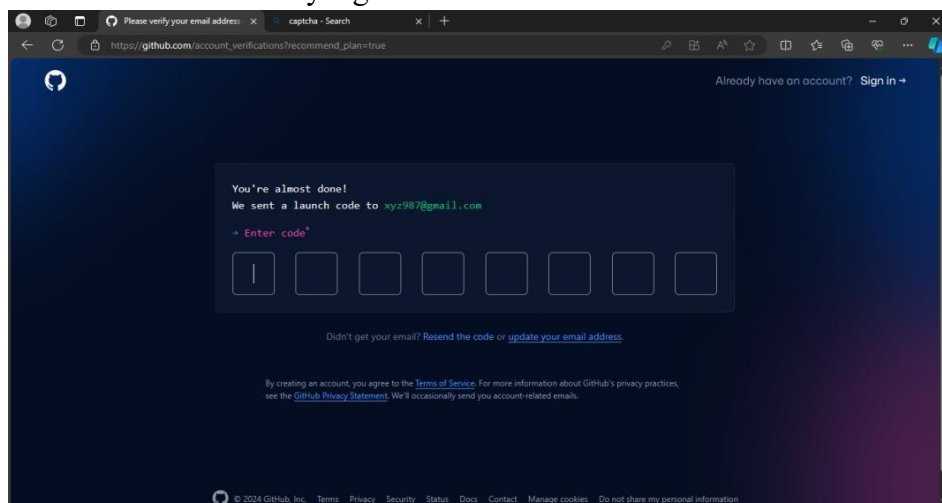
2. Masukan Akun Email, password dan nama untuk profil github.



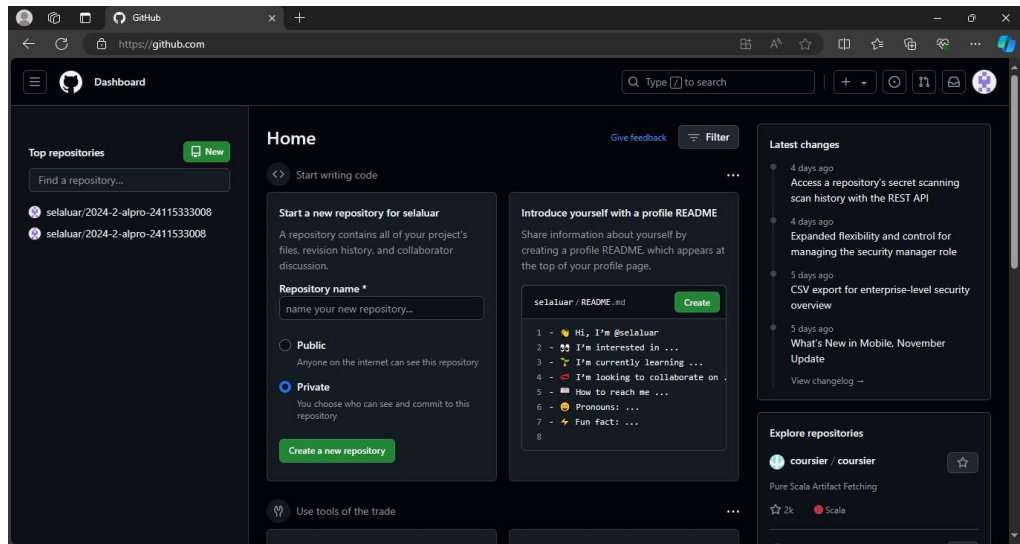
3. Selesaikan captcha untuk mendapatkan kode verifikasi.



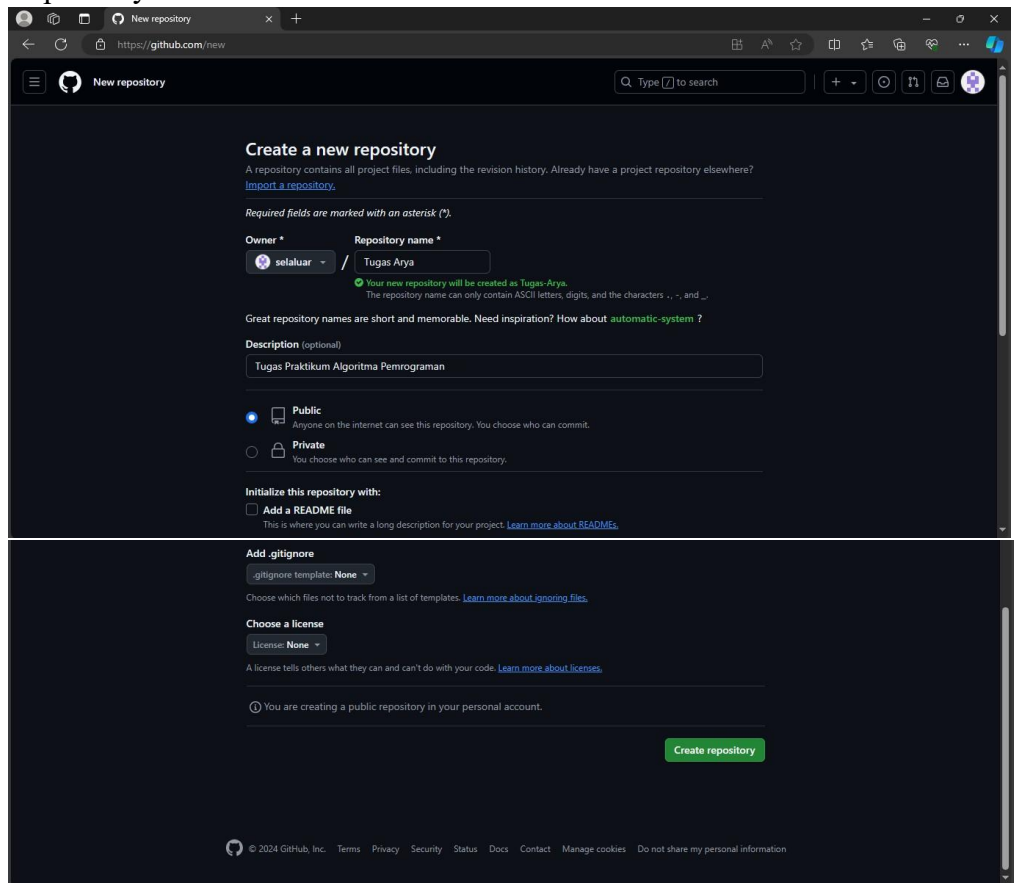
4. Buka email yang sebelumnya digunakan untuk mendaftar akun github untuk melihat kode verifikasi yang dikirimkan. Lalu masukan kode verifikasi tersebut.



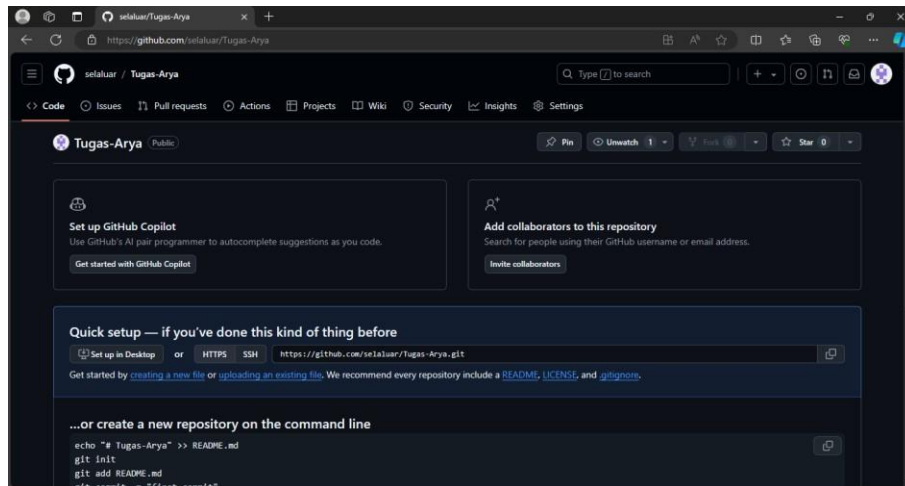
5. Setelah itu, tampilan lawan web akan menjadi seperti berikut. Buat repository dengan menekan tombol "Create Repository" atau "new".



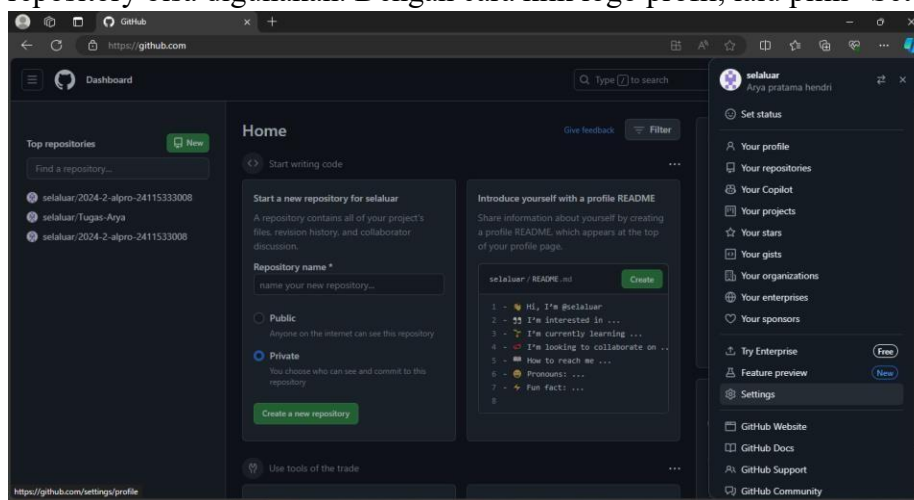
6. Buat nama repository, deskripsi repository, dan pilih apakah proyek dapat dilihat secara publik atau hanya undangan (Private). Kemudian klik “Create Repository” di bawah



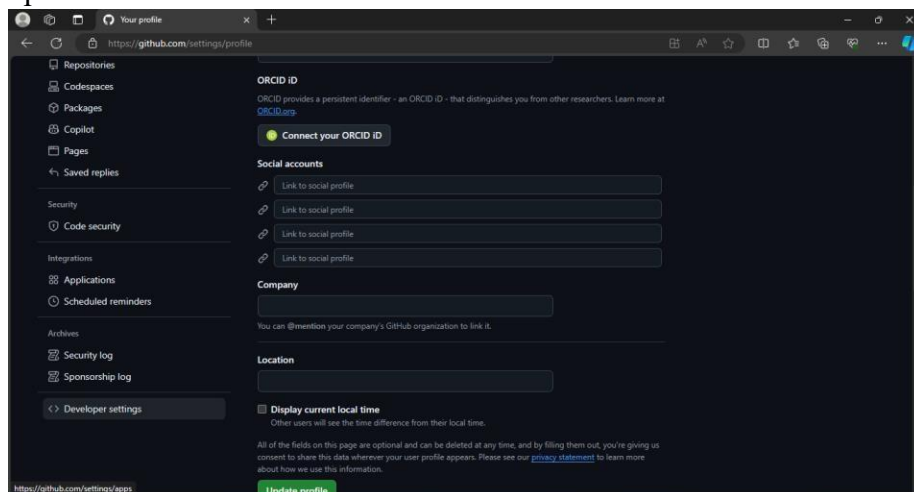
7. Setelah dibuat, tampilan akan seperti berikut. Copy dan simpan link di “Quick setup” yang akan digunakan nanti sebagai link repository.



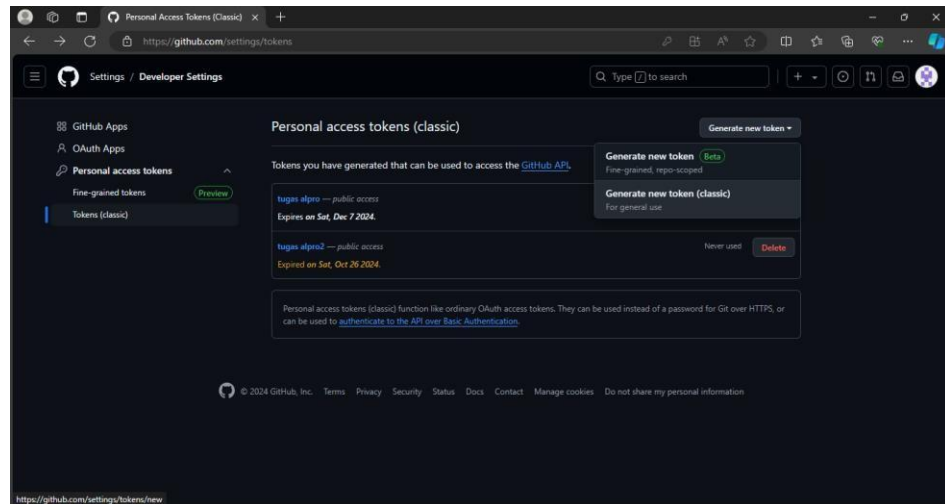
8. Selanjutnya membuat token untuk menghubungkan Eclipse ke GitHub agar repository bisa digunakan. Dengan cara klik logo profil, lalu pilih “Setting”



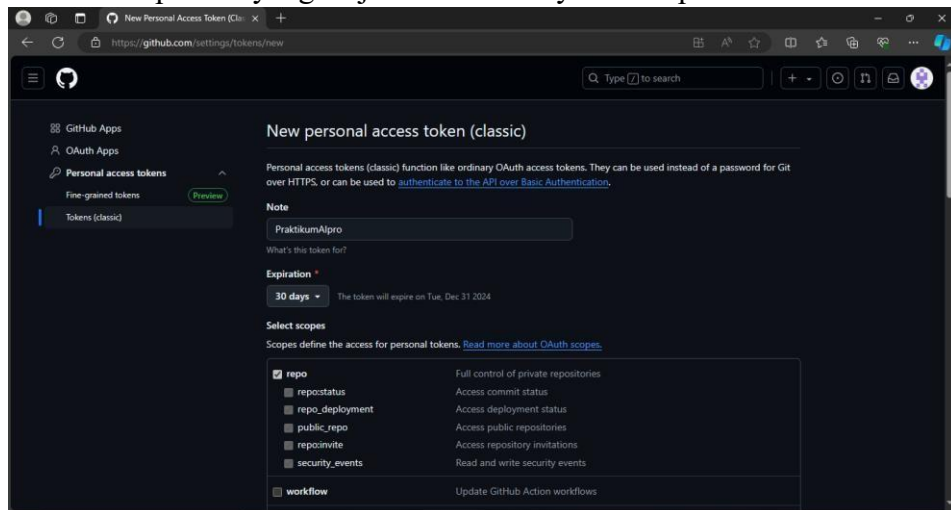
9. Setelah itu, scroll hingga paling bawah sampai menemukan menu “Developer option” lalu klik.



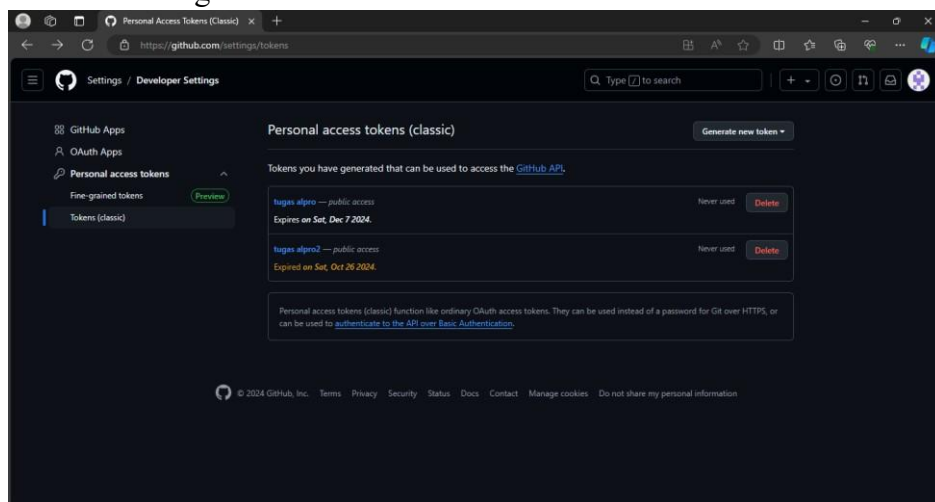
10. Setelah itu, pilih menu “Personal Access Token”, lalu ke “Token (Classic)”, lalu “Generate new token”.



11. Setelah itu, isi note lalu tulis fungsi tokennya. Expiration date menyesuaikan. Lalu untuk pilihan yang wajib di checklist yaitu “Repo”.

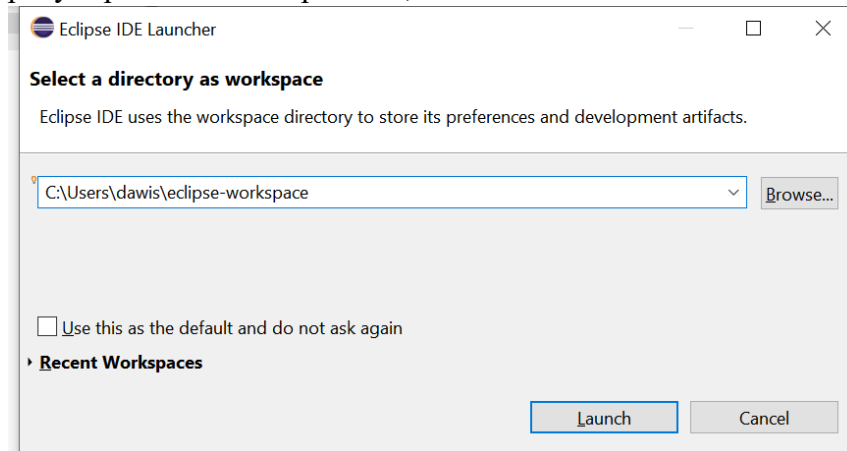


12. Kemudian tampilan akan seperti berikut. Simpan copy dan simpan tokennya karena akan digunakan nanti.

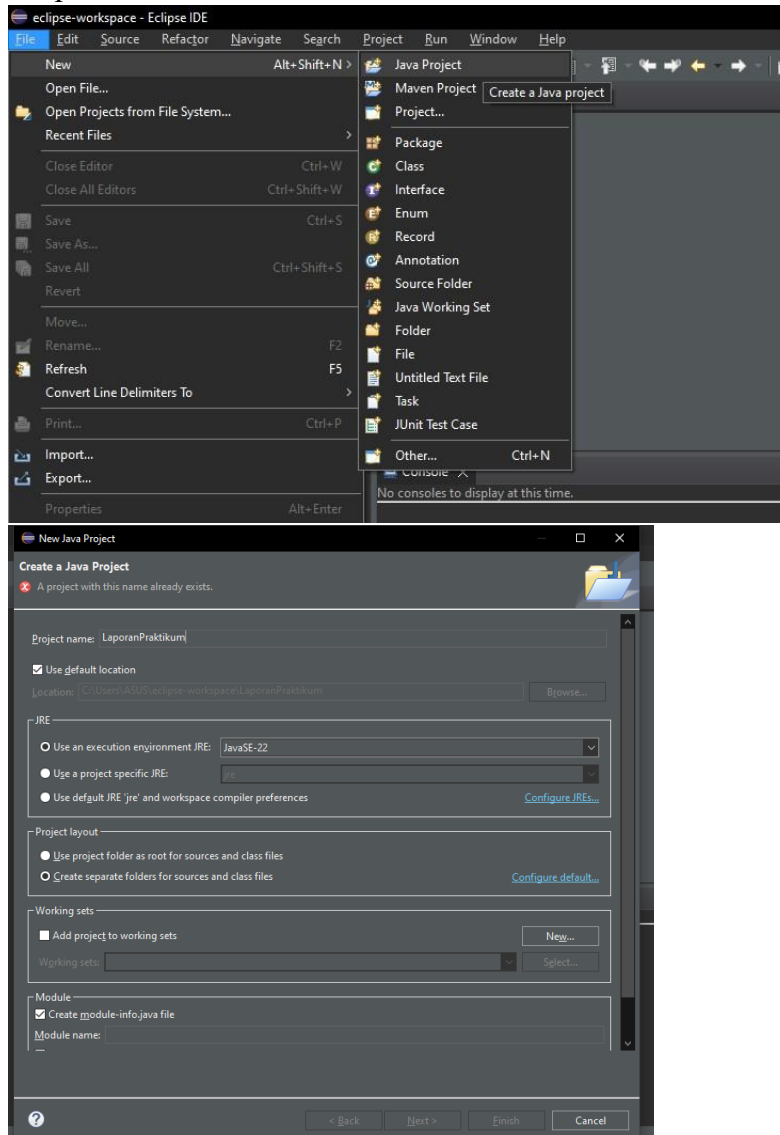


c. Membuat Repository Lokal di Eclipse

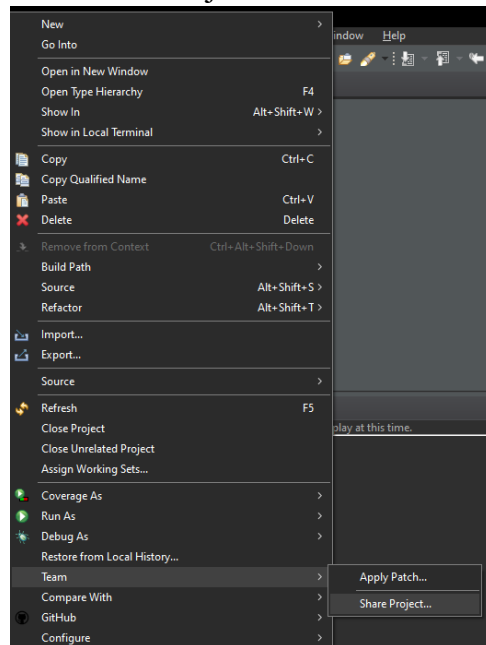
1. Buka Eclipse IDE, buat workspace terlebih dahulu untuk menentukan letak penyimpanan data Eclipse IDE, lalu tekan Launch.



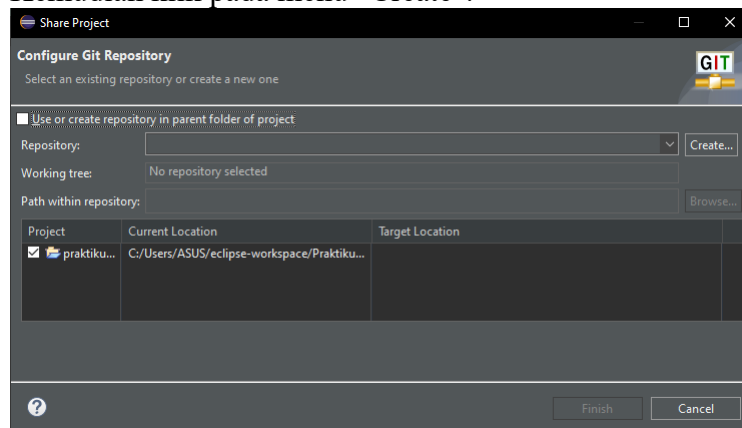
2. Selanjutnya buat Java project baru, caranya dengan klik menu “File” di pojok kiri atas, lalu klik Java project dan tulis nama project dengan “LaporanPraktikum”.



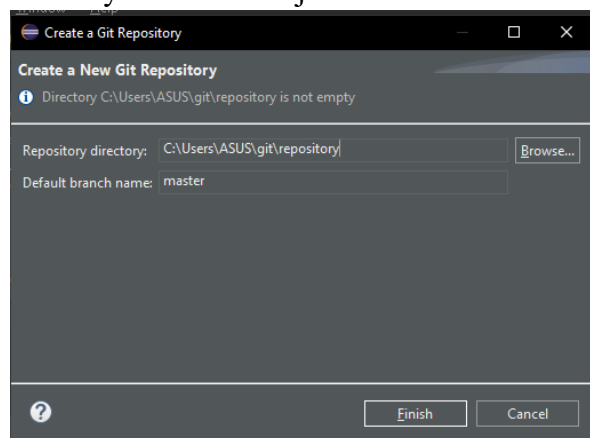
3. Lalu, klik kanan pada file project Java yang telah dibuat, pilih “Team”, lalu klik “Share Project”.



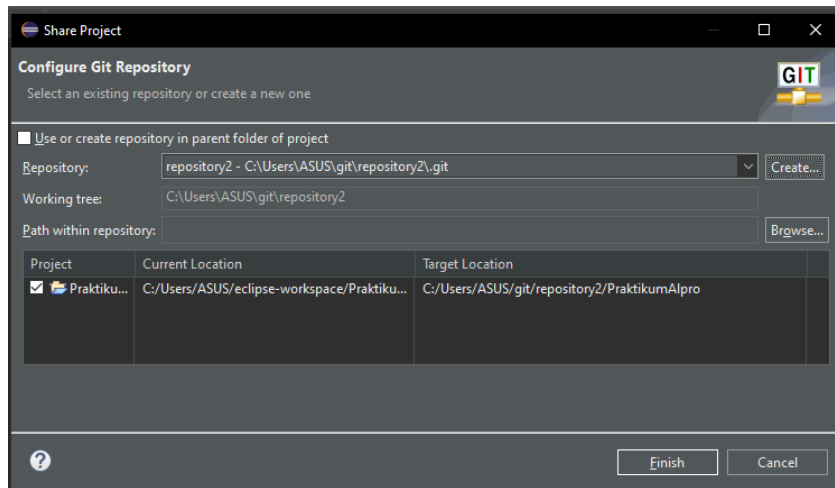
4. Kemudian klik pada menu “Create”.



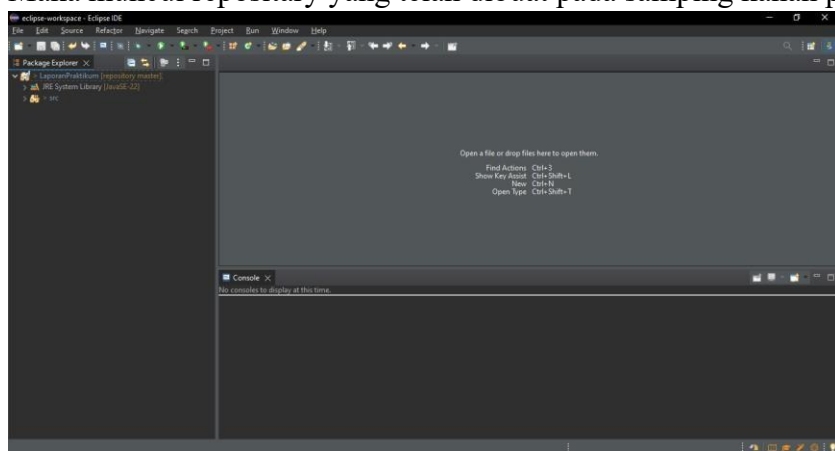
5. Buat nama repository local dengan tekan “Browse”. Untuk default branch name nya dibiarkan saja. Lalu klik “Finish”.



6. Setelah itu, klik “Finish”.

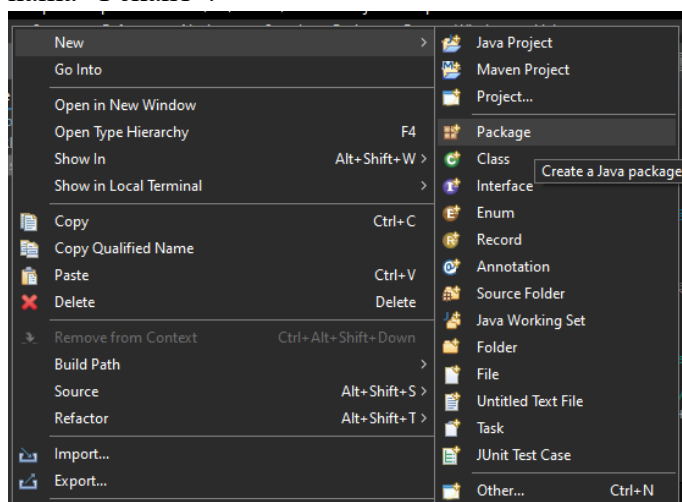


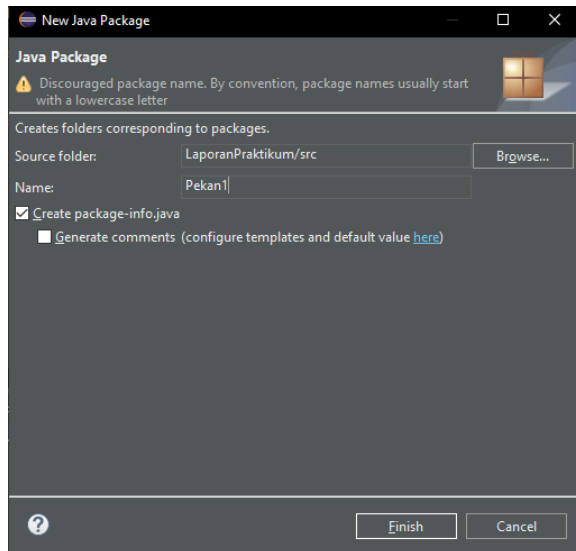
7. Maka muncul repository yang telah dibuat pada samping kanan project



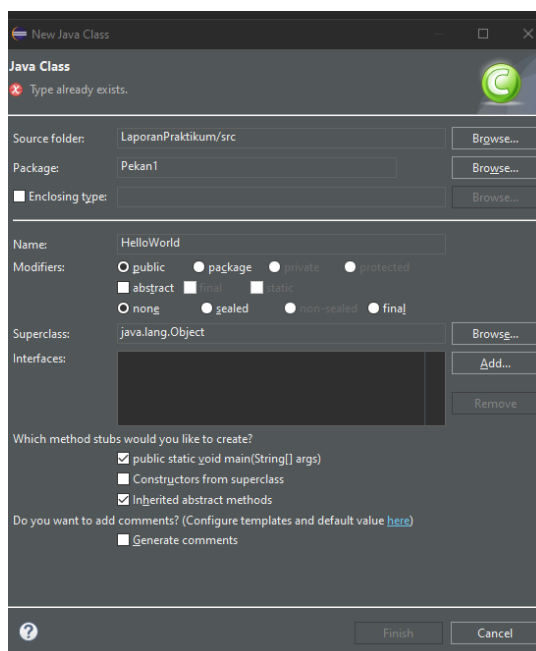
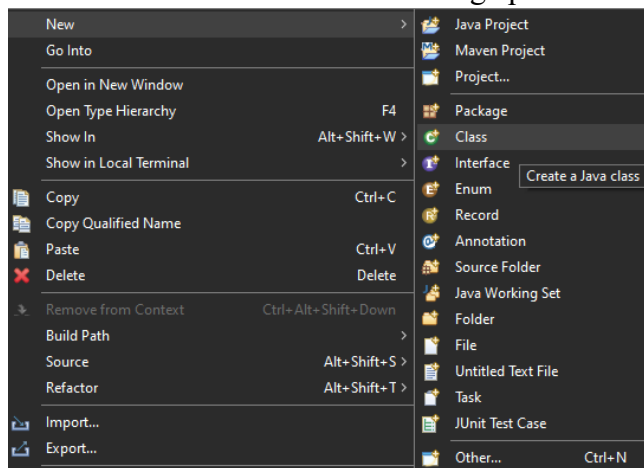
d. Program “Hello World”

1. Buat package terlebih dahulu dengan klik kanan pada folder src, lalu beri nama “Pekan1”.





2. Lalu klik kanan pada package “Pekan1”, pilih „new“ dan buat class baru, dan beri nama “HelloWorld” dan centang “public static void main (string[] args)”.



-
- The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: 'praktikumCoba' (master) contains a 'src' folder with 'praktikumCoba', 'HelloWorld.java', and 'module-info.java'. The main editor window shows the code for 'HelloWorld.java':
- ```

1 package praktikumCoba;
2
3 public class HelloWorld {
4
5 public static void main(String[] args) {
6 // TODO Auto-generated method stub
7
8 }
9
10 }
11

```
- The Outline view on the right shows the class hierarchy: 'praktikumCoba' contains 'HelloWorld', which has a 'main(String[]) void' method. The Console view at the bottom shows the output 'Hello World!'.

- ```
1  HelloWorld.java X
2  package Pekani;
3
4  public class HelloWorld {
5
6      public static void main(String[] args) {
7          // program pertama
8          /*komentar
9           * dua baris atau lebih
10          * pertemuan 1 praktikum alpro
11          */
12          System.out.println("Hello World");
13          System.out.println();
14          System.out.println("program ini menghasilkan");
15          System.out.println("Empat baris output");
16          System.out.println("/ \\ // \\\\ /// \\\\\\\");
17          System.out.println("This"+" program prints \\\n"+ "quote from the Gettysburg Address.");
18      }
19  }
```

-
- The screenshot shows an IDE with two main windows. The left window is the 'Package Explorer' showing a project structure with a package named 'Pekani' containing two classes: 'HelloWorld.java' and 'module-info.java'. The right window is the 'Editor' showing the source code of 'HelloWorld.java'. The code is as follows:
- ```

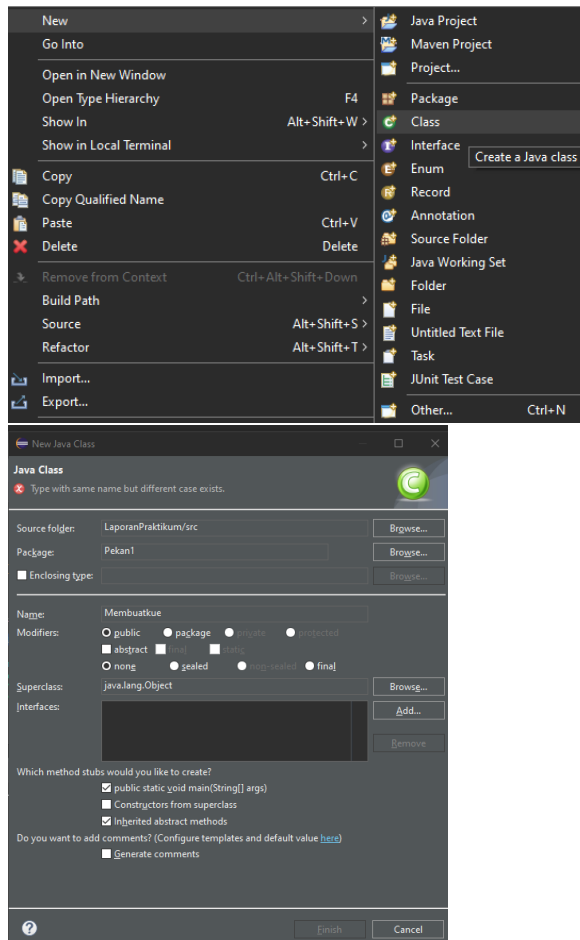
1 package Pekani;
2
3 public class HelloWorld {
4
5 public static void main(String[] args) {
6 // program pertama
7 // pertama
8 // Ask basic about java
9 // kemudian + kemudian akan
10
11 }
12 System.out.println("Hello World");
13 System.out.println();
14 System.out.println("program ini menghasilkan");
15 System.out.println("input basis output");
16 System.out.println("\n // \n \n \n \n \n \n");
17 System.out.println("This a program prints a quote from the Gettysburg Address");
18 }

```
- Below the editor is a 'Console' window showing the output of the program:
- ```

Hello World
program ini menghasilkan
input basis output
\n // \n \n \n \n \n \n
This program prints a
quote from the Gettysburg Address.

```

1. Buat new class dengan cara klik kanan pada package “Pekan1”, pilih „new“ lalu pilih „class“ dan beri nama “Membuatkue” dan centang “public static void main (string[] args)”.



2. Program ini bertujuan untuk memahami dalam memanggil method. Tulis 3 method sebagai berikut dan akhiri dengan buka/tutup kurung „()“ dan semicolon „;“ untuk memanggil method yang berisikan kode program.

```
1 package Pekan1;
2
3 public class membuatkue {
4
5     public static void main(String[] args) {
6         makeBatter();
7         memanggang();
8         menghias();
9     }
10 }
```

3. Tulis method pertama „public static void makeBatter() { }“. Tulis kode program didalam kurawal. Buat komentar sebagai pemberi keterangan. Tuliskan println untuk menghasilkan ouput pada console.

```
11 // langkah 1 : Buat adonan kue.
12 public static void makeBatter() {
13     System.out.println("Campur bahan kering.");
14     System.out.println("Krim mentega dan gula.");
15     System.out.println("Kocok telurnya.");
16     System.out.println("Masukkan bahan kering.");
17 }
```

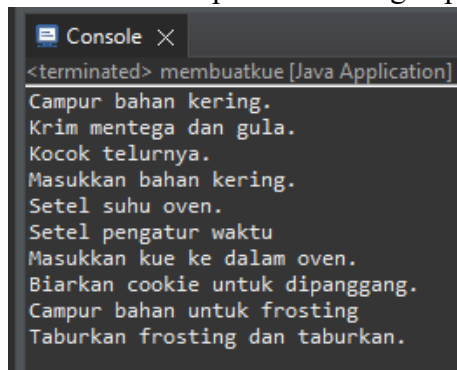
4. Tulis method pertama „public static void memanggang() { }“. Tulis kode program didalam kurawal. Buat komentar sebagai pemberi keterangan. Tuliskan println untuk menghasilkan ouput pada console.

```
18 // langkah 2: Panggang sekumpulan kue.
19 public static void memanggang() {
20     System.out.println("Setel suhu oven.");
21     System.out.println("Setel pengatur waktu");
22     System.out.println("Masukkan kue ke dalam oven.");
23     System.out.println("Biarkan cookie untuk dipanggang.");
24 }
```

5. Tulis method pertama „public static void menghias() { }“. Tulis kode program didalam kurawal. Buat komentar sebagai pemberi keterangan. Tuliskan println untuk menghasilkan ouput pada console.

```
25 // langkah 3: Hiasi cookie.  
26 public static void menghias() {  
27     System.out.println("Campur bahan untuk frosting");  
28     System.out.println("Taburkan frosting dan taburkan.");  
29 }
```

6. Jalankan program dengan klik pada bulatan hijau di bar menu (Run). Maka akan muncul ouput dari codingan program pada console.

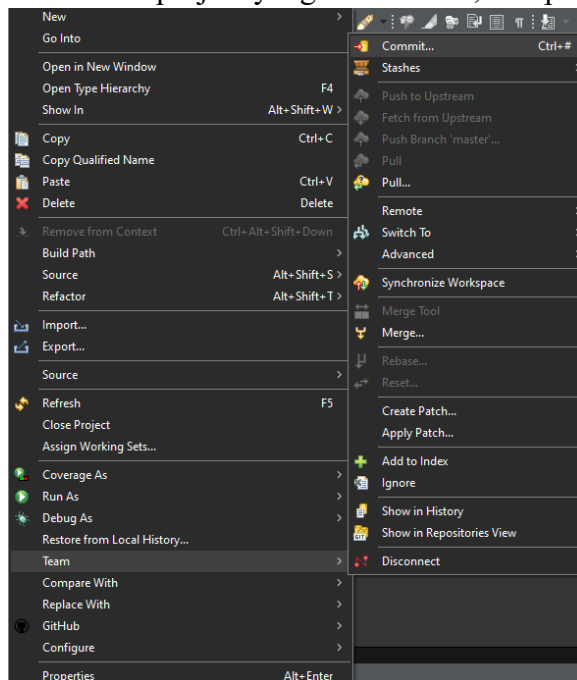


The screenshot shows the Eclipse IDE's console window. The title bar reads "Console X". The main text area displays the output of the program, which is a list of instructions for making cookies. The text is as follows:

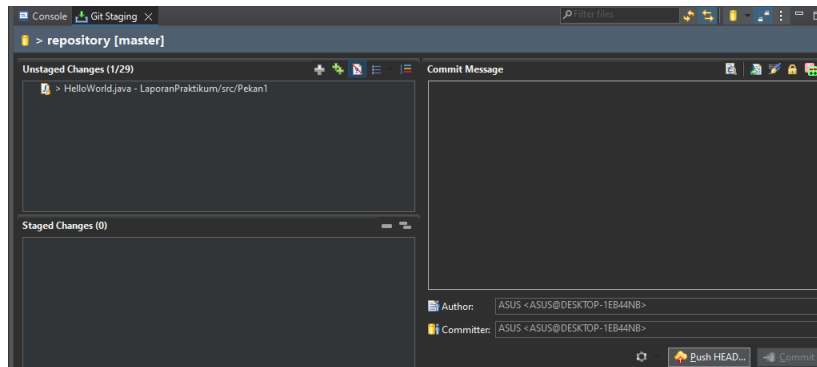
```
<terminated> membuatkue [Java Application]  
Campur bahan kering.  
Krim mentega dan gula.  
Kocok telurnya.  
Masukkan bahan kering.  
Setel suhu oven.  
Setel pengatur waktu  
Masukkan kue ke dalam oven.  
Biarkan cookie untuk dipanggang.  
Campur bahan untuk frosting  
Taburkan frosting dan taburkan.
```

f. Menghubungkan Eclipse ke Github

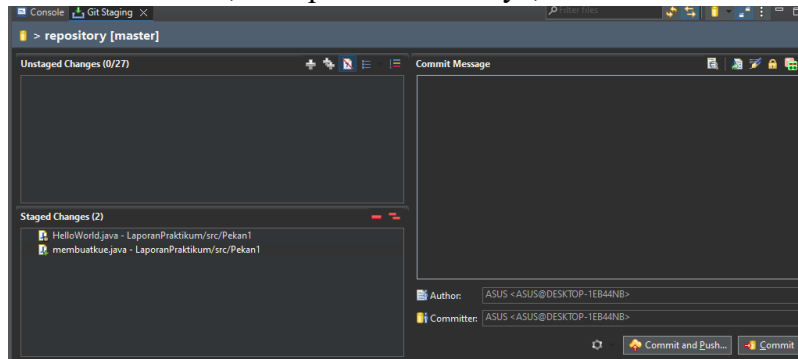
1. Klik kanan project yang telah dibuat, lalu pilih “Team”, lalu pilih “Commit”



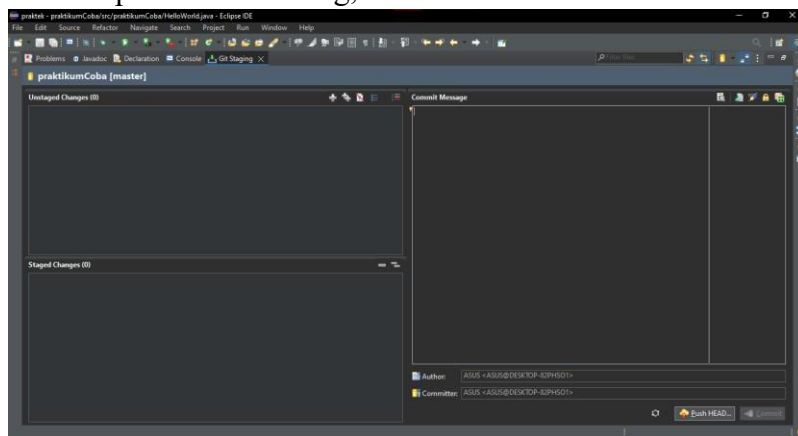
2. Lalu block semua file yang berada di atas, lalu tekan logo tambah plus double (++) untuk memindahkannya kebawah.



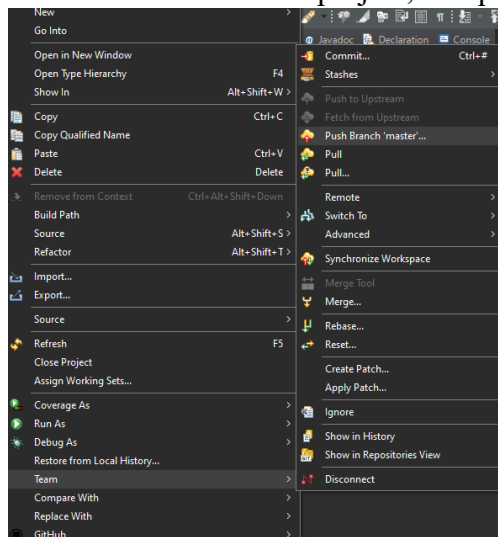
3. Setelah ditambah, buat pesan commitnya, lalu klik “Commit”



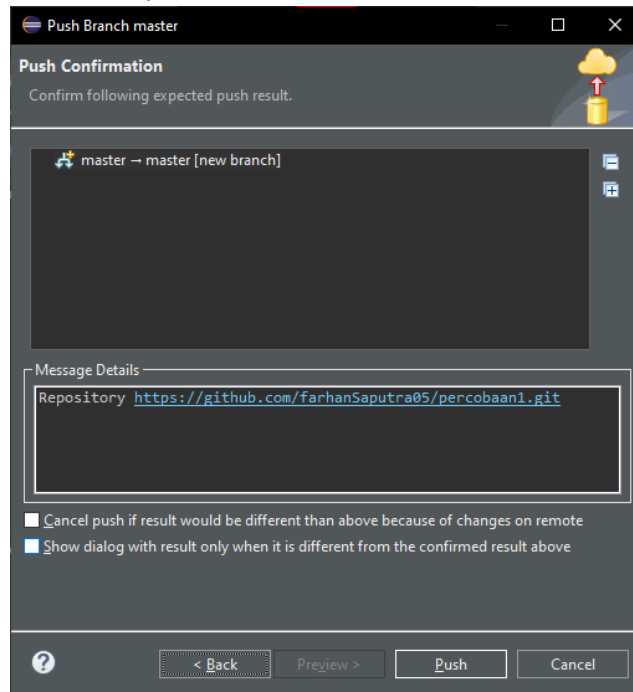
4. Jika tampilan sudah hilang, maka item berhasil di commit.



5. Kembali ke menu awal dengan cara mengklik pojok kiri atas, logo kotak dua, lalu klik kanan kembali project, lalu pilih team, lalu pilih push branch master.



6. Masukkan link URL yang disimpan tadi di awal. Untuk bagian user diisikan dengan username profile, dan untuk password diisikan dengan token yang dibuat di awal, lalu klik “Preview”.
7. Setelah itu, tekan “Push”..



8. Setelahnya file yang berada di Eclipse berhasil disalin di Repository GitHub.

D. Kesimpulan

Melalui praktikum ini dapat mempelajari cara membuat program Java dengan menggunakan Eclipse IDE dan JDK environment sehingga dapat memerintah komputer untuk menampilkan program yang dibuat. Mengetahui syntax Java untuk dapat menampilkan output menggunakan “System.out.println()”. Mengetahui bahasa Java yang bersifat case sensitive. Mempelajari urutan untuk membuat proyek menggunakan Eclipse dimulai dari new project, package, hingga class. Mempelajari pentingnya GitHub untuk programmer sebagai Repository untuk membuat proyek

bersama tim secara lebih fleksibel karena bias mengakses project melalui berbagai device dengan perantara Github.