LAPORAN LATIHAN PRAKTIKUM KE-8

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

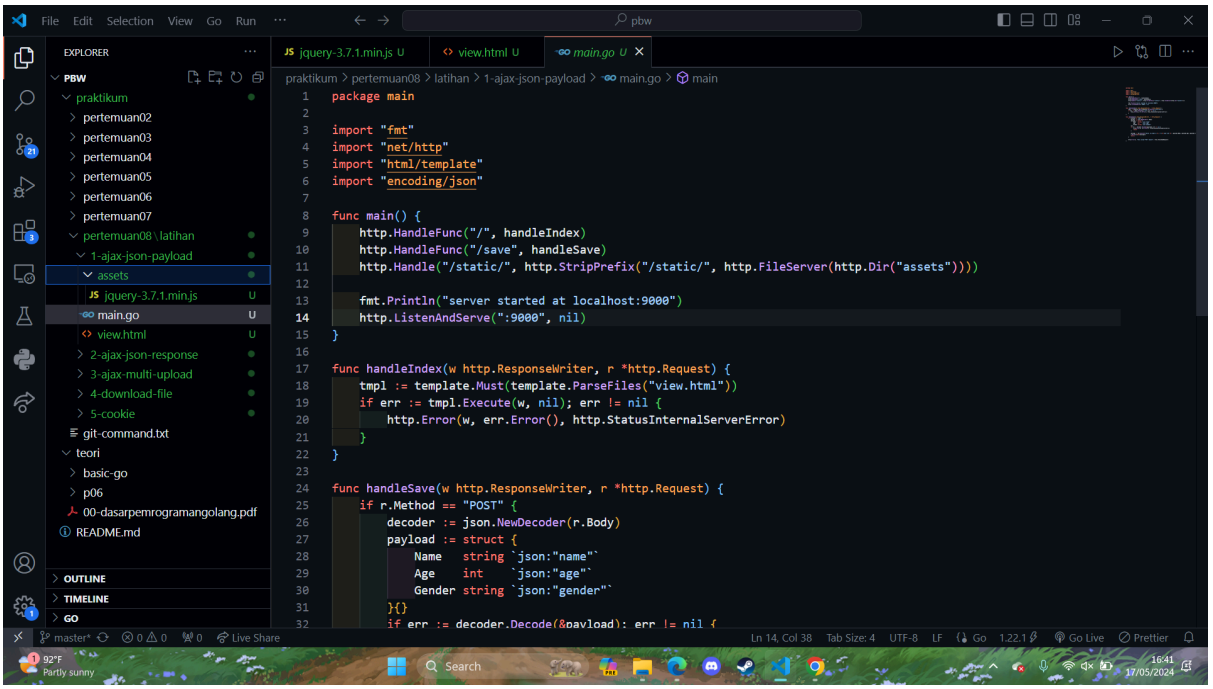
2023/2024

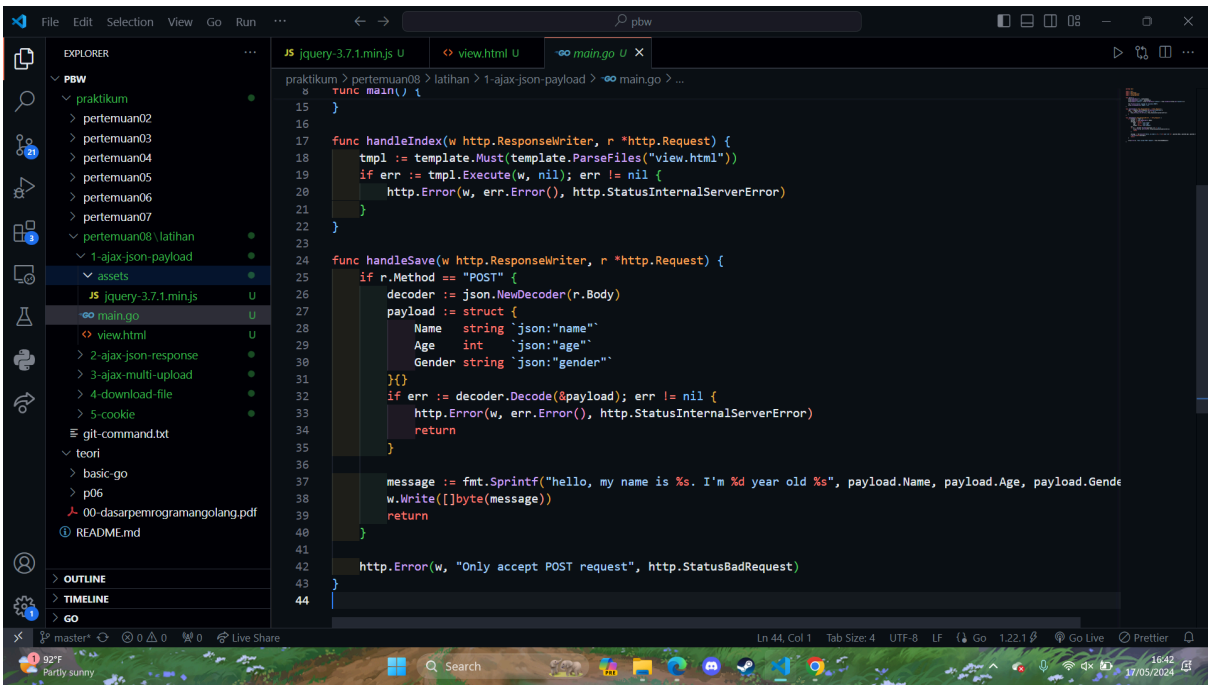
Link Repository Github:
https://github.com/Arsyayd11/pbw/tree/master/praktikum/pertemuan08/latihan

Latihan 1: 1-ajax-json-payload

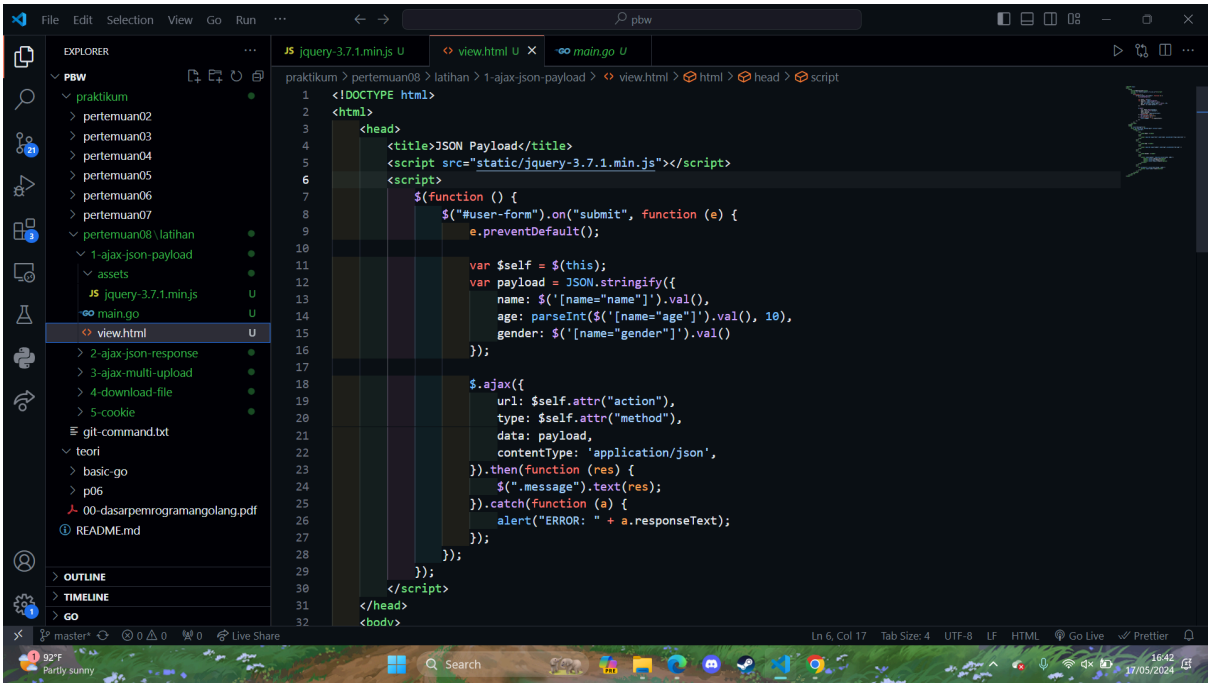
Implementasinya (Source Code):

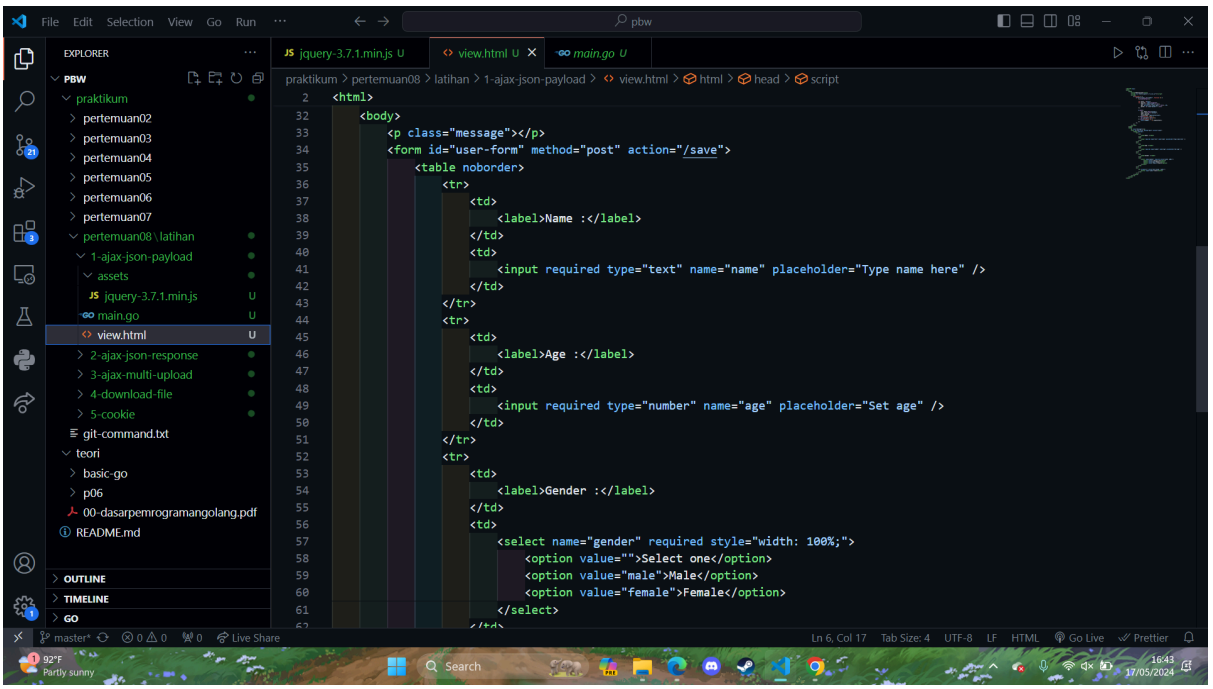
- main.go



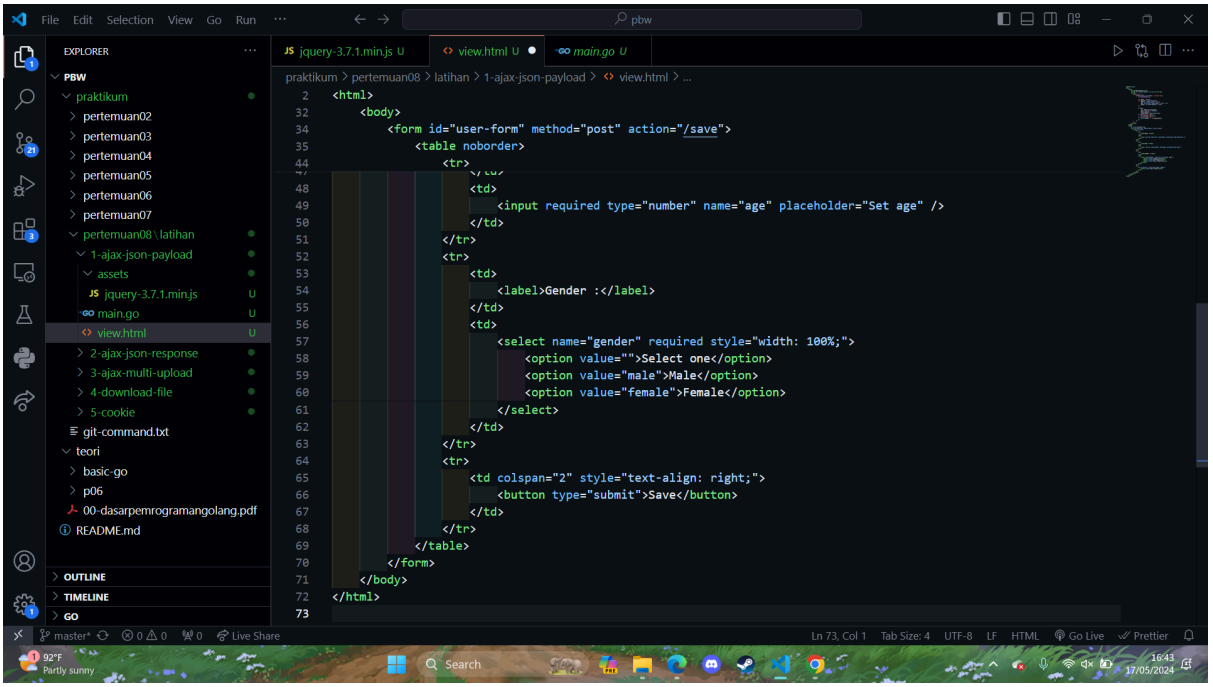


- view.html

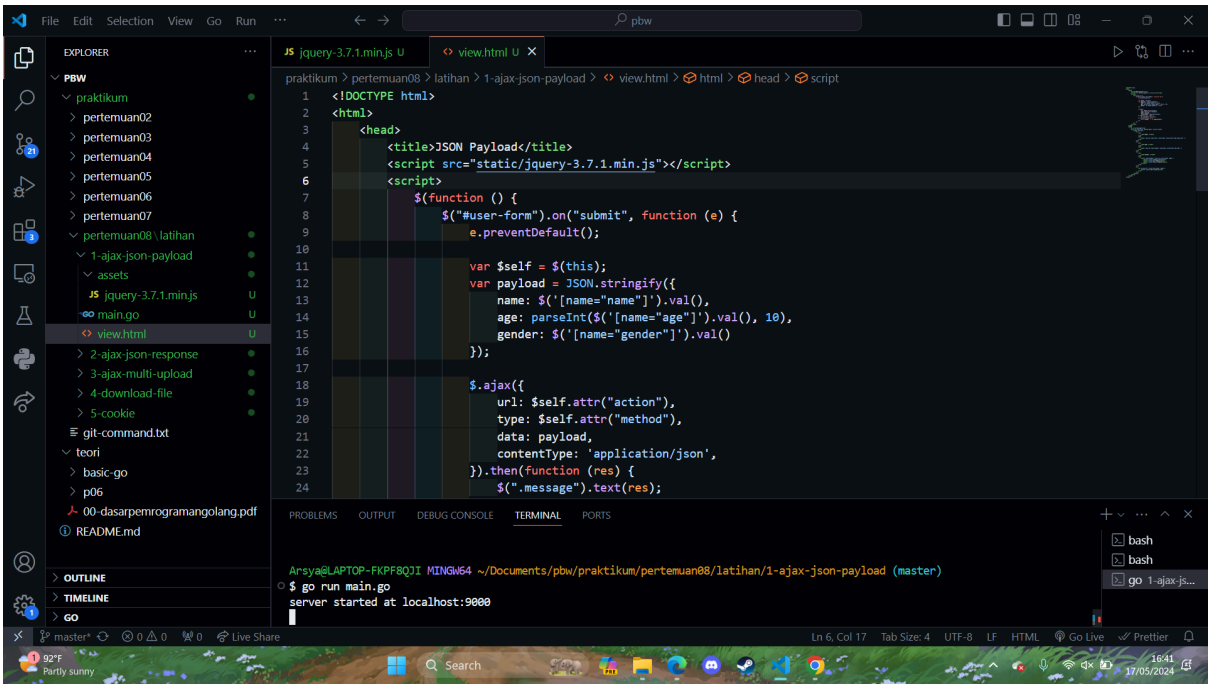

File Edit Selection View Go Run
pbw
EXPLORER
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
assets
jquery-3.7.1.min.js U
main.go U
view.html U
2-ajax-json-response
3-ajax-multi-upload
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
jquery-3.7.1.min.js U
view.html U
main.go U
praktikum > pertemuan08 > latihan > 1-ajax-json-payload > view.html > html > head > script
<!DOCTYPE html>
<html>
<head>
<title>JSON Payload</title>
<script src="static/jquery-3.7.1.min.js"></script>
<script>
$(function () {
$("#user-form").on("submit", function (e) {
e.preventDefault();
var $self = $(this);
var payload = JSON.stringify({
name: $('[name="name"]').val(),
age: parseInt($('[name="age"]').val(), 10),
gender: $('[name="gender"]').val()
});
$.ajax({
url: $self.attr("action"),
type: $self.attr("method"),
data: payload,
contentType: 'application/json',
}).then(function (res) {
$(".message").text(res);
}).catch(function (a) {
alert("ERROR: " + a.responseText);
});
});
});
</script>
</head>
<body>
master*
Live Share
Ln 6, Col 17
Tab Size: 4
UTF-8
LF
HTML
Go Live
Prettier
92°F
Partly sunny
Search
16:42
17/05/2024


File Edit Selection View Go Run
pbw
EXPLORER
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
assets
jquery-3.7.1.min.js U
main.go U
view.html U
2-ajax-json-response
3-ajax-multi-upload
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
jquery-3.7.1.min.js U
view.html U
main.go U
praktikum > pertemuan08 > latihan > 1-ajax-json-payload > view.html > html > head > script
<html>
<body>

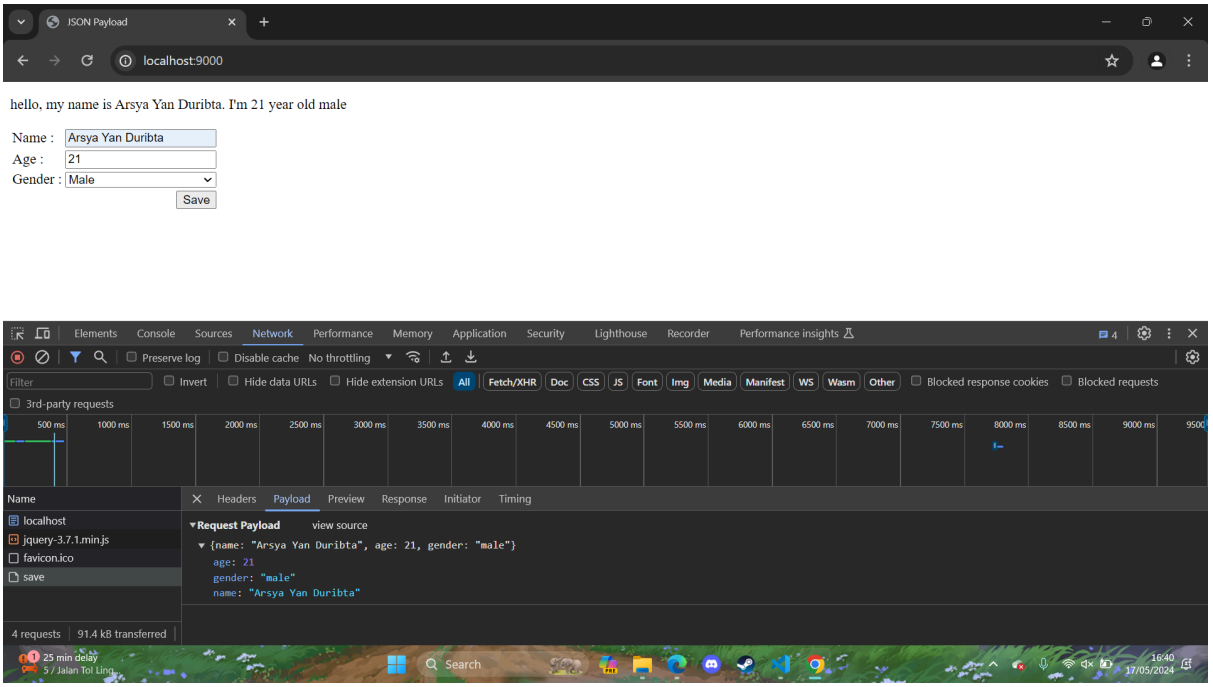
<form id="user-form" method="post" action="/save">
<table noborder>
<tr>
<td>
<label>Name :</label>
</td>
<td>
<input required type="text" name="name" placeholder="Type name here" />
</td>
</tr>
<tr>
<td>
<label>Age :</label>
</td>
<td>
<input required type="number" name="age" placeholder="Set age" />
</td>
</tr>
<tr>
<td>
<label>Gender :</label>
</td>
<td>
<select name="gender" required style="width: 100%;">
<option value="">Select one</option>
<option value="male">Male</option>
<option value="female">Female</option>
</select>
master*
Live Share
Ln 6, Col 17
Tab Size: 4
UTF-8
LF
HTML
Go Live
Prettier
92°F
Partly sunny
Search
16:43
17/05/2024



Run code menggunakan terminal:

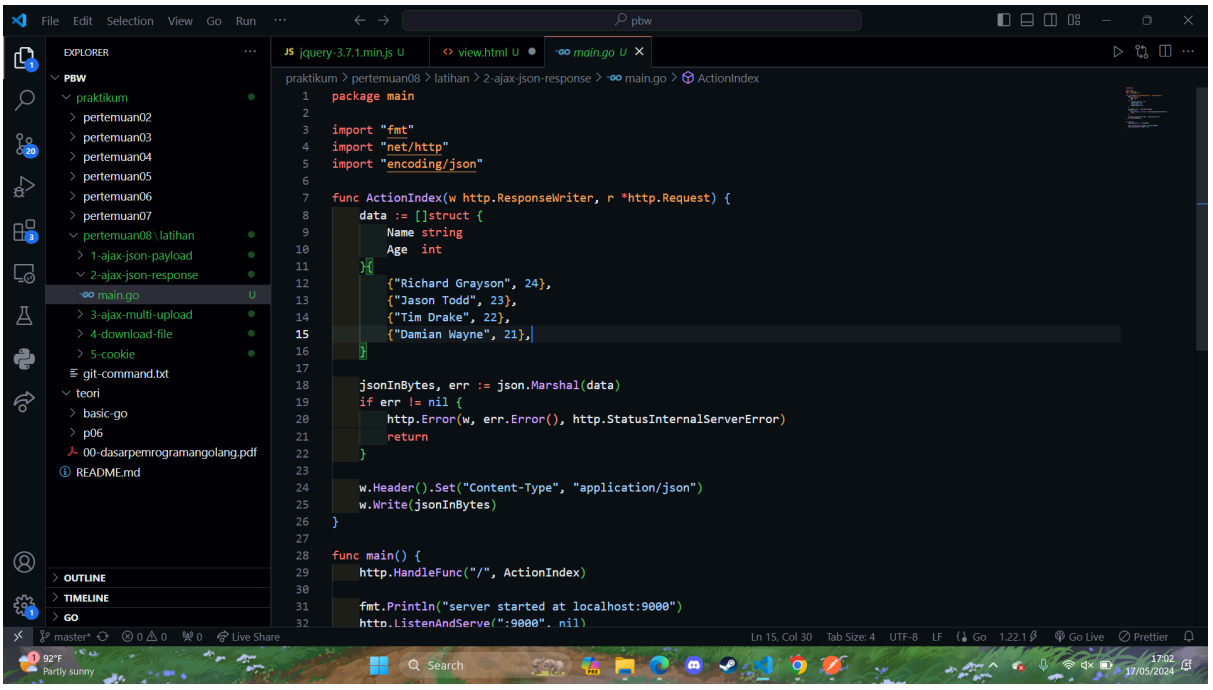


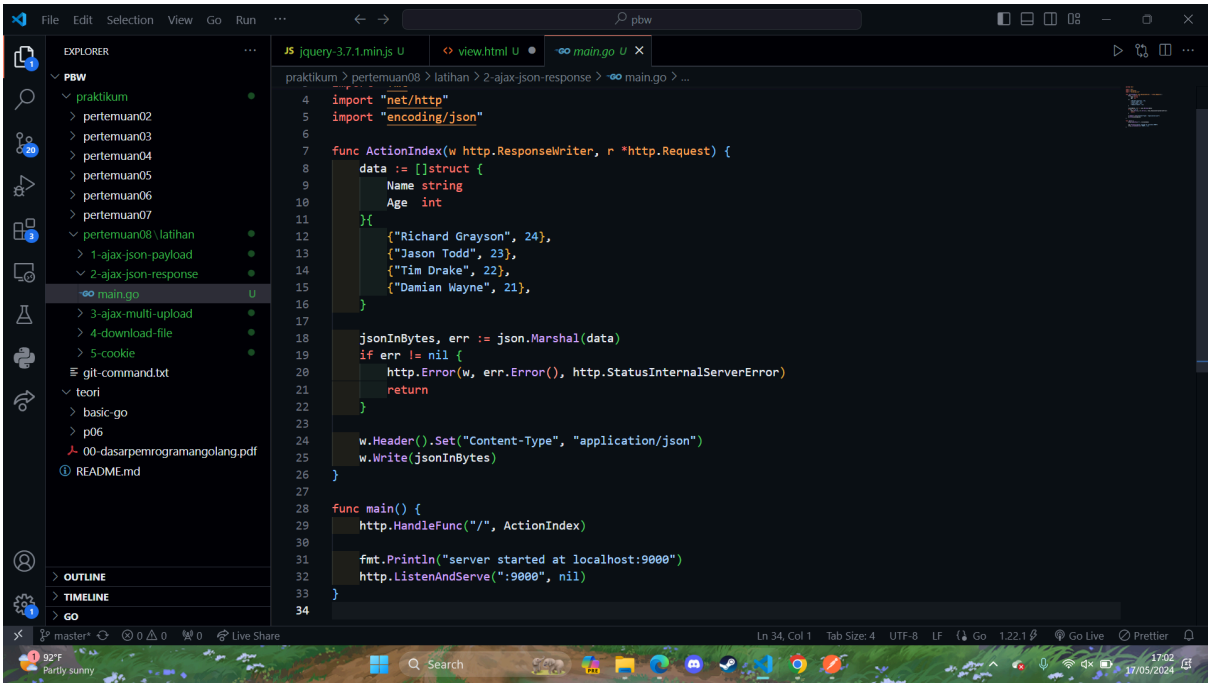
Tampilan localhost pada website:



Latihan 2: 2-ajax-json-response

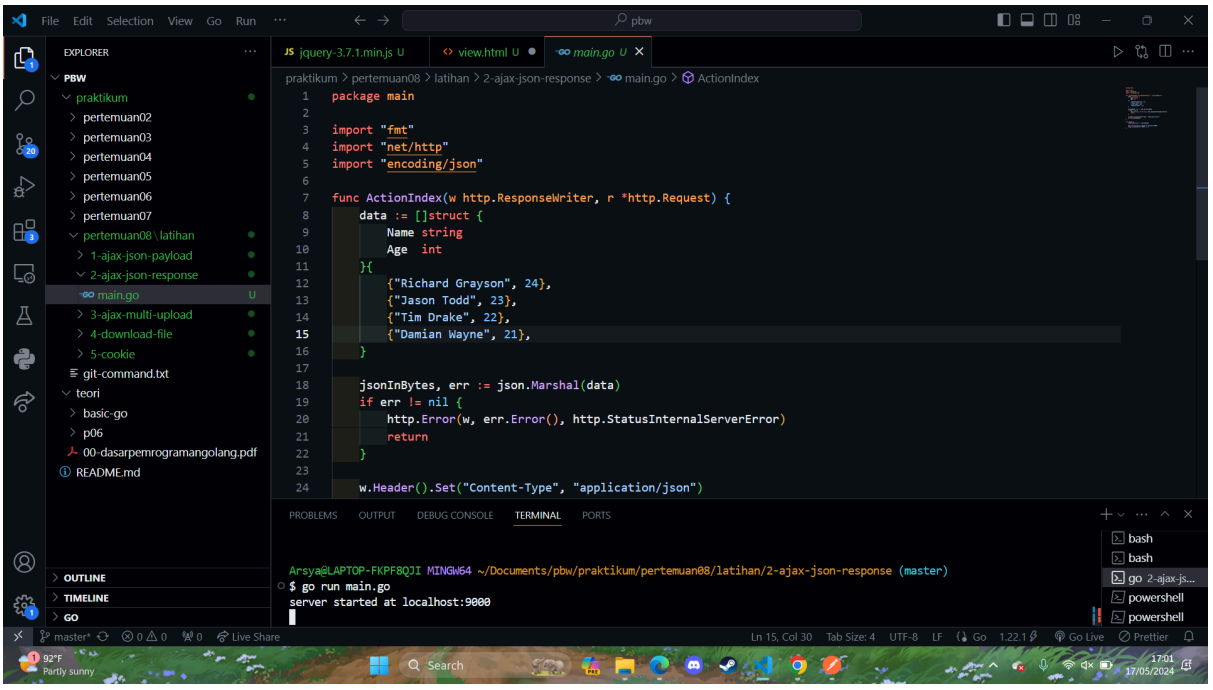
Implementasinya (Source Code):





```
import "net/http"
import "encoding/json"

func ActionIndex(w http.ResponseWriter, r *http.Request) {
    data := []struct {
        Name string
        Age  int
    }{
        {"Richard Grayson", 24},
        {"Jason Todd", 23},
        {"Tim Drake", 22},
        {"Damian Wayne", 21},
    }

    jsonInBytes, err := json.Marshal(data)
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    w.Header().Set("Content-Type", "application/json")
    w.Write(jsonInBytes)
}

func main() {
    http.HandleFunc("/", ActionIndex)

    fmt.Println("server started at localhost:9000")
    http.ListenAndServe(":9000", nil)
}
```

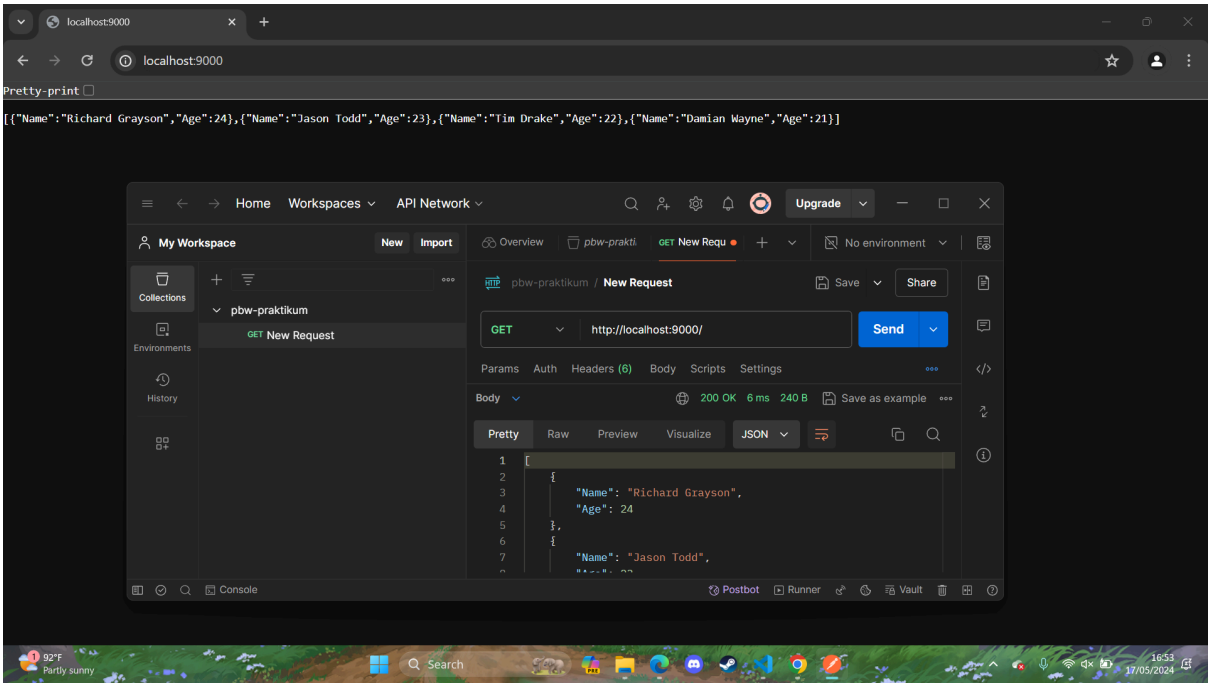
Run code menggunakan terminal:

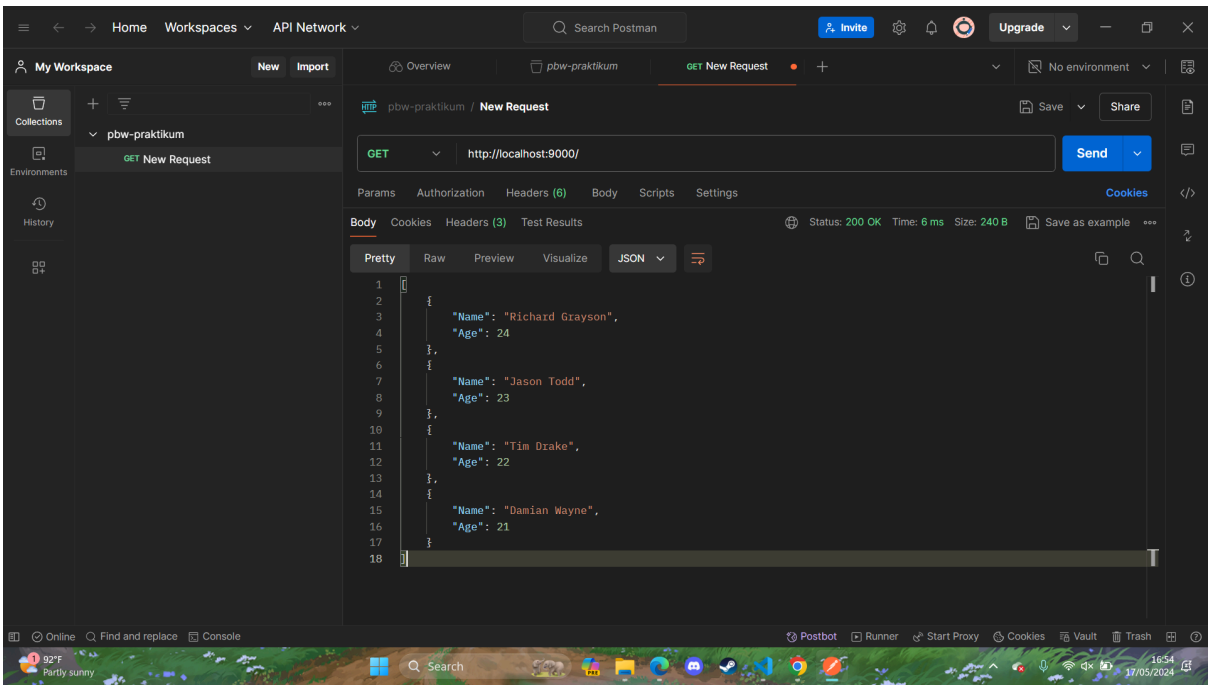


```
package main

import "fmt"
import "net/http"
import "encoding/json"

func ActionIndex(w http.ResponseWriter, r *http.Request) {
    data := []struct {
        Name string
        Age  int
    }{
        {"Richard Grayson", 24},
        {"Jason Todd", 23},
        {"Tim Drake", 22},
        {"Damian Wayne", 21},
    }

    jsonInBytes, err := json.Marshal(data)
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    w.Header().Set("Content-Type", "application/json")
```

```
Arsya@LAPTOP-FKPF8QJI MINGW64 ~/Documents/pbw/praktikum/pertemuan08/latihan/2-ajax-json-response (master)
$ go run main.go
server started at localhost:9000
```

Tampilan localhost pada website:

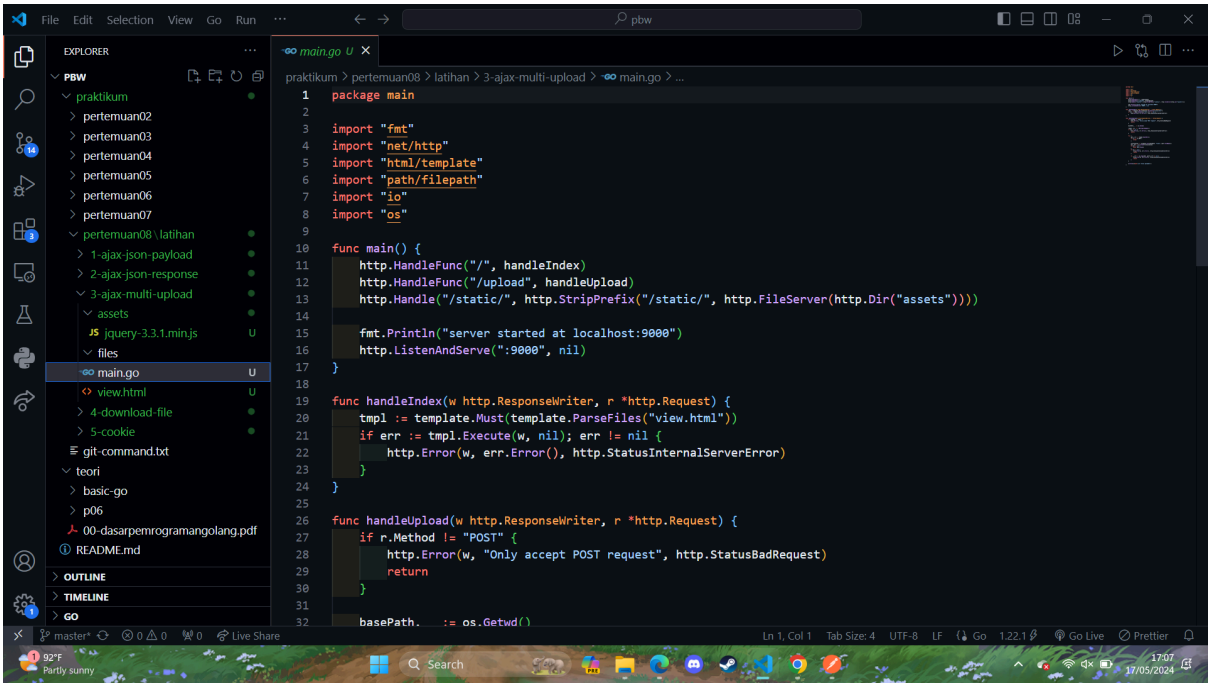


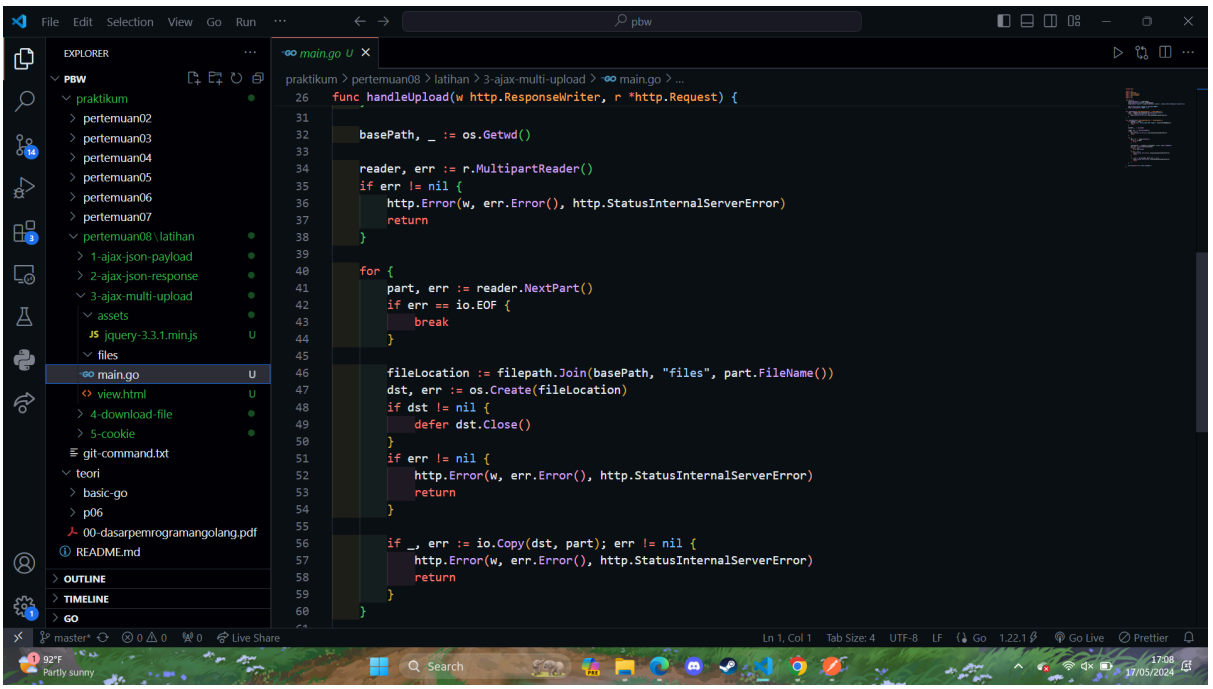


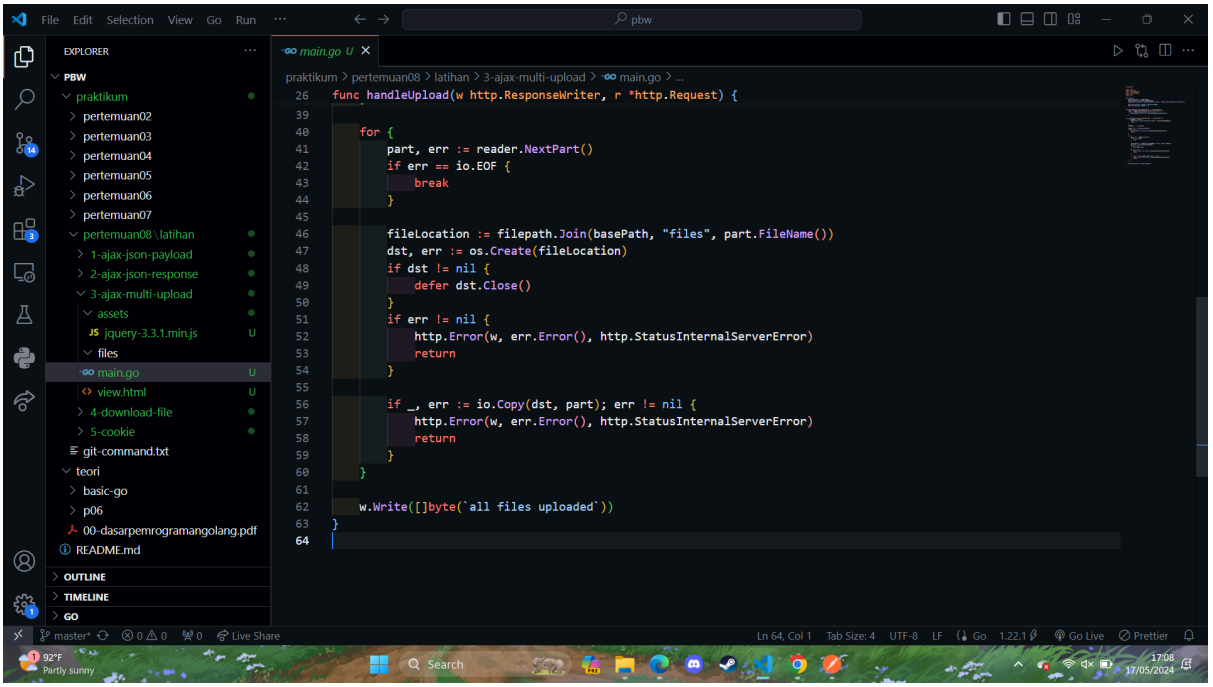
Latihan 3: 3-ajax-multi-upload

Implementasinya (Source Code):

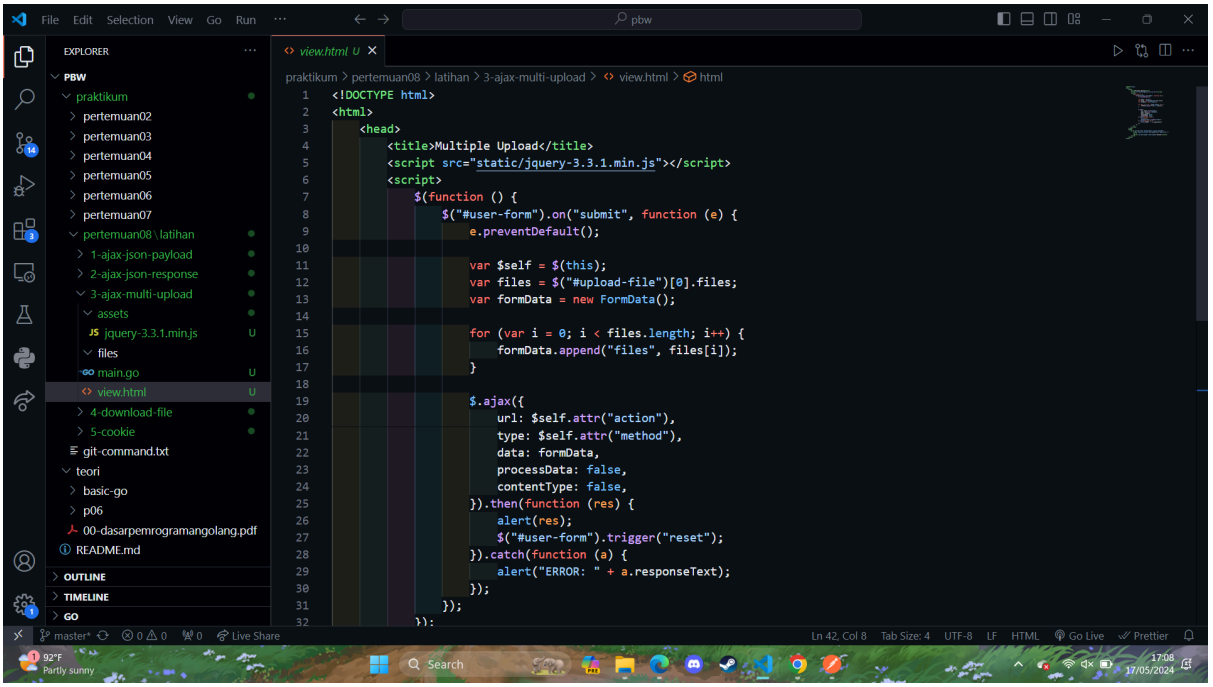
- main.go

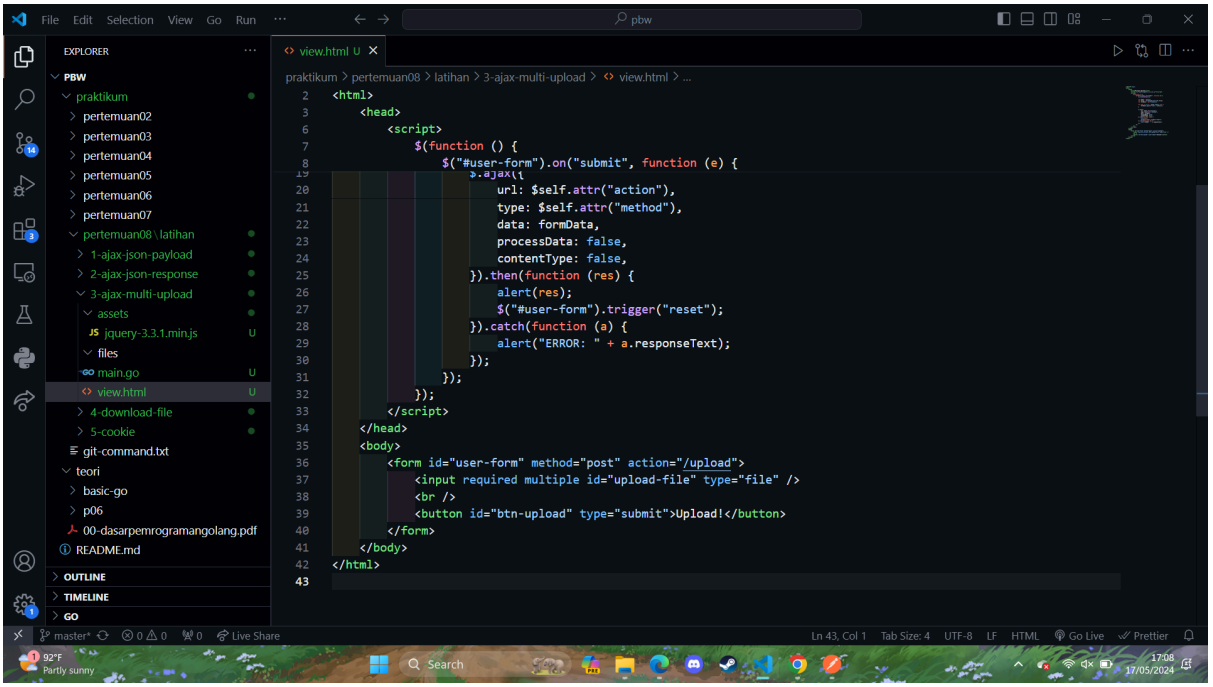

File Edit Selection View Go Run
pbw
EXPLORER
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
2-ajax-json-response
3-ajax-multi-upload
assets
jquery-3.3.1.min.js
files
main.go
view.html
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
main.go U
praktikum > pertemuan08 > latihan > 3-ajax-multi-upload > main.go > ...
package main
import "fmt"
import "net/http"
import "html/template"
import "path/filepath"
import "io"
import "os"
func main() {
http.HandleFunc("/", handleIndex)
http.HandleFunc("/upload", handleUpload)
http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("assets"))))
fmt.Println("server started at localhost:9000")
http.ListenAndServe(":9000", nil)
}
func handleIndex(w http.ResponseWriter, r *http.Request) {
tmpl := template.Must(template.ParseFiles("view.html"))
if err := tmpl.Execute(w, nil); err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
}
}
func handleUpload(w http.ResponseWriter, r *http.Request) {
if r.Method != "POST" {
http.Error(w, "Only accept POST request", http.StatusBadRequest)
return
}
basePath, := os.Getwd()
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Go 1.22.1 Go Live Prettier
master* Live Share
92°F Partly sunny
Search
17:07 17/05/2024


File Edit Selection View Go Run
pbw
EXPLORER
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
2-ajax-json-response
3-ajax-multi-upload
assets
jquery-3.3.1.min.js
files
main.go
view.html
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
main.go U
praktikum > pertemuan08 > latihan > 3-ajax-multi-upload > main.go > ...
func handleUpload(w http.ResponseWriter, r *http.Request) {
basePath, _ := os.Getwd()
reader, err := r.MultipartReader()
if err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
return
}
for {
part, err := reader.NextPart()
if err == io.EOF {
break
}
fileLocation := filepath.Join(basePath, "files", part.FileName())
dst, err := os.Create(fileLocation)
if dst != nil {
defer dst.Close()
}
if err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
return
}
if _, err := io.Copy(dst, part); err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
return
}
}
Ln 1, Col 1 Tab Size: 4 UTF-8 LF Go 1.22.1 Go Live Prettier
master* Live Share
92°F Partly sunny
Search
17:08 17/05/2024

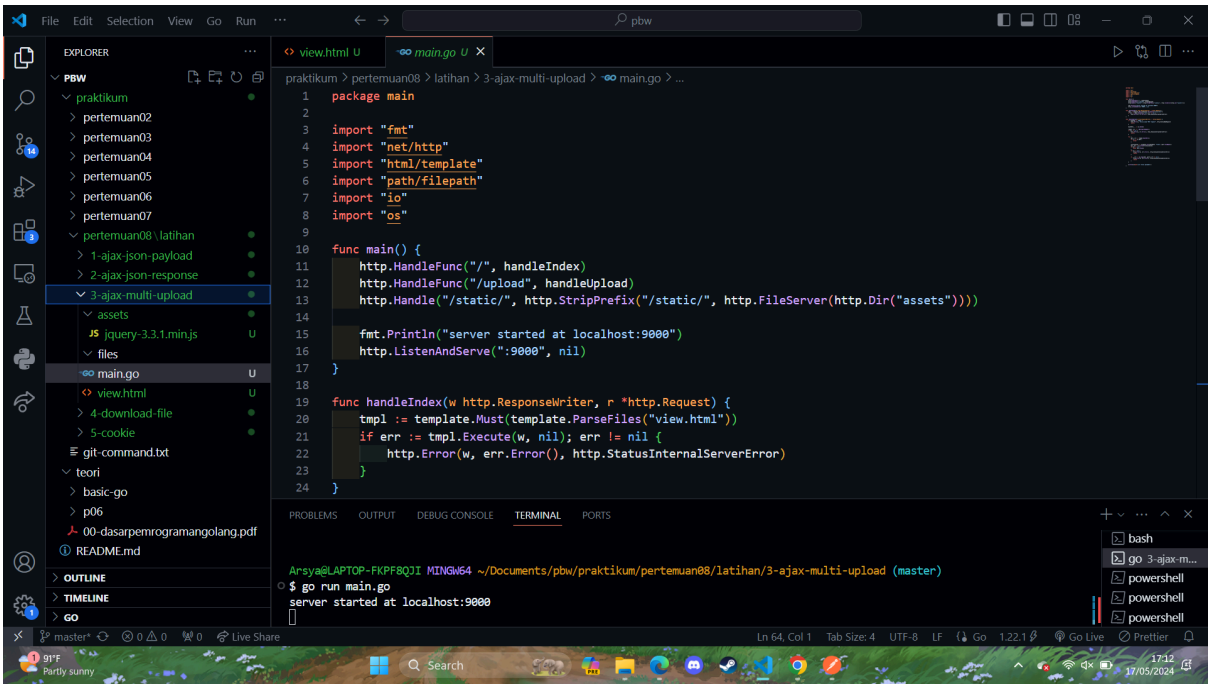

File Edit Selection View Go Run
pbw
EXPLORER
main.go U
praktikum > pertemuan08 > latihan > 3-ajax-multi-upload > main.go > ...
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
2-ajax-json-response
3-ajax-multi-upload
assets
jquery-3.3.1.min.js
files
main.go
view.html
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
func handleUpload(w http.ResponseWriter, r *http.Request) {
for {
part, err := reader.NextPart()
if err == io.EOF {
break
}
fileLocation := filepath.Join(basePath, "files", part.FileName())
dst, err := os.Create(fileLocation)
if dst != nil {
defer dst.Close()
}
if err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
return
}
if _, err := io.Copy(dst, part); err != nil {
http.Error(w, err.Error(), http.StatusInternalServerError)
return
}
}
w.Write([]byte(`all files uploaded`))
}
master*
Live Share
Ln 64, Col 1
Tab Size: 4
UTF-8
LF
Go
1.22.1
Go Live
Prettier
92°F
Partly sunny
Search
17:08
17/05/2024

- view.html


File Edit Selection View Go Run
pbw
EXPLORER
view.html U
praktikum > pertemuan08 > latihan > 3-ajax-multi-upload > view.html > html
PBW
praktikum
pertemuan02
pertemuan03
pertemuan04
pertemuan05
pertemuan06
pertemuan07
pertemuan08 \ latihan
1-ajax-json-payload
2-ajax-json-response
3-ajax-multi-upload
assets
jquery-3.3.1.min.js
files
main.go
view.html
4-download-file
5-cookie
git-command.txt
teori
basic-go
p06
00-dasarpemrogramangolang.pdf
README.md
OUTLINE
TIMELINE
GO
<!DOCTYPE html>
<html>
<head>
<title>Multiple Upload</title>
<script src="static/jquery-3.3.1.min.js"></script>
<script>
$(function () {
$("#user-form").on("submit", function (e) {
e.preventDefault();
var $self = $(this);
var files = $("#upload-file")[0].files;
var formData = new FormData();
for (var i = 0; i < files.length; i++) {
formData.append("files", files[i]);
}
$.ajax({
url: $self.attr("action"),
type: $self.attr("method"),
data: formData,
processData: false,
contentType: false,
}).then(function (res) {
alert(res);
$("#user-form").trigger("reset");
}).catch(function (a) {
alert("ERROR: " + a.responseText);
});
});
master*
Live Share
Ln 42, Col 8
Tab Size: 4
UTF-8
LF
HTML
Go Live
Prettier
92°F
Partly sunny
Search
17:08
17/05/2024



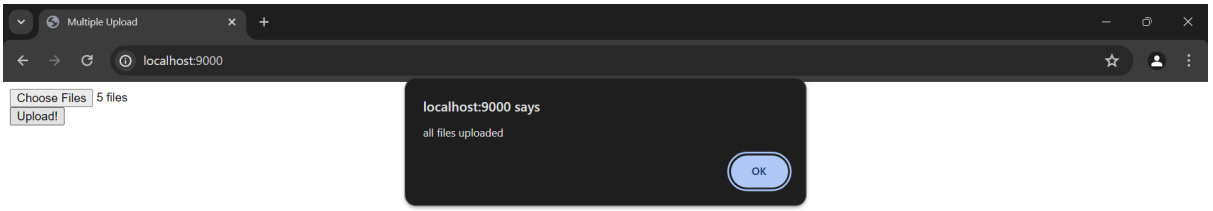
Run code menggunakan terminal:



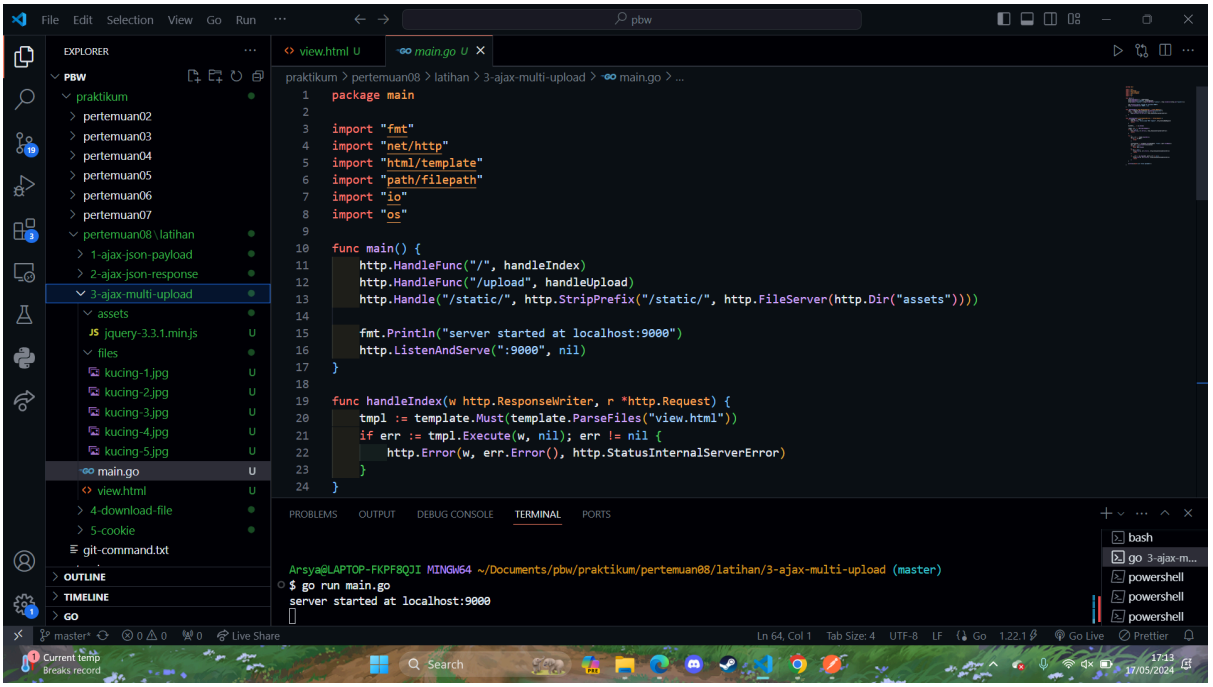
Tampilan localhost pada website:


Multiple Upload
localhost:9000
Choose Files
No file chosen
Upload!


Finance headline
JPMorgan Chas
Search
17:11
17/05/2024


Multiple Upload
localhost:9000
Choose Files
5 files
Upload!
localhost:9000 says
all files uploaded
OK


Current temp
Breaks record
Search
17:13
17/05/2024

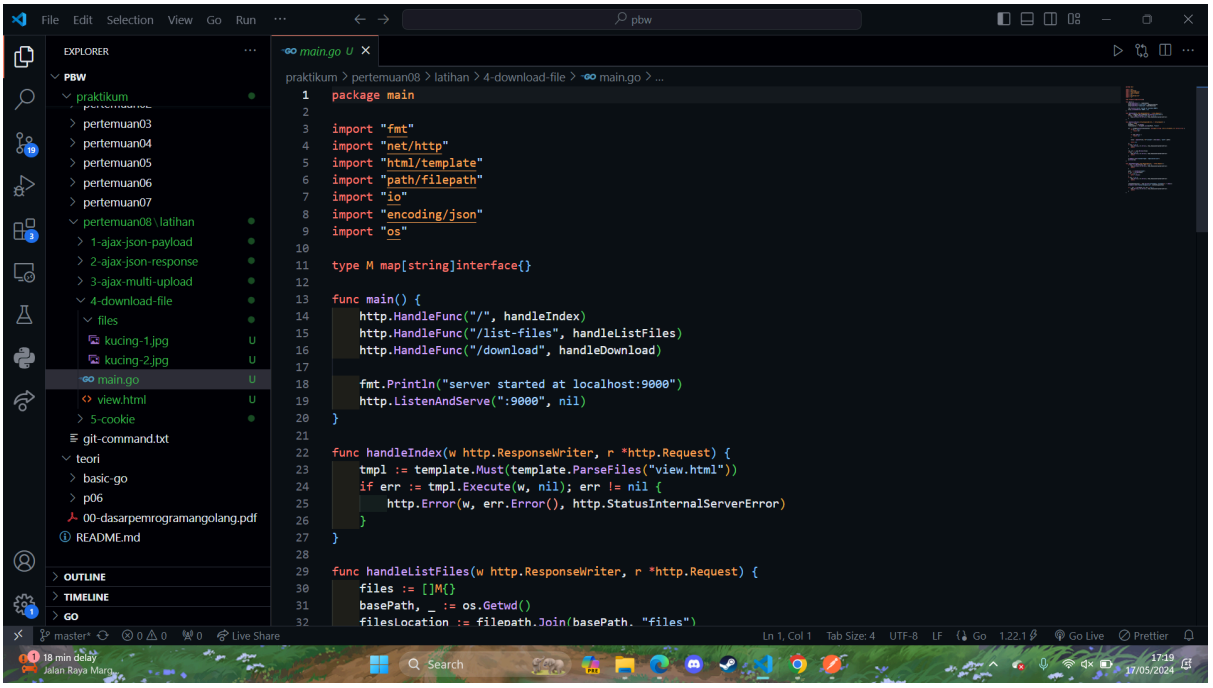



```go
package main

import "fmt"
import "net/http"
import "html/template"
import "path/filepath"
import "io"
import "os"

func main() {
    http.HandleFunc("/", handleIndex)
    http.HandleFunc("/upload", handleUpload)
    http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("assets"))))

    fmt.Println("server started at localhost:9000")
    http.ListenAndServe(":9000", nil)
}

func handleIndex(w http.ResponseWriter, r *http.Request) {
    tmpl := template.Must(template.ParseFiles("view.html"))
    if err := tmpl.Execute(w, nil); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
    }
}
```

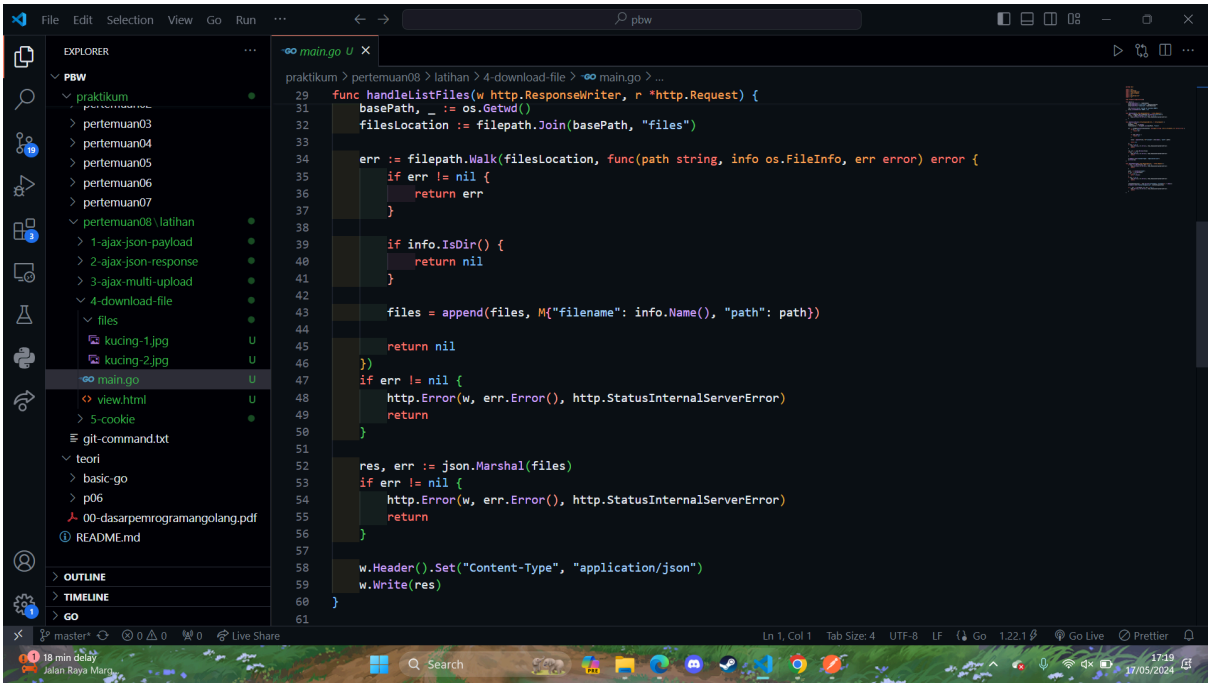
Latihan 4: 4-download-file

Implementasinya (Source Code):

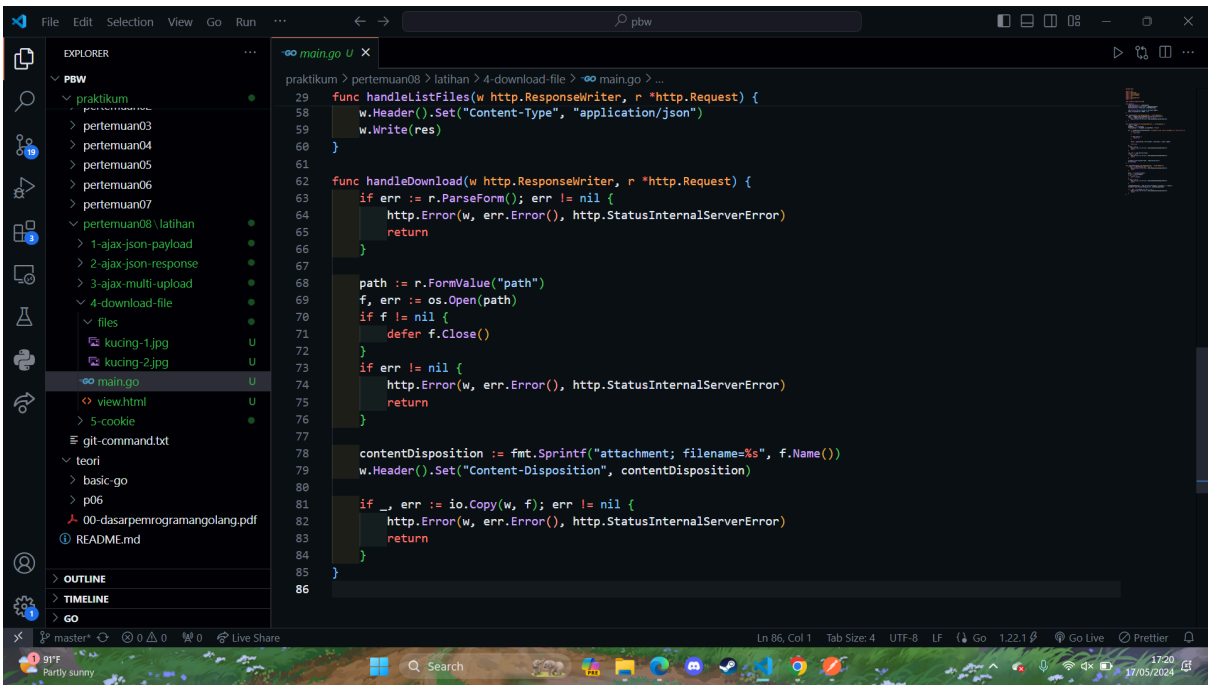
- main.go




```go
package main

import "fmt"
import "net/http"
import "html/template"
import "path/filepath"
import "io"
import "encoding/json"
import "os"

type M map[string]interface{}

func main() {
    http.HandleFunc("/", handleIndex)
    http.HandleFunc("/list-files", handleListFiles)
    http.HandleFunc("/download", handleDownload)

    fmt.Println("server started at localhost:9000")
    http.ListenAndServe(":9000", nil)
}

func handleIndex(w http.ResponseWriter, r *http.Request) {
    tmpl := template.Must(template.ParseFiles("view.html"))
    if err := tmpl.Execute(w, nil); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
    }
}

func handleListFiles(w http.ResponseWriter, r *http.Request) {
    files := []M{}
    basePath, _ := os.Getwd()
    filesLocation := filepath.Join(basePath, "files")
```

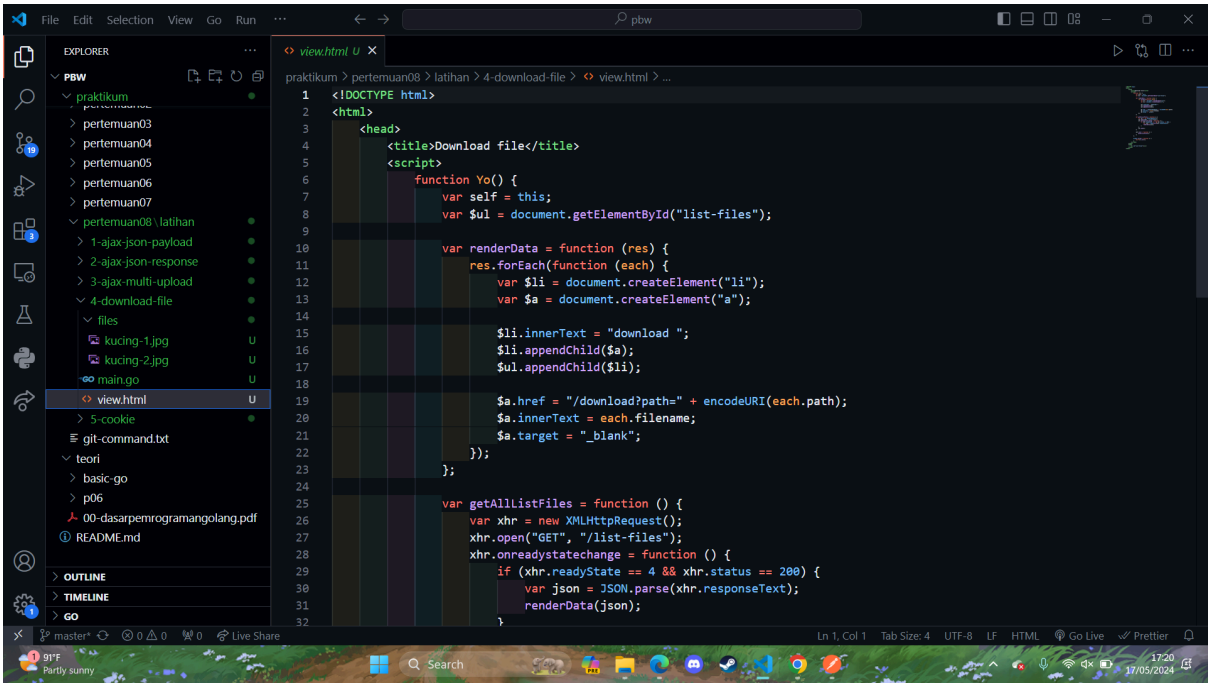


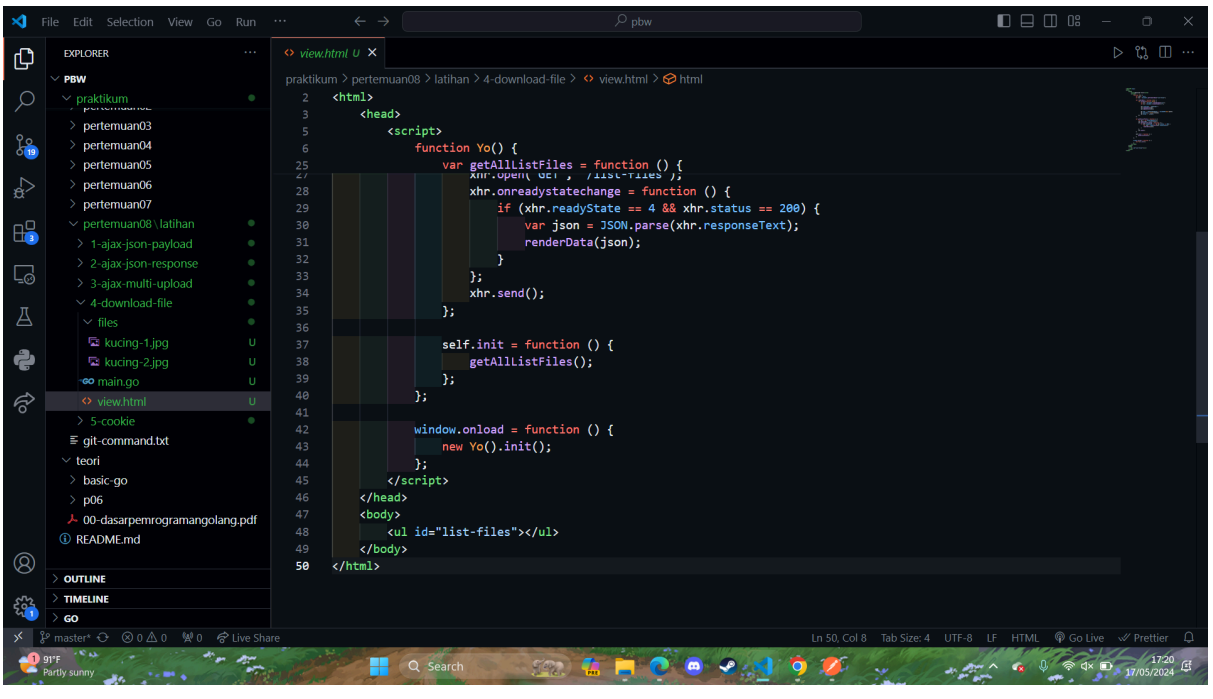
```
func handleListFiles(w http.ResponseWriter, r *http.Request) {
    basePath, _ := os.Getwd()
    filesLocation := filepath.Join(basePath, "files")

    err := filepath.Walk(filesLocation, func(path string, info os.FileInfo, err error) error {
        if err != nil {
            return err
        }

        if info.IsDir() {
            return nil
        }

        files = append(files, M{"filename": info.Name(), "path": path})

        return nil
    })
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    res, err := json.Marshal(files)
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    w.Header().Set("Content-Type", "application/json")
    w.Write(res)
}
```



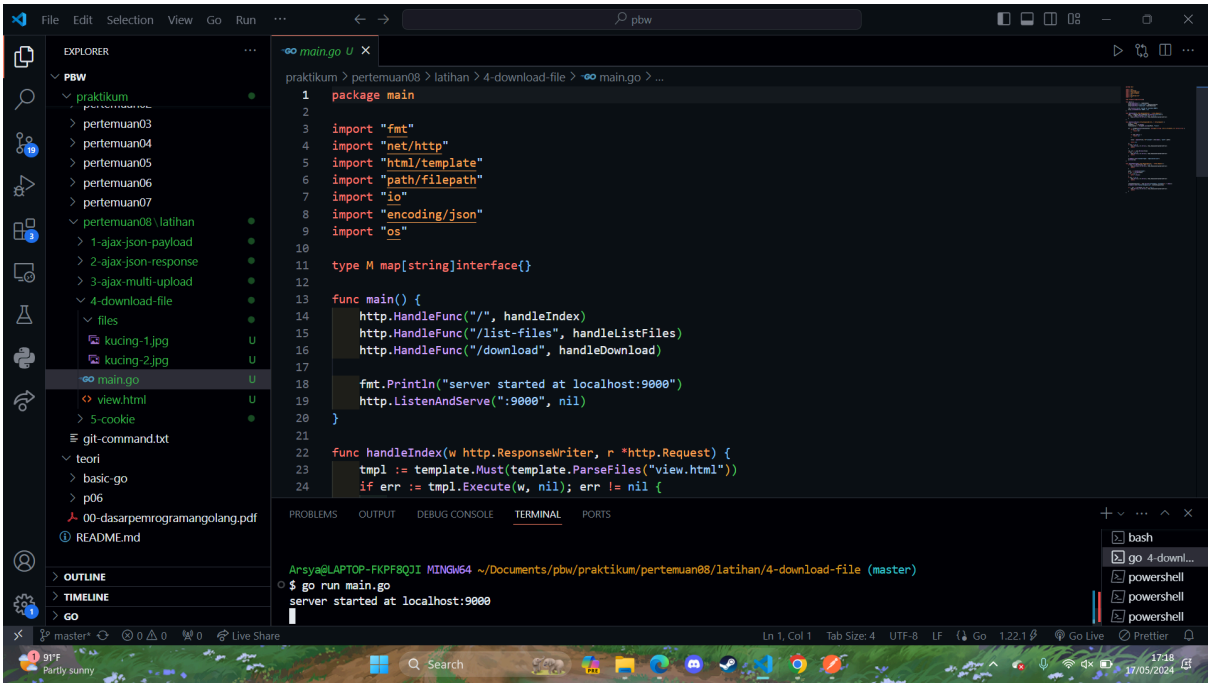
```
func handleListFiles(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-Type", "application/json")
    w.Write(res)
}

func handleDownload(w http.ResponseWriter, r *http.Request) {
    if err := r.ParseForm(); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    path := r.FormValue("path")
    f, err := os.Open(path)
    if f != nil {
        defer f.Close()
    }
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    contentDisposition := fmt.Sprintf("attachment; filename=%s", f.Name())
    w.Header().Set("Content-Disposition", contentDisposition)

    if _, err := io.Copy(w, f); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }
}
```

- view.html

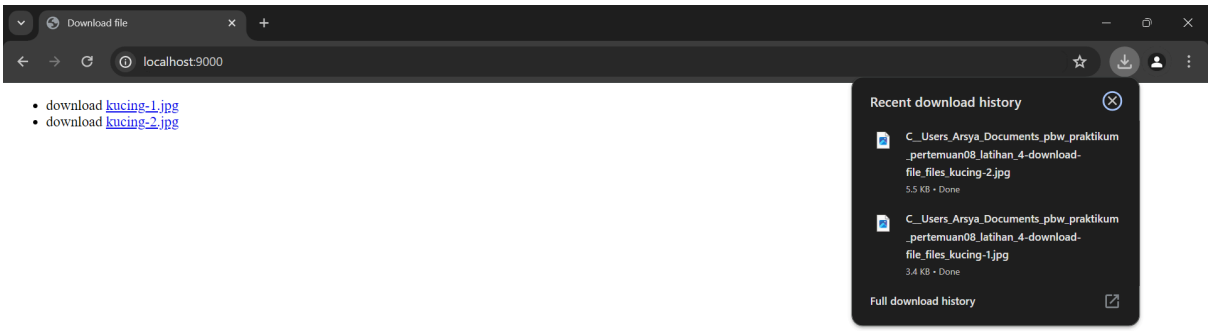




Run code menggunakan terminal:

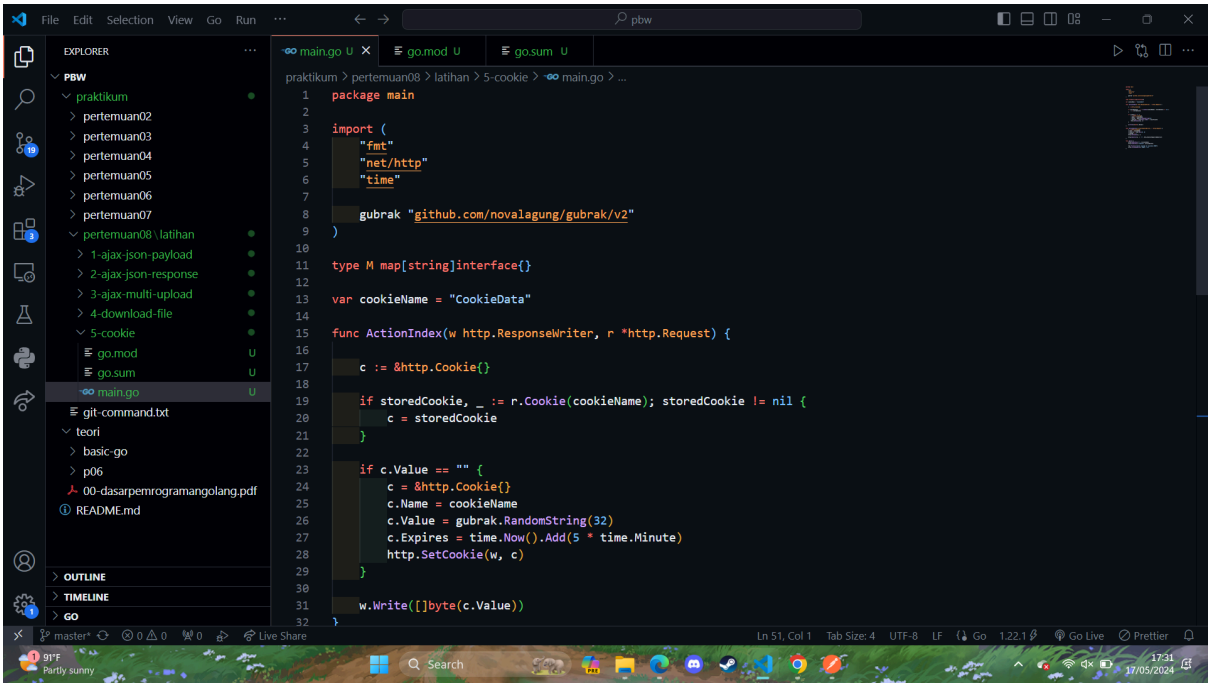


Tampilan localhost pada website:

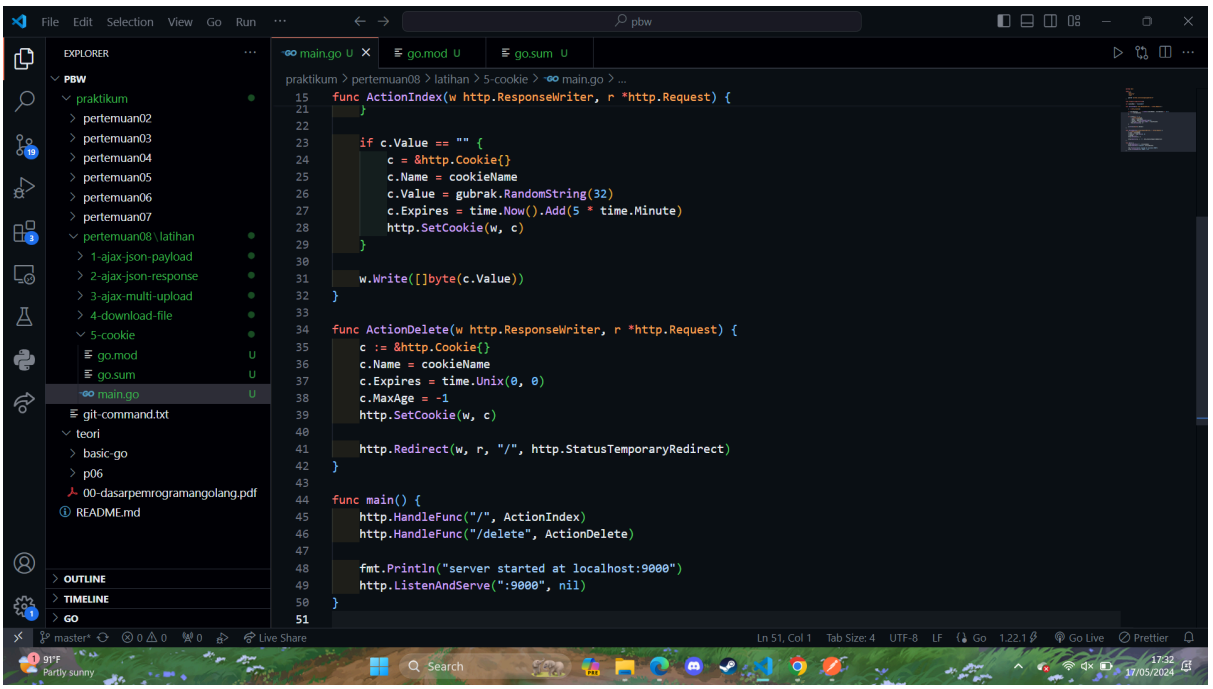


Latihan 5: 5-cookie

Implementasinya (Source Code):

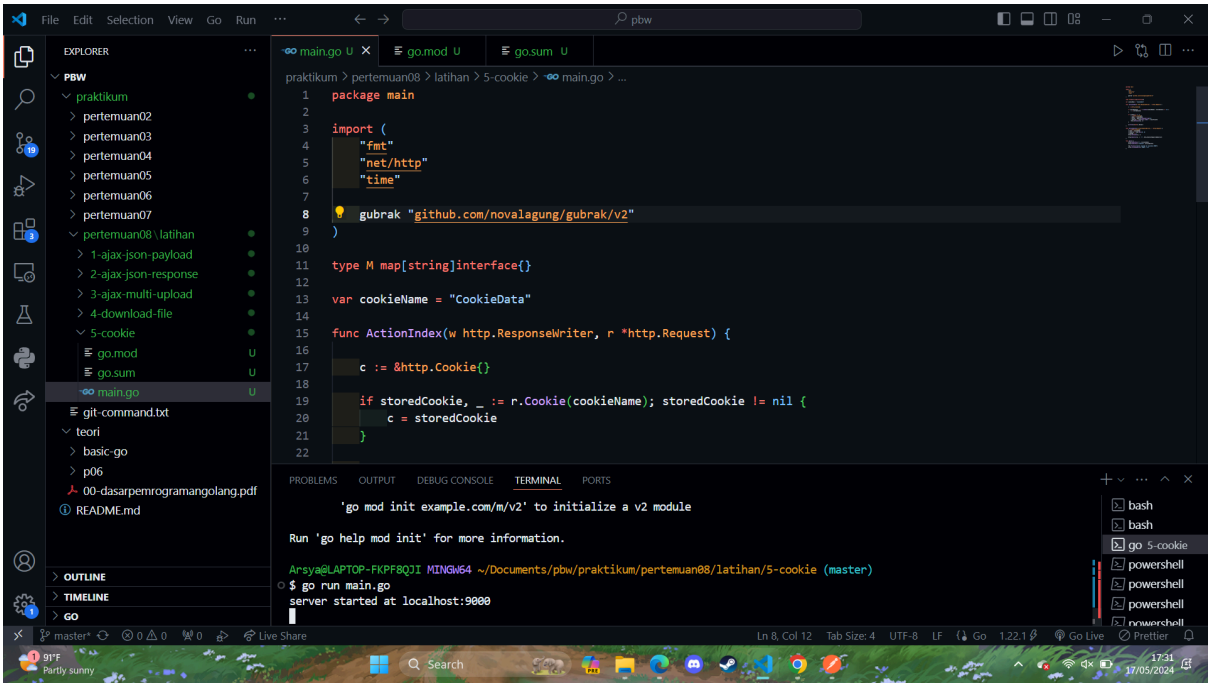


```
package main

import (
    "fmt"
    "net/http"
    "time"

    gubrak "github.com/novalagung/gubrak/v2"
)

type M map[string]interface{}

var cookieName = "CookieData"

func ActionIndex(w http.ResponseWriter, r *http.Request) {

    c := &http.Cookie{}

    if storedCookie, _ := r.Cookie(cookieName); storedCookie != nil {
        c = storedCookie
    }

    if c.Value == "" {
        c = &http.Cookie{}
        c.Name = cookieName
        c.Value = gubrak.RandomString(32)
        c.Expires = time.Now().Add(5 * time.Minute)
        http.SetCookie(w, c)
    }

    w.Write([]byte(c.Value))
}
```

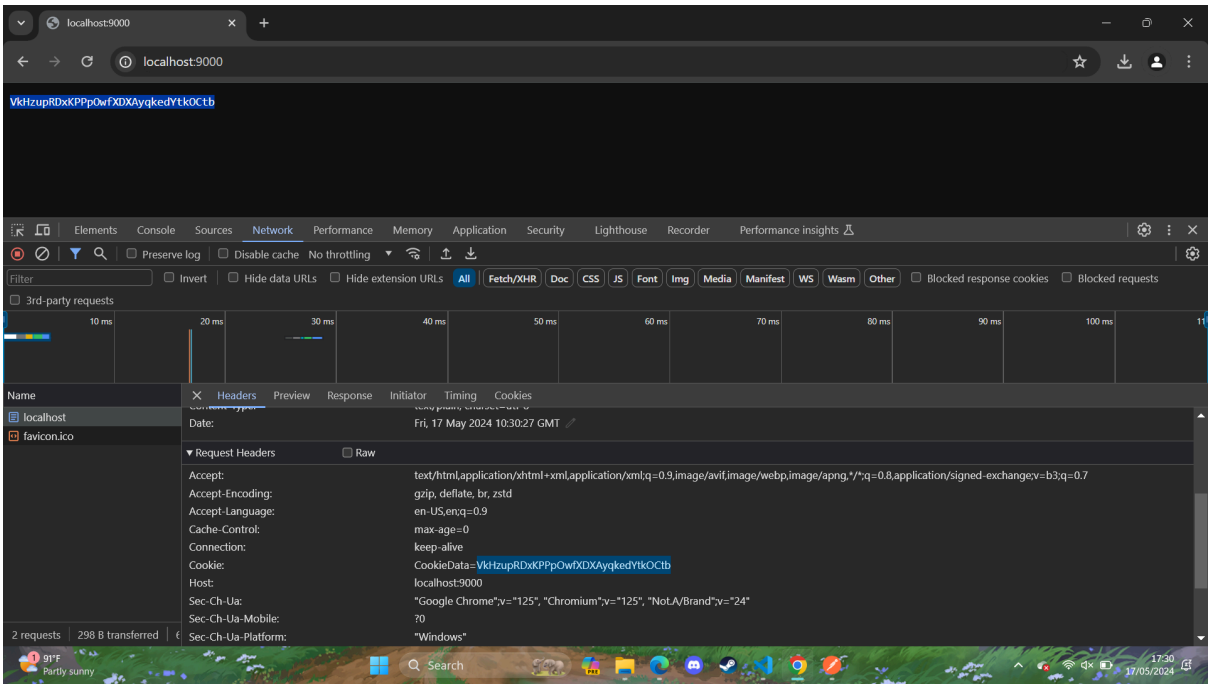


```
func ActionIndex(w http.ResponseWriter, r *http.Request) {
    }

    if c.Value == "" {
        c = &http.Cookie{}
        c.Name = cookieName
        c.Value = gubrak.RandomString(32)
        c.Expires = time.Now().Add(5 * time.Minute)
        http.SetCookie(w, c)
    }

    w.Write([]byte(c.Value))
}

func ActionDelete(w http.ResponseWriter, r *http.Request) {
    c := &http.Cookie{}
    c.Name = cookieName
    c.Expires = time.Unix(0, 0)
    c.MaxAge = -1
    http.SetCookie(w, c)

    http.Redirect(w, r, "/", http.StatusTemporaryRedirect)
}

func main() {
    http.HandleFunc("/", ActionIndex)
    http.HandleFunc("/delete", ActionDelete)

    fmt.Println("server started at localhost:9000")
    http.ListenAndServe(":9000", nil)
}
```

Run code menggunakan terminal:



Tampilan localhost pada website:



Pertanyaan:

a. Apa itu AJAX?

AJAX (Asynchronous JavaScript and XML) merupakan teknik yang digunakan untuk membuat permintaan HTTP ke server dan menerima respon dari server tanpa harus merefresh seluruh halaman web. Teknik ini digunakan aplikasi web untuk memperbarui konten secara dinamis, sehingga meningkatkan interaktivitas dan pengalaman pengguna.

b. Apa itu JSON? Apa kegunaannya?

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan dan mudah dibaca serta ditulis oleh manusia. JSON juga mudah untuk diurai (parse) dan dibentuk (generate) oleh mesin. JSON sering digunakan dalam komunikasi data antara server dan aplikasi web (terutama AJAX), karena kesederhanaannya dan kemampuannya untuk merepresentasikan struktur data kompleks seperti objek dan array.

c. Apa itu ParseMultipartForm dan MultipartReader ? Jelaskan perbedaannya!

- ParseMultipartForm: Fungsi ini adalah bagian dari struct http.Request yang memparsing seluruh body dari permintaan multipart/form-data dan menyimpannya dalam memori atau disk, tergantung pada batasan ukuran yang diberikan.
- MultipartReader: Ini adalah objek reader yang membaca data multipart secara streaming. Ini digunakan ketika kita ingin memproses data multipart dalam bagian-bagian kecil, yang bisa lebih efisien dalam hal penggunaan memory, terutama untuk file berukuran besar.

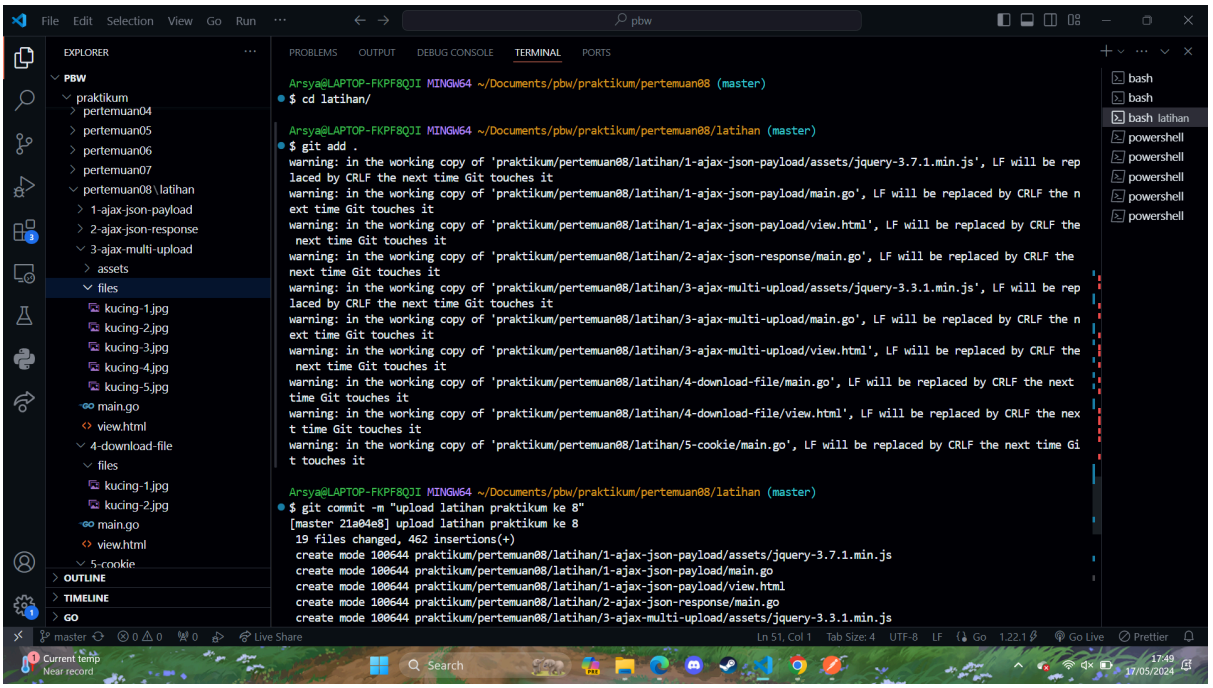
d. Apa itu JavaScript? Jelaskan kegunaannya dalam frontend !

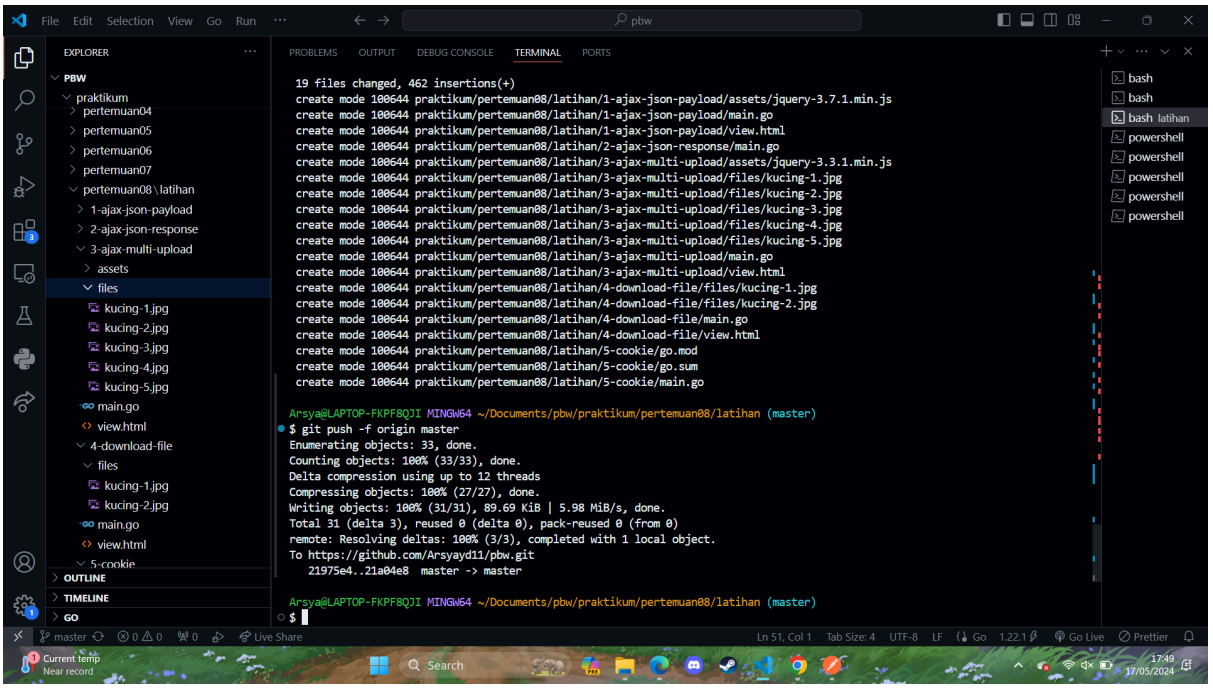
JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web interaktif. Dalam konteks frontend, JavaScript memungkinkan pengembang untuk menambahkan logika dinamis ke dalam situs web, seperti merespons tindakan pengguna (klik, input, dll.), memodifikasi konten halaman secara dinamis, dan berkomunikasi dengan server tanpa merefresh halaman (menggunakan AJAX).

e. Jelaskan apa itu EventHandler!

EventHandler adalah fungsi atau metode yang dijalankan sebagai respons terhadap event tertentu. Dalam konteks pemrograman web, event handler digunakan untuk menangani berbagai jenis event yang dipicu oleh pengguna, seperti klik mouse, penekanan tombol, perubahan nilai input, dan lain-lain.

Upload Ke Git





Kesimpulan

Pada praktikum ini mempelajari penggunaan AJAX untuk mengirim dan menerima JSON, mengunggah beberapa file sekaligus, mengunduh file dari server, dan mengelola cookie HTTP. Latihan ini menunjukkan bagaimana AJAX dapat meningkatkan interaktivitas aplikasi web tanpa perlu memuat ulang halaman, memudahkan pertukaran data dengan format JSON, serta mengoptimalkan pengolahan data multipart untuk file upload. Selain itu, kami mempelajari pengunduhan file dari server dan manajemen sesi pengguna melalui cookie, yang semuanya krusial untuk pengembangan web yang dinamis dan responsif.

Link Repository Github:
https://github.com/Arsyayd11/pbw/tree/master/praktikum/pertemuan08/latihan