

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
LATIHAN 2**



Oleh: Dimas Fanny Hebrasianto Permadi

NAMA: Dimas Ramadhani

NIM: 103112400065

KELAS: IF-12-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. TIPE SOAL A

1. Sewa Sepeda

- Source Code:

```
/*
Nama: Dimas Ramadhani
NIM: 103112400065
*/

package main

import "fmt"

func main() {
    var jam, menit int
    var voucher string
    var member bool
    fmt.Print("Masukkan durasi (jam): ")
    fmt.Scan(&jam)
    fmt.Print("Masukkan durasi (menit): ")
    fmt.Scan(&menit)
    fmt.Print("Apakah member? (true/false): ")
    fmt.Scan(&member)
    fmt.Print("Masukkan nomor voucher (jika ada): ")
    fmt.Scan(&voucher)
    fmt.Printf("Biaya sewa setelah diskon (jika memenuhi syarat): Rp
%.2f", hitungBiaya(jam, menit, member, voucher))
}

func hitungBiaya(jam, menit int, member bool, voucher string) float64 {
    durasi := totalJam(jam, menit)
    var hargaPerjam, harga float64
    if member == true {
        hargaPerjam = 3500.0
    } else {
        hargaPerjam = 5000.0
    }
    harga = durasi * hargaPerjam
    if durasi > 3 && cekVoucher(voucher) {
        harga = harga * 9 / 10
    }
    return harga
}
```

```

func cekVoucher(voucher string) bool {
    var hasil bool
    digit := len(voucher)
    if digit >= 5 {
        hasil = true
    } else {
        hasil = false
    }
    return hasil
}

func totalJam(jam, menit int) float64 {
    var durasi float64
    if jam < 1 {
        durasi = 1
    } else if jam >= 1 && menit <= 10 {
        durasi = float64(jam)
    } else if jam >= 1 && menit > 10 {
        durasi = float64(jam) + float64(menit)/60
    }
    return durasi
}

```

- **Screenshot Hasil Pembahasan:**

```

PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan P
\Collage\Semester 2\Algoritma dan Pemrograman 2\Praktikum\Lap
Masukkan durasi (jam): 2
Masukkan durasi (menit): 30
Apakah member? (true/false): true
Masukkan nomor voucher (jika ada): 123456
Biaya sewa setelah diskon (jika memenuhi syarat): Rp 8750.00

```

- **Penjelasan:**

Program ini bertujuan untuk mengetahui total harga sewa sepeda berdasarkan durasi, keanggotaan dan nomor voucher. Program menggunakan fungsi iteratif untuk mengetahui total harga sewa sepeda.

Pada fungsi utama, dideklarasikan variabel integer yaitu jam dan menit, lalu string yaitu voucher, dan boolean yaitu member. Fungsi ini dimulai dengan mencetak kalimat “Masukkan durasi (jam):” lalu program akan membaca sebagai variabel jam, selanjutnya melakukan hal yang sama lalu membaca variabel menit lalu member lalu terakhir voucher. Setelah kita memasukkan voucher, program akan mencetak hasil dari fungsi hitungBiaya.

Program ini menggunakan prosedur dengan nama hitungBiaya bertipe float64, dengan parameter jam dan menit int, member boolean, voucher string. Prosedur ini berfungsi menghitung total harga sewa sepeda berdasarkan jam, menit, member, dan voucher nya. Hal pertama yang dilakukan fungsi ini mendeklarasikan sekaligus memberikan nilai pada variabel durasi = totalJam(jam, menit). totalJam adalah fungsi lain. Lalu mendeklarasikan hargaPerjam, harga bertipekan float64. Masuk ke percabangan dengan kondisi berdasarkan member jika true akan menjalankan hargaPerjam=3500.0, selain itu akan menjalankan hargaPerjam=5000.0. setelah itu pemberian nilai untuk variabel $harga = durasi \times hargaPerjam$. Lalu masuk ke percabangan dengan kondisi durasi lebih dari 3 dan cekVoucher(voucher) akan menjalankan $harga = harga \times \frac{9}{10}$, cekVoucher juga merupakan fungsi lain dengan parameter voucher. Lalu akan mengembalikan variabel harga.

Lalu ada fungsi yang bernama cekVoucher bertipe boolean, dengan parameter voucher tipe string. Prosedur ini berfungsi untuk mengecek apakah pelanggan memiliki voucher yang valid dengan digit lebih dari sama dengan 5. Fungsi ini akan mendeklarasikan variabel hasil tipe boolean lalu pemberian nilai pada variabel digit=len(voucher), len berfungsi untuk menghitung jumlah digit/huruf pada variabel string. Lalu masuk ke percabangan dengan kondisi digit lebih dari sama dengan 5 akan menjalankan hasil=true,

selain itu akan menjalankan `hasil=false`. Program akan mengembalikan variabel `hasil`.

Lalu ada fungsi yang bernama `totalJam` bertipe `float64`, dengan parameter `jam`, `menit` tipe `integer`. Prosedur ini berfungsi untuk mengetahui total durasi penyewaan sepeda dengan satuan jam. Fungsi ini akan mendeklarasikan variabel `durasi` tipe `float64`. Lalu masuk ke percabangan dengan kondisi `jam` kurang dari 1 akan menjalankan `durasi=1`, selain itu jika kondisi `jam` lebih dari sama dengan 1 dan `menit` kurang dari sama dengan 10 akan menjalankan `durasi=jam`, selain itu akan menjalankan $durasi = jam + \frac{menit}{60}$. Program akan mengembalikan variabel `durasi`.

2. Perfect Number

- Source Code:

```
/*
Nama: Dimas Ramadhani
NIM: 103112400065
*/

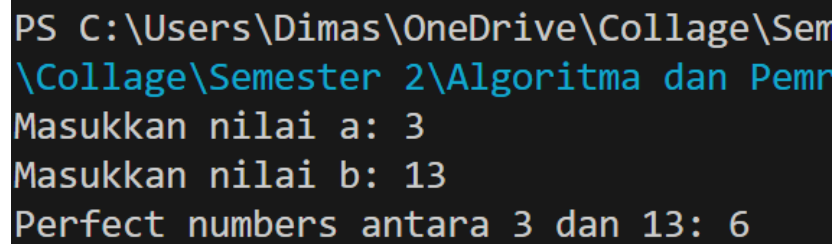
package main

import "fmt"

func main() {
    var a, b int
    fmt.Print("Masukkan nilai a: ")
    fmt.Scan(&a)
    fmt.Print("Masukkan nilai b: ")
    fmt.Scan(&b)
    fmt.Printf("Perfect numbers antara %d dan %d:", a, b)
    perfectNumber(a, b)
}

func perfectNumber(a, b int) {
    var jumlah int
    if a > b {
        return
    }
    for i := 1; i <= a; i++ {
        if a%i == 0 {
            if i != a {
                jumlah += i
            }
        }
    }
    if jumlah == a {
        fmt.Print(" ", jumlah)
    }
    perfectNumber(a+1, b)
}
```

- **Screenshot Hasil Pembahasan:**



```
PS C:\Users\Dimas\OneDrive\Collage\Sem
\Collage\Semester 2\Algoritma dan Pemr
Masukkan nilai a: 3
Masukkan nilai b: 13
Perfect numbers antara 3 dan 13: 6
```

- **Penjelasan:**

Program ini bertujuan untuk mencari bilangan yang termasuk perfect number antara dua bilangan yang kita masukkan. Perfect number adalah bilangan yang berasal dari penjumlahan dari faktor suatu bilangan asal kecuali bilangan asalnya itu sendiri. Program menggunakan rekursi dalam Prosedur perfectNumber untuk mencetak angka yang termasuk perfect number.

Pada program utama, dengan mendeklarasikan variabel a dan b tipe int, berfungsi untuk menyimpan rentang angka yang diinput oleh pengguna. Program akan mencetak kalimat lalu menginputkan bilangan dengan variabel a. Lalu mencetak kalimat lagi dan menginputkan bilangan variabel b. Setelah itu akan tercetak hasilnya dengan memanggil prosedur perfectNumber.

Program ini menggunakan prosedur dengan nama perfectNumber, dengan parameter a dan b int, berfungsi untuk mencetak angka yang termasuk perfect number. Prosedur ini menggunakan pendekatan rekursif untuk mencetak angka. Baris rekursi, jika $a > b$ maka prosedur berhenti (tidak mencetak apa pun). Panggilan rekursif, memanggil prosedur perfectNumber(a+1, b) untuk mencari angkanya dimulai dari a hingga b. Mendeklarasikan variabel jumlah tipe integer. Masuk ke percabangan sebagai struktur berhenti pada rekursif dengan kondisi $a > b$ maka akan return. Lalu masuk ke perulangan dengan iterasi dimulai dari 1, kondisi iterasi $\leq a$, iterasi akan ditambahkan satu setiap perulangan. Didalam perulangan akan menjalankan percangan dengan kondisi a dimoduluskan dengan i sama dengan 0 akan menjalankan struktur percangan lain dengan kondisi i tidak sama dengan a maka akan menjalankan variabel jumlah ditambahkan dengan variabel i. Keluar dari perulangan akan lanjut ke struktur percabangan dengan kondisi jumlah sama dengan a, akan mencetak variabel jumlah. Setelah itu ada bagian rekursinya yaitu perfectNumber(a+1, b).

3. Rendezvous

- **Source Code:**

```
/*
Nama: Dimas Ramadhani
NIM: 103112400065
*/

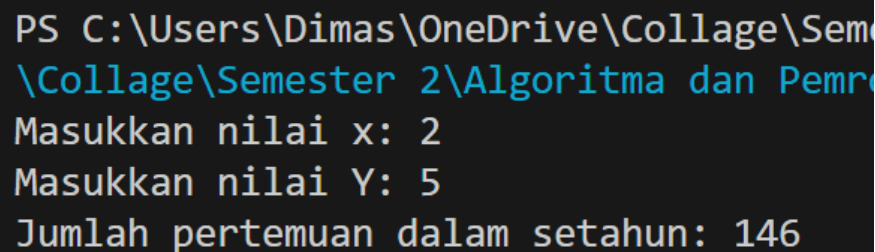
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan nilai Y: ")
    fmt.Scan(&y)
    fmt.Print("Jumlah pertemuan dalam setahun: ", rendezvous(x, y))
}

func rendezvous(x, y int) int {
    var jumlah int
    for i := 1; i <= 365; i++ {
        if i%x == 0 && i%y != 0 {
            jumlah += 1
        }
    }
    return jumlah
}
```

- **Screenshoot Hasil Pembahasan:**



```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman>
Masukkan nilai x: 2
Masukkan nilai Y: 5
Jumlah pertemuan dalam setahun: 146
```

- **Penjelasan:**

Program ini bertujuan untuk menghitung jumlah pertemuan antara dua orang dengan syarat akan bertemu pada hari kelipatan x merupakan inputan pengguna dan tidak akan bertemu pada hari ke y juga dari inputan pengguna. Program ini menggunakan iterasi pada

prosedur rendezvous untuk mencari jumlah hari yang termasuk syaratnya.

Program dimulai dengan fungsi utamanya, pertama mendeklarasikan variabel x dan y tipe integer lalu akan tercetak kalimat dan setelahnya pengguna dapat menginputkan angka akan discan sebagai variabel x, lalu program akan mencetak kalimat lagi dan melakukan scan variabel y. Setelah itu akan mencetak kalimat sekaligus hasil dari prosedur rendezvous.

Prosedur dengan nama rendezvous, dengan parameter variabel x dan y tipe int, untuk variabel x berfungsi sebagai hari kelipatan x sedangkan variabel y berguna untuk hari kelipatan y. Fungsi prosedur ini menggunakan metode iteratif untuk mencari jumlah hari yang termasuk syaratnya. Prosedur ini mendeklarasikan variabel jumlah tipe integer. Lalu melakukan perulangan dengan iterasi awal 1 sampai 365 pada setiap perulangan variabel iterasi akan ditambahkan 1. Di dalam perulangan ada percabangan dengan kondisi i habis dibagi x dan i tidak habis dibagi y maka akan menambahkan variabel jumlah dengan satu. Keluar dari perulangan, prosedur akan mengembalikan nilai variabel jumlah.

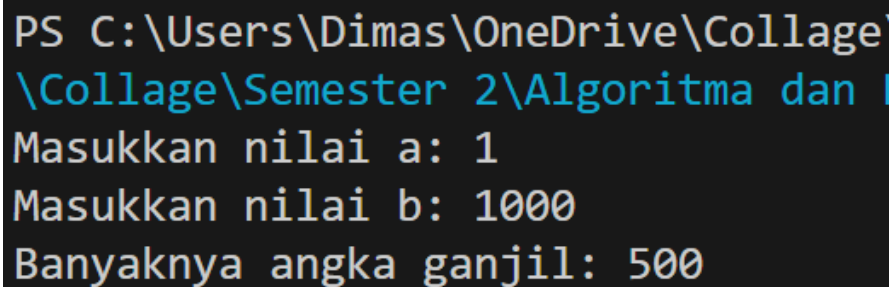
II. TIPE SOAL B

1. Penghitung Angka Ganjil

- **Source Code:**

```
/*  
Nama: Dimas Ramadhani  
NIM: 10311240065  
*/  
package main  
  
import "fmt"  
  
func main() {  
    var bil1, bil2, hasil int  
    fmt.Print("Masukkan nilai a: ")  
    fmt.Scan(&bil1)  
    fmt.Print("Masukkan nilai b: ")  
    fmt.Scan(&bil2)  
    voucher(bil1, bil2, &hasil)  
    fmt.Print("Banyaknya angka ganjil: ", hasil)  
}  
  
func voucher(a, b int, c *int) {  
    for i := a; i <= b; i++ {  
        if i%2 != 0 {  
            *c += 1  
        }  
    }  
}
```

- **Screenshot Hasil Pembahasan:**



```
PS C:\Users\Dimas\OneDrive\Collage\Collage\Semester 2\Algoritma dan  
Masukkan nilai a: 1  
Masukkan nilai b: 1000  
Banyaknya angka ganjil: 500
```

- **Penjelasan:**

Program ini bertujuan untuk menampilkan jumlah suatu bilangan ganjil berdasarkan rentang yang akan diberikan oleh pengguna. Program menggunakan iteratif untuk mencari dan mencatat jumlah suatu bilangan ganjil.

Pada fungsi utama atau prosedur utama, mendeklarasikan variabel bil1, bil2, hasil tipe integer. Lalu akan mencetak kalimat dan melakukan scan sebagai variabel bil1, lalu mencetak kalimat dan scan sebagai variabel bil2. Setelah itu fungsi utama akan menjalankan fungsi voucher untuk mencari banyaknya angka ganjil. Setelah itu akan tercetak hasil dan banyaknya angka ganjil melalui variabel hasil yang pada prosedur voucher sebagai variabel c.

Program ini menggunakan prosedur dengan nama voucher bertipe int, dengan parameter a dan b int dan variabel c pointer integer, dengan a menunjukkan posisi angka terkahir. Prosedur ini berfungsi mencari banyaknya angka ganjil dengan rentang a ke b menggunakan iteratif. Prosedur ini melakukan perulangan dengan iterasi sama dengan a, kondisi i kurang dari sama dengan b, dan setiap perulangan i akan ditambahkan 1. Pada perulangan akan masuk ke percabangan dengan kondisi i tidak habis dibagi 2, maka akan menjumlahkan variabel c dengan 1.

2. Restoran

- Source Code:

```
/*  
Nama: Dimas Ramadhani  
NIM: 10311240065  
*/  
package main  
  
import "fmt"  
  
func main() {  
    var rombongan int  
    fmt.Print("Masukkan jumlah rombongan: ")  
    fmt.Scan(&rombongan)  
    totalBayar(rombongan)  
  
}  
  
func totalBayar(a int) {  
    var jumlahMenu, banyakOrang, sisa, harga int  
    for i := 1; i <= a; i++ {  
        fmt.Print("Masukkan jumlah menu, jumlah orang, dan  
status sisa makanan (0 untuk tidak, 1 untuk iya): ")  
        fmt.Scan(&jumlahMenu, &banyakOrang, &sisa)  
        if jumlahMenu > 50 {  
            harga = 100000  
        } else if jumlahMenu > 3 {  
            harga = 10000 + 2500*jumlahMenu  
        } else if jumlahMenu <= 3 {  
            harga = 10000  
        }  
  
        if sisa == 1 {  
            harga *= banyakOrang  
        }  
        fmt.Printf("Total biaya untuk rombongan %d: Rp %d \n", i,  
harga)  
    }  
}
```

- **Screenshot Hasil Pembahasan:**

```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\Algoritma dan Pemrograman 2\Praktikum\Lapr
ak\103112400065_Latihan2\Soal Tipe B> go run "c:\Users\Dimas\OneDrive\Collage\Semester 2
\Algoritma dan Pemrograman 2\Praktikum\Laprak\103112400065_Latihan2\Soal Tipe B\restoran
_103112400065.go"
Masukkan jumlah rombongan: 2
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya)
: 3 12 0
Total biaya untuk rombongan 1: Rp 10000
Masukkan jumlah menu, jumlah orang, dan status sisa makanan (0 untuk tidak, 1 untuk iya)
: 2 15 1
Total biaya untuk rombongan 2: Rp 150000
```

- **Penjelasan:**

Program ini bertujuan untuk mencetak mengetahui total harga menu dari setiap rombongan berdasarkan jumlah menu, orang dan sisa makanan. Program menggunakan iteratif untuk mencari jumlah harga menu tiap rombongan.

Pada prosedur utama, mendeklarasikan variabel rombongan tipe integer. Lalu mencetak kalimat dan scan variabel rombongan. Setelah itu akan memanggil prosedur totalBayar.

Program ini menggunakan prosedur dengan nama totalBayar, dengan parameter a int, berfungsi untuk menjumlahkan atau mencetak hasil total harga dari menu. Prosedur ini menggunakan pendekatan iteratif untuk mencetak hasil. Mendeklarasikan variabel jumlahMenu, banyakOrang, sisa, harga int. Program ini menjalankan struktur perulangan dengan iterasi dimulai dari 1, kondisi i kurang dari sama dengan a, setiap perulangan variabel i akan ditambahkan satu. Di dalam perulangan akan mencetak kalimat lalu akan scan secara terurut variabel jumlahMenu, banyakOrang, sisa. Setelah itu masuk ke percabangan dengan kondisi jumlahMenu lebih dari 50 akan memberikan nilai pada variabel *harga* = 100000, selain itu jika jumlah menu lebih dari 3 maka variabel *harga* = $10000 + 2500 \times \text{jumlahMenu}$, selain itu jika jumlahMenu kurang dari sama dengan 3 maka *harga* = 10000. Setelah percabangan pertama akan masuk ke percabangan lain dengan kondisi sisa sama dengan 1 maka akan mengalikan variabel harga dengan banyakOrang, fungsi percabangan ini untuk mengecek nilai sisa jika sama dengan 1 artinya rombongan memiliki sisa. Setelah itu akan tercetak kalimat seklaigus hasil total harga menu makanan rombongan.

3. Jumlah Bilangan Positif Kelipatan 4

- Source Code:

```
/*
Nama: Dimas Ramadhani
NIM: 10311240065
*/
package main

import "fmt"

func kelipatanIteratif() {
    var bilangan, hasil int = 0, 0
    fmt.Println("Masukkan bilangan (negatif untuk berhenti): ")
    for bilangan >= 0 {
        fmt.Scan(&bilangan)
        if bilangan%4 == 0 {
            hasil += bilangan
        }
    }
    fmt.Println("Iteratif--Jumlah bilangan kelipatan 4:", hasil)
}

func kelipatanRekursif(a int) int {
    var bil int
    fmt.Scan(&bil)
    if bil < 0 {
        return a
    }
    if bil >= 0 && bil%4 == 0 {
        a += bil
    }
    return kelipatanRekursif(a)
}

func main() {
    kelipatanIteratif()
    fmt.Println("Masukkan bilangan (negatif untuk berhenti): ")
    hasil := kelipatanRekursif(0)
    fmt.Print("Rekursif--Jumlah bilangan kelipatan 4: ", hasil)
}
```

- **Screenshoot Hasil Pembahasan:**

```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2
Collage\Semester 2\Algoritma dan Pemrograman
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Iteratif--Jumlah bilangan kelipatan 4: 4
Masukkan bilangan (negatif untuk berhenti):
2 3 4 5 6 -1
Rekursif--Jumlah bilangan kelipatan 4: 4
```

- **Penjelasan:**

Program ini bertujuan untuk mengecek dan mencari berapa banyak angka yang pengguna masukkan termasuk bilangan kelipatan 4. Program ini menggunakan pendekatan iteratif dan rekursif secara terpisah namun dengan tujuan yang sama.

Program ini menggunakan prosedur dengan nama kelipatanIteratif. Fungsi prosedur ini menggunakan metode iteratif untuk mencari dan menghitung berapa banyak bilangan yang termasuk kelipatan 4. Prosedur ini mendeklarasikan variabel bilangan, hasil tipe integer dan memberikan hasil untuk variabel bilangan sama dengan 0 dan variabel hasil sama dengan 0. Lalu akan mencetak kalimat dan setelahnya program akan masuk ke perulangan. Di perulangan menggunakan kondisi bilangan lebih dari sama dengan 0 akan meminta angka dari pengguna dan di scan sebagai variabel bilangan, lalu masuk ke percabangan dengan kondisi bilangan habis dibagi 4 maka akan menambahkan variabel hasil dengan bilangan. Keluar dari perulangan akan mencetak variabel hasil.

Program ini menggunakan prosedur dengan nama kelipatanRekursif. Prosedur ini menggunakan parameter a tipe integer. Fungsi prosedur ini menggunakan metode rekursif untuk mencari dan menghitung berapa banyak bilangan yang termasuk kelipatan 4. Prosedur ini mendeklarasikan variabel bil tipe integer. Lalu akan melakukan scan dari pengguna dengan variabel bil. Masuk ke percabangan dengan kondisi bil kurang dari 0 maka akan mengembalikan nilai, percabangan ini sebagai tempat atau batasan dari prosedur rekursifnya. Setelahnya akan masuk ke percabangan dengan kondisi bil lebih dari sama dengan 0 dan bil habis dibagi 4 maka akan menambahkan variabel a dengan variabel bil. Keluar dari

percabangan kedua, selanjutnya prosedur ini akan memanggil dirinya sendiri dengan `kelipatanRekursif(a)`.

Masuk ke program utama, dengan langsung memanggil prosedur `kelipatanIteratif` lalu akan tercetak kalimat bermaksud untuk menginputkan angka yang akan dicek. Setelah itu mendeklarasikan sekaligus memberikan nilai variabel hasil dengan prosedur `kelipatanRekursif(0)`, setelahnya akan mencetak kalimat sekaligus hasil dari prosedur `kelipatanRekursif`.

III. TIPE SOAL C

1. Digit

- Source Code:

```
/*  
Nama: Dimas Ramadhani  
NIM: 103112400065  
*/  
  
package main  
  
import "fmt"  
  
func main() {  
    var bil int  
    fmt.Print("Masukkan bilangan bulat positif (>10): ")  
    fmt.Scan(&bil)  
    membagiDigit(bil)  
}  
  
func membagiDigit(a int) {  
    var bil1, bil2, pembagi int  
    pembagi = 1  
    tempBil := a  
    for a >= 10 {  
        pembagi *= 10  
        a = a / pembagi  
    }  
    if tempBil >= 100 {  
        pembagi /= 10  
    }  
    bil1 = tempBil / pembagi  
    bil2 = tempBil % pembagi  
    fmt.Println("Bilangan 1:", bil1)  
    fmt.Println("Bilangan 2:", bil2)  
    fmt.Println("Hasil penjumlahan:", bil1+bil2)  
}
```

- **Screenshot Hasil Pembahasan:**

```
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\A
Collage\Semester 2\Algoritma dan Pemrograman 2\
Masukkan bilangan bulat positif (>10): 15
Bilangan 1: 1
Bilangan 2: 5
Hasil penjumlahan: 6
PS C:\Users\Dimas\OneDrive\Collage\Semester 2\A
Collage\Semester 2\Algoritma dan Pemrograman 2\
Masukkan bilangan bulat positif (>10): 12345
Bilangan 1: 123
Bilangan 2: 45
Hasil penjumlahan: 168
```

- **Penjelasan:**

Program ini bertujuan untuk memotong suatu bilangan bulat positif. Program menggunakan iteratif untuk mencari dan memotong bilangan sehingga terbagi menjadi dua dan menjumlahkan kedua bilangan tersebut.

Pada prosedur utama, mendeklarasikan variabel *bil* tipe *int*. Lalu mencetak kalimat setelah itu pengguna akan memasukkan nilai bilangan lebih dari 10 dan di scan sebagai variabel *bil*. Setelah itu prosedur utama akan memanggil prosedur *membagiDigit*.

Program ini menggunakan prosedur dengan nama *membagiDigit* bertipe *int*, dengan parameter *a* *int*, dengan *a* merupakan angka lebih dari 10 atau angka berdigit 2 atau lebih. Prosedur ini berfungsi mencari dan membagi suatu angka yang diinputkan menjadi dua bagian lalu menjumlahkan 2 angka tersebut. Pertama, prosedur mendeklarasikan variabel *bil1*, *bil2*, *pembagi* tipe *integer*. Lalu memberikan nilai variabel *pembagi* = 1 dan mendeklarasikan variabel *tempBil* tipe *integer* sekaligus memberikan nilai sama dengan *a*. Masuk ke perulangan dengan kondisi *a* lebih dari sama dengan 10 maka akan mengalikan variabel *pembagi* dengan 10 lalu membagi variabel *a* dengan 10. Keluar perulangan akan masuk ke percabangan dengan kondisi *tempBil* lebih dari sama dengan 100 maka akan membagi variabel *pembagi* dengan 10. Keluar percabangan, melakukan pemberian nilai untuk variabel *bil1* =

$\frac{tempBil}{pembagi}$, lalu variabel $bil2 = tempBil \% pembagi$. Setelah itu akan tercetak variabel bil1 beserta kalimatnya, variabel bil2 beserta kalimatnya, dan terakhir penjumlahan dari bil1 dan bil2 beserta kalimatnya.

2. Bukber IF

- Source Code:

```
/*
Nama: Dimas Ramadhani
NIM: 103112400065
*/

package main

import "fmt"

func main() {
    var N, kode, jumA, jumB, jumC int
    fmt.Print("Masukkan jumlah peserta: ")
    fmt.Scan(&N)
    for i := 1; i <= N; i++ {
        fmt.Printf("Masukkan nomor kartu peserta ke-%d: ", i)
        fmt.Scan(&kode)

        if bilanganSama(kode) {
            fmt.Println("Hadiah A")
            jumA++
        } else if bilanganBeda(kode) {
            fmt.Println("Hadiah B")
            jumB++
        } else {
            fmt.Println("Hadiah C")
            jumC++
        }
    }
    fmt.Println()
    fmt.Println("Jumlah yang memperoleh Hadiah A:", jumA)
    fmt.Println("Jumlah yang memperoleh Hadiah B:", jumB)
    fmt.Println("Jumlah yang memperoleh Hadiah C:", jumC)
}

func bilanganSama(a int) bool {
    var b1, b2, b3 int
    b1 = a / 100
    b2 = (a / 10) % 10
```

```

b3 = a % 10
if b1 == b2 && b2 == b3 && b3 == b1 {
    return true
}
return false
}

func bilanganBeda(a int) bool {
var b1, b2, b3 int
b1 = a / 100
b2 = (a / 10) % 10
b3 = a % 10
if b1 != b2 && b2 != b3 && b3 != b1 {
    return true
}
return false
}

```

- **Screenshot Hasil Pembahasan:**

```

PS C:\Users\Dimas\OneDrive\Collage\Semester
Collage\Semester 2\Algoritma dan Pemrogram
Masukkan jumlah peserta: 3
Masukkan nomor kartu peserta ke-1: 333
Hadiah A
Masukkan nomor kartu peserta ke-2: 123
Hadiah B
Masukkan nomor kartu peserta ke-3: 898
Hadiah C

Jumlah yang memperoleh Hadiah A: 1
Jumlah yang memperoleh Hadiah B: 1
Jumlah yang memperoleh Hadiah C: 1

```

- **Penjelasan:**

Program ini bertujuan untuk memberikan hadiah untuk peserta berdasarkan kode voucher yang dimasukkan oleh pengguna. Program ini menggunakan iteratif pada prosedur bilanganSama, bilanganBeda.

Pada prosedur utama, mendeklarasikan variabel N, kode, jumA, jumB, jumC tipe integer. Lalu akan mencetak kalimat dan setelahnya pengguna menginputkan bilangan sebagai jumlah peserta pada variabel N. Setelah itu masuk ke perulangan dengan iterasi awal sama dengan 1, kondisi untuk iterasi kurang dari sama dengan N, setiap perulangan nilai iterasi akan ditambahkan dengan 1. Di dalam perulangan akan mencetak kalimat lalu variabel i dan setelahnya pengguna memasukkan angka sebagai variabel kode. Setelah itu akan masuk ke percabangan dengan kondisi prosedur bilanganSama maka akan mencetak kalimat “Hadiah A” lalu menambahkan nilai variabel jumA dengan 1. Selain itu jika prosedur bilanganBeda maka akan mencetak “Hadiah B” dan menambahkan variabel jumB dengan 1. Selain itu akan mencetak kalimat “Hadiah C” dan menambahkan nilai variabel jumC dengan satu. Di luar dari perulangan akan tercetak kalimat dan jumA, lalu kalimat dan jumB, lalu kalimat dan jumC.

Program ini menggunakan prosedur dengan nama bilanganSama, dengan parameter a int, berfungsi untuk mengecek apakah nomor kartu memiliki angka yang sama semua atau tidak. Prosedur ini menggunakan pendekatan iteratif untuk mengecek nomor kartu peserta. Prosedur ini mendeklarasikan variabel b1, b2, b3 tipe int. Lalu memberikan nilai variabel $b1 = \frac{a}{100}$, lalu variabel $b2 = \frac{a}{10} \% 10$, dan $b3 = a \% 10$. Lalu masuk ke percabangan dengan kondisi b1 sama dengan b2 dan b2 sama dengan b3 dan b3 sama dengan b1 maka akan mengembalikan nilai true. Selain itu akan mengembalikan nilai false.

Program ini menggunakan prosedur dengan nama bilanganBeda, dengan parameter a int, berfungsi untuk mengecek apakah nomor kartu memiliki angka yang berbeda semua atau tidak. Prosedur ini menggunakan pendekatan iteratif untuk mengecek nomor kartu peserta. Prosedur ini mendeklarasikan variabel b1, b2, b3 tipe int. Lalu memberikan nilai variabel $b1 = \frac{a}{100}$, lalu variabel $b2 = \frac{a}{10} \% 10$, dan $b3 = a \% 10$. Lalu masuk ke percabangan dengan kondisi b1 tidak sama dengan b2 dan b2 tidak sama dengan b3 dan b3 tidak sama dengan b1 maka akan mengembalikan nilai true. Selain itu akan mengembalikan nilai false.

3. Perkalian dengan Cara Penjumlahan

- **Source Code:**

```
/*  
Nama: Dimas Ramadhani  
NIM: 103112400065  
*/  
  
package main  
  
import "fmt"  
  
func perkalianIteratif(n, m int) int {  
    var hasil int  
    for i := n; i >= 1; i-- {  
        hasil += m  
    }  
    return hasil  
}  
  
func perkalianRekursif(n, m, hasil int) int {  
    if n == 0 {  
        return hasil  
    }  
    return perkalianRekursif(n-1, m, hasil+m)  
}  
  
func main() {  
    var n, m int  
    fmt.Print("Masukkan bilangan n: ")  
    fmt.Scan(&n)  
    fmt.Print("Masukkan bilangan m: ")  
    fmt.Scan(&m)  
    hasilIteratif := perkalianIteratif(n, m)  
    hasilRekursif := perkalianRekursif(n, m, 0)  
    fmt.Printf("Iteratif---Hasil dari %d x %d =%d\n", n, m, hasilIteratif)  
    fmt.Printf("Rekursif---Hasil dari %d x %d =%d\n", n, m, hasilRekursif)  
}
```

- **Screenshoot Hasil Pembahasan:**

```
PS C:\Users\Dimas\OneDrive\Collage\Semest
Collage\Semester 2\Algoritma dan Pemrogra
Masukkan bilangan n: 5
Masukkan bilangan m: 6
Iteratif---Hasil dari 5 x 6 =30
Rekursif---Hasil dari 5 x 6 =30
```

- **Penjelasan:**

Program ini bertujuan untuk mengalikan suatu bilangan namun tanpa struktur perkalian, menggunakan penjumlahan. Program ini menggunakan iteratif dan rekrusif secara terpisah namun dengan fungsi yang sama.

Program ini menggunakan prosedur dengan nama perkalianIteratif, dengan parameter variabel n dan m tipe int, untuk variabel n adalah angka yang berfungsi sebagai penjumlahan berulang untuk variabel m. Fungsi prosedur ini menggunakan metode iteratif untuk menjumlahkan variabel m sebanyak n kali. Prosedur ini mendeklarasikan variabel hasil tipe integer. Lalu masuk ke perulangan dengan iterasi awal sama dengan n, kondisi iterasi lebih dari sama dengan 1, setiap perulangan variabel iterasi akan dikurangi dengan 1. Di dalam perulangan akan menjumlahkan hasil dengan variabel m. Di luar dari perulangan akan mengembalikan variabel hasil.

Program ini menggunakan prosedur dengan nama perkalianRekursif, dengan parameter variabel n, m dan hasil tipe int, untuk variabel n adalah angka yang berfungsi sebagai penjumlahan berulang untuk variabel m, sedangkan hasil adalah sebagai angka akhir total dari penjumlahannya. Fungsi prosedur ini menggunakan metode rekursif untuk menjumlahkan variabel m sebanyak n kali. Prosedur ini menjalankan percabangan dengan kondisi n sama dengan 0 maka akan mengembalikan nilai hasil, disini sebagai struktur berhentinya prosedur rekrusif. Setelah percabangan akan mengembalikan nilai atau memanggil dirinya sendiri yaitu perkalianRekursif(n-1, m, hasil+m)

Masuk ke program utama, dengan mendeklarasikan variabel n dan m tipe int, variabel n untuk angka pengali variabel m. Lalu program

akan mencetak kalimat dan memasukkan angka dari pengguna di scan sebagai variabel n , lalu akan tercetak kalimat lagi dan memasukkan angka dari pengguna di scan sebagai variabel m . Lalu melakukan pendeklarasian dan pemberian nilai $hasilIteratif = perkalianIteratif(n, m)$, dan variabel $hasilRekursif(n, m, 0)$. Setelah itu akan mencetak kalimat dan $hasilIteratif$ tadi lalu mencetak kalimat dan $hasilRekursif$ tadi.