

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

Nama Mahasiswa: Akbar Agus Wijaya	NIM: 13011572	Nilai:
Nama Mahasiswa: Stana Edro Swargara	NIM: 1301188539	Nilai:
Nama Mahasiswa: Dzulfikar Nur Ahmad Faisal	NIM: 1301188596	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. 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:

```

Kam 3 Okt 21:41
wijaya@192:~/go/src/Tugas5Akbar

[wijaya@192 ~]$ cd go
[wijaya@192 go]$ cd src/Tugas5Akbar
[wijaya@192 Tugas5Akbar]$ go run tugasno1.go
# command-line-arguments
./tugasno1.go:9:22: invalid character literal (more than one character)
./tugasno1.go:10:21: invalid character literal (more than one character)
./tugasno1.go:18:9: syntax error: unexpected err at end of statement
./tugasno1.go:22:5: syntax error: non-declaration statement outside function body
[wijaya@192 Tugas5Akbar]$ ^C
[wijaya@192 Tugas5Akbar]$ go run tugasno1.go
{"firstName":"John","lastName":"Dow"}
[wijaya@192 Tugas5Akbar]$

```

Fungsi `JSON.Marshal` digunakan untuk decoding data ke json string. Sumber data bisa berupa variable objek cetakan struct. Hasil konversi berupa []byte, casting terlebih dahulu ke tipe string agar bisa ditampilkan bentuk json stringnya. Marshal memiliki beberapa keterbatasan ketat:

- Peta kunci harus string
- Nilai-nilai peta harus jenis Defender oleh paket json

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

- Jenis berikut tidak didukung :Channel, kompleks, dan fungsi
- Struktur data siklik yang tidak didukung
- Petunjuk akan dikodekan sebagai nilai-nilai yang menunjukkan null

Soal No 2 (JSON Unmarshal)

```
package main

import (
    "encoding/json"
    "fmt"
)

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

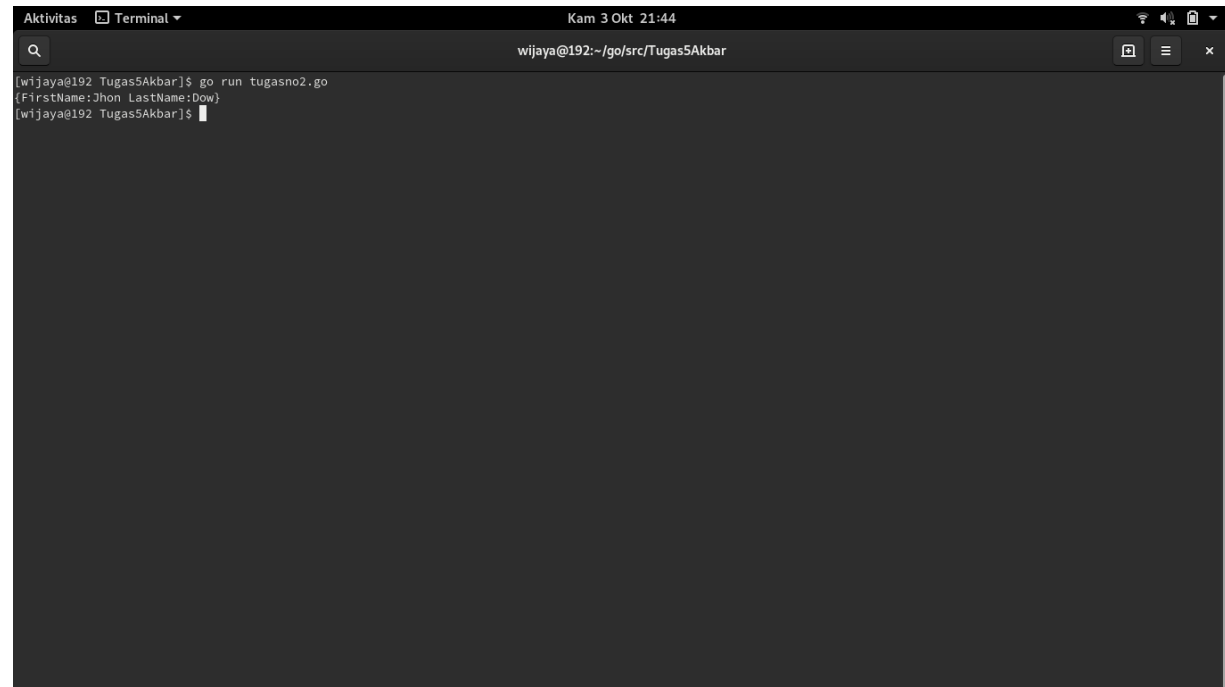
func main() {
    in := `{"firstName":"John","lastName":"Dow"}`
    bytes := []byte(in)

    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:



```
Kam 3 Okt 21:44
wijaya@192:~/go/src/TugasSAkbar

[wijaya@192 TugasSAkbar]$ go run tugasno2.go
{FirstName:Jhon LastName:Dow}
[wijaya@192 TugasSAkbar]$
```

Fungsi JSON.Unmarshal yaitu mengambil sepotong byte yang mudah-mudahan mewakili berlaku JSON dan antarmuka tujuan, yang biasanya pointer ke struct atau jenis dasar.

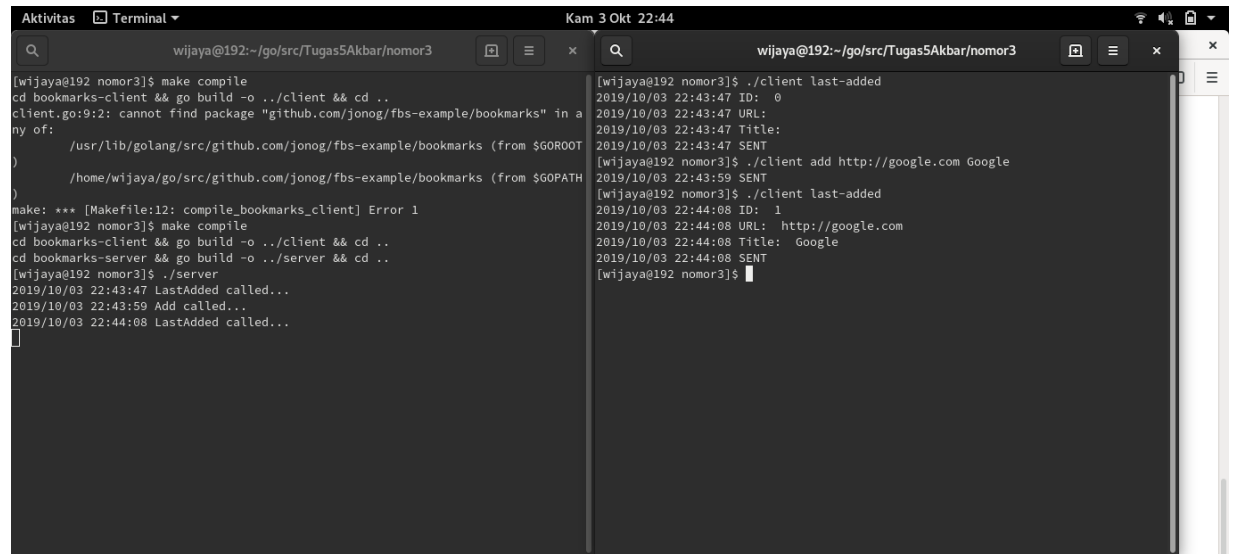
**Soal No 3 (Flatbuffer dan Protocol Buffer)**

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

Berikan analisis berupa:

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

Jawaban:



```
wijaya@192:~/go/src/Tugas5Akbar/nomor3
[wijaya@192 nomor3]$ make compile
cd bookmarks-client && go build -o ../client && cd ..
client.go:9:2: cannot find package "github.com/jonog/fbs-example/bookmarks" in any of:
    /usr/lib/golang/src/github.com/jonog/fbs-example/bookmarks (from $GOROOT)
    /home/wijaya/go/src/github.com/jonog/fbs-example/bookmarks (from $GOPATH)
make: *** [Makefile:12: compile_bookmarks_client] Error 1
[wijaya@192 nomor3]$ make compile
cd bookmarks-client && go build -o ../client && cd ..
cd bookmarks-server && go build -o ../server && cd ..
[wijaya@192 nomor3]$ ./server
2019/10/03 22:43:47 LastAdded called...
2019/10/03 22:43:59 Add called...
2019/10/03 22:44:08 LastAdded called...
[wijaya@192 nomor3]$ ./client last-added
2019/10/03 22:43:47 ID: 0
2019/10/03 22:43:47 URL:
2019/10/03 22:43:47 Title:
2019/10/03 22:43:47 SENT
[wijaya@192 nomor3]$ ./client add http://google.com Google
2019/10/03 22:43:59 SENT
[wijaya@192 nomor3]$ ./client last-added
2019/10/03 22:44:08 ID: 1
2019/10/03 22:44:08 URL: http://google.com
2019/10/03 22:44:08 Title: Google
2019/10/03 22:44:08 SENT
[wijaya@192 nomor3]$
```

Download dependencies via [Cinub Releases](#)

- Compile flatc, e.g. `cmake -G"Unix Makefiles" then run make`

gRPC Setup

```
go get google.golang.org/grpc
```

Analisis perbedaan :

- Protocol Buffer : metode serialisasi data terstruktur. Digunakan dalam mengembangkan program untuk berkomunikasi satu sama lain melalui kabel atau menyimpan data.
- Flatbuffer : digunakan untuk mengakses data serial yang tidak memerlukan penyalinan pertama ke bagian memori yang terpisah, yang membuat pengaksesan data dalam format ini jauh lebih cepat daripada data dalam format yang membutuhkan pemrosesan yang lebih luas.

