LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PERULANGAN *WHILE* DAN *DO WHILE* DALAM PEMROGRAMAN JAVA

Disusun Oleh:

Haliya Isma Husna Putri Ahmadi
2511532002

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

Asisten Praktikum:
Muhammad Zaky Al Hafiz

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji dan Syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas Rahmat dan karunia-Nya, penulis dapat menyelesaikan lapora praktikum Algoritma dan Pemrograman pekan 6 dengan judul "Perulangan *While* dan *Do While* dalam Pemrograman Java" tepat pada waktunya.

Penyusunan laporan ini bertujuan untuk memenuhi salah satu tugas dalam mata kuliah Praktikum Algoritma dan Pemrograman sekaligus sebagai bentuk pembelajaran bagi penulis mengenai implementasi konsep perulanganaan dalam pemrograman *Java,* khususnya penggunaan perulangan *while* dan *do-while*. Melalui praktikum ini, penulis dapat mempelajari konsep dan logika perulangan untuk menjalankan suatu program secara efisien, berdasarkan kondisi tertentu .

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

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

Padang, 4 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR ..... **ii**
DAFTAR ISI ..... **iii**
BAB I PENDAHULUAN **4**
1.1 Latar Belakang **4**
1.2 Tujuan **4**
1.3 Manfaat **4**
BAB II PEMBAHASAN **5**
2.1 Dasar Teori **5**
2.1.1 Perulangan *While* **5**
2.1.2 Perulangan *Do-While* **5**
2.2 Langkah Kerja **6**
2.2.1 Pembuatan *Package* dan *Class* **6**
2.2.2 Program Pertama: perulanganWhile1_2511532002 ..... **8**
2.2.3 Program Kedua: Lempardadu **9**
2.2.4 Program Ketiga: GamePenjumlahan **10**
2.2.5 Program Keempat: SentinelLoop **12**
2.2.6 Program Kelima: doWhile1 ..... **13**
2.2.7 *Upload* ke GitHub **15**
BAB III KESIMPULAN **18**
3.1 Kesimpulan **18**
3.2 Saran **18**
DAFTAR PUSTAKA **19**

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam mempelajari bahasa pemrograman, pemahaman konsep terhadap perulangan (*looping*) merupakan salah satu konsep dasar yang sangat penting. Dengan perulangan, kita dapat menjalankan suatu intruksi secara berulang dengan kondisi tertentu tanpa harus menuliskan perintah berulang kali. Adanya struktur perulangan membuat penulisan kode menjadi lebih efisien dan mudah dibaca.

Salah satu jenis perulangan yang umum digunakan dalam bahasa pemrograman *Java* adalah perulangan *while* dan *do-while* yang bekerja berdasarkan logika. Struktur *while* mengecek kondisi terlebih dahulu sebelum menjalankan perintah. Sedangkan *do-while*, menjalankan perintah minimal satu kali sebelum memeriksa kondisi.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1.2.1 Memahami konsep dasar, logika dan struktur *syntax* perulangan *while* dan *do-while* dalam bahasa pemrogram *Java*.

1.2.2 Mampu mengimplemantasikan perulangan *while* dan *do-while* untuk berbagai kondisi, sesuai dengan kasus yg dibutuhkan..

1.2.3 Membangun logika pemrograman dalam menyusun programan dan algoritma perulangan.

1.3 Manfaat

Manfaat yang diharapkan dari pelaksanaan praktikum ini antara lain:

1.3.1 Memberikan pemahaman mengenai konsep dasar perulangan, khususnya perulangan *while* dan *do-while* dalam bahasa *Java*.

1.3.2 Memberikan pengalaman langsung dalam mengimplementasikan konsep perulangan untuk suatu kasus *real*.

BAB II
PEMBAHASAN

2.1 Dasar Teori

Perulangan (*looping)* adalah proses menjalakan satu atau beberapa perintah secara berulang hingga mencapai kondisi tertentu terpenuhi. Perulangan ini sangat berguna agara *user* tidak perlu menulis kode perintah yang sama berulang-ulang secara manual.

Dalam *Java,* perulangan dapat dibagii menjadi dua jenis utama:

1. *Counted Loop*

   Perulangan yang jumlah pengulangannya sudah diketahui atau terhitung sebelumnya. Contoh nya perulangan *for.*

2. *Uncounted Loop*

   Perulangan yang jumlah perulangannya tidak tentu atau berdasarkan kondisi tertentu selama proses *looping*. Contoh *while* dan *do-while.*

Pada pekan ini akan lebih difokuskan dengan perulangan *uncounted while* dan *do-while.* Perulangan yang digunakan untuk mempersingkat kode saat harus mengulang proses yang sama berkali-kali, tanpa mengetahui jumlah perulangan yang akan dilakukan.

2.1.1 Perulangan *While*

Perulangan *while* adalah struktur perulangan di mana kondisi dicek sebelum blok perulangan dieksekusi. Jika kondisi bernilai *true*, maka blok dieksekusi, jika *false*, maka program berakhir.

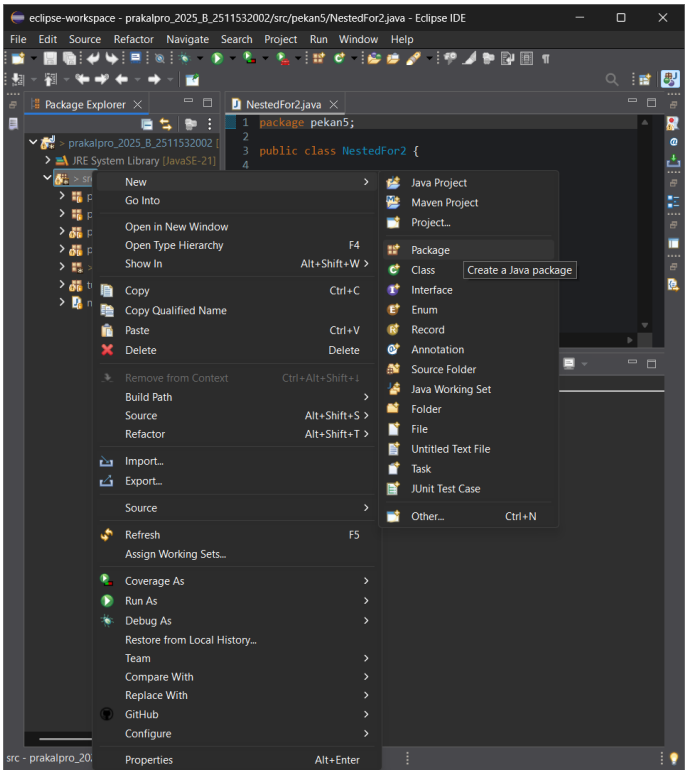
2.1.2 Perulangan *Do-While*

Perulangan *do-while* adalah struktur perulangan di mana mengeksekusi blok perintah minimal satu kali, sebelum kondisi diperiksa. Artinya, program akan menjalankan perulangan terlebih dahulu, kemudian mengevaluasi kondisi yang ada. Jika kondisi masih bernilai *true*, maka perulangan terus dijalankan. Namun jika kondisi bernilai *false*, maka program berakhir

2.2 Langkah Kerja

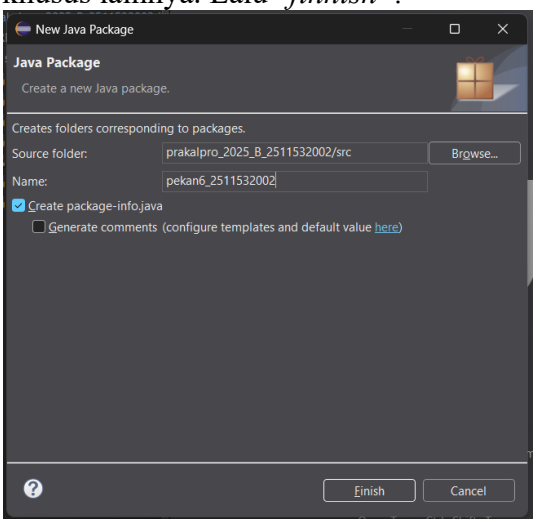
2.2.1 Pembuatan *Package* dan *Class*

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



2. 1 Pembuatan Package

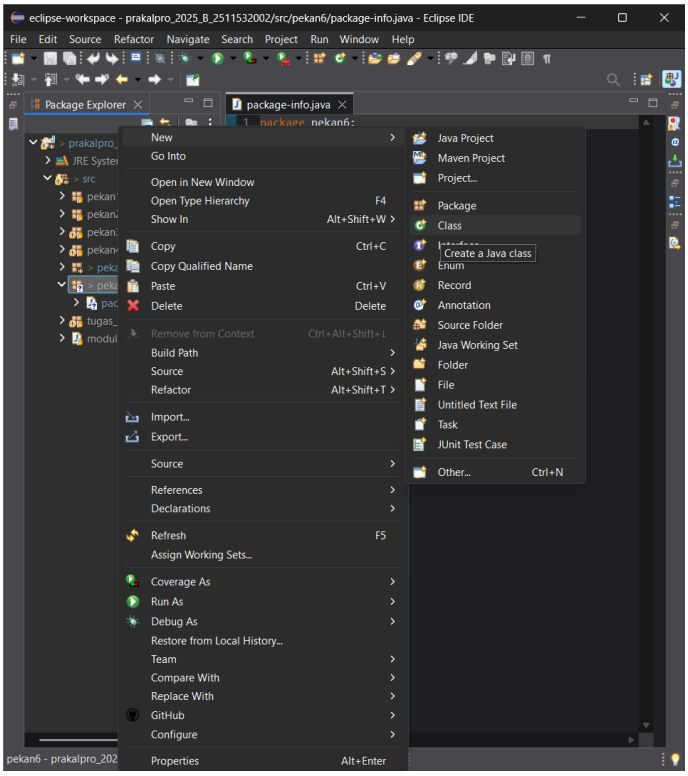
2. Setelah itu akan muncul menu "*Java Package*", buat nama *package* dengan ketentuan tanpa *capslock, space*, dan karakter khusus lainnya. Lalu "*finnish*".



2. 2 Pembuatan Package

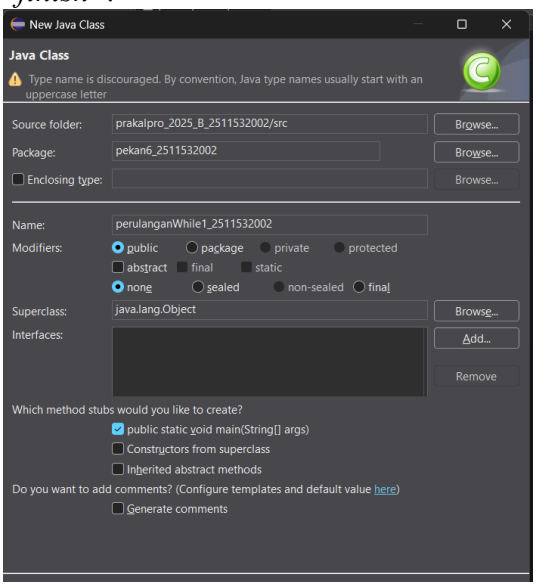
Maka *user* telah berhasil membuat *package.*

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



2. 3 Pembuatan Class

4. Setelah itu pada menu "*Java Class"* buat nama *class*, dengan ketentuan nama harus *capslock* diawal kalimat dan tanpa spasi. Lalu pilih "*Public static void main (String[] grs)"* agar tidak perlu membuat *methode main()* secara manual. Selanjutnya "*finish".*

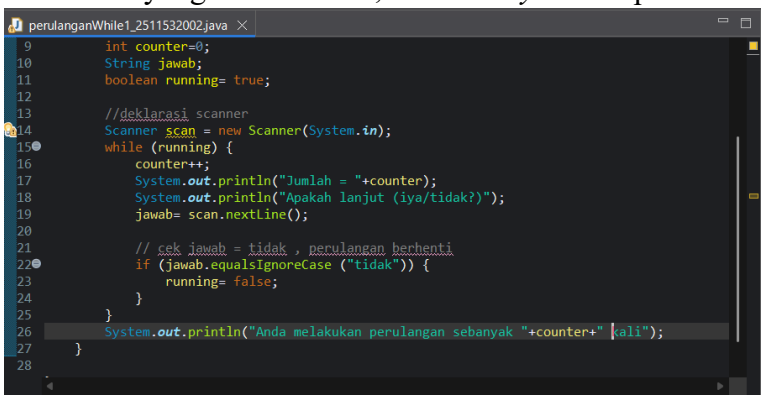


2. 4 Pebuatan Class

2.2.2 Program Pertama: perulanganWhile1_2511532002

Pada *class* ini mengenai penerapan *while* sederhana. Dengan tujuan mengeksekusi sebuah blok perintah tanpa mengetahui junlah perulanggannya.

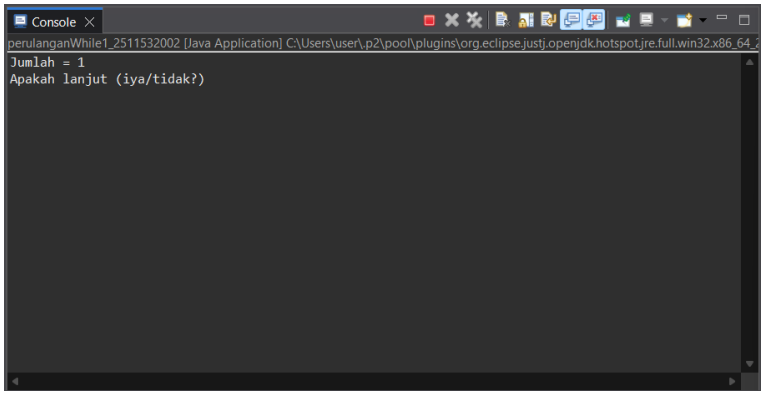
1. Pada *class* yang sudah dibuat, masukan *syintax* seperti berikut.



2. 5 Syntax Program perulanganWhile1_2511532002

2. Setelah itu jalan kan program dengan klik tomol *Run* di pojok kiri atas pada *toolbar.*

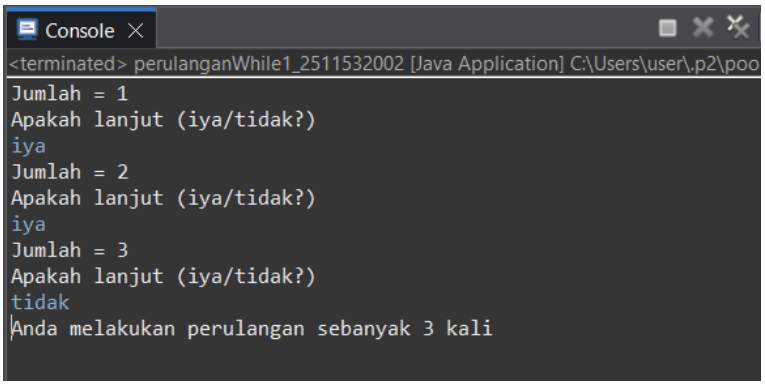
   Maka pada menu *console* akan muncul hasil program seperti berikut, disini *user* dapat mencoba program berulang kali sesuai dengan *input* yang diberikan, dengan memasukan nilai "ya/tidak".



2. 6 Running Program perulanganWhile1_2511532002

   Dimana jika *user* mengetik "iya" (selain "tidak") maka program akan terus berjalan. Jika *user* mengtik "tidak/Tidak" maka program akan berhenti.

3. Disini saya akan menguji, dengan mencoba mengeti “iya” sebanyak 2x dan “tidak” ti perulangan ke-3. Maka nantinya akan menampilakn hasil program seperti berikut.

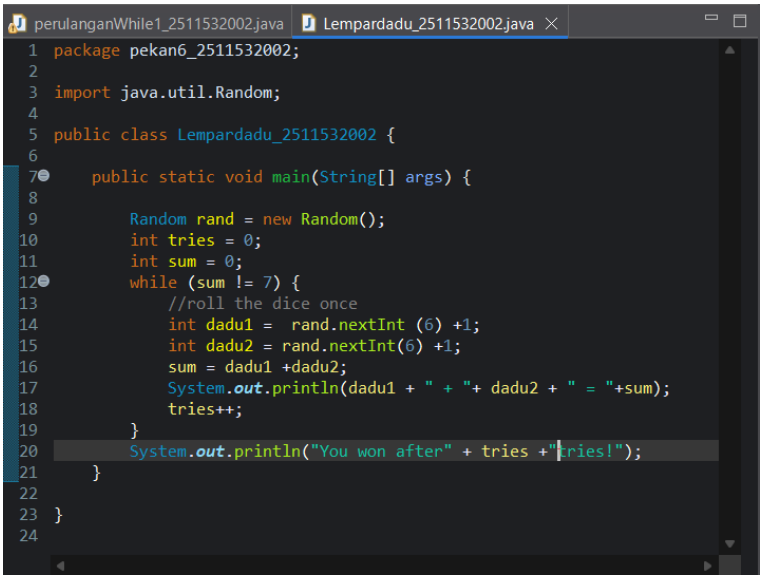


2. 7 Hasil Uji Coba Program perulanganWhile1_2511532002

Hasil menunjukan total perulangan 3x (Sesuai).

2.2.3 Program Kedua: Lempardadu

1. Buat kembali *class* baru pada *package* pekan6 dengan nama kelas “*Lempardadu_2511532002*”. Selanjutnya masukan *syntax* seperti berikut.

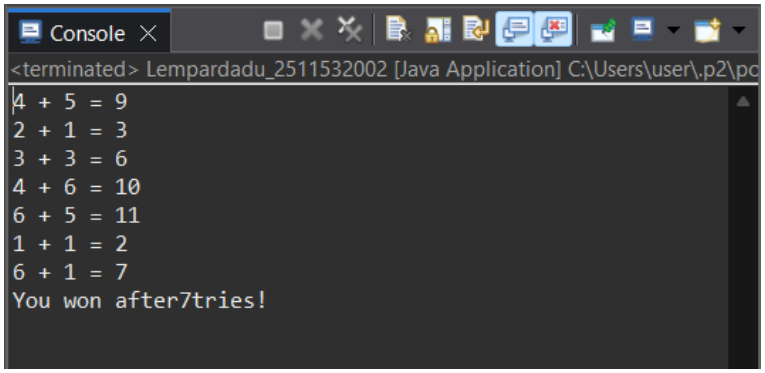


2. 8 Syntax Program Lempardadu_2511532002

Program ini merupakan simulasi permainan lempar dadu, menggunakan perulangan *while* dengan variabel *random* yang berhenti saat total nilai kedua dadu sama dengan 7.

4. Lalu jalankan program dengan klik tomol *Run* di pojok kiri atas pada *toolbar.*

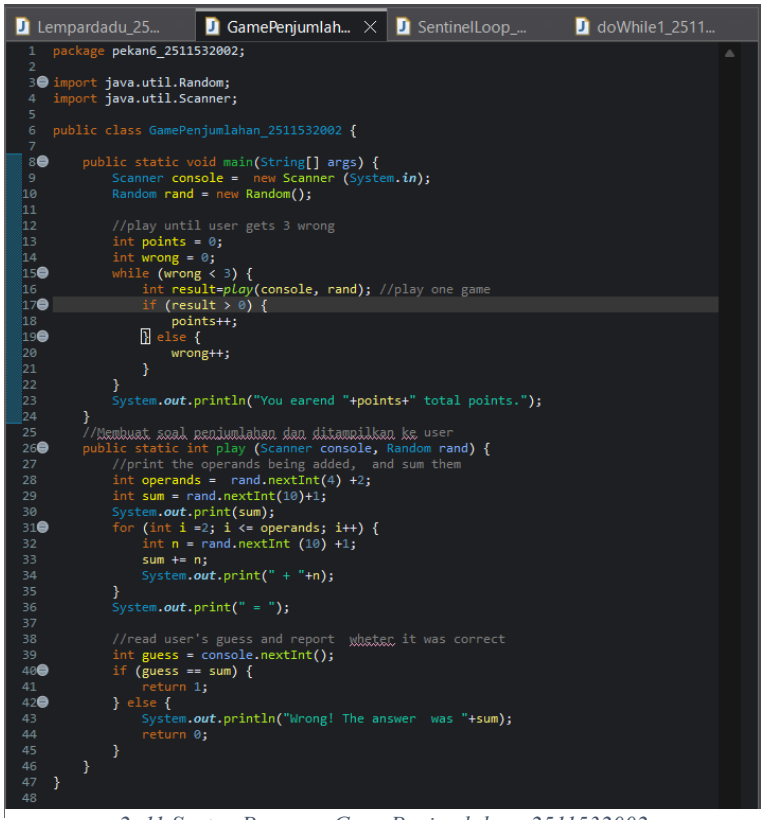
   Maka pada menu *console* akan muncul hasil program seperti berikut, dimana akan muncul total putaran/ perulangan "dadu" secara random yang kita dapatkan.



2. 10 Hasil Program Lempardadu_2511532002

2.2.4 Program Ketiga: GamePenjumlahan

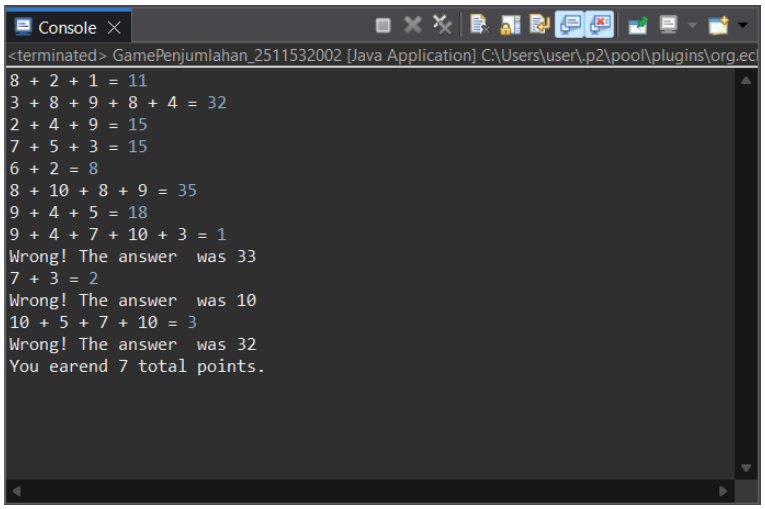
1. Buat kembali *class* baru dengan nama kelas "GamePenjumlahan_2511532002". Selanjutnya masukan *syntax* seperti berikut.



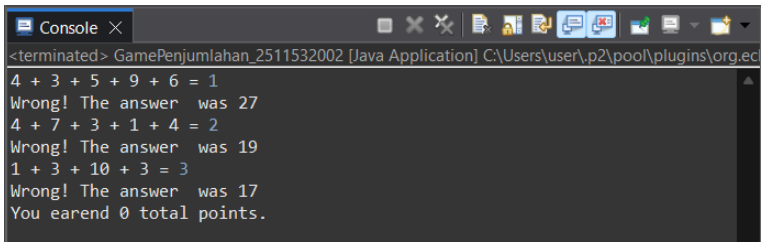
2. 11 Syntax Program GamePenjumlahan_2511532002

Program ini merupakan simulasi permainan penjumlahan sederhana menggunakan struktur perulangan *while*. Dimana *user* akan menjawab soal penjumlahan dengan bilangan *random,* yang akan berhenti jika *user* gagal menjawab sebanyak 3x.

2. Lalu jalankan program, dengan klik tomol *Run* di pojok kiri atas pada *toolbar.*
   Maka pada menu *console* akan muncul hasil program seperti berikut, dimana akan menampilkan soal penjumlahan acak. Setiap jawaban benar akan menambah poin, dan permainan akan berakhir setelah 3 kali kesalahan.



2. 12 Hasil Uji Coba Program GamePenjumlahan_2511532002



2. 13 Hasil Uji Coba Program GamePenjumlahan_2511532002

2.2.5 Program Keempat: SentinelLoop

1. Buat kembali *class* baru dengan nama kelas "SentinelLoop_2511532002". Selanjutnya masukan *syntax* seperti berikut.

```java
package pekan6_2511532002;

import java.util.Scanner;

public class SentinelLoop_2511532002 {

    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        int sum =0;
        int number =12; //"dummy value", anything but 0

        while (number != 0) {
            System.out.println("Masukkan angka (0 untuk keluar): ");
            number = console.nextInt();
            sum=sum +number;
        }
        System.out.println("Totalnya adalah "+sum);
    }
}
```

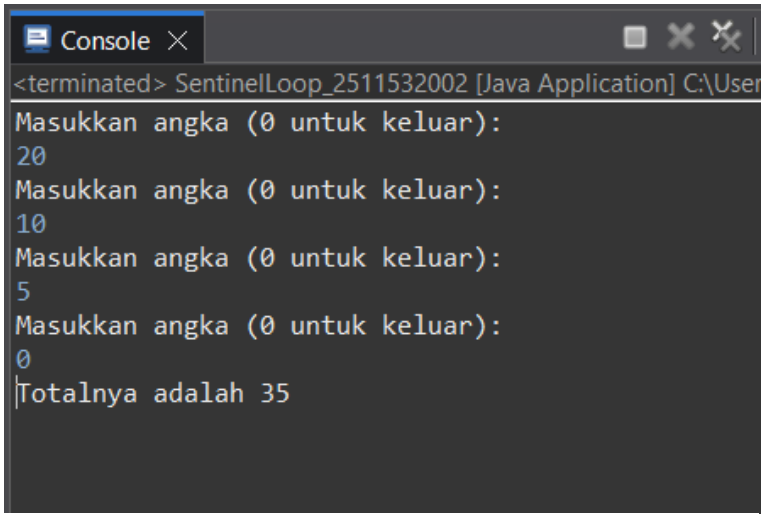
2. 14 Syntax Program SentinelLoop_2511532002

Program ini digunakan untuk menghitung jumlah (total) dari sekumpulan angka yang di *input* oleh pengguna hingga mengetikkan 0, yang berfungsi sebagai nilai sentinel untuk menghentikan perulangan.

3. Lalu jalankan program, dengan klik tomol *Run* di pojok kiri atas pada *toolbar.* Maka pada menu *console* akan muncul hasil program, setelah itu masukan nilai *input* yang diinginkan hingga 0 untuk menghentingkan. Sehingga menghasilkan program seperti berikut.

```
Console
<terminated> SentinelLoop_2511532002 [Java Application] C:\Use
Masukkan angka (0 untuk keluar):
5
Masukkan angka (0 untuk keluar):
4
Masukkan angka (0 untuk keluar):
3
Masukkan angka (0 untuk keluar):
2
Masukkan angka (0 untuk keluar):
1
Masukkan angka (0 untuk keluar):
0
Totalnya adalah 15
```

2. 15 Hasil Program SentinelLoop_2511532002

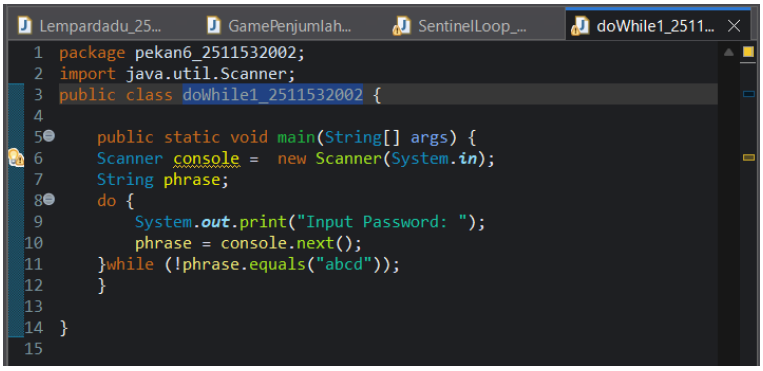


2. 16 Hasil Program SentinelLoop_2511532002

2.2.6 Program Kelima: doWhile1

Program berikut menerapkan struktur perulangan *do-while*, di mana isi perulangan akan dijalankan minimal, sebelum kondisi diperiksa.

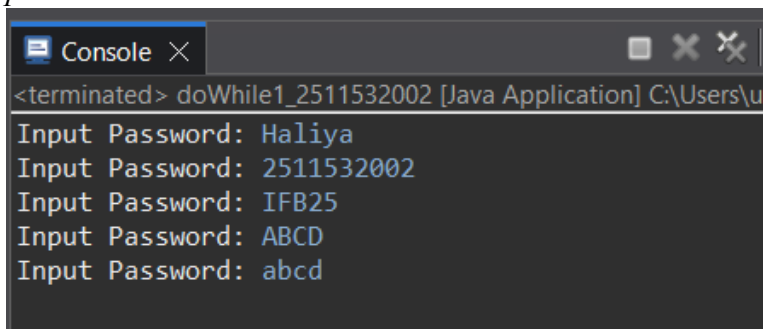
2. Buat kembali *class* baru dengan nama kelas "doWhile1_2511532002".Selanjutnya masukan *syntax* seperti berikut.



2. 17Syntax Program doWhile1_2511532002

Program ini berfungsi untuk meminta *input password* dari *user* berulang kali sampai *password* yang dimasukkan sesuai ("abcd").

3. Lalu jalankan program, dengan klik tomol *Run* di pojok kiri atas pada *toolbar*. Maka pada menu *console* akan muncul hasil program, setelah itu *user* dapat mencoba memasukan nilai *password*.

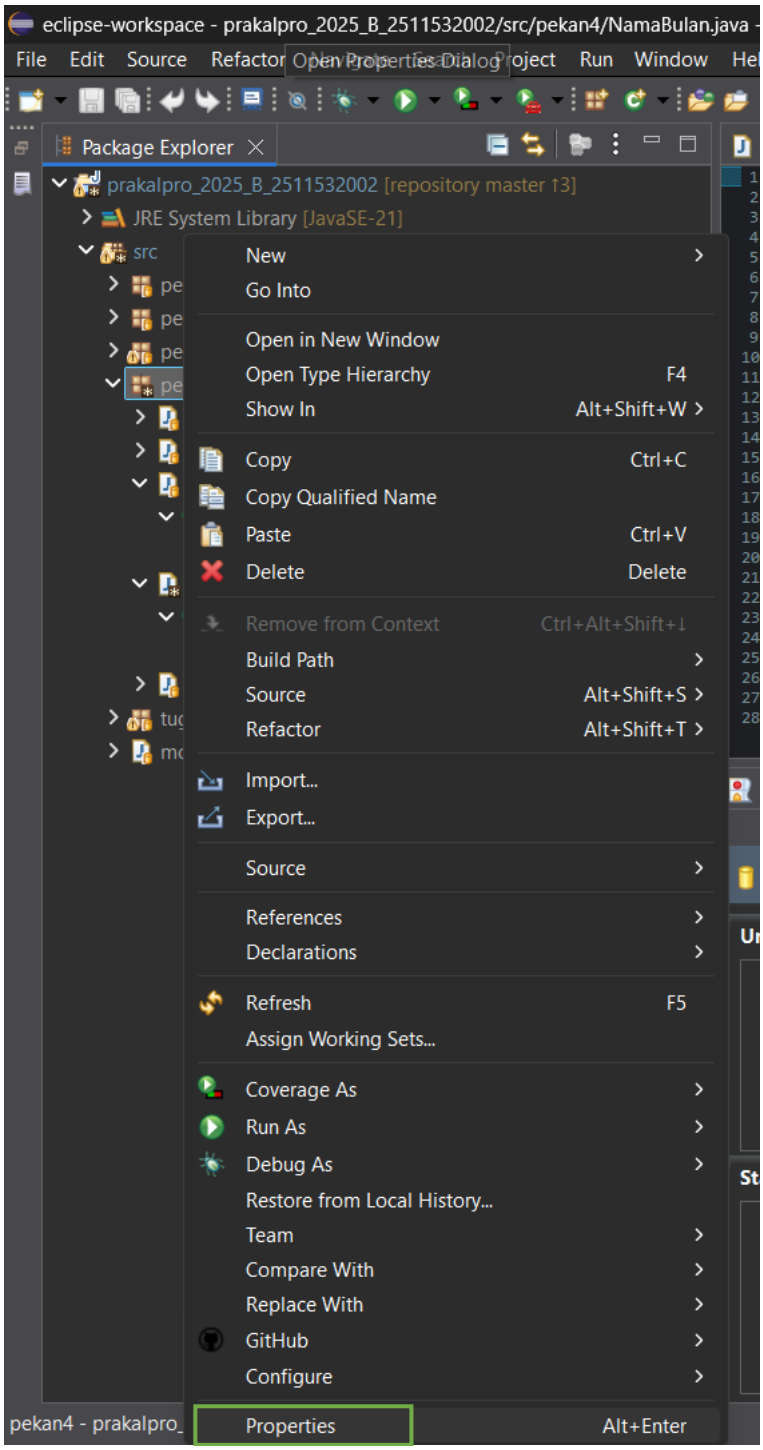


2. 18 Hasil Program doWhile1_2511532002

Dari hasil tersebut terlihat bahwa program akan terus meminta *input password* sampai pengguna memasukkan "*abcd*".

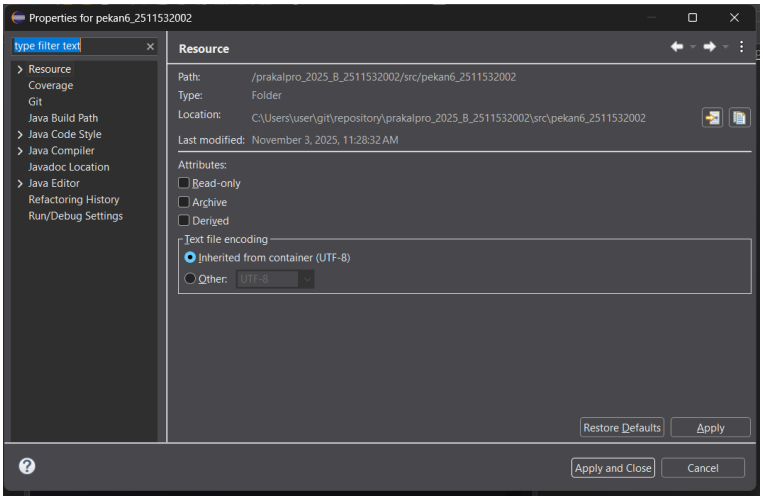
2.2.7 *Upload* ke GitHub

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



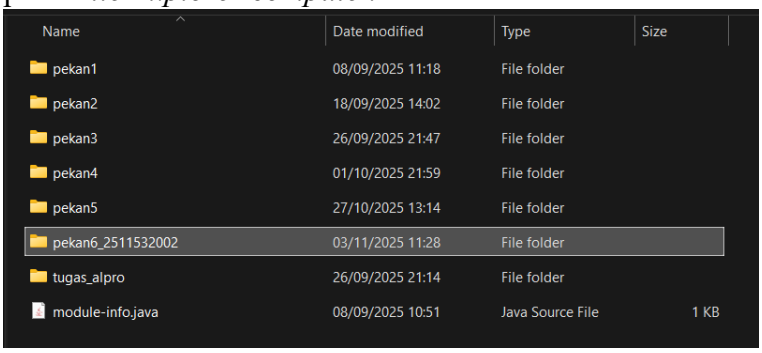
2. 19 Package Properties

2. Maka akan muncul *pop up menu "Properteis for pekan6_2511532002"*. Lalu klik *icon "Show in System Folder"*.



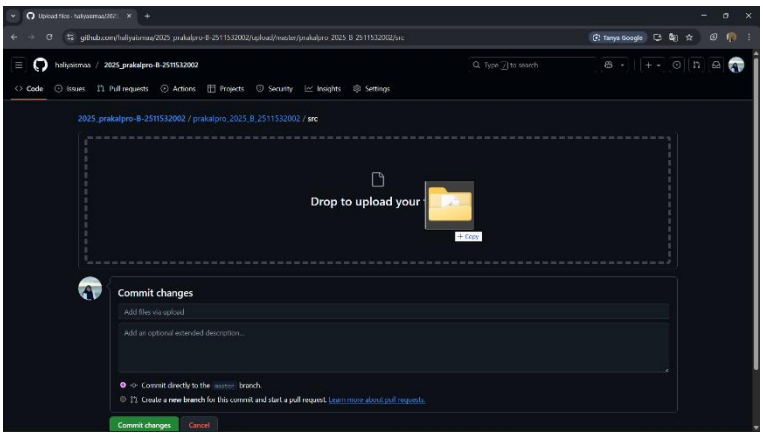
2. 20 Properteis for pekan6_2511532002

3. Maka akan menunjukan lokasi *file "pekan6_2511532002"* pada *File Explorer computer.*



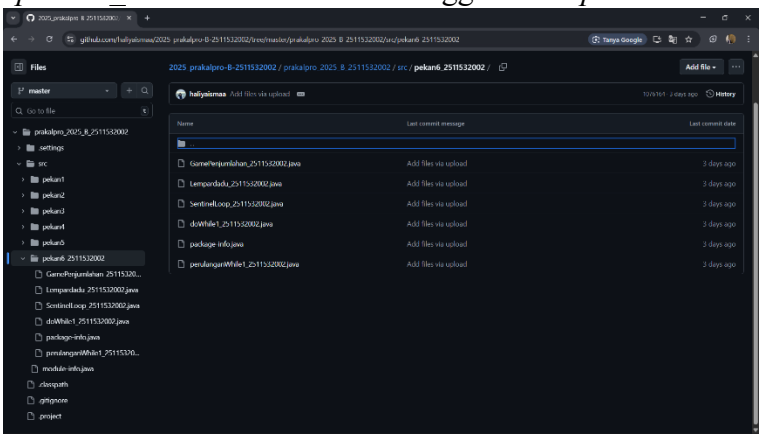
2. 21 Lokasi File pekan6_2511532002 pada File Explorer

4. Setelah itu *drag and drop file "pekan6_2511532002"* ke GitHub. Lalu tambahkan keterangan jika perlu. Setelah itu klik *upload.*



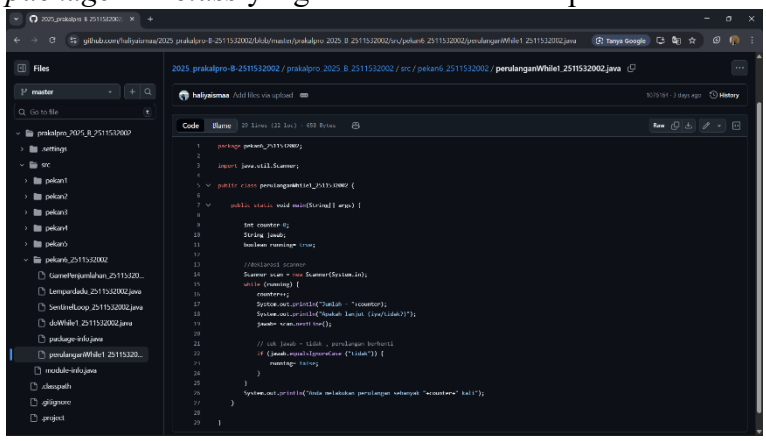
2. 22 Upload Package pekan6_2511532002

5. Maka dapat dilihat pada *repository GitHub* bahwa file "*pekan6_2511532002*" telah terunggah via *Upload.*



2. 23 Upload Package pekan6_2511532002

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



2. 24 Package pekan6_2511532002 di GitHub

BAB III
KESIMPULAN

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan dengan topik *Uncounted loop* atau "Perulangan *while* dan *do-while*". Dapat disimpulkan bahwa:

1. Perulangan (*looping*) digunakan untuk mengeksekusi blok kode secara berulang selama kondisi tertentu terpenuhi, sehingga dapat menghemat waktu dan baris kode dalam pemrograman..
2. *Uncounted loop* adalah perulangan yang jumlaa perulangannya tidak tentu atau berdasarkan kondisi tertentu selama proses *looping*. Contoh *while* dan *do-while*. Dimana *while* kondisi diperiksa terlebih dahulu sebelum perulangan dijalankan, sedangkan *do-while* kode dijalankan dulu, baru kemudian kondisi diperiksa di akhir.
3. Sehingga jika syarat perulangannya tidak berkaitan dengan hasil hitung pada blok kode yang diulang, maka *user* dapat menggunakan *while*. Tetapi, jika syarat perulangan berkaitan dengan hasil perhitungan di blok kode yang diulang, maka *do- while*.

3.2 Saran

Saran untuk pelaksanaan praktikum kedepannya adalah diharapkan agar mahasiswa lebih teliti dalam memahami *syntax* agar meminimalisir kesalahan penulisan kode. Terutama saat menggunakan perulangan *while*, pastikan kondisi pengontrol dan nilai variabel benar, agar program tidak mengalami *infinite loop*.

DAFTAR PUSTAKA

[1] Oracle, “The while Statement (Java Tutorials > Learning the Java Language Basics),” [Daring]. Tersedia pada: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html [Diakses: 06-Nov-2025].

[2] Petani Kode, “Belajar Java: Perulangan (Looping) - For, While, dan Do While,” [Daring]. Tersedia pada: https://www.petanikode.com/java-perulangan/ [Diakses: 06-Nov-2025].

[3] Petani Kode, “Perbedaan Perulangan While dan Do While,” [Daring]. Tersedia pada: https://www.petanikode.com/perbedaan-perulangan-while-dan-do-while/ [Diakses: 06-Nov-2025].