

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
PERULANGAN
PEKAN 5



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 dentan materi perulangan yang dibuat pada pekan 5 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, 31 Oktober 2025

Lexi Mulia Yunaspi

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Praktikum.....	22
BAB III PENUTUP.....	23
3.1 Kesimpulan.....	24
DAFTAR PUSTAKA.....	25

BAB I

PENDAHULUAN

1.1.Latar Belakang

Dalam pemrograman, perulangan (looping) merupakan salah satu konsep dasar yang sangat penting untuk mengotomatisasi tugas yang berulang. Tanpa perulangan, programmer harus menulis instruksi yang sama berkali-kali, sehingga kode menjadi panjang, tidak efisien, dan sulit dipelihara.

Melalui praktikum ini, mahasiswa diperkenalkan pada berbagai jenis struktur perulangan seperti `for`, `while`, dan `do-while`, serta penerapannya dalam menyelesaikan berbagai masalah komputasi. Pemahaman konsep perulangan akan mempermudah pengembangan logika program dan menjadi dasar bagi konsep algoritma yang lebih kompleks seperti pengolahan data dan rekursi.

1.2. Tujuan Praktikum

1. Memahami konsep dasar dan logika kerja dari struktur perulangan.
2. Mampu menggunakan berbagai jenis perulangan (**for**, **while**, **do-while**) secara tepat sesuai kebutuhan program.
3. Melatih kemampuan menyusun algoritma yang melibatkan proses berulang.
4. Meningkatkan keterampilan dalam menulis kode yang efisien dan mudah dipahami.
5. Mampu mengimplementasikan perulangan dalam studi kasus sederhana, seperti perhitungan deret angka, pencetakan pola, atau pemrosesan data.

1.3. Manfaat Praktikum

1. Mahasiswa memahami pentingnya efisiensi dalam menulis program.
2. Mahasiswa dapat berpikir logis dan sistematis dalam menyusun solusi berbasis algoritma.

3. Memberikan dasar kuat untuk mempelajari struktur kontrol yang lebih kompleks seperti nested loop dan loop with condition.
4. Menumbuhkan keterampilan analitis untuk mengidentifikasi kapan dan bagaimana prulangan digunakan secara optimal.
5. Menjadi bekal penting untuk praktikum atau proyek pemrograman di tingkat lanjut.

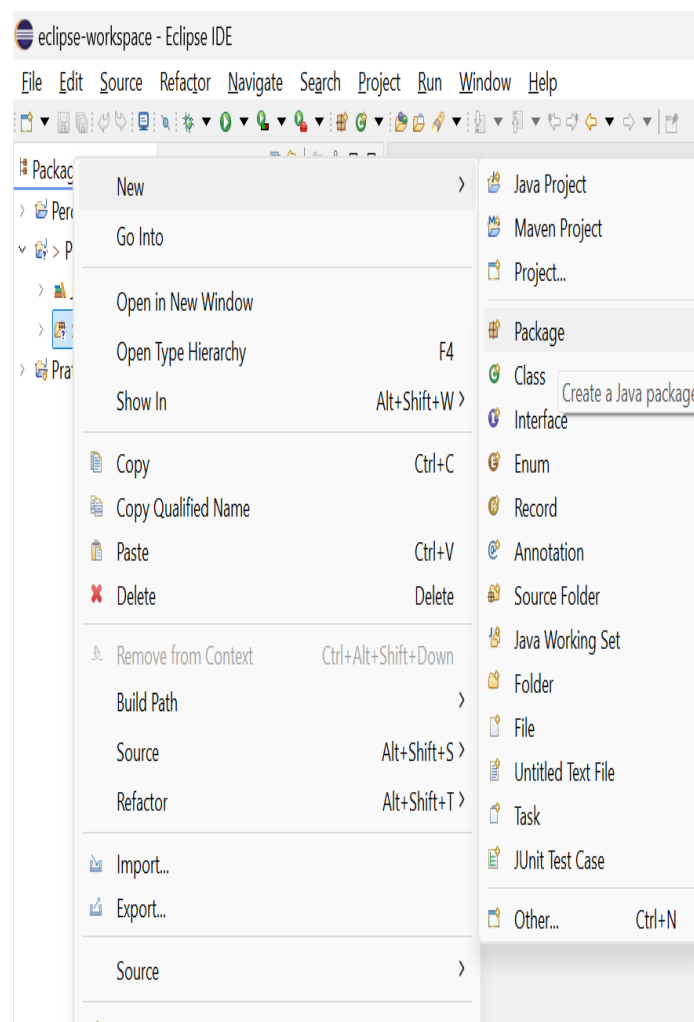
BAB II

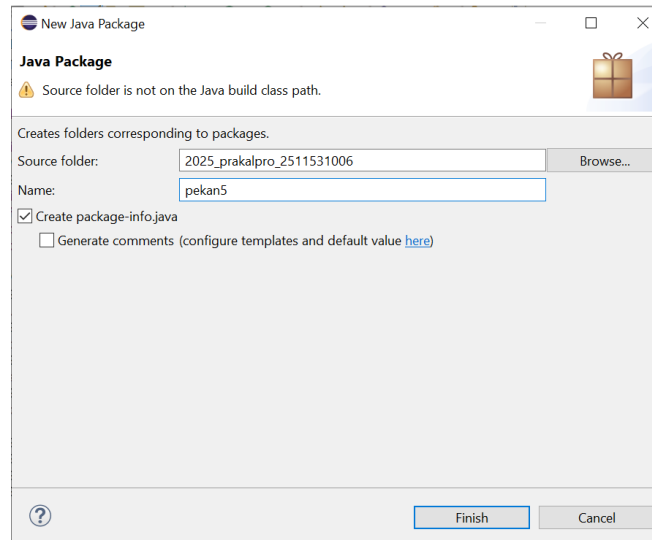
PEMBAHASAN

2.1.Langkah Kerja Praktikum

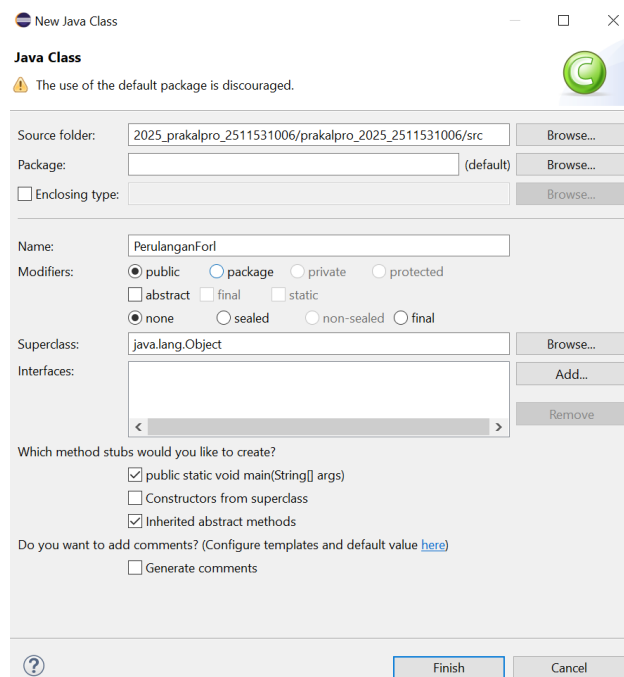
1.Program Pertama

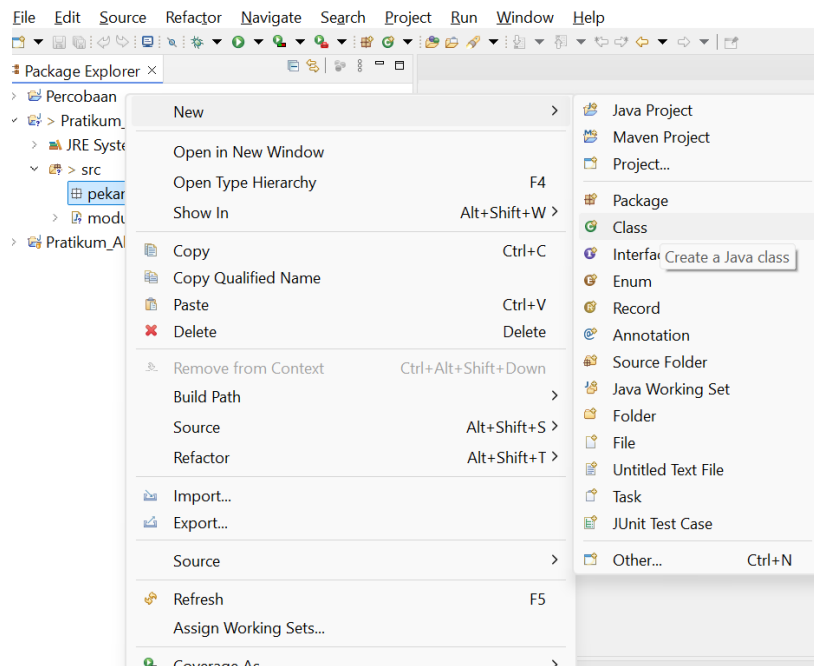
a.Buatlah package terlebih dahulu dengan mengklik kanan di folder src. Setelah itu beri nama pada package tanpa huruf kapital, karakter khusus serta tanpa “space.





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. Potongan kode tersebut adalah bagian awal dari program Java yang mendefinisikan sebuah kelas bernama `PerulanganFor1`. Di dalamnya terdapat metode utama `main(String[] args)`, yang merupakan titik awal eksekusi program Java. Pada bagian ini biasanya akan ditulis logika utama program, seperti perulangan, percabangan, atau operasi lainnya yang ingin dijalankan ketika program dieksekusi. Struktur ini merupakan kerangka dasar yang wajib ada agar program Java dapat berjalan.

```
public class PerulanganFor1 {
    public static void main(String[] args) {
```

d. Potongan kode tersebut merupakan struktur perulangan for dalam bahasa Java yang digunakan untuk mencetak angka dari 1 hingga 10. Variabel `i` diinisialisasi dengan nilai 1, kemudian selama `i` kurang dari atau sama dengan 10, perulangan akan terus berjalan. Setiap kali perulangan dijalankan, nilai `i` akan ditingkatkan satu per satu dengan `i++`. Di dalam blok perulangan, perintah `System.out.println(i);` mencetak nilai `i` ke layar di setiap baris baru, sehingga hasil akhirnya adalah daftar angka dari 1 sampai 10 secara berurutan.


```

        for (int i = 1; i <= 10; i++) {
            System.out.println (i);
        }
    }
}

```

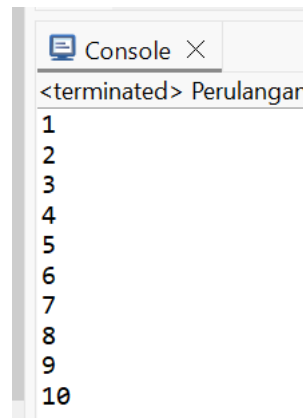
e.Jadinya akan seperti ini

```

public class PerulanganFor1 {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.println (i);
        }
    }
}

```

f.Hasil run program



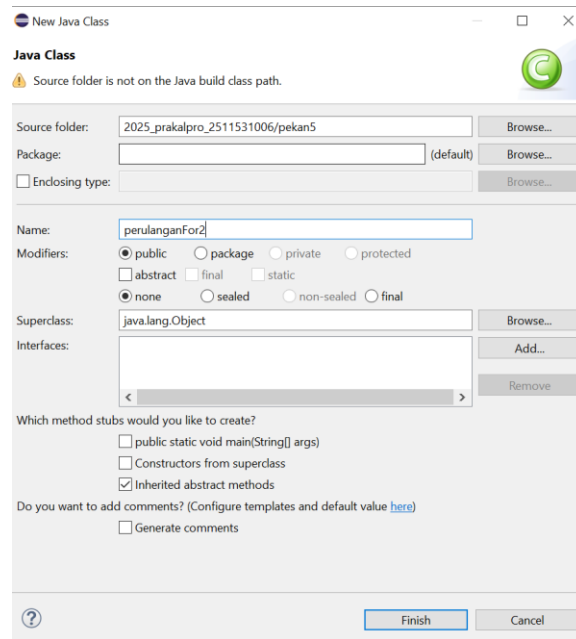
```

<terminated> Perulangan
1
2
3
4
5
6
7
8
9
10

```

2.Program Dua

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



b. Maka selanjutnya tekan finish dan keluar seperti ini

```
public class perulanganFor2 {  
    public static void main(String[] args) {
```

c. Potongan kode tersebut menampilkan perulangan for dalam Java yang mencetak angka dari 1 sampai 10 dalam satu baris. Variabel `i` diinisialisasi dengan nilai 1 dan akan terus bertambah satu per satu hingga mencapai 10. Di dalam perulangan, perintah `System.out.print(i + " ");` mencetak nilai `i` diikuti dengan spasi tanpa berpindah ke baris baru.

```
        for (int i = 1; i <= 10; i++) {  
            System.out.print(i + " ");  
        }  
    }  
}
```

d. Programnya akan jadi seperti ini

```

public class perulanganFor2 {

    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            System.out.print(i + " ");
        }
    }
}

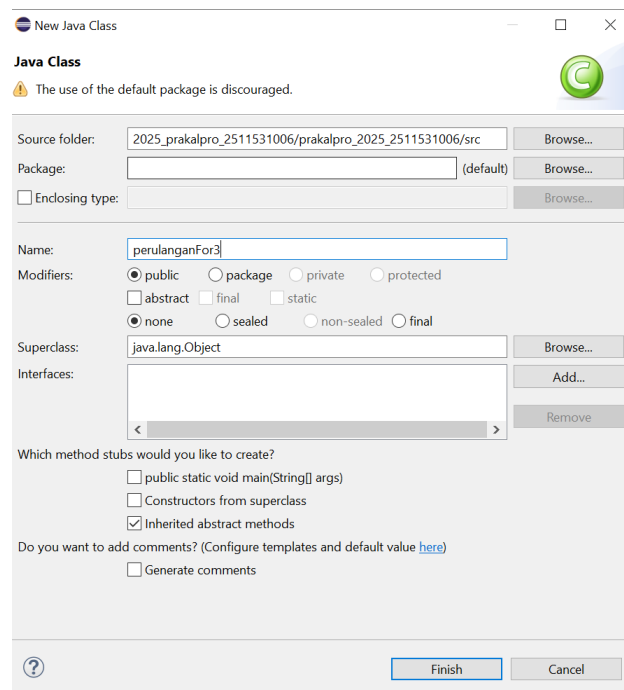
```

e. Hasil run program

1 2 3 4 5 6 7 8 9 10

3. Program 3

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



b. Maka akan muncul seperti ini jika sudah tekan finish.

```
public class perulanganFor3 {  
    public static void main(String[] args) {
```

c. Potongan kode tersebut menampilkan perulangan for yang mencetak angka dari 1 hingga 10 dengan tanda “+” di antara setiap angka, kecuali setelah angka terakhir. Variabel jumlah diinisialisasi dengan nilai 0 dan digunakan untuk menampung hasil penjumlahan seluruh angka dalam perulangan. Setiap kali perulangan berjalan, nilai i ditambahkan ke jumlah, lalu program mencetak nilai i. Kondisi if ($i < 10$) memastikan tanda “+” hanya dicetak di antara angka, bukan di akhir.

```
int jumlah=0;  
for (int i=1;i<=10;i++) {  
    System.out.print (i);  
    jumlah= jumlah+i;  
    if (i<10) {  
        System.out.print(" + ");  
    }  
}
```

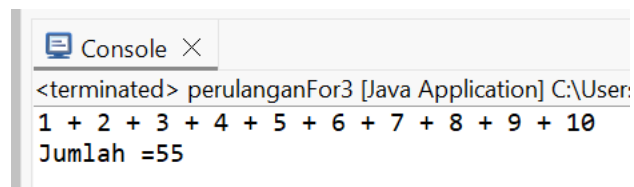
d. Potongan kode tersebut digunakan untuk menampilkan hasil akhir dari program perulangan sebelumnya. Baris `System.out.println();` berfungsi untuk membuat baris baru setelah deretan angka selesai dicetak agar hasil berikutnya ditampilkan di bawahnya. Selanjutnya, `System.out.println("Jumlah =" + jumlah);` mencetak teks “Jumlah =” diikuti dengan nilai total dari variabel jumlah, yang merupakan hasil penjumlahan semua angka dalam perulangan.

```
    System.out.println();  
    System.out.println("Jumlah =" + jumlah) ;  
}  
}
```

e. Maka kode programnya akan seperti ini

```
public class perulanganFor3 {  
    public static void main(String[] args) {  
        int jumlah=0;  
        for (int i=1;i<=10;i++) {  
            System.out.print (i);  
            jumlah= jumlah+i;  
            if (i<10) {  
                System.out.print(" + ");  
            }  
        }  
        System.out.println();  
        System.out.println("Jumlah =" + jumlah) ;  
    }  
}
```

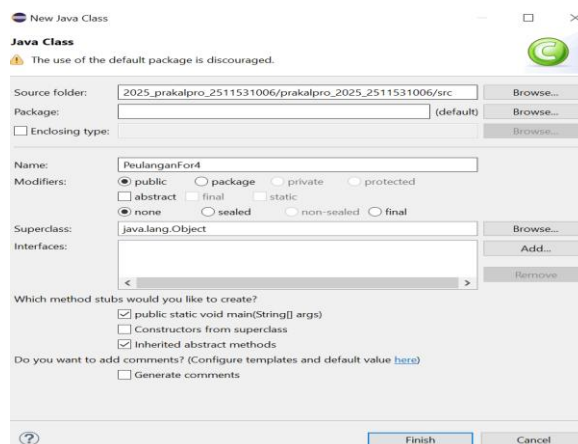
f. Hasil run kode program



The screenshot shows a console window titled "Console" with the following output:
<terminated> perulanganFor3 [Java Application] C:\User:
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
Jumlah =55

4. Program 4

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



b. Maka akan muncul seperti ini jika sudah tekan finish.

```
import java.util.Scanner;

public class PeulanganFor4 {

    public static void main(String[] args) {
```

c. Potongan kode Java pada gambar tersebut berfungsi untuk mendeklarasikan dua variabel bertipe data int (bilangan bulat), yaitu jumlah dan batas. Variabel jumlah diinisialisasi dengan nilai awal 0, yang biasanya digunakan untuk menyimpan hasil penjumlahan atau perhitungan sementara. Sedangkan variabel batas hanya dideklarasikan tanpa nilai awal dan nantinya akan diisi melalui input pengguna atau pernyataan program lainnya.

```
int jumlah=0;
int batas;
```

d. Potongan kode Java tersebut digunakan untuk membaca input dari pengguna menggunakan kelas Scanner. Pertama, objek Scanner bernama input dibuat untuk mengambil data dari keyboard (System.in). Program kemudian menampilkan pesan "Masukkan nilai batas =" agar pengguna tahu harus memasukkan nilai. Nilai yang dimasukkan oleh pengguna disimpan ke dalam variabel batas menggunakan input.nextInt(). Setelah itu, input.close() digunakan untuk menutup objek Scanner dan membebaskan sumber daya yang digunakan.

```
Scanner input=new Scanner (System.in);
System.out.print("Masukkan nilai batas = ");
batas= input.nextInt();
input.close();
```

e. Potongan kode Java tersebut menggunakan perulangan for untuk menampilkan deret angka dari 1 hingga nilai batas yang dimasukkan pengguna. Di setiap iterasi, angka i dicetak ke layar dan nilainya dijumlahkan ke variabel jumlah. Setelah mencetak angka, terdapat kondisi if yang memeriksa apakah i masih lebih kecil dari batas; jika ya, maka program

menampilkan tanda “ + ” di antara angka-angka. Namun jika i sudah sama dengan batas, program menampilkan tanda “ = ” sebagai penutup deret.

```
for (int i=1;i<=batas;i++) {  
    System.out.print(i);  
    jumlah= jumlah+i;  
    if (i<batas) {  
        System.out.print(" + ");  
    } else {  
        System.out.print(" = ");  
    }  
}
```

f. Potongan kode Java tersebut berfungsi untuk menampilkan hasil akhir dari penjumlahan deret angka yang telah dihitung sebelumnya dan disimpan dalam variabel jumlah. Pernyataan `System.out.println(jumlah);` mencetak nilai total penjumlahan ke layar pada baris baru setelah tanda sama dengan (=).

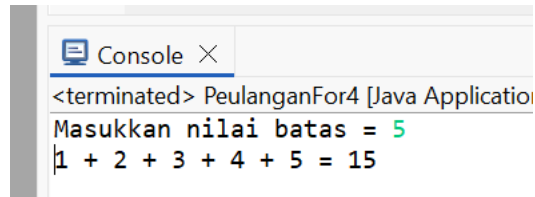
```
    System.out.println(jumlah);  
}
```

```
}
```

g. Jadinya kode program akan seperti ini

```
1 import java.util.Scanner;  
2  
3 public class PeulanganFor4 {  
4  
5     public static void main(String[] args) {  
6         int jumlah=0;  
7         int batas;  
8         Scanner input=new Scanner (System.in);  
9         System.out.print("Masukkan nilai batas = ");  
10        batas= input.nextInt();  
11        input.close();  
12        for (int i=1;i<=batas;i++) {  
13            System.out.print(i);  
14            jumlah= jumlah+i;  
15            if (i<batas) {  
16                System.out.print(" + ");  
17            } else {  
18                System.out.print(" = ");  
19            }  
20        }  
21        System.out.println(jumlah);  
22    }  
23 }  
24  
25  
26  
27  
28  
29 }
```

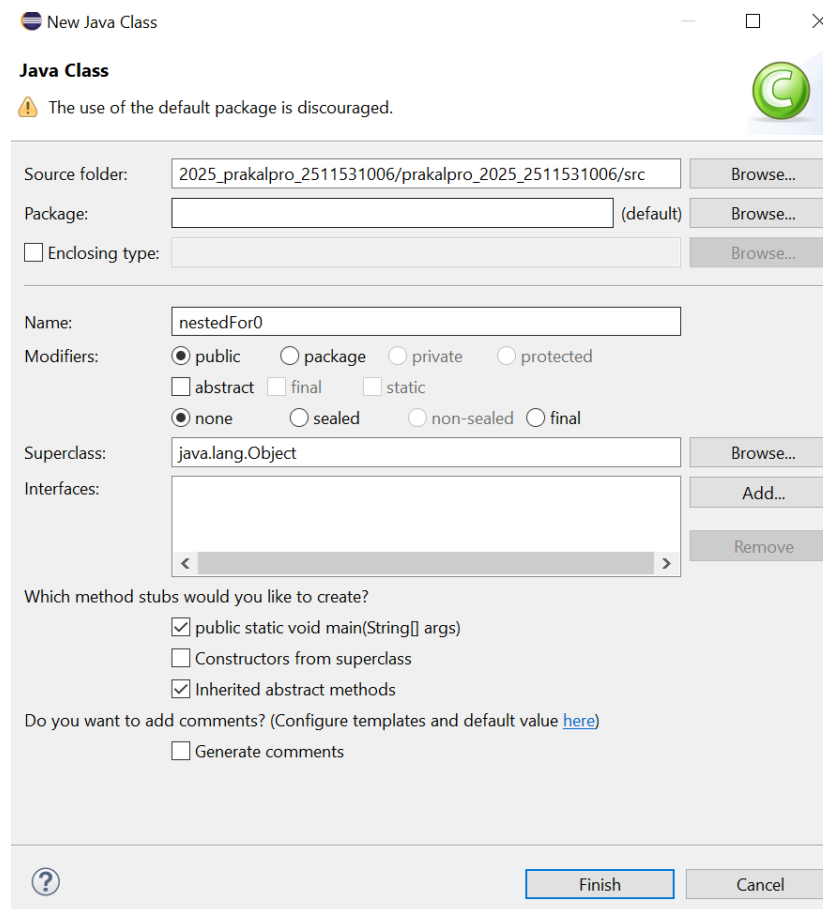
h. Hasil run kode program



```
<terminated> PeulanganFor4 [Java Application]
Masukkan nilai batas = 5
1 + 2 + 3 + 4 + 5 = 15
```

5. Program 5

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



New Java Class

The use of the default package is discouraged.

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

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: nestedFor0

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. Maka akan muncul seperti ini jika sudah tekan finish.


```
public class nestedFor0 {
    public static void main(String[] args) {
```

c. Potongan kode Java tersebut menggunakan perulangan bersarang (nested loop) untuk mencetak pola titik (.) secara menurun. Perulangan luar (for (int line = 1; line <= 5; line++)) mengatur jumlah baris dari 1 hingga 5, sedangkan perulangan dalam menentukan jumlah titik pada setiap baris.

```
    for (int line = 1; line <= 5; line++) {
        for (int j = 1; j <= (-1 * line + 5); j++) {
            System.out.print(".");
        }
```

d. Potongan kode Java tersebut digunakan untuk menampilkan pola angka yang disertai dengan titik-titik di depannya. Setelah mencetak sejumlah titik sesuai baris menggunakan perulangan dalam, perintah System.out.print(line); akan menampilkan nomor baris tersebut di akhir setiap baris titik. Lalu System.out.println(); digunakan untuk pindah ke baris baru sebelum mencetak pola berikutnya.

```
            System.out.print(line);
            System.out.println();
        }
    }
}
```

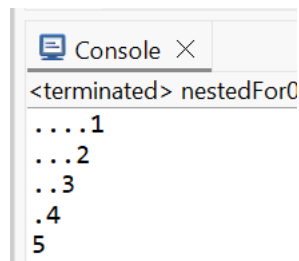
e. Jadinya kode program akan seperti ini

```

public class nestedFor0 {
    public static void main(String[] args) {
        for (int line = 1; line <= 5; line++) {
            for (int j = 1; j <= (-1 * line + 5); j++) {
                System.out.print(".");
            }
            System.out.print(line);
            System.out.println();
        }
    }
}

```

f. Hasil run kode program



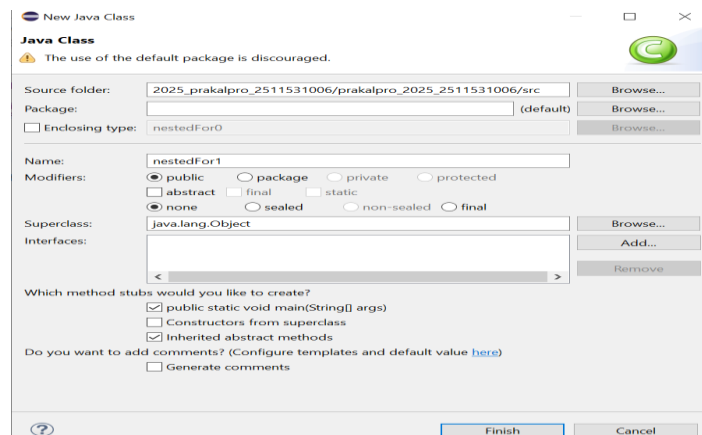
```

<terminated> nestedFor0
....1
...2
..3
.4
5

```

6. Program 6

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



New Java Class

The use of the default package is discouraged.

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

Package: (default) Browse...

☐ Enclosing type: nestedFor0 Browse...

Name: nestedFor1

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. Maka akan muncul seperti ini jika sudah tekan finish.

```
public class nestedFor1 {  
    public static void main(String[] args) {
```

c. Kode itu pakai dua perulangan for buat nampilin bintang (*) dalam bentuk kotak. Perulangan luar ngatur barisnya dari 1 sampai 5, sedangkan perulangan dalam ngatur berapa banyak bintang di setiap baris (juga 5 kali). Jadi setiap baris isinya 5 bintang, dan totalnya ada 5 baris.

```
for (int i = 1; i <= 5; i++) {  
    for (int j = 1; j <= 5; j++) {  
        System.out.print("*");  
    }
```

d. Bagian kode itu dipakai buat ganti baris setelah satu baris bintang selesai dicetak. Jadi `System.out.println();` fungsinya biar kursor pindah ke bawah, biar baris berikutnya bisa mulai dari awal lagi. Komentarnya `// to end the line` cuma ngasih tau kalau perintah itu buat nutup atau ngakhirin baris. Intinya, tanpa baris ini, semua bintang bakal nyatu di satu baris panjang.

```
        System.out.println();  
        // to end the line  
    }  
}
```

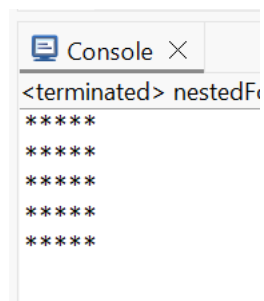
e. Jadinya kode program akan seperti ini

```

1
2 public class nestedFor1 {
3
4     public static void main(String[] args) {
5         for (int i = 1; i <= 5; i++) {
6             for (int j = 1; j <= 5; j++) {
7                 System.out.print("*");
8             }
9             System.out.println();
10            // to end the line
11        }
12    }
13 }

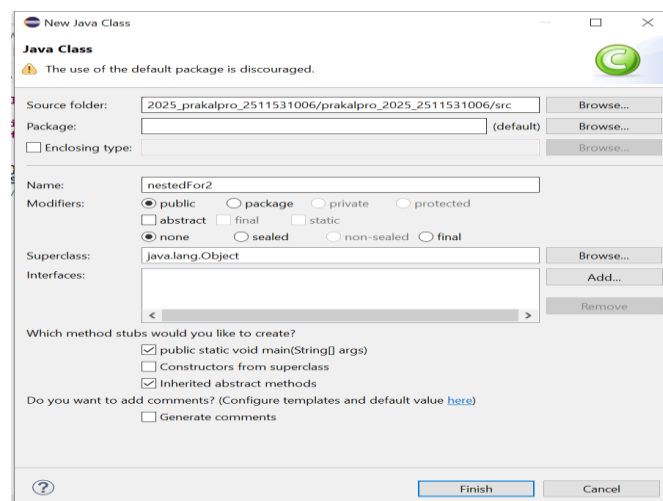
```

f. Hasil run kode program



7. Program 7

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



b. Maka akan muncul seperti ini jika sudah tekan finish.

```
2 public class nestedFor2 {  
3  
4 public static void main(String[] args) {  
--
```

c. Kode itu pakai dua perulangan for buat nampilin hasil penjumlahan dari dua variabel i dan j. Nilai i dan j masing-masing jalan dari 0 sampai 5. Di setiap langkah, program ngeprint hasil i + j diikuti spasi biar rapi. Jadi, setiap baris nunjukin hasil penjumlahan dengan nilai i yang sama tapi j-nya berubah-ubah.

```
for (int i = 0; i <= 5; i++) {  
    for (int j = 0; j <= 5; j++) {  
        System.out.print(i + j + " ");  
    }  
}
```

d. Bagian kode itu fungsinya buat ganti baris setelah satu baris angka selesai dicetak. Jadi setiap kali System.out.println(); dijalankan, program bakal pindah ke baris baru biar hasil output-nya rapi dan nggak nyatu semua di satu baris. Komentarnya // to end the line cuma ngasih tau kalau perintah itu dipakai buat nutup atau ngakhirin baris sebelum lanjut ke baris berikutnya.

```
9         System.out.println();  
10        // to end the line  
11    }  
12 }  
13 }
```

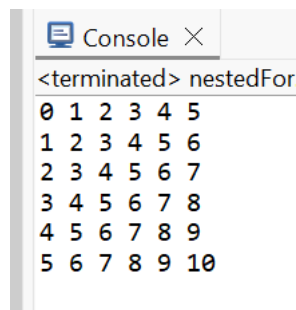
e. Jadinya kode program akan seperti ini.

```

2 public class nestedFor2 {
3
4     public static void main(String[] args) {
5         for (int i = 0; i <= 5; i++) {
6             for (int j = 0; j <= 5; j++) {
7                 System.out.print(i + j + " ");
8             }
9             System.out.println();
10            // to end the line
11        }
12    }
13 }

```

f. Hasil run kode program



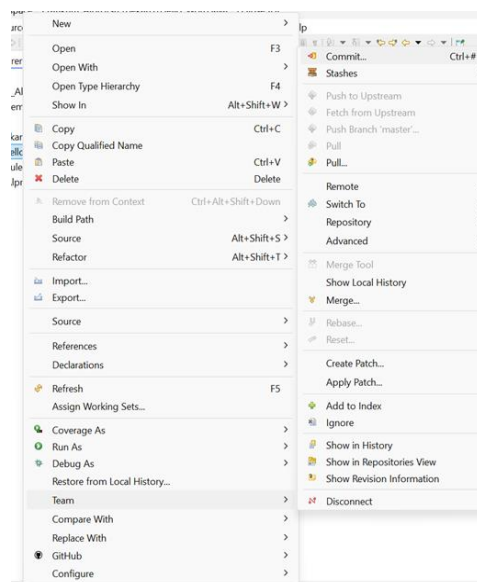
```

<terminated> nestedFor
0 1 2 3 4 5
1 2 3 4 5 6
2 3 4 5 6 7
3 4 5 6 7 8
4 5 6 7 8 9
5 6 7 8 9 10

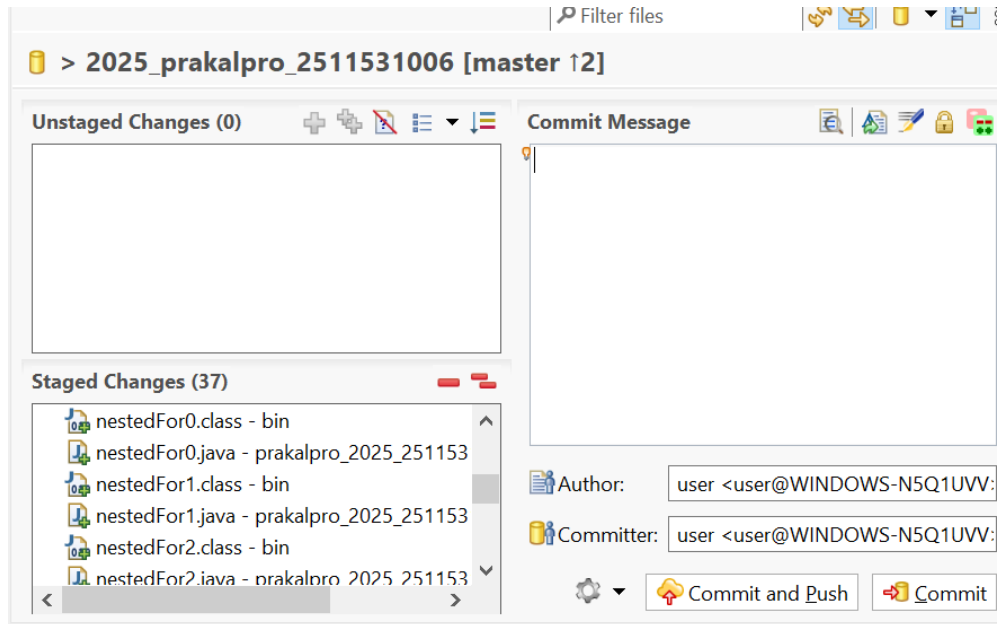
```

8. Menghubungkan Eclipse ke Github

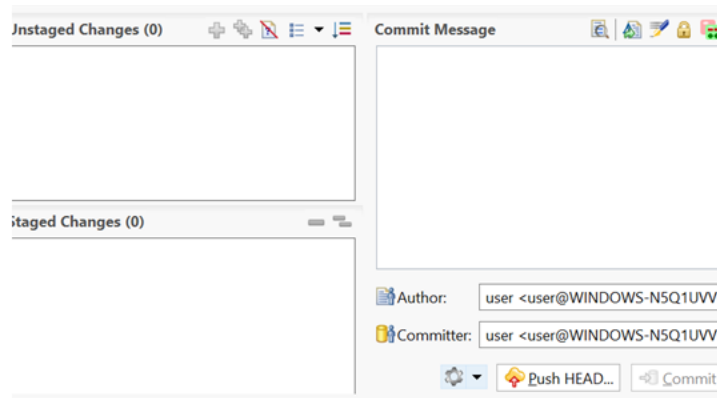
a. Klik kanan project yang telah dibuat, lalu pilih “Team”, lalu pilih “Commit”.



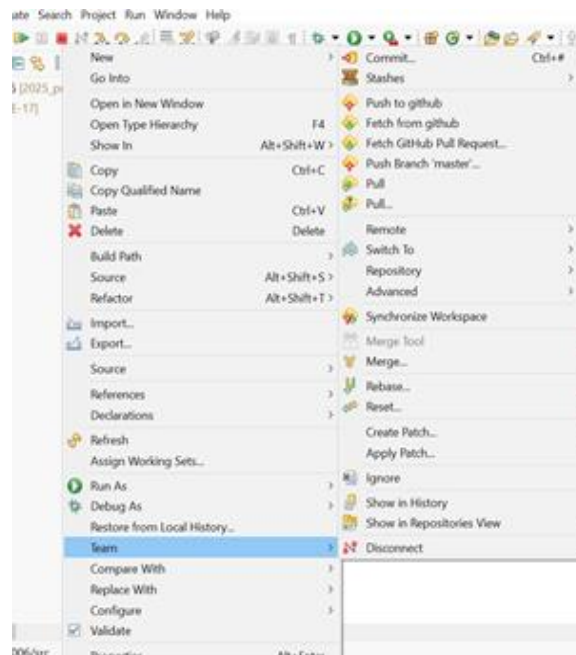
b. Block semua yang berada di atas dengan , lalu tekan logo tambah dan Setelah ditambah, buat pesan commitnya, lalu klik “Commit”.



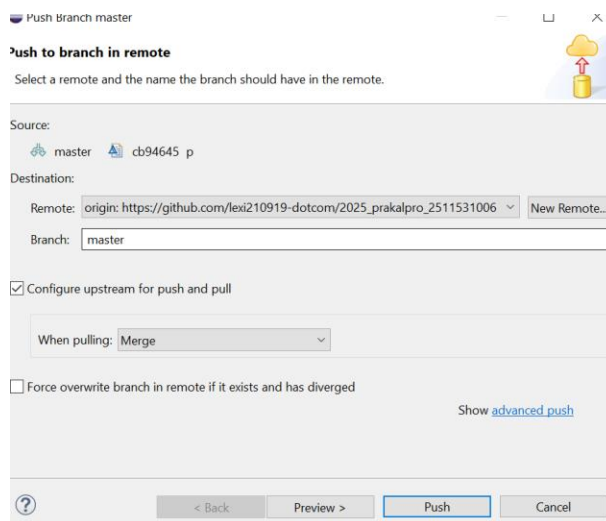
c. Jika tampilan sudah hilang, maka item berhasil di commit.



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



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



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

lexi210919-dotcom / 2025_prakalpro_2511531006

Type [Z] 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](#)

Releases

No releases published

[Create a new release](#)

Packages

Activate Windows

No packages published

[Go to Settings to activate Windows](#)

[Dibeli oleh Ben Markana](#)

BAB III

PENUTUP

3.1.KESIMPULAN

Dari hasil praktikum minggu ini, bisa disimpulkan kalau materi tentang perulangan itu sangat penting buat dipahami dalam dunia pemrograman. Lewat latihan-latihan yang udah dikerjain, aku jadi ngerti gimana cara kerja perulangan kayak *for*, *while*, dan *do-while* dalam menyelesaikan masalah yang butuh proses berulang. Awalnya emang agak ribet karena tiap jenis perulangan punya aturan sendiri, tapi setelah dicoba langsung di program, ternyata konsepnya cukup simpel asal ngerti alurnya. Dari situ aku juga belajar kalau perulangan bisa bantu banget buat bikin kode jadi lebih singkat, efisien, dan nggak perlu nulis hal yang sama berulang-ulang. Selain itu, aku jadi terbiasa berpikir lebih logis dan sistematis waktu nyusun alur program. Jadi, praktikum ini bener-bener ngebantu buat ngasah kemampuan dasar dalam bikin logika program dan juga jadi bekal buat belajar algoritma yang lebih kompleks di tahap selanjutnya.

3.2.SARAN

Menurut aku, praktikum ini udah cukup bagus karena langkah-langkahnya jelas dan mudah diikuti. Tapi biar makin menarik, mungkin bisa ditambah contoh kasus yang lebih beragam, misalnya kasus yang mirip sama situasi di kehidupan nyata biar kita bisa lihat langsung manfaatnya. Selain itu, penjelasan teori di awal praktikum bisa dibuat lebih sederhana atau disertai contoh kecil biar lebih gampang dipahami sebelum langsung masuk ke koding. Kalau bisa juga, waktu asistensi ditambah sedikit biar mahasiswa bisa tanya hal-hal yang belum paham tanpa terburu-buru. Dengan begitu, pembelajaran tentang perulangan ini bisa lebih maksimal, dan semua peserta bisa benar-benar ngerti bukan cuma sekedar menyalin kode, tapi juga paham cara kerjanya.

DAFTAR PUSTAKA

Wahana Komputer. (2019). Belajar Pemrograman Java untuk Pemula. Yogyakarta: *Andi Publisher*.

Nugroho, Adi. (2020). Algoritma dan Pemrograman dengan Java. Bandung: *Informatika*