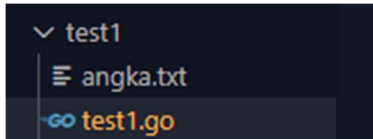


Link Github : <https://github.com/mnasruls/Upscale>

1. Code :



```
package main

import (
    "io/ioutil"
    "log"
    "strconv"
    "strings"
)

func main() {
    // read file
    numbers, err := ReadFile("test1/angka.txt")
    if err != nil {
        log.Println("Error read file : ", err)
        panic(err)
    }

    log.Println("Total angka pada file : ", len(numbers))

    // sum numbers
    sum, err := SumNum(numbers)
    if err != nil {
        log.Println("Error sum number : ", err)
        panic(err)
    }

    log.Println("Jumlah semua angka : ", sum)
}

func ReadFile(filename string) ([]string, error) {

    file, err := ioutil.ReadFile(filename)
    if err != nil {
        return nil, err
    }
}
```

```

    dataFile := string(file)

    numbers := strings.Fields(dataFile)

    return numbers, nil
}

func SumNum(numbers []string) (int, error) {

    var result int
    // loop to sum numbers
    for _, numS := range numbers {
        num, err := strconv.Atoi(numS)
        if err != nil {
            return 0, err
        }

        result += num
    }

    return result, nil
}

```

Result :

```

[New Branch] main -> main
mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test1/test1.go
2023/05/24 23:58:55 Total angka pada file : 5
2023/05/24 23:58:55 Jumlah semua angka : 15

```

2. Code :

```

package main

import (
    "bufio"
    "log"
    "os"
    "strings"
)

func main() {
    var inputText string
    log.Print("Masukkan kalimat/kata : ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()
}

```

```

    inputText = scan.Text()
    if CheckIfPolindrom(inputText) {
        log.Println("Kata/kalimat yang diinput :", inputText, "\nMerupakan
palindrom")
    } else {
        log.Println("Kata/kalimat yang diinput :", inputText, "\nBukan merupakan
palindrom")
    }
}

func CheckIfPolindrom(text string) bool {
    newText := strings.ToLower(text)

    for i := 0; i < len(newText)/2; i++ {
        if newText[i] != newText[len(newText)-(i+1)] {
            return false
        }
    }
    return true
}

```

Result:

```

● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test2/test2.go
2023/05/25 00:01:49 Masukkan kalimat/kata :
Kasur ini rusak
2023/05/25 00:02:03 Kata/kalimat yang diinput : Kasur ini rusak
Merupakan palindrom
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test2/test2.go
2023/05/25 00:02:07 Masukkan kalimat/kata :
Ibu ratna antar ubi
2023/05/25 00:02:27 Kata/kalimat yang diinput : Ibu ratna antar ubi
Merupakan palindrom
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test2/test2.go
2023/05/25 00:02:29 Masukkan kalimat/kata :
Malam ini aku terbangun
2023/05/25 00:02:37 Kata/kalimat yang diinput : Malam ini aku terbangun
Bukan merupakan palindrom
○ mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ 

```

3. Code :

```

package main

import (
    "encoding/json"
    "errors"
    "log"
    "net/http"
    "time"
)

type data struct {
    Id    int    `json:"id"`
    Title string `json:"title"`
    Body  string `json:"body"`
}

type datas []data

func main() {
    response, err := GetListData()
    if err != nil {
        log.Println("Error : ", err.Error())
        panic(err)
    }
    log.Println(JsonEncode(response))
}

func GetListData() (result datas, err error) {
    client := http.Client{
        Timeout: 3 * time.Second,
    }

    request, err := http.Get("https://jsonplaceholder.typicode.com/posts")
    if err != nil {
        return
    }
    response, err := client.Do(request.Request)
    if err != nil {
        return
    }

    defer response.Body.Close()

    if response.StatusCode >= 300 {
        err = errors.New("something error : " + response.Status)
    }
}

```

```

        return
    }

    err = json.NewDecoder(response.Body).Decode(&result)
    if err != nil {
        return
    }

    return
}

func JsonEncode(obj interface{}) string {
    json, _ := json.MarshalIndent(obj, "", " ")
    return string(json)
}

```

Resultst :

```

● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test3/test3.go
2023/05/25 00:04:47 [
    {
        "id": 1,
        "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
        "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
    },
    {
        "id": 2,
        "title": "qui est esse",
        "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
    },
    {
        "id": 3,
        "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
        "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmoles tiae porro eius odio et labore et velit aut"
    },
    {
        "id": 4,
        "title": "eum et est occaecati",
        "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
    },
    {
        "id": 5,
        "title": "nesciunt quas odio",
        "body": "repudiandae veniam quaerat sunt sed\nnalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nne st aut tenetur dolor neque"
    },
    {
        "id": 6,
        "title": "dolorem eum magni eos aperiam quia",
        "body": "ut aspernatur corporis harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab repreh enderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"
    },
    {
        "id": 7,
        "title": "magnam facilis autem",
        "body": "dolore placeat quibusdam ea quo vitae\nmagni quis enim qui quis quo nemo aut saepe\nquidem repellat excepturi ut quia\nsunt ut sequi eos ea sed quas"
    },
    {
        "id": 8,

```

4. Code :

```
package main

import (
    "fmt"
    "math/rand"
    "time"
)

func main() {
    numbers := []int{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

    for range numbers {

        rand.Seed(time.Now().UnixNano())
        n := len(numbers)
        if n <= 1 {
            return
        }

        for i := n - 1; i > 0; i-- {
            j := rand.Intn(i + 1)
            numbers[i], numbers[j] = numbers[j], numbers[i]
        }
    }

    for _, num := range numbers {
        fmt.Print(num, " ")
    }
    fmt.Println()
}
```

Result :

```

● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
10 9 6 3 2 1 5 7 4 8
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
9 1 7 4 5 10 6 8 3 2
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
3 8 10 5 9 1 4 7 2 6
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
4 10 9 2 5 8 6 7 1 3
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
6 2 7 10 1 3 5 9 8 4
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
5 6 4 2 1 3 9 10 8 7
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test4/test4.go
5 9 2 3 10 1 6 7 8 4
○ mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ █

```

5. Code :

```

package main

import (
    "bufio"
    "errors"
    "fmt"
    "log"
    "os"
    "strconv"
    "strings"
)

func main() {
    var time12 string
    log.Print("Masukkan waktu dalam format 12 jam (HH:MM:SS AM/PM): ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()

    time12 = scan.Text()

    time24, err := convertTime(time12)
    if err != nil {
        log.Println(err)
        return
    }

    log.Println("Waktu dalam format 24 jam : ", time24)
}

```

```

func convertTime(time12 string) (string, error) {
    partsTime := strings.Split(time12, " ")
    if len(partsTime) != 2 {
        return "", errors.New("Invalid input format")
    }

    timeStr := partsTime[0]
    meridiem := strings.ToUpper(partsTime[1])

    if meridiem != "AM" && meridiem != "PM" {
        return "", errors.New("Invalid input format")
    }

    timeParts := strings.Split(timeStr, ":")
    if len(timeParts) != 3 {
        return "", errors.New("Invalid input format")
    }

    hourStr := timeParts[0]

    minute, err := strconv.Atoi(timeParts[1])
    if err != nil {
        return "", errors.New("Invalid input format")
    }

    if minute >= 60 || minute < 0 {
        return "", errors.New("Invalid input format")
    }

    second, err := strconv.Atoi(timeParts[2])
    if err != nil {
        return "", errors.New("Invalid input format")
    }

    if second >= 60 || second < 0 {
        return "", errors.New("Invalid input format")
    }

    minuteStr := timeParts[1] + ":" + timeParts[2]

    hour, err := convertHour(hourStr, meridiem)
    if err != nil {
        return "", err
    }
}

```



```

    return fmt.Sprintf("%02d:%s", hour, minuteStr), nil
}

func convertHour(hourStr string, meridiem string) (int, error) {
    hour, err := strconv.Atoi(hourStr)
    if err != nil {
        return 0, errors.New("Invalid input format")
    }

    if hour < 1 || hour > 12 {
        return 0, errors.New("Invalid input format")
    }

    switch meridiem {
    case "PM":
        if hour != 12 {
            hour += 12
        }
    case "AM":
        if hour == 12 {
            hour = 0
        }
    }

    return hour, nil
}

```

Result :

```

● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test5/test5.go
2023/05/25 00:11:36 Masukkan waktu dalam format 12 jam (HH:MM:SS AM/PM):
12:14:09 AM
2023/05/25 00:11:48 Waktu dalam format 24 jam : 00:14:09
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test5/test5.go
2023/05/25 00:11:56 Masukkan waktu dalam format 12 jam (HH:MM:SS AM/PM):
13:12:11 AM
2023/05/25 00:12:08 Invalid input format
● mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test5/test5.go
2023/05/25 00:12:11 Masukkan waktu dalam format 12 jam (HH:MM:SS AM/PM):
01:12:60 PM
2023/05/25 00:12:33 Waktu dalam format 24 jam : 13:12:60
○ mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ █

```

6. Code :

```
package main

import (
    "bufio"
    "fmt"
    "log"
    "os"
    "sort"
    "strconv"
    "strings"
    "unicode"
)

func main() {
    FizzBuzz()           // bagian 1
    Palindrom()          // bagian 2
    Factorial()          // bagian 3
    BiggestSmallest()    // bagian 4
    LettersAndDigits()   // bagian 5
}

// bagian 1
func FizzBuzz() {
    var inputInt string
    log.Print("Masukkan angka : ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()

    inputInt = scan.Text()

    input, err := strconv.Atoi(inputInt)
    if err != nil {
        log.Println("format bilangan salah")
    }

    for i := 1; i <= input; i++ {
        switch {
        case i%3 == 0 && i%5 == 0:
            fmt.Print("FizzBuzz ")
        case i%3 == 0:
            fmt.Print("Fizz ")
        case i%5 == 0:
            fmt.Print("Buzz ")
        }
    }
}
```

```

        default:
            fmt.Print(i, " ")
        }
    }
    fmt.Println()
}

// bagian 2
func Palindrom() {
    var inputText string
    log.Print("Masukkan kalimat/kata : ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()

    inputText = scan.Text()
    newText := strings.ToLower(inputText)

    var isPalindrom = true
    for i := 0; i < len(newText)/2; i++ {
        if newText[i] != newText[len(newText)-(i+1)] {
            isPalindrom = false
            break
        }
    }

    if isPalindrom {
        log.Println("Kata/kalimat yang diinput :", inputText, "\nMerupakan palindrom")
    } else {
        log.Println("Kata/kalimat yang diinput :", inputText, "\nBukan merupakan palindrom")
    }
}

// bagian 3
func Factorial() {
    var num int
    var result = 1
    log.Print("Masukkan bilangan: ")
    fmt.Scanln(&num)

    for i := num; i >= 1; i-- {
        result *= i
    }
    log.Println("Faktorial dari", num, "adalah", result)
}

```

```

}

// bagian 4
func BiggestSmallest() {
    var inputSInt string
    log.Print("Masukkan angka - angka (gunakan spasi untuk memisahkan urutan
bilangan) : ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()

    inputSInt = scan.Text()

    inputInt := strings.Split(inputSInt, " ")

    var arrInt []int
    for _, value := range inputInt {
        num, err := strconv.Atoi(value)
        if err != nil {
            log.Println("format angka salah")
            panic(err)
        }

        arrInt = append(arrInt, num)
    }

    sort.Ints(arrInt)

    log.Println("Bilangan terbesar adalah : ", arrInt[len(arrInt)-1])
    log.Println("Bilangan terkecil adalah : ", arrInt[0])
}

// bagian 5
func LettersAndDigits() {
    var inputText string
    var letters, digits, else int
    log.Print("Masukkan kalimat/kata : ")
    scan := bufio.NewScanner(os.Stdin)
    scan.Scan()

    inputText = scan.Text()

    for _, value := range inputText {
        switch {
        case unicode.IsLetter(value):
            letters++

```

```

        case unicode.IsDigit(value):
            digits++
        default:
            elses++
    }
}

log.Println("Jumlah angka : ", digits)
log.Println("Jumlah huruf : ", letters)
log.Println("Jumlah lain-lain : ", elses)
}

```

Result :

```

mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$ go run test6/test6.go
2023/05/25 00:23:23 Masukkan angka :
100
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 FizzBuzz 16 17 Fizz 19 Buzz Fizz 22 23 Fizz Buzz 26 Fizz 28 29 FizzBuzz 31 32 Fizz 34 Bu
zz Fizz 37 38 Fizz Buzz 41 Fizz 43 44 FizzBuzz 46 47 Fizz 49 Buzz Fizz 52 53 Fizz Buzz 56 Fizz 58 59 FizzBuzz 61 62 Fizz 64 Buzz Fizz 67
68 Fizz Buzz 71 Fizz 73 74 FizzBuzz 76 77 Fizz 79 Buzz Fizz 82 83 Fizz Buzz 86 Fizz 88 89 FizzBuzz 91 92 Fizz 94 Buzz Fizz 97 98 Fizz Buz
z
2023/05/25 00:23:24 Masukkan kalimat/kata :
kasur ini rusak
2023/05/25 00:23:38 Kata/kalimat yang diinput : kasur ini rusak
Merupakan palindrom
2023/05/25 00:23:38 Masukkan bilangan:
12
2023/05/25 00:23:52 Faktorial dari 12 adalah 479001600
2023/05/25 00:23:52 Masukkan angka - angka (gunakan spasi untuk memisahkan urutan bilangan) :
10 12 45 2 17 47 134 23
2023/05/25 00:24:22 Bilangan terbesar adalah : 134
2023/05/25 00:24:22 Bilangan terkecil adalah : 2
2023/05/25 00:24:22 Masukkan kalimat/kata :
Hallo 12 Upscale 23
2023/05/25 00:25:13 Jumlah angka : 4
2023/05/25 00:25:13 Jumlah huruf : 12
2023/05/25 00:25:13 Jumlah lain-lain : 3
mnasrul@LAPTOP-ROI9GIMP:~/Playground/golang/Upscale$

```

7.
  - a. Untuk mengambil data dari API menggunakan Golang, Anda dapat menggunakan package **net/http** untuk membuat HTTP request ke API dan package **encoding/json** untuk menguraikan data JSON yang diterima.  
Berikut adalah contoh sederhana bagaimana Anda dapat mengambil data dari API yang mengembalikan data JSON:

```

package main

import (
    "encoding/json"
    "errors"
    "log"
    "net/http"

```

```

        "time"
    )

type data struct {
    Id      int    `json:"id"`
    Title   string  `json:"title"`
    Body    string  `json:"body"`
}

type datas []data

func main() {
    response, err := GetListData()
    if err != nil {
        log.Println("Error : ", err.Error())
        panic(err)
    }
    log.Println(JsonEncode(response))
}

func GetListData() (result datas, err error) {
    client := http.Client{
        Timeout: 3 * time.Second,
    }

    request, err := http.Get("https://jsonplaceholder.typicode.com/posts")
    if err != nil {
        return
    }
    response, err := client.Do(request.Request)
    if err != nil {
        return
    }

    defer response.Body.Close()

    if response.StatusCode >= 300 {
        err = errors.New("something error : " + response.Status)
        return
    }

    err = json.NewDecoder(response.Body).Decode(&result)
    if err != nil {
        return
    }
}

```

```

    return
}

func JsonEncode(obj interface{}) string {
    json, _ := json.MarshalIndent(obj, "", " ")
    return string(json)
}

```

- b. berikut adalah beberapa langkah yang dapat diikuti:
- i. Identifikasi area yang mempengaruhi performa: Perhatikan bagian mana dari aplikasi yang mengalami masalah performa. Apakah ada bagian tertentu yang lambat atau mengkonsumsi banyak sumber daya? Anda dapat menggunakan profilis untuk mengidentifikasi area yang mempengaruhi performa, seperti **go tool pprof** atau paket pihak ketiga seperti **pprof** atau **net/http/pprof**.
  - ii. Gunakan pemantauan dan logging: Implementasikan pemantauan dalam aplikasi Anda untuk melacak metrik performa yang relevan, seperti waktu respons, penggunaan memori, atau jumlah permintaan per detik. Selain itu, pastikan Anda memiliki logging yang memadai untuk mencatat aktivitas penting, termasuk waktu eksekusi atau pemanggilan fungsi yang berpotensi mempengaruhi performa.
  - iii. Profil dan analisis kinerja: Gunakan profilis untuk memperoleh informasi tentang penggunaan CPU, alokasi memori, pemanggilan fungsi, dan lainnya. Analisis hasil profilis dapat membantu Anda mengidentifikasi hot spot yang berpotensi mempengaruhi performa. Gunakan alat profilis bawaan seperti **go tool pprof**, atau alat profilis pihak ketiga seperti **pprof** atau **stackimpact** untuk memperoleh data yang lebih terperinci.
  - iv. Optimasi kode: Setelah mengidentifikasi area yang mempengaruhi performa, tinjau kode yang terkait dengan area tersebut. Perhatikan algoritma dan logika yang digunakan. Anda dapat mencari cara untuk mengoptimalkan atau menyederhanakan kode. Misalnya, menghindari pengulangan yang tidak perlu, menggunakan struktur data yang lebih efisien, atau meminimalkan akses ke I/O yang mahal.
  - v. Caching: Pertimbangkan penggunaan caching untuk mengurangi waktu eksekusi dan beban sistem. Anda dapat menggunakan cache di tingkat aplikasi atau memanfaatkan solusi cache seperti Redis atau memcached untuk menyimpan data yang sering diakses.
  - vi. Optimasi database: Jika masalah performa berkaitan dengan database, pastikan Anda telah mengindeks dengan benar dan mempertimbangkan skema database yang efisien. Lakukan query tuning untuk meningkatkan performa query yang sering dieksekusi. Gunakan fitur-fitur database seperti caching query, connection pooling, atau partitioning untuk meningkatkan performa.
  - vii. Pembaruan versi Golang: Pastikan Anda menggunakan versi Golang yang terbaru. Terkadang, pembaruan versi Golang dapat memberikan peningkatan performa dan perbaikan bug yang relevan.
  - viii. Uji dan pemantauan lanjutan: Setelah melakukan perbaikan, lakukan uji coba dan pemantauan ulang untuk memverifikasi apakah perubahan yang Anda lakukan telah

meningkatkan performa aplikasi. Lakukan tes beban (load testing) dan uji skenario ekstrim untuk memvalidasi performa setelah perbaikan dilakukan.

- c. Untuk memulai dan mengembangkan aplikasi backend dengan Golang yang menerima request HTTP dan mengembalikan response berupa data JSON, berikut adalah langkah-langkah yang dapat Anda ikuti:
  - i. Persiapan awal:
    1. Pastikan Anda telah menginstal Golang di komputer Anda dan mengatur GOPATH dan PATH dengan benar.
    2. Buatlah direktori proyek baru untuk aplikasi Anda.
    3. Inisialisasi modul Go dengan menjalankan perintah `go mod init nama-modul` di direktori proyek Anda.
  - ii. Pemahaman kebutuhan aplikasi:
    1. Tentukan secara rinci kebutuhan fungsional dan non-fungsional dari aplikasi Anda.
    2. Identifikasi entitas utama, endpoint HTTP yang akan digunakan, dan operasi yang akan dilakukan.
  - iii. Desain dan implementasi:
    1. Mulailah dengan membuat file utama `main.go`.
    2. Gunakan package `net/http` untuk menangani request dan response HTTP atau bisa menggunakan framework seperti `echo`, `fiber`, `gin` dan lain - lain.
    3. Implementasikan handler fungsi untuk setiap endpoint HTTP yang Anda butuhkan.
    4. Gunakan package `encoding/json` untuk menghasilkan response JSON atau yang sudah disiapkan oleh framework.
    5. Pastikan Anda memahami prinsip-prinsip RESTful API dan mengikuti konvensi yang baik dalam desain API Anda, seperti menggunakan metode HTTP yang sesuai (GET, POST, PUT, DELETE), menyusun URL dengan baik, dan menggunakan kode status HTTP yang benar dalam respons.
  - iv. Keamanan:
    1. Pastikan Anda melakukan validasi input yang masuk untuk mencegah serangan seperti SQL injection atau XSS (cross-site scripting).
    2. Periksa dan implementasikan mekanisme otentikasi dan otorisasi jika diperlukan.
    3. Selalu gunakan HTTPS untuk mengamankan komunikasi antara aplikasi Anda dan klien.
  - v. Logging dan pemantauan:
    1. Implementasikan sistem logging untuk mencatat kegiatan penting dalam aplikasi.
    2. Pertimbangkan penggunaan alat pemantauan seperti `sentry` atau `slack`.
  - vi. Dokumentasi:
    1. Buatlah dokumentasi yang jelas dan mudah dipahami untuk API Anda.
    2. Gunakan alat-alat seperti `Swagger` atau `OpenAPI` untuk menghasilkan dokumentasi API yang interaktif.