# Golang For Intermediate

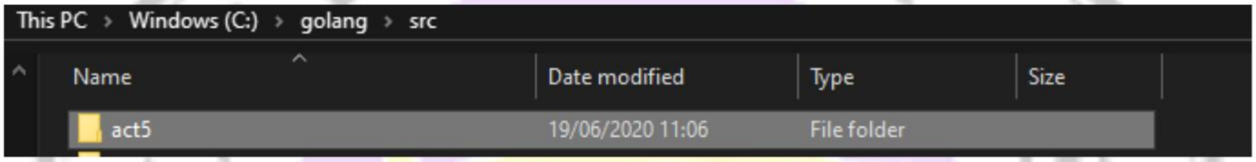
# ACTIVITY 5

# (*Data Binding*)

1. Apa yang dimaksud dengan data binding? (**POIN 30**)
2. Buatlah output seperti gambar dibawah! (**POIN 70**)

## Langkah 1

Persiapkan folder project pada **C:\golang\src\** atau pada GOPATH yang sudah kalian buat pada saat installasi. Setelah itu buatlah folder baru untuk menaruh project kalian.



This PC › Windows (C:) › golang › src

| Name | Date modified | Type | Size |
|---|---|---|---|
| act5 | 19/06/2020 11:06 | File folder | |

Setelah itu buatlah database mysql dengan query sebagai berikut:

```
CREATE DATABASE `nama_npm`;

CREATE TABLE `employee` (
`id` int(6) unsigned NOT NULL AUTO_INCREMENT,
`name` VARCHAR(30) NOT NULL, `city`
VARCHAR(30) NOT NULL,
PRIMARY KEY (`id`));
```

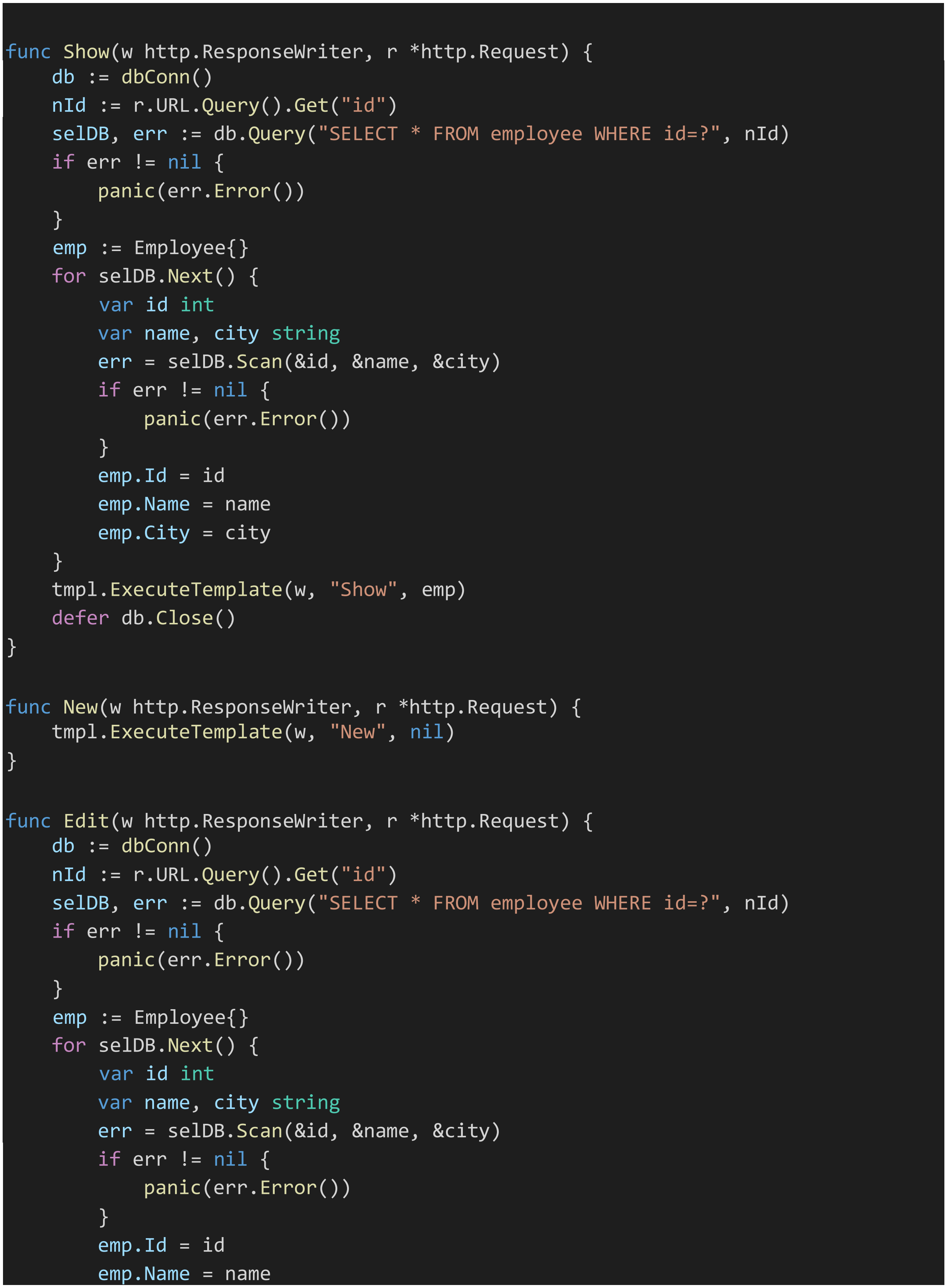
## Langkah 2

Buat file baru di dalam folder **act5** lalu beri nama **main.go**. Setelah itu masukan kode program di bawah ini:

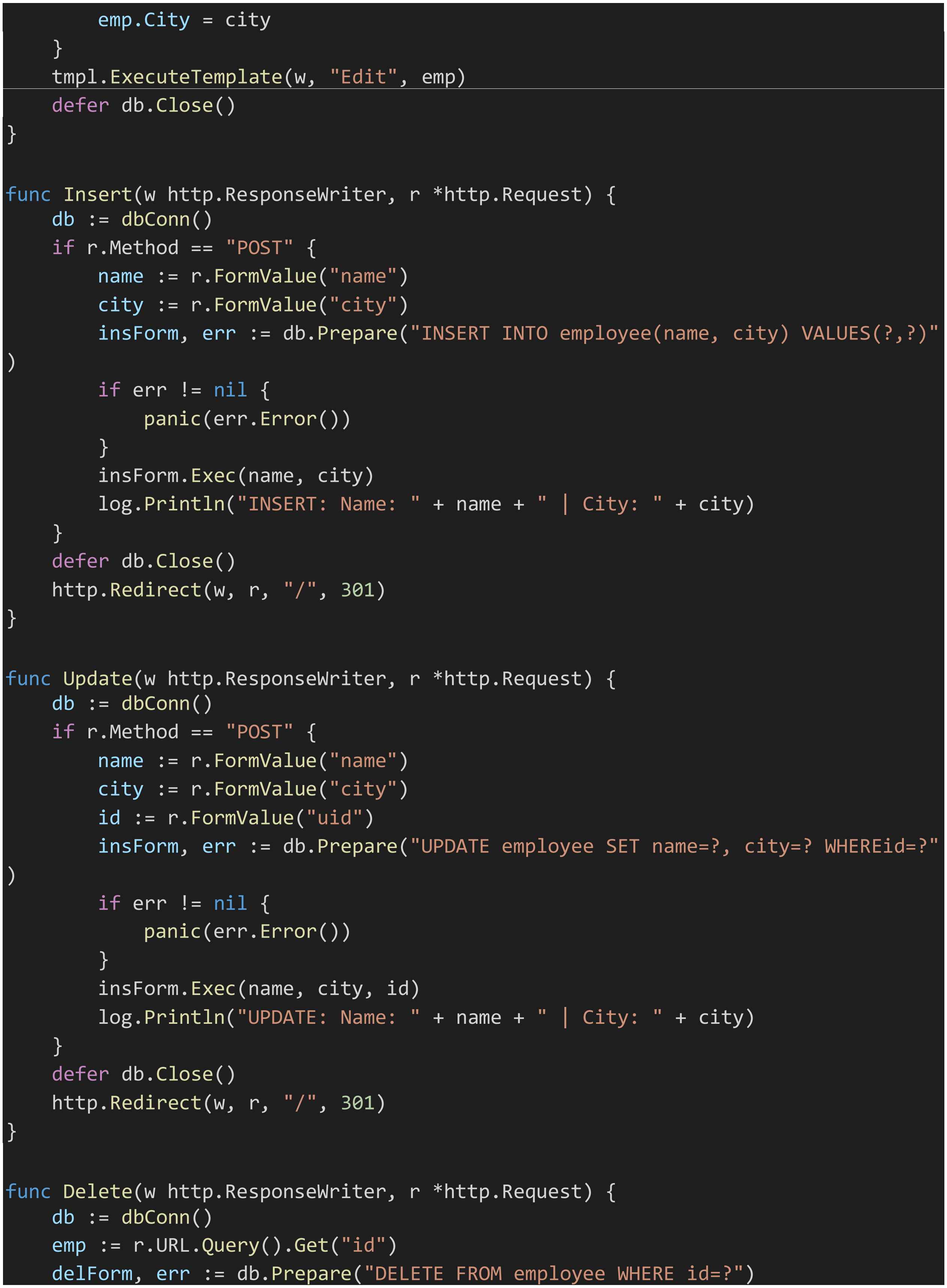
**main.go**

```
package main

import (
    "database/sql"
    "html/template"
    "log"
    "net/http"

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

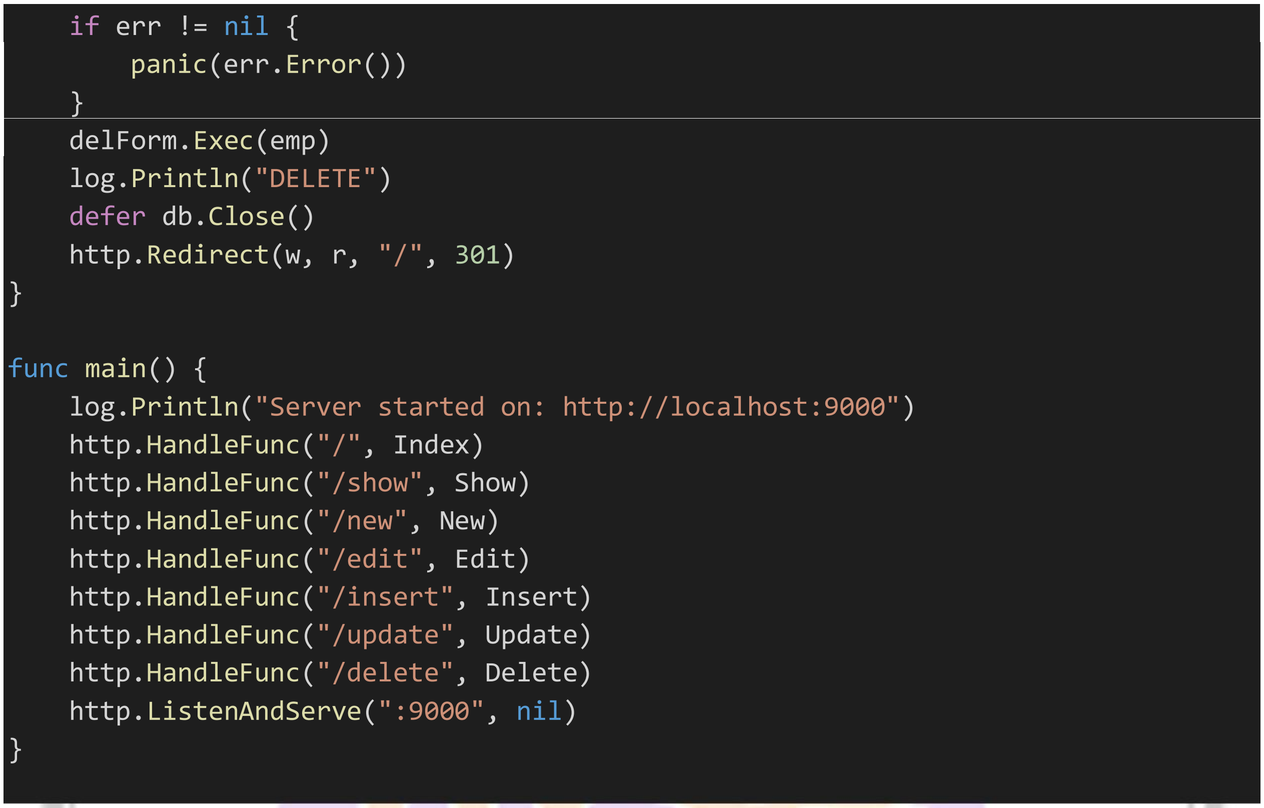
```
type Employee struct {
    Id   int
    Name string
    City string
}

func dbConn() (db *sql.DB) {
    dbDriver := "mysql"
    dbUser := "root"
    dbPass := "root"
    dbName := "act5"
    db, err := sql.Open(dbDriver,
        dbUser+":"+dbPass+"@tcp(localhost)/"+dbName)
    if err != nil {
        panic(err.Error())
    }
    return db
}

var tmpl = template.Must(template.ParseGlob("form/*"))

func Index(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    selDB, err := db.Query("SELECT * FROM employee ORDER BY id DESC")
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    res := []Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
        emp.City = city
        res = append(res, emp)
    }
    tmpl.ExecuteTemplate(w, "Index", res)
    defer db.Close()
}
```



```go
func Show(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    nId := r.URL.Query().Get("id")
    selDB, err := db.Query("SELECT * FROM employee WHERE id=?", nId)
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
        emp.City = city
    }
    tmpl.ExecuteTemplate(w, "Show", emp)
    defer db.Close()
}

func New(w http.ResponseWriter, r *http.Request) {
    tmpl.ExecuteTemplate(w, "New", nil)
}

func Edit(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    nId := r.URL.Query().Get("id")
    selDB, err := db.Query("SELECT * FROM employee WHERE id=?", nId)
    if err != nil {
        panic(err.Error())
    }
    emp := Employee{}
    for selDB.Next() {
        var id int
        var name, city string
        err = selDB.Scan(&id, &name, &city)
        if err != nil {
            panic(err.Error())
        }
        emp.Id = id
        emp.Name = name
```



```
        emp.City = city
    }
    tmpl.ExecuteTemplate(w, "Edit", emp)
    defer db.Close()
}

func Insert(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    if r.Method == "POST" {
        name := r.FormValue("name")
        city := r.FormValue("city")
        insForm, err := db.Prepare("INSERT INTO employee(name, city) VALUES(?,?)"
)
        if err != nil {
            panic(err.Error())
        }
        insForm.Exec(name, city)
        log.Println("INSERT: Name: " + name + " | City: " + city)
    }
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}

func Update(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    if r.Method == "POST" {
        name := r.FormValue("name")
        city := r.FormValue("city")
        id := r.FormValue("uid")
        insForm, err := db.Prepare("UPDATE employee SET name=?, city=? WHEREid=?"
)
        if err != nil {
            panic(err.Error())
        }
        insForm.Exec(name, city, id)
        log.Println("UPDATE: Name: " + name + " | City: " + city)
    }
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}

func Delete(w http.ResponseWriter, r *http.Request) {
    db := dbConn()
    emp := r.URL.Query().Get("id")
    delForm, err := db.Prepare("DELETE FROM employee WHERE id=?")
```



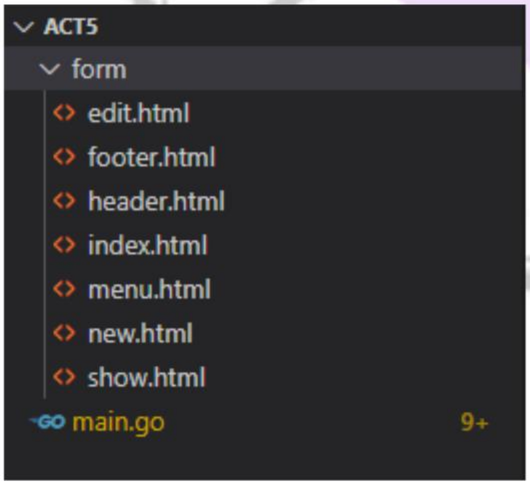
```
    if err != nil {
        panic(err.Error())
    }
    delForm.Exec(emp)
    log.Println("DELETE")
    defer db.Close()
    http.Redirect(w, r, "/", 301)
}

func main() {
    log.Println("Server started on: http://localhost:9000")
    http.HandleFunc("/", Index)
    http.HandleFunc("/show", Show)
    http.HandleFunc("/new", New)
    http.HandleFunc("/edit", Edit)
    http.HandleFunc("/insert", Insert)
    http.HandleFunc("/update", Update)
    http.HandleFunc("/delete", Delete)
    http.ListenAndServe(":9000", nil)
}
```

**Langkah 3**

Setelah membuat dan menyelesaikan kode program **main.go**, selanjutnya adalah membuat folder **form** di dalam direktori **C:\golang\src\act5** atau direktori project kalian. Dalam folder **form** buatlah file baru dengan nama:

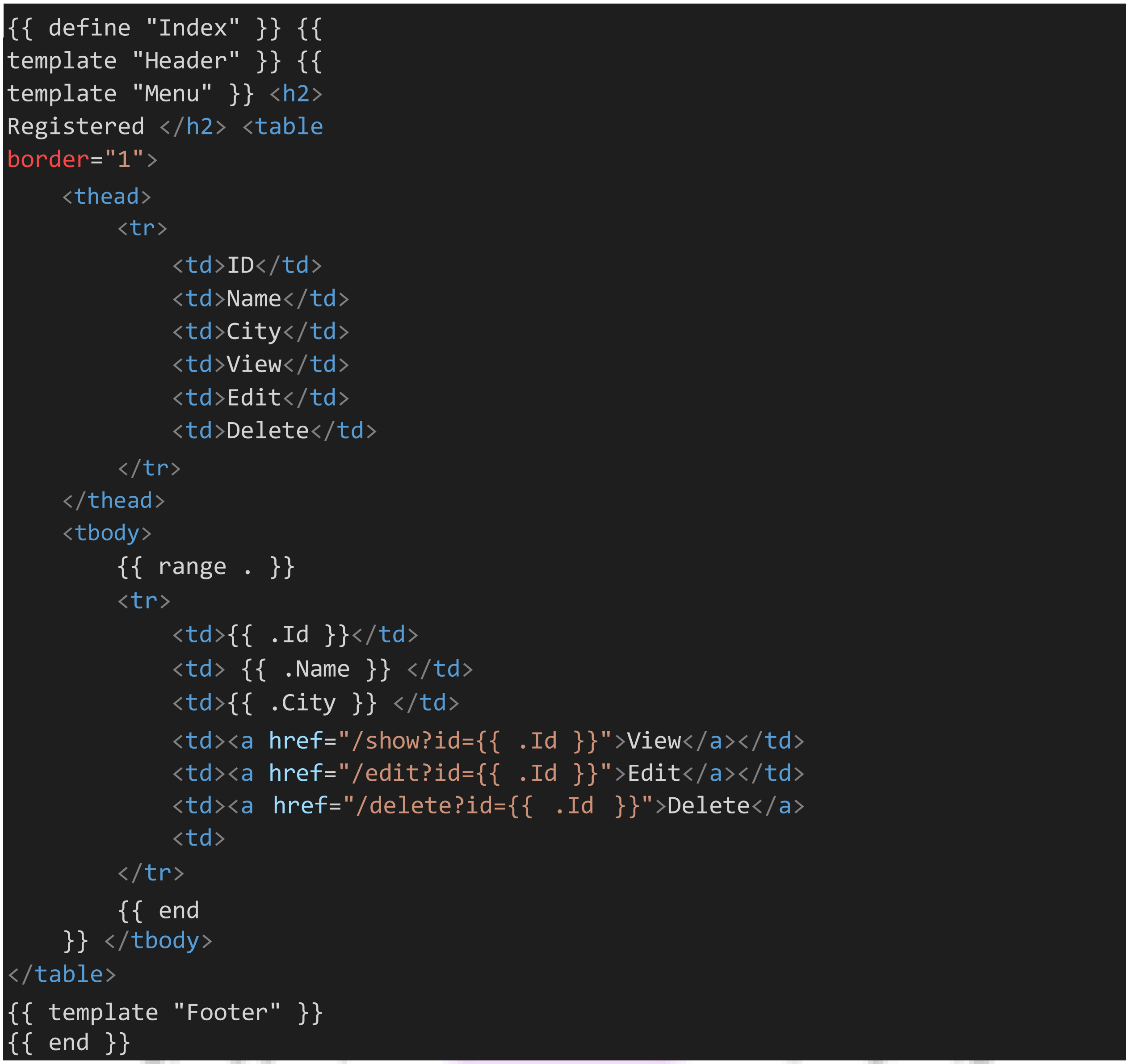
**index.html**, **header.html**, **footer.html**, **show.html**, **menu.html**, **new.html**, **edit.html**

Berikut adalah struktur folder setelah kalian membuat folder **form** dan file **html** di atas.



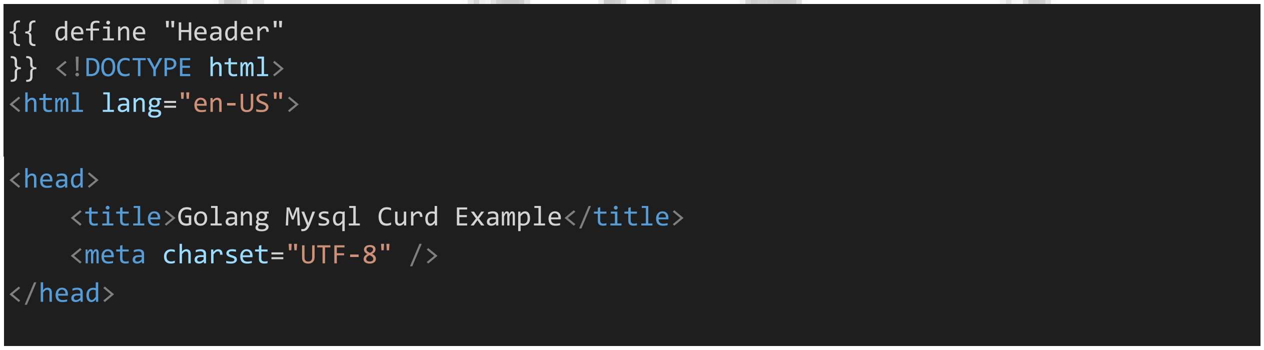
```
ACT5
  form
    edit.html
    footer.html
    header.html
    index.html
    menu.html
    new.html
    show.html
  main.go                    9+
```

## index.html

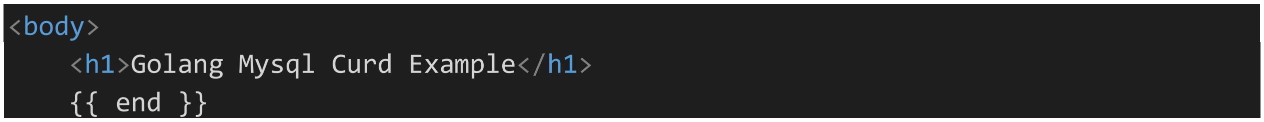


```
{{ define "Index" }} {{
template "Header" }} {{
template "Menu" }} <h2>
Registered </h2> <table
border="1">
    <thead>
        <tr>
            <td>ID</td>
            <td>Name</td>
            <td>City</td>
            <td>View</td>
            <td>Edit</td>
            <td>Delete</td>
        </tr>
    </thead>
    <tbody>
        {{ range . }}
        <tr>
            <td>{{ .Id }}</td>
            <td> {{ .Name }} </td>
            <td>{{ .City }} </td>
            <td><a href="/show?id={{ .Id }}">View</a></td>
            <td><a href="/edit?id={{ .Id }}">Edit</a></td>
            <td><a  href="/delete?id={{  .Id  }}">Delete</a>
            <td>
        </tr>
        {{ end
    }} </tbody>
</table>
{{ template "Footer" }}
{{ end }}
```

## header.html

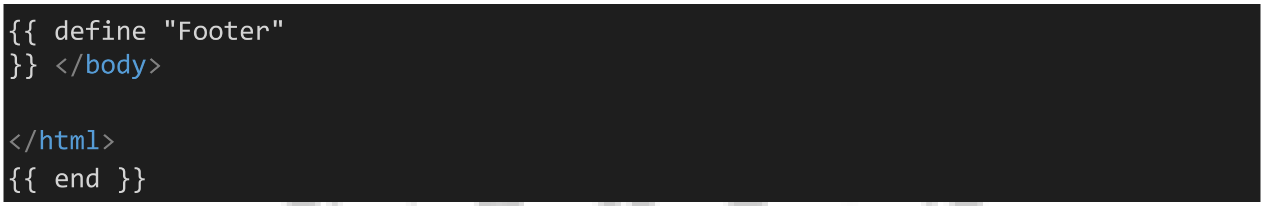


```
{{ define "Header"
}} <!DOCTYPE html>
<html lang="en-US">

<head>
    <title>Golang Mysql Curd Example</title>
    <meta charset="UTF-8" />
</head>
```



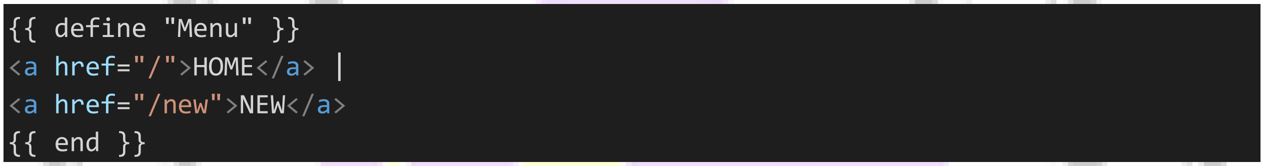
```
<body>
    <h1>Golang Mysql Curd Example</h1>
    {{ end }}
```

### footer.html



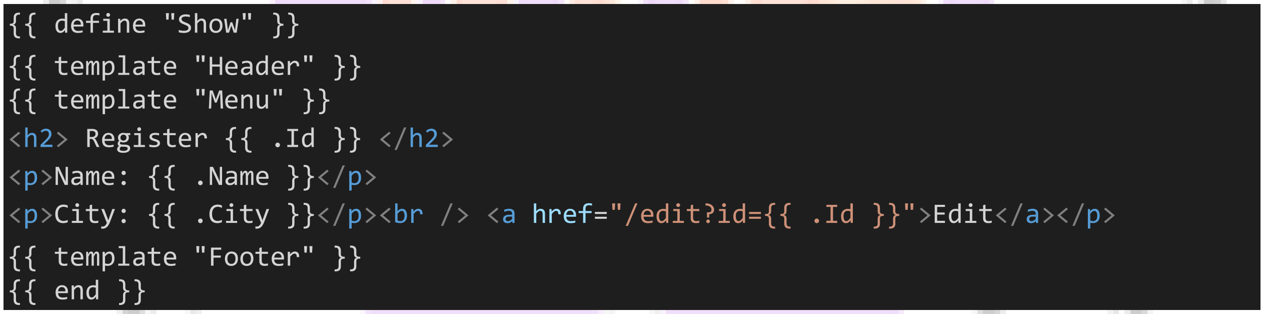
```
{{ define "Footer"
}} </body>

</html>
{{ end }}
```

### show.html



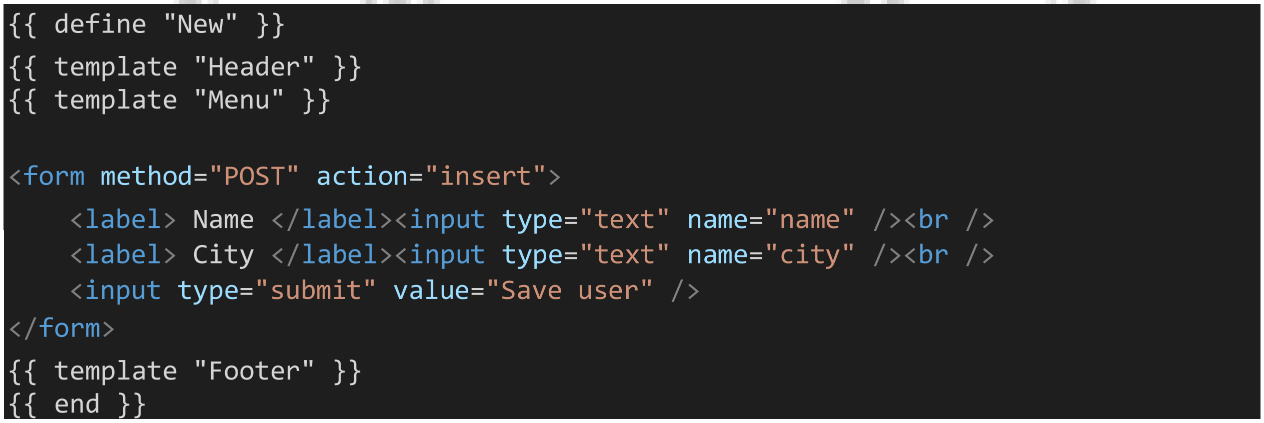
```
{{ define "Menu" }}
<a href="/">HOME</a> |
<a href="/new">NEW</a>
{{ end }}
```

### menu.html



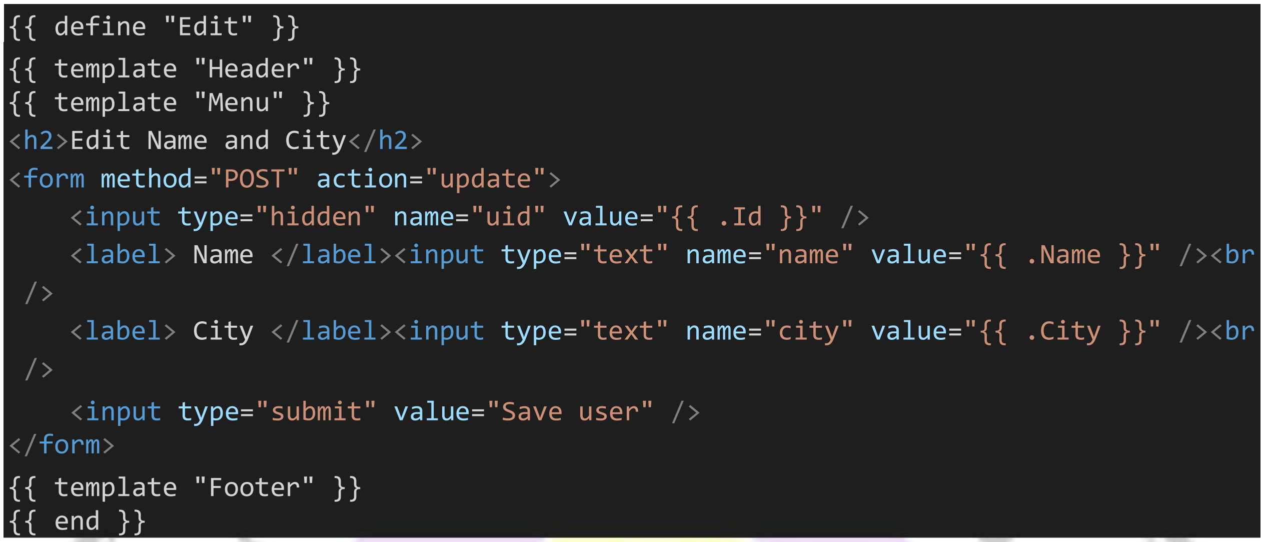
```
{{ define "Show" }}
{{ template "Header" }}
{{ template "Menu" }}
<h2> Register {{ .Id }} </h2>
<p>Name: {{ .Name }}</p>
<p>City: {{ .City }}</p><br /> <a href="/edit?id={{ .Id }}">Edit</a></p>
{{ template "Footer" }}
{{ end }}
```

### new.html



```
{{ define "New" }}
{{ template "Header" }}
{{ template "Menu" }}

<form method="POST" action="insert">
    <label> Name </label><input type="text" name="name" /><br />
    <label> City </label><input type="text" name="city" /><br />
    <input type="submit" value="Save user" />
</form>
{{ template "Footer" }}
{{ end }}
```

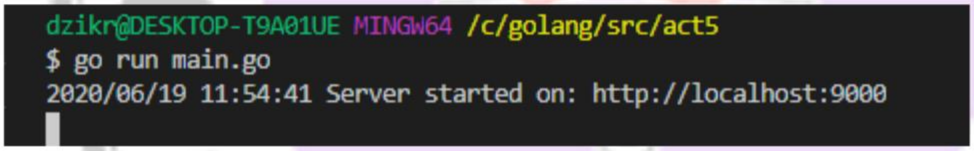
**edit.html**



```
{{ define "Edit" }}
{{ template "Header" }}
{{ template "Menu" }}
<h2>Edit Name and City</h2>
<form method="POST" action="update">
    <input type="hidden" name="uid" value="{{ .Id }}" />
    <label> Name </label><input type="text" name="name" value="{{ .Name }}" /><br
 />
    <label> City </label><input type="text" name="city" value="{{ .City }}" /><br
 />
    <input type="submit" value="Save user" />
</form>
{{ template "Footer" }}
{{ end }}
```

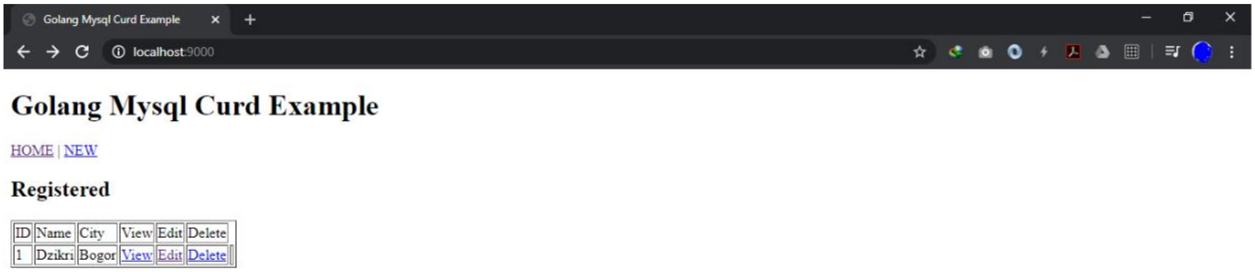
Jika kode program **main.go** dan template pada folder **form** tidak terdapat error, maka jalankan program dengan perintah berikut:

**go run main.go**



```
dzikr@DESKTOP-T9A01UE MINGW64 /c/golang/src/act5
$ go run main.go
2020/06/19 11:54:41 Server started on: http://localhost:9000
```

Setelah itu buka http://localhost:9000, maka akan tampil output seperti gambar di bawah:



# Golang Mysql Curd Example

HOME | NEW

### Registered

| ID | Name | City | View | Edit | Delete |
|---|---|---|---|---|---|
| 1 | Dzikri | Bogor | View | Edit | Delete |