|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **WINDA DWI SALISTIANNA**  **G1F024004** | **IF DAN SWICH** | **23 SEPTEMBER 2024** |
| **UNIT 1 IF** | | |
| **Latihan 1:**  1.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!  1.2. Cermati contoh 2, analisa kondisi pada IF bersarang!          Tambahkan satu kondisi IF dengan satu nilai input Quiz (nilaiQ). Jika nilai UTS, Tugas, dan Quiz lebih besar sama dengan 80        maka siswa mendapat nilai A.        Apakah ketiga kondisi IF pada Contoh 1.2. dapat diringkas menjadi satu kondisi?        Rancanglah kode yang dapat menggantikan ketiga kondisi tersebut menggunakan          operator boolen ( && atau ||) !  1.3.  Konstruksikan kode program untuk menghasilkan luaran berdasarkan informasi berikut!         Persyaratan kelulusan bagi mahasiswa Informatika dan Sistem Informasi Universitas Bengkulu yaitu:        a)    Jumlah sks yang lulus minimum = 144 sks. Jika kurang dari 144 maka perlu mengulang mengambil mata kuliah hingga memenuhi        b)    Nilai tes toefl minimum = 430. Jika skor belum mencapai maka ulangi tes toefl maksimal 3x atau hingga skor tercapai        c)    Ujian skripsi = Lulus, jika belum lulus maka perlu ujian ulang.         d)    Menulis artikel ilmiah dari skripsi yang dibuat dalam status = submitted. Jika bukti penerimaan submitted tidak dilampirkan         maka persyaratan belum terpenuhi.         Jika keempat syarat ini terpenuhi maka Mahasiswa layak lulus dengan gelar S.Kom. (Sarjana Komputer).  1.4.    Desain gambar diagram flowchart dari Latihan 1.3!  JAWABAN  **Latihan 1.1 contoh 1**  Kode contoh 1 memiliki beberapa kesalahan yang menyebabkan error saat kompilasi. Kesalahan-kesalahan tersebut antara lain:   * Variabel nilai tidak dideklarasikan: Variabel nilai digunakan sebelum dideklarasikan, sehingga compiler tidak mengenali variabel tersebut. * Objek Scanner salah nama: Objek Scanner yang digunakan untuk membaca input pengguna dinamakan masuk, bukan input. * Operator perbandingan salah: Penggunaan = pada kondisi if sebenarnya adalah operator assignment (penugasan), bukan operator perbandingan (sama dengan). Untuk membandingkan nilai, harus menggunakan == .   Kode yang benar :    *Gambar 1.1 codingan if*  Penjelasan kode:  Kode pada gambar 2.1 meminta pengguna untuk memasukkan sebuah angka. Angka yang dimasukkan akan disimpan dalam variabel nilai. Kemudian, kode akan memeriksa apakah nilai nilai sama dengan 1000. Jika ya, maka akan dicetak "Seribu", jika tidak, maka akan dicetak "Nilai Bukan Seribu".  Output:  Sebuah gambar berisi teks, cuplikan layar, Font  Deskripsi dibuat secara otomatis  Gambar 1.2 output if  **Contoh 2**  Kode contoh 1 memiliki beberapa kesalahan yang menyebabkan error saat kompilasi. Kesalahan-kesalahan tersebut antara lain:   * **Variabel nilaiU tidak didefinisikan:** Kode mencoba mengakses variabel nilaiU pada kondisi if (nilaiU >= 80), namun variabel ini tidak pernah dideklarasikan. * **Penggunaan Scanner yang berlebihan:** Dibuat dua objek Scanner yang berbeda (varT dan varQ) untuk membaca input, padahal satu objek Scanner sudah cukup. * **Logika percabangan:** Logika percabangan hanya memeriksa nilai tugas. Nilai kuis tidak digunakan dalam perhitungan nilai akhir.   Kode yang benar :  Sebuah gambar berisi teks, cuplikan layar, Font, nomor  Deskripsi dibuat secara otomatis  *Gambar 1.3 kode if bersarang*  Penjelasan kode :  Kode di atas akan meminta pengguna untuk memasukkan nilai tugas dan nilai quiz. Kemudian, nilai tugas dan nilai quiz akan dirata-ratakan untuk mendapatkan nilai akhir. Jika nilai akhir lebih besar atau sama dengan 80, maka akan dicetak "Anda mendapatkan nilai A", jika tidak, maka akan dicetak "Anda TIDAK mendapatkan nilai A".  Satu Objek **Scanner** Digunakan satu objek Scanner untuk membaca semua input, yaitu input.Penamaan Variabel Nama variabel diganti menjadi lebih deskriptif, yaitu nilaiTugas dan nilaiQuiz.Perhitungan Nilai Akhir: Ditambahkan perhitungan nilai akhir dengan cara menjumlahkan nilai tugas dan nilai quiz, kemudian dibagi dua untuk mendapatkan rata-rata.Kondisi **if:** Kondisi if kini memeriksa nilai akhir, bukan hanya nilai tugas.  Output :  Sebuah gambar berisi teks, cuplikan layar, Font, garis  Deskripsi dibuat secara otomatis  *Gambar 1.4 output if bersarang*  **Latihan 1.2** Menambahkan kondisi UTS: Diperlukan input baru untuk nilai UTS dan kondisi tambahan untuk memeriksa apakah nilai UTS juga lebih besar sama dengan 80. Menggabungkan kondisi: Ketiga kondisi (nilaiT, nilaiQ, dan nilaiU) dapat digabungkan menjadi satu kondisi menggunakan operator logika AND (&&). Menggunakan operator boolean: Operator && akan mengembalikan nilai true jika semua kondisi yang dihubungkan olehnya bernilai true.  Sebuah gambar berisi teks, cuplikan layar, Font, nomor  Deskripsi dibuat secara otomatis  *Gambar 1.5 code if*  Penjelasan :  Pada kode gambar 1.5 memeriksa apakah nilai tugas, quiz, dan UTS semuanya memenuhi syarat (lebih besar sama dengan 80). Jika ketiga syarat terpenuhi, maka siswa akan mendapatkan nilai A. Penggunaan operator && memastikan bahwa semua kondisi harus benar agar keseluruhan kondisi bernilai benar. Input Nilai UTS : Ditambahkan input untuk nilai UTS.Kondisi Gabungan: Digunakan operator && untuk menggabungkan ketiga kondisi. Kondisi ini akan bernilai true jika nilai tugas, quiz, dan UTS semuanya lebih besar atau sama dengan 80.  Output :  *Gambar 1.6 Output if*  **Sebuah gambar berisi teks, cuplikan layar, Font, nomor  Deskripsi dibuat secara otomatisLatihan 1.3**  *Gambar 1.6 kodingan*  Penjelasan kode:  Kode di atas memberikan solusi yang sederhana dan efektif untuk memeriksa kelulusan mahasiswa berdasarkan persyaratan yang diberikan. Dengan menggunakan variabel boolean dan operator logika, kita dapat dengan mudah mengevaluasi semua kondisi dan menentukan apakah seorang mahasiswa layak lulus atau tidak.   * Input Data: Program meminta pengguna untuk memasukkan jumlah SKS, skor TOEFL, status ujian skripsi, dan status artikel ilmiah. * Variabel Boolean: Setiap kondisi diwakili oleh sebuah variabel boolean. Variabel ini akan bernilai true jika kondisi terpenuhi dan false jika tidak. * Kondisi Kelulusan: Menggunakan operator && untuk memeriksa apakah semua kondisi bernilai true. Jika ya, maka mahasiswa dinyatakan lulus.   **1.4 FLOWCHART**  START  Input jumlah sks  Input skor toefl  Input status ujian skripsi  Input status artikel  If lulusSKS && lulusToefl&& lulusSkrippsi && lulusArtikel  *false*  Mahasiswa belum memenuhi syarat kelulusan  *True*  Mahasiswa layak lulus dengan gelar S.Kom  Selesai | | |
| **UNIT 2 SWITCH** | | |
| **Latihan 2:**  2.1. Cermati kode pada Contoh 3.         Evaluasi penyebab kesalahan dan perbaiki kode tersebut!        Hapuslah kode break; pada //baris 1, lalu eksekusi kembali.         Kemudian hapuslah kode break; pada //baris 2, lalu eksekusi kembali.        Simpulkan kegunaan break pada switch!  2.2. Cermati kode pada Contoh 4. Evaluasi apakah penulisan kode tersebut sudah efisien?         Apakah ada penulisan informasi yang diulangi?        Jika ada, susun kembali penulisan kode yang tepat!        Simpulkan perbedaan antara kode Contoh 4 dengan kode yang kalian susun!  2.3. Cermati permasalahan yang dipecahkan pada Contoh 3.         Apakah masalah ini bisa diubah menjadi perintah IF?        Jika bisa, susun kembali kode Contoh 3 dari perintah SWITCH menjadi IF!       Simpulkan perbandingan masalah dan kode program yang dapat diselesaikan percabangan dengan IF atau  SWITCH !  2.4. Desain gambar flowchart dari Latihan 2.1. dan Latihan 2.3!  JAWABAN  **Latihan 2.1** Evaluasi dan Perbaikan Kode pada Contoh 3  Kode yang ada pada Contoh 3 mengandung beberapa kesalahan, antara lain:   1. Pada baris char data = data.next().charAt(0); seharusnya menggunakan objek masukData, bukan data, karena objek data belum dideklarasikan sebelumnya. Jadi seharusnya: char data = masukData.next().charAt(0); 2. Pada perintah switch, terdapat kesalahan penulisan tanda titik dua (:) setelah switch(data). Seharusnya, tanda kurung kurawal { digunakan untuk memulai blok switch. Jadi: switch(data) { 3. Pada perintah case A, seharusnya 'A' ditulis dalam tanda petik tunggal karena 'A' merupakan karakter, bukan variabel. Jadi: case 'A': 4. Pada blok default, kurang tanda titik dua setelah kata default. Seharusnya: default:   Sebuah gambar berisi teks, cuplikan layar, Font, nomor  Deskripsi dibuat secara otomatisKode yang benar :  *Gambar 2.1 code switch*  Penjelasan kode:  Kode ini mendefinisikan sebuah program Java sederhana yang menggunakan struktur untuk membuat keputusan berdasarkan input pengguna. Program ini dimulai dengan deklarasi paket , yang berfungsi untuk mengorganisasi kelas-kelas dalam suatu proyek. Kemudian, kelas didefinisikan sebagai wadah utama untuk kode program ini. Di dalam kelas ini, terdapat metode yang merupakan titik awal eksekusi program.  Sebuah gambar berisi teks, cuplikan layar, Font, garis  Deskripsi dibuat secara otomatisDalam metode , program meminta pengguna untuk memilih antara karakter 'A' atau 'B'. Input pengguna kemudian disimpan dalam variabel bertipe karakter. Selanjutnya, struktur digunakan untuk mengevaluasi nilai variabel . Jika nilai adalah 'A', maka pesan "Anda sudah rajin belajar" akan ditampilkan. Jika nilai adalah 'B', maka pesan "Anda perlu kurangi main game" akan ditampilkan. Jika nilai tidak sesuai dengan 'A' atau 'B', maka pesan default akan ditampilkan.  *Gambar 2.2 output switch*  Sebuah gambar berisi teks, cuplikan layar, Font, garis  Deskripsi dibuat secara otomatisHasil output jika break baris satu di hapus :  *Gambar 2.3 output*  Jadi **Break** berguna untuk menghentikan eksekusi dalam blok switch setelah suatu kondisi terpenuhi dan mencegah program menjalankan kondisi lain yang tidak seharusnya.  **Latihan 2.2**  Penulisan kode pada Contoh 4 dapat dianggap kurang efisien karena banyak kode yang diulangi, khususnya dalam penentuan jumlah hari pada bulan tertentu yang jumlah harinya sama.  Kode yang disusun :  Sebuah gambar berisi teks, cuplikan layar, Font, nomor  Deskripsi dibuat secara otomatis  *Gambar 2.4 codingan switch*  Output :  Sebuah gambar berisi teks, cuplikan layar, Font, garis  Deskripsi dibuat secara otomatis  *Gambar 2.5 output*  Perbedaan Contoh 4 dan Kode yang Disusun:  Pada kode yang disusun, beberapa bulan dengan jumlah hari yang sama digabungkan ke dalam satu case, sehingga lebih efisien dan penulisan lebih ringkas dan tidak mengulangi informasi yang sama.  **Latihan 2.3** Pengubahan Contoh 3 dari switch menjadi if:    *Gambar 2.6 codingan if*  Perbandingan antara IF dan SWITCH:  switch lebih cocok digunakan untuk memeriksa nilai spesifik seperti karakter atau angka, sementara if lebih fleksibel untuk digunakan dalam berbagai jenis kondisi, termasuk rentang nilai atau kondisi yang lebih kompleks.  switch biasanya lebih efisien dan mudah dibaca jika ada banyak nilai yang diperiksa satu per satu, sedangkan if lebih cocok untuk kondisi kompleks.  **2.4**. Desain gambar flowchart dari Latihan 2.1. dan Latihan 2.3!  Flowchart 2.1  Mulai  *true*  Anda sudah rajin belajar  CaseA  Input nilai A atau B  *false*  Anda perlu kurangi main game  Case B  Pilihan anda diluar A atau B  Default  Selesai  Flowchart 2.3  Mulai  Anda sudah rajin belajar  DataA  Input nilai A atau B  Anda perlu kurangi main game  Data B  Pilihan anda diluar A atau B  Selesai | | |
| **KESIMPULAN** | | |
| Kesimpulan pada materi IF, SWITCH yaitu menunjukkan bahwa masing-masing memiliki peran penting dalam pengambilan keputusan. Pernyataan if lebih fleksibel untuk kondisi kompleks, sedangkan switch lebih efisien untuk situasi dengan banyak pilihan berdasarkan nilai yang sama. Flowchart, di sisi lain, merupakan alat visual yang membantu merencanakan dan memahami alur logika dari program secara keseluruhan, memudahkan analisis dan komunikasi ide. Kombinasi ketiganya memungkinkan pengembang untuk menulis kode yang lebih terstruktur dan mudah dipahami. | | |
|  | | |