

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
KONDISIONAL STATEMENT



disusun Oleh:
Annisa Layli Ramadhani
2511532024

Dosen Pengampu: Wahyudi. Dr.. S.T.M.T
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 29 September 2025 dapat diselesaikan. Laporan ini membahas materi tentang Kondisional Statement sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 29 September 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I.....	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat	5
BAB II	6
2.1 Dasar Teori	6
2.1.1 If-Else	6
2.1.2 Switch Case.....	6
2.2 Langkah Kerja.....	7
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 4	7
2.2.2 Program Pertama: LatIf1.....	8
2.2.3 Program Kedua: Ifelse1.....	9
2.2.4 Program Ketiga: MultiIf	10
2.2.5 Program Keempat: Nilai	10
2.2.6 Program Kelima: Nama Bulan	11
2.2.7 <i>Commit</i> dan <i>Push</i> ke GitHub.....	12
BAB III.....	15
3.1 Kesimpulan	15
3.2 Saran.....	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, pengambilan keputusan merupakan salah satu aspek yang sangat penting untuk menentukan alur jalannya program. Pada bahasa pemrograman seperti Java, proses pengambilan keputusan dapat dilakukan dengan menggunakan kondisional *statement*. Dua bentuk kondisional *statement* yang paling umum digunakan adalah *if-else* dan *switch case*.

Pernyataan *if-else* digunakan ketika program perlu melakukan pemeriksaan kondisi logika tertentu. Jika suatu kondisi bernilai benar (*true*), maka blok perintah tertentu akan dijalankan, sedangkan jika kondisi salah (*false*), blok perintah lain dapat dijalankan. Struktur ini memberikan fleksibilitas yang tinggi, terutama untuk kondisi yang kompleks atau bercabang banyak.

Sementara itu, *switch case* biasanya dipakai untuk menangani banyak pilihan (*multiple choice*) dengan cara yang lebih sederhana dan lebih terstruktur dibandingkan *if-else* bertingkat. *Switch case* membuat kode program menjadi ringkas, mudah dipahami, serta mempermudah pengelolaan alur program ketika terdapat banyak kondisi yang harus diperiksa berdasar nilai variabel tertentu.

Kedua jenis pernyataan kondisional ini memiliki peran penting dalam membangun logika program. Pemahaman terhadap penggunaannya akan membantu programmer dalam menulis kode yang lebih efisien, terstruktur, serta sesuai dengan kebutuhan aplikasi yang sedang dikembangkan. Oleh karena itu, pembahasan mengenai *if-else* dan *switch case* menjadi bagian fundamental dalam materi dasar pemrograman.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.2.1 Memahami konsep dasar pernyataan kondisional pada bahasa pemrograman, khususnya *if-else* dan *switch case*.
- 1.2.2 Mampu mengimplementasikan logika percabangan dengan menggunakan struktur *if-else* maupun *switch case* sesuai kebutuhan program.
- 1.2.3 Melatih keterampilan dalam menyusun algoritma yang melibatkan proses pengambilan keputusan.
- 1.2.4 Mengetahui kelebihan dan kekurangan dari masing-masing pernyataan kondisional sehingga dapat memilih struktur yang tepat dalam pemecahan masalah.

1.3 Manfaat

Manfaat dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.3.1 Memberikan pemahaman dasar kepada mahasiswa/pelajar tentang pentingnya penggunaan struktur kendali dalam pemrograman.
- 1.3.2 Menjadi bekal untuk mengerjakan proyek pemrograman yang lebih kompleks di masa depan.
- 1.3.3 Mempermudah proses *debugging* dan pemeliharaan program karena alur logika lebih jelas.
- 1.3.4 Menambah wawasan praktis mengenai penerapan teori logika ke dalam bentuk nyata melalui bahasa pemrograman.

BAB II

PEMBAHASAN

2.1 Dasar Teori

2.1.1 *If-Else*

Pernyataan *if-else* adalah salah satu bentuk *control structure* dalam pemrograman yang digunakan untuk melakukan pengambilan keputusan. Dengan *if-else*, program dapat menentukan blok kode mana yang akan dijalankan berdasarkan hasil evaluasi suatu kondisi logika (benar/*true* atau salah/*false*).

Struktur dasar *if-else* sebagai berikut:

```
If (kondisi) {  
    // blok kode dijalankan jika  
    kondisi bernilai true  
}  
Else {  
    // blok kode dijalankan jika  
    kondisi bernilai false  
}
```

Jika kondisi pada bagian *if* bernilai *true*, maka perintah di dalam blok *if* akan dieksekusi, sedangkan blok *else* akan dilewati. Sebaliknya, jika kondisi pada *if* bernilai *false*, maka blok kode pada *else* yang akan dijalankan.

2.1.2 *Switch Case*

Switch case adalah salah satu bentuk pernyataan kondisional dalam pemrograman yang digunakan untuk memilih satu blok kode dari beberapa pilihan berdasarkan nilai sebuah variabel atau ekspresi. *Switch case* sering dipakai ketika terdapat banyak kondisi yang perlu diperiksa, khususnya jika kondisi tersebut berupa nilai diskrit (seperti angka, karakter, atau, string).

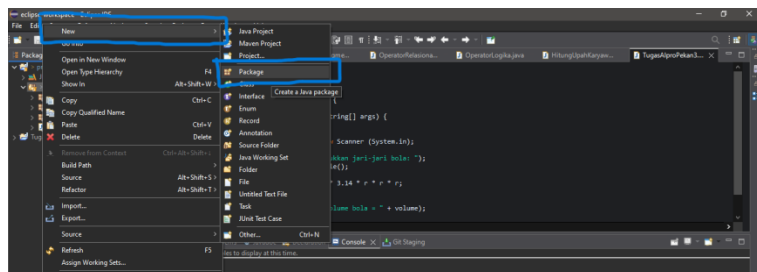
Struktur dasar *switch case* dalam Java adalah:

```
Switch (ekspresi) {
    Case nilai1:
        // blok kode jika ekspresi == nilai1
        Break;
    Case nilai2:
        // blok kode jika ekspresi == nilai2
        Break;
    ...
    default:
        // blok kode jika tidak ada nilai
        yang cocok
}
```

2.2 Langkah Kerja

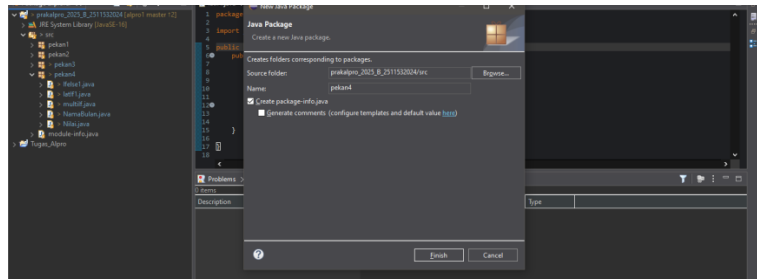
2.2.1 Pembuatan *Package* dan *Class* Pekan 4

1. Buka aplikasi eclipse, lalu buat package baru pada bagian src klik kanan, lalu pilih “New” dan klik tulisan Package.



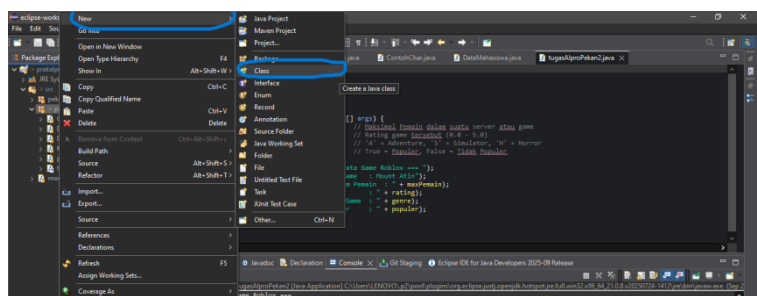
2.2.1 pembuatan package

2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. Beri nama “pekan4”. Lalu klik finish.



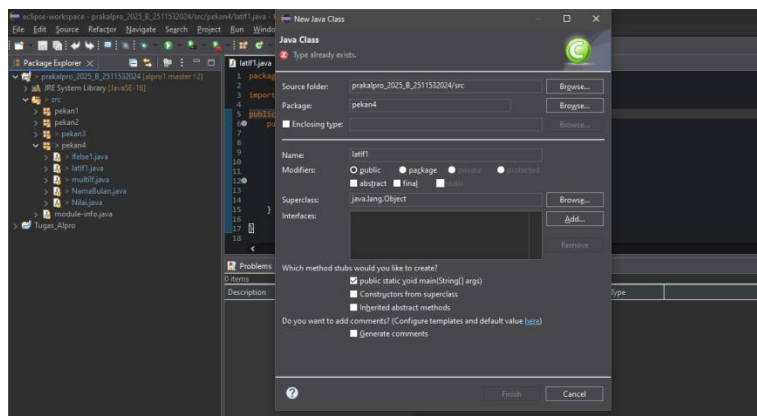
2.2.2 pembuatan package

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan4”. Lalu klik “New” dan pilih bagian “Class” untuk memulai membuat program.



2.2.3 pembuatan class

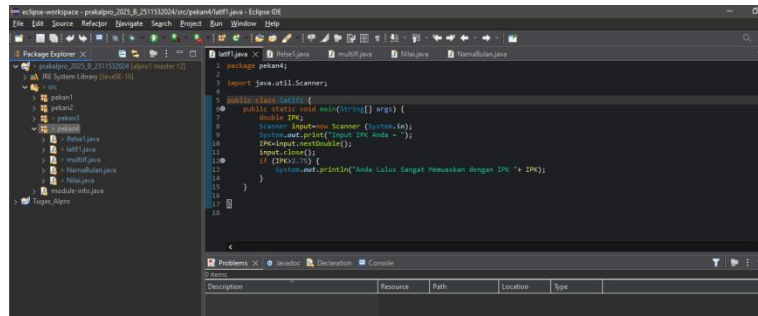
4. Buat nama Class program yang akan dibuat pada bagian nama tanpa spasi dan menggunakan huruf kapital pada awal kata, lalu klik bagian “public static void main(String[] args)”. Kemudian klik “finish”.



2.2.4 pembuatan class

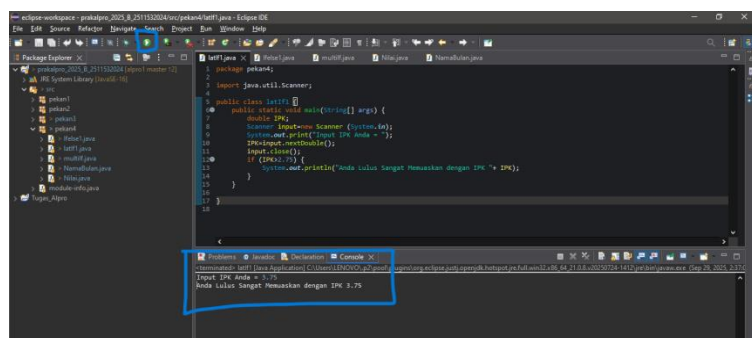
2.2.2 Program Pertama: LatIf1

1. Masukkan sintaks berikut.



2.2.5 sintaks latIf1

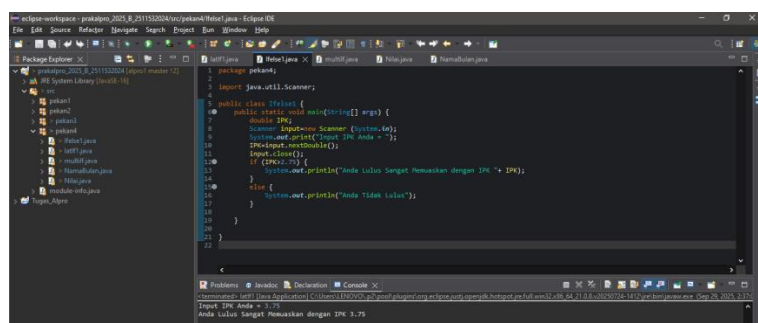
2. Lalu klik bagian “run” untuk melihat hasil sintaks yang sudah di inputkan. Lalu akan menghasilkan output seperti berikut.



2.2.6 Running code dan output sintaks

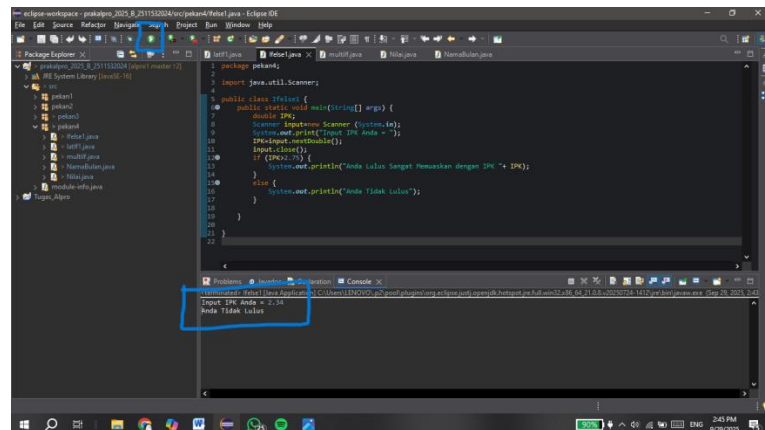
2.2.3 Program Kedua: Ifelse1

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “Ifelse1”. Lalu masukkan sintaks seperti berikut.



2.2.7 sintaks ifelse1

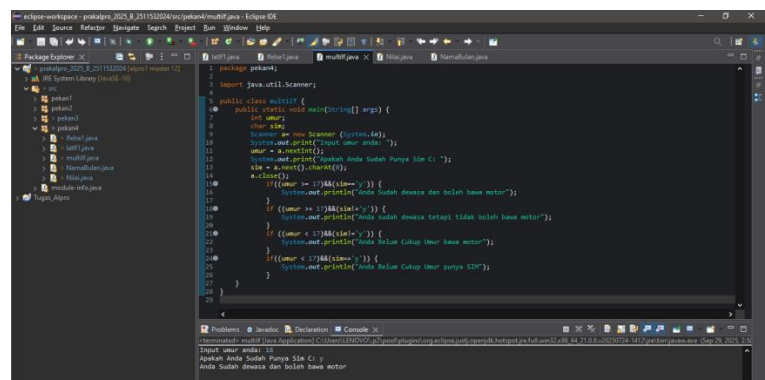
2. Klik “run” untuk melihat hasil sintaks yang sudah di buat.



2.2.8 running kode dan output sintaks

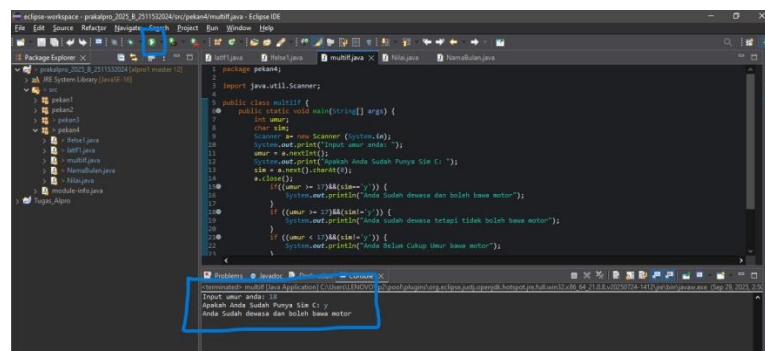
2.2.4 Program Ketiga: MultiIf

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “MultiIf”. Lalu masukkan sintaks seperti berikut.



2.2.9 sintaks multiif

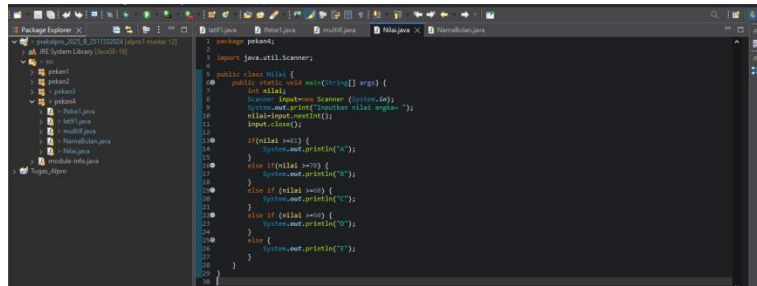
2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



2.2.10 running kode dan output sintaks

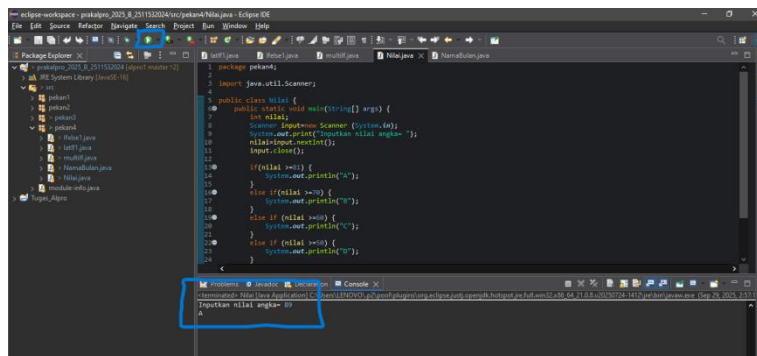
2.2.5 Program Keempat: Nilai

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “Nilai”. Lalu masukkan sintaks seperti berikut.



2.2.11 sintaks nilai

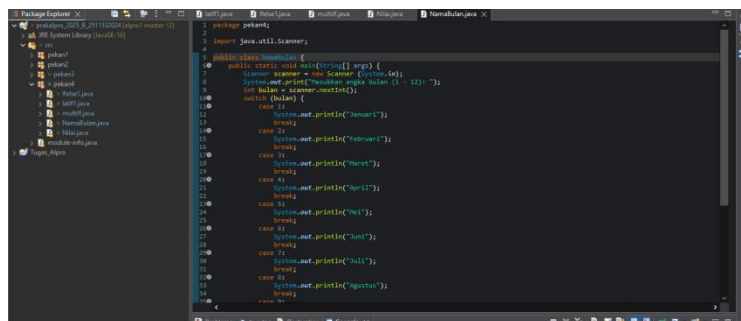
2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



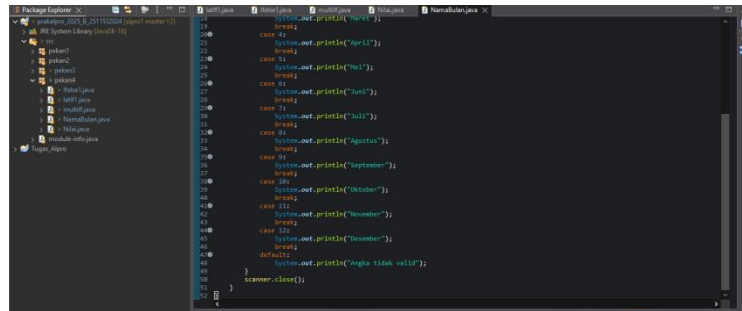
2.2.12 running kode dan output sintaks

2.2.6 Program Kelima: Nama Bulan

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “NamaBulan”. Lalu masukkan sintaks seperti berikut.

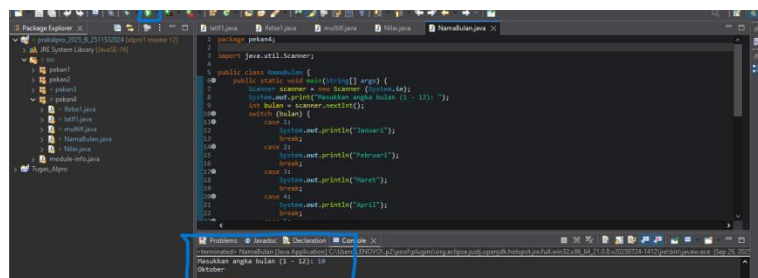


2.2.13 sintaks nama bulan



2.2.14 sintaks nama bulan

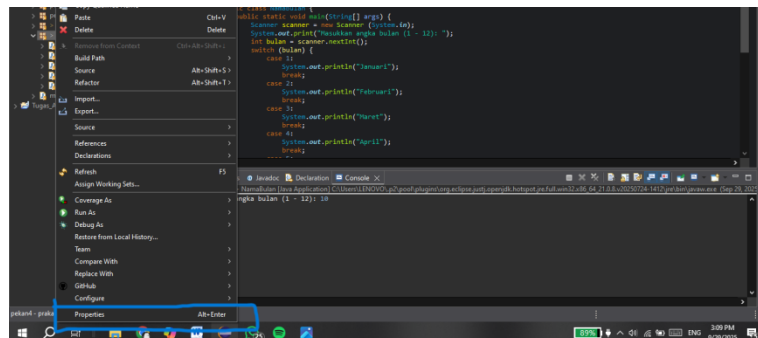
2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



2.2.15 running code dan output sintaks

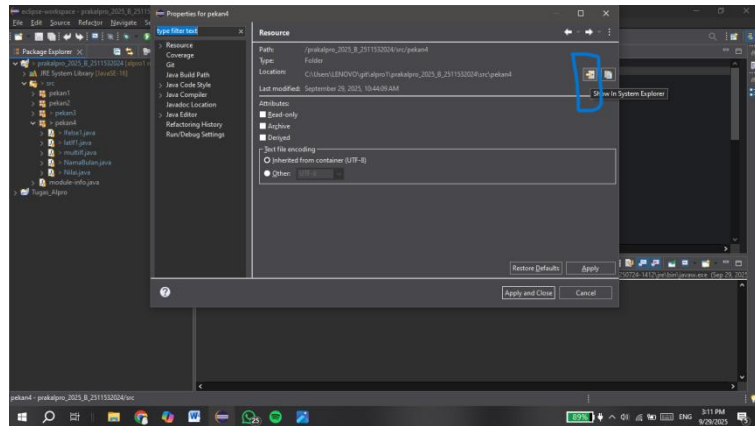
2.2.7 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan4”. Kemudian pilih bagian “properties”.



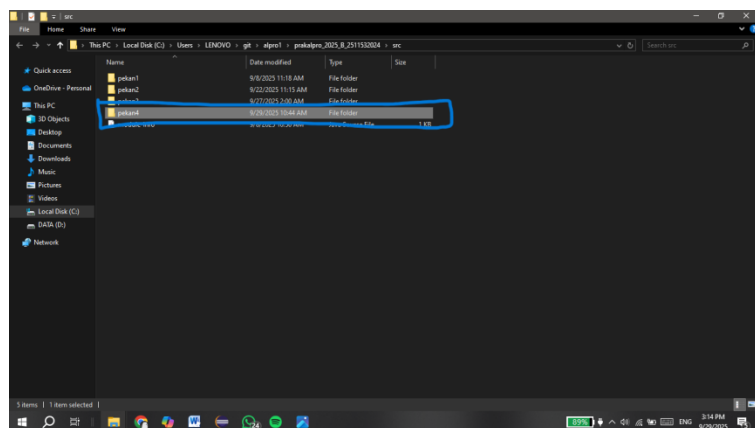
2.2.16 proses commit dan push ke GitHub

2. Setelah itu keluar opsi properties. Kemudian klik “Show In System Explorer”.



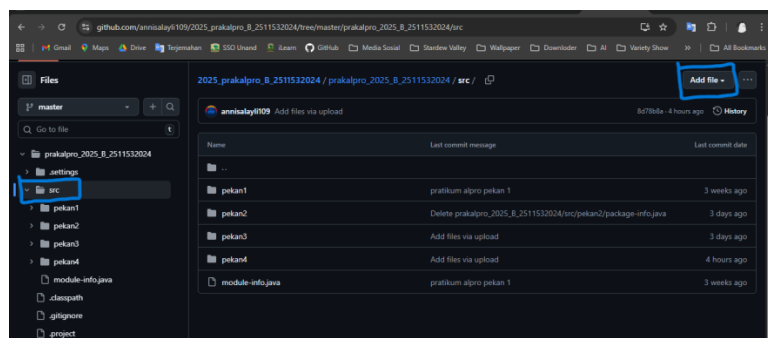
2.2.17 proses commit dan push ke GitHub

3. Setelah itu akan terbuka folder file tersebut di file explorer.



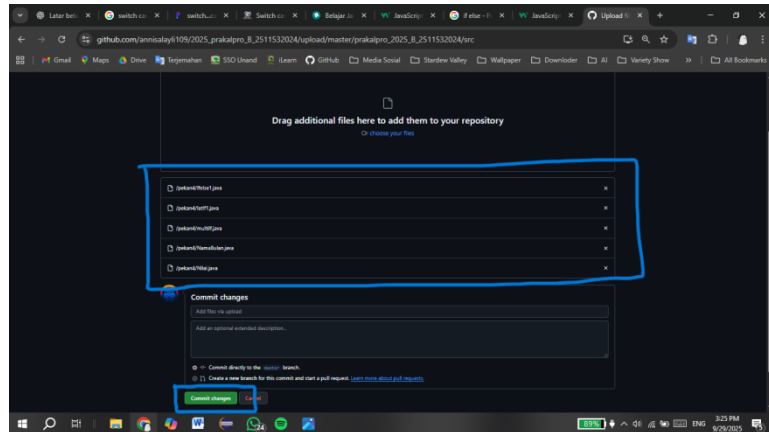
2.2.18 proses commit dan push ke GitHub

4. Kemudian buka akun Github dan buka pada bagian src. Dan klik pada bagian “add file”. Dan pilih “upload file”.



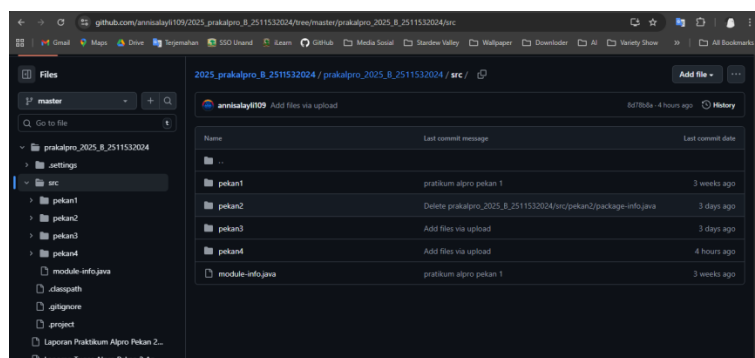
2.2.19 proses commit dan push ke GitHub

- Setelah itu tarik file “pekan4” dari file explorer ke bagian upload file dan kemudian tunggu loading file diunggah ke GitHub. Dan setelah itu klik “Commit changes”.



2.2.20 proses commit dan push ke GitHub

- Kemudian cek akun GitHub untuk melihat apakah file yang di upload sudah masuk atau belum. Jika sudah maka upload sudah selesai.



2.2.21 proses commit dan push ke GitHub

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan pembahasan mengenai kondisional statement *if-else* dan *switch case*, dapat disimpulkan bahwa keduanya merupakan struktur kendali yang sangat penting dalam pemrograman untuk mengatur alur logika program. Pernyataan *if-else* digunakan ketika program perlu mengambil keputusan berdasarkan kondisi logis yang lebih kompleks, sehingga fleksibel dalam berbagai situasi. Sedangkan *switch case* lebih efektif dan terstruktur apabila program harus menangani banyak pilihan dengan nilai diskrit tertentu, karena membuat kode lebih ringkas dan mudah dibaca.

Dengan memahami serta mampu menerapkan kedua bentuk pernyataan kondisional ini, kita dapat menulis program lebih efisien, teratur, dan sesuai dengan kebutuhan. Selain itu, penguasaan terhadap *if-else* dan *switch case* menjadi dasar penting sebelum mempelajari konsep pemrograman yang lebih lanjut seperti perulangan, fungsi, maupun struktur data.

3.2 Saran

Disarankan agar dalam menulis program, penggunaan *if-else* maupun *switch case* dipilih sesuai kebutuhan, yaitu *if-else* untuk kondisi yang lebih kompleks dan *switch case* untuk pilihan yang lebih sederhana serta terstruktur. Selain itu, sebaiknya kita terus berlatih membuat program sederhana dengan kedua struktur kondisional tersebut agar terbiasa dalam menentukan logika yang tepat. Latihan yang konsisten akan membantu meningkatkan kemampuan berpikir logis dan memudahkan dalam memahami materi pemrograman lanjutan.

DAFTAR PUSTAKA

- [1] GeeksforGeeks, “Decision Making in Java (if-else, switch, break, continue, jump),” *GeeksforGeeks*, Aug. 14, 2025. [Online]. Available: <https://www.geeksforgeeks.org/java/decision-making-javaif-else-switch-break-continue-jump>
- [2] Oracle, “The if-then and if-then-else Statements,” *Java Documentation*, 2025. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>
- [3] Oracle, “The switch Statement,” *Java Documentation*, 2025. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>