

**LAPORAN PRAKTIKUM
ALGORITME DAN PEMROGRAMAN 2**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

MUHAMMAD AMIR SALEH

2311102233

IF - 11 - 06

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. Dasar Teori

Sintaks Golang (Go) dirancang dengan kesederhanaan dan keterbacaan sebagai prioritas utama. Setiap program dimulai dengan deklarasi paket, dan jika programnya ingin dijalankan sebagai aplikasi, paket main wajib digunakan. Setelah itu, kita bisa mengimpor berbagai paket lain, seperti `fmt` yang umum dipakai untuk menangani input dan output. Di dalam Go, semua program dieksekusi dari fungsi `main`, yang menjadi titik masuk pertama kali. Untuk mendeklarasikan variabel, kita bisa menggunakan kata kunci `var`, atau dengan cara yang lebih singkat melalui operator `:=`, yang memungkinkan kita mendeklarasikan sekaligus menginisialisasi variabel secara bersamaan. Go juga memastikan penggunaan tipe data yang ketat, meskipun bisa secara otomatis menginferensikan tipe dari nilai yang diberikan. Kesederhanaan ini membuat Golang menjadi bahasa yang efisien untuk dipelajari dan digunakan.

II. Guided (Modul 2A)

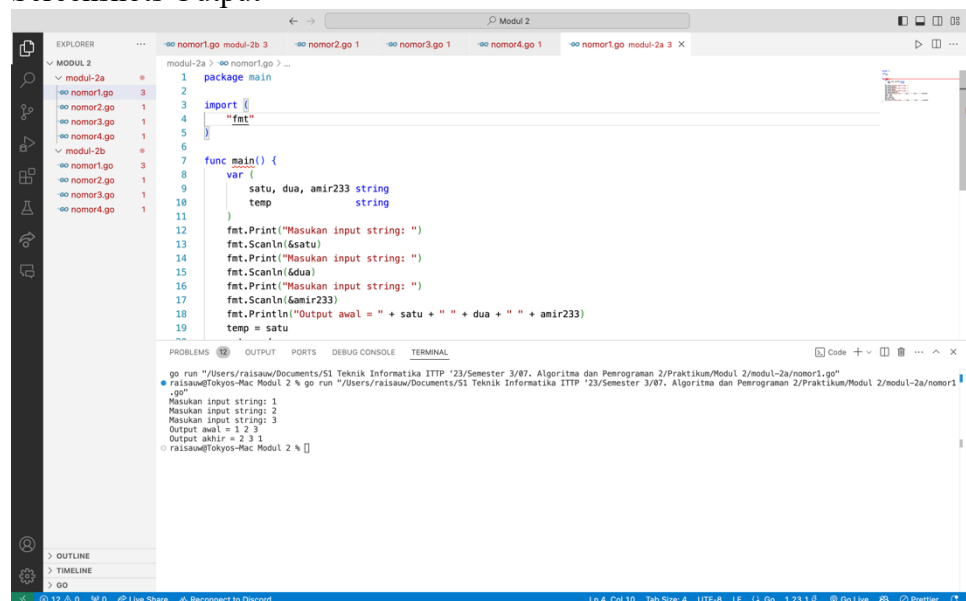
Guided 1

```
package main

import (
    "fmt"
)

func main() {
    var (
        satu, dua, amir233 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(&amir233)
    fmt.Println("Output awal = " + satu + " " + dua + " " +
amir233)
    temp = satu
    satu = dua
    dua = amir233
    amir233 = temp
    fmt.Println("Output akhir = " + satu + " " + dua + " " +
amir233)
}
```

Screenshots Output



The screenshot displays a Go IDE interface with the following components:

- EXPLORER:** Shows a project structure with folders 'MODUL 2' and 'modul-2a', and files 'modul-2a.go', 'modul-2b.go', 'modul-2c.go', 'modul-2d.go', 'modul-2e.go', 'modul-2f.go', 'modul-2g.go', 'modul-2h.go', 'modul-2i.go', 'modul-2j.go', 'modul-2k.go', 'modul-2l.go', 'modul-2m.go', 'modul-2n.go', 'modul-2o.go', 'modul-2p.go', 'modul-2q.go', 'modul-2r.go', 'modul-2s.go', 'modul-2t.go', 'modul-2u.go', 'modul-2v.go', 'modul-2w.go', 'modul-2x.go', 'modul-2y.go', 'modul-2z.go'.
- EDITOR:** Displays the code for 'modul-2a.go', which is the same code as shown in the previous block.
- OUTPUT:** Shows the output of the program, which is the same as shown in the previous block.
- TERMINAL:** Shows the command to run the program: `go run "C:/Users/raisaaw/Documents/S1 Teknik Informatika ITTP '23/Semester 3/87. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2a/modul-2a.go"`.

Deskripsi:

Kode ini adalah program untuk menukar nilai dari tiga variabel string yang diinputkan oleh pengguna. Pertama, program meminta pengguna memasukkan tiga string dan menyimpannya dalam variabel satu, dua, dan amir233. Setelah itu, program menampilkan ketiga string tersebut dalam urutan aslinya. Selanjutnya, program menukar nilai variabel-variabel tersebut menggunakan variabel sementara bernama temp. Nilai dari satu dipindahkan ke dua, nilai dari dua dipindahkan ke amir233, dan nilai dari amir233 dipindahkan ke satu. Akhirnya, program menampilkan urutan baru dari ketiga string tersebut setelah penukaran.

Guided 2

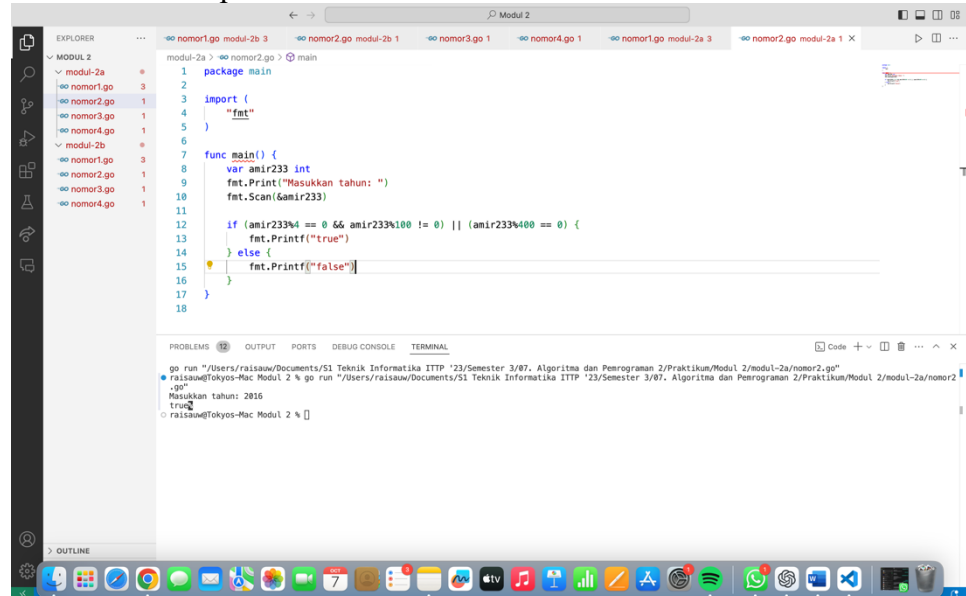
```
package main

import (
    "fmt"
)

func main() {
    var amir233 int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&amir233)

    if (amir233%4 == 0 && amir233%100 != 0) || (amir233%400 ==
0) {
        fmt.Printf("true")
    } else {
        fmt.Printf("false")
    }
}
```

Screenshots Output



Deskripsi:

Kode ini adalah program untuk menukar nilai dari tiga variabel string yang diinputkan oleh pengguna. Pertama, program meminta pengguna memasukkan tiga string dan menyimpannya dalam variabel satu, dua, dan amir233. Setelah itu, program menampilkan ketiga string tersebut dalam urutan aslinya. Selanjutnya, program menukar nilai variabel-variabel tersebut menggunakan variabel sementara bernama temp. Nilai dari satu dipindahkan ke dua, nilai dari dua dipindahkan ke amir233, dan nilai dari amir233 dipindahkan ke satu. Akhirnya, program menampilkan urutan baru dari ketiga string tersebut setelah penukaran.

Guided 3

```
package main

import "fmt"

func volumeBola(jari float64) float64 {
    const pi = 3.1415926535
    return (4.0 / 3.0) * pi * jari * jari * jari
}

func luasPermukaanBola(jari float64) float64 {
    const pi = 3.1415926535
    return 4 * pi * jari * jari
}

func main() {
```

```

var amir233 float64

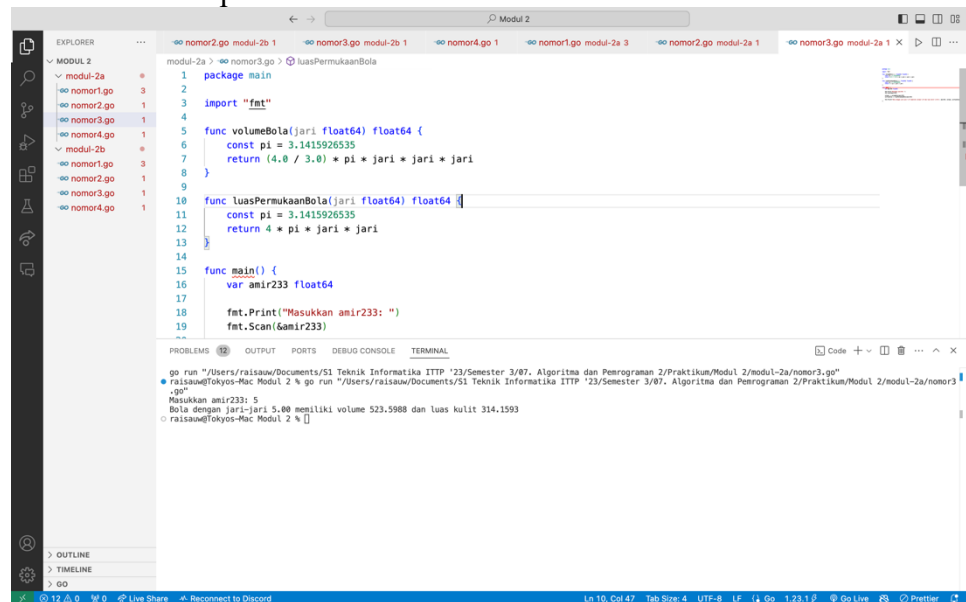
fmt.Print("Masukkan amir233: ")
fmt.Scan(&amir233)

volume := volumeBola(amir233)
surfaceArea := luasPermukaanBola(amir233)

fmt.Printf("Bola dengan jari-jari %.2f memiliki volume %.4f
dan luas kulit %.4f\n", amir233, volume, surfaceArea)
}

```

Screenshots Output



Deskripsi:

Kode ini adalah program dalam bahasa Go yang menghitung volume dan luas permukaan sebuah bola berdasarkan jari-jari yang dimasukkan pengguna. Program memiliki dua fungsi: `volumeBola`, yang menghitung volume bola menggunakan rumus, dan `luasPermukaanBola`, yang menghitung luas permukaan bola dengan rumus. Di dalam fungsi `main`, program meminta pengguna untuk memasukkan nilai jari-jari yang disimpan dalam variabel `amir233`, lalu memanggil kedua fungsi tersebut untuk menghitung volume dan luas permukaan bola. Hasil perhitungan kemudian ditampilkan dengan format yang rapi, menunjukkan jari-jari, volume, dan luas permukaan bola hingga dua dan empat angka desimal, masing-masing.

```
package main

import "fmt"

func volumeBola(jari float64) float64 {
    const pi = 3.1415926535
    return (4.0 / 3.0) * pi * jari * jari * jari
}

func luasPermukaanBola(jari float64) float64 {
    const pi = 3.1415926535
    return 4 * pi * jari * jari
}

func main() {
    var amir233 float64

    fmt.Print("Masukkan amir233: ")
    fmt.Scan(&amir233)

    volume := volumeBola(amir233)
    surfaceArea := luasPermukaanBola(amir233)

    fmt.Printf("Bola dengan jari-jari %.2f memiliki volume %.4f
dan luas kulit %.4f\n", amir233, volume, surfaceArea)
}
```

Screenshots Output



Deskripsi:

Kode ini adalah program untuk mengonversi suhu dari derajat Celsius ke derajat Fahrenheit. Program ini terdiri dari fungsi `amir233`, yang menerima parameter suhu dalam Celsius dan mengembalikan nilai suhu dalam Fahrenheit menggunakan rumus konversi. Dalam fungsi `main`, pengguna diminta untuk memasukkan nilai suhu dalam Celsius, yang disimpan dalam variabel `celsius`. Kemudian, program memanggil fungsi `amir233` untuk melakukan konversi dan menyimpan hasilnya dalam variabel `fahrenheit`. Akhirnya, hasil konversi suhu dalam Fahrenheit ditampilkan di layar.

III. Unguided (Modul 2B)

Unguided 1

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    amir233 := []string{"merah", "kuning", "hijau", "ungu"}

    numExperiments := 5

    for {
        success := true

        for i := 1; i <= numExperiments; i++ {
            var colors [4]string
            fmt.Printf("Percobaan %d: ", i)
            for j := 0; j < 4; j++ {
                fmt.Scan(&colors[j])
            }

            for j := 0; j < 4; j++ {
                if strings.ToLower(colors[j]) != amir233[j] {
                    success = false
                    break
                }
            }

            fmt.Printf("Percobaan %d: %s %s %s %s\n", i,
                colors[0], colors[1], colors[2], colors[3])
        }

        if success {
            fmt.Println("BERHASIL: true")
        } else {
            fmt.Println("BERHASIL: false")
        }

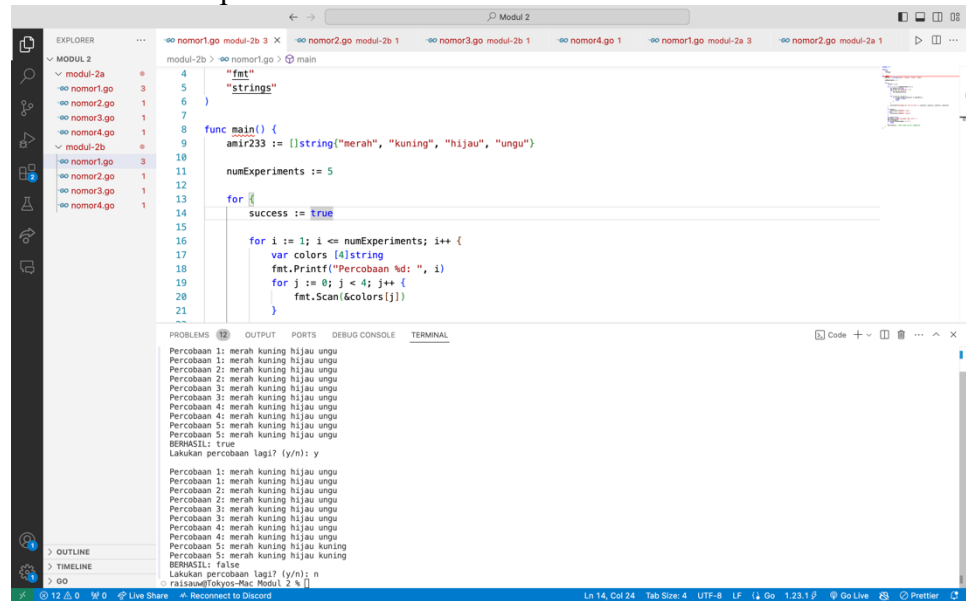
        var again string
        fmt.Print("Lakukan percobaan lagi? (y/n): ")
        fmt.Scan(&again)
        if strings.ToLower(again) != "y" {
            break
        }
    }
}
```

```

    }
    fmt.Println() // Add a blank line for readability
}
}

```

Screenshots Output



Deskripsi:

Kode ini adalah program untuk mencocokkan urutan warna tertentu. Program mendefinisikan array `amir233` yang berisi empat warna: "merah", "kuning", "hijau", dan "ungu". Pengguna diminta untuk melakukan sejumlah percobaan (ditentukan oleh variabel `numExperiments`, yang diatur ke 5) untuk memasukkan empat warna. Setelah pengguna memasukkan warna dalam setiap percobaan, program akan memeriksa apakah urutan warna yang dimasukkan sesuai dengan urutan dalam array `amir233`. Jika semua warna cocok, program mencetak "BERHASIL: true"; jika ada yang tidak cocok, ia mencetak "BERHASIL: false". Setelah semua percobaan selesai, pengguna ditanya apakah ingin melakukan percobaan lagi. Jika pengguna menjawab "y", program akan diulang; jika tidak, program akan berhenti.

Unguided 2

```

package main

import (
    "fmt"
    "strings"
)

```

```

func main() {
    var amir233 string
    var jumlahBunga int

    for {
        var namaBunga string
        jumlahBunga++
        fmt.Printf("Bunga %d: ", jumlahBunga)
        fmt.Scan(&namaBunga)

        if strings.ToUpper(namaBunga) == "SELESAI" {
            jumlahBunga--
            break
        }

        if amir233 == "" {
            amir233 = namaBunga
        } else {
            amir233 += " - " + namaBunga
        }
    }

    fmt.Println("amir233:", amir233)
    fmt.Println("Bunga:", jumlahBunga)
}

```

Screenshots Output

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Shows a project structure with files named `modul-2a`, `modul-2b`, `modul-2c`, `modul-2d`, `modul-2e`, `modul-2f`, `modul-2g`, `modul-2h`, `modul-2i`, `modul-2j`, `modul-2k`, `modul-2l`, `modul-2m`, `modul-2n`, `modul-2o`, `modul-2p`, `modul-2q`, `modul-2r`, `modul-2s`, `modul-2t`, `modul-2u`, `modul-2v`, `modul-2w`, `modul-2x`, `modul-2y`, `modul-2z`.
- EDITOR:** Displays the source code for `modul-2b`, which is the same code as shown in the previous block.
- TERMINAL:** Shows the output of the program:


```

go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/modul-2b.go"
raisaaw@Tokyo-Mac Modul 2 % go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/modul-2b.go"
Bunga 1: bunga1
Bunga 2: bunga2
Bunga 3: bunga3
Bunga 4: bunga4
Bunga 5: bunga5
Bunga 6: SELESAI
amir233: bunga1 - bunga2 - bunga3 - bunga4 - bunga5
Bunga: 5
raisaaw@Tokyo-Mac Modul 2 %

```

Deskripsi:

Kode ini adalah program untuk memasukkan nama-nama bunga satu per satu hingga pengguna mengetikkan “SELESAI”. Program dimulai dengan mendeklarasikan dua variabel: `amir233` untuk menyimpan daftar nama bunga, dan `jumlahBunga` untuk menghitung jumlah bunga yang dimasukkan. Dalam loop tanpa akhir, program meminta pengguna untuk memasukkan nama bunga, dan setiap kali pengguna memasukkan nama, variabel `jumlahBunga` akan bertambah. Jika pengguna mengetikkan “SELESAI” (dalam huruf besar), loop akan dihentikan, dan jumlah bunga akan dikurangi satu agar tidak menghitung kata “SELESAI” sebagai bunga. Selanjutnya, program akan mencetak daftar nama bunga yang dimasukkan, dipisahkan oleh “ - “, dan juga mencetak jumlah total bunga yang dimasukkan sebelum kata “SELESAI”.

Unguided 3

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var kantong1, kantong2 float64
    amir233 := 0.0

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        _, err := fmt.Scanf("%f %f", &kantong1, &kantong2)
        if err != nil {
            fmt.Println("Input tidak valid. Mohon masukkan dua angka.")
            continue
        }

        if kantong1 < 0 || kantong2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        amir233 = kantong1 + kantong2
        selisih := math.Abs(kantong1 - kantong2)
        akanOlen := selisih >= 9

        fmt.Printf("Sepeda motor pak Andi akan olen: %t\n",
            akanOlen)
```

```

        if amir233 > 150 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}

```

Screenshots Output

The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program with a package main, imports for fmt and math, and a main function. The main function defines two float64 variables, kantung1 and kantung2, and a float64 variable amir233. It enters a for loop where it prompts the user to enter weights for two bags. It uses fmt.Scanf to read input and checks for errors. If the input is not a valid float, it prints an error message and continues the loop. The terminal output shows the program execution with sample inputs and the resulting calculations and messages.

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var kantung1, kantung2 float64
    amir233 := 0.0

    for {
        fmt.Println("Masukan berat belanja di kedua kantong: ")
        err := fmt.Scanf("%f %f", &kantung1, &kantung2)
        if err != nil {
            fmt.Println("Input tidak valid. Mohon masukkan dua angka.")
            continue
        }
    }
}

```

```

go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITTP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/nomor3.go"
raisaaw@Tokyo-Mac Modu 2 % go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITTP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/nomor3.go"
Masukan berat belanja di kedua kantong: 50.0 59.0
Sepeda motor pak Andi akan oleng: true
Masukan berat belanja di kedua kantong: 50.0 58.0
Sepeda motor pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 50.0 280.0
Sepeda motor pak Andi akan oleng: true
Proses selesai.
raisaaw@Tokyo-Mac Modu 2 %

```

Deskripsi:

Kode ini adalah program untuk menghitung dan memantau berat belanjaan dalam dua kantong. Program meminta pengguna untuk memasukkan berat dari kedua kantong dalam satu baris. Jika input tidak valid (misalnya, bukan angka), program akan memberi tahu pengguna dan meminta input kembali. Pengguna dapat memasukkan berat kantong selama nilainya tidak negatif; jika salah satu dari kantong bernilai negatif, program akan menampilkan pesan “Proses selesai” dan berhenti. Program juga menghitung total berat belanjaan dari kedua kantong dan selisih berat antara keduanya. Jika selisih tersebut lebih besar atau sama dengan 9, program akan menyatakan bahwa sepeda motor Pak Andi “akan oleng”. Jika total berat belanjaan melebihi 150, program juga akan berhenti dan mencetak pesan “Proses selesai”.

Unguided 4

```

package main

import (

```

```

    "fmt"
    "math"
)

func f(k float64) float64 {
    amir233 := math.Pow(4*k+2, 2)
    denominator := (4*k + 1) * (4*k + 3)
    return amir233 / denominator
}

func approximateSqrt2(K int) float64 {
    result := 1.0
    for k := 0; k <= K; k++ {
        result *= f(float64(k))
    }
    return result
}

func main() {
    var K int

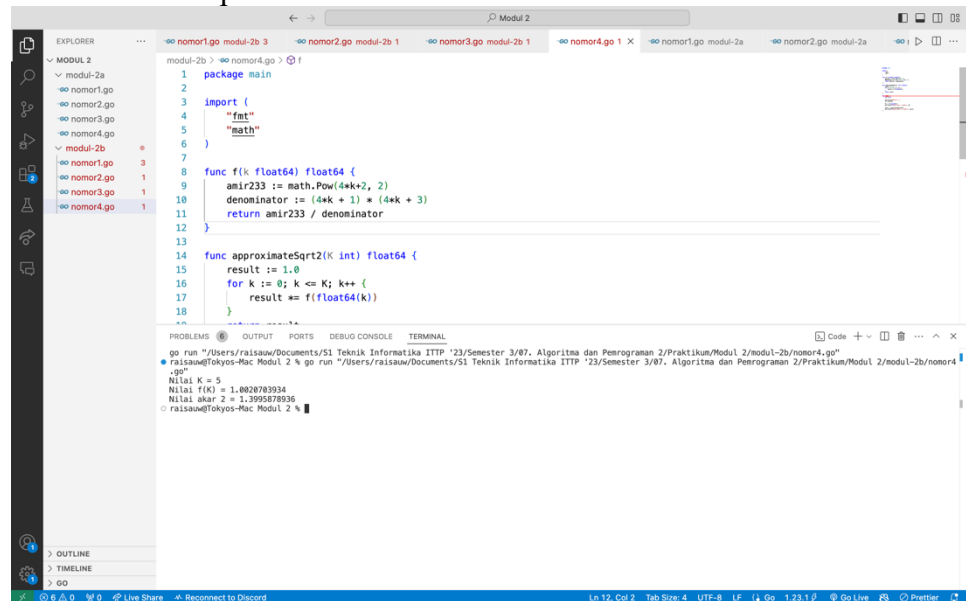
    fmt.Print("Nilai K = ")
    fmt.Scan(&K)

    fK := f(float64(K))
    fmt.Printf("Nilai f(K) = %.10f\n", fK)

    sqrt2 := approximateSqrt2(K)
    fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)
}

```

Screenshots Output



The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program with two functions: `f(k float64) float64` and `approximateSqrt2(K int) float64`. The terminal shows the output of running the program, which includes the value of `f(5)` and the approximate square root of 2.

```
package main

import (
    "fmt"
    "math"
)

func f(k float64) float64 {
    amir233 := math.Pow(4*k+2, 2)
    denominator := (4*k + 1) * (4*k + 3)
    return amir233 / denominator
}

func approximateSqrt2(K int) float64 {
    result := 1.0
    for k := 0; k <= K; k++ {
        result *= f(float64(k))
    }
}

func main() {
    K := 5
    result := approximateSqrt2(K)
    fmt.Println("Nilai akar 2 = ", result)
}
```

Terminal Output:

```
go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/nomor4.go"
raisaaw@Tokyo-Mac Modul 2 % go run "/Users/raisaaw/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/Modul 2/modul-2b/nomor4.go"
Nilai k = 5
Nilai f(k) = 1.0020789394
Nilai akar 2 = 1.3995878936
raisaaw@Tokyo-Mac Modul 2 %
```

Deskripsi:

Kode ini adalah program untuk menghitung nilai dari fungsi tertentu dan memberikan pendekatan untuk menghitung akar kuadrat dari 2 menggunakan deret produk. Program ini terdiri dari dua fungsi utama: `f(k float64)`, yang menghitung nilai berdasarkan rumus dengan pembilang sebagai kuadrat dari $4k + 2$ dan penyebut sebagai produk dari $(4k + 1)(4k + 3)$, serta `approximateSqrt2(K int)`, yang menghitung perkalian dari nilai `f(k)` untuk semua nilai `k` dari 0 hingga `K` untuk memperkirakan nilai akar kuadrat dari 2. Di dalam fungsi `main`, pengguna diminta untuk memasukkan nilai `K`, setelah itu program menghitung dan mencetak nilai dari fungsi `f(K)` serta nilai pendekatan akar kuadrat dari 2 berdasarkan `K` yang diberikan, dengan format yang menunjukkan sepuluh angka desimal.