**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2**

**REVIEW STRUKTUR KONTROL**

**Sebuah gambar berisi logo, simbol, Grafis, Font

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**RAIA GUSTI PRAYOGO**

**2311102238**

**IF-11-06**

**Dosen Pengampu :**

**ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY 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.

Komentar, bukan bagian dari kode program, dan dapat ditulis di mana saja di dalam program:

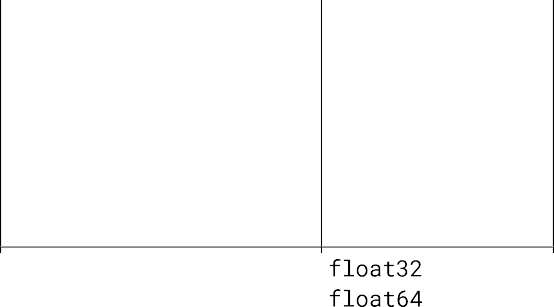
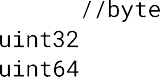
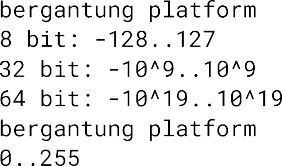
* Satu baris teks yang diawali dengan garis miring ganda ('Il') s.d. akhir baris, atau.
* Beberapa baris teks yang dimulai dengan pasangan karakter 'l\*' dan diakhiri dengan '\*l'.
* // Setiap program utama dimulai dengan "package main"



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

' Nama variabel dimulai dengan huruf dan dapat dikuti dengan sejumlah huruf, angka, atau garisbawah.

Contoh: ketemu, found, rerata, mhsl, data 2



Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string. Lihat tabel berikut ini untuk variasi tipe data yang disediakan dalam bahasa Go.

* + Nilal data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya.

Contoh: Menyebutkan nama found akan mengambil nilai tersimpan dalam memori untuk variabel found, pastinya.

* + Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks & di depan nama variabel tersebut.

Contoh: &found akan mendapatkan alamat memori untuk menyimpan data pada found.

* + Jika variabel berisi alamat memori, prefiks \* pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.

**Struktur Kontrol Perulangan**

Co 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 repeat-until

|  |
| --- |
|  |
|  |
|  |
|  |

1. Bentuk While-Loop

Bentuk while-loop memastikan setiap kali memasuki loop, ada kondisi yang harus terpenuhi (benarltrue). Inijuga berarti saat keluar dari loop, maka nilai kondisi tersebut pasti salahlfalse!

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. Bentuk Repeat-Until

Bentuk repeat-until di perulangan dilakukan terus menerus sampai kondisi keluar terpenuhi. Artinya selama kondisi belum terpenuhi (salahlfalse) maka perulangan akan terus dilakukan.

Pada saat keluar dari loop maka nilai kondisi pasti benar/true!

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |

**Struktur Kontrol Percabangan**

Untuk analisa kasus, bahasa Go mendukung dua bentuk percabangan, yaitu if-else dan switch-case.

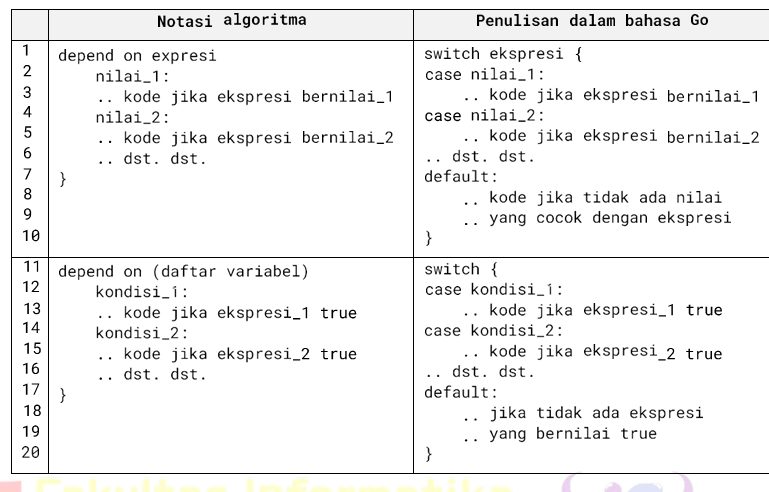
1. Bentuk If-Else

Berikut ini bentuk-bentuk if-else yang mungkin dilakukan dalam bahasa Go. Semua bentuk di bawah merupakan satu instruksi if-else-endif saja (hanya satu endif). Bentuk if-else yang bersarang (dengan beberapa endif) dapat dibentuk dengan komposisi beberapa if-else-endif tersebut.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Bentuk Switch-Case

Dalam bahasa Co ada dua variasi bentuk switch-case. Bentuk yang biasa digunakan adalah ekspresi ditulis pada perintah switch dan nilai ditulis dalam setiap label case-nya. Bentuk yang kedua mempunyai switch tanpa ekspresi, tetapi setiap case boleh berisi ekspresi boolean.

Tentunya bentuk yang kedua lebih bersifat umum, dan merupakan penyederhanaan bentuk

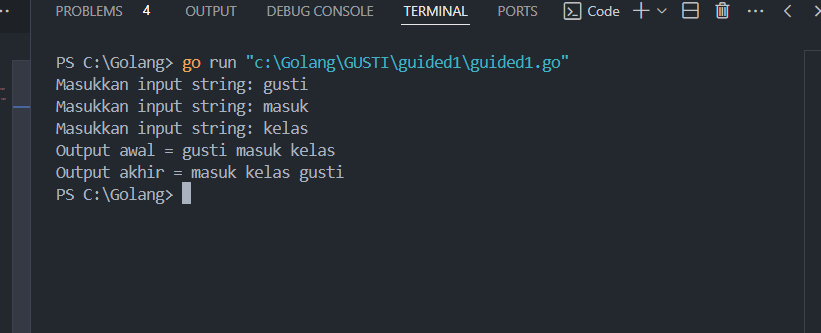
1. **GUIDED**
   * + 1. **Soal Studi Case**

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data vang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {  var satu, dua, tiga string  var temp string  fmt.Print("Masukkan input string: ")  fmt.Scanln(&satu)  fmt.Print("Masukkan input string: ")  fmt.Scanln(&dua)  fmt.Print("Masukkan 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 Output**

****

**Deskripsi Program**

Program ini merupkan program untuk menukar nilai inputan. Terdapat variabel satu, dua ,tiga dan temp dengan tipe data string. Kemudian program akan menerima input. Output akan ditampilkan sesuai urutan inputan. Setelah itu program melakukan pertukaran nilai variabel. Nilai satu menjadi dua, dua menjadi tiga dan tiga menjadi satu, kemudian ditampilkan di output

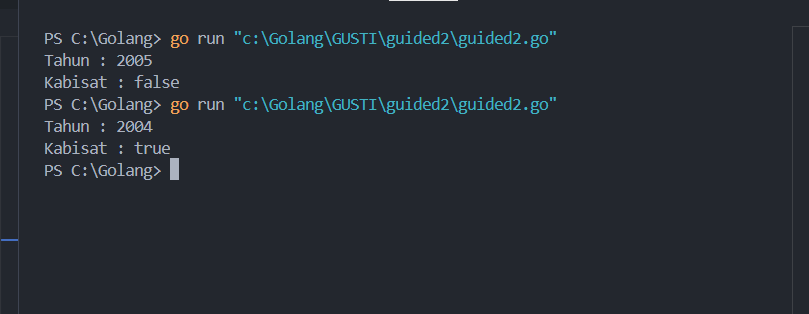
* + - 1. **Soal Studi Case**

Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false).

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {  var tahun int  fmt.Print("Tahun : ")  fmt.Scan(&tahun)  fmt.Printf("Kabisat : %t \n", (tahun%4 == 0 && (tahun%100 != 0 || tahun%400 == 0)))  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program untuk memnentukan apakah tahun tersebut adalah tahun kabisat. Dengan variabel tahun tipe data int. Kemudian program menerima inputan dari user. Dan tahun akan dicek ketika tahun habis dibagi 400 dan tidak habis dibagi 100 maka output akan true yang berarti tahun tersebut adalah kabisat

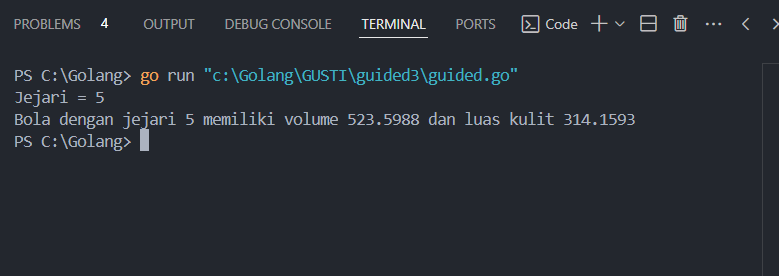
* + - 1. **Soal Studi Case**

Buat program Bola yang menerima inputjari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola. volumebola = - 4/3r 3 dan luasbola = 4nr 2 (Tr3.1415926535).

**Sourcecode**

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

**Screenshoot Output**

****

**Deskripsi Program**

Program ini untuk mencari luas dan volume bola dengan inputan jari jari dari user. Terdapat variabel vlomue dan luas dengan tipe data float Kemudian akan diproses dengan rumus luas dan volume. Dan program akan menampilkan output berupa volume dan luas kulit dalam nilai desimal

1. **UNGUIDED**
   * + 1. **Soal Studi Case**

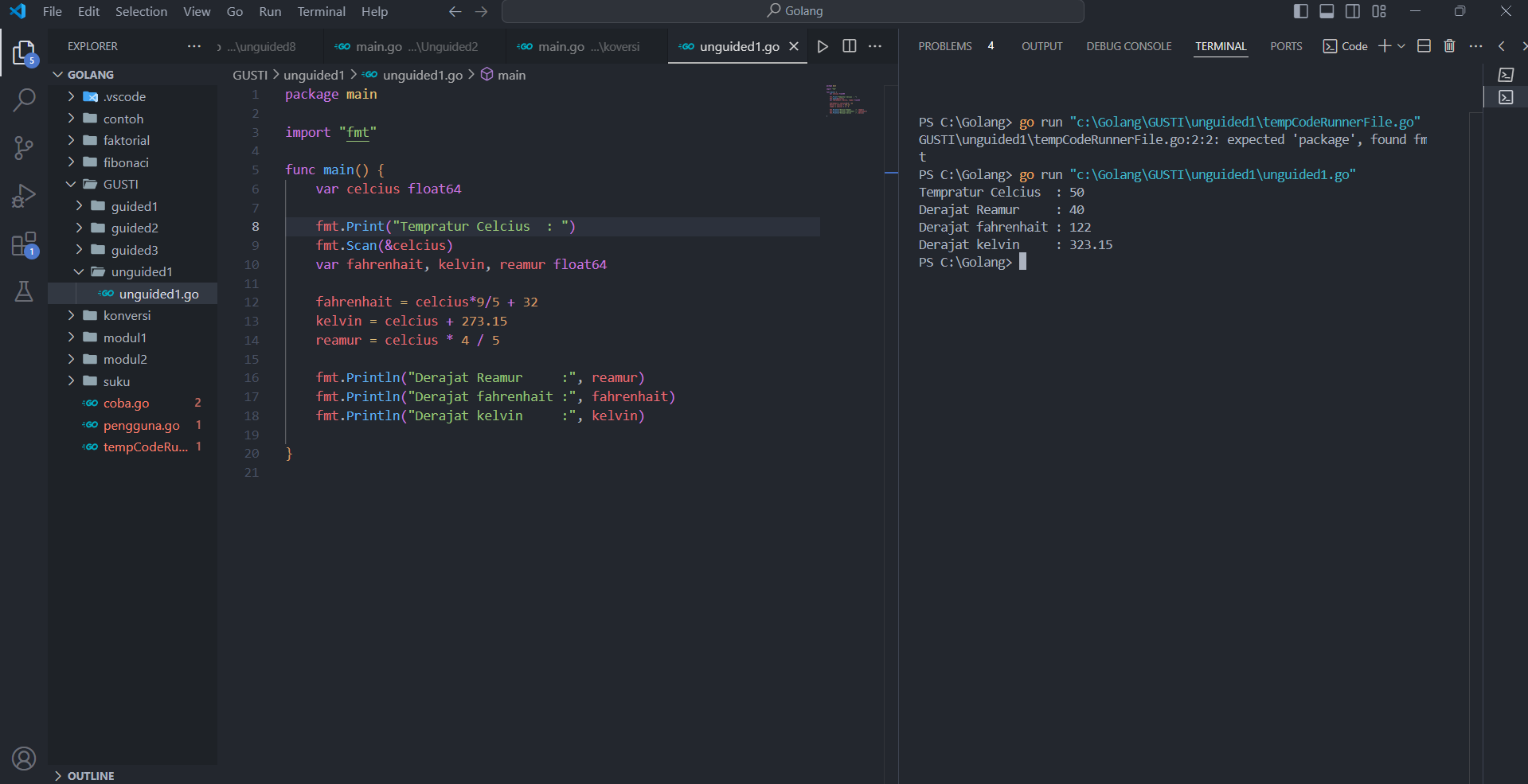
Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam



**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var celcius float64      fmt.Print("Tempratur Celcius  : ")      fmt.Scan(&celcius)      var fahrenhait, kelvin, reamur float64      fahrenhait = celcius\*9/5 + 32      kelvin = celcius + 273.15      reamur = celcius \* 4 / 5      fmt.Println("Derajat Reamur     :", reamur)      fmt.Println("Derajat fahrenhait :", fahrenhait)      fmt.Println("Derajat kelvin     :", kelvin)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini adalah program untuk mengkonversi suhu dalam derajat celcius ke fahrenhait, kelvin dan reamur. Terdapat deklarasi variabel celcius dengan tipe data float yang kemudian akan diinput oleh user. Selajutnya terdapat rumus untuk mengkonversi kemasing masing derajat. Kemudian hasil dari konversi itu akan ditampilkan dalam output

* + - 1. **Studi Case**

Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja.

Buat program ASCII yang akan membaca 5 buat data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII)

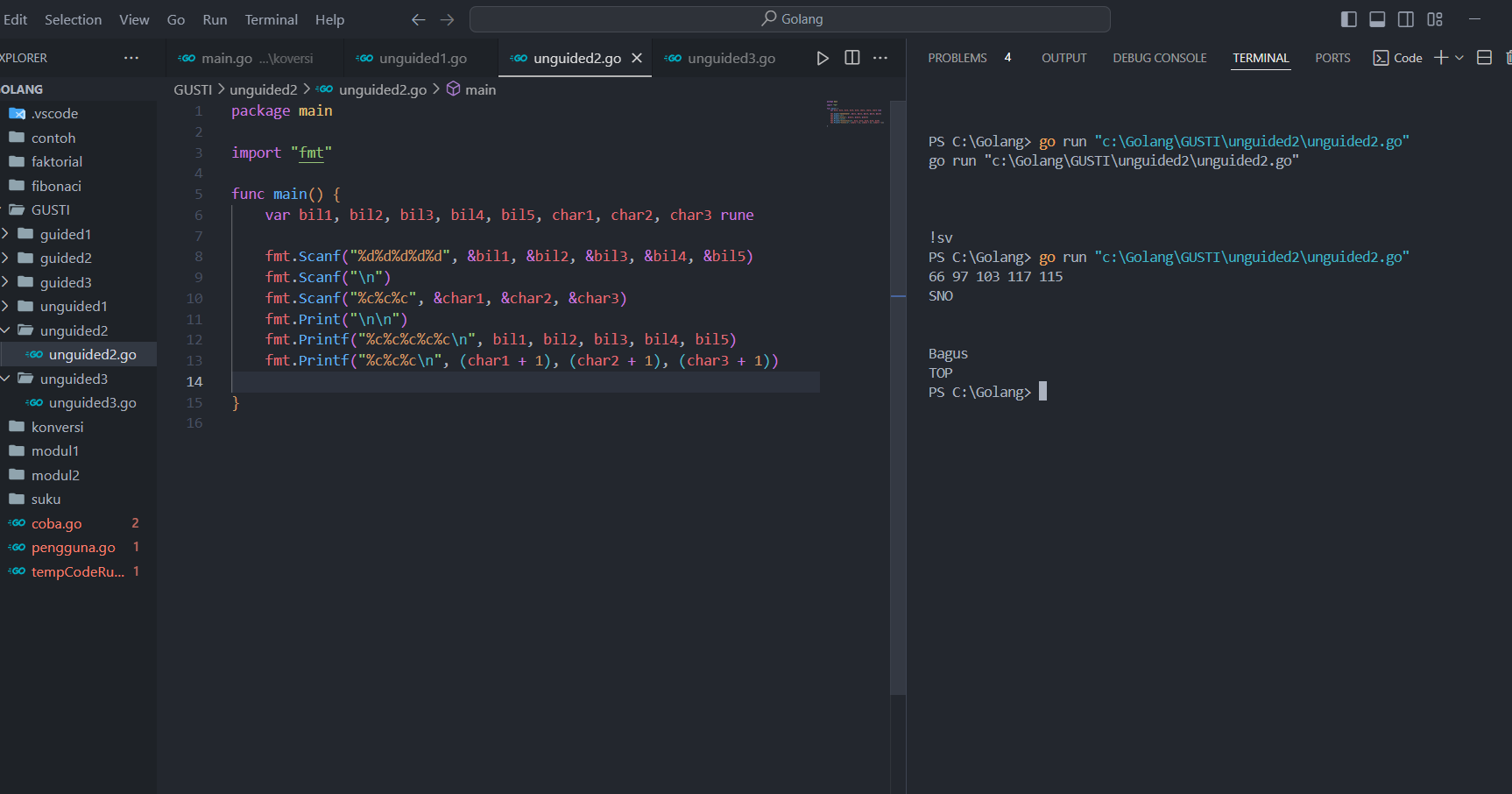
Masukan terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

Keluaranjuga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var bil1, bil2, bil3, bil4, bil5, char1, char2, char3 rune      fmt.Scanf("%d%d%d%d%d", &bil1, &bil2, &bil3, &bil4, &bil5)      fmt.Scanf("\n")      fmt.Scanf("%c%c%c", &char1, &char2, &char3)      fmt.Print("\n\n")      fmt.Printf("%c%c%c%c%c\n", bil1, bil2, bil3, bil4, bil5)      fmt.Printf("%c%c%c\n", (char1 + 1), (char2 + 1), (char3 + 1))  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program untuk program ASCII yang akan mengubah integer ke dalam karakter. Dalam program terdapat variabel bil1 sampai bil5 dan char1 sampai 3 dengan tipe data rune(merupakan tipe data unicode). Kemudian user akan menginput intger sebanayak 5 setelah itu user akan menginput lagi 3 karakter. Kemudian akan ditempilkan output yang berupa karakter sesuai dengan tabel ASCII dan tiga karakter yang merupakan karakter setelahnya

* + - 1. **Soal Studi Case**

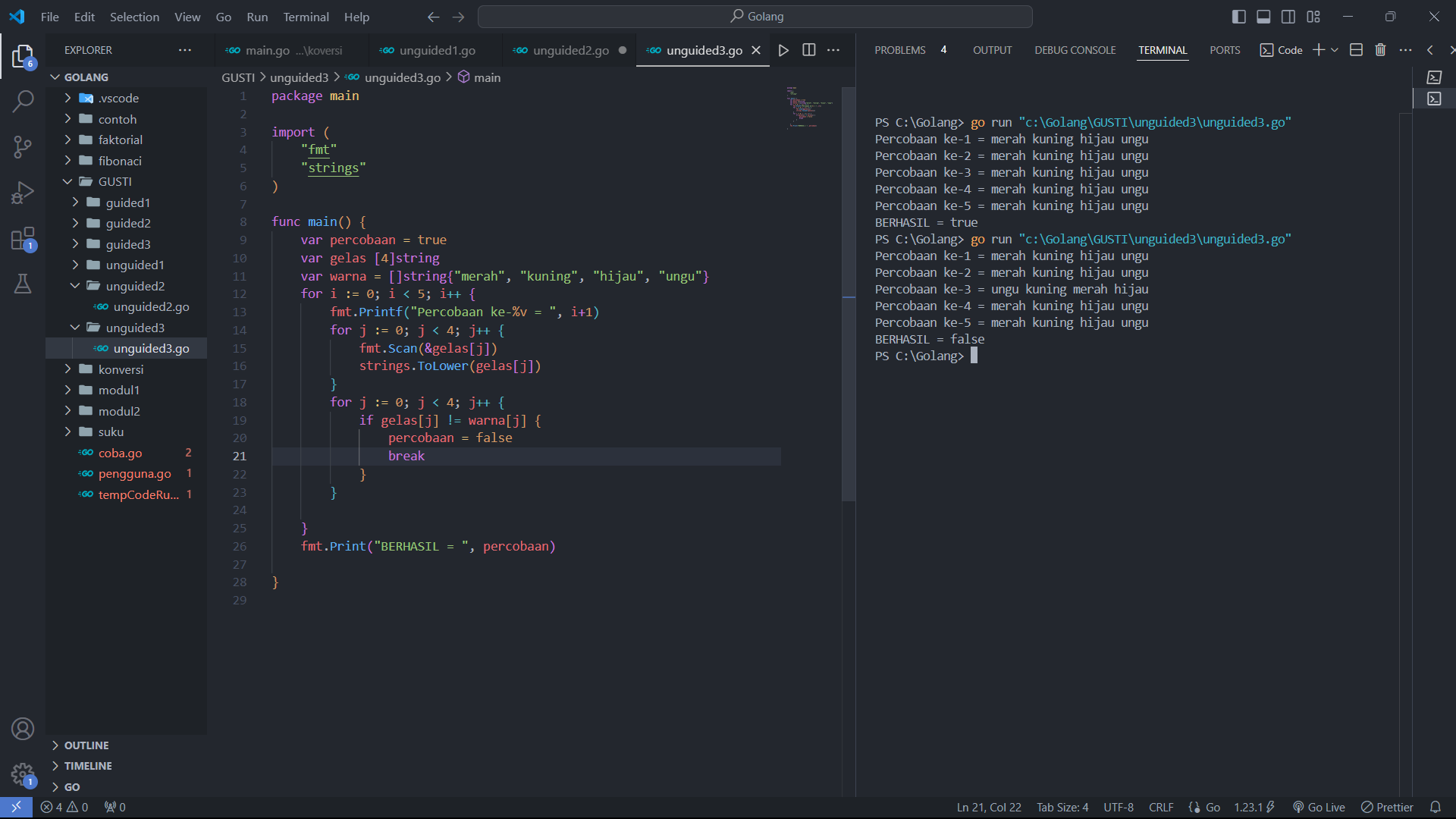
Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturutan adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "strings"  )  func main() {      var percobaan = true      var gelas [4]string      var warna = []string{"merah", "kuning", "hijau", "ungu"}      for i := 0; i < 5; i++ {          fmt.Printf("Percobaan ke-%v = ", i+1)          for j := 0; j < 4; j++ {              fmt.Scan(&gelas[j])              strings.ToLower(gelas[j])          }          for j := 0; j < 4; j++ {              if gelas[j] != warna[j] {                  percobaan = false                  break              }          }      }      fmt.Print("BERHASIL = ", percobaan)  } |

**Screenshoot Output**



**Deskripsi Program**

Program ini adalah program untuk mealukan percobaan. Terdapat deklarasi variabel gelas dan warna yang merupakn array dengan size 4 dengan tipe data string, dan variabel percobaan dengan tipe data boolean. Terjadi perulangan untuk menginput 4 warna dalam 4 gelas sebanyak 5 kali perulangan. Kemudian terdapat percabagan if untuk meriksa apakah warna yang diinput sama dengan warna yang benar. Ketika warna benar selama 5 kali percobaan maka akan mernampilkan output true. Namun ketika dalam 5 percobaan ada yang tidak benar maka output akan false

* + - 1. **Soal Studi Case**

Suatu pita (string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan

Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak not) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita.

(Petunjuk: gunakan operasi penggabungan string dengan operator "+

Tampilkan isi pita setelah proses input selesai.

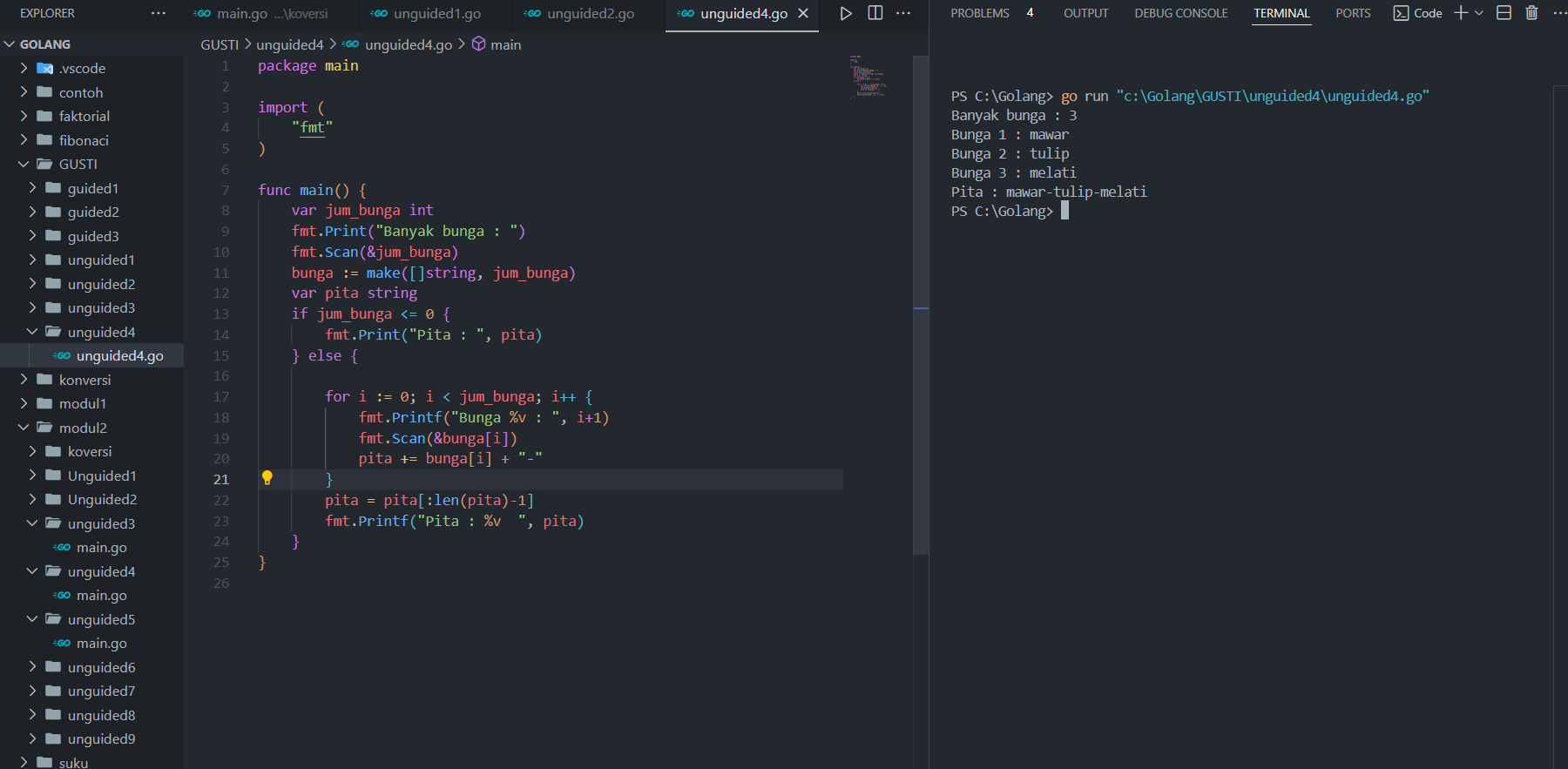
**Sourcecode**

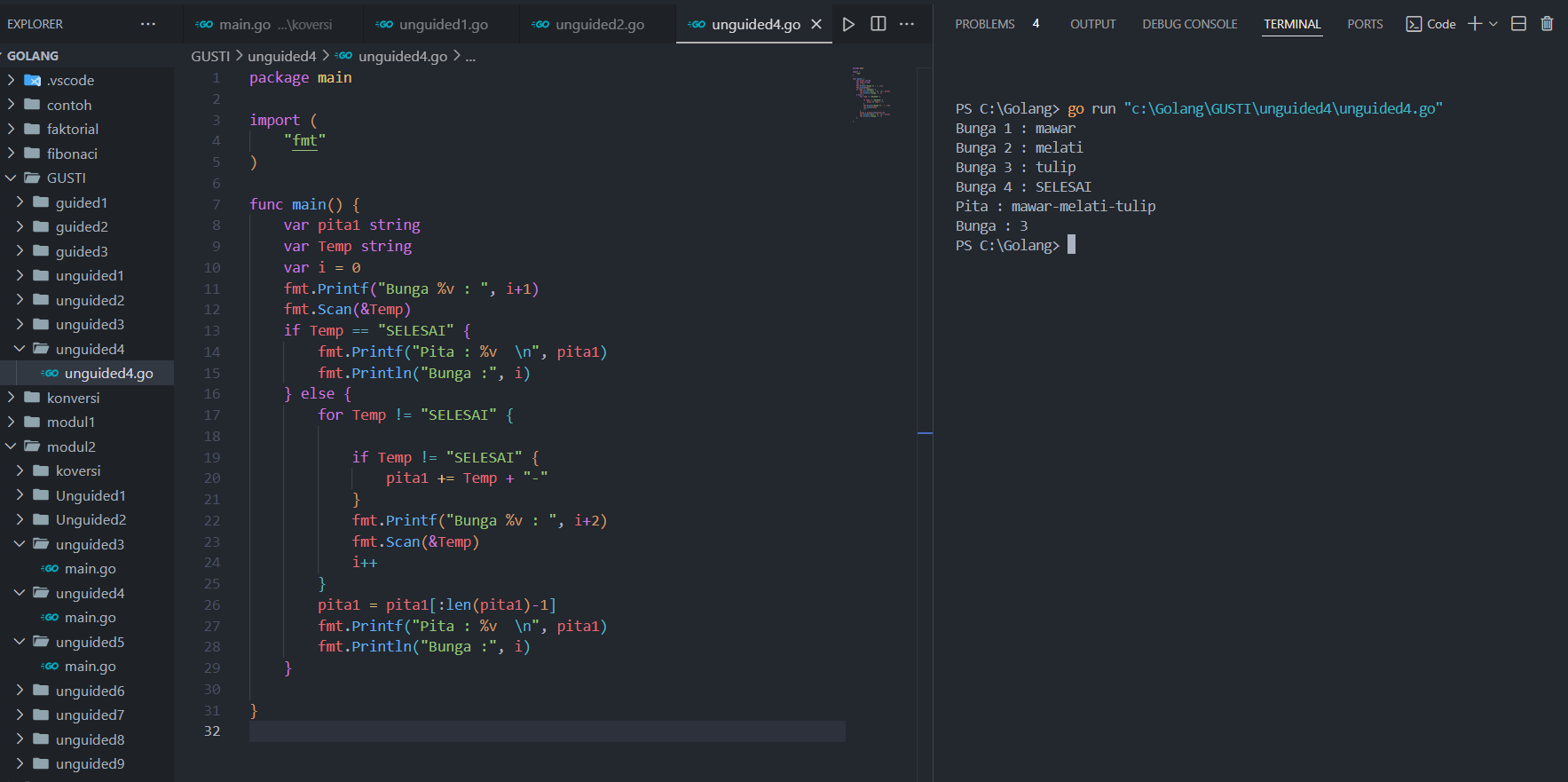
|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var jum\_bunga int      fmt.Print("Banyak bunga : ")      fmt.Scan(&jum\_bunga)      bunga := make([]string, jum\_bunga)      var pita string      if jum\_bunga <= 0 {          fmt.Print("Pita : ", pita)      } else {          for i := 0; i < jum\_bunga; i++ {              fmt.Printf("Bunga %v : ", i+1)              fmt.Scan(&bunga[i])              pita += bunga[i] + "-"          }          pita = pita[:len(pita)-1]          fmt.Printf("Pita : %v  ", pita)      }  } |

**Sourcecode (MODIFIKASI)**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var pita1 string      var Temp string      var i = 0      fmt.Printf("Bunga %v : ", i+1)      fmt.Scan(&Temp)      if Temp == "SELESAI" {          fmt.Printf("Pita : %v  \n", pita1)          fmt.Println("Bunga :", i)      } else {          for Temp != "SELESAI" {              if Temp != "SELESAI" {                  pita1 += Temp + "-"              }              fmt.Printf("Bunga %v : ", i+2)              fmt.Scan(&Temp)              i++          }          pita1 = pita1[:len(pita1)-1]          fmt.Printf("Pita : %v  \n", pita1)          fmt.Println("Bunga :", i)      }  } |

**Screenshoot Output**

****

****

**Deskripsi Program**

Program untuk mengiput nam bunga dan kemudian ditampilkan. Terdapat variabel jum\_bunga tipe data int untuk menginput jumlah bunga. Kemudian terdapat deklarasi array dengan panjang sesuai jumlah bunga yang diinput. Percabanagn if dengan jumlah bungan kurang atau sama dengan 0 maka output hanya pita kosong saja. Namunketika kondisi if tidak terpenuhi maka akan berjalan perulngan untuk menginput bunga 1 dan seterusnya. Inputan kemdudian akan digabungakan dengan tanda – dalam pita. Sebelum ditampilkan pita akan dikurangi sebanyak 1 karakter untuk menghapus tanda – yang berlebih. Kemudian output berupa nama bunga dengan penghubung ”- ”.

Selanjutnya program akan dimodifikasi untuklangsung menginput nama bunga tanpa menginput jumlah bunga terlebih dahulu. Dengan menggunkan perulangan kondisi inputan bukan sama dengan selesai maka perulangan akan berlanjut hingga user menginput ”SELESAI”. Kemudian output akan ditampilkan seperti sebelumnya dengan tambahan jumlah bunga di akhir

* + - 1. **Soal Studi Case**

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan olengjika selisih berat barang di kedua kantong Sisi tidak lebih dari 9 kg.

Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih.

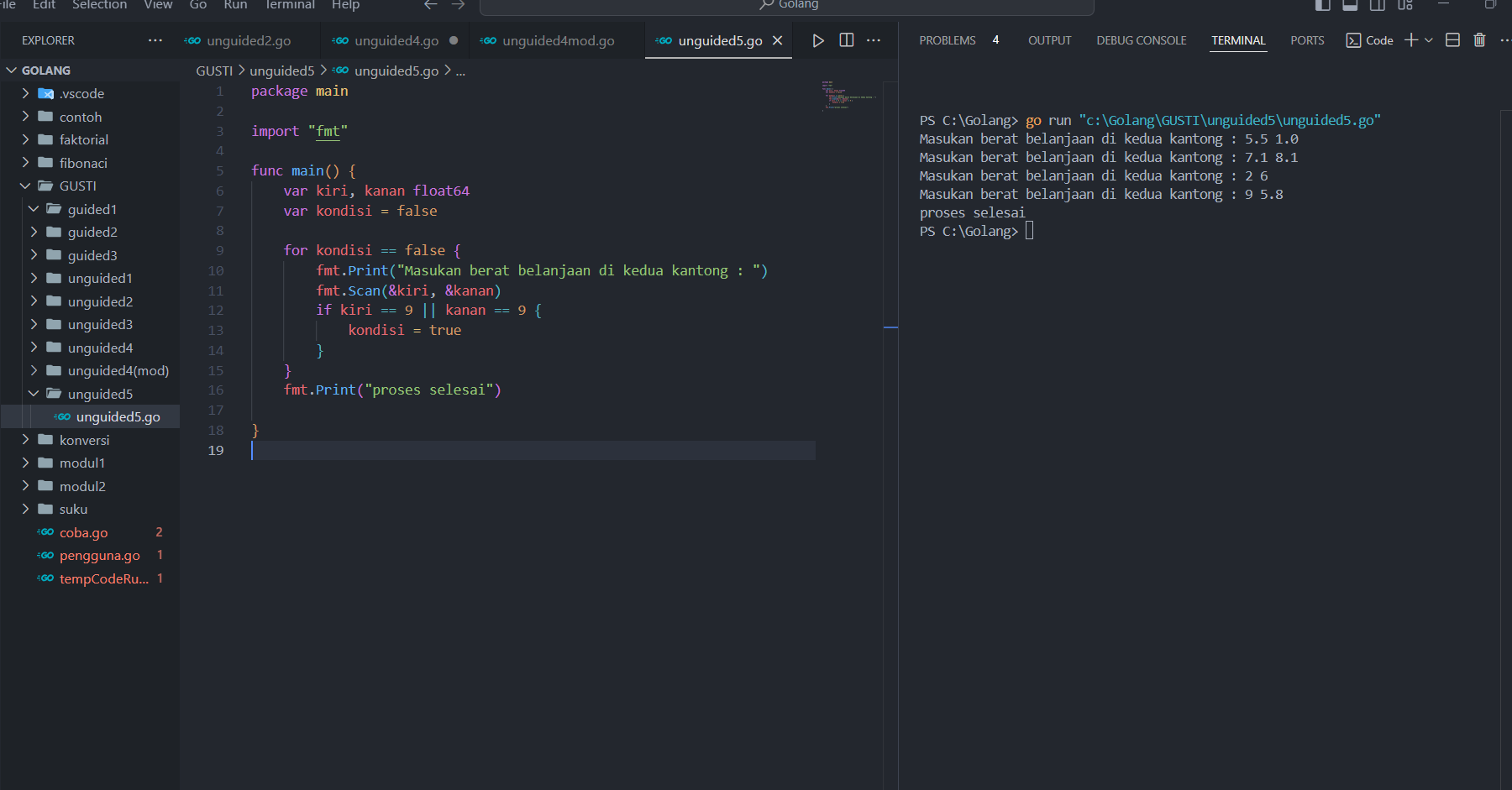
Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan 9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif.

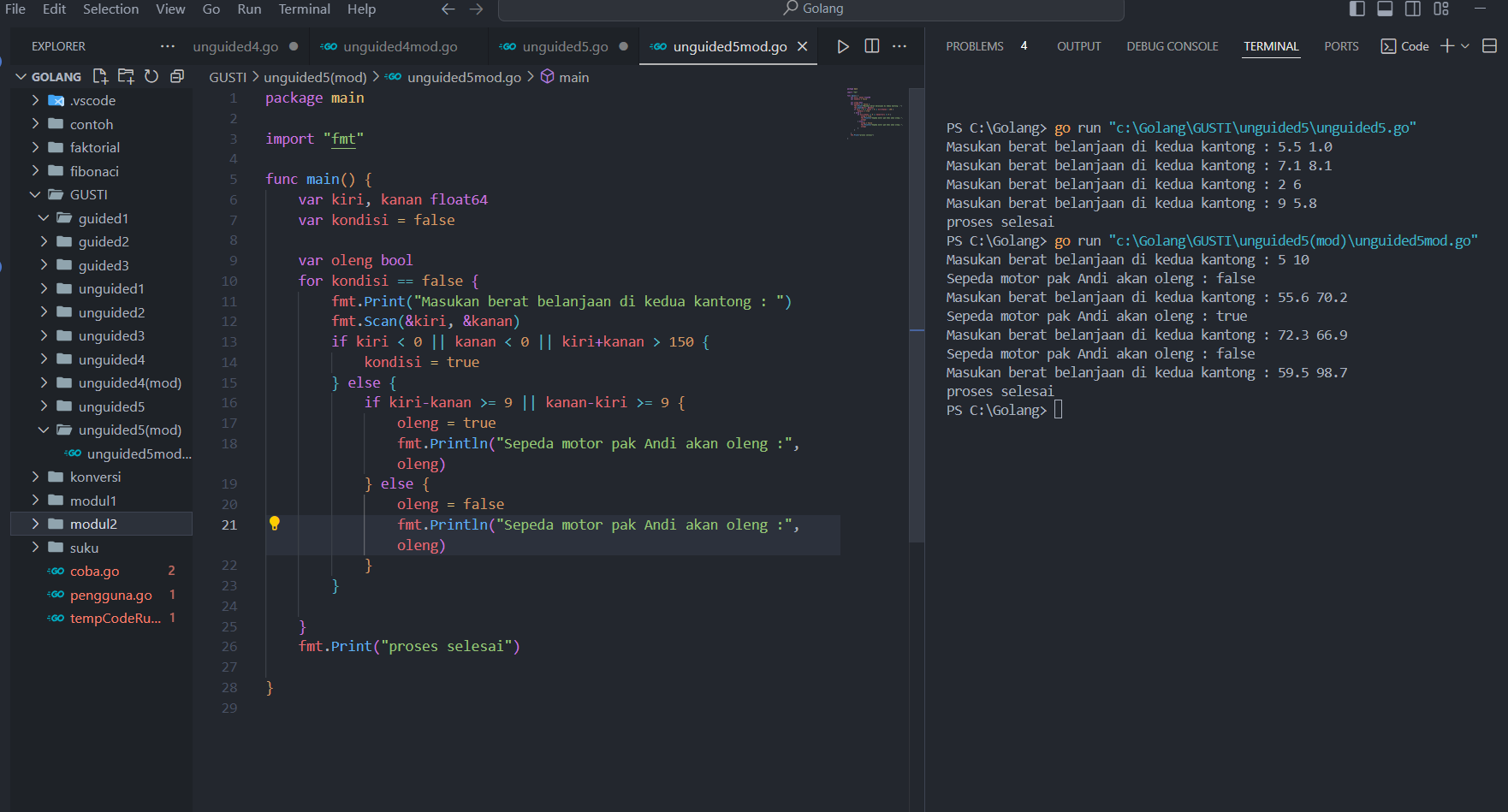
**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var kiri, kanan float64      var kondisi = false      for kondisi == false {          fmt.Print("Masukan berat belanjaan di kedua kantong : ")          fmt.Scan(&kiri, &kanan)          if kiri == 9 || kanan == 9 {              kondisi = true          }      }      fmt.Print("proses selesai")  } |

**Sourcecode(MODIFIKASI)**

|  |
| --- |
| package main  import "fmt"  func main() {      var kiri, kanan float64      var kondisi = false      var oleng bool      for kondisi == false {          fmt.Print("Masukan berat belanjaan di kedua kantong : ")          fmt.Scan(&kiri, &kanan)          if kiri < 0 || kanan < 0 || kiri+kanan > 150 {              kondisi = true          } else {              if kiri-kanan >= 9 || kanan-kiri >= 9 {                  oleng = true                  fmt.Println("Sepeda motor pak Andi akan oleng :", oleng)              } else {                  oleng = false                  fmt.Println("Sepeda motor pak Andi akan oleng :", oleng)              }          }      }      fmt.Print("proses selesai")  } |

**Screenshoot Output** ****

****

**Deskripsi Program**

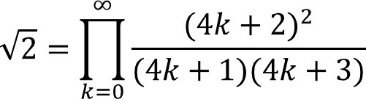
Program untuk menegcek berat belanjaan barang pak Andi dengan motor yang berada disebalah kanandan kiri. Terdapat variabel kana dan kiri untuk menginput berat barang pak Andi terdapat juga variabel kondisi dengan tipe data boolean. Program akan meminta user menginput berat barang di kanan dan kiri. Dengan menggunakan perulangan dengaan kondisi false sehingga porgram akan terus berjalan . kemudian if berat kiri atau kanan sama dengan 9 maka kondisi akan true sehingga perulangan akan terhenti dan program selesai

Modifikasi program untuk menampilkan apakan motor pak Andi oleng atau tidek dengan syarat tidak boleh selisih 9. sehingga ditambah varibel oleng dengan tipe data boolean. Kemduian pada perulangan sebelumnya ditambahkan kondisi if ketika selisih lebih dari 9 maka nilai oleng akan true dan program akan menampilkkan true. perulangan untuk menginput akan terus berjalan hingga nilai dari salah satu inputan nagatif atau jumlah keduanya 150 maka program akan berhenti

* + - 1. **Soal Studi Case**

Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai f(K) sesuai persamaan di atas.

Akar 2 merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihampiri dengan rumus berikut:



Modifikasi program sebelumnya yang menerima input integer K dan menghitung untuk K tersebut. Hampiran dituliskan dalam ketelitian 10 angka di belakang koma

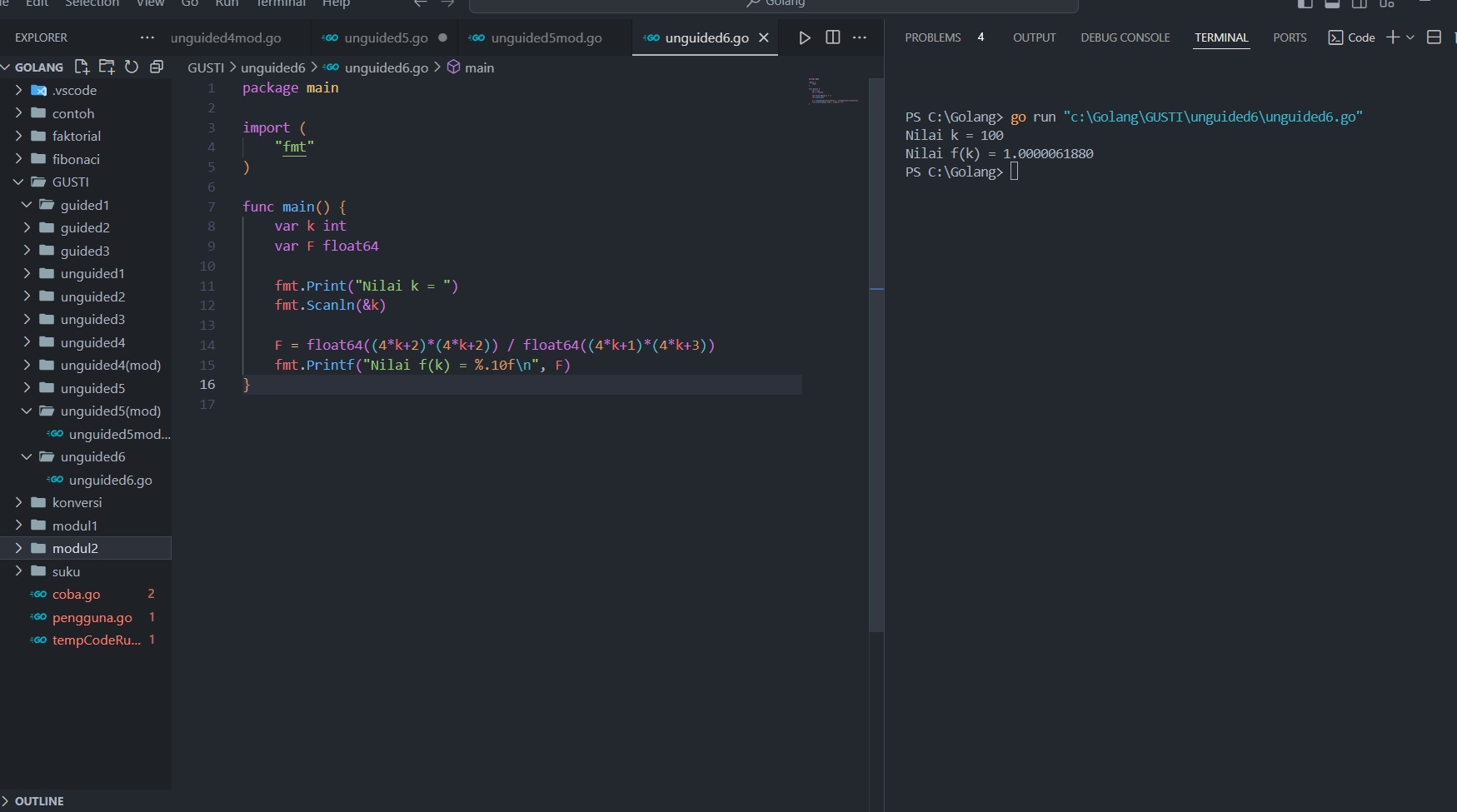
**Sourcecode**

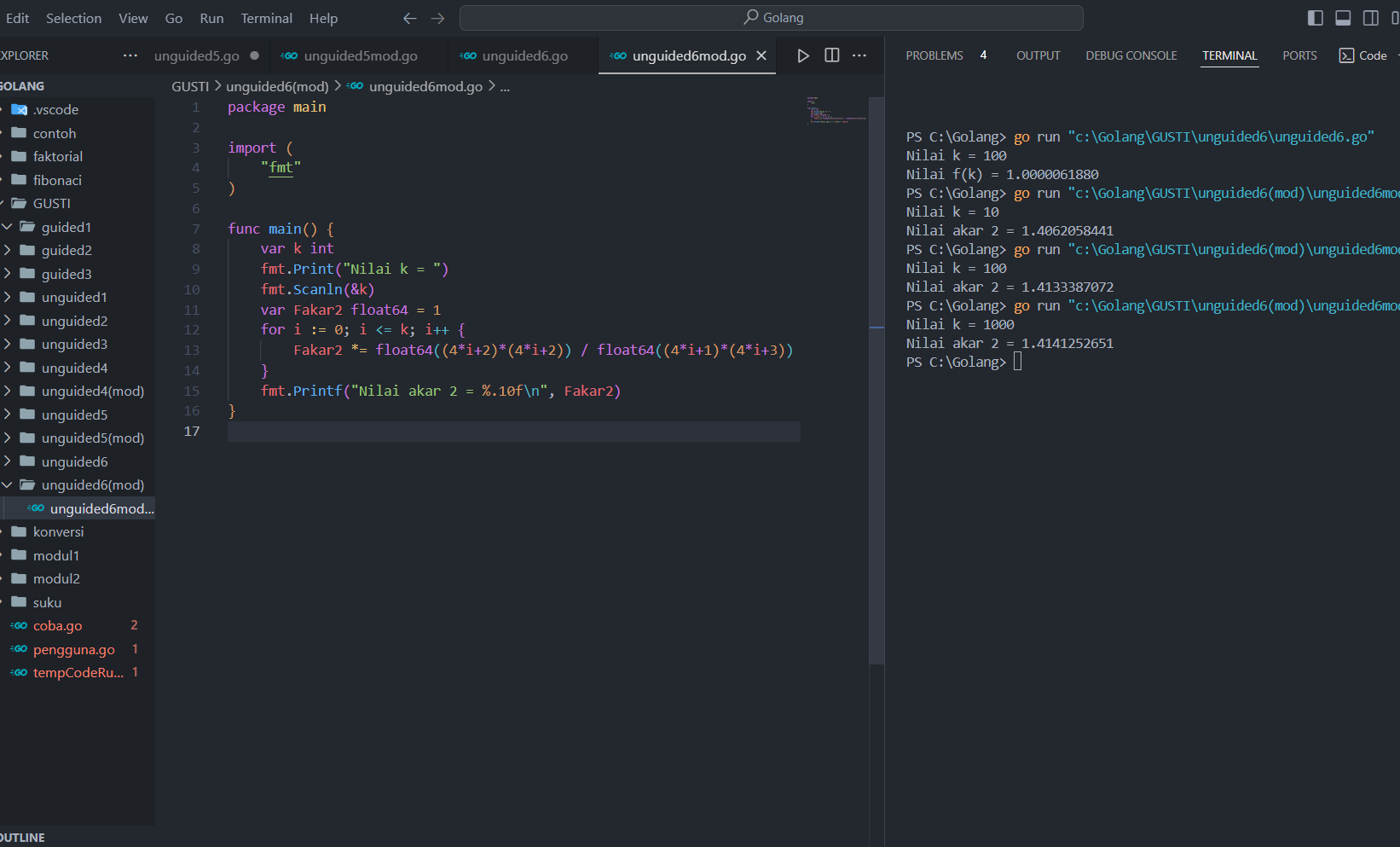
|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var k int      var F float64      fmt.Print("Nilai k = ")      fmt.Scanln(&k)      F = float64((4\*k+2)\*(4\*k+2)) / float64((4\*k+1)\*(4\*k+3))      fmt.Printf("Nilai f(k) = %.10f\n", F)  } |

**Sourcecode(MODIFIKASI)**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var k int      fmt.Print("Nilai k = ")      fmt.Scanln(&k)      var Fakar2 float64 = 1      for i := 0; i <= k; i++ {          Fakar2 \*= float64((4\*i+2)\*(4\*i+2)) / float64((4\*i+1)\*(4\*i+3))      }      fmt.Printf("Nilai akar 2 = %.10f\n", Fakar2)  } |

**Screenshoot Output**

****

****

**Deskripsi Program**

Program ini adalah program untuk menghitung nilai dari fung f(k). Dengan inisiasi varrabel k sebagi inputan. Kemudian nilai k akan dimasukan kedalam rumus dan hasilnya akan ditampilkan di output

Modifikasi program untuk mencari nilai dari akar 2 dengan menggunakan rumus yang sama hanya saja kita akan mengalikan hasilnya dari nilai k = 0 hingga kondisi yang ditentukan dalam hal ini batas k akan menjadi inputan user. Untuk menghitung akar 2 rumus akan dimasukan kedalam perulangan aghar dapat mengalikan hasilnya. Dengan kondisi perulanagn berjalan hingga nilai k. Kemudian output berupa hasil akhir dari perkalian berulang.

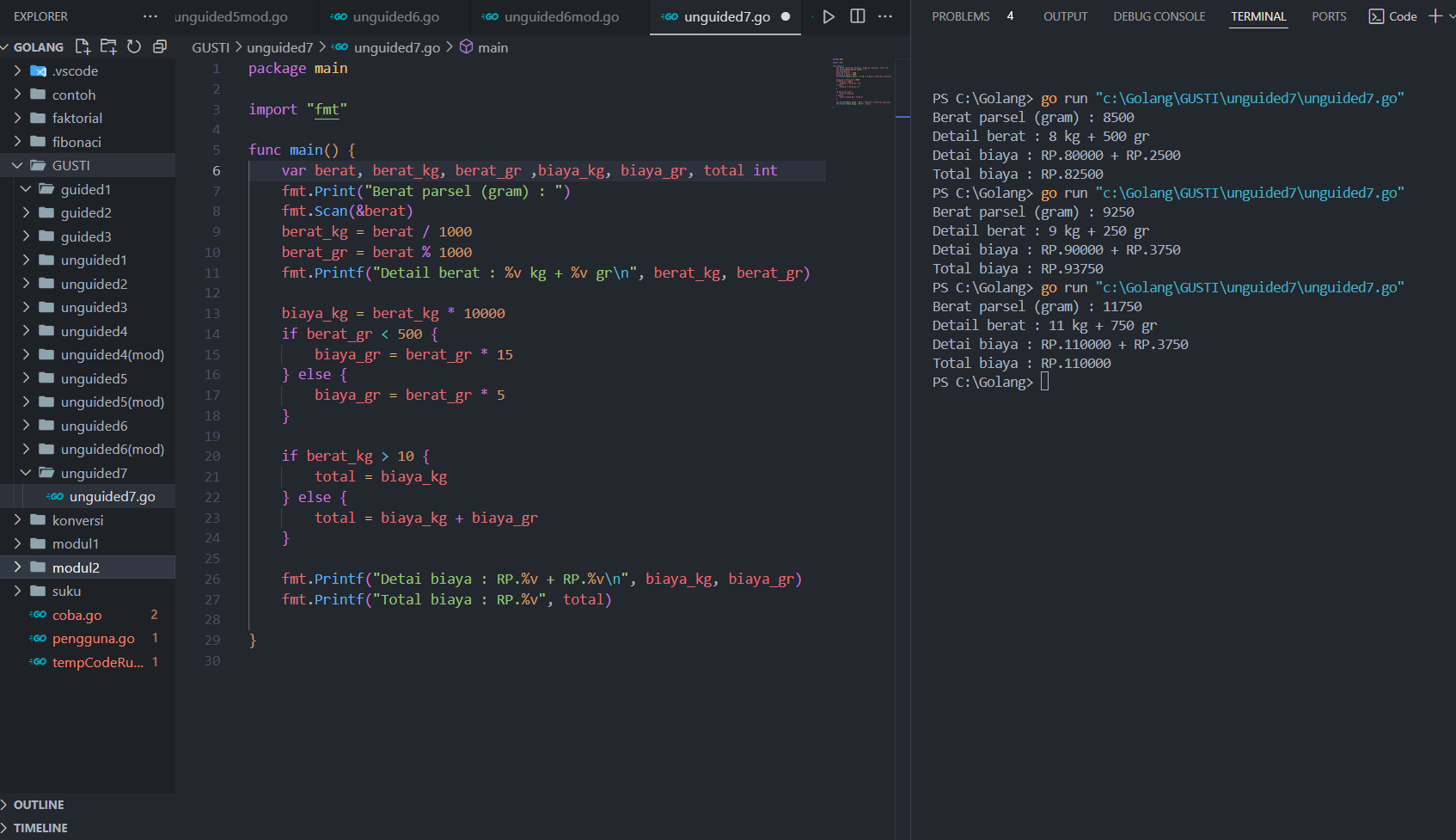
* + - 1. **Soal Studi Case**

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parsel. Maka, buatlah program BlayaPos untuk menghitung blaya pengiriman tersebut dengan ketentuan sebagai berlkut! Dari berat parsel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah RP. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya RP. 5,- per gram saja. Tetapijika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar RP. 15,- per gram. Sisa berat (yang kurang dari lkg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var berat, berat\_kg, berat\_gr ,biaya\_kg, biaya\_gr, total int      fmt.Print("Berat parsel (gram) : ")      fmt.Scan(&berat)      berat\_kg = berat / 1000      berat\_gr = berat % 1000      fmt.Printf("Detail berat : %v kg + %v gr\n", berat\_kg, berat\_gr)      biaya\_kg = berat\_kg \* 10000      if berat\_gr < 500 {          biaya\_gr = berat\_gr \* 15      } else {          biaya\_gr = berat\_gr \* 5      }      if berat\_kg > 10 {          total = biaya\_kg      } else {          total = biaya\_kg + biaya\_gr      }      fmt.Printf("Detai biaya : RP.%v + RP.%v\n", biaya\_kg, biaya\_gr)      fmt.Printf("Total biaya : RP.%v", total)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini adalah aplikasi perhitungan biaya kirim berdasarkan berat parsel. Terdapat deklarasi variabel untuk menyatakan berat, biaya dan total harga. Program akan menerima input an dari user berupa berat dengan satuan gram. Kemudian berat akan dibagi 1000 untuk menentukan dalam satuan kilo dan berat di modulus kan 1000 untuk menentukan sisa berat dalam satuan gram. Selanjutnya akan ditampilkan output dari detail beratnya berupa satuan KG dan Gram. Untuk menghitung biaya, untuk yang satuan kg akan dikalikan dengan 10.000 dan gram akan dikalikan tergantung kondisi if gram kurang dari 500 maka dikali 15, dan ketika 500 atau lebih maka akan dikali 5. Kemudian ioutput berupa detail biaya dan selanjutnya total keseluruhan. Terdapat juga kondisi di mana jika berat dalam kg lebih dari 10 maka harga dalam gram digratiskan.

* + - 1. **Soal Studi Case**

Diberikan sebuah nilai akhir mata kuliah (NAM) [O.. 100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

|  |  |
| --- | --- |
| NAM | NMK |
| NAM>80 |  |
| 72.5 < NAM < = 80 | AB |

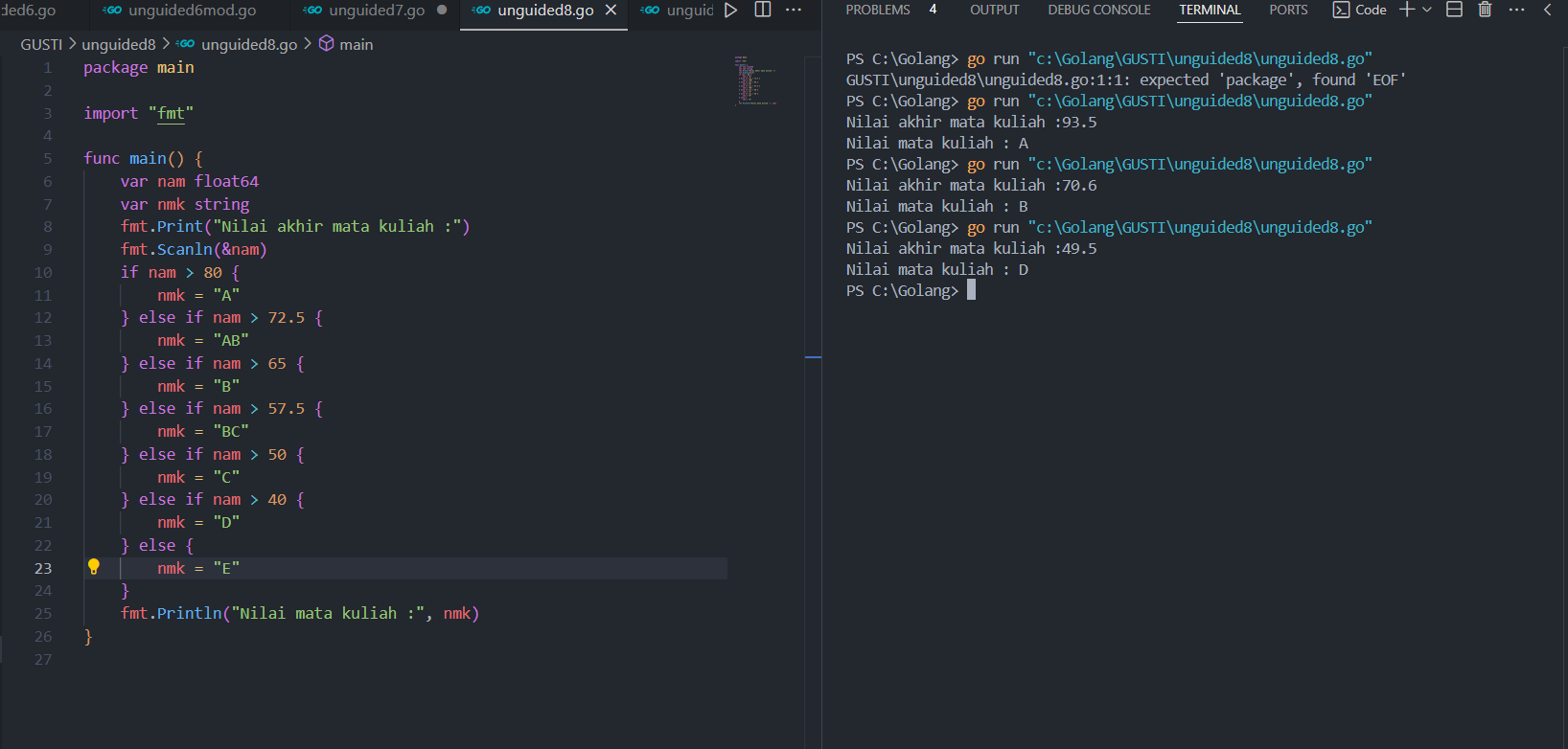
|  |  |
| --- | --- |
| 65 < NAM < = 72.5 |  |
| 57.5 < NAM < = 65 | BC |
| 50 < NAM < = 57.5 | c |
| 40 < NAM < = 50 | D |
| NAM |  |

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya

**Sourcecode**

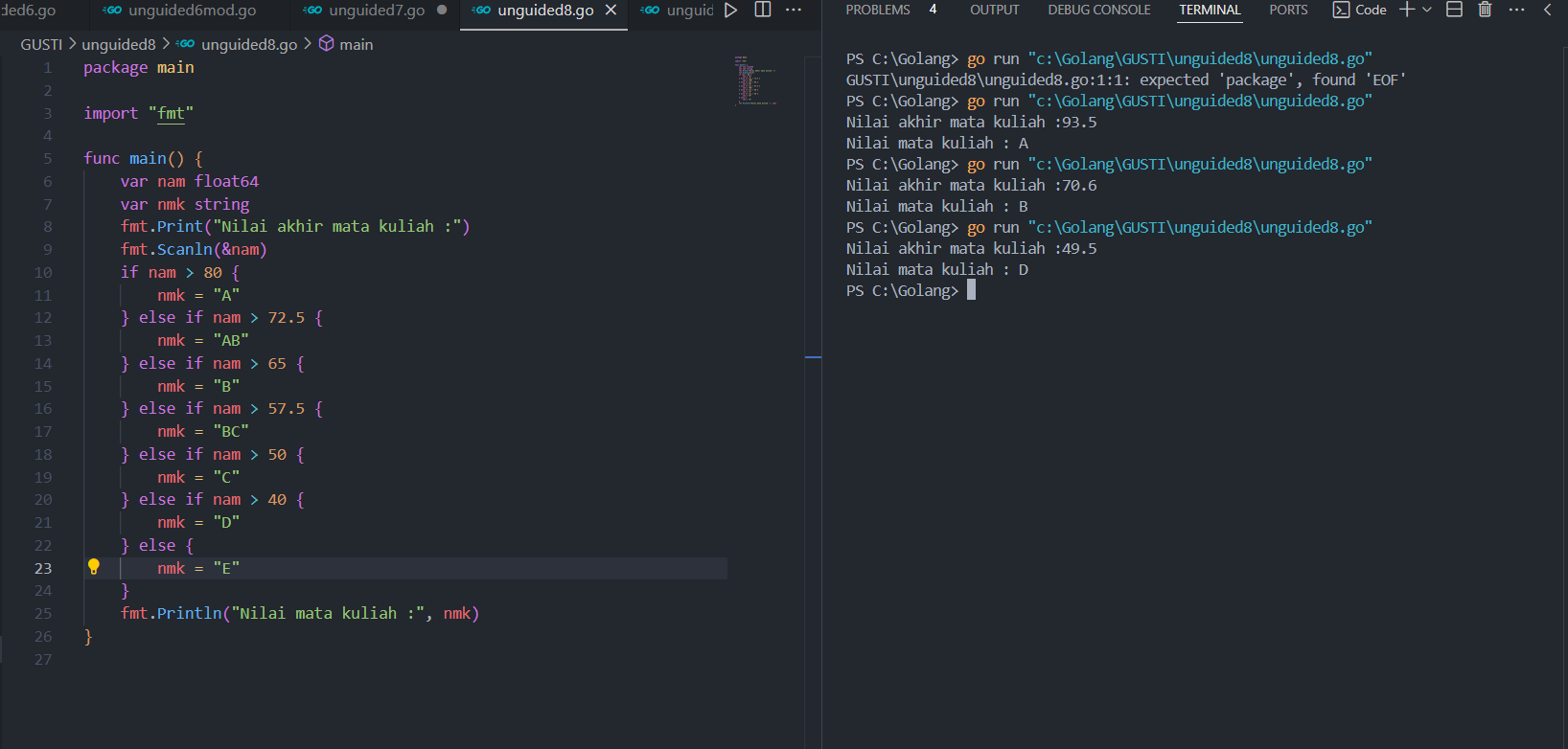
|  |
| --- |
| package main  import "fmt"  func main() {      var nam float64      var nmk string      fmt.Print("Nilai akhir mata kuliah :")      fmt.Scanln(&nam)      if nam > 80 {          nmk = "A"      } else if nam > 72.5 {          nmk = "AB"      } else if nam > 65 {          nmk = "B"      } 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 Output**



**Deskripsi Program**

1. Jika nam diberikan 80.1 maka output seharusnya adalah A namun sourcecode nya salah karena nilai nam adalah int bukan A yang seharunya itu adalah nilai dari var nmk  
   (nialai nmk tidak di inisiasikan dalam if) dan eksekusi program tidak sesuai dengan soal
2. Kelasahan dalam program tersebut adalah variabel nmk tidak digunakan. Kondisi if tidak lengkap yang mana tidak ada blok else ketika tidak dapat menangani kondisi if. Program seharusnya adalah dimulai dari user menginput nam . kemudian kondisi if-else if untuk menentukan katogori nilai nam dengan nmk yang berupa karakter dan yang terakhir menampikan output ( sesuai dengan sourcecode diatas)



* + - 1. **Soal Studi Case**

Sebuah bilangan bulat b memiliki faktor bilangan f > Ojika f habis membagi b. Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2.

Buatlah program yang menerima input sebuah bilangan bulat b dan b > 1. Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut!

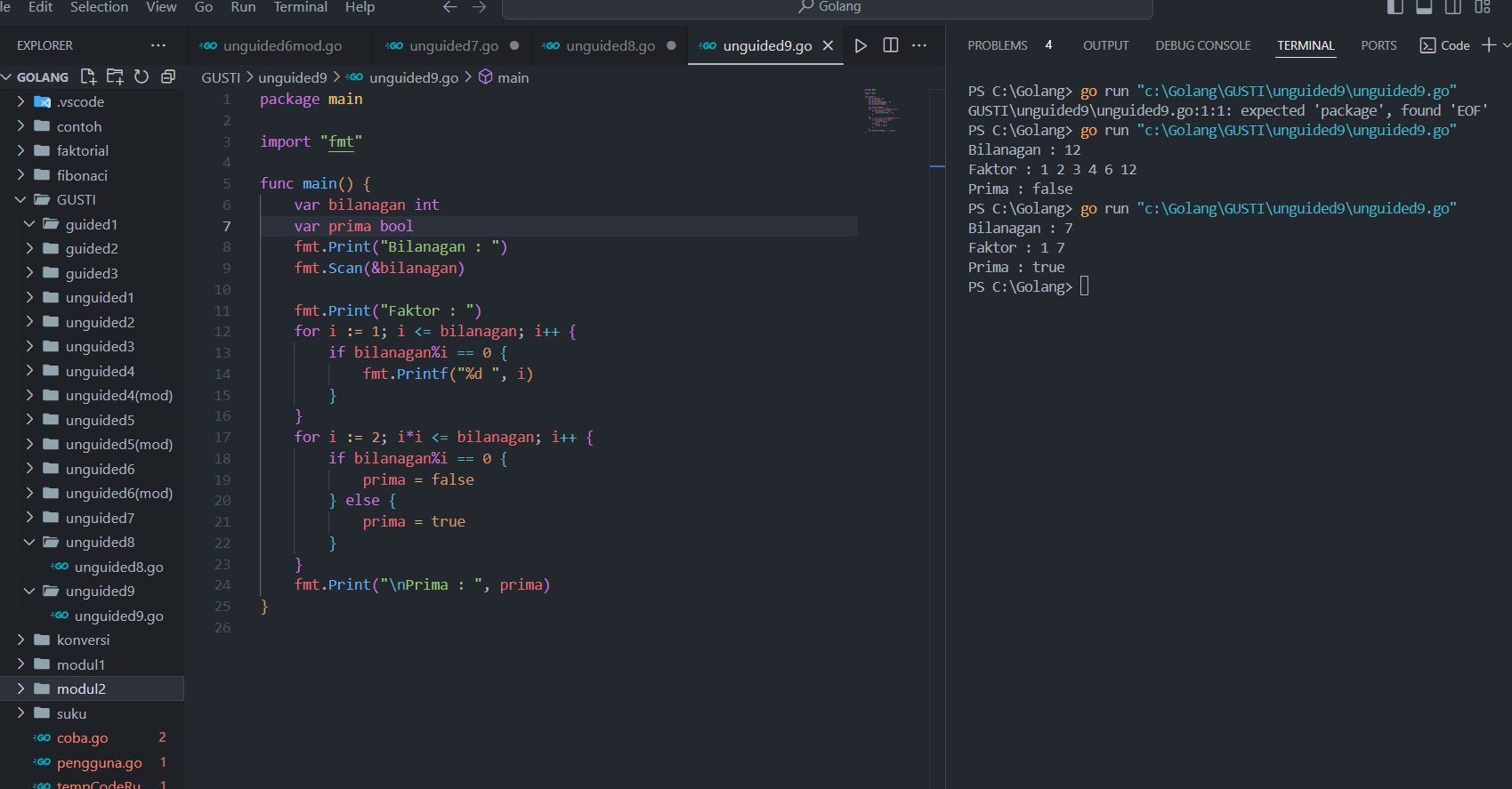
Bilangan bulat b > 0 merupakan bilangan prima pjika dan hanyajika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri.

Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat b > 0. Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah b merupakan bilangan prima.

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var bilanagan int      var prima bool      fmt.Print("Bilanagan : ")      fmt.Scan(&bilanagan)      fmt.Print("Faktor : ")      for i := 1; i <= bilanagan; i++ {          if bilanagan%i == 0 {              fmt.Printf("%d ", i)          }      }      for i := 2; i\*i <= bilanagan; i++ {          if bilanagan%i == 0 {              prima = false          } else {              prima = true          }      }      fmt.Print("\nPrima : ", prima)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini adalah program untuk menentukan faktor dari bilangan yang diinput dan menentukan apakah bilangan tersebut adalag bilaangan prima atau tidak. Terdapat deklarasivar bilanagan dengan tipe data int dan var prima dengan tipe data boolean untuk. Inputan dari user untuk mengisi nilai bilanagn. Perulanagn for pertama berfungsi untuk menentukan faktor dengan batas bilangan yang diinput kemudian terdapat if bilangan habis dibagi nilai i ketika kondisi terpenuhi maka akan menmapilkan nilai dari i tersebut atau bisa dibilang faktor dari bilangan yang diinput. Kemudian perulangan kedua untuk mengecek apakah bilangan yang dinput adalah bilangan prima dengan kondisi atas adalah akar kuadrat dari bilangan tersebut. If bilanagn habis dibagi i maka bukan prima, namun jika tidak maka bilangan tersebut adalah prima. Output kemudian akan menampilkan prima true