

CRUD RESTFUL API GOLANG DENGAN FRAMEWORK GIN GONIC
Untuk Memenuhi Ulangan Tengah Semester Genap Pemrograman Cloud Computing

Dosen Pengampu : Sopingi, M.Kom



Disusun Oleh

Liana Trihardianingsih (170101095)

SISTEM INFORMASI (A3)

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DUTA BANGSA SURAKARTA

2020

UJIAN TENGAH SEMESTER
PEMROGRAMAN CLOUD COMPUTING

A. Membuat Kode RestFul API Golang

1. Setelah menginstall Golang pada laptop/komputer, selanjutnya install driver MySQL dan framework Gin Gonic. Pastikan ada koneksi internet yang cukup memadai (lancar dan tidak lemot), karena ini akan mengunduh paket-paket yang digunakan untuk menjalankannya.
 - a. Install MySQL, ketikan perintah dibawah ini pada Git Bash atau CMD,
`go get -u github.com/go-sql-driver/mysql`
 - b. Install framework Gin Gonic, ketikan perintah dibawah ini pada Git Bash atau CMD,
`go get -u github.com/gin-gonic/gin`

2. Buat database animal_shelter dan tabel hewan

| Field | Type | Keterangan |
|-------------|--------------|-------------|
| id_hewan | int (13) | Primary Key |
| nama_hewan | Varchar (50) | |
| jenis_hewan | Varchar (50) | |
| jekel | Varchar(20) | |
| warna | Varchar(50) | |

3. Buatlah folder bernama UTS, lalu buka folder UTS itu pada Visual Code atau kode editor yang digunakan.
4. Buat file **hewan.go** dan mengisi **import**, kodenya seperti dibawah :

```
package main
import (
    "bytes"
    "database/sql"
    "fmt"
    "net/http"

    "github.com/gin-gonic/gin"
    _ "github.com/go-sql-driver/mysql"
)
```

Untuk memanggil framework yang sudah diinstal pada poin 1b agar framework bisa dijalankan

5. Buat **koneksi** di dalam **func main**, dan membuat fungsi menjalankan server :

```
func main() {
    db, err := sql.Open("mysql",
        "root:@tcp(127.0.0.1:3306)/animal_shelter")
    err = db.Ping()
    if err != nil {
        panic("Gagal Menghubungkan ke Database")
    }
    defer db.Close()

    router.Run(":8080")
}
```

Digunakan untuk menjalankan server localhost:8080

*kodingan selanjutnya akan ditulis pada jeda antara **defer db.Close()** dan **router.Run(":8080")**

UJIAN TENGAH SEMESTER
PEMROGRAMAN CLOUD COMPUTING

6. Buat **struct** di dalam *func main*, *struct* ini berisi dari *field* yang ada di tabel hewan :

```
type Hewan struct {
    Id_hewan    int    `json: "id_hewan"`
    Nama_hewan  string `json: "nama_hewan"`
    Jenis_hewan string `json: "jenis_hewan"`
    Jekel       string `json: "jekel"`
    Warna       string `json: "warna"`
}
```

7. Buat perintah untuk melakukan **routing**, dan buat fungsi untuk menampilkan (GET) data **hewan** berdasarkan **id_hewan** di dalam *func main* :

```
router := gin.Default()
```

Menunjukkan Gin Gonic yang nantinya akan melakukan Routing

```
router.GET("/:id", func(c *gin.Context) {
    var (
        hewan Hewan
        result gin.H
    )
    id_hewan := c.Param("id_hewan")
    row := db.QueryRow("select * from hewan where id_hewan = ?;",
        id_hewan)
    err = row.Scan(&hewan.Id_hewan, &hewan.Nama_hewan,
        &hewan.Jenis_hewan, &hewan.Jekel, &hewan.Warna)
    if err != nil {
        result = gin.H{
            "Hasilnya": "Tidak ada data hewan yang ditemukan",
        }
    } else {
        result = gin.H{
            "Hasilnya": hewan,
        }
    }
})
```

"Hewan" ini mengambil dari Struct di atas

8. Buat fungsi untuk menampilkan (GET) seluruh data **hewan** di dalam *func main* :

```
router.GET("/", func(c *gin.Context) {
    var (
        hewan Hewan
        hewans []Hewan
    )
    rows, err := db.Query("select * from hewan;")
    if err != nil {
        fmt.Print(err.Error())
    }
    for rows.Next() {
        err = rows.Scan(&hewan.Id_hewan, &hewan.Nama_hewan,
            &hewan.Jenis_hewan, &hewan.Jekel, &hewan.Warna)
        hewans = append(hewans, hewan)
        if err != nil {
            fmt.Print(err.Error())
        }
    }
    defer rows.Close()
    c.JSON(http.StatusOK, gin.H{
        "Hasilnya": hewans,
        "Jumlahnya": len(hewans),
    })
})
```

"Hewan" ini mengambil dari Struct di atas

UJIAN TENGAH SEMESTER
PEMROGRAMAN CLOUD COMPUTING

9. Buat fungsi untuk menambahkan (POST) data ke tabel **hewan** di dalam *func main*:

```
router.POST("/", func(c *gin.Context) {
    var buffer bytes.Buffer
    id_hewan := c.PostForm("id_hewan")
    nama_hewan := c.PostForm("nama_hewan")
    jenis_hewan := c.PostForm("jenis_hewan")
    jekel := c.PostForm("jekel")
    warna := c.PostForm("warna")
    stmt, err := db.Prepare("insert into hewan (id_hewan,
    nama_hewan, jenis_hewan, jekel, warna) values(?,?,?,?);")
    if err != nil {
        fmt.Print(err.Error())
    }
    _, err = stmt.Exec(id_hewan, nama_hewan, jenis_hewan, jekel,
    warna)

    if err != nil {
        fmt.Print(err.Error())
    }

    buffer.WriteString(nama_hewan)
    buffer.WriteString(", jenis:")
    buffer.WriteString(jenis_hewan)
    buffer.WriteString(" ")
    buffer.WriteString(jekel)
    buffer.WriteString(", warna:")
    buffer.WriteString(warna)
    defer stmt.Close()
    datanya := buffer.String()
    c.JSON(http.StatusOK, gin.H{
        "Pesannya": fmt.Sprintf(" Berhasil menambahkan Hewan %s
        ", datanya),
    })
})
```

10. Buat fungsi untuk merubah (PUT) data dari tabel **progdi** berdasarkan **id** di dalam *func main*:

```
router.PUT("/:id", func(c *gin.Context) {
    var buffer bytes.Buffer
    id_hewan := c.PostForm("id_hewan")
    nama_hewan := c.PostForm("nama_hewan")
    jenis_hewan := c.PostForm("jenis_hewan")
    jekel := c.PostForm("jekel")
    warna := c.PostForm("warna")
    stmt, err := db.Prepare("update hewan set nama_hewan= ?,
    jenis_hewan = ?, jekel= ?, warna= ? where id_hewan= ?;")
    if err != nil {
        fmt.Print(err.Error())
    }
    _, err = stmt.Exec(nama_hewan, jenis_hewan, jekel, warna,
    id_hewan)
    if err != nil {
        fmt.Print(err.Error())
    }

    buffer.WriteString(nama_hewan)
    buffer.WriteString(", jenis:")
    buffer.WriteString(jenis_hewan)
    buffer.WriteString(" ")
    buffer.WriteString(jekel)
```

UJIAN TENGAH SEMESTER PEMROGRAMAN CLOUD COMPUTING

```
buffer.WriteString(",warna:")
buffer.WriteString(warna)
defer stmt.Close()
datanya := buffer.String()
c.JSON(http.StatusOK, gin.H{
    "Pesannya": fmt.Sprintf("Berhasil Merubah Id %s Menjadi %s", id_hewan, datanya),
})
})
```

11. Buat fungsi untuk menghapus (DELETE) data dari tabel **progdi** berdasarkan **id** di dalam

func main :

```
router.DELETE("/", func(c *gin.Context) {
    id_hewan := c.PostForm("id_hewan")
    stmt, err := db.Prepare("delete from hewan where id_hewan= ?;")
    if err != nil {
        fmt.Print(err.Error())
    }
    _, err = stmt.Exec(id_hewan)
    if err != nil {
        fmt.Print(err.Error())
    }
    c.JSON(http.StatusOK, gin.H{
        "Pesannya": fmt.Sprintf("Berhasil Menghapus %s", id_hewan),
    })
})
```

B. Menjalankan RestFul API Menggunakan Postman

1. Jalankan XAMPP lalu mulai Apache dan MySql
2. Buka folder menyimpan kode go kedalam **Git Bash**
3. Ketikan perintah dibawah untuk menjalankan file go
go run hewan.go lalu klik enter
4. Tunggu hingga muncul jendela *Firewall* lalu klik **Allow Access**
5. Akan muncul seperti dibawah jika berhasil, disana sudah ada informasi server berjalan dimana dan cara menggunakan method bagaimana :

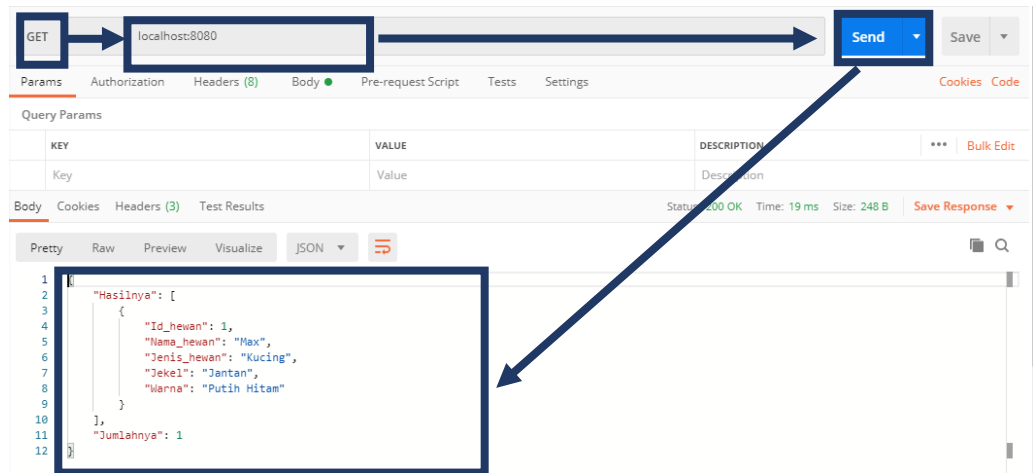
```
[GIN-debug] GET    /:id_hewan      --> main.main.func1 (3 handlers)
[GIN-debug] GET    /               --> main.main.func2 (3 handlers)
[GIN-debug] POST   /               --> main.main.func3 (3 handlers)
[GIN-debug] PUT    /               --> main.main.func4 (3 handlers)
[GIN-debug] DELETE /              --> main.main.func5 (3 handlers)
[GIN-debug] Listening and serving HTTP on :8080
```

6. Untuk melakukan ujicoba API, laptop/komputer harus sudah terinstall aplikasi **Postman** (kalo belum tahu bisa lihat di youtube/google) dan dijalankan.
7. Berikut merupakan hasilnya sekaligus tahapan mengeksekusi setiap method:

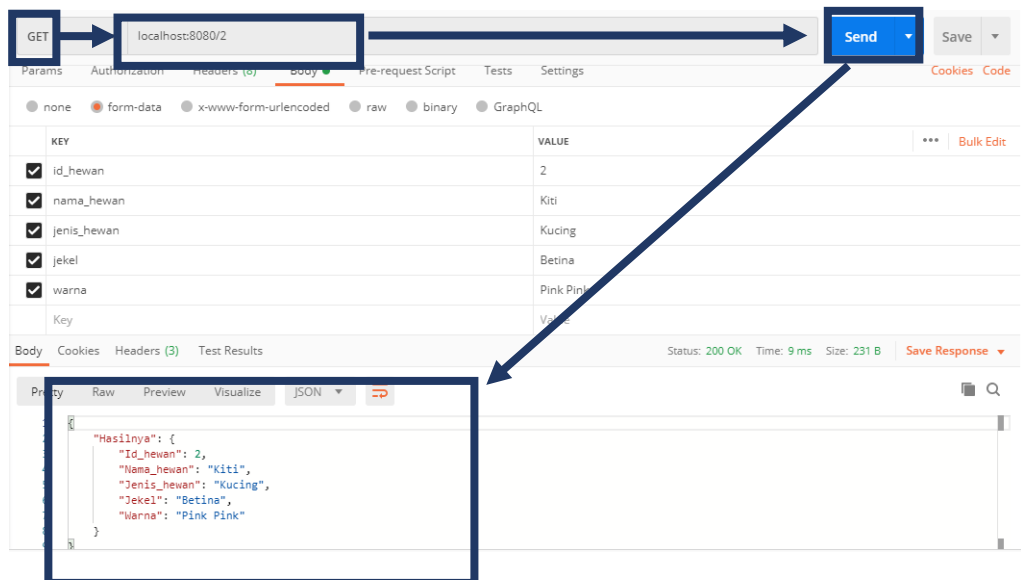
UJIAN TENGAH SEMESTER PEMROGRAMAN CLOUD COMPUTING

a. Method GET

GET untuk menampilkan seluruh data – <http://localhost:8080/>



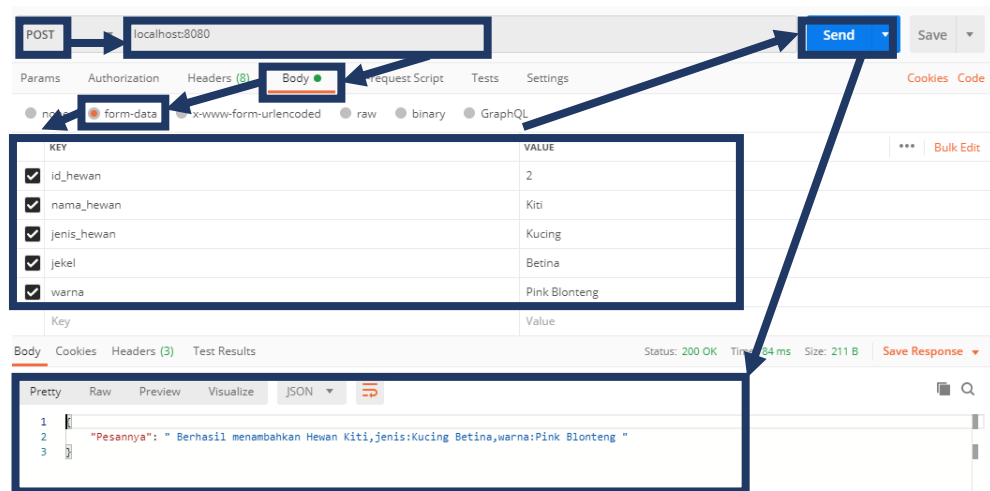
GET untuk menampilkan seluruh data – http://localhost:8080/id_hewan



UJIAN TENGAH SEMESTER PEMROGRAMAN CLOUD COMPUTING

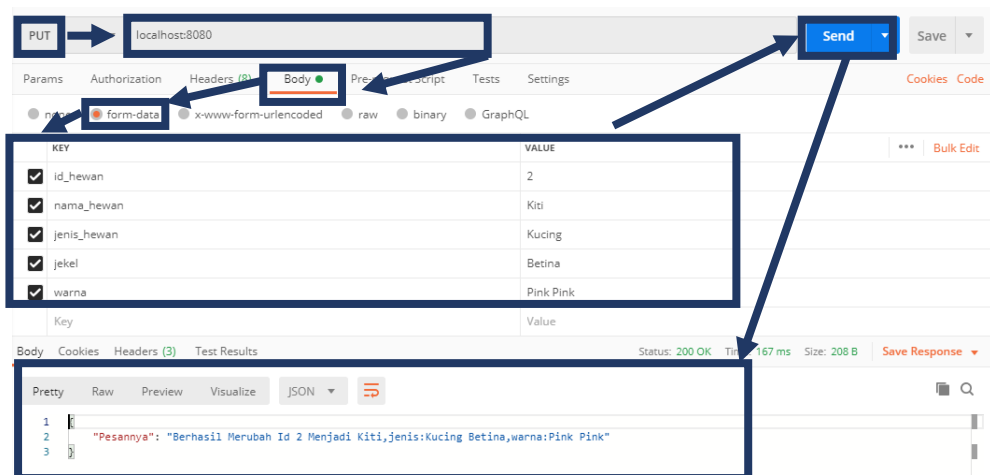
b. Method POST

POST untuk menambah data – <http://localhost:8080/>



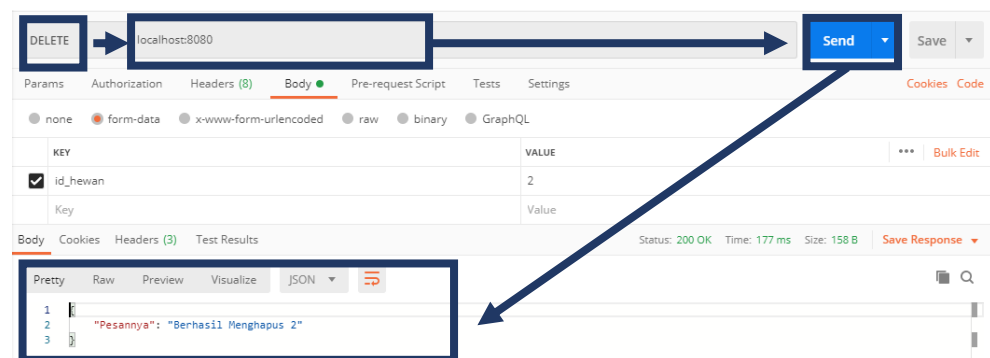
c. Method PUT

PUT untuk merubah data – [http://localhost:8080/\(id yang ingin di ubah\)](http://localhost:8080/(id yang ingin di ubah))



d. Method DELETE

DELETE untuk menghapus data – [http://localhost:8080/\(id yang ingin di hapus\)](http://localhost:8080/(id yang ingin di hapus))



**Untuk informasi lebih lengkap, dapat dilihat di channel Youtube Liana Trihardianingsih*