**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 2**

**REVIEW STRUKTUR KONTROL**



Oleh:

NAMA : Alberta Aurora Adiswari

NIM : 2311102220

KELAS : S1-IF-11-02

**S1 TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

1. **DASAR TEORI**

Dalam kerangka program yang ditulis dalam bahasa pemrograman Go, program utama selalu mempunyai dua komponen berikut:

* **package main** merupakan penanda bahwa file ini berisi program utama.
* **func main()** berisi kode utama dari sebuah program Go.

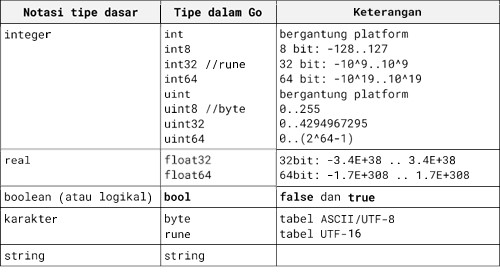
Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam Bahasa Go:

* Panggil shell atau terminal (program/utility cmd.exe di Windows)
* Masuk ke dalam (cd) folder program (normalnya ada di C:\Users\go\src\ atau yang sejenis)
* Kemudian panggil perintah go build atau go build file.go untuk mengkompilasi file.go
* Jika gagal, akan muncul pesan eror yang sesuai, pelajari dengan baik pesan tersebut, perbaiki teks program sumber, kemudian ulangi proses buildnya.
* Jika berhasil maka pada folder tersebut akan dibuat program dengan nama yang sama dan diakhiri dengan .exe (untuk Windows)
* Panggil program eksekutabel tersebut dari terminal yang sama. Jangan memanggil program tersebut denan mengklik eksekutabel tersebut dari folder karena program kalian hanya berbasis teks, bukan/belum dirancang dengan tampilan Windows.

Semua proses terkait bahasa Go dilakukan melalui utilitas go. Beberapa opsi dengan utilitas go:

* **go build**: mengkompilasi program sumber yang ada dalam folder menjadi sebuah program.
* **go build file.go**: mengkompilasi program sumber file.go saja.
* **go fmt**: membaca semua program sumber dalam folder dan mereformat penulisannya agarsesuai dengan standar penulisan program sumber Go.
* **go clean**: membersihkan file-file dalam folder sehingga tersisa program sumber nya saja.

Variabel adalah nama dari suatu Lokasi di memori, yang data dengan tipe tertentu dapat disimpan.



Operasi yang dapat dilakukan terhadap tipe data di atas adalah

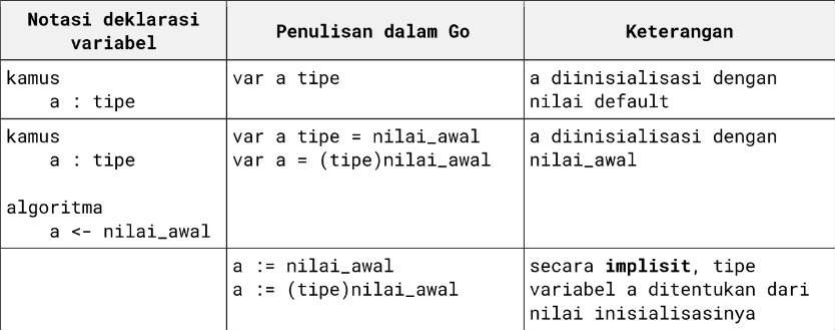


Bahasa Go menggunakan kesesuaian tipe data yang ketat. Tipe data yang berbeda tidak dapat dicampur dalam satu ekspresi, termasuk tipe yang sejenis semial sama – sama integer. Untuk menyesuaikan tipe data, ada beberapa cara yang dapat dilakukan:

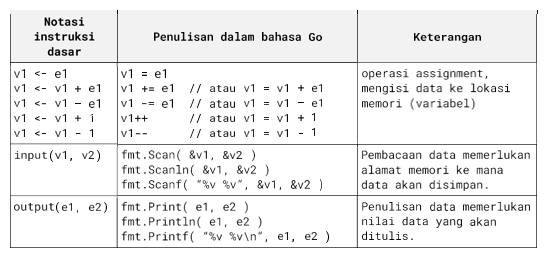
* Casting, **tipe(data)**, mengubah tipe dari data yang diberikan menjadi tipe data yang diinginkan.
* Memanfaatkan fungsi **Sprint** dan **Sscan** pada paket **fmt**.
* Memanfaatkan fungsi pada paket **strconv**, seperti **Atol**, **Itoa** dan **ParseBool**.



Sebelum digunakan variable harus dideklarasikan dan diinisialisasi terlebih dahulu agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Dalam Bahasa Go, variable yang tidak diinisialisasi lebih dahulu otomatis diisi dengan nilai default yang ekuivalen dengan bit 0.

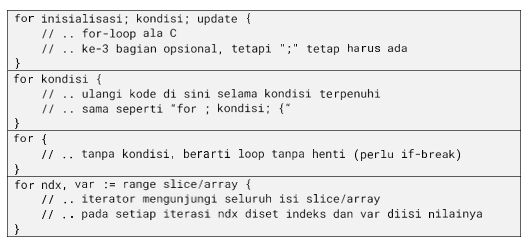


Instruksi Dasar

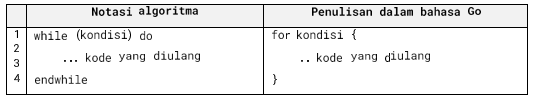


Go hanya mempunyai kata kunci for untuk semua jenis perulangan yang kita pelajari dalam notasi algoritma. Dua bentuk yang kita gunakan di sini adalah struktur while-loop dan repat-until.

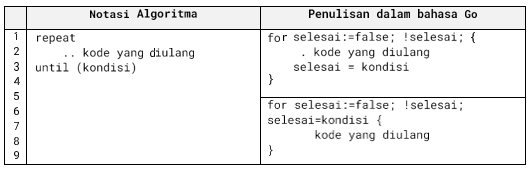
Bentuk perulangan dalam Bahasa Go



Bentuk while-loop memastikan setiap kali memasuki loop, ada kondisi yang harus terpenuhi, yang berarti saat keluar dari loop nilai kondisi tersebut pasti salah.



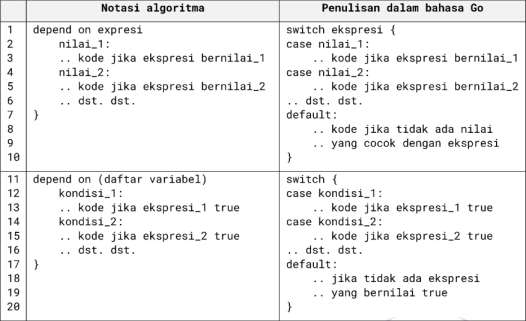
Bentuk repeat-until di perulangan dilakukan terus menerus sampai kondisi keluar terpenuhi. Artinya selama kondisi belum terpenuhi maka perulangan akan terus dilakukan. Pada saat keluar dari loop maka nilai kondisi pasti benar.



Bentuk if-else



Bentuk Switch-case

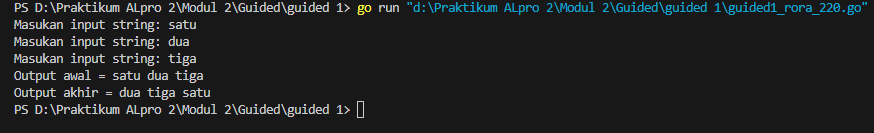


1. **GUIDED**
   * + 1. **Guided 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var (          satu, dua, tiga string          temp            string      )      fmt.Print("Masukan input string: ")      fmt.Scanln(&satu)      fmt.Print("Masukan input string: ")      fmt.Scanln(&dua)      fmt.Print("Masukan input string: ")      fmt.Scanln(&tiga)      fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)      temp = satu      satu = dua      dua = tiga      tiga = temp      fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)  } |

**Screenshoot program**

****

**Deskripsi program**

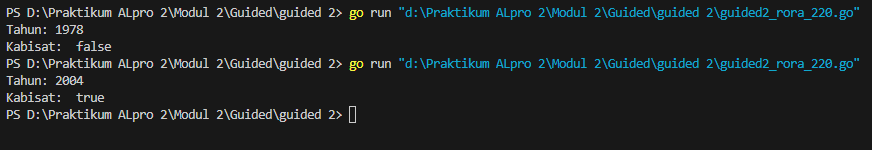
Pada program ini menggunakan Bahasa pemrograman Go yang dimana program meminta pengguna untuk menginputkan 3 inputan string yang disimpan ke dalam variable satu, dua, dan tiga. Lalu menghasilkan outputan awal dengan mengurutkan ketiga inputan string sesuai dengan urutan inputannya.

* + - 1. **Guided 2**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      year := 0      fmt.Print("Tahun: ")      fmt.Scan(&year)      fmt.Println("Kabisat: ", year%4 == 0 && year%1000 != 0)  } |

**Screenshoot program**

****

**Deskripsi program**

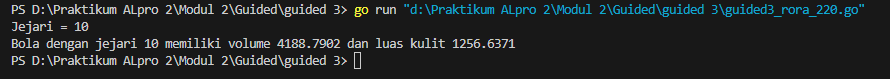
Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini digunakan untuk menentukan benar tidaknya perhitungan tahun kabisat. Pada program pengguna diminta untuk menginputkan tahun yang nantinya program akan mengecek dengan menggunakan rumus yang telah dirancang untuk menentukan apakah tahun yang diinputkan habis dibagi 4, apabila kondisi ini terpenuhi, maka tahun yang diinputkan adalah tahun kabisat.

* + - 1. **Guided 3**

**Source code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var jari float64      fmt.Print("Jejari = ")      fmt.Scanln(&jari)      volume := (4.0 / 3.0) \* math.Pi \* math.Pow(jari, 3)      luas := 4 \* math.Pi \* math.Pow(jari, 2)      fmt.Printf("Bola dengan jejari %v memiliki volume %.4f dan luas kulit %.4f", jari, volume, luas)  } |

**Screenshoot program**

****

**Deskripsi program**

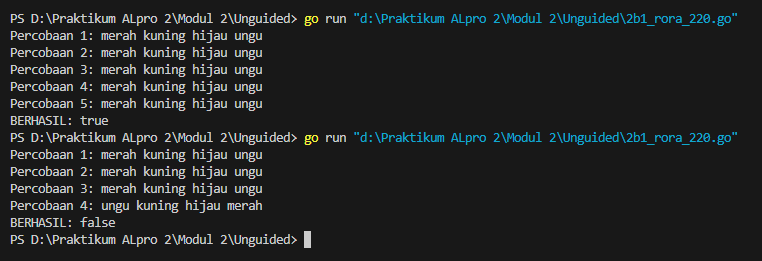
Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini menghitung volume dan luas permukaan bola berdasarkan jari-jari yang diinputkan oleh pengguna. Pengguna akan diminta untuk menginputkan jari-jari yang lalu program akan menghitung volume bola dengan rumus 4/3 \* π \* r^3 dan luas permukaan bola menggunakan rumus 4 \* π \* r^2.

1. **UNGUIDED**
   * + 1. **Unguided 2b1**

**Source code**

|  |
| --- |
| package main  import (      "bufio"      "fmt"      "os"      "strings"  )  func main() {      cairanWarna\_220 := []string{"merah", "kuning", "hijau", "ungu"}      scanner := bufio.NewScanner(os.Stdin)      success := true      for i := 1; i <= 5; i++ {          fmt.Printf("Percobaan %d: ", i)          if !scanner.Scan() {              fmt.Println("Input error")              return          }          warna := strings.Fields(scanner.Text())          if len(warna) != 4 {              success = false              break          }          for j, warna := range warna {              if warna != cairanWarna\_220[j] {                  success = false                  break              }          }          if !success {              break          }      }      fmt.Printf("BERHASIL: %v\n", success)  } |

**Screenshoot program**

****

**Deskripsi program**

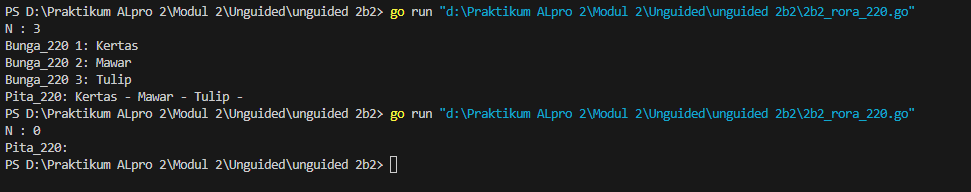
Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini digunakan untuk memeriksa apakah input dari pengguna sesuai dengan urutan warna yang diharapkan dalam lima percobaan. Pada program dilakukan lima kali percobaan sehingga pengguna diminta untuk menginputkan empat warna dalam satu baris, setiap inputan warna dipisahkan menggunakan spasi dan dibandingkan dengan warna target. Apabila urutan yang diinputkan tidak sesuai atau kurang dari 4 warna, maka program akan menampilkan BERHASIL: false. Namun jika urutan yang diinputkan sesuai maka program akan menampilkan BERHASIL: true sebagai outputan.

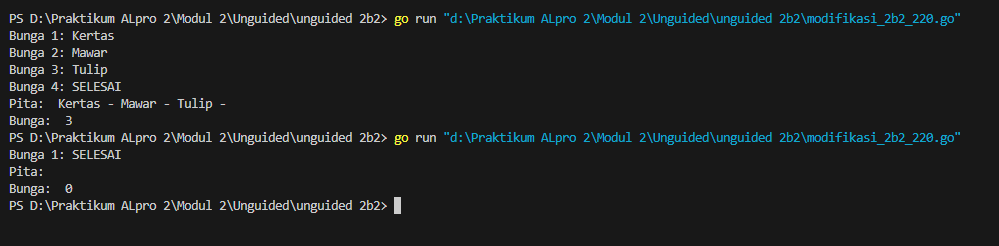
* + - 1. **Unguided 2b2**

**Source code**

|  |
| --- |
| package main  import (      "bufio"      "fmt"      "os"      "strings"  )  func main() {      pita\_220 := ""      var bunga\_220List []string      scanner := bufio.NewScanner(os.Stdin)      fmt.Print("N: ")      scanner.Scan()      input := scanner.Text()      if input == "0" {          fmt.Println("Pita:")          fmt.Println("Bunga: 0")          return      }      count := 0      for {          count++          fmt.Printf("Bunga %d: ", count)          scanner.Scan()          bunga\_220 := scanner.Text()          if strings.ToUpper(bunga\_220) == "SELESAI" {              count--              break          }          if pita\_220 == "" {              pita\_220 = bunga\_220          } else {              pita\_220 += "-" + bunga\_220          }          bunga\_220List = append(bunga\_220List, bunga\_220)      }      fmt.Println("Pita:", pita\_220)      fmt.Printf("Bunga: %d\n", len(bunga\_220List))  } |
| package main  import (      "fmt"      "strings"  )  func main() {      var n int      var bunga\_220, pita\_220 string      for i := 0; ; i++ {          n = i          fmt.Print("Bunga ", i+1, ": ")          fmt.Scan(&bunga\_220)          if strings.ToLower(bunga\_220) == "selesai" {              break          }          pita\_220 += bunga\_220 + " - "      }      fmt.Println("Pita: ", pita\_220)      fmt.Println("Bunga: ", n)  } |

**Screenshoot program**

****

****

**Deskripsi progam**

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program ini meminta pengguna untuk menginputkan nama bunga, yang kemudian nantinya program akan menambahkan setiap nama bunga ke dalam string pita, dimana setiap nama bunga akan dipisahkan dengan tanda “-“. Program pertama-tama akan meminta pengguna untuk menginputkan integer n untuk menyimpan jumlah bunga yang akan kita masukkan. Dalam loop sebanyak n kali, pengguna diminta untuk menginputkan nama bunga, lalu setiap nama bunga ditambahkan ke dalam variable pita dan diikuti dengan tanda ”-“. Setelah semua bunga dimasukkan, program akan menampilkan hasil akhir string pita dengan format “bunga 1 – bunga 2 - …” sebagai outputan.

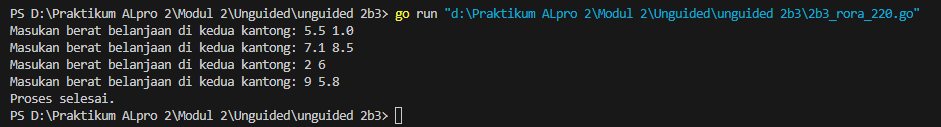
Pada modifikasi program pada loop tidak terhingga sehingga program meminta pengguna untuk memasukkan nama bunga dan setiap nama bunga akan dimasukkan ke dalam string pita, dan dipisahkan dengan “-“. Ketika pengguna mengetik selesai program keluar dari loop sehingga program mencetak hasil akhir dari string pita yang berisi nama-nama bunga yang diinputkan serta total jumlah yang dicatat pada variable n pada outputan.

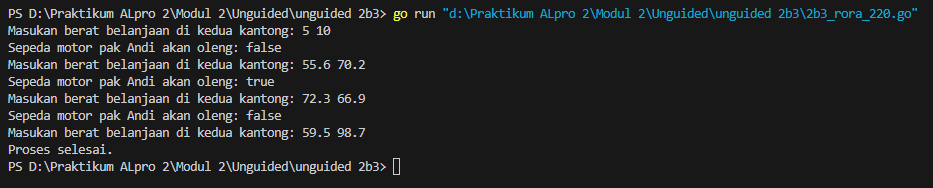
* + - 1. **Unguided 2b3**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var kiri\_220, kanan\_220 float32      for kiri\_220 < 9 && kanan\_220 < 9 {          fmt.Print("Masukan berat belanjaan di kedua kantong: ")          fmt.Scan(&kiri\_220, &kanan\_220)      }      fmt.Println("Proses selesai.")  } |
| package main  import (      "fmt"      "math"  )  func main() {      var kiri\_220, kanan\_220 float32      for {          fmt.Print("Masukan berat belanjaan di kedua kantong: ")          fmt.Scan(&kiri\_220, &kanan\_220)            if kiri\_220 < 0 || kanan\_220 < 0 || (kiri\_220+kanan\_220) > 150 {              fmt.Println("Proses selesai.")              break          }            selisih := math.Abs(float64(kiri\_220 - kanan\_220))          oleng := selisih >= 9          fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)      }  } |

**Screenshoot program**

****

****

**Deskripsi program**

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menerima inputan berjalan dari dua kantong, kiri dan kanan dan untuk menentukan apakah sepeda motor pak Andi akan oleng berdasarkan perbandingan berat kedua kantong tersebut. Program meminta pengguna untuk menginputkan berat belanjaan di kedua kantong secara berulang selama total berat belanjaan pada kantong kurang dari 150.

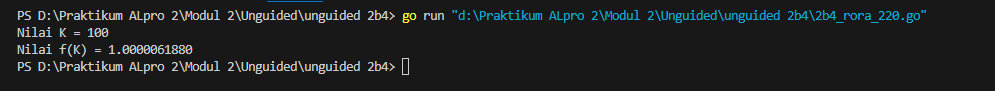
Pada modifikasi program pada loop, setelah diinputkan berat belanjaan pada kedua kantong program mencetak pernyataan apakah sepeda motor pak andi akan oleng atau tidak, dengan menggunakan kondisi kiri > kanan + 9 dan kanan > kiri + 9. Apabila satu kantong beratnya lebih dari 9kg dan melebihi berat kantong yang lain maka sepeda motor pak andi akan oleng. Setelah total berat melebihi 150, program akan keluar dari loop dan mencetak “Proses selesai” sebagai outputan.

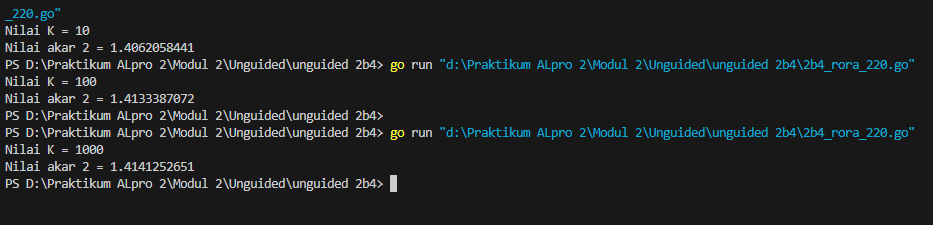
* + - 1. **Unguided 2b4**

**Source code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var k float64      fmt.Print("Nilai K = ")      fmt.Scan(&k)      numerator\_220 := math.Pow((4\*k + 2), 2)      denominator\_220 := (4\*k + 1) \* (4\*k + 3)      fk := numerator\_220 / denominator\_220      fmt.Printf("Nilai f(K) = %.10f\n", fk)  } |
| package main  import (      "fmt"      "math"  )  func main() {      var K int      fmt.Print("Nilai K = ")      fmt.Scan(&K)      result := 1.0      for k := 0; k <= K; k++ {          numerator := math.Pow((4\*float64(k) + 2), 2)          denominator := (4\*float64(k) + 1) \* (4\*float64(k) + 3)          result \*= numerator / denominator      }      fmt.Printf("Nilai akar 2 = %.10f\n", result)  } |

**Screenshoot program**

****

****

**Deskripsi program**

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menghitung nilai fungsi matematika berdasarkan input nilai k yang diinputkan oleh pengguna. Program meminta pengguna untuk menginputkan nilai k sehingga program menggunakan rumus pada soal untuk menghitung hasilnya. Hasil akhir dicetak dengan format yang menunjukkan 10 angka decimal setelah titik decimal.

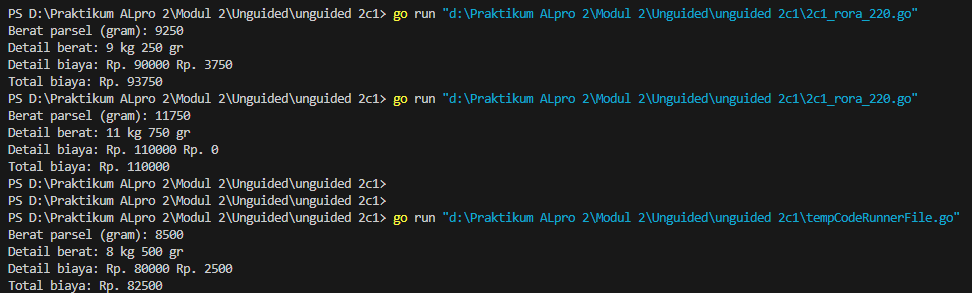
Pada modifikasi program setelah pengguna menginputkan nilai k, program melakukan iterasi dari 0 hingga k-1 menggunakan loop for. Dimana pada setiap iterasinya program menghitung nilai berdasarkan rumus deret akar 2 dan mengalikan hasilnya dengan variable hasil yang diinisialisasi dengan 1. Setelah semua iterasi diselesaikan program mencetak hasil akhir dengan format yang sama.

* + - 1. **Unguided 2c1**

**Source code**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var beratParcel int      // Meminta input berat parsel dalam gram      fmt.Print("Berat parsel (gram): ")      fmt.Scan(&beratParcel)      // Menghitung total berat dalam kg dan sisa dalam gram      totalKg\_220 := beratParcel / 1000      sisaGr\_220 := beratParcel % 1000      // Menghitung biaya dasar      biayaDasar := totalKg\_220 \* 10000      var biayaSisa int      // Menghitung biaya tambahan berdasarkan sisa berat      if totalKg\_220 > 10 {          biayaSisa = 0 // Sisa berat gratis jika total lebih dari 10kg      } else {          if sisaGr\_220 >= 500 {              biayaSisa = sisaGr\_220 \* 5 // Rp. 5,- per gram          } else {              biayaSisa = sisaGr\_220 \* 15 // Rp. 15,- per gram          }      }      // Menampilkan detail berat dan biaya      fmt.Printf("Detail berat: %d kg %d gr\n", totalKg\_220, sisaGr\_220)      fmt.Printf("Detail biaya: Rp. %d Rp. %d\n", biayaDasar, biayaSisa)      fmt.Printf("Total biaya: Rp. %d\n", biayaDasar+biayaSisa)  } |

**Screenshoot program**

****

**Deskripsi program**

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menghitung total biaya pengiriman berdasarkan berat parsel yang dimasukkan oleh pengguna dalam gram. Biaya pengiriman dasar dihitung sebagai original, yang merupakan biaya untuk setiap kilogram, ditetapkan pada Rp. 10.000 per kilogram. Jika berat kilogram kurang dari 10, program kemudian menentukan biaya tambahan berdasarkan berat gram: jika kurang dari 500 gram, biaya tambahan dihitung dengan mengalikan gram dengan 15; jika 500 gram atau lebih, biaya tambahan dihitung dengan mengalikan gram dengan 5. Setelah semua biaya dihitung, program mencetak detail berat, biaya dasar, biaya tambahan, dan total biaya pengiriman.

* + - 1. **Unguided 2c2**
         1. **Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?**

Keluaran dari program tersebut jika nam yang diberikan adalah 80.1 adalah nilai “A”. Eksekusi program ini benar apabila hanya meninjau kondisi bahwa nam lebih besar dari 80. Akan pada program ini tidak sesuai dengan spesifikasi soal karena pada table, nilai yang diberikan untuk 80 < nam ≤ 100 adalah “A”, sedangkan kondisi 72.5 < nam ≤ 80 adalah “AB”.

* 1. **Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!**

Kesalahan dari program ini pada penggunaan operator perbandingan, terutama ketika menangani batasan rentang nilai, yang mana pada program, kondisi rentang nilai terdapat yang salah.

Sebagai contoh, yang mana pada tabel seharusnya:

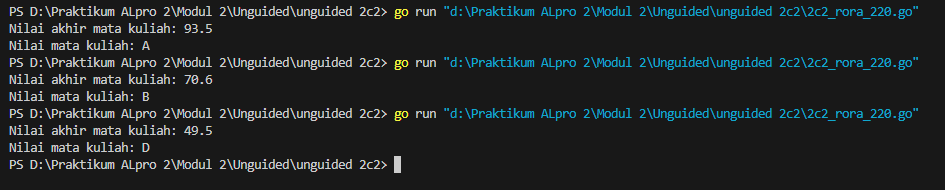
72.5 < NAM ≤ 80 yaitu senilai dengan “AB” akan tetapi dalam program tidak ada kondisi untuk batas atas 80, sehingga semua nilai lebih besar dari 80 akan menghasilkan A. selain itu seharusnya program memeriksa kondisi rentang nilai dengan kondisi untuk batasnya.

* 1. Perbaiki program tersebut! Ujilah dengan masukan 93.5; 70.6; dan 49,5. Seharusnya keluaran yang diperoleh adalah ‘A’,’B’ dan ‘D”

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var nam float64      var nmk string      fmt.Print("Nilai akhir mata kuliah: ")      fmt.Scanln(&nam)      if nam > 80 {          nmk = "A"      } else if nam > 72.5 {          nmk = "AB"      } else if nam > 65 {          nmk = "B"      } else if nam > 57.5 {          nmk = "BC"      } else if nam > 50 {          nmk = "C"      } else if nam > 40 {          nmk = "D"      } else {          nmk = "E"      }      fmt.Println("Nilai mata kuliah:", nmk)  } |

**Screenshoot program**

****

**Deskripsi program**

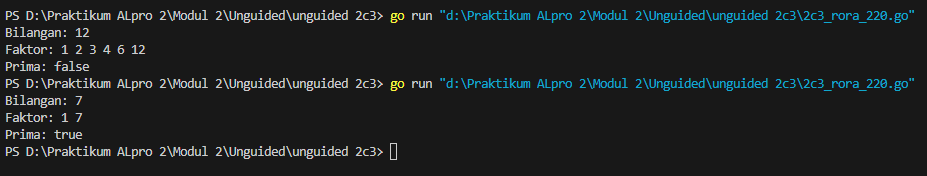
Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menentukan nilai huruf dari nilai akhir mata kuliah berdasarkan input nilai yang diinputkan oleh pengguna. Program meminta pengguna menginputkan nilai akhir dalam bentuk angka decimal, program kemudian menggunakan serangkaian pernyataan kondisional if dan else if untuk menentukan kategori nilai huruf yang sesuai.

* + - 1. **Unguided 2c3**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var bil\_220 int      var prima bool = true      fmt.Print("Bilangan: ")      fmt.Scan(&bil\_220)      fmt.Print("Faktor: ")      for i := 1; i <= bil\_220; i++ {          if bil\_220%i == 0 {              fmt.Print(i, " ")              if !(i == 1 || i == bil\_220) {                  prima = false              }          }      }      fmt.Println()      fmt.Print("Prima: ", prima)  } |

**Screenshoot program**

****

**Deskripsi program**

Pada program ini menggunakan Bahasa pemrograman Go yang dimana program menentukan faktor-faktor dari suatu bilangan dan memeriksa apakah bilangan tersebut merupakan bilangan prima. Program meminta pengguna untuk menginputkan sebuah bilangan, sehingga program akan mencetak semua factor dari bilangan dengan menggunakan loop. Di dalam loop, program memeriksa apakah factor tersebut bukan 1 atau bilangan itu sendiri. Jika ada factor lain, maka variable prima diubah menjadi false. Setelah loop selesai program akan mencetak semua factor dan menunjukkan apakah bilangan tersebut adalah bilangan prima atau tidak.