

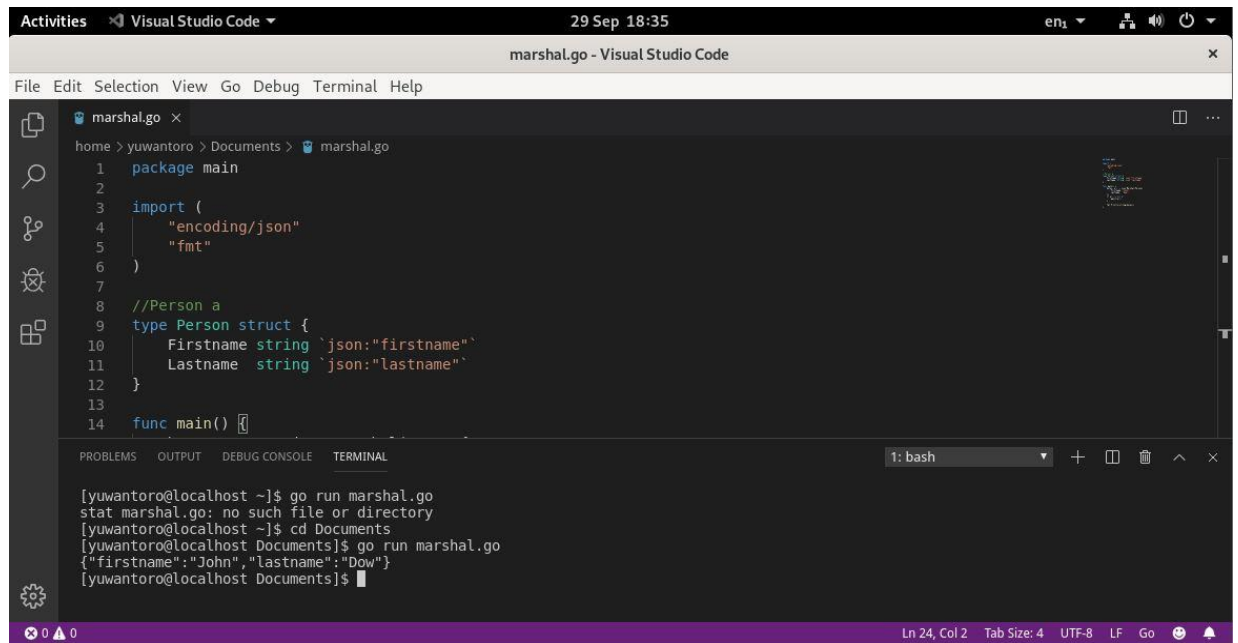
Nama = Yuwantoro M. (1301150042)

Nama = Lukman Budiman (1301164725)

Nama = Salma F.S (1301164442)

Tugas 5 – Pemrograman Jaringan

1.



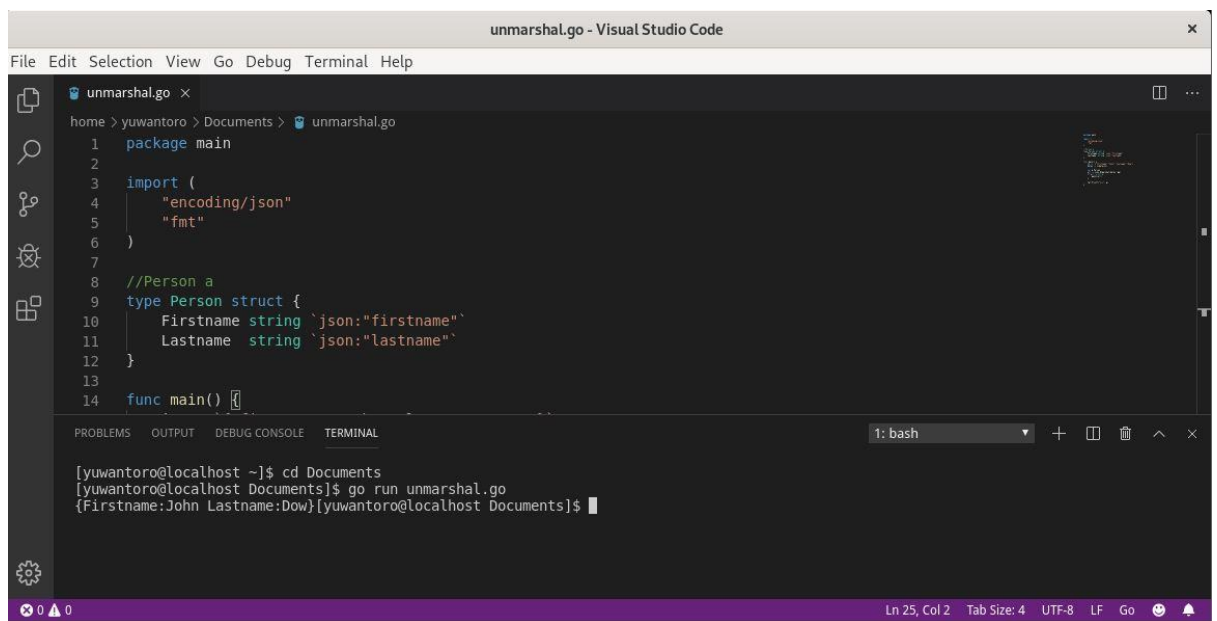
```
Activities Visual Studio Code 29 Sep 18:35 en1
marshal.go - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

marshal.go x
home > yuwantoro > Documents > marshal.go
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6 )
7
8 //Person a
9 type Person struct {
10     Firstname string `json:"firstname"`
11     Lastname  string `json:"lastname"`
12 }
13
14 func main() {
15     // ...
16 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[yuwantoro@localhost ~]$ go run marshal.go
stat marshal.go: no such file or directory
[yuwantoro@localhost ~]$ cd Documents
[yuwantoro@localhost Documents]$ go run marshal.go
{"firstname":"John","lastname":"Dow"}
[yuwantoro@localhost Documents]$
```

Penjelasannya adalah Struct Person diinisialisasi pada variabel bytes dengan atribut nama depan (Firstname) “John” dan nama belakang (Lastname) “Dow”. Kemudian, bytes diserialisasikan kedalam bentuk JSON dan dicetak menghasilkan keluaran struct Person dalam bentuk JSON ({"firstname": "John", "lastname": "Dow"}).

2.



The screenshot shows the Visual Studio Code editor with a file named `unmarshal.go` open. The code defines a `Person` struct with `Firstname` and `Lastname` fields, both of type `string`. The `main` function is partially visible. Below the editor, the terminal window shows the execution of the program. The output indicates that the program successfully unmarshaled a JSON input into a `Person` struct, displaying the result as `{Firstname:John Lastname:Dow}`.

```
unmarshal.go - Visual Studio Code
File Edit Selection View Go Debug Terminal Help

home > yuwantoro > Documents > unmarshal.go
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6 )
7
8 //Person a
9 type Person struct {
10     Firstname string `json:"firstname"`
11     Lastname  string `json:"lastname"`
12 }
13
14 func main() {
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: bash
[yuwantoro@localhost ~]$ cd Documents
[yuwantoro@localhost Documents]$ go run unmarshal.go
{Firstname:John Lastname:Dow}[yuwantoro@localhost Documents]$
```

Ln 25, Col 2 Tab Size: 4 UTF-8 LF Go

Penjelasannya adalah Struct Person dalam bentuk JSON di-assign ke variabel `in`, yang kemudian variabel `in` menjadi nilai untuk variabel bytes dalam tipe data `[]byte`. Kemudian, variabel bytes di decode dari bentuk JSON menjadi bentuk struct Person dan ditampung pada variabel `p`. Sehingga, keluaran yang dihasilkan dari mencetak variabel `p` adalah `{Firstname:John Lastname:Dow}`.

3.

```

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./server
2019/10/03 20:23:36 LastAdded called...

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./client
2019/10/03 20:23:21 Insufficient args provided
[root@localhost grpc-flatbuffers-example]# ./client last-added
2019/10/03 20:23:36 ID: 0
2019/10/03 20:23:36 URL:
2019/10/03 20:23:36 Title:
2019/10/03 20:23:36 SENT

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./server
2019/10/03 20:23:36 LastAdded called...
2019/10/03 20:24:54 Add called...

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./client
2019/10/03 20:23:21 Insufficient args provided
[root@localhost grpc-flatbuffers-example]# ./client last-added
2019/10/03 20:23:36 ID: 0
2019/10/03 20:23:36 URL:
2019/10/03 20:23:36 Title:
2019/10/03 20:23:36 SENT
[root@localhost grpc-flatbuffers-example]# ./client add http://google.com Google
2019/10/03 20:24:54 SENT
[root@localhost grpc-flatbuffers-example]#

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./server
2019/10/03 20:23:36 LastAdded called...
2019/10/03 20:24:54 Add called...
2019/10/03 20:26:00 LastAdded called...

root@localhost:~/grpc-flatbuffers-example
File Edit View Search Terminal Help
[root@localhost grpc-flatbuffers-example]# ./client
2019/10/03 20:23:21 Insufficient args provided
[root@localhost grpc-flatbuffers-example]# ./client last-added
2019/10/03 20:23:36 ID: 0
2019/10/03 20:23:36 URL:
2019/10/03 20:23:36 Title:
2019/10/03 20:23:36 SENT
[root@localhost grpc-flatbuffers-example]# ./client add http://google.com Google
2019/10/03 20:24:54 SENT
2019/10/03 20:26:00 ID: 1
2019/10/03 20:26:00 URL: http://google.com
2019/10/03 20:26:00 Title: Google
2019/10/03 20:26:00 SENT
[root@localhost grpc-flatbuffers-example]#

```

Perbedaan antara protocol buffer dengan flatbuffer adalah pada representasi in-memory dan wire format-nya. Protocol buffer memisahkan representasi in-memory dengan wire protocolnya (butuh parsing dan serialisasi), sementara flatbuffer tidak memisahkan keduanya (serialisasi terjadi disaat pembuatan objek flatbuffer pada representasi in-memory). Hal yang sama terjadi saat objek protocol buffer ingin dikembalikan ke asalnya, terjadi parsing dan deserialisasi lagi. Namun pada flatbuffer, yang ditampilkan adalah pointer kepada objek flatbuffer yang ada.