

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN

PEKAN 6



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 6 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, 08 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 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Praktikum.....	26
BAB III PENUTUP.....	27
3.1 Kesimpulan.....	27
DAFTAR PUSTAKA.....	28

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Perkembangan teknologi informasi dan komputer pada saat ini telah memberikan pengaruh besar dalam berbagai aspek kehidupan. Salah satu dasar utama dalam bidang ilmu komputer adalah algoritma dan pemrograman, karena keduanya berperan penting dalam merancang solusi dari suatu permasalahan dengan cara yang sistematis, efisien, dan dapat diimplementasikan dalam bahasa pemrograman.

Algoritma merupakan urutan langkah-langkah logis yang disusun secara terstruktur untuk menyelesaikan suatu masalah. Sementara itu, pemrograman adalah proses penerjemahan algoritma ke dalam bentuk bahasa yang dapat dimengerti dan dijalankan oleh komputer. Pemahaman yang baik mengenai algoritma akan sangat membantu dalam menghasilkan program yang tidak hanya benar secara logika, tetapi juga efisien dalam penggunaan waktu dan sumber daya. Dalam konteks pemrograman, pemahaman terhadap struktur kontrol sangatlah penting. Salah satunya adalah struktur perulangan seperti **Do-While**. Struktur **Do-While** memastikan bahwa blok kode di dalamnya akan dieksekusi setidaknya satu kali sebelum kondisi perulangan diperiksa. Hal ini berbeda dengan **While** yang memeriksa kondisi di awal. Penggunaan perulangan **Do-While** sangat krusial dalam kasus-kasus seperti meminta input pengguna berulang kali hingga input yang valid dimasukkan, atau menjalankan proses minimum satu kali sebelum mengecek kebutuhan untuk iterasi berikutnya.

Melalui praktikum Algoritma dan Pemrograman, saya diharapkan mampu menguasai keterampilan dasar dalam merancang, menuliskan, serta menguji algoritma menggunakan berbagai bahasa pemrograman. Dengan adanya praktikum ini, mahasiswa tidak hanya memahami teori, tetapi juga memperoleh pengalaman langsung dalam mengimplementasikan konsep algoritma menjadi sebuah program yang dapat dijalankan oleh komputer. Hal ini akan

memperkuat pemahaman, menambah keterampilan, serta mempersiapkan mahasiswa untuk menghadapi tantangan di dunia kerja.

1.2. Tujuan Praktikum

- Memahami konsep dasar algoritma sebagai langkah-langkah logis dan sistematis dalam penyelesaian masalah.
- Melatih keterampilan dalam menerjemahkan algoritma ke dalam bentuk kode program menggunakan bahasa pemrograman.
- Mengembangkan kemampuan berpikir kritis, analitis, dan terstruktur dalam menyelesaikan suatu permasalahan.
- Mengetahui cara menuliskan program yang benar, efisien, dan mudah dipahami, termasuk memahami dan menerapkan struktur kontrol perulangan seperti **Do-While**.
- Melatih ketelitian serta pemahaman logika dalam proses debugging dan pengujian program.

1.3. Manfaat Praktikum

- **Peningkatan Keterampilan Praktis:** Saya memperoleh pengalaman langsung dalam menulis, mengkompilasi, dan menjalankan kode program, menjembatani kesenjangan antara teori dan implementasi.
- **Penguatan Logika Berpikir:** Melalui pemecahan masalah (problem solving) secara terstruktur dan implementasi struktur kontrol (termasuk Do-While), kemampuan berpikir logis dan analitis saya terasah.
- **Kesiapan Karir:** Membekali diri saya dengan keterampilan dasar pemrograman yang sangat dicari di industri teknologi, mempersiapkan mereka untuk peran developer atau analis.
- **Pemahaman Mendalam:** Praktikum membantu memahami konsep-konsep abstrak seperti efisiensi kode, debugging, dan testing dengan cara yang lebih nyata dan aplikatif.

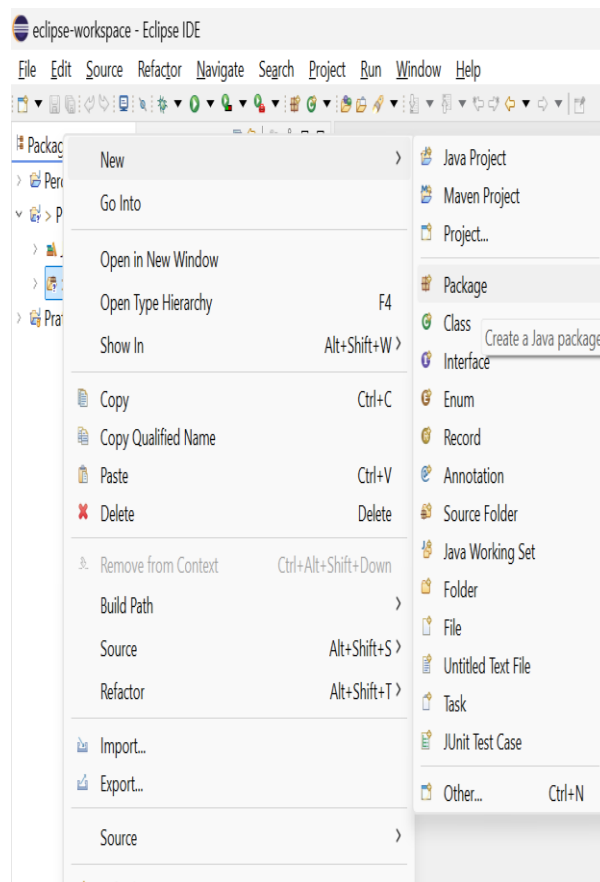
BAB II

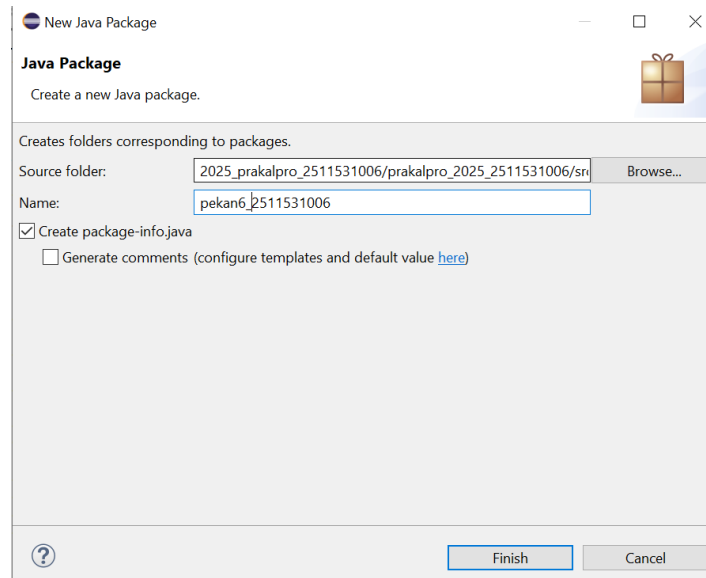
PEMBAHASAN

2.1.Langkah Kerja Praktikum

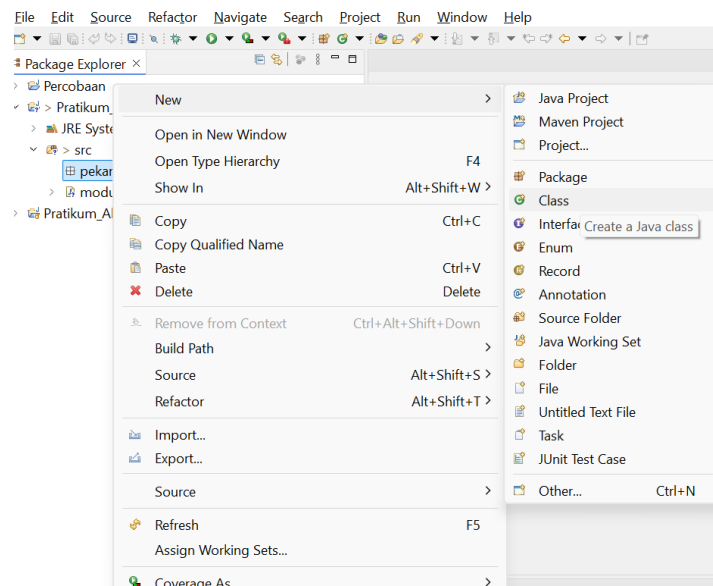
1.Program Pertama

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. Maka Tampilan akan seperti berikut.

```

1 package pekan6_2511531006;
2
3 import java.util.Random;
4
5 public class LemparDadu_2511531006 {
6
7     public static void main(String[] args) {

```

d. Paragraf ini merupakan inisialisasi awal dan kontrol utama dari sebuah program simulasi acak, kemungkinan besar program pelemparan dadu seperti yang terlihat di kode sebelumnya. Baris 8, `Random rand = new Random();`, berfungsi untuk menginisialisasi objek `Random` dari library Java, yang akan digunakan untuk menghasilkan angka acak (simulasi pelemparan dadu). Selanjutnya, dua variabel integer diinisialisasi: `int tries = 0;` (Baris 9) yang berfungsi sebagai penghitung berapa kali proses perulangan atau percobaan telah dilakukan, dan `int sum = 0;` (Baris 10) yang menyimpan jumlah total dari pelemparan dadu. Setelah inisialisasi, program memasuki struktur kontrol perulangan utama yaitu `while (sum != 7);` (Baris 11). Perulangan `while` ini akan terus mengeksekusi blok kode di dalamnya (yang tidak terlihat dalam potongan ini, tetapi berisi logika pelemparan dadu) selama nilai variabel `sum` (jumlah dadu) tidak sama dengan 7. Dengan kata lain, program akan terus melempar dadu dan menghitung totalnya sampai hasil pelemparan mencapai angka 7.

```

8     Random rand = new Random();
9     int tries = 0;
10    int sum = 0;
11    while (sum != 7); {

```

e. Potongan kode pada gambar tersebut merupakan bagian dari program Java yang berfungsi untuk mensimulasikan pelemparan dua buah dadu sekali. Pada baris `int dadu1 = rand.nextInt(6) + 1;` dan `int dadu2 = rand.nextInt(6) + 1;`, program menggunakan objek `rand` (tipe `Random`) untuk menghasilkan dua bilangan acak antara 1 hingga 6, yang masing-masing mewakili hasil dari lemparan dadu pertama dan kedua. Nilai kedua dadu kemudian dijumlahkan dan disimpan dalam variabel `sum`. Selanjutnya, perintah `System.out.print(dadu1 + " + " + dadu2 + " = " + sum);` digunakan untuk menampilkan hasil lemparan kedua dadu beserta totalnya dalam

format seperti “3 + 4 = 7”. Terakhir, `tries++`; berfungsi untuk menambah nilai variabel penghitung `tries` sebanyak satu, menandakan bahwa satu kali percobaan lempar dadu telah dilakukan.

```
12         //roll the dice once
13         int dadu1 = rand.nextInt(6) + 1;
14         int dadu2 = rand.nextInt(6) + 1;
15         sum = dadu1 + dadu2;
16         System.out.print(dadu1 + " + " + dadu2 + " = " + sum);
17         tries++;
18
19     }
```

f. Potongan kode `System.out.println("You won after " + tries + " tries!");` pada gambar tersebut berfungsi untuk menampilkan pesan ke layar (console) yang memberi tahu pemain berapa kali percobaan yang dibutuhkan untuk menang dalam sebuah permainan. Perintah `System.out.println` digunakan untuk mencetak teks diikuti dengan nilai variabel `tries`, yang menyimpan jumlah percobaan yang dilakukan pemain. Tanda `+` berfungsi sebagai operator penggabung (concatenation) antara teks dan nilai variabel. Jadi, jika misalnya `tries` bernilai 3, maka hasil yang muncul di layar adalah: "You won after 3 tries!". Baris kode ini biasanya diletakkan di akhir program setelah kondisi kemenangan tercapai.

```
20         System.out.println("You won after " + tries + " tries!");
21
22     }
23
24
25
26 }
```

g. Maka akhirnya menjadi seperti ini

```

1 package pekan6_2511531006;
2
3 import java.util.Random;
4
5 public class LemparDadu_2511531006 {
6
7     public static void main(String[] args) {
8         Random rand = new Random();
9         int tries = 0;
10        int sum = 0;
11        while (sum != 7); {
12            //roll the dice once
13            int dadu1 = rand.nextInt(6) + 1;
14            int dadu2 = rand.nextInt(6) + 1;
15            sum = dadu1 + dadu2;
16            System.out.print(dadu1 + " + " + dadu2 + " = " + sum);
17            tries++;
18        }
19        System.out.println("You won after " + tries + " tries!");
20
21    }
22
23 }
24
25 }

```

h.Hasil Run

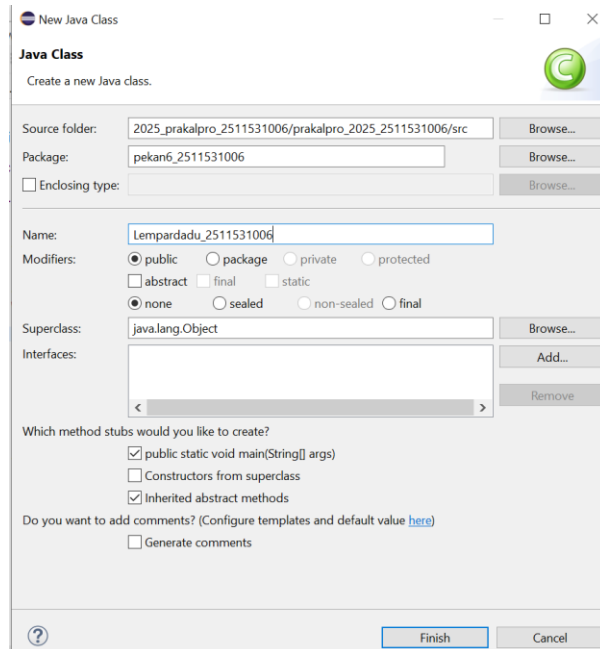
```

6 + 2 = 8
3 + 6 = 9
3 + 1 = 4
3 + 4 = 7
You won after 4 tries!

```

2.Kode Program Kedua

a. Pertama tama 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 di tekan finish, maka Tampilan akan seperti berikut

```
1 package pekan6_2511531006;
2
3 import java.util.Random;
4
5
6 public class GamePenjumlahan_2511531006 {
7
8     public static void main(String[] args) {
```

c. Potongan kode tersebut menampilkan dua baris perintah penting dalam bahasa pemrograman Java. Baris pertama, `Scanner console = new Scanner(System.in);`, berfungsi untuk membuat objek Scanner yang digunakan agar program bisa menerima input dari pengguna melalui keyboard. Artinya, setiap kali pengguna mengetik sesuatu, nilai tersebut bisa dibaca dan disimpan oleh program. Baris kedua, `Random rand = new Random();`, digunakan untuk membuat objek Random yang berfungsi menghasilkan angka acak, misalnya untuk membuat soal atau pilihan secara acak. Komentar pada baris berikutnya, `// play until user gets 3 wrong`, menjelaskan bahwa program akan terus dijalankan sampai pengguna melakukan tiga kesalahan. Jadi, bagian kode ini adalah tahap persiapan untuk membaca input pengguna dan menghasilkan angka acak dalam permainan atau kuis yang sedang dibuat.

```

9      Scanner console = new Scanner(System.in);
10     Random rand = new Random();
11     // play until user gets 3 wrong

```

d. Potongan kode ini menunjukkan logika utama dari sebuah permainan atau kuis sederhana. Dua variabel awal, yaitu `points` dan `wrong`, masing-masing diinisialisasi dengan nilai 0 untuk menyimpan jumlah poin benar dan jumlah kesalahan pengguna. Selanjutnya, terdapat perulangan `while` (`wrong < 3`) yang berarti program akan terus berjalan selama jumlah kesalahan pengguna masih kurang dari tiga. Di dalam perulangan, baris `int result = play(console, rand);` memanggil metode `play()` untuk menjalankan satu ronde permainan, di mana pengguna memberikan jawaban dan sistem menentukan hasilnya. Jika nilai `result` lebih besar dari 0 (berarti jawaban benar), maka `points++` menambah satu poin. Namun, jika hasilnya tidak lebih besar dari 0 (berarti salah), maka `wrong++` menambah jumlah kesalahan. Dengan demikian, program ini akan terus menjalankan permainan hingga pengguna salah sebanyak tiga kali.

```

12     int points = 0;
13     int wrong = 0;
14     while (wrong < 3) {
15         int result = play(console, rand); // play one game
16         if (result > 0) {
17             points++;
18         } else {
19             wrong++;
20         }
21     }
22 }

```

e. Potongan kode ini merupakan bagian akhir dari program yang berfungsi untuk menampilkan hasil permainan kepada pengguna. Baris `System.out.println("You earned " + points + " total points.");` digunakan untuk mencetak teks ke layar. Teks yang dicetak akan menampilkan kalimat "You earned X total points.", di mana X merupakan jumlah poin yang telah dikumpulkan pengguna selama permainan berlangsung. Nilai `points` berasal dari variabel yang dihitung dalam perulangan sebelumnya setiap kali pengguna menjawab dengan benar.

Dengan kata lain, baris ini memberikan umpan balik kepada pengguna berupa total skor akhir setelah permainan selesai.

```
23     System.out.println("You earned " + points + " total points.");
24 }
```

f. Potongan kode ini merupakan bagian dari metode `play()` yang digunakan untuk membuat dan menampilkan soal penjumlahan secara acak kepada pengguna. Metode ini bersifat `public static` dan mengembalikan nilai bertipe `int`. Di dalam parameter, terdapat dua objek yaitu `Scanner console` untuk membaca input dari pengguna dan `Random rand` untuk menghasilkan angka acak. Baris `int operands = rand.nextInt(4) + 2;` digunakan untuk menentukan jumlah angka (operand) yang akan dijumlahkan secara acak antara 2 sampai 5. Kemudian `int sum = rand.nextInt(10) + 1;` menghasilkan sebuah angka acak antara 1 hingga 10 yang akan menjadi salah satu bilangan dalam soal penjumlahan. Terakhir, `System.out.print(sum);` berfungsi untuk menampilkan nilai tersebut di layar, sehingga pengguna dapat melihat angka yang akan digunakan dalam soal.

```
25 // membuat soal penjumlahan dan ditampilkan ke user
26 public static int play(Scanner console, Random rand) {
27     // print the operands being added, and sum them
28     int operands = rand.nextInt(4) + 2;
29     int sum = rand.nextInt(10) + 1;
30     System.out.print(sum);
```

g. Potongan kode ini merupakan bagian dari metode `play()` yang berfungsi untuk membentuk soal penjumlahan dengan beberapa bilangan acak. Baris `int i;` hanya mendeklarasikan variabel sementara (meskipun tidak digunakan). Kemudian, perulangan `for (int i1 = 2; i1 <= operands; i1++)` digunakan untuk mengulangi proses penambahan angka sesuai jumlah operand yang telah ditentukan sebelumnya. Di dalam perulangan, `int n = rand.nextInt(10) + 1;` menghasilkan angka acak antara 1 hingga 10. Nilai ini kemudian dijumlahkan ke variabel `sum` menggunakan `sum += n;`, yang berarti total penjumlahan akan terus bertambah setiap kali loop dijalankan. Selanjutnya, `System.out.print(" + " + n);` berfungsi menampilkan tanda tambah

dan angka acak tersebut ke layar, sehingga program akan mencetak bentuk soal seperti “5 + 3 + 7”. Setelah loop selesai, baris `System.out.print(" = ");` mencetak tanda sama dengan di akhir, menandakan bahwa pengguna harus mengisi hasil dari penjumlahan tersebut.

```
31     int i;  
32     for (int i1 = 2; i1 <= operands; i1++); {  
33         int n = rand.nextInt(10) + 1;  
34         sum += n;  
35         System.out.print(" + " + n);  
36     }  
37     System.out.print(" = ");
```

h. Potongan kode ini adalah bagian akhir dari metode `play()` yang berfungsi untuk mengecek apakah jawaban pengguna benar atau salah. Baris `int guess = console.nextInt();` digunakan untuk membaca input angka dari pengguna melalui keyboard — angka ini merupakan tebakan atau jawaban pengguna terhadap soal penjumlahan yang telah ditampilkan sebelumnya.

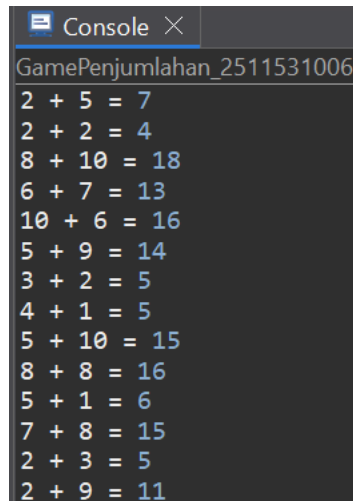
Selanjutnya, perintah `if (guess == sum)` digunakan untuk membandingkan jawaban pengguna dengan hasil penjumlahan yang benar (`sum`). Jika nilainya sama, maka program akan mengembalikan nilai 1 menggunakan `return 1;`, yang menandakan jawaban benar dan nantinya akan menambah poin pengguna di bagian utama program. Namun, jika jawaban salah, blok `else` akan dijalankan. Di dalamnya terdapat `System.out.println("Wrong! The answer was " + sum);` yang akan menampilkan pesan bahwa jawaban salah, serta menunjukkan jawaban yang benar. Setelah itu, `return 0;` dikembalikan untuk menandakan jawaban salah, sehingga jumlah kesalahan pengguna akan bertambah.

```
38  
39     // read user's guess and report whether it was correct  
40     int guess = console.nextInt();  
41     if (guess == sum) {  
42         return 1;  
43     } else {  
44         System.out.println("Wrong! The answer was " + sum);  
45         return 0;  
46     }  
47  
48 }  
49  
50 }
```

i. Tampilan semua kode program kedua

```
1 package pekan6_2511531006;
2
3 import java.util.Random;
4
5
6 public class GamePenjumlahan_2511531006 {
7
8     public static void main(String[] args) {
9         Scanner console = new Scanner(System.in);
10        Random rand = new Random();
11        // play until user gets 3 wrong
12        int points = 0;
13        int wrong = 0;
14        while (wrong < 3) {
15            int result = play(console, rand); // play one game
16            if (result > 0) {
17                points++;
18            } else {
19                wrong++;
20            }
21        }
22        System.out.println("You earned " + points + " total points.");
23    }
24    // membuat soal penjumlahan dan ditampilkan ke user
25    public static int play(Scanner console, Random rand) {
26        // print the operands being added, and sum them
27        int operands = rand.nextInt(4) + 2;
28        int sum = rand.nextInt(10) + 1;
29        System.out.print(sum);
30        int i;
31        for (int i1 = 2; i1 <= operands; i1++); {
32            int n = rand.nextInt(10) + 1;
33            sum += n;
34            System.out.print(" + " + n);
35        }
36        System.out.print(" = ");
37
38        // read user's guess and report whether it was correct
39        int guess = console.nextInt();
40        if (guess == sum) {
41            return 1;
42        } else {
43            System.out.println("Wrong! The answer was " + sum);
44            return 0;
45        }
46    }
47
48 }
49
50 }
```

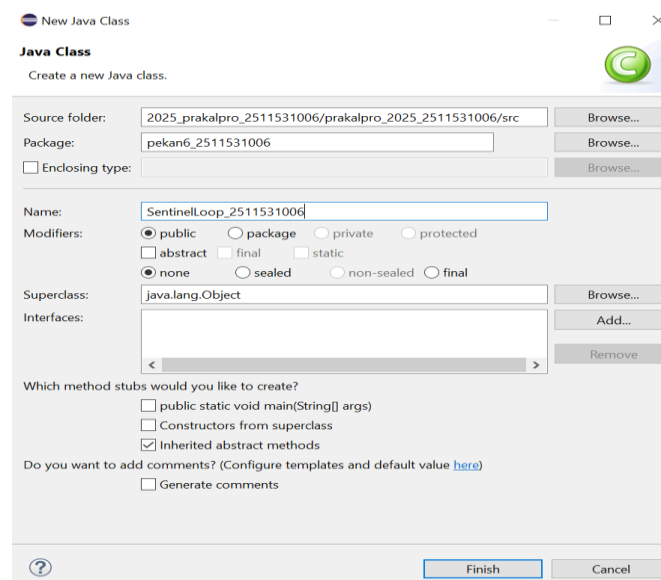
j.Hasil Run



```
GamePenjumlahan_2511531006
2 + 5 = 7
2 + 2 = 4
8 + 10 = 18
6 + 7 = 13
10 + 6 = 16
5 + 9 = 14
3 + 2 = 5
4 + 1 = 5
5 + 10 = 15
8 + 8 = 16
5 + 1 = 6
7 + 8 = 15
2 + 3 = 5
2 + 9 = 11
```

3.Kode Program Ketiga

a. Pertama tama 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 di tekan finish, maka Tampilan akan seperti berikut.

```
1 package pekan6_2511531006;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511531006 {
6
7     public static void main(String[] args) {
```

c. Potongan kode tersebut membuat objek Scanner bernama console untuk menerima input dari pengguna melalui keyboard. Variabel sum diinisialisasi dengan nilai 0 untuk menyimpan total penjumlahan nanti, sedangkan variabel number diberi nilai awal 12 sebagai “dummy value” atau nilai sementara yang bisa diganti nanti sesuai input atau kebutuhan program.

```
8     Scanner console = new Scanner (System.in);
9     int sum = 0;
10    int number=12; // "dummy value", anything
11
```

d. Potongan kode itu membuat program yang terus meminta saya memasukkan angka sampai saya mengetik angka 0. Selama angka yang dimasukkan bukan 0, program akan terus berjalan. Baris `System.out.print("Masukkan angka (0 untuk keluar): ");` digunakan untuk menampilkan perintah agar saya memasukkan angka. Setelah itu, `number = console.nextInt();` akan menyimpan angka yang saya masukkan ke variabel number. Lalu, `sum = sum + number;` akan menambahkan angka tersebut ke total penjumlahan. Jika saya memasukkan 0, kondisi `while (number != 0)` menjadi salah sehingga perulangan berhenti dan program selesai dijalankan. Jadi, program ini berfungsi untuk menjumlahkan semua angka yang saya masukkan sampai saya mengetik 0 sebagai tanda keluar.

```
12    while (number != 0) {
13        System.out.print("Masukkan angka (0 untuk keluar_); ");
14        number = console.nextInt();
15        sum = sum + number;
16    }
```

e. Bagian kode ini berfungsi untuk menampilkan hasil akhir dari penjumlahan semua angka yang sudah saya masukkan sebelumnya. Baris `System.out.println("totalnya adalah " + sum);` akan mencetak teks “totalnya adalah” diikuti dengan nilai dari variabel `sum`. Jadi, setelah saya selesai memasukkan angka dan mengetik 0 untuk keluar, program akan menampilkan jumlah totalnya. Lalu, baris `console.close();` digunakan untuk menutup objek `Scanner` agar input dari keyboard tidak lagi digunakan dan supaya program lebih rapi serta bebas dari peringatan sumber daya yang masih terbuka.

```
16     }
17     System.out.println("totalnya adalah " + sum);
18     console.close();
19     }
20 }
```

f. Tampilan semua kode program ketiga.

```
1 package pekan6_2511531006;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511531006 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner (System.in);
9         int sum = 0;
10        int number=12; // "dummy value", anything
11
12        while (number != 0) {
13            System.out.print("Masukkan angka (0 untuk keluar_); ");
14            number = console.nextInt();
15            sum = sum + number;
16        }
17        System.out.println("totalnya adalah " + sum);
18        console.close();
19    }
20 }
```

g. Hasil Run kode program

```
Console X
<terminated> SentinelLoop_2511531006 [Java Appli
Masukkan angka (0 untuk keluar_); 0
totalnya adalah 0
```

4.Kode Program keempat

a. Pertama tama 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)”.

New Java Class

Java Class

Create a new Java class.

Source folder: 2025_prakalpro_2511531006/prakalpro_2025_2511531006/src Browse...

Package: pekan6_2511531006 Browse...

Enclosing type: Browse...

Name: PerulanganWhilel_2511531006

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

b. Jika sudah di tekan finish, maka akan jadi seperti ini.

```

1 package pekan6_2511531006;
2
3 import java.util.Scanner;
4
5 public class PerulanganWhile1_2511531006 {
6
7     public static void main(String[] args) {

```

c. Potongan kode ini berisi tiga variabel dengan tipe data berbeda. Variabel counter bertipe int dan diisi 0, biasanya dipakai untuk menghitung berapa kali proses atau perulangan dijalankan. Variabel jawab bertipe String, disiapkan untuk menyimpan jawaban yang saya masukkan, misalnya “ya” atau “tidak”. Sedangkan running bertipe boolean dan diisi true, yang artinya program dalam keadaan berjalan. Nantinya nilai running bisa diubah jadi false kalau saya mau menghentikan program.

```

8         int counter=0;
9         String jawab;
10        boolean running = true;

```

d. Bagian kode ini mendeklarasikan objek Scanner bernama scan yang berfungsi untuk membaca input dari keyboard. Setelah itu ada perulangan while (running) yang akan terus berjalan selama nilai variabel running masih true. Di dalam perulangan, ada baris counter++ yang artinya setiap kali loop dijalankan, nilai counter akan bertambah satu. Jadi, bagian ini bisa dipakai untuk menghitung berapa kali perulangan terjadi selama program masih aktif.

```

11        //deklarasi scanner
12        Scanner scan = new Scanner (System.in);
13        while (running) {
14            counter++;

```

e. Potongan kode ini menampilkan hasil dari variabel counter dan meminta saya memutuskan apakah ingin melanjutkan program atau tidak. Baris `System.out.println("Jumlah = " + counter);` akan menampilkan jumlah perulangan yang sudah terjadi. Lalu,

System.out.println("Apakah lanjut (ya / tidak?"); menanyakan ke saya apakah mau lanjut. Setelah itu, jawab = scan.nextLine(); digunakan untuk membaca jawaban yang saya ketik di keyboard, misalnya “ya” untuk lanjut atau “tidak” untuk berhenti.

```
15         System.out.println("Jumlah = "+counter);
16         System.out.println("Apakah lanjut (ya / tidak?");
17         jawab= scan.nextLine();
```

f. Potongan kode ini berfungsi untuk mengecek apakah saya ingin menghentikan perulangan atau tidak. Komentarnya menjelaskan bahwa kalau jawaban saya adalah “tidak”, maka perulangan akan berhenti. Baris if (jawab.equalsIgnoreCase(jawab)) sebenarnya kurang tepat, karena membandingkan variabel yang sama. Seharusnya ditulis seperti if (jawab.equalsIgnoreCase("tidak")), agar program bisa mendeteksi kalau saya mengetik “tidak” (dengan huruf besar atau kecil apa pun). Jika kondisi itu benar, maka running = false; dijalankan, artinya program akan menghentikan perulangan while.

```
18         //cek jawab = tidak, perulangan berhenti
19         if (jawab.equalsIgnoreCase(jawab)) {
20             running= false;
21         }
22     }
```

g. Potongan kode ini digunakan untuk menampilkan hasil akhir setelah perulangan selesai. Baris System.out.println("Anda sudah melakukan perulangan sebanyak " + counter + " kali"); akan mencetak pesan yang memberitahu saya berapa kali perulangan sudah dijalankan selama program aktif. Nilai dari variabel counter diambil dari hitungan yang terus bertambah setiap kali loop berjalan. Jadi, setelah saya memilih untuk berhenti (menjawab “tidak”), program akan menampilkan total jumlah perulangan yang sudah saya lakukan.

```

23     System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
24
25
26 }
27
28
29
30
31
32 }

```

h. Tampilan semua kode program keempat

```

1  package pekan6_2511531006;
2
3  import java.util.Scanner;
4
5  public class PerulanganWhile1_2511531006 {
6
7      public static void main(String[] args) {
8          int counter=0;
9          String jawab;
10         boolean running = true;
11         //deklarasi scanner
12         Scanner scan = new Scanner (System.in);
13         while (running) {
14             counter++;
15             System.out.println("Jumlah = "+counter);
16             System.out.println("Apakah lanjut (ya / tidak?)");
17             jawab= scan.nextLine();
18             //cek jawab = tidak, perulangan berhenti
19             if (jawab.equalsIgnoreCase(jawab)) {
20                 running= false;
21             }
22         }
23         System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
24
25     }
26
27
28
29
30
31
32 }

```

g. Hasil run kode program

```

Console X
<terminated> perulanganWhile1_2511531006 [Java Application] C:\
Jumlah = 1
Apakah lanjut (ya / tidak?)
ya
Anda sudah melakukan perulangan sebanyak 1 kali

```

5.Kode Program Kelima

a. Pertama tama 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)”

Java Class
Create a new Java class.

Source folder: 2025_prakalpro_2511531006/prakalpro_2025_2511531006/src Browse...

Package: pekan6_2511531006 Browse...

☐ Enclosing type: Browse...

Name: Dowhile1_2511531006

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add...
Remove

Which method stubs would you like to create?

☒ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

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

```
1 package pekan6_2511531006;  
2  
3 import java.util.Scanner;  
4  
5 public class Dowhile1_2511531006 {  
6  
7     public static void main(String[] args) {
```

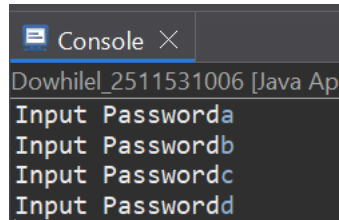
c. Potongan kode di atas adalah bagian dari program Java yang menggunakan kelas Scanner untuk membaca input dari pengguna melalui keyboard. Baris `Scanner console = new Scanner(System.in);` berfungsi untuk membuat objek bernama `console` yang akan digunakan untuk menerima data dari pengguna lewat konsol. Selanjutnya, terdapat deklarasi variabel `String phrase;` yang berarti program menyiapkan variabel bernama `phrase` dengan tipe data string untuk menyimpan teks yang dimasukkan oleh pengguna nanti. Setelah itu, baris `do {` menandakan bahwa program akan memulai sebuah perulangan `do-while`, yaitu jenis perulangan yang menjalankan blok kode di dalamnya terlebih dahulu sebelum memeriksa kondisi di bagian `while`. Biasanya struktur seperti ini digunakan untuk meminta input berulang kali sampai pengguna memberikan jawaban tertentu.

```
8      Scanner console = new Scanner(System.in);
9      String phrase;
10     do {
```

d. Potongan kode pada gambar tersebut merupakan lanjutan dari program sebelumnya yang berfungsi untuk meminta pengguna memasukkan password hingga sesuai dengan yang ditentukan. Pada baris `System.out.print("Input Password");`, program menampilkan pesan agar pengguna mengetikkan password. Input dari pengguna kemudian dibaca menggunakan `phrase = console.next();`, yang berarti nilai yang diketik akan disimpan ke dalam variabel `phrase`. Setelah itu, perulangan `do-while` akan terus berjalan selama kondisi `!phrase.equals("abcd")` bernilai benar. Tanda `!` berarti “tidak sama dengan”, sehingga perulangan akan berhenti hanya jika pengguna memasukkan kata “abcd” dengan tepat. Dengan kata lain, program ini memaksa pengguna untuk terus menginput password sampai password yang dimasukkan benar.

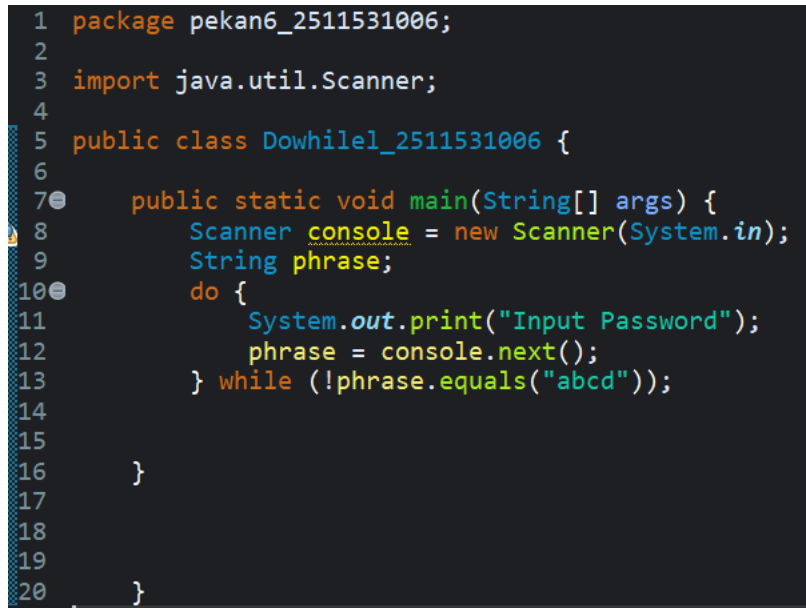
```
11         System.out.print("Input Password");
12         phrase = console.next();
13     } while (!phrase.equals("abcd"));
14
15
16     }
17
18
19
20 }
```


e.Run kode program



```
Console ×
Dowhilel_2511531006 [Java Ap
Input Passworda
Input Passwordb
Input Passwordc
Input Passwordd
```

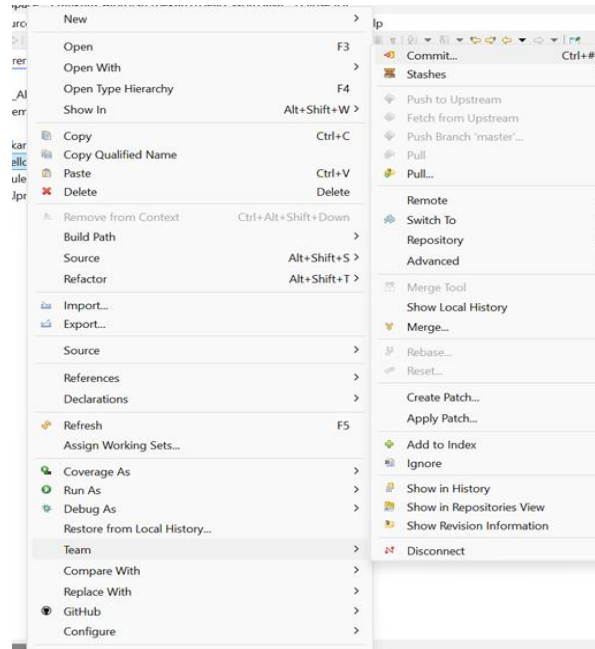
f. Tampilan semua kode program kelima.



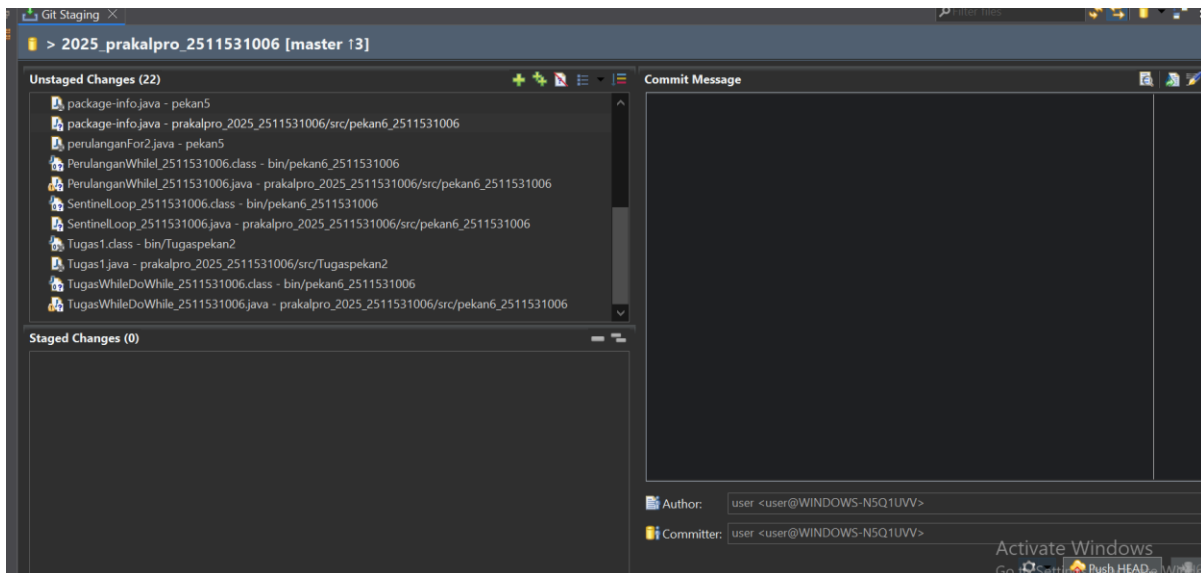
```
1 package pekan6_2511531006;
2
3 import java.util.Scanner;
4
5 public class Dowhilel_2511531006 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner(System.in);
9         String phrase;
10        do {
11            System.out.print("Input Password");
12            phrase = console.next();
13        } while (!phrase.equals("abcd"));
14
15
16    }
17
18
19
20 }
```

6.Menghubungkan ke github

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

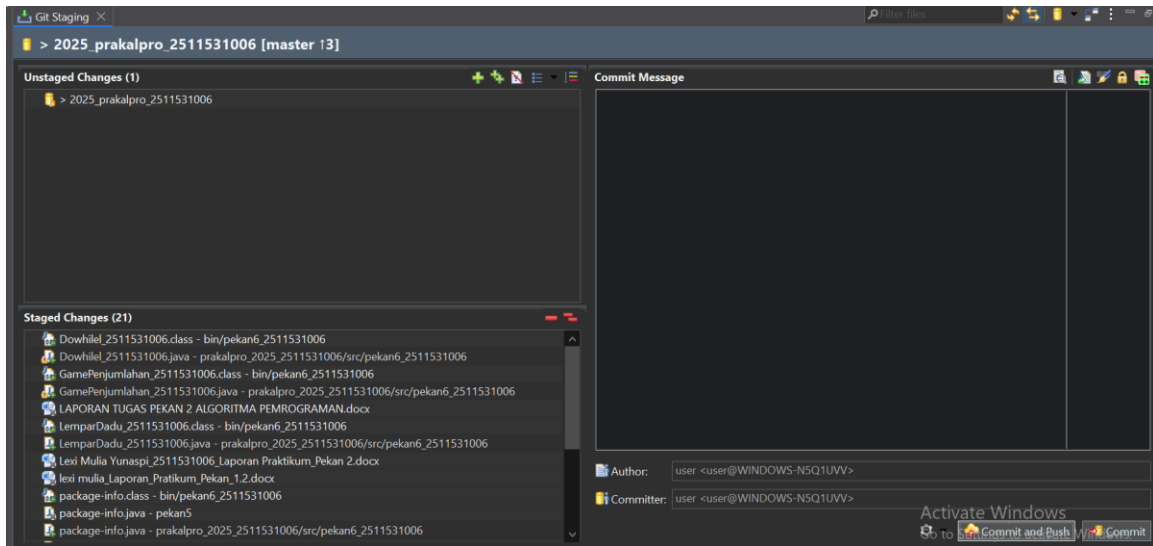


b. Block semua yang berada di atas dengan , lalu tekan logo tambah.

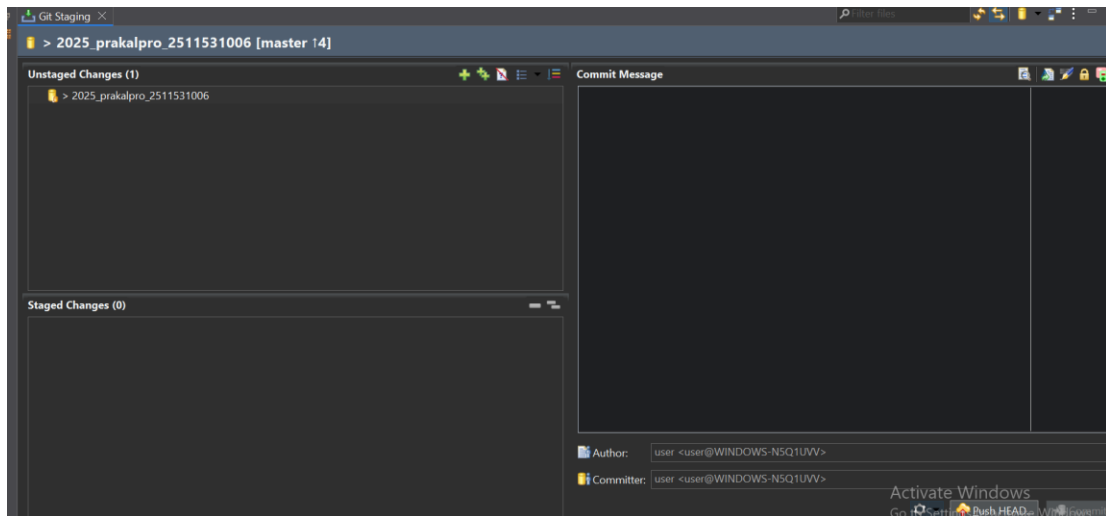


c.

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



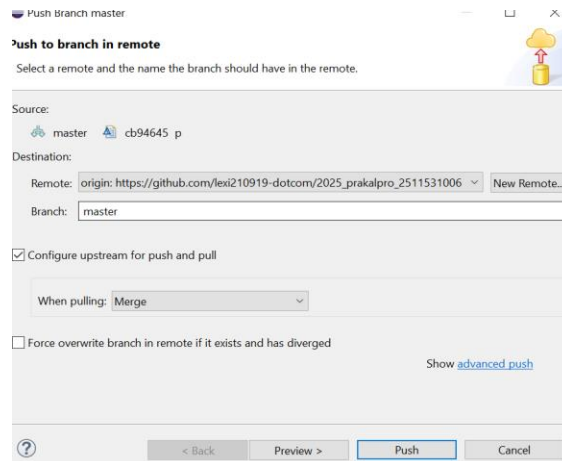
d. Jika tampilan sudah hilang, maka item berhasil di commit.



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



f. 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”.



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

lexi210919-dotcom / 2025_prakalpro_2511531006

Type ↵ to search

+

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

2025_prakalpro_2511531006

Public

Pin

Watch 0

Fork 0

Star 0

master 1 Branch 0 Tags

Go to file ↵

Add file

<> Code

About

user and user tugas 6b1234d · yesterday 3 Commits

prakalpro_2025_2511531006 tugas yesterday

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

Repositori praktikum Algoritma kelas B

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

Activate Windows

Go to Settings to activate Windows

No packages published

Unpublish your first package

BAB III

PENUTUP

3.1 Kesimpulan

Dari hasil praktikum Algoritma dan Pemrograman pada pekan ini, dapat disimpulkan bahwa pemahaman terhadap struktur kontrol seperti perulangan (looping) dan kondisional sangat penting dalam membangun logika program yang efisien dan dinamis. Melalui berbagai latihan program seperti simulasi lempar dadu, perhitungan penjumlahan, hingga validasi input password, saya belajar bagaimana menerapkan konsep do-while, while, dan if-else secara tepat untuk mengatur alur program. Praktikum ini juga memberikan pengalaman langsung dalam menggunakan kelas Scanner untuk membaca input pengguna dan kelas Random untuk menghasilkan nilai acak. Dengan demikian, praktikum ini telah membantu memperkuat pemahaman saya terhadap logika algoritma, sintaks Java, serta penerapan konsep dasar pemrograman yang terstruktur dan efisien.

B.Saran

Untuk praktikum selanjutnya, sebaiknya diberikan lebih banyak variasi soal yang menantang agar mahasiswa dapat mengasah kemampuan logika dan analisis dalam memecahkan masalah pemrograman yang lebih kompleks. Selain itu, penjelasan mengenai kesalahan umum (error handling) dan praktik penggunaan GitHub juga dapat lebih diperluas agar mahasiswa terbiasa dengan proses kerja pengembang perangkat lunak secara profesional. Saya juga akan terus berlatih menulis kode yang bersih, terstruktur, dan mudah dipahami, karena keterampilan tersebut sangat penting dalam dunia kerja di bidang teknologi informasi.

DAFTAR PUSTAKA

Dony, R. (2020). *Algoritma dan Pemrograman Dasar dengan Java*. Yogyakarta: Deepublish.

Munir, Rinaldi. (2022). *Algoritma dan Pemrograman dalam Bahasa Java*. Bandung:
Informatika.