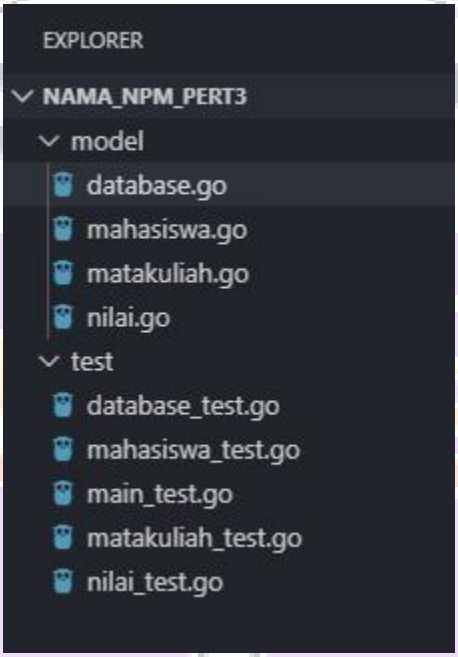
GOLANG FOR INTERMEDIATE

ACTIVITY 1

(CRUD MYSQL PADA GOLANG)

Struktur Folder pada Pertemuan 3 :



Pada **GOPATH** yaitu **c://golang/src/** buatlah sebuah folder bernama **NamaDepan_NPM_Pert3** (nama folder tidak boleh ada spasi)

Contoh : **ilhamhmmd _13116429_Pert3**

Didalam folder **Nama_NPM_Pert3** buatlah 2 buah folder yaitu **model** dan **test**

Pada **folder model** silahkan buat file go berikut ini :

database.go **(POIN 15)**

```
package model
```

```go
import (
    "database/sql"
    "fmt"
    _ "github.com/go-sql-driver/mysql"
)

type Table interface {
    Name() string
    Field() ([]string, []interface{})
}

func Connect(username string, password string, host string, database string) (*sql.DB, error) {
    conn := fmt.Sprintf("%s:%s@tcp(%s:3306)/%s", username, password, host, database)
    db, err := sql.Open("mysql", conn)
    return db, err
}

func CreateDB(db *sql.DB, name string) error {
    query := fmt.Sprintf("CREATE DATABASE %v", name)
    _, err := db.Exec(query)
    return err
}

func CreateTable(db *sql.DB, query string) error {
    _, err := db.Exec(query)
    return err
}

func DropDB(db *sql.DB, name string) error {
    query := fmt.Sprintf("DROP DATABASE  %v", name)
    _, err := db.Exec(query)
    return err
}
```

mahasiswa.go **(POIN 15)**

```go
package model

import (
    "database/sql"
    "fmt"
    "strings"
)

var TabelMahasiswa string = `
    CREATE TABLE mahasiswa(
        npm VARCHAR(10) PRIMARY KEY,
        nama VARCHAR(30),
        kelas VARCHAR(10)
    );
`

type Mahasiswa struct {
    NPM   string `json:"NPM"`
    Nama  string `json:"Nama"`
    Kelas string `json:"Kelas"`
}

func (m *Mahasiswa) Fields() ([]string, []interface{})
    { fields := []string{"npm", "nama", "kelas"}
    temp := []interface{}{&m.NPM, &m.Nama, &m.Kelas}
    return fields, temp
}

func (m *Mahasiswa) Structur() *Mahasiswa {
    return &Mahasiswa{}
}

func (m *Mahasiswa) Insert(db *sql.DB) error {
    query := fmt.Sprintf("INSERT INTO %v values(?,?,?)", "mahasiswa")
    _, err := db.Exec(query, &m.NPM, &m.Nama, &m.Kelas)
    return err
}

func (m *Mahasiswa) Update(db *sql.DB, data map[string]interface{}) er
ror {

    var kolom = []string{}
```

```
var args []interface{}
```

```
    i := 1

    // Ini loop data untuk dimasukan kedalam
    set, for key, value := range data {
        updateData := fmt.Sprintf("%v = ?",
        strings.ToLower(key)) kolom = append(kolom, updateData)
        args = append(args,
        value) i++
    }

    // Ubah array menjadi string dengan pemisah koma
    dataUpdate := strings.Join(kolom, ",")

    query := fmt.Sprintf("UPDATE %s SET %s WHERE %s = ?",
 "mahasiswa", dataUpdate, "NPM")
    args = append(args, m.NPM)
    // Exec dengan query yang ada
    _, err := db.Exec(query, args...)
    return err
}

func (m *Mahasiswa) Delete(db *sql.DB) error {
    query := fmt.Sprintf("DELETE FROM %s WHERE %s = ?", "mahasiswa", "
NPM")
    // Exec dengan query yang ada
    _, err := db.Exec(query, m.NPM)
    return err
}

func GetMahasiswa(db *sql.DB, id string) (*Mahasiswa, error) {

    m := &Mahasiswa{}
    each := m.Structur()
    _, dst := each.Fields()
    query := fmt.Sprintf("SELECT * FROM %s WHERE %s = ?",
 "mahasiswa", "NPM")

    // isinya akan dimasukan kedalam var dst yang dideklarasikan diata
s
    err := db.QueryRow(query, id).Scan(dst...)
    if err != nil {
        fmt.Println(err)
        return nil, err
    }
    return each, nil
}
```

```go
func GetAllMahasiswa(db *sql.DB) ([]*Mahasiswa, error)
    { m := &Mahasiswa{}
    query := fmt.Sprintf("SELECT * FROM %s", "mahasiswa")
    data, err := db.Query(query)
    if err != nil {
        return nil, err
    }

    defer data.Close()

    var result []*Mahasiswa

    for data.Next() {
        each := m.Structur()
        _, dst := each.Fields()

        err := data.Scan(dst...)
        if err != nil {
            return nil, err
        }
        fmt.Println(each)
        result = append(result, each)
    }

    return result, nil

}
```

Pada **folder test** silahkan buat file go berikut ini, namun perlu

database_test.go **(POIN 15)**

```go
package test

import (
    "database/sql"
    "fmt"
    "nama_npm_pert3/model" //sesuaikan dengan nama folder (case sensit
ive)
```

```
    "testing"
)

var username, password, host, namaDB, defaultDB string

func init() {
    username = "CPC_noPC" // Misal : CPC21
    password = "lepkom@123"
    host = "dbms.lepkom.f4.com"
    namaDB = "db_npm" // Misal : db_13116429
    defaultDB = "mysql"
}

func TestDatabase(t *testing.T) {

    t.Run("Testing untuk membuat database", func(t *testing.T) {
        db, err := model.Connect(username, password, host, defaultDB)

        defer db.Close()

        if err != nil {
            t.Fatal(err)
        }

        err = model.CreateDB(db, namaDB)

        if err != nil {
            t.Fatal(err)
        }

    })

    t.Run("Testing untuk memeriksa koneksi database", func(t *testing.
T) {

        db, err := model.Connect(username, password, host, defaultDB)

        defer db.Close()

        if err != nil {
            t.Fatal(err)
        }
    })

```

```
    /* Function Testing dibawah ini berfungsi untuk menghapus DB (Apab
ila ingin melakukan testing,
        silahkan di comment terlebih dahulu. Abaikan apabila telah ter
-comment)
    */

    // t.Run("Testing untuk menghapus database", func(t *testing.T) {

    //  db, err := model.Connect(username, password, host, defaultDB)

    //  defer db.Close()

    //  if err != nil {
    //      t.Fatal(err)
    //  }

    //  err = model.DropDB(db, namaDB)

    //  if err != nil {
    //      t.Fatal(err)
    //  }
    // })
}

func initDatabase() (*sql.DB, error) {

    dbInit, err := model.Connect(username, password, host, defaultDB)

    if err != nil {
        fmt.Println("Gagal melakukan koneksi")
    }

    if err = model.DropDB(dbInit, namaDB); err != nil
        { fmt.Println("Gagal menghapus database")
        return nil, err
    }

    if err = model.CreateDB(dbInit, namaDB); err != nil
        { fmt.Println("Gagal membuat database")
        return nil, err
    }

    dbInit.Close()

    db, err := model.Connect(username, password, password, namaDB)
```

```
    if err != nil {
        fmt.Println("Gagal melakukan koneksi")
        return nil, err
    }

    if err = model.CreateTable(db, model.TabelMahasiswa); err != nil
        { fmt.Println("Gagal membuat table mahasiswa")
        return nil, err
    }



    /*
      Silahkan dibaca tak perlu ditulis komentar multiline ini
      Catatan :
        1.  Baris kode yang dikomentari di bawah ini merupakan baris
            Kode untuk melakukan pembuatan tabel Matkul dan Nilai
        2.  Silahkan hilangkan komentar baris kode dibawah
            apabila telah membuat model dan test berupa :
             - matakuliah.go
             - nilai.go
             - matakuliah_test.go
             - nilai_test.go
    */

    // if err = model.CreateTable(db, model.TabelMatkul); err != nil {
    //  fmt.Println("Gagal membuat table matkul")
    //  return nil, err
    // }

    // if err = model.CreateTable(db, model.TabelNilai); err != nil {
    //  fmt.Println("Gagal membuat table nilai")
    //  return nil, err
    // }

    return db, nil
}
```

mahasiswa_test.go (POIN 15)

```go
package test

import (
    "nama_npm_pert3/model" //sesuaikan dengan nama folder (case sensit
ive)
    "testing"
)

func TestMahasiswa(t *testing.T) {

    var dataInsertMhs = []model.Mahasiswa{
        model.Mahasiswa{
            NPM:   "12345678",
            Nama:  "Budi Doremi",
            Kelas: "3KA20",
        },
        model.Mahasiswa{
            NPM:   "19283746",
            Nama:  "Doremi Budi",
            Kelas: "4KA20",
        },
        model.Mahasiswa{
            NPM:   "44444444",
            Nama:  "DoBud",
            Kelas: "4KA21",
        },
    }

    db, err := initDatabase()

    if err != nil {
        t.Fatal(err)
    }

    t.Run("Testing insert mahasiswa", func(t *testing.T) {
        for _, dataInsert := range dataInsertMhs {
            err := dataInsert.Insert(db)
            if err != nil {
                t.Fatal(err)
            }
        }
    })

    t.Run("Testing update mahasiswa", func(t *testing.T) {
        var updateData = map[string]interface{}{
            "nama": "Abdi Teh Ayeuna",
```

```
        }

        data := dataInsertMhs[0]
        if err := data.Update(db, updateData); err != nil
            { t.Fatal(err)
        }

    })

    t.Run("Testing Get mahasiswa", func(t *testing.T)
        { _, err := model.GetMahasiswa(db, "44444444")
        if err != nil {
            t.Fatal(err)
        }
    })

    t.Run("Testing Get mahasiswa", func(t *testing.T)
        { _, err := model.GetAllMahasiswa(db)
        if err != nil {
            t.Fatal(err)
        }

    })

    t.Run("Testing delete mahasiswa", func(t *testing.T) {
        data := dataInsertMhs[0]
        if err := data.Delete(db); err != nil {
            t.Fatal(err)
        }
    })
    defer db.Close()
}
```

Setelah itu silahkan jalankan Unit Testing database_test.go terlebih dahulu

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\laragon\www\go\src\nama_npm_pert3> cd test
PS C:\laragon\www\go\src\nama_npm_pert3\test> go test -run TestDatabase
PASS
ok      nama_npm_pert3/test     0.950s
PS C:\laragon\www\go\src\nama_npm_pert3\test>
```

Kemudian jalankan Unit Testing mahasiswa_test.go

```
PS C:\laragon\www\go\src\nama_npm_pert3\test> go test -run TestMahasiswa
&{12345678 Abdi Teh Ayeuna 3KA20}
&{19283746 Doremi Budi 4KA20}
&{44444444 DoBud 4KA21}
PASS
ok      nama_npm_pert3/test     2.088s
```

Sejauh ini program kamu sudah dapat berjalan sebagai mana mestinya

Selanjutnya silahkan lanjutkan dengan membuat dan jalankan Unit Testing-nya file go berikut ini:

model :

- **matakuliah.go** **(POIN 10)**
- **nilai.go** **(POIN 10)**

test :

- **matakuliah_test.go** **(POIN 10)**
- **nilai_test.go** **(POIN 10)**

Penjelasan penuh pada Activity ini terdapat di video, namun kaidah pengerjaannya terdapat di dokumen ini. Silahkan diselaraskan.

Selamat Mengerjakan Calon Orang-Orang Sukses,

Jangan Lupa Kurung Kurawalnya Harus Pas Hiya Hiya ☺