

**LAPORAN LATIHAN PRAKTIKUM KE-5**  
**PEMROGRAMAN BERBASIS WEB**  
**KELAS A**

Disusun Untuk Memenuhi Tugas Mata Kuliah Prak. Pemrograman Berbasis Web



Disusun oleh:

Arsya Yan Duribta

4522210117

Dosen Pengampu:

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

**Program Studi Teknik Informatika**  
**Fakultas Teknik Universitas Pancasila**  
**2023/2024**

## Link Repository Github:

<https://github.com/Arsyayd11/pbw/tree/master/praktikum/pertemuan05/51-web-server>

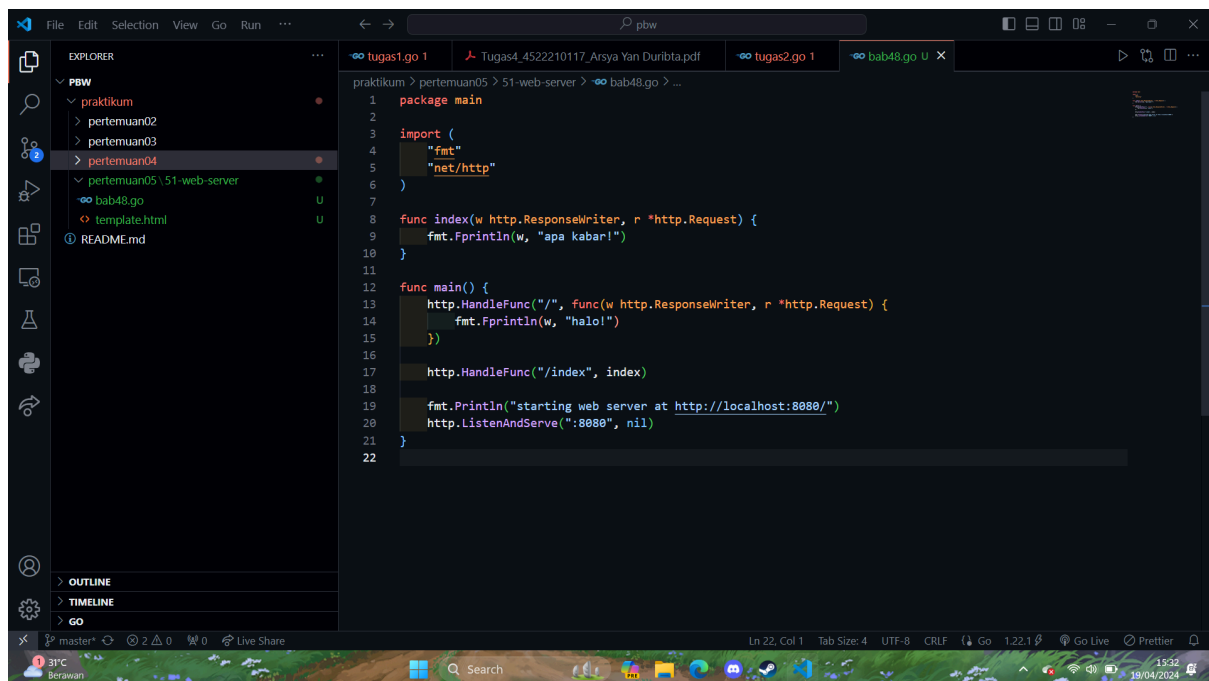
## Web Server

Go menyediakan package `net/http`, yang menyediakan berbagai macam fitur untuk pembuatan aplikasi berbasis web, termasuk web server, routing, dan templating. Go memiliki web server bawaan, yang berbeda dengan bahasa pemrograman lain yang memerlukan server terpisah (seperti Apache untuk PHP atau IIS untuk .NET). Pada bagian ini, mempelajari pembuatan aplikasi web sederhana dan pemanfaatan template untuk mendesain tampilan.

### 51.1. Membuat Aplikasi Web Sederhana

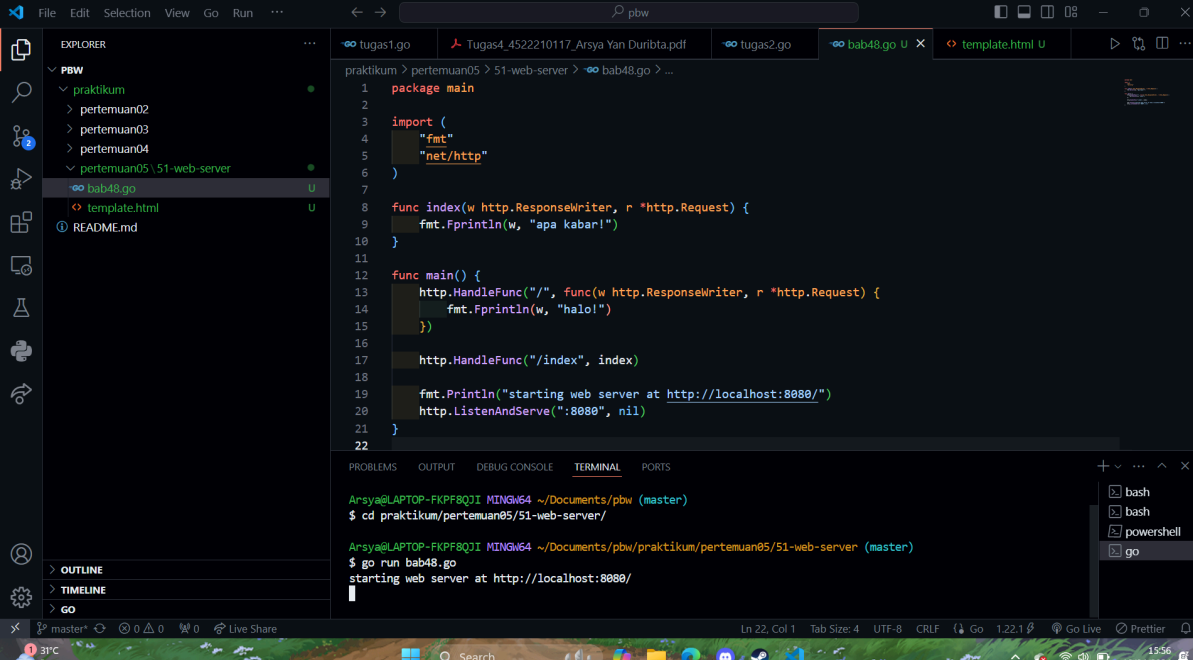
Berikut adalah contoh sederhana untuk menampilkan teks di web ketika URL tertentu diakses:

Implementasinya (Source Code):

A screenshot of a code editor showing a Go web server implementation. The Explorer on the left shows a project structure with folders 'praktikum', 'pertemuan02', 'pertemuan03', 'pertemuan04', and 'pertemuan05'. Under 'pertemuan05', there is a sub-folder '51-web-server' containing 'bab48.go', 'template.html', and 'README.md'. The main editor displays the source code for 'bab48.go'. The code defines a package 'main', imports 'fmt' and 'net/http', and implements an 'index' handler function and a 'main' function. The 'index' function prints 'apa kabar!' to the response writer. The 'main' function sets up a web server on 'localhost:8080' that responds to the root path with 'halo!' and to the '/index' path with the 'index' handler. The status bar at the bottom indicates the file is at line 22, column 1, with a tab size of 4, UTF-8 encoding, and CRLF line endings. The system tray at the very bottom shows a temperature of 31°C and the date 19/04/2024.

```
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
17     http.HandleFunc("/index", index)
18
19     fmt.Println("starting web server at http://localhost:8080/")
20     http.ListenAndServe(":8080", nil)
21 }
22
```

Run code menggunakan terminal:



The image shows a Visual Studio Code editor window with a Go web server code file named `bab48.go` and a terminal window running the code.

**Code File: `bab48.go`**

```
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
17     http.HandleFunc("/index", index)
18
19     fmt.Println("starting web server at http://localhost:8080/")
20     http.ListenAndServe(":8080", nil)
21 }
22
```

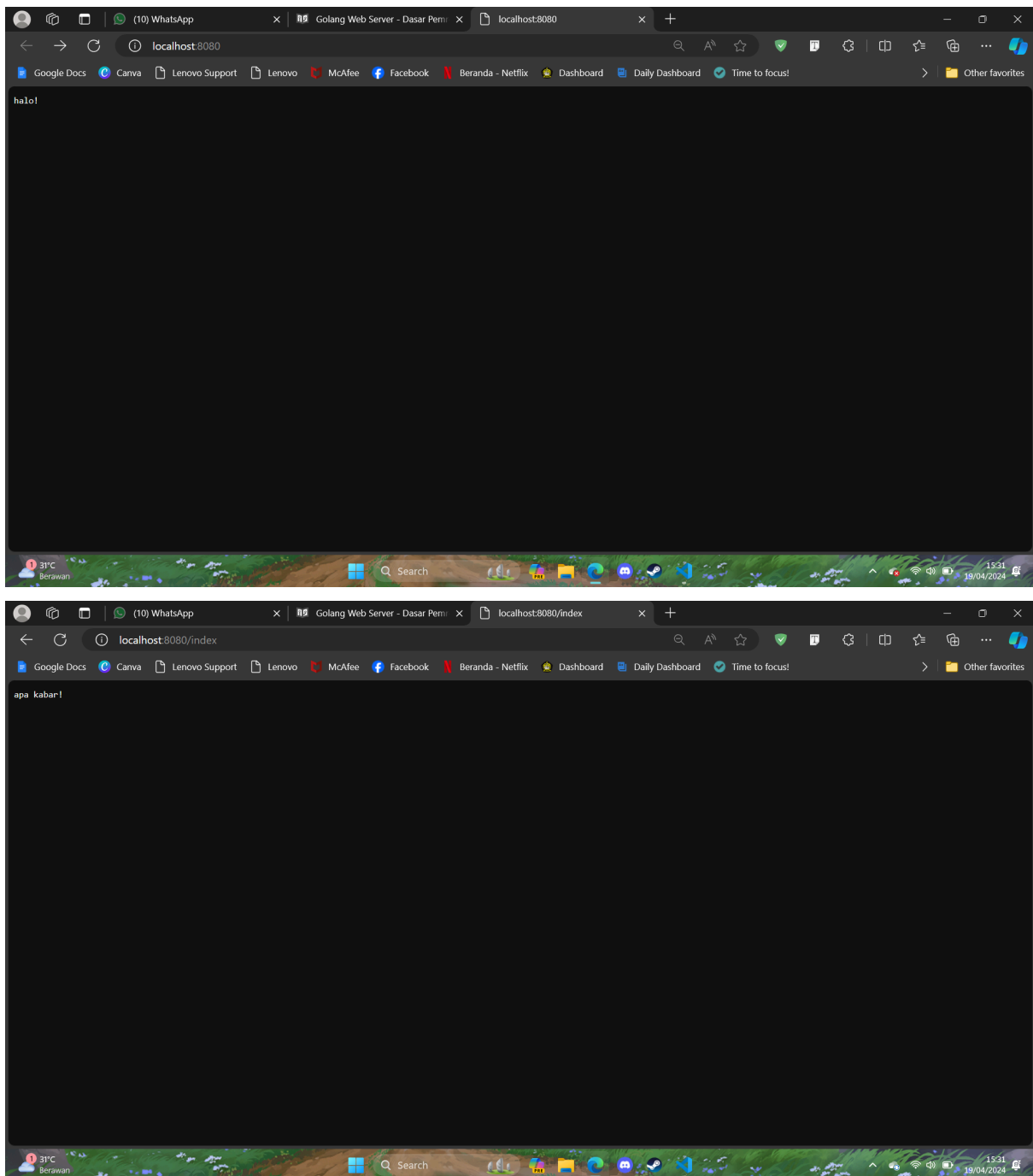
**Terminal Output:**

```
Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw (master)
$ cd praktikum/pertemuan05/51-web-server/

Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw/praktikum/pertemuan05/51-web-server (master)
$ go run bab48.go
starting web server at http://localhost:8080/
```

The terminal output shows the command `go run bab48.go` being executed, which starts the web server at `http://localhost:8080/`.

Output ketika menampilkannya di web browser:



Penjelasan:

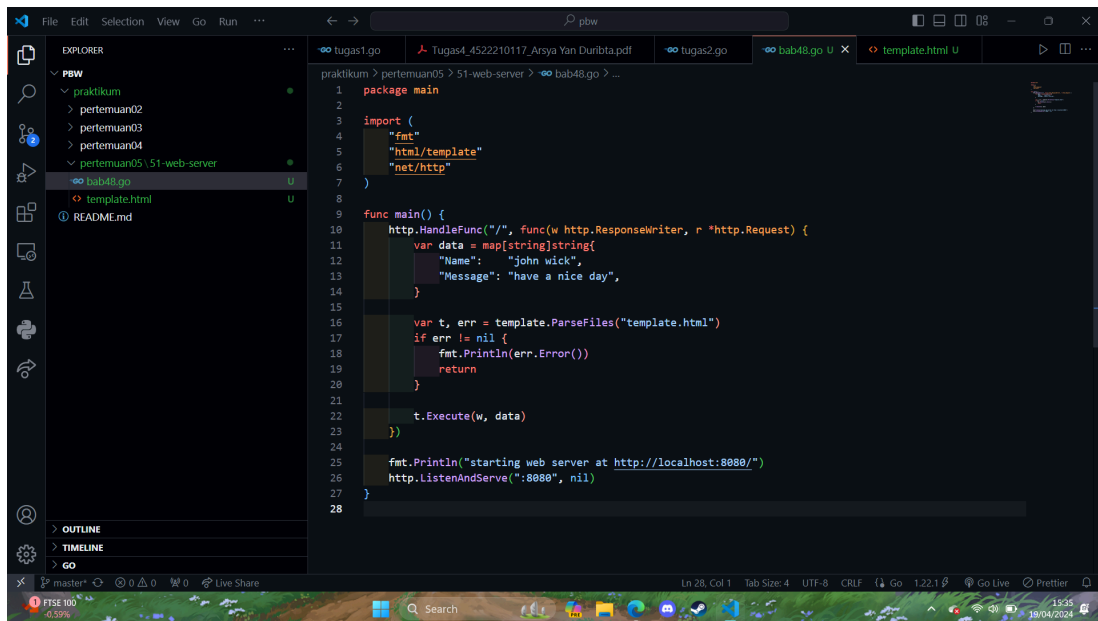
Fungsi `http.HandleFunc()` digunakan untuk routing aplikasi web. Dua rute didaftarkan: `/` dan `/index`. Rute `/` menampilkan teks "halo" sedangkan `/index` menampilkan teks "apa kabar!".

## 51.2. Penggunaan Template Web

Berikut contoh penggunaan template untuk menampilkan data dari html:

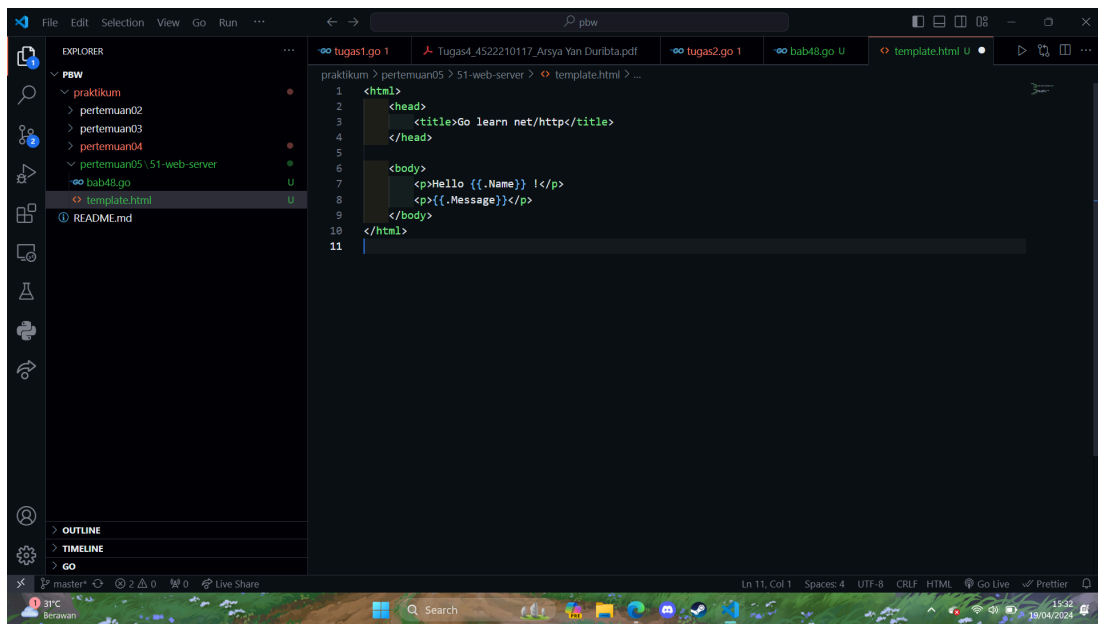
Implementasinya (Source Code):

- bab48.go:



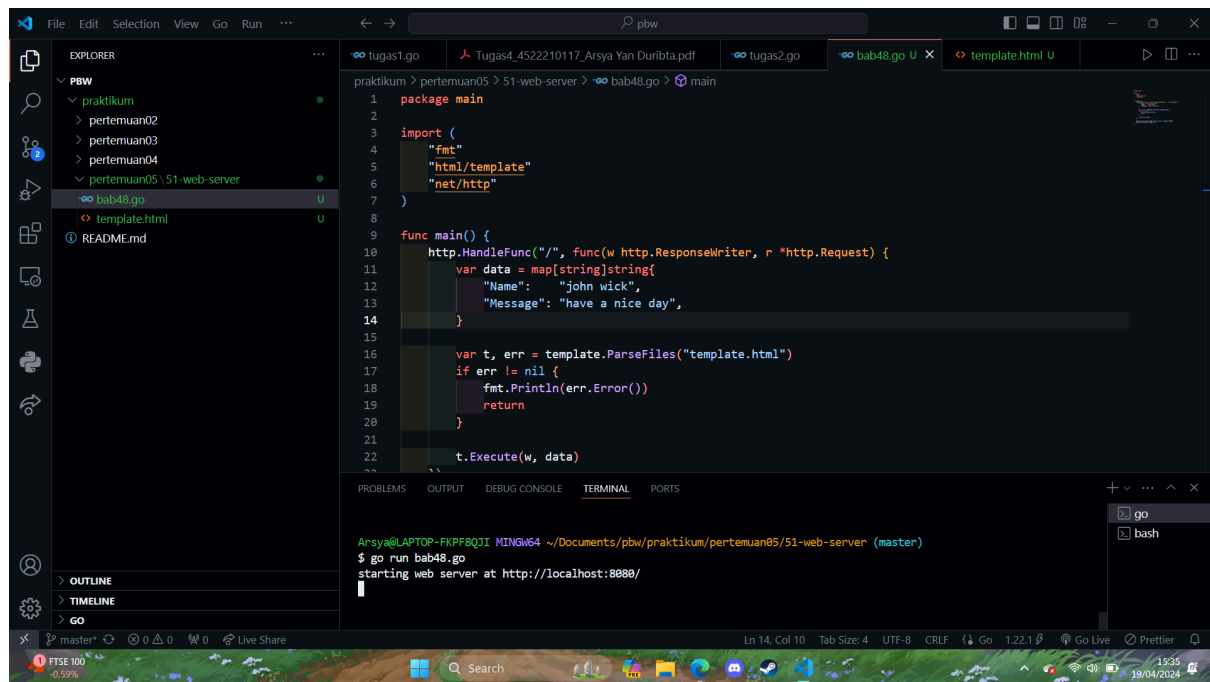
```
1 package main
2
3 import (
4     "fmt"
5     "html/template"
6     "net/http"
7 )
8
9 func main() {
10     http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
11         var data = map[string]string{
12             "Name": "john wick",
13             "Message": "have a nice day",
14         }
15
16         var t, err = template.ParseFiles("template.html")
17         if err != nil {
18             fmt.Println(err.Error())
19             return
20         }
21
22         t.Execute(w, data)
23     })
24
25     fmt.Println("starting web server at http://localhost:8080/")
26     http.ListenAndServe(":8080", nil)
27 }
28
```

- template.html:



```
1 <html>
2 <head>
3 <title>Go learn net/http</title>
4 </head>
5
6 <body>
7 <p>Hello {{.Name}} !</p>
8 <p>{{.Message}}</p>
9 </body>
10 </html>
11
```

Run code menggunakan terminal:

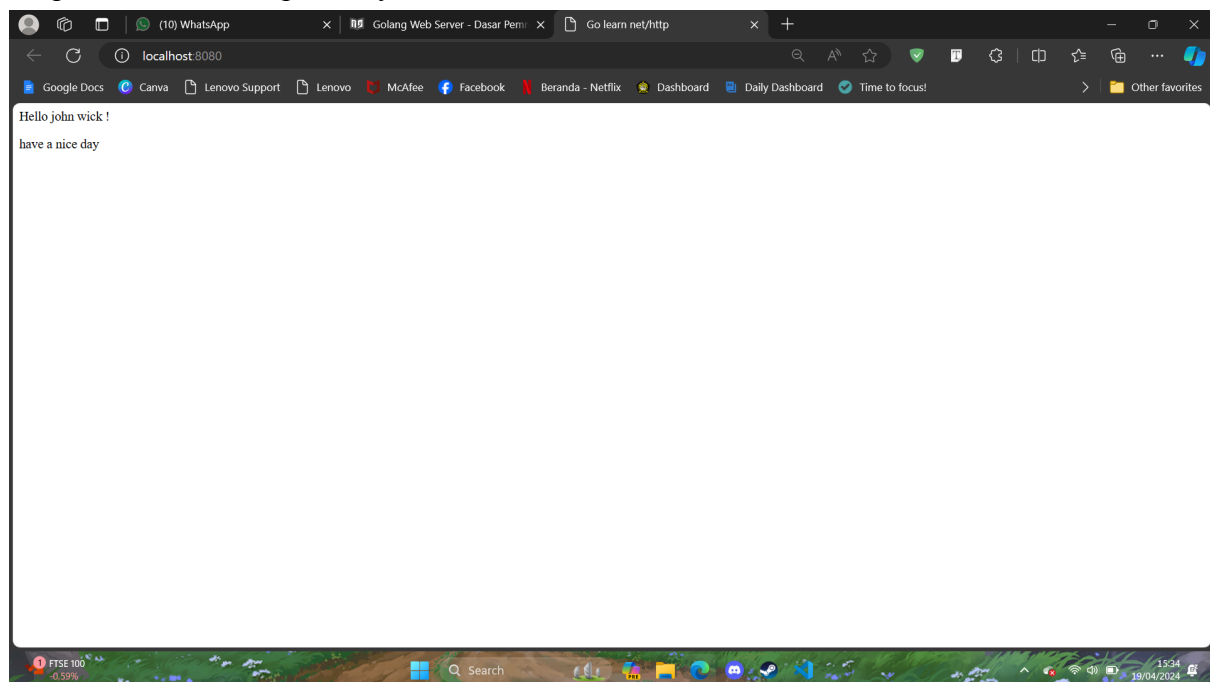


The screenshot shows a VS Code editor with a Go web server project. The Explorer panel on the left shows the project structure. The main editor displays the `tugas4.go` file, which contains a Go program using the `template` package. The program defines a `main` function that handles HTTP requests, parses a template, and executes it with data. The terminal at the bottom shows the command `$ go run bab48.go` being executed, resulting in the output `starting web server at http://localhost:8080/`.

```
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
22         t.Execute(w, data)
23     })
24 }
```

Arnya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw/praktikum/pertemuan05/51-web-server (master)  
\$ go run bab48.go  
starting web server at http://localhost:8080/

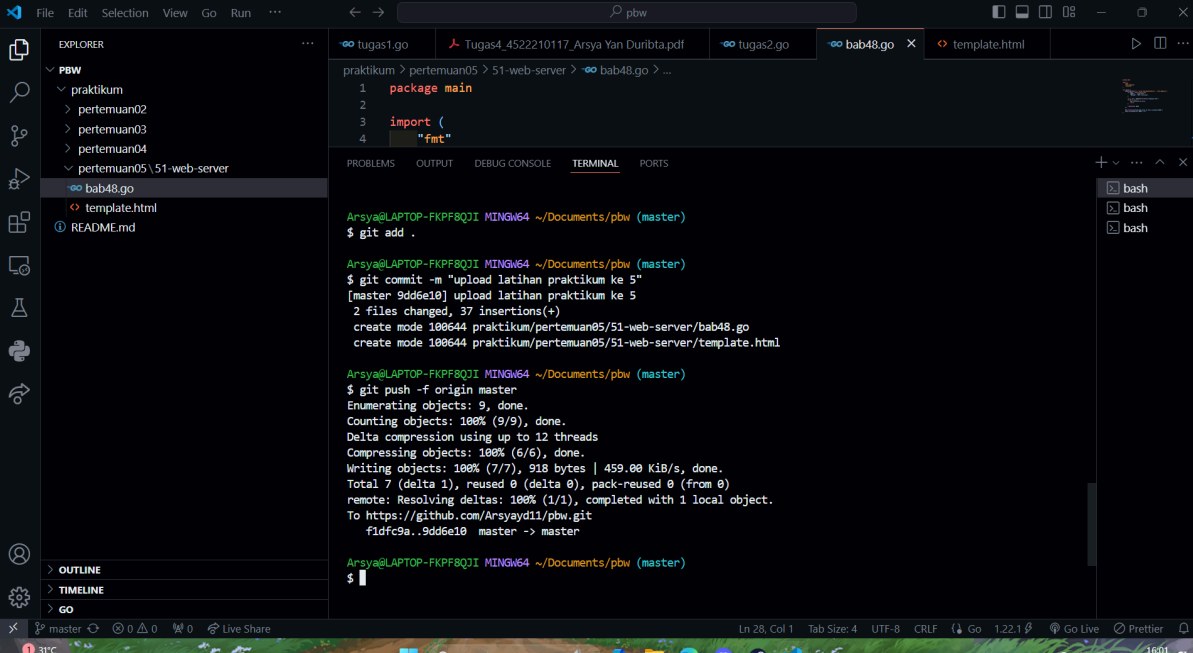
Output ketika menampilkannya di web browser:



Penjelasan:

Fungsi `template.ParseFiles()` digunakan untuk mem-parsing template. Pemanggilan method `Execute()` menghasilkan tampilan pada browser. Data disisipkan sebagai parameter kedua dalam method `Execute()`, dan bisa diakses di-view dengan menggunakan notasi `{{.Nama_Property}}`.

## Upload Ke Git



The screenshot shows a Visual Studio Code editor with a Go web server project. The Explorer panel on the left shows the project structure: praktikum > pertemuan05 > 51-web-server > bab48.go. The main editor shows the content of bab48.go, which is a Go package main with an import statement for "fmt". The terminal panel at the bottom shows the following commands and output:

```
Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw (master)
$ git add .

Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw (master)
$ git commit -m "upload latihan praktikum ke 5"
[master 9dd6e10] upload latihan praktikum ke 5
2 files changed, 37 insertions(+)
create mode 100644 praktikum/pertemuan05/51-web-server/bab48.go
create mode 100644 praktikum/pertemuan05/51-web-server/template.html

Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw (master)
$ git push -f origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 918 bytes | 459.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Arsyayd11/pbw.git
 f1dfc9a..9dd6e10 master -> master

Arsya@LAPTOP-FKPF8Q3I MINGW64 ~/Documents/pbw (master)
$
```

## Kesimpulan

Praktikum ini mengenalkan konsep dasar pembuatan aplikasi web menggunakan bahasa pemrograman Go. Mempelajari penggunaan package `net/http` untuk membuat web server, routing, dan templating. Dalam praktikum ini, berhasil membuat aplikasi web sederhana yang merespons permintaan dengan teks dan menggunakan template untuk menampilkan data. Penggunaan template engine dalam Go mempermudah desain tampilan aplikasi web.

## Link Repository Github:

<https://github.com/Arsyayd11/pbw/tree/master/praktikum/pertemuan05/51-web-server>