Telkom University	**Tugas 4** **Pemrograman Jaringan (CSH4V3)** **----------** **Semester Genap 2018 - 2019** **Dosen: Aulia Arif Wardana, S.Kom., M.T. (UIW)**

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

Nama Mahasiswa: **Aup Hakim Nurzaman** ……	**NIM:** **1301164395** ……	**Nilai:** ……
Nama Mahasiswa: ……	**NIM:** ……	**Nilai:** ……
Nama Mahasiswa: ……	**NIM:** ……	**Nilai:** ……

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. Disarankan untuk 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. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.
5. Lakukan instalasi flatbuffer (https://google.github.io/flatbuffers/) untuk mengerjakan salah satu tugas pada modul ini.

Soal No 1 (JSON Marshal)

Nama:	NIM:	Nilai:

```go
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:

```
[aupnurzaman@localhost tugas4]$ go run jsonmarshal.go
{"firstName":"John","lastName":"Dow"}
[aupnurzaman@localhost tugas4]$
```

Fungsi *json.Marshal* digunakan untuk decoding data ke json. Data tersebut bisa berupa variable objek cetakan struct map[string]interface{}, bisa juga bertipe array.

Program pada nomor 1 adalah contoh cara encode data ke bentuk json.

- Pertama import package yang dibutuhkan dan siapkan struct *Person.*
- Hasil encode nantinya akan disimpan ke variable objek cetakan struct *Person.* Buat contoh struct *Person.*
- Buat json dari contoh data Buat pesan ketika error
- Hasil encode adalah bertipe [ ]byte. Casting ke *string* bisa digunakan untuk menampilkan data.

Soal No 2 (JSON Unmarshal)

Nama:	NIM:	Nilai:

```
package main

import (
        "encoding/json"
        "fmt"
)

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

func main() {
        in := `{"firstName":"John","lastName":"Dow"}`
        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:

Pada *code* di atas, ***json*** dikonversi kedalam bentuk objek. Hasil konversi berupa []bytes, lalu dicasting terlebih dahulu ke tipe []byte, karena fungsi ***json.Unmarshal*** hanya menerima data bertipe []byte. Dalam fungsi json.Unmarshal, variabel penampung hasil decode harus di-pass dalam bentuk pointer, contohnya seperti **&p**. seperti output :

```
[aupnurzaman@localhost tugas4]$ go run jsonunmarshal.go
{FirstName:John LastName:Dow}[aupnurzaman@localhost tugas4]$
```

Nama:	NIM:	Nilai:

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!