

**LAPORAN PRAKTIKUM ALGORITMA DAN
PEMROGRAMAN
PEKAN 5 PERULANGAN FOR DI JAVA**



Disusun Oleh :

DIKA GIOWANDA

NIM 2311533025

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

JOVANTRI IMMANUEL GULO

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, OKTOBER 2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan praktikum mata kuliah Pemrograman Berorientasi Objek dengan judul "Penerapan Perulangan *for* dan *nested for* dalam Bahasa Pemrograman Java" ini tepat pada waktunya.

Laporan ini disusun sebagai bentuk dokumentasi dan refleksi dari kegiatan praktikum yang telah dilakukan, dengan tujuan untuk memahami konsep dasar perulangan *for* dan *nested for* dalam pemrograman java, khususnya struktur kendali *for* dan *nested for*. Melalui praktikum ini, penulis berkesempatan mengaplikasikan teori yang telah dipelajari di kelas ke dalam bentuk program nyata, seperti pembuatan pola bintang, perhitungan matematis sederhana, serta manipulasi data berulang yang efisien.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Ucapan terima kasih disampaikan kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan, ilmu, dan motivasi selama proses praktikum berlangsung.

Akhir kata, semoga laporan ini dapat bermanfaat sebagai bahan pembelajaran dan referensi bagi penulis maupun pembaca dalam memahami pentingnya struktur perulangan dalam pengembangan program berbasis Java.

Padang, 2025

Penyusun

.DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN.....	1
1.1 Pengertian Praktikum.....	1
1.2 Tujuan Praktikum.....	1
1.3 Persyaratan Praktikum.....	1
1.4 Tempat dan Waktu Praktikum.....	2
1.5 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN PRAKTIKUM.....	3
2.1 Pengertian Perulangan For.....	3
2.2 Pengertian Nested For.....	4
2.3 Langkah Langkah Pengerjaan Praktikum Pekan 5.....	6
BAB 3 PENUTUP.....	23
3.1 Kesimpulan.....	23
3.2 Saran.....	23
DAFTAR PUSTAKA.....	24

BAB 1

PENDAHULUAN

1.1 Pengertian Praktikum

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

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini antara lain sebagai berikut :

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

1.3 Persyaratan Praktikum

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

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

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

1.4 Waktu dan Tempat Praktikum

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

1.5 Manfaat Praktikum

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

BAB 2

PEMBAHASAN PRAKTIKUM

2.1 Pengertian Perulangan For

Perulangan *for* di Java adalah sebuah struktur kontrol yang memungkinkan eksekusi blok kode berulang kali berdasarkan kondisi tertentu, biasanya digunakan saat jumlah iterasi sudah diketahui sebelumnya. Sintaks dasarnya adalah *for* (*inisialisasi*; *kondisi*; *increment/decrement*) di mana blok kode akan dieksekusi selama kondisi bernilai *true*.

Struktur dasar:

1. *inisialisasi*: Pernyataan ini dieksekusi hanya sekali sebelum perulangan dimulai. Biasanya digunakan untuk mendeklarasikan dan menginisialisasi variabel kontrol (contoh: *int i = 0*).
2. *kondisi*: Kondisi ini dievaluasi sebelum setiap iterasi. Jika kondisi bernilai *true*, blok kode di dalamnya akan dieksekusi. Jika *false*, perulangan berhenti.
3. *Increment/Decrement*: Pernyataan ini dieksekusi setelah setiap iterasi. Biasanya digunakan untuk memperbarui nilai variabel kontrol (contoh: *i++* untuk menaikkan nilai *i* atau *i--* untuk menurunkannya).

Contoh perulangan for:

Contoh sederhana untuk mencetak angka dari 1 hingga 5:

```
public class ContohFor {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 5; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

Cara kerjanya:

1. *Inisialisasi*: `int i = 1` menetapkan nilai awal untuk variabel `i` menjadi 1.
2. *Kondisi*: `i <= 5` diperiksa. Selama nilai `i` kurang dari atau sama dengan 5, perulangan akan terus berjalan.
3. *Eksekusi*: `System.out.println(i)` akan mencetak nilai `i` saat ini.
4. *Perubahan Nilai*: `i++` akan menambah nilai `i` sebanyak 1.
5. *Proses berulang*: Setelah `i++`, kondisi `i <= 5` diperiksa lagi. Proses ini berlanjut hingga kondisi menjadi salah (yaitu, saat `i` menjadi 6).
6. *Penjelasan komponen for*
7. *Inisialisasi* (`int i = 1`): Pernyataan ini dijalankan hanya sekali di awal perulangan. Biasanya digunakan untuk mendeklarasikan dan menginisialisasi variabel *counter*.
8. *Kondisi* (`i <= 5`): Kondisi ini dievaluasi sebelum setiap iterasi. Jika kondisinya `true`, blok kode di dalam loop dijalankan. Jika `false`, perulangan berhenti.
9. *Perubahan nilai* (`i++`): Pernyataan ini dijalankan setelah setiap iterasi selesai. Biasanya digunakan untuk menambah atau mengurangi nilai *counter*.

2.2 Pengertian Nested For

Nested for di Java adalah perulangan *for* yang berada di dalam perulangan *for* lainnya. Perulangan ini terdiri dari satu perulangan luar (*outer loop*) dan satu atau lebih perulangan dalam (*inner loop*). Setiap kali perulangan luar selesai satu iterasi, perulangan dalam akan dijalankan seluruhnya dari awal sebelum melanjutkan ke iterasi perulangan luar berikutnya.

Cara kerja :

1. *Perulangan luar*: Perulangan luar berjalan sesuai dengan kondisinya.
2. *Perulangan dalam*: Setiap kali perulangan luar berulang, perulangan dalam akan dieksekusi penuh. Setelah perulangan

dalam selesai, kontrol kembali ke perulangan luar untuk memulai iterasi berikutnya.

3. Ilustrasi: Jika perulangan luar berjalan 5 kali, dan perulangan dalamnya berjalan 2 kali, maka total 10 kali eksekusi akan terjadi (karena $(5 \times 2 = 10)$). Contoh penggunaan :
4. Membuat tabel: nested for sering digunakan untuk membuat struktur seperti tabel, di mana perulangan luar mengulang baris dan perulangan dalam mengulang kolom.
5. Mengulang data: Mengulang setiap elemen dalam suatu struktur data yang memiliki dimensi lebih dari satu, seperti matriks.
6. Mengulang tugas berulang: Melakukan tugas berulang yang bergantung pada iterasi sebelumnya, misalnya, menghitung jumlah kombinasi dari dua set data yang berbeda.

Contoh nested for:

// Contoh pendek nested for

```
public class ContohNestedFor {  
    public static void main(String[] args) {  
        // Perulangan luar untuk baris  
        for (int i = 1; i <= 3; i++) {  
            // Perulangan dalam untuk kolom  
            for (int j = 1; j <= 5; j++) {  
                System.out.print("* "); // Cetak bintang dan spasi  
            }  
            System.out.println(); // Pindah baris setelah satu baris selesai  
        }  
    }  
}
```

Cara kerja contoh di atas:

1. Perulangan luar (*for (int i = 1; i <= 3; i++)*) akan berjalan sebanyak 3 kali (untuk $i = 1$, $i = 2$, dan $i = 3$).
2. Perulangan dalam (*for (int j = 1; j <= 5; j++)*) akan dieksekusi sepenuhnya untuk setiap iterasi perulangan luar.

3. Saat *i* bernilai 1, perulangan dalam akan berjalan 5 kali, mencetak "*" sebanyak 5 kali. Setelah itu, *System.out.println()* akan membuat baris baru.
4. Proses ini diulang saat *i* bernilai 2 dan *i* bernilai 3.
5. Hasil akhirnya adalah tiga baris, di mana setiap baris berisi 5 bintang.

* * * *

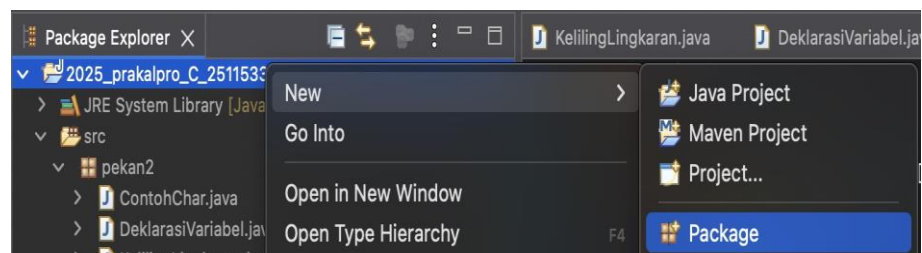
* * * *

* * * *

2.3 Langkah Langkah Pengerjaan Praktikum Pekan 5

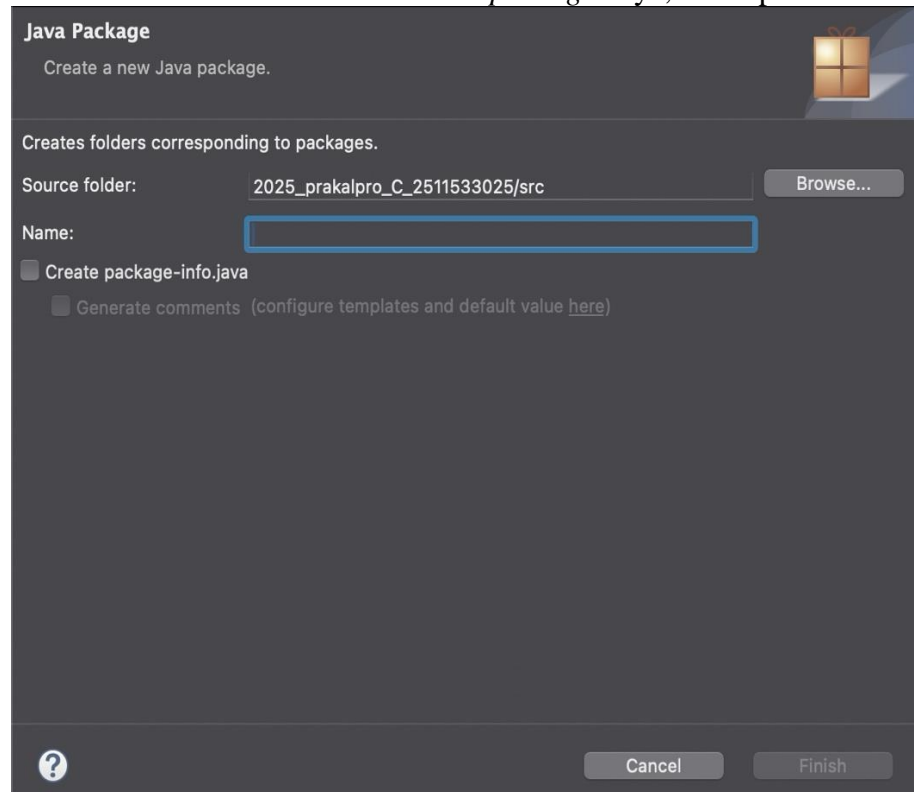
Sebelum ke langkah Langkah pengerjaan contoh proyek, saya akan menjelaskan terlebih dahulu gimana caranya menambahkan proyek nya terlebih dahulu :

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

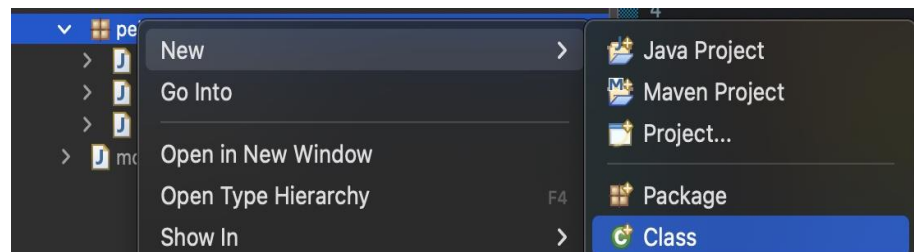


Setelah di click akan muncul halaman java *package*, dimana kita

disuruh bikin nama untuk folder *package* nya, lalu pilih finish.



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



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

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

Catatan :

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

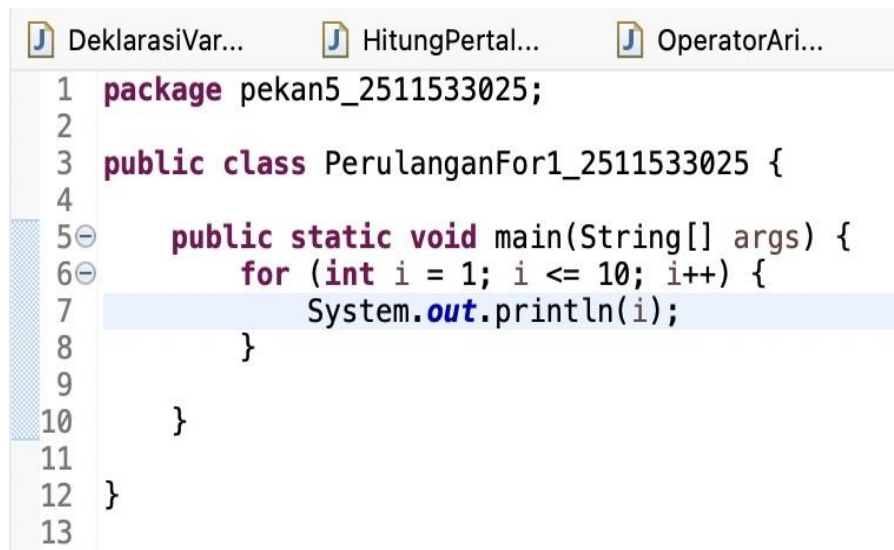
A. Baiklah karena proses menambahkan proyek sudah dijelaskan, selanjutnya langkah langkah pengerjaan proyek :

1. Membuat Perulangan For1

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

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Lalu masukan kode *int i=1* sebagai inisialisasi yaitu nilai awal, kode *i<=10* sebagai kondisi yaitu perulangan nilai sampai batas yang ditentukan, kode *i++* sebagai increment yaitu penambahan nilai *i*.

3. Masukan kode *system.out.println* untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya, jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar perulangan for1:



```
1 package pekan5_2511533025;
2
3 public class PerulanganFor1_2511533025 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(i);
8         }
9     }
10 }
11
12 }
13
```

4. Hasil projek Perulangan For 1.



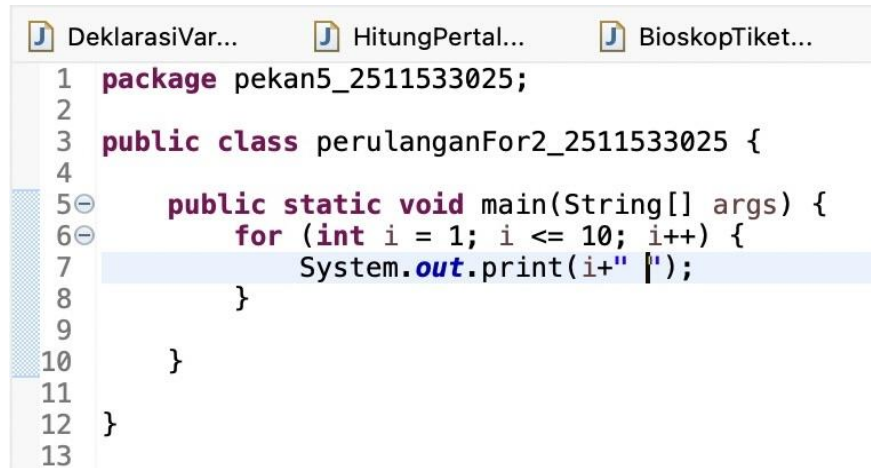
```
<terminated> PerulanganFor1_2511533025 [Java Application] /
1
2
3
4
5
6
7
8
9
10
```

2. Membuat Projek Perulangan For 2

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa pembuka sebagai pembuka perintah.

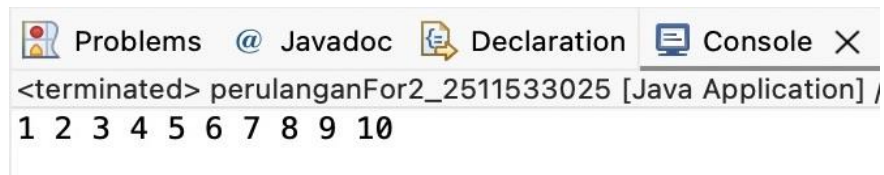
2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Lalu masukan kode *int i=1* sebagai inisialisasi yaitu nilai awal, kode *i<=10* sebagai kondisi yaitu perulangan nilai sampai batas yang ditentukan, kode *i++* sebagai increment yaitu penambahan nilai *i*.

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



```
1 package pekan5_2511533025;
2
3 public class perulanganFor2_2511533025 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 10; i++) {
7             System.out.print(i+" ");
8         }
9     }
10 }
11
12 }
13
```

4. Hasil projek Perulangan For 2.



```
<terminated> perulanganFor2_2511533025 [Java Application] /
1 2 3 4 5 6 7 8 9 10
```

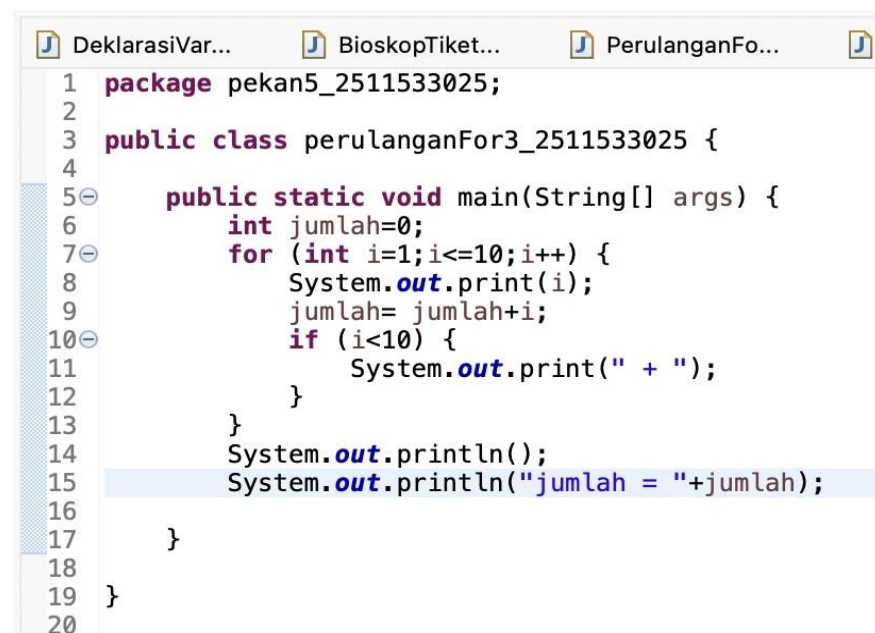
3. Membuat Projek Perulangan For 3

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah.

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat class baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *int jumlah=0* untuk mendeklarasikan variabel jumlah dengan nilai awal 0 dan si *int* menyimpan total penjumlahan, Lalu masukan kode *int i=1* sebagai inisialisasi yaitu nilai awal, kode *i<=10* sebagai kondisi yaitu perulangan nilai sampai batas yang ditentukan, kode *i++* sebagai increment yaitu penambahan nilai *i*. pada *system.out.print(i)* untuk mencetak nilai *i* saat ini, kenapa tidak menggunakan *println*? Karena supaya otuputnya menyamping. Supaya outputnya penjumlahan kita masukan kode *jumlah = jumlah+i*. Lalu mencetak tanda tanda tambah dengan kode *if (i<10)* { lalu tekan enter supaya dibawahnya masukan kode *system.out.print(" + ")* yaitu mencetak tanda + setelah setiap angka

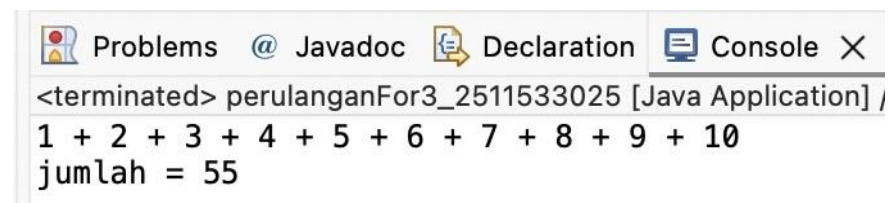
kecuali setelah angka terakhir yaitu 10. *If* berfungsi untuk mengontrol alur program dengan mengeksekusi blok kode tertentu hanya jika kondisi yang diberikan bernilai benar (*true*).

3. Masukkan kode *system.out.println* untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Kenapa kita menggunakan *print* bukan yang *println* pada pertengahan proses memasukan kode karena supaya outputnya menyamping. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya, jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar perulangan for 3 :



```
1 package pekan5_2511533025;
2
3 public class perulanganFor3_2511533025 {
4
5     public static void main(String[] args) {
6         int jumlah=0;
7         for (int i=1;i<=10;i++) {
8             System.out.print(i);
9             jumlah= jumlah+i;
10            if (i<10) {
11                System.out.print(" + ");
12            }
13        }
14        System.out.println();
15        System.out.println("jumlah = "+jumlah);
16    }
17 }
18
19 }
20 }
```

4. Hasil projek Perulangan For 3.



```
<terminated> perulanganFor3_2511533025 [Java Application] /
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10
jumlah = 55
```

d. Membuat projek Perulangan For 4

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setingnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah.

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *int jumlah=0* dan *int batas* untuk mendeklarasikan variabel jumlah dengan nilai awal 0 dan nilai batas, si *int* menyimpan total penjumlahan, masukan kode *Scanner input= new Scanner(System.in)* untuk membuat objek *Scanner* baru bernama *input* yang digunakan untuk membaca input dari keyboard(*System.in*), kemudian masukan kode *System.out.print("Masukan nilai batas =")* untuk menampilkan pesan "Masukan nilai batas =" di konsol untuk memberitahu pengguna agar memasukan nilai, masukan kode *batas=input.nextInt()* untuk membaca nilai integer yang dimasukkan oleh pengguna dari keyboard dan menyimpannya ke dalam variabel. Lalu masukan kode *int i=1* sebagai inisialisasi yaitu nilai awal, kode *i<=batas* sebagai kondisi yaitu perulangan nilai sampai batas yang ditentukan, kode *i++* sebagai increment yaitu penambahan nilai *i*. Lalu mencetak tanda tambah dengan kode *if (i<batas) {* lalu tekan enter supaya dibawahnya masukan kode *system.out.print(" + ")* yaitu mencetak tanda + setelah setiap angka yang bernilai *true* kecuali setelah angka terakhir pada batas. *If* berfungsi untuk mengontrol alur program dengan mengeksekusi blok kode tertentu hanya jika kondisi yang diberikan bernilai benar (*true*). Kemudian masukan kode *else* yang berfungsi untuk apabila kondisi perintah salah, maka kode didalam blok tidak dijalankan, dan lanjut ke *System.out.print(" = ")* untuk mencetak "=" ke layar.

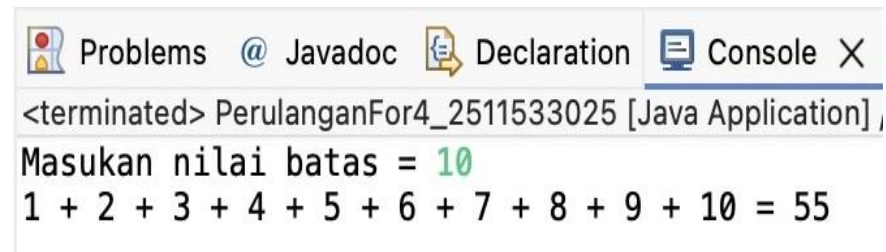
3. Masukan kode *system.out.println* untuk melanjutkan proses atau mencetak programnya dimana akan memindahkan kursor ke garis baru. Kenapa kita menggunakan *print* bukan yang *println* pada pertengahan proses memasukan kode karena supaya outputnya menyamping. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya, jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa setiap kode program per baris selalu diakhiri dengan tanda (; dan {) tergantung apa isi kode programnya di baris tersebut, jika sudah selesai memasukan kode program

diakhiri dengan kurung kurawa tertutup. Berikut adalah gambar perulangan *for* 4 :



```
1 package pekan5_2511533025;
2
3 import java.util.Scanner;
4
5 public class PerulanganFor4_2511533025 {
6
7     public static void main(String[] args) {
8         int jumlah=0;
9         int batas;
10        Scanner input= new Scanner(System.in);
11        System.out.print("Masukan nilai batas = ");
12        batas= input.nextInt();
13        input.close();
14        for (int i=1;i<=batas;i++) {
15            System.out.print(i);
16            jumlah= jumlah+i;
17            if (i<batas) {
18                System.out.print(" + ");
19            } else {
20                System.out.print(" = ");
21            }
22        }
23        System.out.println(jumlah);
24    }
25 }
26
27 }
```

5. Hasil proyek Perulangan For 4.



```
<terminated> PerulanganFor4_2511533025 [Java Application]
Masukan nilai batas = 10
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55
```

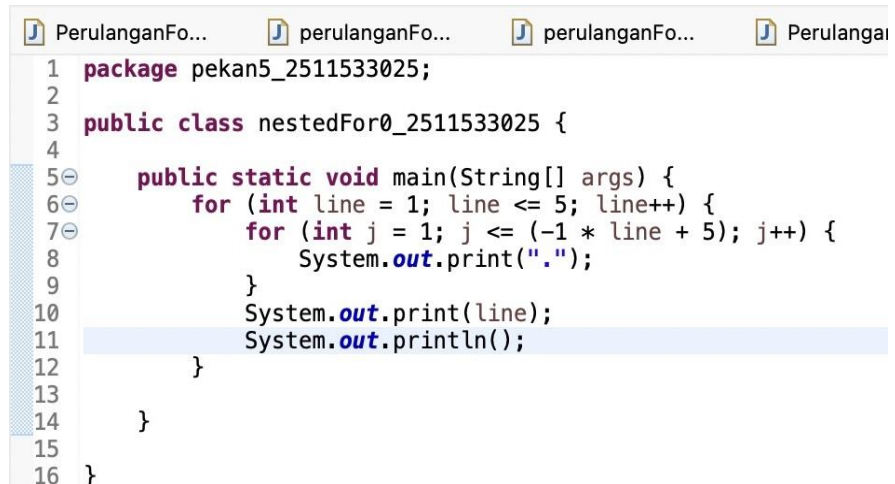
e. Membuat proyek Perulangan Nested For 0

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah.

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java*

Virtual Machine) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *for* karena kode ini merupakan kata kunci yang berfungsi untuk menjalankan blok kode di dalamnya sampai batas yang ditentukan. Lalu masukan kode *Inisialisasi*: *int line = 1* yaitu perulangan dimulai dengan membuat variabel integer bernama *line* dan memberinya nilai awal 1. Kode *line* berfungsi sebagai *counter* atau penghitung perulangan *for* sedangkan kode *int* berfungsi mendeklarasikan variabel yang akan menyimpan nilai dan total nilai penjumlahan. Masukan kode Kondisi: *line <= 5* yaitu perulangan akan terus berjalan selama nilai dari *line* kurang dari atau sama dengan 5, masukan kode *Iterasi*: *line++* yaitu setelah setiap *iterasi* (setiap kali blok kode selesai dijalankan), nilai *line* akan bertambah 1. Lalu masukan kode *for (int j = 1* yaitu inisialisasi *loop* dimana sebuah variabel *integer* bernama *j* dibuat dan diberi nilai awal 1, masukan kode *j <= (-1 * line + 5)* yaitu kondisi *loop* dimana *loop* akan terus berjalan selama nilai *j* kurang dari atau sama dengan hasil dari ekspresi $(-1 * line + 5)$, masukan kode *j++* yaitu operasi yang dilakukan setelah setiap iterasi dimana nilai *j* akan bertambah satu setelah setiap pengulangan selesai. Masukan kode *System.out.print(".")* untuk mencetak karakter titik (.) ke konsol tanpa membuat baris baru. Masukan kode *System.out.print(line)* untuk mencetak nilai dari variabel *line* ke *konsol* tanpa membuat baris baru.

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



```

1 package pekan5_2511533025;
2
3 public class nestedFor0_2511533025 {
4
5     public static void main(String[] args) {
6         for (int line = 1; line <= 5; line++) {
7             for (int j = 1; j <= (-1 * line + 5); j++) {
8                 System.out.print(".");
9             }
10            System.out.print(line);
11            System.out.println();
12        }
13    }
14 }
15
16 }

```

5. Hasil projek Perulangan Nested For 0.



```

<terminated> nestedFor0_2511533025 [Java Application] /App
....1
...2
..3
.4
5

```

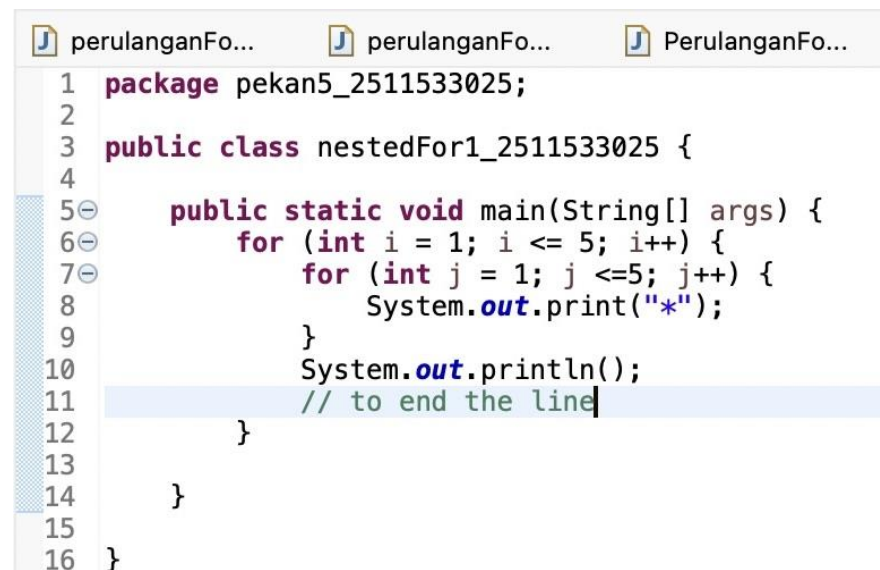
f. Membuat projek Perulangan Perulangan Nested For 1

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setingnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah.

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *for* karena kode ini merupakan kata kunci yang berfungsi untuk menjalankan blok kode di dalamnya sampai batas yang ditentukan. Setelah itu masukan kode *for* karena kode ini berfungsi untuk menjalankan blok kode di dalamnya sampai batas yang ditentukan, lalu masukan kode *int i = 1* yaitu Inisialisasi variabel *i* dengan nilai 1 yang merupakan titik awal perulangan, masukan kode *i <= 5* yaitu kondisi yang harus dipenuhi agar perulangan terus berjalan dimana perulangan akan berhenti ketika nilai *i* lebih besar dari 5, masukan kode *i++* yaitu ekspresi yang akan

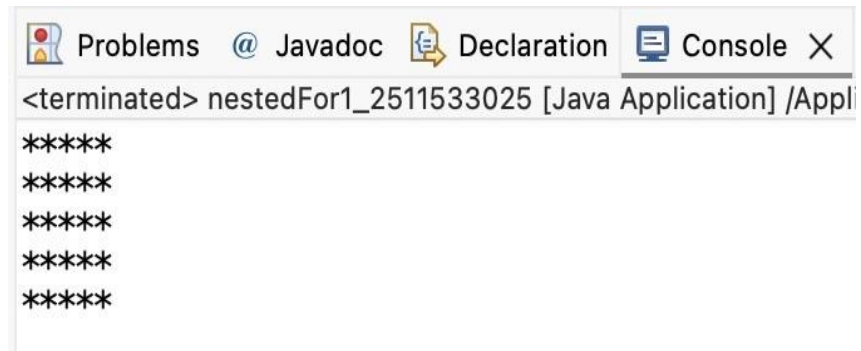
dijalankan di akhir setiap iterasi yang akan menambah nilai *i* sebanyak 1 setiap kali perulangan selesai, kode *int* berfungsi mendeklarasikan variabel yang akan menyimpan nilai dan total nilai penjumlahan. Lalu tahap selanjutnya, masukan kode (*int j = 1; j <= 5; j++*) kode ini adalah bagian dari perulangan yang mengatur prosesnya, dimana kode *int j = 1* yaitu *inisialisasi* dimana variabel *j* dimulai dari nilai 1, masukan kode *j <= 5* yaitu kondisi dimana perulangan akan terus berjalan selama nilai *j* kurang dari atau sama dengan 5, masukan kode *j++* yaitu Iterasi dimana nilai *j* akan bertambah 1 setiap kali perulangan selesai satu kali. Kemudian masukan kode *System.out.print("*")* dimana akan mencetak karakter * tanpa pindah ke baris baru, tekan enter pada kode *}* untuk blok kode sebelumnya, masukan kode *System.out.println()* dimana akan mencetak baris kosong dan memindahkan kursor ke baris baru yang berfungsi untuk mengakhiri baris saat ini, masukan kode *// to end the line* ini sebagai komentar yang menjelaskan tujuan dan fungsi dari baris kode di atasnya.

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



```
1 package pekan5_2511533025;
2
3 public class nestedFor1_2511533025 {
4
5     public static void main(String[] args) {
6         for (int i = 1; i <= 5; i++) {
7             for (int j = 1; j <= 5; j++) {
8                 System.out.print("*");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14 }
15
16 }
```

4. Hasil projek Perulangan Nested For 1.



```
<terminated> nestedFor1_2511533025 [Java Application] /Appli
*****
*****
*****
*****
*****
```

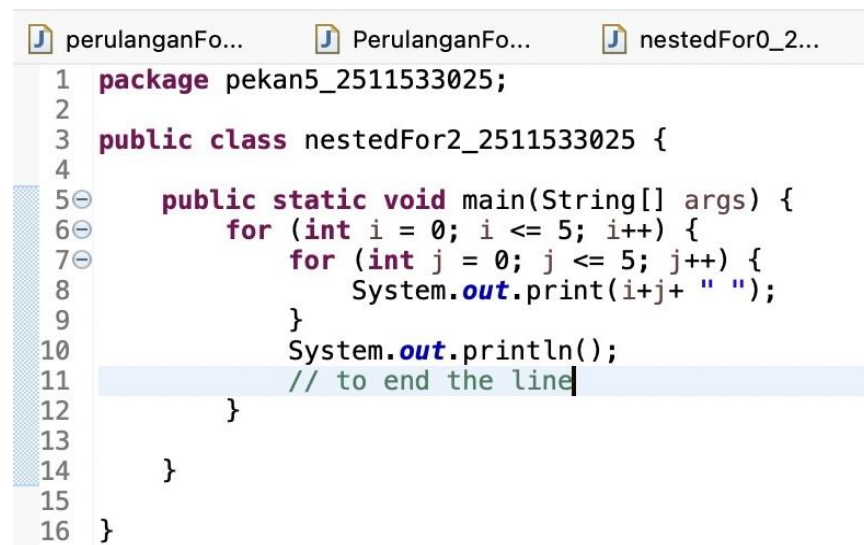
g. Membuat projek Perulangan Nested For 2

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah.

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (*Java Virtual Machine*) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *for* karena kode ini merupakan kata kunci yang berfungsi untuk menjalankan blok kode di dalamnya sampai batas yang ditentukan. Setelah itu masukan kode *for* karena kode ini berfungsi untuk menjalankan blok kode di dalamnya sampai batas yang ditentukan, lalu masukan kode *int i = 0* yaitu Inisialisasi variabel *i* dengan nilai 0 yang merupakan titik awal perulangan, masukan kode *i <= 5* yaitu kondisi yang harus dipenuhi agar perulangan terus berjalan dimana perulangan akan berhenti ketika nilai *i* lebih besar dari 5, masukan kode *i++* yaitu ekspresi yang akan dijalankan di akhir setiap iterasi yang akan menambah nilai *i* sebanyak 1 setiap kali perulangan selesai, kode *int* berfungsi mendeklarasikan variabel yang akan menyimpan nilai dan total nilai penjumlahan. Lalu tahap selanjutnya, masukan kode (*int j = 1; j <= 5; j++*) kode ini adalah bagian dari perulangan yang mengatur prosesnya, dimana kode *int j = 1* yaitu *inisialisasi* dimana variabel *j* dimulai dari nilai 1, masukan kode *j <= 5* yaitu kondisi dimana perulangan akan terus berjalan selama nilai *j* kurang dari atau sama dengan 5, masukan kode *j++* yaitu Iterasi dimana nilai *j* akan bertambah 1 setiap kali perulangan selesai satu kali. Kemudian masukan kode *System.out.print(i+j+ " ")* yaitu mencetak hasil penjumlahan variabel *i* dan *j* yang diikuti dengan spasi tanpa membuat baris baru, tekan enter pada kode *}* untuk blok kode

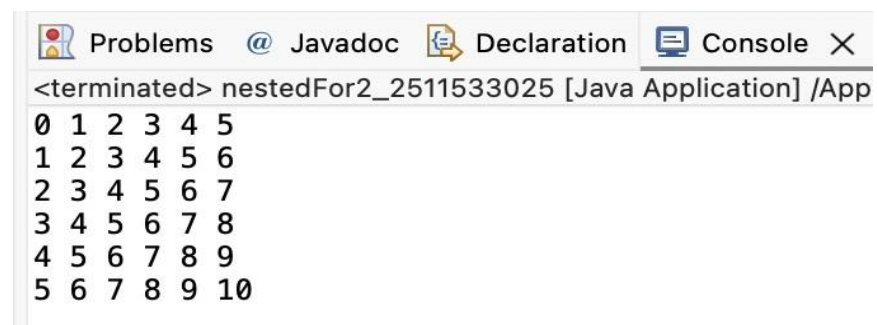
sebelumnya, masukan kode `System.out.println()` yaitu mencetak baris baru setelah loop atau blok kode selesai, sehingga output berikutnya akan dimulai di baris baru, masukan kode `// to end the line` ini sebagai komentar yang menjelaskan tujuan dan fungsi dari baris kode sebelumnya.

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



```
1 package pekan5_2511533025;
2
3 public class nestedFor2_2511533025 {
4
5     public static void main(String[] args) {
6         for (int i = 0; i <= 5; i++) {
7             for (int j = 0; j <= 5; j++) {
8                 System.out.print(i+j+ " ");
9             }
10            System.out.println();
11            // to end the line
12        }
13    }
14 }
15
16 }
```

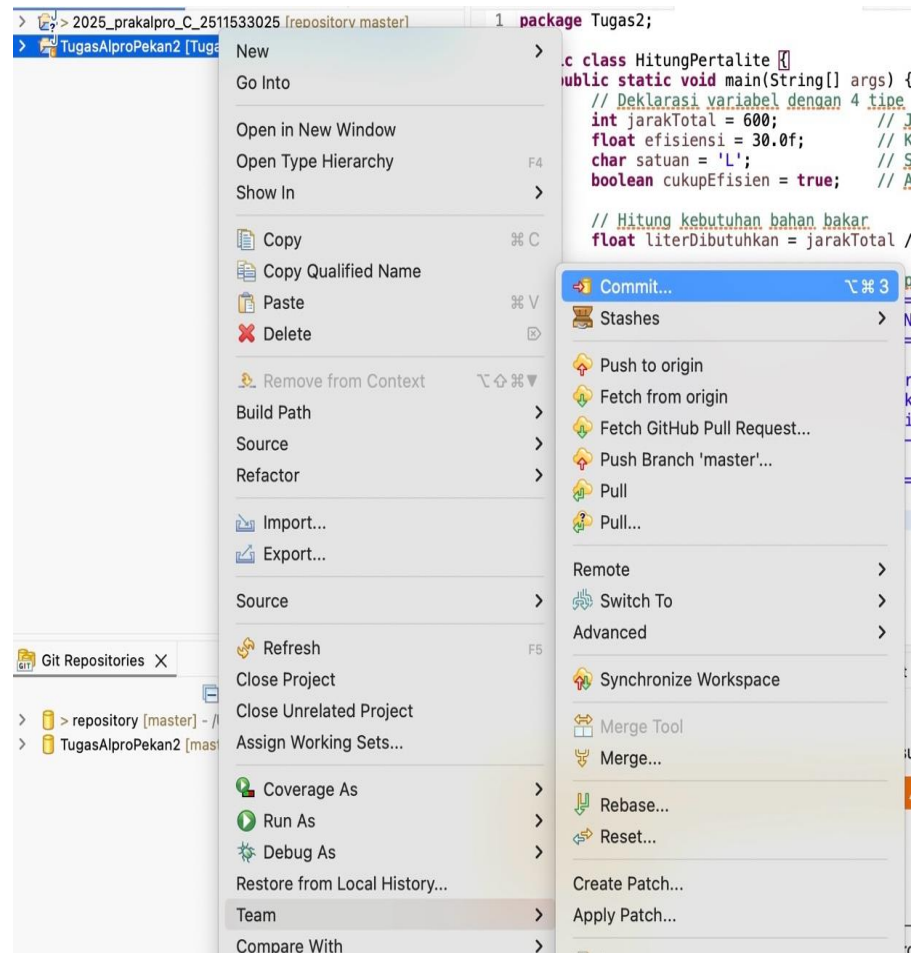
4. Hasil projek Perulangan Nested For 2.



```
Problems @ Javadoc Declaration Console X
<terminated> nestedFor2_2511533025 [Java Application] /App
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
```


B. Langkah Penyimpanan

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



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



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

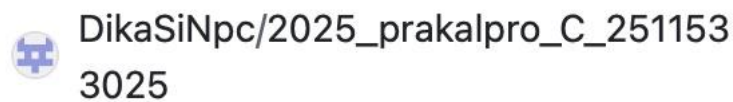
Commit Message

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

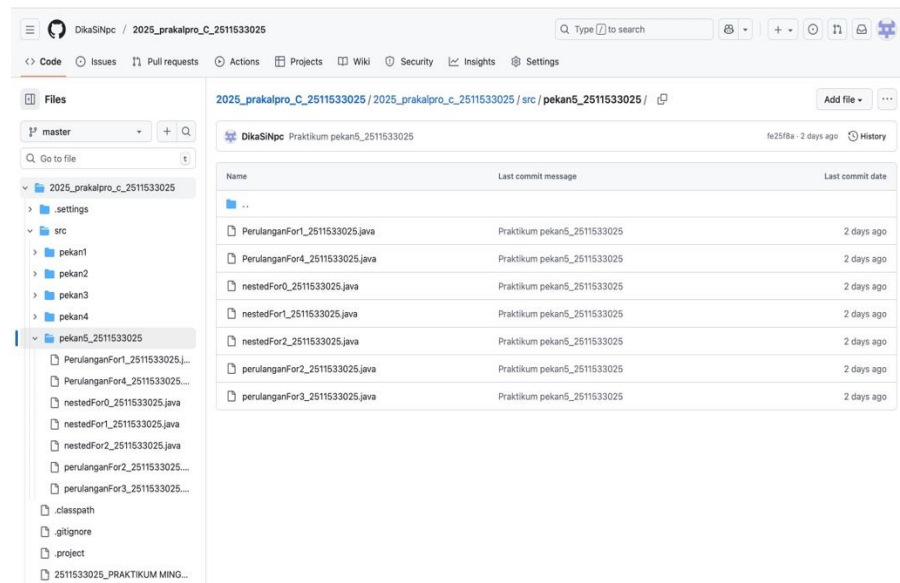
- Setelah itu tekan *commit* and pus, lalu ikutin arahan untuk memasukan nama dan pw akun git hub kalian, maka proyek kalian sudah tersimpan di akun github kalian.



- Proyek yang sudah disimpan di akun proyek.



- Gambar proyek apabila sudah masuk di akun github.



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



Tapi pada program yang sekarang berbeda dari sebelumnya karena pada program ini ada yang inputnya kita masukan angka atau huruf sesuai keinginan kita pada kolom *console*, tapi ingat harus sesuai perintah kode yang kita masukan.

BAB 3

PENUTUP

3.1 Kesimpulan

Dari praktikum yang telah dilakukan, maka dapat diambil kesimpulan bahwa Perulangan for adalah struktur penting dalam Java untuk menjalankan kode berulang kali dengan jumlah iterasi yang sudah diketahui. Strukturnya yang terdiri dari inisialisasi, kondisi, dan increment/decrement perubahan nilai membuatnya efisien dan mudah dikendalikan. Sementara itu, nested for (perulangan for di dalam for) memungkinkan kita menangani masalah dua dimensi, seperti membuat pola bintang, mengolah matriks, atau menampilkan data dalam bentuk tabel. Kedua konsep ini melatih logika berpikir terstruktur dan menjadi dasar penting dalam pemrograman. Tapi jika salah satu aja kode yang dibuat maka akan terjadi error. Oleh karena itu kita harus berhati hati ketika memasukan kode dan ikutin proses sesuai arahan yang diberikan. Dan saya sangat senang melaksanakan praktikum ini karena bisa menambah ilmu pengetahuan dan wawasan akan kemajuan teknologi.

3.2 Saran

Untuk menguasai materi ini, seringlah berlatih membuat berbagai pola atau mengolah array dua dimensi. Perhatikan alur perulangan agar tidak terjadi kesalahan logika atau infinite loop. Gunakan nama variabel yang jelas (misalnya baris dan kolom) agar kode mudah dibaca. Hindari penggunaan nested for yang berlebihan pada data besar karena bisa memperlambat program. Terakhir, kombinasikan perulangan ini dengan konsep lain seperti percabangan atau input pengguna untuk memperdalam pemahaman.

DAFTAR PUSTAKA

- Sumber daring (website):
[1] W3Schools, "Java For Loop," 2025. [Daring]. Tersedia pada: https://www.w3schools.com/java/java_for_loop.asp. [Diakses: 31-Oct-2025].

[2] GeeksforGeeks, "Loops in Java," 07-Apr-2025. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/java/loops-in-java/>. [Diakses: 31-Oct-2025].

[3] Programiz, "Java Nested Loop," 2025.[Daring]. Tersedia pada: <https://www.programiz.com/java-programming/nested-loop>. [Diakses: 31-Oct-2025].

[4] Sciencetech Easy, "Nested For Loop in Java," 2021. [Daring]. Tersedia pada: <https://www.scientecheasy.com/2021/05/nested-for-loop-injava>. [Diakses: 31-Oct-2025].