

LAPORAN LATIHAN PRAKTIKUM-5

PEMROGRAMAN BERBASIS WEB

KELAS A

Disusun Untuk Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Web



Disusun oleh:

Gibran Kahlil

4522210128

Dosen Pengampu:

Adi Wahyu Pribadi , S.Si., M.Kom

Program Studi Teknik Informatika

Fakultas Teknik Universitas Pancasila

2023/2024

Link GitHub : <https://github.com/gibrankahlil260404/PBW/tree/master/p05>

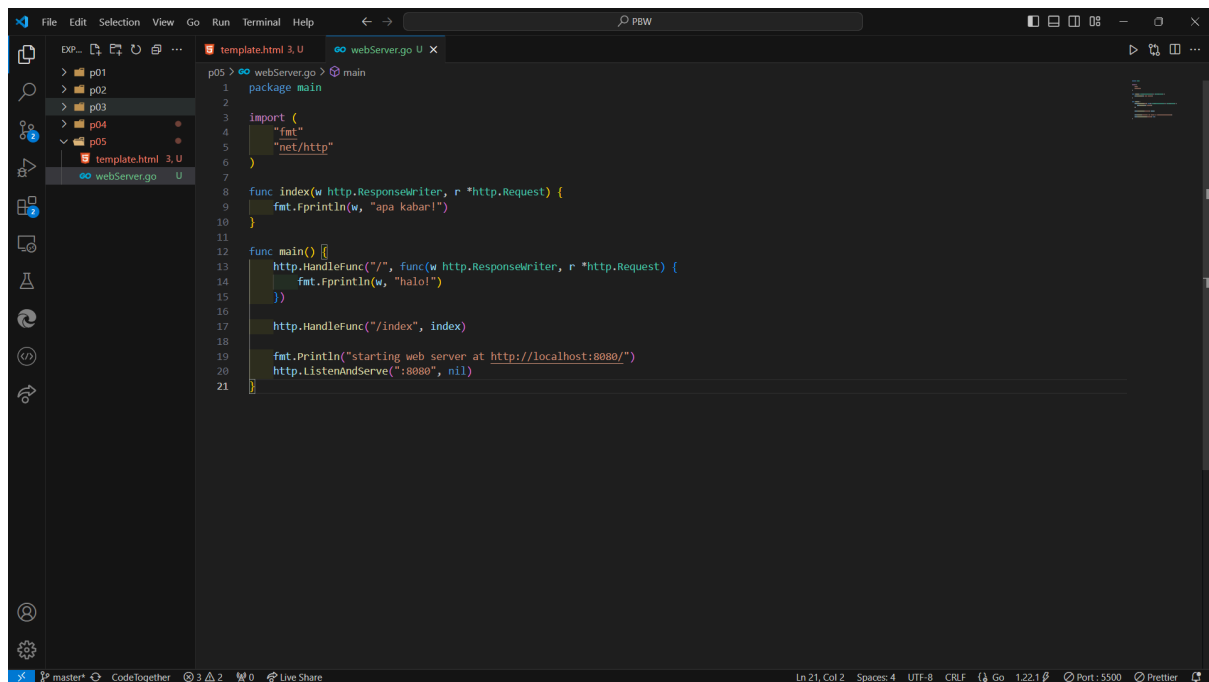
Web Server

Komponen penting dari paket `net/http` dalam Go adalah web server, yang memberikan pengembang kemampuan untuk membuat dan menjalankan server HTTP. Dengan menggunakan paket ini, pengembang dapat membuat aplikasi web dengan cepat tanpa perlu menginstal server terpisah seperti Apache atau Nginx.

Membuat aplikasi web sederhana

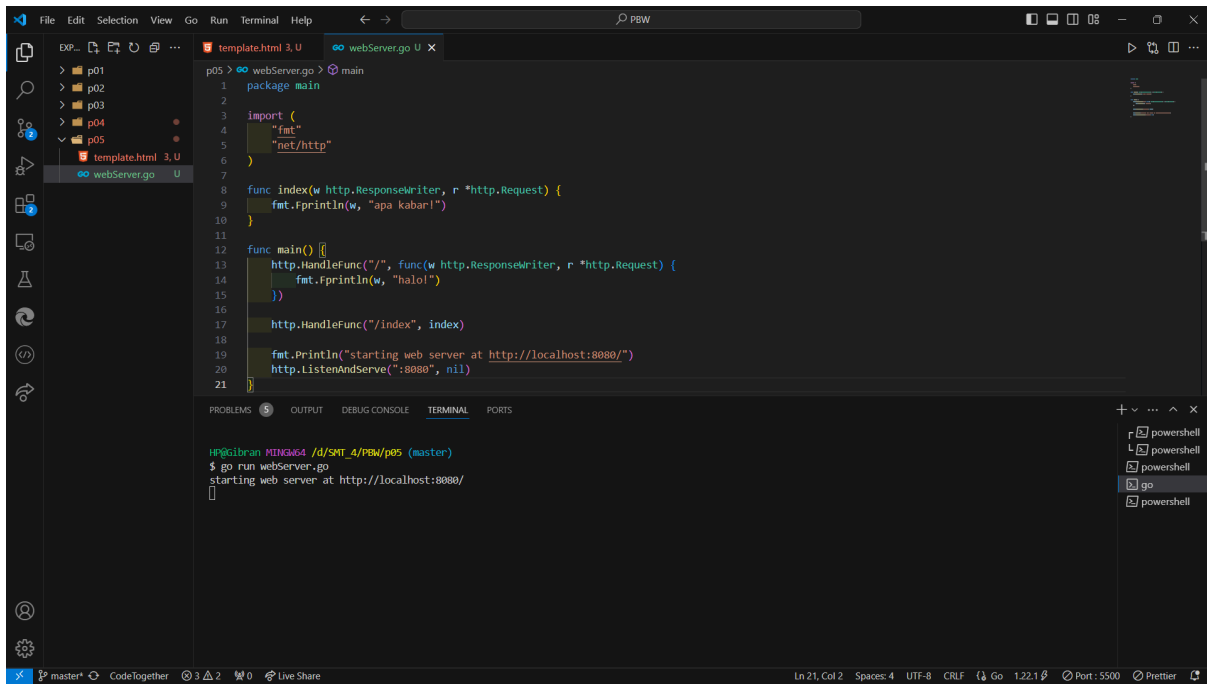
Contoh sederhana untuk memunculkan text di web ketika url tertentu diakses

- Source Code



```
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6 )
7
8 func index(w http.ResponseWriter, r *http.Request) {
9     fmt.Fprintln(w, "apa kabar!")
10 }
11
12 func main() {
13     http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
14         fmt.Fprintln(w, "halo!")
15     })
16     http.HandleFunc("/index", index)
17
18     fmt.Println("starting web server at http://localhost:8080/")
19     http.ListenAndServe(":8080", nil)
20 }
21 }
```

Jalankan program dengan mengetik “`go run WebServer.go`”, setelah berhasil di running akan muncul localhost di dalam terminal dan jika muncul dialog **Do you want the application “bab48” to accept incoming network connections?** atau sejenis, pilih allow.



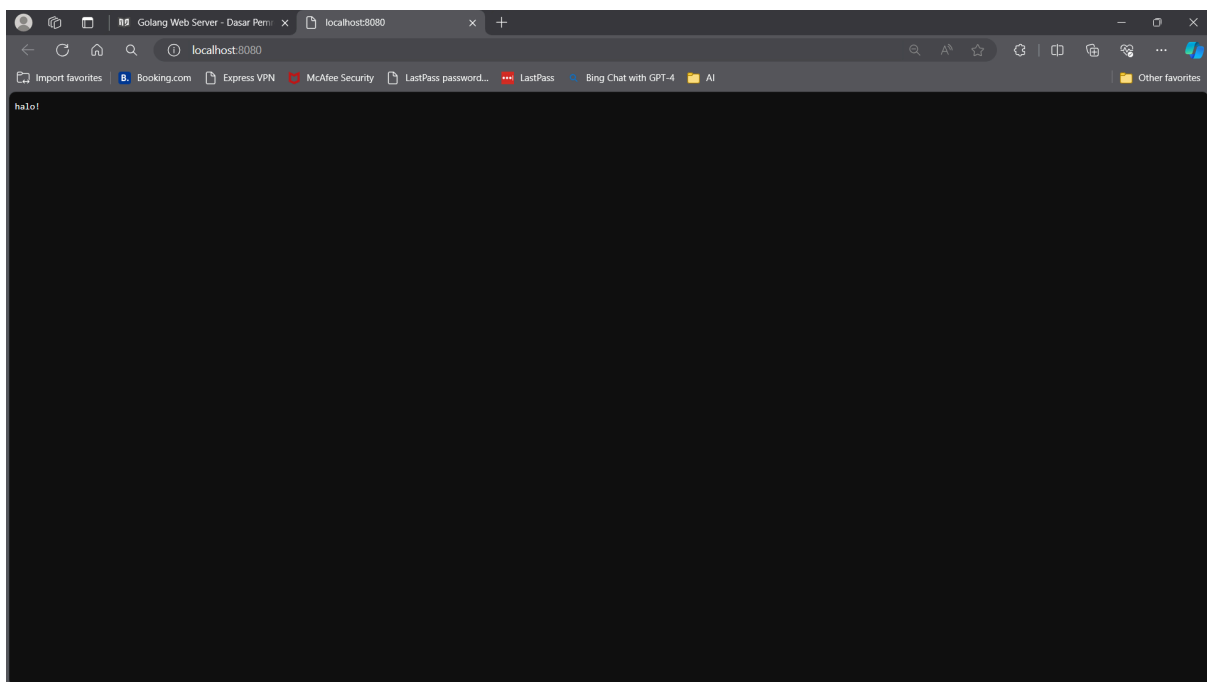
The screenshot shows the Visual Studio Code interface. The left sidebar displays a file explorer with a project structure including folders p01 through p05, and files template.html and webServer.go. The main editor window is open to webServer.go, which contains the following Go code:

```
1 package main
2
3 import (
4     "fmt"
5     "net/http"
6 )
7
8 func index(w http.ResponseWriter, r *http.Request) {
9     fmt.Fprintln(w, "apa kabar!")
10 }
11
12 func main() {
13     http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
14         fmt.Fprintln(w, "halo!")
15     })
16     http.HandleFunc("/index", index)
17     fmt.Println("starting web server at http://localhost:8080/")
18     http.ListenAndServe(":8080", nil)
19 }
20
21
```

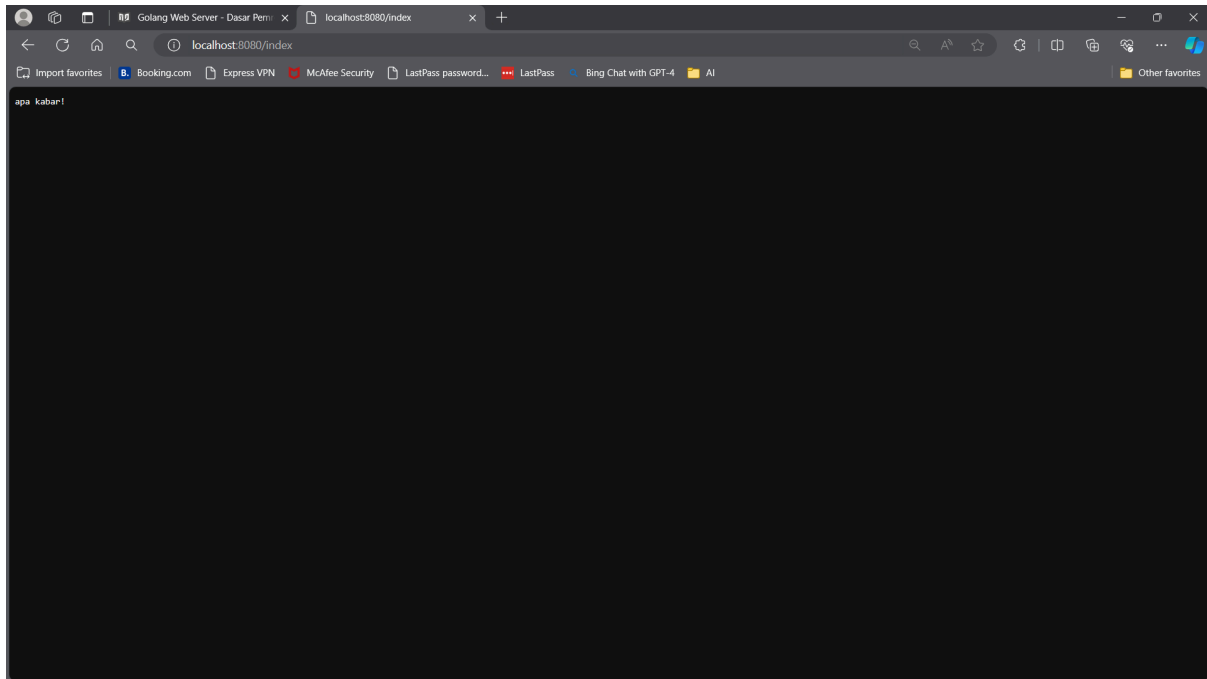
Below the editor, the TERMINAL tab is active, showing the command prompt output:

```
HP@Gihnan MINGW64 /J/SMT_A/PBW/p05 (master)
$ go run webServer.go
starting web server at http://localhost:8080/
```

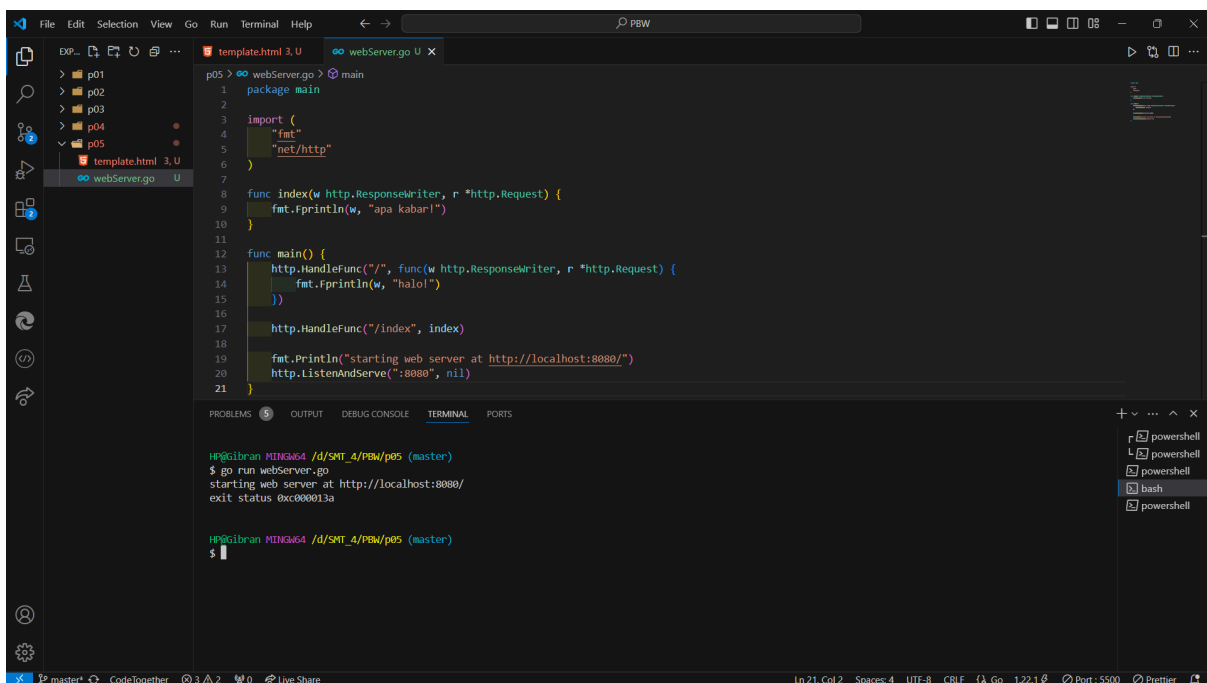
- Hasil running dengan membuka localhost:8080



- Hasil running dengan membuka localhost:8080/index



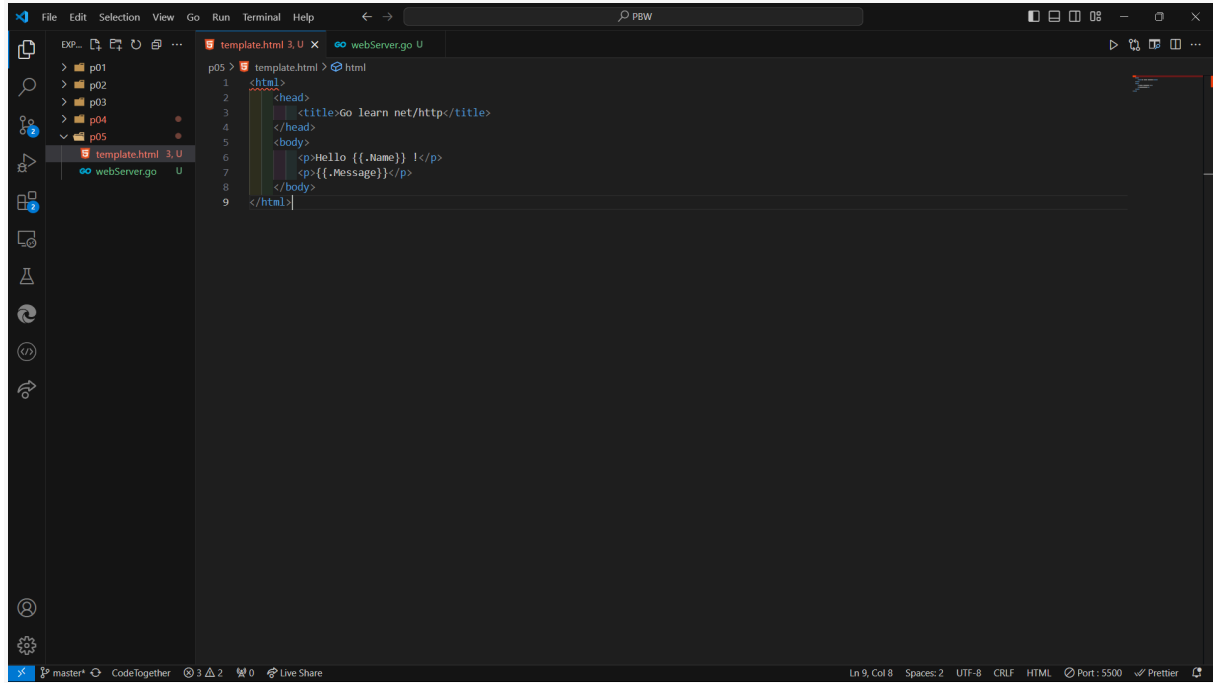
Note: Setiap perubahan pada file .go, go run harus dipanggil ulang, sebelum running ulang terlebih dahulu keluar server dengan mengetik **CTRL + C**, setelah itu akan ada keterangan “exit status 0xc000013a” yang artinya sudah keluar dari server



Penggunaan Template Web

Buatlah file html dengan nama `template.html`

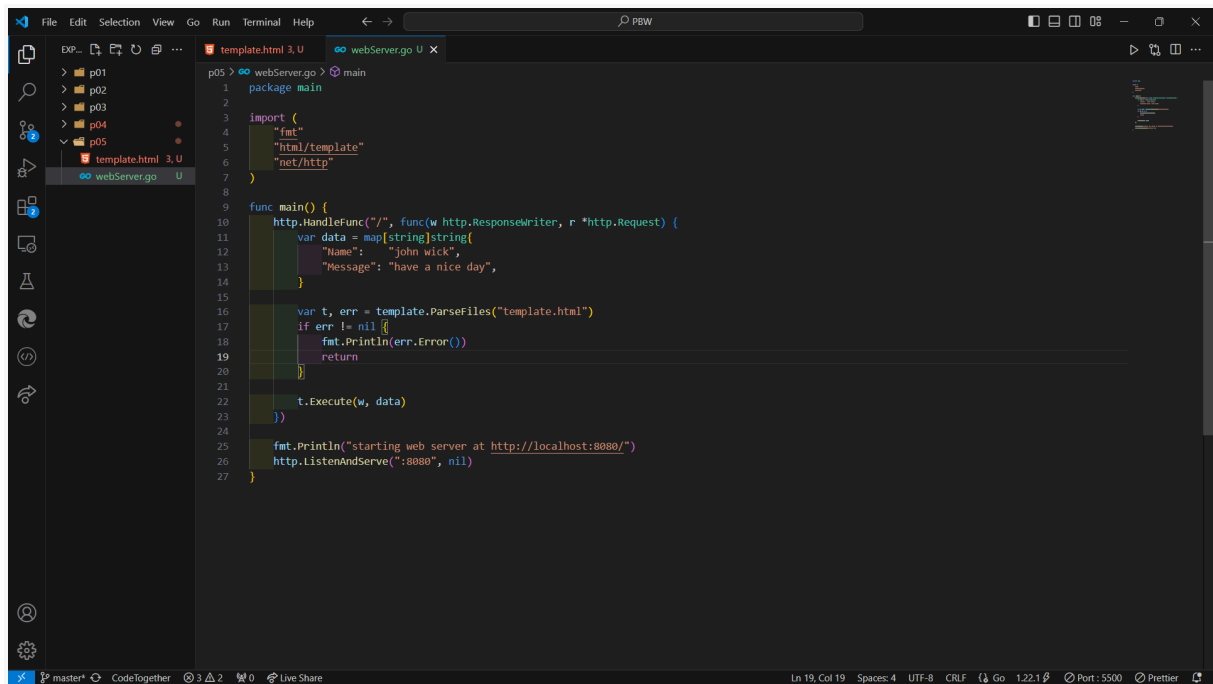
- **Source Code**



A screenshot of the Visual Studio Code editor. The file explorer on the left shows a project structure with folders p01 through p05, and files template.html and webServer.go. The main editor window displays the content of template.html, which is an HTML template with placeholders for Name and Message.

```
1 <html>
2 <head>
3   <title>Go learn net/http</title>
4 </head>
5 <body>
6   <p>Hello {{.Name}} !</p>
7   <p>{{.Message}}</p>
8 </body>
9 </html>
```

- **Ubah Source Code Go sebelumnya menjadi berikut ini**

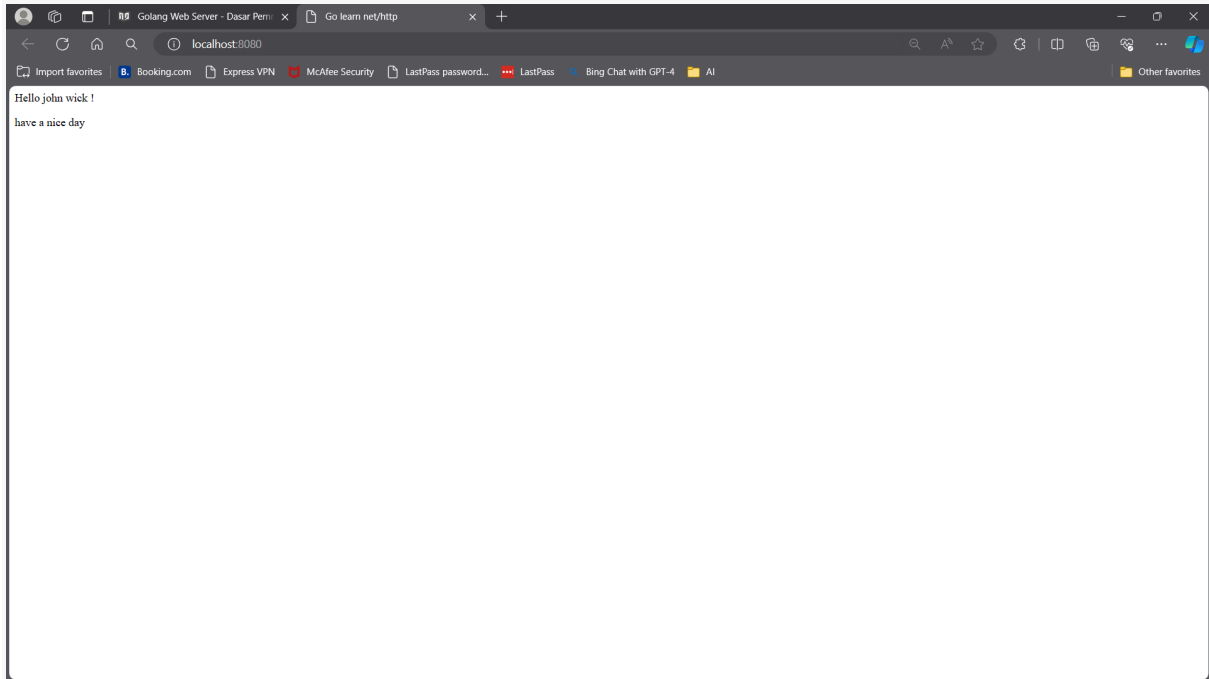


A screenshot of the Visual Studio Code editor showing the updated Go source code for webServer.go. The code uses the Go template package to render the HTML template.

```
1 package main
2
3 import (
4     "fmt"
5     "html/template"
6     "net/http"
7 )
8
9 func main() {
10     http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
11         var data = map[string]string{
12             "Name": "john wick",
13             "Message": "have a nice day",
14         }
15
16         var t, err = template.ParseFiles("template.html")
17         if err != nil {
18             fmt.Println(err.Error())
19             return
20         }
21         t.Execute(w, data)
22     })
23
24     fmt.Println("starting web server at http://localhost:8080/")
25     http.ListenAndServe(":8080", nil)
26 }
27
```

Jalankan ulang program dengan mengetikkan “go run WebServer.go” seperti langkah running di awal serta buka ulang localhost yang berada di terminal.

- **Hasil running dengan membuka localhost**



Kesimpulan: Dari praktikum 05 ini saya mengerti dasar dari Web Server mulai dari, apa itu web server, cara kerja web server, dan cara membuat web server.