

Nama : Ardhian Maulana Tanjung

NPM : 19242503

Kelas : 3IA01

---

1. Dalam pembuatan REST API, kita sering menggunakan method GET , POST , PUT, dan DELETE. Jelaskan perbedaan keempat method tersebut ! (5 poin)

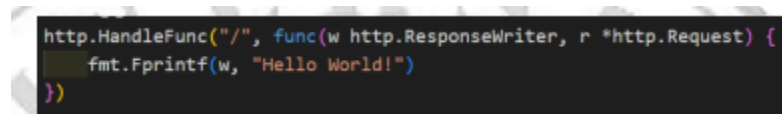
**Jawab** : - Metode GET : Berfungsi untuk membaca dan mengambil data

- Metode POST : Berfungsi untuk mengirim data ke server untuk membuat atau memperbarui data yang ada

- Metode PUT : Berfungsi untuk mengubah data yang ada pada server

- Metode DELETE : Berfungsi untuk menghapus data yang ada

2. Perhatikan potongan code berikut :



```
http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {  
    fmt.Fprintf(w, "Hello World!")  
})
```

Jelaskan secara singkat cara kerja dari potongan kode diatas ! (5 poin)

**Jawab** : Cara kerja code di atas yaitu HTTP merespon permintaan yang ada pada request dan kemudian akan menampilkan kata " Hello World! " sebagai output menggunakan kode `fmt.Fprintf` untuk menjalankan code.

ACT (total : 90 poin)

Anda baru saja bergabung dengan startup Filmku, sebuah startup yang bergerak di bidang perfilman. Anda bertugas sebagai backend developer. CTO meminta Anda membuat prototype REST API sederhana untuk mengelola katalog film. Karena masih tahap prototype, data akan disimpan dalam file JSON terlebih dahulu.

Tim product memberikan requirement sebagai berikut :

- API harus bisa membaca dan menyimpan informasi film (ID, Judul, Tahun Rilis, Genre, dan Rating) ke dalam file JSON
- API harus menyediakan endpoint untuk operasi dasar (lihat semua film, tambah film, update film, dan hapus film)
- Response API harus dalam format JSON

Lakukan Langkah pengerjaan berikut :

1. Importkan Library yang diperlukan ! (5 poin)
2. Buatlah struct film untuk menyimpan data sesuai struktur yang diminta, Struktur yang diharapkan (5 poin) :



3. Buatlah variable films bertipe array dengan struct film (5 poin)
4. Buatlah sistem untuk membaca dan menulis ke file JSON dengan spesifikasi berikut (5 poin) :
  - a. Fungsi loadData()
    - Membaca file films.json
    - Unmarshal data ke variable films
    - Handle error jika file tidak ditemukan
  - b. Fungsi saveData
    - Marshal data films ke format JSON
    - Simpan ke file films.json
    - Handle error saat proses penyimpanan
5. Buatlah helper function untuk memformat response dalam bentuk string ataupun JSON (10 poin) seperti code dibawah func respond(w http.ResponseWriter, message

Jawab :

Main.go

```
package main

//1
import (
    "encoding/json"
    "fmt"
    "log"
    "net/http"
    "os"
    "strconv"
)
```

```

// no 2,tulis dibawah ini
type Film struct {
    ID          int    `json:"id"`
    Judul       string  `json:"judul"`
    TahunRilis  int    `json:"tahun_rilis"`
    Genre       string  `json:"genre"`
    Rating      float64 `json:"rating"`
}

// no 3, tulis dibawah ini
var films []Film

func main() {
    loadData()
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        respond(w, "Selamat datang di API Film! Gunakan /films untuk mengakses
data film.", http.StatusOK)
    })
    http.HandleFunc("/films", handleFilms)
    http.HandleFunc("/films/", handleFilmByID)

    fmt.Println("Server berjalan di port 8003")
    log.Fatal(http.ListenAndServe(":8003", nil))
}

// no 4a, tulis dibawah ini
func loadData() {
    file, err := os.ReadFile("films.json")
    if err == nil {
        json.Unmarshal(file, &films)
    }
}

// no4b, tulis dibawah ini
func saveData() {
    data, _ := json.MarshalIndent(films, "", " ")
    os.WriteFile("films.json", data, 0644)
}

// no5
func respond(w http.ResponseWriter, message string, statusCode int) {
    w.Header().Set("Content-Type", "application/json")
    w.WriteHeader(statusCode)
    json.NewEncoder(w).Encode(map[string]string{"message": message})
}

```

```

}

func respondJSON(w http.ResponseWriter, data interface{}, statusCode int) {
    w.Header().Set("Content-Type", "application/json")
    w.WriteHeader(statusCode)
    json.NewEncoder(w).Encode(data)
}

// no 10
func handleFilms(w http.ResponseWriter, r *http.Request) {
    if r.Method == http.MethodGet {
        respondJSON(w, films, http.StatusOK)
    } else if r.Method == http.MethodPost {
        var newFilm Film
        if err := json.NewDecoder(r.Body).Decode(&newFilm); err == nil {
            newFilm.ID = len(films) + 1
            films = append(films, newFilm)
            saveData()
            respondJSON(w, newFilm, http.StatusCreated)
        } else {
            respond(w, "Format data tidak valid", http.StatusBadRequest)
        }
    } else {
        respond(w, "Metode tidak didukung", http.StatusMethodNotAllowed)
    }
}

func handleFilmByID(w http.ResponseWriter, r *http.Request) {
    id, err := strconv.Atoi(r.URL.Path[len("/films/"):])
    if err != nil {
        respond(w, "ID tidak valid", http.StatusBadRequest)
        return
    }

    for i, film := range films {
        if film.ID == id {
            switch r.Method {
            case http.MethodGet:
                respondJSON(w, film, http.StatusOK)
            case http.MethodPut:
                var updatedFilm Film
                if json.NewDecoder(r.Body).Decode(&updatedFilm) == nil {
                    updatedFilm.ID = id
                    films[i] = updatedFilm
                    saveData()
                }
            }
        }
    }
}

```

```

        respondJSON(w, updatedFilm, http.StatusOK)
    } else {
        respond(w, "Format data tidak valid", http.StatusBadRequest)
    }
}
case http.MethodDelete:
    films = append(films[:i], films[i+1:]...)
    saveData()
    respond(w, "data berhasil dihapus", http.StatusOK)
default:
    respond(w, "Metode tidak didukung", http.StatusMethodNotAllowed)
}
return
}
}
respond(w, "Film tidak ditemukan", http.StatusNotFound)
}

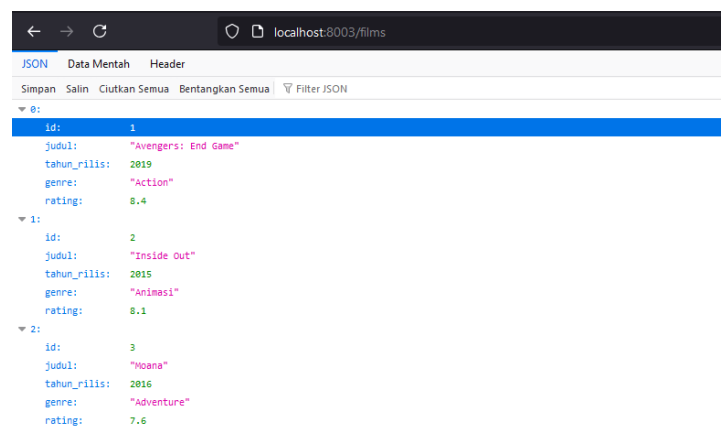
```

```

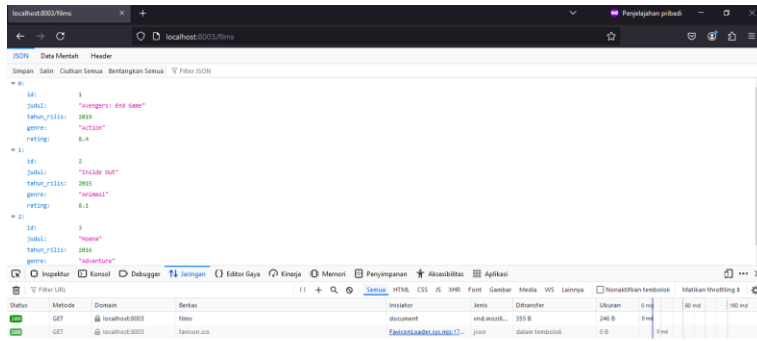
PS C:\go\src\Pert5_19242503> go run main.go
Server berjalan di port 8003

```

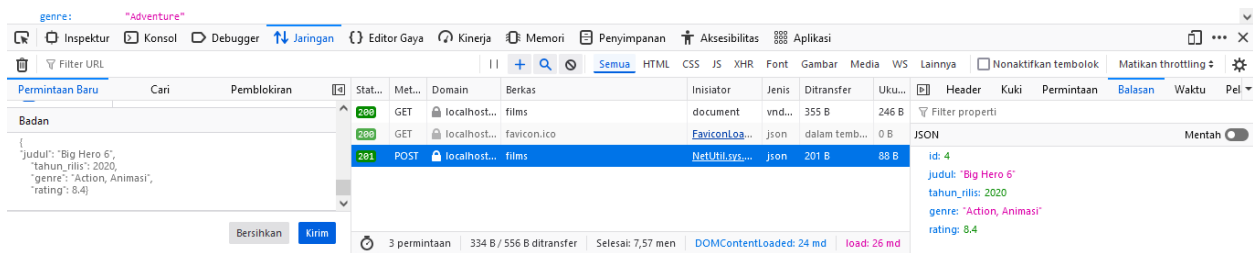
6. Buatlah endpoint dengan method GET dengan route “/films” untuk menampilkan semua film (5 poin)



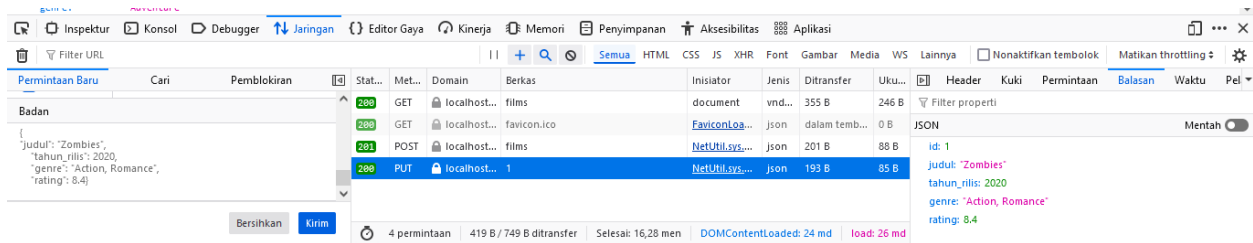
	id	Judul	tahun_rilis	genre	rating
0:	1	"Avengers: End Game"	2019	"Action"	8.4
1:	2	"Inside Out"	2015	"Animasi"	8.1
2:	3	"Moana"	2016	"Adventure"	7.6



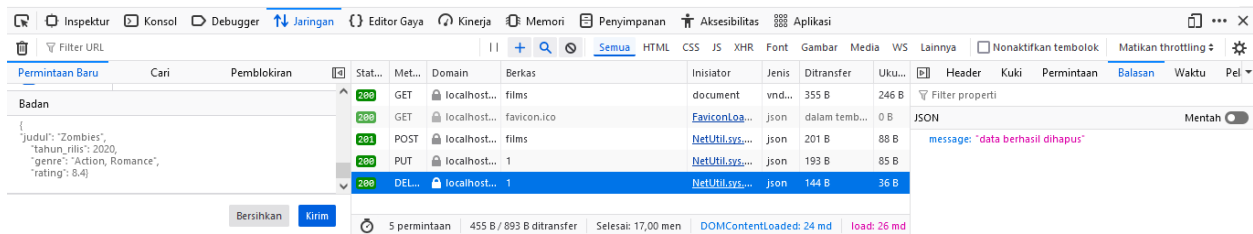
7. Buatlah endpoint dengan method POST dengan route “/films” untuk menambahkan film baru dengan ID auto-increment (10 poin)



8. Buatlah endpoint dengan method PUT dengan route “/films/{id}” untuk mengupdate informasi film berdasarkan ID (10 poin)



9. Buatlah endpoint dengan method DELETE dengan route “/films/{id}” untuk menghapus film berdasarkan ID (10 poin)



## 10. Buatlah Error Handling & Response format untuk penanganan scenario (25 poin) :

The screenshot shows the Chrome DevTools Network tab with the 'Balasan' (Response) pane selected. The response is a JSON object with a message: "Film tidak ditemukan". The request was a GET to http://localhost:8003/films/9.

Stat...	Met...	Domain	Berkas	Initiator	Jenis	Ditransfer	Uk...
200	GET	localhost...	films	document	vnd...	355 B	246 B
200	GET	localhost...	favicon.ico	FaviconLoa...	json	dalam temb...	0 B
201	POST	localhost...	films	NetUtil.sys...	json	201 B	88 B
200	PUT	localhost...	1	NetUtil.sys...	json	193 B	85 B
200	DEL...	localhost...	1	NetUtil.sys...	json	144 B	36 B
404	GET	localhost...	9	NetUtil.sys...	json	150 B	35 B

6 permintaan 490 B / 1,04 kB ditransfer Selesai: 19,35 men DOMContentLoaded: 24 md load: 26 m

The screenshot shows the Chrome DevTools Network tab with the 'Balasan' (Response) pane selected. The response is a JSON object with a message: "ID tidak valid". The request was a GET to http://localhost:8003/films/gaada.

Stat...	Met...	Domain	Berkas	Initiator	Jenis	Ditransfer	Uk...
200	GET	localhost...	favicon.ico	FaviconLoa...	json	dalam temb...	0 B
201	POST	localhost...	films	NetUtil.sys...	json	201 B	88 B
200	PUT	localhost...	1	NetUtil.sys...	json	193 B	85 B
200	DEL...	localhost...	1	NetUtil.sys...	json	144 B	36 B
400	GET	localhost...	9	NetUtil.sys...	json	150 B	35 B
400	GET	localhost...	gaada	NetUtil.sys...	json	146 B	29 B

7 permintaan 519 B / 1,19 kB ditransfer Selesai: 20,91 men DOMContentLoaded: 24 md load: 26 m

The screenshot shows the Chrome DevTools Network tab with the 'Balasan' (Response) pane selected. The response is a JSON object with a message: "Format data tidak valid". The request was a POST to http://localhost:8003/films/ with a body containing film details.

Stat...	Met...	Domain	Berkas	Initiator	Jenis	Ditransfer	Uk...
200	PUT	localhost...	1	NetUtil.sys...	json	193 B	85 B
200	DEL...	localhost...	1	NetUtil.sys...	json	144 B	36 B
400	GET	localhost...	9	NetUtil.sys...	json	150 B	35 B
400	GET	localhost...	gaada	NetUtil.sys...	json	146 B	29 B
400	POST	localhost...	/films/	NetUtil.sys...	json	146 B	29 B
400	POST	localhost...	films	NetUtil.sys...	json	155 B	38 B

9 permintaan 586 B / 1,49 kB ditransfer Selesai: 23,31 men DOMContentLoaded: 24 md load: 26 m