**LAPORAN PRAKTIKUM 7**



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

Nama: Gilang Shirel Paski  
NIM: 2311533017

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

**2024**

1. **Pendahuluan**
2. Tujuan Praktikum

* Memahami dasar-dasar pemrograman Java dan penerapannya dalam pembuatan program sederhana.
* Menerapkan konsep pemrograman berorientasi objek (OOP), seperti enkapsulasi melalui penggunaan getter dan setter.
* Mengimplementasikan struktur kontrol alur, seperti if-else dan switch-case, untuk menangani logika pemilihan dalam program.
* Membuat program berbasis GUI (Graphical User Interface) menggunakan library Swing untuk membangun antarmuka yang interaktif.
* Mempelajari penggunaan event-driven programming dengan menerapkan ActionListener untuk menangani aksi pengguna.
* Mengembangkan keterampilan dalam membangun, menguji, dan menjalankan program sederhana yang melibatkan operasi aritmatika dan pemrosesan data.
* Meningkatkan pemahaman dalam pembuatan program yang modular, efisien, dan mudah dipelihara dengan menerapkan prinsip-prinsip pemrograman.

1. Kajian Teori

Pemrograman Java merupakan salah satu bahasa pemrograman yang populer dan digunakan secara luas dalam pengembangan perangkat lunak modern. Java dikenal sebagai bahasa pemrograman yang berorientasi objek (Object-Oriented Programming/OOP), di mana setiap komponen dalam program dipandang sebagai objek yang memiliki atribut (data) dan metode (perilaku). Konsep utama dalam pemrograman berorientasi objek mencakup enkapsulasi, inheritance (pewarisan), polimorfisme, dan abstraksi. Enkapsulasi adalah proses membatasi akses ke atribut dan metode dalam sebuah kelas untuk menjaga integritas data. Hal ini biasanya dilakukan dengan menjadikan atribut bersifat private dan menyediakan metode getter dan setter untuk mengakses atau memodifikasi nilai atribut tersebut.

Dalam pembangunan antarmuka grafis, Java menyediakan library Swing, yang memungkinkan pengembang membuat aplikasi berbasis GUI (Graphical User Interface). Swing adalah bagian dari Java Foundation Classes (JFC) yang menawarkan komponen seperti JFrame untuk jendela utama, JTextField untuk input teks, JButton untuk tombol, serta JComboBox untuk menu pilihan. Pemrograman GUI di Java umumnya bersifat event-driven, artinya eksekusi program dipicu oleh aksi pengguna, seperti menekan tombol atau memilih item dari daftar. Untuk menangani aksi tersebut, digunakan mekanisme event listener seperti ActionListener, yang mengatur logika program ketika suatu peristiwa terjadi.

Struktur kontrol alur dalam pemrograman merupakan dasar penting dalam mengatur logika program agar berjalan sesuai tujuan. Struktur ini mencakup percabangan (decision making) seperti if-else dan switch-case, serta perulangan seperti for, while, dan do-while. Struktur switch-case digunakan untuk membuat percabangan berdasarkan nilai variabel, di mana setiap nilai akan mengeksekusi blok kode yang berbeda. Struktur ini efektif untuk situasi di mana pilihan logis memiliki rentang nilai yang terbatas dan dapat ditentukan secara eksplisit.

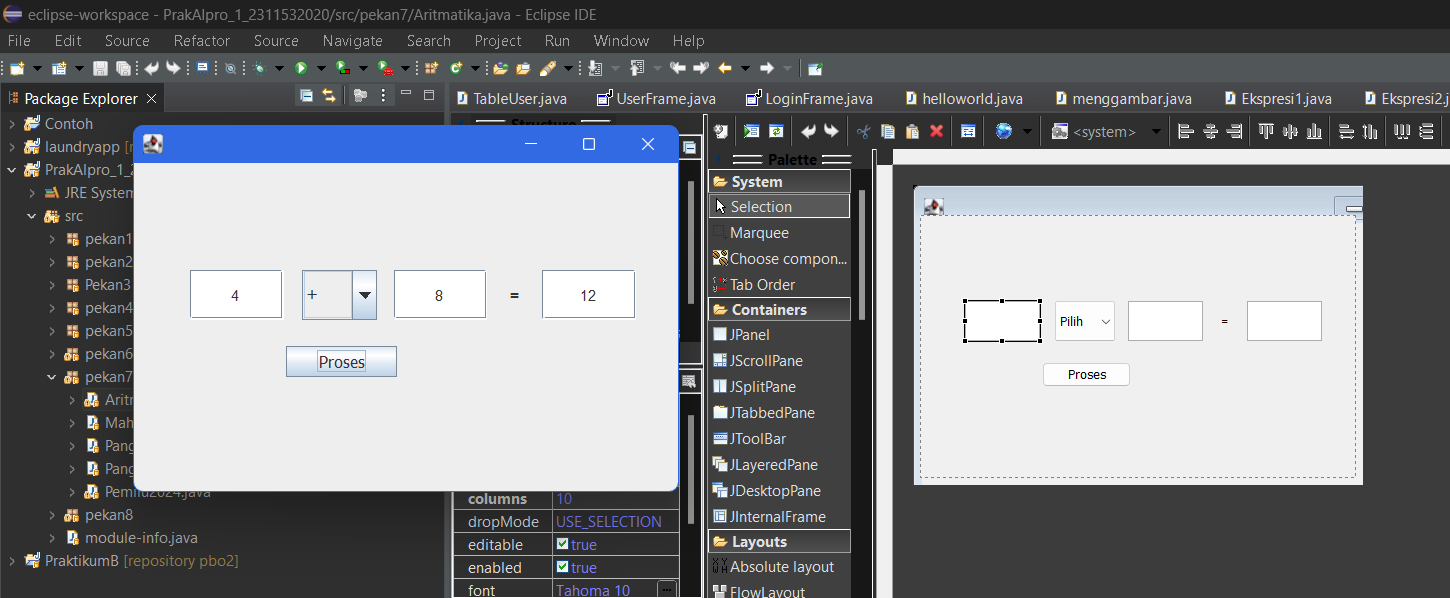
Selain itu, dalam pengolahan input dan output, Java menyediakan kelas Scanner yang digunakan untuk membaca data yang dimasukkan oleh pengguna melalui konsol. Kelas ini mendukung berbagai jenis input, seperti String, integer, float, dan tipe data lainnya. Pemrosesan data input memungkinkan program menjadi lebih interaktif, di mana pengguna dapat memberikan data secara dinamis dan program merespons sesuai logika yang telah ditentukan.

Dalam studi kasus pembuatan kelas Mahasiswa, prinsip dasar OOP seperti enkapsulasi dan pemanggilan metode telah digunakan. Metode getter dan setter memungkinkan data dapat diakses secara aman dan terstruktur. Selain itu, program ini menekankan pentingnya penggunaan kelas sebagai wadah untuk merepresentasikan objek-objek nyata, seperti mahasiswa dengan atribut NIM dan nama. Pendekatan ini membantu dalam mengorganisir kode menjadi lebih modular, sehingga mudah dipahami, diubah, dan dipelihara.

Penerapan operasi aritmatika melalui antarmuka grafis dalam program Aritmatika menunjukkan bagaimana input pengguna dapat diproses secara real-time menggunakan komponen GUI dan logika aritmatika sederhana. Operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus dilakukan dengan memanfaatkan input dari JTextField dan pilihan dari JComboBox, kemudian hasilnya ditampilkan kembali dalam komponen output. Konsep ini mengajarkan bagaimana antarmuka pengguna dapat memfasilitasi proses perhitungan sederhana dengan pendekatan yang lebih visual dan ramah pengguna.

Dalam studi kasus Pemilu2024, implementasi struktur switch-case menunjukkan bagaimana program dapat memberikan respon yang berbeda berdasarkan input numerik dari pengguna. Hal ini memberikan pemahaman mendalam tentang penggunaan kontrol alur untuk memecahkan masalah yang memerlukan pemilihan logis. Selain itu, pengolahan input menggunakan kelas Scanner dan penanganan nilai yang tidak valid melalui blok default memperlihatkan pendekatan yang sistematis dalam validasi data pengguna.

1. **Pembahasan**
2. GUI Aritmatika



Kode Java ini merupakan implementasi antarmuka grafis pengguna (GUI) sederhana menggunakan library Swing untuk melakukan operasi aritmatika dasar, seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus. Program ini berfungsi menerima dua input bilangan bulat melalui dua komponen JTextField dan menampilkan hasil perhitungan di bidang output (JTextField) setelah pengguna memilih operasi dari daftar pilihan (JComboBox) dan menekan tombol "Proses".

Komponen GUI diatur dalam metode initialize() di dalam kelas Aritmatika. Komponen utama meliputi tiga JTextField (A, B, dan C), satu JComboBox untuk memilih operasi aritmatika, dan satu tombol JButton sebagai pemicu proses perhitungan. Metode ActionListener di dalam tombol "Proses" mengatur logika perhitungan berdasarkan nilai yang dipilih di JComboBox. Indeks operasi ditentukan menggunakan metode getSelectedIndex() dari JComboBox, di mana indeks 1 hingga 5 digunakan untuk masing-masing operasi (+, -, \*, /, %). Selanjutnya, nilai dari JTextField input (A dan B) dikonversi menjadi tipe data integer menggunakan Integer.parseInt() dan hasilnya ditampilkan kembali dalam bidang output JTextField C menggunakan setText().

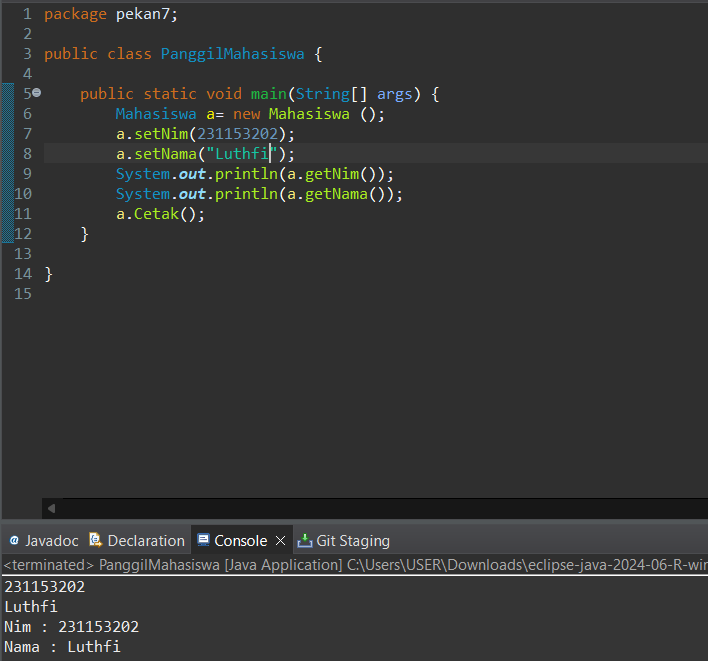
Program ini dijalankan melalui metode main, di mana EventQueue digunakan untuk memastikan thread-safe dalam menjalankan GUI. Kerangka utama GUI ditampilkan menggunakan JFrame dengan tata letak absolut (setLayout(null)) dan ukuran frame tetap (setBounds). Dengan demikian, program ini memungkinkan pengguna untuk melakukan perhitungan aritmatika dasar dengan antarmuka yang sederhana dan interaktif.

1. Class Mahasiswa

Kode Java ini mendefinisikan sebuah kelas bernama Mahasiswa yang berfungsi sebagai representasi dari objek mahasiswa dengan atribut utama berupa NIM (Nomor Induk Mahasiswa) dan nama mahasiswa. Kelas ini menggunakan prinsip enkapsulasi, di mana variabel global nim, nim2, dan nama dideklarasikan sebagai private sehingga hanya dapat diakses melalui metode getter dan setter yang disediakan. Metode setter (mutator) seperti setNim(int nim), setNim2(String nim2), dan setNama(String nama) digunakan untuk mengatur nilai variabel nim, nim2, dan nama. Sedangkan metode getter (accessor) seperti getNim(), getNim2(), dan getNama() memungkinkan pengambilan nilai dari variabel-variabel tersebut.

Selain itu, kelas ini juga memiliki dua metode tambahan, Cetak() dan Cetak2(), yang digunakan untuk mencetak informasi mahasiswa ke konsol. Metode Cetak() menampilkan NIM dalam format integer (nim) dan nama mahasiswa, sedangkan metode Cetak2() menampilkan NIM dalam format string (nim2) dan nama mahasiswa. Perbedaan antara kedua metode ini terletak pada jenis data NIM yang digunakan. Kode ini memberikan fleksibilitas dalam menangani variasi format data NIM, yang dapat berupa integer maupun string. Dengan penggunaan metode getter dan setter, kelas Mahasiswa mengimplementasikan prinsip data hiding dalam pemrograman berorientasi objek (OOP), yang memastikan bahwa data hanya dapat diakses atau dimodifikasi melalui metode yang telah ditentukan. Ini membantu menjaga integritas data dan memisahkan logika pengolahan data dari akses langsung terhadap variable

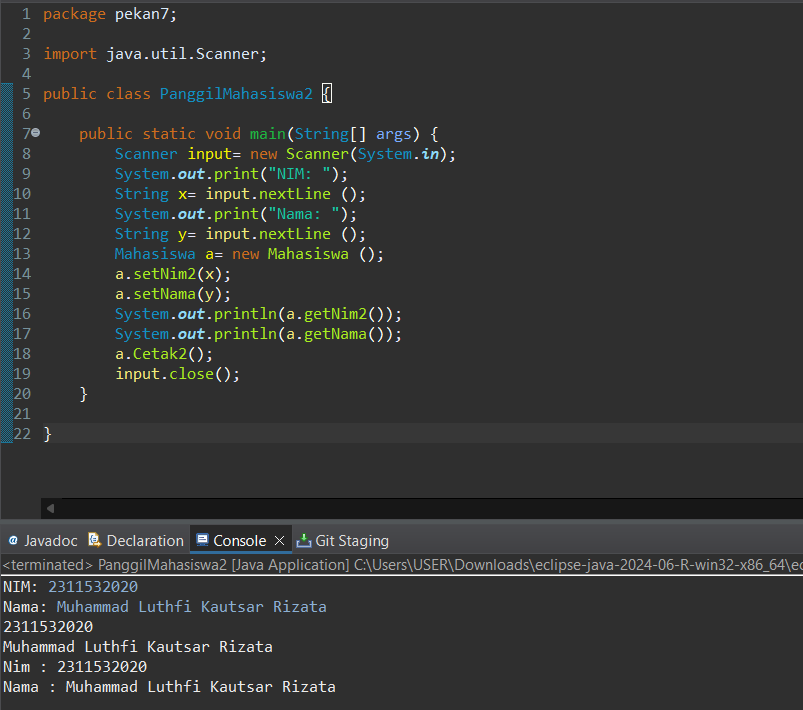
1. Class PanggilMahasiswa



Kode program di atas merupakan implementasi sederhana dari permainan lempar dadu yang bertujuan untuk mendapatkan jumlah angka dadu sebesar tujuh. Program ini dimulai dengan mengimpor kelas Random dari pustaka Java, yang digunakan untuk menghasilkan angka acak. Di dalam metode main, objek Random dibuat untuk menghasilkan angka acak yang merepresentasikan hasil lemparan dua dadu. Program kemudian menginisialisasi dua variabel, yaitu tries untuk menghitung jumlah percobaan lemparan dan sum untuk menyimpan hasil penjumlahan dari kedua dadu.

Selama jumlah dadu tidak sama dengan tujuh, program akan terus melakukan lemparan dadu. Setiap kali dadu dilempar, dua angka acak dihasilkan, masing-masing untuk dadu pertama dan kedua, dengan rentang dari 1 hingga 6. Hasil dari kedua dadu tersebut dijumlahkan dan ditampilkan ke layar. Setelah mendapatkan jumlah tujuh, program akan keluar dari loop dan menampilkan pesan yang menunjukkan berapa kali percobaan dilakukan sebelum mencapai hasil yang diinginkan. Program ini mencerminkan penggunaan kontrol alur dengan loop while dan pemanfaatan kelas Random untuk menghasilkan angka acak, serta memberikan gambaran sederhana tentang bagaimana simulasi permainan dapat diterapkan dalam pemrograman Java.

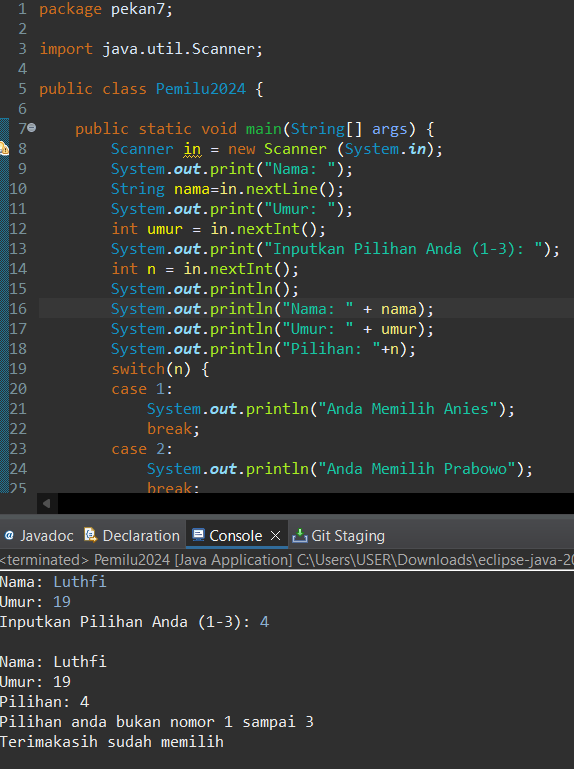
1. Class PanggilMahasiswa2



Kode Java ini berfungsi sebagai program utama untuk mengakses dan memanipulasi data dari kelas Mahasiswa. Dalam program ini, class Scanner digunakan untuk menerima input dari pengguna melalui konsol. Program meminta pengguna untuk memasukkan NIM (Nomor Induk Mahasiswa) dalam bentuk string dan nama mahasiswa, yang kemudian disimpan dalam variabel x dan y. Selanjutnya, sebuah objek a dari kelas Mahasiswa dideklarasikan dan digunakan untuk memanggil metode setter setNim2() dan setNama() guna menyimpan nilai NIM dan nama ke dalam atribut kelas Mahasiswa.

Untuk mengakses dan menampilkan data yang telah dimasukkan, program memanggil metode getter getNim2() dan getNama() yang mengembalikan nilai NIM dan nama mahasiswa ke output konsol. Selain itu, metode Cetak2() dari kelas Mahasiswa dipanggil untuk menampilkan nilai NIM (dalam format string) dan nama mahasiswa dalam format yang telah ditentukan.

Program ini menunjukkan bagaimana prinsip enkapsulasi dalam pemrograman berorientasi objek (OOP) diimplementasikan, di mana data diatur melalui penggunaan metode setter dan getter untuk menjaga akses yang terkontrol terhadap variabel privat. Penggunaan Scanner untuk menerima input pengguna memperlihatkan interaktivitas program, dan pemanggilan metode Cetak2() memastikan data ditampilkan secara terstruktur di konsol. Secara keseluruhan, program ini mengilustrasikan praktik pemrograman yang baik dalam mengelola data melalui objek dan metode dalam paradigma OOP.

1. Class Pemilu2024

Kode Java ini adalah program sederhana berbasis konsol yang digunakan untuk **simulasi pemilihan umum (Pemilu)**. Program meminta input dari pengguna berupa **nama**, **umur**, dan pilihan kandidat yang tersedia (nomor 1-3). Input dimasukkan melalui kelas **Scanner**, yang berfungsi untuk membaca data dari pengguna. Variabel nama menyimpan input nama pengguna sebagai tipe data String, sementara variabel umur dan n digunakan untuk menyimpan umur dan pilihan kandidat sebagai tipe data int.

Program kemudian menampilkan kembali data pengguna dengan mencetak nama, umur, dan pilihan ke layar. Proses pengambilan keputusan kandidat dilakukan menggunakan struktur kontrol **switch-case**, di mana setiap nilai dari variabel n akan menghasilkan output yang berbeda. Jika pengguna memilih angka **1**, maka program mencetak "Anda Memilih Anies". Pilihan **2** mencetak "Anda Memilih Prabowo", dan pilihan **3** mencetak "Anda Memilih Ganjar". Apabila input tidak sesuai dengan rentang angka 1-3, blok **default** akan menampilkan pesan "Pilihan anda bukan nomor 1 sampai 3".

Penggunaan switch-case dalam program ini memperjelas alur logika pemilihan berdasarkan input pengguna. Selain itu, program menyertakan pesan penutup "Terimakasih sudah memilih" sebagai bentuk penyelesaian eksekusi. Program ini merupakan contoh penerapan dasar **kontrol alur** dalam pemrograman Java yang memadukan penggunaan input dari pengguna, percabangan keputusan, dan pencetakan output yang sesuai.

1. **Kesimpulan**

Dalam praktikum Algoritma Pemrograman ini, telah dikembangkan beberapa program sederhana berbasis bahasa pemrograman Java yang berfokus pada implementasi konsep dasar pemrograman, seperti struktur kontrol alur, input/output data, dan pemrograman berorientasi objek (OOP). Program-program ini memberikan pemahaman yang lebih mendalam mengenai bagaimana data diolah, bagaimana logika program dijalankan, serta bagaimana antarmuka grafis sederhana dapat dibuat untuk meningkatkan interaksi dengan pengguna. Praktikum ini mencakup tiga studi kasus utama, yakni program Aritmatika, program Mahasiswa, dan program Pemilu2024.

Pada program Aritmatika, antarmuka grafis dibuat menggunakan library Swing, yang merupakan bagian dari Java untuk membangun GUI (Graphical User Interface). Program ini memungkinkan pengguna untuk memasukkan dua buah bilangan, memilih operasi aritmatika yang diinginkan (seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus), serta menampilkan hasil perhitungan. Antarmuka terdiri dari beberapa komponen utama, seperti JTextField untuk input bilangan, JComboBox untuk memilih operasi, dan JButton sebagai tombol untuk memicu proses perhitungan. Logika program ini dijalankan melalui metode ActionListener, di mana setiap kali tombol "Proses" ditekan, program akan membaca input dari pengguna, memproses data sesuai dengan operasi yang dipilih, dan menampilkan hasilnya dalam komponen output. Program ini menunjukkan bagaimana event-driven programming digunakan dalam Java dan bagaimana antarmuka grafis dapat dibuat secara sederhana namun fungsional.

Selanjutnya, dalam kelas Mahasiswa dan program PanggilMahasiswa2, prinsip dasar pemrograman berorientasi objek (OOP) diterapkan dengan menggunakan konsep enkapsulasi. Kelas Mahasiswa mendefinisikan atribut privat berupa NIM dan nama mahasiswa, yang hanya dapat diakses atau dimodifikasi melalui metode getter dan setter. Konsep ini merupakan implementasi dari data hiding, di mana akses langsung terhadap variabel privat dihindari demi menjaga integritas data. Selain itu, kelas Mahasiswa juga memiliki dua metode tambahan, yaitu Cetak() dan Cetak2(), yang digunakan untuk mencetak data mahasiswa ke konsol dalam format yang berbeda. Metode ini menunjukkan fleksibilitas dalam menampilkan data dengan variasi format. Program PanggilMahasiswa2 kemudian digunakan untuk menguji kelas Mahasiswa. Program ini mengambil input NIM dan nama mahasiswa dari pengguna menggunakan Scanner, menyimpan data tersebut melalui metode setter, dan menampilkannya kembali melalui metode getter dan metode Cetak2(). Melalui studi kasus ini, prinsip OOP seperti enkapsulasi dan pemanggilan metode dalam objek dapat dipahami dengan lebih baik, yang merupakan salah satu pilar utama dalam pengembangan perangkat lunak modern.

Pada program Pemilu2024, struktur kontrol alur berbasis switch-case diterapkan untuk menentukan kandidat yang dipilih oleh pengguna. Program ini meminta input berupa nama, umur, dan pilihan kandidat (dalam bentuk angka 1-3). Berdasarkan input pengguna, program menampilkan output yang sesuai, seperti "Anda Memilih Anies", "Anda Memilih Prabowo", atau "Anda Memilih Ganjar". Jika pengguna memasukkan angka di luar rentang 1-3, program akan mengeksekusi blok default untuk menampilkan pesan "Pilihan anda bukan nomor 1 sampai 3". Selain itu, program ini memastikan informasi input pengguna, seperti nama dan umur, dicetak kembali ke layar sebagai bentuk validasi. Studi kasus ini menunjukkan bagaimana struktur kontrol alur, seperti switch-case, digunakan untuk menangani percabangan logika yang sederhana namun efektif dalam pemrograman.

Secara keseluruhan, praktikum ini memberikan pemahaman mendalam tentang konsep dasar pemrograman Java dan penerapannya dalam menyelesaikan masalah praktis. Program Aritmatika menunjukkan cara membangun antarmuka grafis menggunakan Swing dan bagaimana event-driven programming dapat memicu proses perhitungan sesuai interaksi pengguna. Program Mahasiswa dan PanggilMahasiswa2 menekankan pentingnya penggunaan prinsip enkapsulasi dalam pemrograman berorientasi objek, di mana data dikelola melalui metode setter dan getter untuk menjaga keamanan serta keteraturan pengolahan data. Sementara itu, program Pemilu2024 memperkenalkan struktur kontrol alur switch-case sebagai solusi untuk menangani pemilihan berdasarkan input pengguna.

Melalui praktikum ini, keterampilan dasar dalam memproses input/output data, menerapkan struktur kontrol alur, serta membangun program berbasis objek telah dipelajari dengan baik. Selain itu, penggunaan GUI sederhana memberikan wawasan tambahan tentang bagaimana program dapat dibuat lebih interaktif dan mudah digunakan oleh pengguna. Dengan memahami konsep-konsep tersebut, mahasiswa diharapkan mampu mengembangkan program yang lebih kompleks dan efisien di masa mendatang dengan memanfaatkan prinsip pemrograman Java yang telah dipelajari.