|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| Nadya Putri Marsela  G1F024036  Karina Hodiyah Ramadona  G1F024040 | Operator Aritmatika | 17 September 2024 |
| **[No.1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable 2. Bila kalian kuliah selama 4 tahun. Susunlah kode java untuk menghitung jumlah yang harus dibayar selama kuliah untuk setiap anggota kelompok! (Asumsi: setiap orang memiliki jumlah yang dibayarkan berbeda karena perbedaan jalur masuk, UKT, SPP 3. Rincikan sumber informasi yang relevan (buku / webpage) 4. Video penjelasan pembelajaran dapat diakses pada Chanel Youtube Rumah Ilmu Raflesia <https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMyIw>  [Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman](https://www.youtube.com/watch?v=PzCMZObexZM) 5. <https://www.youtube.com/watch?v=PzCMZObexZM>  [Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman](https://www.youtube.com/watch?v=LcFgl0yrKEw) <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[No.1] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan. 2. Konsistensi dalam penamaan variabel:   Ubah nama variabel agar menggunakan gaya penulisan camelCase, misalnya Uangpangkal menjadi uangPangkal. Ini membuat kode lebih mudah dibaca dan mengikuti standar penamaan di Java.   1. Format tampilan uang:   Gunakan NumberFormat untuk memformat angka yang berkaitan dengan uang. Ini akan menambahkan pemisah ribuan, sehingga angka seperti 3440000 akan ditampilkan sebagai Rp.3.440.000.   1. Perhitungan total biaya 4 tahun:   Hitung total biaya pendidikan selama 4 tahun dengan cara yang sudah benar, tetapi tambahkan format uang agar hasil perhitungannya tampil rapi, dengan pemisah ribuan dan mata uang.   1. Fungsi modular untuk menampilkan data:   Pisahkan bagian program yang menampilkan data tiap anggota ke dalam fungsi terpisah. Ini membuat kode lebih rapi, mudah dimengerti, dan lebih mudah dikelola jika ingin menambah atau mengubah cara data ditampilkan.   1. Struktur kode yang lebih teratur:   Dengan pemisahan fungsi dan penggunaan format yang jelas, program akan lebih terstruktur, sehingga jika ada perubahan atau penambahan anggota tidak perlu memodifikasi banyak bagian dalam program.   1. Analisis solusi, kaitkan dengan permasalahan.   1. Penamaan variabel yang konsisten  Variabel Uangpangkal tidak mengikuti aturan penulisan yang konsisten dapat diubah menjadi uangPangkal agar sesuai dengan standar Java.  2. Format tampilan uang  Angka uang seperti 3440000 ditampilkan tanpa pemisah ribuan, sulit dibaca sehingga menggunakan NumberFormat untuk menambahkan pemisah ribuan.  3. Perhitungan total biaya 4 tahun  Perhitungan sudah benar, tapi tampilannya kurang rapi sehingga format hasil perhitungan menggunakan NumberFormat.  4. Modularitas kode  Kode tidak terstruktur karena semua operasi ada di satu perulangan, pisahkan logika tampilan data anggota ke dalam fungsi terpisah. | | |
| **[No.1] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma 2. Deklarasikan nilai nya 3. Lakukan looping 4. Menghitung total biaya yang harus dibayar selama 4 tahun 5. Menampilkan luaran 6. Selesai. 7. Tuliskan kode program dan luaran     **Luaran** | | |
| **[No.1] Kesimpulan** | | |
| 1. Evaluasi 2. Apa konsekuensi dari skenario pemprograman ini?   Konsekuensi utama dari skenario pemrograman di atas adalah bahwa tanpa perbaikan dalam hal penamaan variabel, modularitas, dan format tampilan, program mungkin akan sulit dipahami, dipelihara, dan digunakan oleh pengguna. Dengan perbaikan yang diajukan, kode akan lebih profesional, mudah dipelihara, dan memberikan pengalaman pengguna yang lebih baik. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| Nadya Putri Marsela  G1F024036  Karina Hodiyah Ramadona  G1F024040 | Operator Aritmatika | 17 September 2024 |
| **[No.2] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   2. Gunakan operator ternary. Rancanglah kode Java untuk menghitung jumlah yang harus dibayar jika kalian bisa selesai 4 tahun atau jika selesai 5 tahun!   1. Rincikan sumber informasi yang relevan (buku / webpage)  * <https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMyIw>  [Video Materi 1 – Definisi operator, unary, binary, ternary, operasi aritmatika, penugasan, relasional, increment/decrement pada laman](https://www.youtube.com/watch?v=PzCMZObexZM) * <https://www.youtube.com/watch?v=PzCMZObexZM>  [Video Materi 2 – Operator Logika, Kondisional, Bitwise, dan contoh pembahasan soal pada laman](https://www.youtube.com/watch?v=LcFgl0yrKEw) <https://www.youtube.com/watch?v=LcFgl0yrKEw> | | |
| **[No.2] Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Berikut adalah uraian rancangan solusi yang diusulkan dari kode \*\*OperatorTernary\*\* dengan bahasa yang lebih sederhana dan manusiawi:  1. Penggunaan Operator Ternary untuk Menentukan Total Biaya  Kode ini menggunakan operator ternary untuk menentukan apakah seorang mahasiswa menjalani perkuliahan selama 4 atau 5 tahun, dan menghitung total biaya pendidikan berdasarkan lamanya studi. Jika lama kuliah 4 tahun, program menghitung biaya 8 semester. Jika 5 tahun, program menghitung biaya 10 semester.  Dengan operator ternary, kode menjadi lebih singkat dan efisien. Tanpa harus menggunakan pernyataan `if-else`, program dapat langsung memutuskan perhitungan mana yang akan digunakan berdasarkan lama kuliah setiap anggota.  2. Penggunaan Array untuk Data Mahasiswa  Kode menyimpan data anggota berupa nama, NPM, uang pangkal, UKT, dan lama kuliah dalam array, lalu mengaksesnya menggunakan perulangan.  Dengan menyimpan data dalam array, kita dapat dengan mudah menambah atau mengurangi data tanpa harus mengubah banyak kode. Penggunaan perulangan juga memungkinkan program untuk menangani banyak mahasiswa tanpa harus menulis baris kode yang berulang-ulang.  3. Memisahkan Biaya Berdasarkan Lama Studi  Kode menghitung dua kemungkinan total biaya kuliah: satu untuk 4 tahun (8 semester) dan satu lagi untuk 5 tahun (10 semester). Kemudian, menggunakan operator ternary untuk memilih perhitungan yang sesuai.  Memisahkan logika biaya berdasarkan lama kuliah memberikan fleksibilitas pada kode, sehingga program dapat menangani berbagai skenario lama kuliah dengan cara yang efisien. Jika nanti ada anggota dengan lama kuliah yang berbeda (misalnya 3 atau 6 tahun), program dapat dengan mudah diadaptasi.  4. Menggunakan Perulangan untuk Menampilkan Data  Kode menggunakan perulangan `for` untuk menampilkan data setiap mahasiswa, seperti nama, NPM, lama kuliah, dan total biaya.  Dengan perulangan, program dapat dengan mudah menangani beberapa mahasiswa tanpa perlu menulis kode secara manual untuk setiap orang. Ini membuat kode lebih singkat dan lebih mudah diubah jika data mahasiswa bertambah.  5. Penanganan Data yang Tidak Konsisten  Ada inkonsistensi pada ukuran array, misalnya array `Uangpangkal` memiliki tiga elemen sedangkan array lainnya hanya memiliki dua elemen.  Pastikan semua array memiliki jumlah elemen yang sama untuk menghindari kesalahan saat mengakses data. Misalnya, ubah array `Uangpangkal` menjadi `{0, 0}` agar sesuai dengan jumlah anggota lainnya.   1. Analisis solusi, kaitkan dengan permasalahan. 2. Ukuran Array yang Tidak Sama 3. Menghitung Total Biaya Berdasarkan Lama Kuliah 4. Menggunakan Array untuk Menyimpan Data Mahasiswa 5. Tampilan Biaya yang Lebih Jelas 6. Pemeliharaan Kode | | |
| **[No.2] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma    1. Deklarasikan nilai nya    2. Lakukan looping    3. Menghitung total biaya yang harus dibayar selama 4 tahun dan atau 5 tahun    4. Menampilkan luaran    5. Selesai. 2. Tuliskan kode program dan luaran     **Luaran** | | |
| **[No.2] Kesimpulan** | | |
| 1) Evaluasi  Apa konsekuensi dari skenario pemprograman ini?   * Error dan ketidakstabilan program akan terjadi jika ukuran array tidak sama. * Kurang efisien jika logika pengambilan keputusan dilakukan dengan if-else yang panjang. * Tampilan biaya yang tidak jelas dapat mengganggu pengalaman pengguna. * Kode yang tidak modular akan sulit dipelihara dan dikembangkan di masa depan. | | |
|  | | |