

***Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.***  
***Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.***  
***Selamat belajar, semoga sukses !***

<b>Nama Mahasiswa:</b> <b>Andi Yulia.H</b> .....	<b>NIM:</b> <b>1301164235</b> .....	<b>Nilai:</b> .....
<b>Nama Mahasiswa:</b> <b>Nanda Safira</b> .....	<b>NIM:</b> <b>1301164235</b> .....	<b>Nilai:</b> .....
<b>Nama Mahasiswa:</b> <b>Khulafaur Rasyidin</b> .....	<b>NIM:</b> <b>1301188571</b> .....	<b>Nilai:</b> .....

**Siapkan tools berikut sebelum mengerjakan:**

1. Go Programming Language (<https://golang.org/dl/>).
2. Visual Studio Code (<https://code.visualstudio.com/>) atau LiteIDE (<https://github.com/visualfc/liteide>).
3. Harus menggunakan linux dengan distro fedora (<https://getfedora.org/id/workstation/>).
4. Buatlah git repository pada <https://github.com/> kemudian push semua kode dan hasil laporan anda ke dalam repository github yang sudah anda buat.
5. Lakukan instalasi flatbuffer (<https://google.github.io/flatbuffers/>) untuk mengerjakan salah satu tugas pada modul ini.
6. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.
7. Link repository harus berbeda untuk setiap tugasnya. Buatlah markdown yang rapi disetiap repository tugas yang anda kumpulkan.
8. Printscreen program harus dari desktop kelompok anda sendiri, dan harus dari linux yang sudah diinstall. Jika tidak, maka harus mengulang pengerjaan tugasnya.
9. Jangan lupa untuk menuliskan NAMA dan NIM pada laporan.
10. Laporan berbentuk PDF dan dikumpulkan pada link repository github beserta kodenya.
11. Walaupun tugas berkelompok tapi pengumpulan link github harus individu, jika tidak mengumpulkan maka dianggap tidak mengerjakan.

Nama:	NIM:	Nilai:
-------	------	--------

### Soal No 1 (JSON Marshal)

```
package main

import (
    "encoding/json"
    "fmt"
)

type Person struct {
    FirstName string `json:"firstName"`
    LastName  string `json:"lastName"`
}

func main() {
    bytes, err := json.Marshal(Person{
        FirstName: "John",
        LastName:  "Dow",
    })
    if err != nil {
        panic(err)
    }
    fmt.Println(string(bytes))
}
```

Jalankan program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:

Cara kerjanya yaitu :

Data json dituliskan sebagai string. Program diatas adalah contoh cara decoding json ke bentuk objek. Dengan menggunakan json.Unmarshal, json string bisa dikonversi menjadi bentuk objek. Pertama import package, lalu siapkan struct Person yang nantinya digunakan untuk membuat variabel baru. Data json string pada kode di atas di casting terlebih dahulu ke tipe byte sebelum dipergunakan pada fungsi unmarshal. Data json yang akan diparsing memiliki dua property yaitu FirstName dan LastName. Dengan menambahkan tag json, maka property FirstName dan LastName struct akan secara otomatis menampung data json property FirstName dan LastName.

### Soal No 2 (JSON Unmarshal)

```
package main

import (
    "encoding/json"
    "fmt"
)

type Person struct {
    FirstName string `json:"firstName"`
    LastName  string `json:"lastName"`
}

func main() {
    in := `{"firstName":"John","lastName":"Doe"}`
    bytes := []byte(in)

    var p Person
    err := json.Unmarshal(bytes, &p)
    if err != nil {
        panic(err)
    }

    fmt.Printf("%v", p)
}
```

Jalankan program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:

Cara kerjanya yaitu :

Pada program diatas, data slice struct dikonversi ke dalam bentuk json string. Hasilnya dikonversi ke dalam bentuk byte. Fungsi marshal berfungsi untuk decoding data ke json string. Fungsi marshal akan digunakan untuk melakukan encode data objek ke bentuk json string.

### Soal No 3 (Flatbuffer dan Protocol Buffer)

Jalankan program pada repository github berikut: <https://github.com/jonog/grpc-flatbuffers-example>

Berikan analisis berupa:

1. Apakah outputnya (berikan printscreen)!
2. Jelaskan cara kerjanya dan buatlah diagram FSMnya!
3. Analisis perbedaan dari protocol buffer dan flatbuffer!

Jawaban: