

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 6
“Array”



DISUSUN OLEH:
RAIHAN ADI ARBA
103112400071
S1 IF-12-01
DOSEN:
Dimas Fanny Hebrasianto Permadi, S.ST., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI

Modul ini membahas tentang tipe data buatan dan struktur data array dalam pemrograman Go. Tipe data buatan terdiri dari dua jenis, yaitu alias yang berfungsi untuk menyederhanakan penamaan tipe data yang sudah ada, serta struct yang digunakan untuk mengelompokkan berbagai data terkait menjadi satu unit. Selain itu, modul ini memperkenalkan array sebagai struktur data statis dengan ukuran tetap, sementara slice dan map ditawarkan sebagai solusi yang lebih dinamis untuk menangani data yang bersifat fleksibel dan asosiatif.

Beberapa contoh aplikasi yang dibahas meliputi analisis geometri untuk menentukan posisi suatu titik terhadap lingkaran, pemeriksaan teks palindrom, serta pengelolaan data seperti statistik pertandingan olahraga atau nilai siswa. Penekanan modul ini terletak pada pemilihan struktur data yang tepat agar penyelesaian masalah dapat dilakukan secara efisien dan terorganisir. Dengan memahami materi ini, pembaca diharapkan mampu mengimplementasikan tipe data buatan, array, slice, dan map dalam pengembangan program menggunakan bahasa Go.

A. GUIDED

1. Source code :

```
package main

import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name string
    Price float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date time.Time
    Items []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
    r.TotalAmount = total
}

func (r *Receipt) PrintReceipt() {
    fmt.Println("=====")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah",
    "Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name,
        item.Price, item.Quantity, itemTotal)
    }

    fmt.Println("=====")
    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
```

```

    fmt.Println("=====")
    fmt.Println("Terima kasih telah berbelanja!")
}

func main() {
    receipt := Receipt{
        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
        Date:      time.Now(),
        Items: []Item{
            {Name: "Beras", Price: 12000, Quantity: 5},
            {Name: "Gula", Price: 15000, Quantity: 2},
            {Name: "Minyak", Price: 20000, Quantity: 1},
            {Name: "Telur", Price: 2000, Quantity: 10},
        },
    }

    receipt.CalculateTotal()
    receipt.PrintReceipt()
}

```

Output :

```

raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run
"/Users/raihan/Documents/GitHub/waffer/Laprak-Mod
u-6/103112400071_MODUL6/tempCodeRunnerFile.go"

```

```

=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 14-06-2025 10:09
=====

```

Item	Harga	Jumlah	Total
Beras	Rp12000.00	5	Rp60000.00
Gula	Rp15000.00	2	Rp30000.00
Minyak	Rp20000.00	1	Rp20000.00
Telur	Rp2000.00	10	Rp20000.00

```

=====
Total Belanja:                Rp130000.00
=====

```

Terima kasih telah berbelanja!

Deskripsi :

Aplikasi ini memungkinkan pengguna untuk menyimpan informasi toko, daftar barang belanjaan beserta harga dan jumlahnya, serta secara otomatis menghitung total belanja. Struk yang dihasilkan menampilkan informasi transaksi seperti nama toko, tanggal

pembelian, detail item (termasuk harga per unit, jumlah, dan subtotal), serta total harga yang harus dibayar. Program menggunakan struct Item untuk merepresentasikan barang belanjaan dan struct Receipt untuk menyimpan data struk, dilengkapi dengan method CalculateTotal() untuk menghitung total belanja dan PrintReceipt() untuk menampilkan struk dalam format yang rapi dan mudah dibaca. Contoh output menunjukkan struk pembelian sembako dengan empat item berbeda yang total harganya dihitung secara otomatis.

2. Source code :

```
package main

import "fmt"

func main() {
    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa")
    fmt.Println("=====")

    for i, nilai := range nilaiMahasiswa {
        fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
    }

    var total int
    for _, nilai := range nilaiMahasiswa {
        total += nilai
    }
    rataRata := float64(total) / float64(len(nilaiMahasiswa))

    fmt.Println("=====")
    fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

    // Mencari nilai tertinggi dan terendah
    tertinggi := nilaiMahasiswa[0]
    terendah := nilaiMahasiswa[0]

    for _, nilai := range nilaiMahasiswa {
        if nilai > tertinggi {
            tertinggi = nilai
        }
        if nilai < terendah {
            terendah = nilai
        }
    }

    fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
    fmt.Printf("Nilai terendah: %d\n", terendah)

    // Contoh array 2 dimensi
```

```

fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1,
nilai[0], nilai[1])
}
}

```

Output :

```

raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run "/Users/raihan/Documents/GitHub/waffer/Laprak-Mod
u-6/103112400071_MODUL6/103112400071_Guided2.go"
Data Nilai Mahasiswa
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95

```

Deskripsi :

Aplikasi ini bekerja dengan dua jenis array: array satu dimensi untuk menyimpan nilai individual mahasiswa dan array dua dimensi untuk mencatat nilai ujian dari beberapa mata kuliah. Pada array satu dimensi, program menampilkan seluruh nilai mahasiswa, kemudian menghitung statistik penting seperti rata-rata nilai (87.20), nilai tertinggi (95), dan nilai terendah (78). Bagian kedua program mendemonstrasikan penggunaan array dua dimensi untuk menyimpan nilai ujian tiga mahasiswa dalam dua mata kuliah berbeda (Matematika dan Bahasa), kemudian menampilkannya dalam format yang terstruktur. Program ini secara efektif menunjukkan operasi dasar pengolahan array di Go, termasuk iterasi, perhitungan, dan pencarian nilai ekstrem, yang dapat menjadi fondasi untuk pengembangan sistem pengolahan data akademik yang lebih kompleks. Output yang dihasilkan disajikan secara rapi dan informatif, memudahkan pembacaan dan analisis data nilai.

B. UNGUIDED

1. Latihan 1

Source Code:

```
// Raihan Adi Arba
// 103112400071

package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    pusatlingkaran titik
    radius          int
}

func jarak(p, q titik) float64 {
    cx := p.x - q.x
    cy := p.y - q.y
    return math.Sqrt(float64(cx*cx + cy*cy))
}

func didalam(c lingkaran, p titik) bool {
    jarakkepusat := jarak(c.pusatlingkaran, p)
    return jarakkepusat <= float64(c.radius)
}

func main() {
    var (
        lingkaran1, lingkaran2    lingkaran
        p                          titik
        dlmlingkaran1, dlmlingkaran2 bool
    )

    fmt.Scan(&lingkaran1.pusatlingkaran.x, &lingkaran1.pusatlingkaran.y,
    &lingkaran1.radius)
    fmt.Scan(&lingkaran2.pusatlingkaran.x, &lingkaran2.pusatlingkaran.y,
    &lingkaran2.radius)
    fmt.Scan(&p.x, &p.y)

    dlmlingkaran1 = didalam(lingkaran1, p)
    dlmlingkaran2 = didalam(lingkaran2, p)
}
```

```

switch {
case dlmlingkar1 && dlmlingkar2:
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
case dlmlingkar1:
    fmt.Println("Titik di dalam lingkaran 1")
case dlmlingkar2:
    fmt.Println("Titik di dalam lingkaran 2")
default:
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output :

```

raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run "/Users/raihan/Documents/GitHub/waffer/Laprak-Modu-6/103112400071_MODUL6/103112400071_Unguided1.go"
1 1 5
2 2 4
7 8
Titik di luar lingkaran 1 dan 2
raihan@Raihans-MacBook-Pro Laprak-Modu-6 %

```

Penjelasan Program:

Program ini merupakan implementasi sederhana dalam bahasa Go untuk menentukan posisi suatu titik terhadap dua lingkaran. Dengan menggunakan struktur data titik dan lingkaran, program menghitung jarak antara titik pusat lingkaran dengan titik yang dimasukkan, kemudian memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran berdasarkan perbandingan jarak dengan radius. Hasilnya ditampilkan dalam bentuk pesan yang sesuai, seperti "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2". Program ini berguna untuk analisis geometri dasar dalam sistem koordinat dua dimensi.

2. Latihan 2

Source Code:

```

// Raihan Adi Arba
// 103112400071

package main

import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n)
}

```



```

arr := make([]int, n)

fmt.Println("Masukkan elemen array:")
for i := 0; i < n; i++ {
    fmt.Printf("Elemen ke-%d: ", i+1)
    fmt.Scan(&arr[i])
}

for {
    fmt.Println("\n=====")
    fmt.Println("    SELAMAT DATANG DI MENU ARRAY    ")
    fmt.Println("=====")
    fmt.Println("1. TAMPILKAN SEMUA ARRAY")
    fmt.Println("2. TAMPILKAN SEMUA ARRAY GENAP")
    fmt.Println("3. TAMPILKAN SEMUA ARRAY GANJIL")
    fmt.Println("4. TAMPILKAN ARRAY DENGAN INDEKS KELIPATAN X")
    fmt.Println("5. HAPUS ELEMEN PADA INDEKS TERTENTU")
    fmt.Println("6. HITUNG RATA-RATA ARRAY")
    fmt.Println("7. HITUNG STANDAR DEVIASI ARRAY")
    fmt.Println("8. HITUNG FREKUENSI BILANGAN TERTENTU")
    fmt.Println("9. KELUAR")
    fmt.Print("Pilihan Anda: ")

    var choice int
    fmt.Scan(&choice)

    if choice == 1 {
        displayAll(arr)
    } else if choice == 2 {
        displayEvenIndex(arr)
    } else if choice == 3 {
        displayOddIndex(arr)
    } else if choice == 4 {
        displayMultiples(arr)
    } else if choice == 5 {
        arr = deleteElement(arr)
    } else if choice == 6 {
        calculateAverage(arr)
    } else if choice == 7 {
        calculateStdDev(arr)
    } else if choice == 8 {
        calculateFrequency(arr)
    } else if choice == 9 {
        fmt.Println("Keluar dari program.")
        return
    } else {
        fmt.Println("Pilihan tidak valid!")
    }
}
}

```

```

func displayAll(arr []int) {
    fmt.Println("\nIsi array:")
    for i, val := range arr {
        fmt.Printf("Indeks %d: %d\n", i, val)
    }
}

func displayEvenIndex(arr []int) {
    fmt.Println("\nElemen dengan indeks genap:")
    for i := 0; i < len(arr); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, arr[i])
    }
}

func displayOddIndex(arr []int) {
    fmt.Println("\nElemen dengan indeks ganjil:")
    for i := 1; i < len(arr); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, arr[i])
    }
}

func displayMultiples(arr []int) {
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)

    fmt.Printf("\nElemen dengan indeks kelipatan %d:\n", x)
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Printf("Indeks %d: %d\n", i, arr[i])
        }
    }
}

func deleteElement(arr []int) []int {
    var index int
    fmt.Print("Masukkan indeks yang akan dihapus: ")
    fmt.Scan(&index)

    if index < 0 || index >= len(arr) {
        fmt.Println("Indeks tidak valid!")
        return arr
    }

    newArr := append(arr[:index], arr[index+1:]...)

    fmt.Println("\nArray setelah penghapusan:")
    for i, val := range newArr {
        fmt.Printf("Indeks %d: %d\n", i, val)
    }
}

```

```

        return newArr
    }

func calculateAverage(arr []int) {
    if len(arr) == 0 {
        fmt.Println("Array kosong")
        return
    }

    sum := 0
    for _, val := range arr {
        sum += val
    }
    average := float64(sum) / float64(len(arr))
    fmt.Printf("\nRata-rata: %.2f\n", average)
}

func calculateStdDev(arr []int) {
    if len(arr) == 0 {
        fmt.Println("Array kosong")
        return
    }

    sum := 0
    for _, val := range arr {
        sum += val
    }
    mean := float64(sum) / float64(len(arr))

    variance := 0.0
    for _, val := range arr {
        variance += math.Pow(float64(val)-mean, 2)
    }
    variance /= float64(len(arr))

    stdDev := math.Sqrt(variance)
    fmt.Printf("\nStandar deviasi: %.2f\n", stdDev)
}

func calculateFrequency(arr []int) {
    var target int
    fmt.Print("Masukkan bilangan yang akan dihitung frekuensinya: ")
    fmt.Scan(&target)

    count := 0
    for _, val := range arr {
        if val == target {
            count++
        }
    }
}

```

```
    fmt.Printf("\nFrekuensi bilangan %d: %d\n", target, count)
}
```

Output:

```
raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run "/Users/raihan/Documents/GitHub/waffer/Laprak-Modu-6/103112400071_MODUL6/103112400071_unguided2.go"
Masukkan jumlah elemen array: 6
Masukkan elemen array:
Elemen ke-1: 2
Elemen ke-2: 1
Elemen ke-3: 3
Elemen ke-4: 2
Elemen ke-5: 4
Elemen ke-6: 7

=====
                SELAMAT DATANG DI MENU ARRAY
=====

1. TAMPILKAN SEMUA ARRAY
2. TAMPILKAN SEMUA ARRAY GENAP
3. TAMPILKAN SEMUA ARRAY GANJIL
4. TAMPILKAN ARRAY DENGAN INDEKS KELIPATAN X
5. HAPUS ELEMEN PADA INDEKS TERTENTU
6. HITUNG RATA-RATA ARRAY
7. HITUNG STANDAR DEVIASI ARRAY
8. HITUNG FREKUENSI BILANGAN TERTENTU
9. KELUAR
Pilihan Anda: 1

Isi array:
Indeks 0: 2
Indeks 1: 1
Indeks 2: 3
Indeks 3: 2
Indeks 4: 4
Indeks 5: 7
```

Pilihan Anda: 2

Elemen dengan indeks genap:

Indeks 0: 2

Indeks 2: 3

Indeks 4: 4

Pilihan Anda: 3

Elemen dengan indeks ganjil:

Indeks 1: 1

Indeks 3: 2

Indeks 5: 7

Pilihan Anda: 4

Masukkan nilai x: 3

Elemen dengan indeks kelipatan 3:

Indeks 0: 2

Indeks 3: 2

Pilihan Anda: 6

Rata-rata: 3.17

Pilihan Anda: 7

Standar deviasi: 1.95

Pilihan Anda: 8

Masukkan bilangan yang akan dihitung frekuensinya: 4

Frekuensi bilangan 4: 1

Deskripsi Program:

Aplikasi ini memungkinkan pengguna untuk melakukan berbagai operasi pada array integer melalui antarmuka menu yang user-friendly. Setelah pengguna memasukkan jumlah elemen dan nilai-nilai array, program menampilkan menu utama dengan sembilan pilihan operasi, termasuk menampilkan seluruh array, menampilkan elemen berdasarkan indeks (genap, ganjil, atau kelipatan tertentu), menghapus elemen, serta melakukan perhitungan statistik seperti rata-rata, standar deviasi, dan frekuensi kemunculan bilangan.

Program ini dirancang dengan struktur yang modular, di mana setiap fungsi operasi diimplementasikan secara terpisah, seperti `displayAll()` untuk menampilkan array, `calculateAverage()` untuk menghitung rata-rata, dan `deleteElement()` untuk menghapus elemen tertentu. Contoh eksekusi program menunjukkan kemampuannya dalam mengolah data array, seperti menghitung rata-rata 3.17 dan standar deviasi 1.95 dari array [2, 1, 3, 2, 4, 7], serta menentukan frekuensi kemunculan bilangan tertentu (misalnya bilangan 4 muncul 1 kali).

3. Latihan 3

Source Code:

```
// Raihan Adi Arba
// 103112400071
package main

import "fmt"

func main() {
    var klub [2]string
    var pemenang [100]string
    var skorA, skorB int
    var i int

    fmt.Print("Klub A : ")
    fmt.Scan(&klub[0])

    fmt.Print("Klub B : ")
    fmt.Scan(&klub[1])

    for {
        fmt.Printf("Pertandingan %d : ", i+1)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }

        if skorA > skorB {
            pemenang[i] = klub[0]
        } else if skorB > skorA {
            pemenang[i] = klub[1]
        } else {
            pemenang[i] = "Draw"
        }
        i++
    }

    fmt.Println()
    for j := 0; j < i; j++ {
```

```

        fmt.Printf("Hasil %d : %s\n", j+1, pemenang[j])
    }
    fmt.Println("Pertandingan selesai")
}

```

Output:

```

● raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run "/Users/raihan/D
ocuments/GitHub/waffer/Laprak-Modu-6/103112400071_MODUL6/103112400
071_unguided3.go"
Klub A : 11
Klub B : 22
Pertandingan 1 : 2
1
Pertandingan 2 : 3 4
Pertandingan 3 : 9 6
Pertandingan 4 : 3 9
Pertandingan 5 : 4 7
Pertandingan 6 : 3 6
Pertandingan 7 : -3 0

Hasil 1 : 11
Hasil 2 : 22
Hasil 3 : 11
Hasil 4 : 22
Hasil 5 : 22
Hasil 6 : 22
Pertandingan selesai
○ raihan@Raihans-MacBook-Pro Laprak-Modu-6 % █

```

Deskripsi Program:

Aplikasi ini dirancang untuk memudahkan pencatatan serangkaian pertandingan antara dua klub sepak bola. Pengguna pertama-tama diminta untuk memasukkan nama kedua klub yang akan bertanding, kemudian program akan memproses input skor dari setiap pertandingan secara berulang.

Sistem bekerja dengan membandingkan skor kedua tim untuk setiap pertandingan, kemudian secara otomatis menentukan pemenangnya (klub pertama, klub kedua, atau hasil seri). Program menggunakan array dengan kapasitas 100 elemen untuk menyimpan hasil pertandingan, sehingga mampu mencatat hingga 100 pertandingan. Proses input akan berhenti secara otomatis ketika pengguna memasukkan skor negatif, yang kemudian program akan menampilkan laporan lengkap semua hasil pertandingan yang telah dicatat.

4. Latihan 4

Source code :

```

// Raihan Adi Arba
// 103112400071

package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {

    var input rune
    *n = 0
    fmt.Println("Masukkan karakter [akhirki dengan . ]:")

    for {
        fmt.Scanf("%c", &input)

        if input == '\n' {
            continue
        }
        if input == '.' || *n >= NMAX {
            break
        }
        t[*n] = input
        (*n)++
    }
}

func cetakArray(t tabel, n int) {

    fmt.Print("Isi array: ")
    for i := 0; i < n; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Println()
}

func balikanArray(t *tabel, n int) {

    for i := 0; i < n/2; i++ {
        t[i], t[n-1-i] = t[n-1-i], t[i]
    }
}

func isPalindrom(t tabel, n int) bool {

    for i := 0; i < n/2; i++ {
        if t[i] != t[n-1-i] {

```



```

        return false
    }
}
return true
}

func main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

    fmt.Println("\nArray sebelum dibalik:")
    cetakArray(tab, m)

    if isPalindrom(tab, m) {
        fmt.Println("Array ini palindrom")
    } else {
        fmt.Println("Array bukan palindrom")
    }

    balikanArray(&tab, m)

    fmt.Println("\nArray setelah dibalik:")
    cetakArray(tab, m)

    if isPalindrom(tab, m) {
        fmt.Println("Array ini palindrom")
    } else {
        fmt.Println("Array bukan palindrom")
    }
}

```

Output :

```

raihan@Raihans-MacBook-Pro Laprak-Modu-6 % go run "/Users/raihan/Documents/GitHub/waffer/Laprak-Modu-6/103112400071_MODUL6/103112400071_unguided4.go"
Masukkan karakter [akhirki dengan . ]:
mancing.

Array sebelum dibalik:
Isi array: m a n c i n g
Array bukan palindrom

Array setelah dibalik:
Isi array: g n i c n a m
Array bukan palindrom

```

Deskripsi :

Program ini merupakan aplikasi manipulasi array karakter yang mengimplementasikan operasi dasar seperti input data, pencetakan array, pembalikan urutan, dan pengecekan palindrom. Dengan kapasitas maksimum 127 elemen, program menerima input karakter dari pengguna hingga menemukan tanda titik (.) sebagai penanda akhir. Fitur utamanya mencakup kemampuan memeriksa apakah susunan karakter membentuk palindrom - suatu pola yang sama ketika dibaca maju maupun mundur - serta fungsi untuk membalik urutan array. Seperti terlihat pada contoh eksekusi dengan input "mancing", program berhasil memproses array, melakukan pembalikan, dan memberikan verifikasi status palindrom secara akurat.

DAFTAR PUSTAKA

Prayogo, N. A. (2021). *Dasar Pemrograman Go. Ebook*