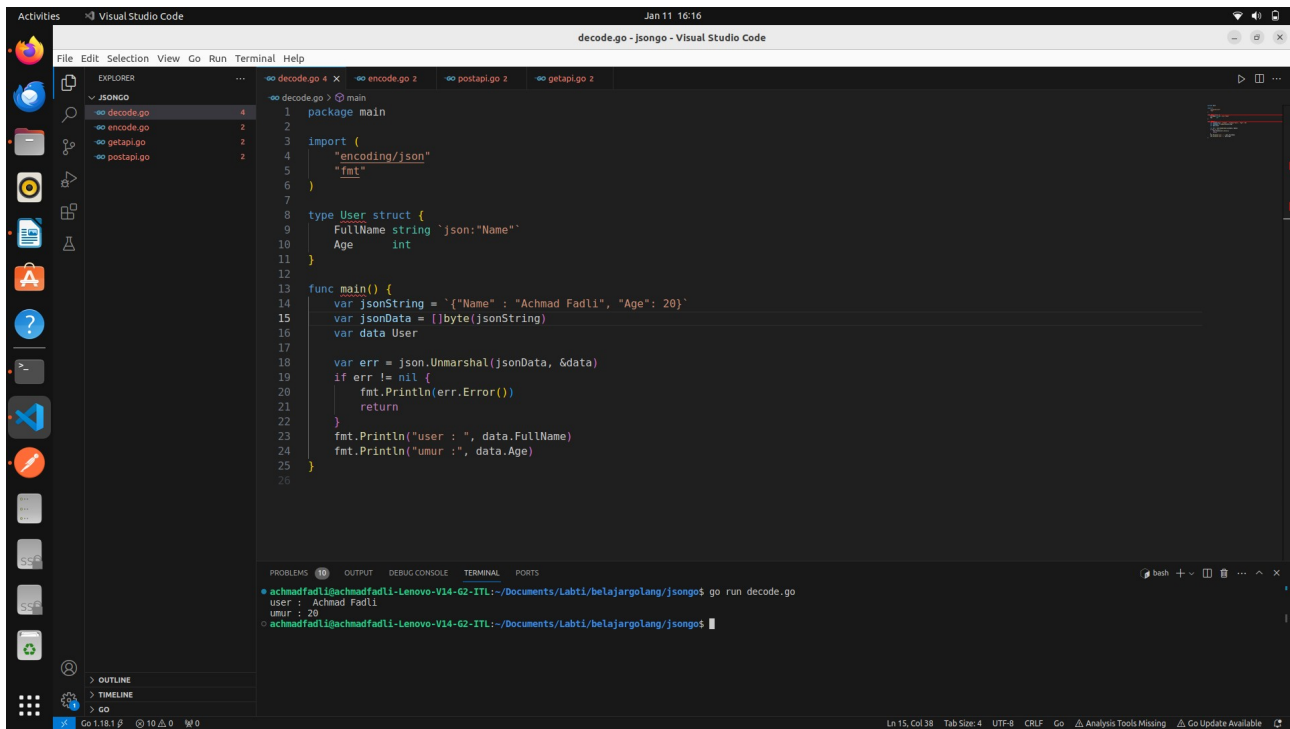


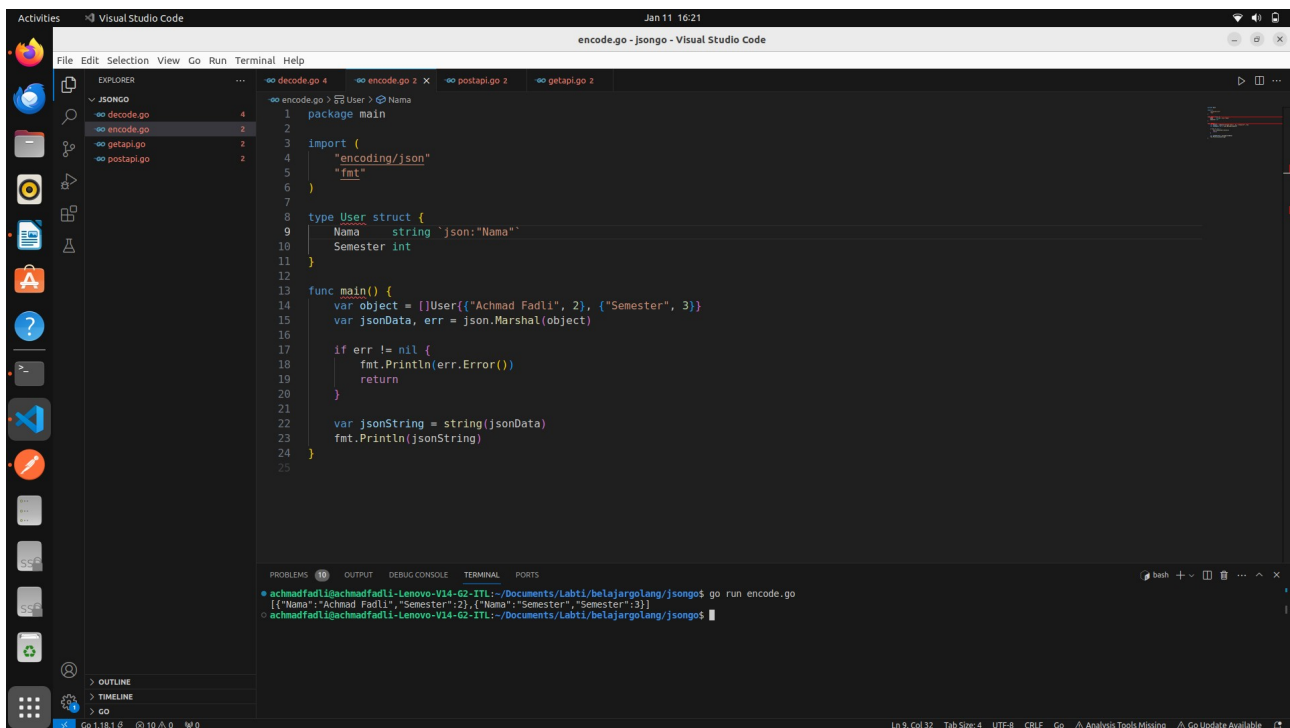
Nama : Achmad Fadli Iskandar  
NPM : 50422069  
Kelas : 2IA13



The screenshot shows the Visual Studio Code editor with the file 'decode.go' open. The code defines a Go struct 'User' with fields 'FullName' and 'Age', and a 'main' function that unmarshals a JSON string into a 'User' struct and prints its details. The terminal output shows the successful execution of the program, displaying the user's name and age.

```
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6 )
7
8 type User struct {
9     FullName string `json:"Name"`
10    Age      int
11 }
12
13 func main() {
14     var jsonString = `{"Name": "Achmad Fadli", "Age": 20}`
15     var jsonData = []byte(jsonString)
16     var data User
17
18     var err = json.Unmarshal(jsonData, &data)
19     if err != nil {
20         fmt.Println(err.Error())
21         return
22     }
23     fmt.Println("user : ", data.FullName)
24     fmt.Println("umur :", data.Age)
25 }
```

```
achmadfadi@achmadfadi-Lenovo-V14-G2-ITL:~/Documents/Labti/belajargolang/jsongos$ go run decode.go
user : Achmad Fadli
umur : 20
achmadfadi@achmadfadi-Lenovo-V14-G2-ITL:~/Documents/Labti/belajargolang/jsongos$
```



The screenshot shows the Visual Studio Code editor with the file 'encode.go' open. The code defines a Go struct 'User' with fields 'Nama' and 'Semester', and a 'main' function that marshals a 'User' struct into a JSON string and prints it. The terminal output shows the successful execution of the program, displaying the JSON string representation of the user data.

```
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6 )
7
8 type User struct {
9     Nama      string `json:"Nama"`
10    Semester  int
11 }
12
13 func main() {
14     var object = []User{{"Achmad Fadli", 2}, {"Semester", 3}}
15     var jsonData, err = json.Marshal(object)
16
17     if err != nil {
18         fmt.Println(err.Error())
19         return
20     }
21
22     var jsonString = string(jsonData)
23     fmt.Println(jsonString)
24 }
25
```

```
achmadfadi@achmadfadi-Lenovo-V14-G2-ITL:~/Documents/Labti/belajargolang/jsongos$ go run encode.go
[{"Nama": "Achmad Fadli", "Semester": 2}, {"Nama": "Semester", "Semester": 3}]
achmadfadi@achmadfadi-Lenovo-V14-G2-ITL:~/Documents/Labti/belajargolang/jsongos$
```

Visual Studio Code interface showing the editor for `getapi.go`. The code defines a REST API for managing students (Mahasiswa).

```
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6     "log"
7     "net/http"
8 )
9
10 type Mahasiswa struct {
11     ID      int    `json:"id"`
12     NPM     int    `json:"npm"`
13     Name    string `json:"name"`
14 }
15
16 func NewMahasiswa() []Mahasiswa {
17     mhs := []Mahasiswa{
18         Mahasiswa{
19             ID: 1,
20             NPM: 50422069,
21             Name: "Achmad Fadli Iskandar",
22         },
23         Mahasiswa{
24             ID: 2,
25             NPM: 51419643,
26             Name: "Dariwan",
27         },
28         Mahasiswa{
29             ID: 3,
30             NPM: 50422102,
31             Name: "Ardian",
32         },
33     }
34     return mhs
35 }
36
37 func GetMahasiswa(w http.ResponseWriter,
38     r *http.Request) {
39     if r.Method == "GET" {
40         mhs := NewMahasiswa()
41         datamahasiswa, err := json.Marshal(mhs)
42         if err != nil {
43             http.Error(w, err.Error(),
44                 http.StatusInternalServerError)
```

Visual Studio Code interface showing the editor for `getapi.go`. The code continues with the `main` function and error handling.

```
45         http.Error(w, err.Error(),
46             http.StatusInternalServerError)
47         w.Header().Set("Content-Type", "application/json")
48         w.WriteHeader(http.StatusOK)
49         w.Write(datamahasiswa)
50         return
51     }
52     http.Error(w, "haya mau ngapain",
53         http.StatusNotFound)
54 }
55
56 func main() {
57     http.HandleFunc("/mahasiswa", GetMahasiswa)
58     fmt.Println("server running...")
59     if err := http.ListenAndServe(":8080", nil); err != nil {
60         log.Fatal(err)
61     }
62 }
```

Terminal output:

```
achmadfadli@achmadfadli-Lenovo-V14-G2-ITL:~/Documents/Labti/belajargolang/jsongo$ go run getapi.go
server running...
```

