

UJIAN



Mata Praktikum : GOLANG INTERMEDIET
Kelas : 3IA20
Praktikum ke- : UJIAN
Tanggal : 13 November 2022
Materi : (CRUD MYSQL PADA GOLANG)
NPM : 50420900
Nama : Muhammad Reza Hidayat
Ketua Asisten :
Nama Asisten :
Paraf Asisten :
Jumlah Lembar :



LABORATORIUM TEKNIK INFORMATIKA
UNIVERSITAS GUNADARMA

2022

A. LISTING PROGRAM

1. Apa yang dimaksud dengan crud? (10 poin)

Jawab :

CRUD merupakan istilah yang digunakan dalam pembuatan atau pengolahan struktur basis data. CRUD terdiri dari perintah Create, Read, Update, dan Delete.

- ◆ Create => merupakan struktur pertama pada CRUD yang berfungsi untuk membuat record baru pada sistem basis data. Dengan perintah ini memungkinkan administrator untuk membuat database atau table baru.
- ◆ Read => merupakan perintah yang digunakan untuk mengambil atau mencari data tertentu dalam suatu database atau table. Fungsi read kurang lebih memiliki fungsi yang sama dengan perintah select ataupun search.
- ◆ Update => merupakan fungsi yang sering digunakan untuk melakukan perubahan pada data dalam suatu database. Perintah update adalah perintah yang digunakan untuk melakukan modifikasi pada data yang sudah disipkan sebelumnya.
- ◆ Delete => merupakan perintah yang biasa digunakan untuk melakukan penghapusan terhadap suatu data. Ketika suatu data sudah tidak diperlukan atau terjadi kesalahan dalam penginputan, maka perintah ini dapat digunakan untuk mengatasinya.

2. Sebutkan struktur crud pada Go Language! (10 poin)

Struktur Create

```
reza_50420900_pert7 - main.go
1 func createMahasiswa(w http.ResponseWriter, r *http.Request) {
2     w.Header().Set("Content-type", "application/json")
3
4     var response ResponseMessage
5
6     db := dbConn()
7     defer db.Close()
8
9     err := r.ParseForm()
10    if err != nil {
11        log.Print(err.Error())
12    }
13
14    npm := r.Form.Get("npm")
15    nama := r.Form.Get("nama")
16    kelas := r.Form.Get("kelas")
17    profile := "gambar1.jpg"
18
19    rows, err := db.Prepare("INSERT INTO mahasiswa(npm, nama, kelas, profile) VALUES(?, ?, ?, ?)")
20
21    if err != nil {
22        log.Print(err.Error())
23    }
24
25    rows.Exec(npm, nama, kelas, profile)
26
27    response.Status = true
28    response.Message = "Mahasiswa berhasil ditambahkan"
29
30    log.Print(response.Message)
31
32    json.NewEncoder(w).Encode(response)
33 }
```

Struktur Update

```
reza_50420900_pert7 - main.go

1 func updateMahasiswa(w http.ResponseWriter, r *http.Request) {
2     w.Header().Set("Content-type", "application/json")
3
4     var response ResponseMessage
5     var responseErr ResponseError
6     var mahasiswa Mahasiswa
7
8     db := dbConn()
9     defer db.Close()
10
11     err := r.ParseForm()
12     if err != nil {
13         log.Print(err.Error())
14     }
15
16     id := r.Form.Get("id")
17     npm := r.Form.Get("npm")
18     nama := r.Form.Get("nama")
19     kelas := r.Form.Get("kelas")
20
21     rows := db.QueryRow("SELECT id FROM mahasiswa WHERE id=?", id)
22     if err := rows.Scan(&mahasiswa.Id); err != nil && err == sql.ErrNoRows {
23         responseErr.Status = false
24         responseErr.Error = "Mahasiswa tidak ditemukan"
25         w.WriteHeader(http.StatusNotFound)
26         json.NewEncoder(w).Encode(responseErr)
27         return
28     }
29
30     update, err := db.Prepare("UPDATE mahasiswa SET npm=?, nama=?, kelas=? WHERE id=?")
31
32     if err != nil {
33         log.Print(err.Error())
34     }
35
36     update.Exec(npm, nama, kelas, id)
37     response.Status = true
38     response.Message = "Data mahasiswa berhasil diubah"
39
40     log.Print(response.Message)
41
42     json.NewEncoder(w).Encode(response)
43     return
44 }
```

Struktur Delete

```
reza_50420900_pert7 - main.go
1 func deleteMahasiswa(w http.ResponseWriter, r *http.Request) {
2     w.Header().Set("Content-type", "application/json")
3
4     var mahasiswa Mahasiswa
5     var response ResponseMessage
6     var responseErr ResponseError
7
8     db := dbConn()
9     defer db.Close()
10
11     params := mux.Vars(r)
12
13     rows := db.QueryRow("SELECT id FROM mahasiswa WHERE id=?", params["id"])
14     if err := rows.Scan(&mahasiswa.Id); err != nil && err == sql.ErrNoRows {
15         responseErr.Status = false
16         responseErr.Error = "Mahasiswa tidak ditemukan"
17         w.WriteHeader(http.StatusNotFound)
18         json.NewEncoder(w).Encode(responseErr)
19         return
20     }
21
22     delete, err := db.Prepare("DELETE FROM mahasiswa WHERE id=?")
23
24     if err != nil {
25         log.Print(err.Error())
26     }
27
28     delete.Exec(params["id"])
29
30     response.Status = true
31     response.Message = "Data mahasiswa berhasil dihapus"
32
33     log.Print(response.Message)
34
35     json.NewEncoder(w).Encode(response)
36     return
37 }
```

```
reza_50420900_pert7 - main.go
1 func main() {
2     r := mux.NewRouter().StrictSlash(true)
3     mime.AddExtensionType(".js", "application/javascript")
4
5     r.HandleFunc("/api/mahasiswa", getAllMahasiswa).Methods("GET")
6     r.HandleFunc("/api/mahasiswa/{id}", getMahasiswa).Methods("GET")
7     r.HandleFunc("/api/mahasiswa", createMahasiswa).Methods("POST")
8     r.HandleFunc("/api/mahasiswa", updateMahasiswa).Methods("PUT")
9     r.HandleFunc("/api/mahasiswa/{id}", deleteMahasiswa).Methods("DELETE")
10
11     r.HandleFunc("/mahasiswa/search/{keyword}", getMahasiswaByName).Methods("GET")
12
13     spa := spaHandler{staticPath: "polymer", indexPath: "index.html"}
14     r.PathPrefix("/").Handler(spa)
15
16     srv := &http.Server{
17         Handler:      r,
18         Addr:          "127.0.0.1:8000", // 2 angka belakang port diganti menjadi 2 angka dibelakang npm masing-masing
19         WriteTimeout:  15 * time.Second,
20         ReadTimeout:   15 * time.Second,
21     }
22
23     log.Print("Server berjalan di http://127.0.0.1:8000") // 2 angka belakang port diganti menjadi 2 angka dibelakang npm masing-masing
24     srv.ListenAndServe()
25 }
```

3. Membuat dan menginput data pada table database! (20 poin)

SQL COMMAND

```
undefined - pertemuan7.sql

1  --Membuat database reza_50420900_pert7
2  CREATE DATABASE reza_50420900_pert7;
3
4  --menggunakan database reza_50420900_pert7
5  use reza_50420900_pert7;
6
7  --membuat tabel mahasiswa
8  CREATE TABLE mahasiswa (
9      id int(6) unsigned AUTO_INCREMENT primary key,
10     npm char(8) NOT NULL,
11     nama varchar(30) NOT NULL,
12     kelas char(5) NOT NULL,
13     profile varchar(30)NOT NULL);
14
15 --menampilkan tabel employee
16 select * from mahasiswa;
17
18 --menampilkan database employee
19 desc mahasiswa;
```

Main.go

```
1 package main
2
3 import (
4     "database/sql"
5     "encoding/json"
6     "fmt"
7     "log"
8     "net/http"
9     "os"
10    "path/filepath"
11    "time"
12
13    _ "github.com/go-sql-driver/mysql"
14    "github.com/gorilla/mux"
15)
16
17 type spotondier struct {
18     staticPath string
19     indexPath string
20 }
21
22 type Mahasiswa struct {
23     Id      int    `json:"id"`
24     Npm     string `json:"npm"`
25     Nama    string `json:"nama"`
26     Kelas   string `json:"kelas"`
27     Profile string `json:"profile"`
28 }
29
30 type ResponseAllData struct {
31     Status bool    `json:"status"`
32     Data   []Mahasiswa `json:"data"`
33 }
34
35 type ResponseData struct {
36     Status bool    `json:"status"`
37     Data   Mahasiswa `json:"data"`
38 }
39
40 type ResponseMessage struct {
41     Status bool    `json:"status"`
42     Message string `json:"message"`
43 }
44
45 type ResponseError struct {
46     Status bool    `json:"status"`
47     Error  string `json:"error"`
48 }
49
50 func dbConn() (db *sql.DB) {
51     dbDriver := "mysql"
52     dbName := "reza50420900_port7" // Diganti menjadi nama database kalian masing-masing
53     dbUser := "root"
54     dbPass := ""
55     db, err := sql.Open(dbDriver, dbUser+"@"tcp(localhost)/"+dbName)
56     if err != nil {
57         panic(err.Error())
58     }
59     return db
60 }
61
62 func getAllMahasiswa(w http.ResponseWriter, r *http.Request) {
63     w.Header().Set("Content-type", "application/json")
64
65     var response ResponseAllData
66     var mahasiswa Mahasiswa
67     var mhs []Mahasiswa
68
69     db := dbConn()
70     rows, err := db.Query("SELECT * FROM mahasiswa")
71     defer db.Close()
72
73     if err != nil {
74         log.Println(err.Error())
75     }
76
77     for rows.Next() {
78         err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.Kelas, &mahasiswa.Profile)
79
80         if err != nil {
81             log.Println(err.Error())
82         } else {
83             mhs = append(mhs, mahasiswa)
84         }
85     }
86
87     response.Status = true
88     response.Data = mhs
89     json.NewEncoder(w).Encode(response)
90     return
91 }
92
93 func getMahasiswa(w http.ResponseWriter, r *http.Request) {
94     w.Header().Set("Content-type", "application/json")
95
96     var response ResponseData
97     var responseErr ResponseError
98     var mahasiswa Mahasiswa
99
100     db := dbConn()
101     defer db.Close()
102
103     params := mux.Vars(r)
104
105     rows := db.QueryRow("SELECT * FROM mahasiswa WHERE id=?", params["id"])
106     err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.Kelas, &mahasiswa.Profile)
107
108     if err != nil {
109         if err == sql.ErrNoRows {
110             responseErr.Status = false
111             responseErr.Error = "Mahasiswa tidak ditemukan"
112             w.WriteHeader(http.StatusNotFound)
113             json.NewEncoder(w).Encode(responseErr)
114             return
115         }
116     }
117
118     response.Status = true
119     response.Data = mahasiswa
120     json.NewEncoder(w).Encode(response)
121     return
122 }
123
124 func getMahasiswaByName(w http.ResponseWriter, r *http.Request) {
125     w.Header().Set("Content-type", "application/json")
126
127     var mahasiswa Mahasiswa
128     var mhs []Mahasiswa
129     var response ResponseAllData
130
131     db := dbConn()
132     defer db.Close()
133
134     params := mux.Vars(r)
135     query := fmt.Sprintf("SELECT * FROM mahasiswa WHERE nama LIKE '%s'", params["keyword"])
136     rows, err := db.Query(query)
137
138     if err != nil {
139         log.Println(err.Error())
140     }
141
142     for rows.Next() {
143         if err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.Kelas, &mahasiswa.Profile); err != nil {
144             log.Println(err.Error())
145         }
146
147         mhs = append(mhs, mahasiswa)
148     }
149
150     response.Status = true
151     response.Data = mhs
152     json.NewEncoder(w).Encode(response)
153     return
154 }
```

```

mux 00400000_port7 - mahgo

func createMahasiswa(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-type", "application/json")

    var response ResponseMessage

    db := dbConn()
    defer db.Close()

    err := r.ParseForm()
    if err != nil {
        log.Print(err.Error())
    }

    npm := r.Form.Get("npm")
    nama := r.Form.Get("nama")
    kelas := r.Form.Get("kelas")
    profile := "gambar1.jpg"

    rows, err := db.Prepare("INSERT INTO mahasiswa(npm, nama, kelas, profile) VALUES(?, ?, ?, ?)")

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

    rows.Exec(npm, nama, kelas, profile)

    response.Status = true
    response.Message = "Mahasiswa berhasil ditambahkan"

    log.Print(response.Message)

    json.NewEncoder(w).Encode(response)
}

func updateMahasiswa(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-type", "application/json")

    var response ResponseMessage
    var responseErr ResponseError
    var mahasiswa Mahasiswa

    db := dbConn()
    defer db.Close()

    err := r.ParseForm()
    if err != nil {
        log.Print(err.Error())
    }

    id := r.Form.Get("id")
    npm := r.Form.Get("npm")
    nama := r.Form.Get("nama")
    kelas := r.Form.Get("kelas")

    rows := db.QueryRow("SELECT id FROM mahasiswa WHERE id=?", id)
    if err := rows.Scan(&mahasiswa.Id); err != nil && err == sql.ErrNoRows {
        responseErr.Status = false
        responseErr.Error = "Mahasiswa tidak ditemukan"
        w.WriteHeader(http.StatusNotFound)
        json.NewEncoder(w).Encode(responseErr)
        return
    }

    update, err := db.Prepare("UPDATE mahasiswa SET npm=?, nama=?, kelas=? WHERE id=?")

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

    update.Exec(npm, nama, kelas, id)
    response.Status = true
    response.Message = "Data mahasiswa berhasil diubah"

    log.Print(response.Message)

    json.NewEncoder(w).Encode(response)
    return
}

func deleteMahasiswa(w http.ResponseWriter, r *http.Request) {
    w.Header().Set("Content-type", "application/json")

    var mahasiswa Mahasiswa
    var response ResponseMessage
    var responseErr ResponseError

    db := dbConn()
    defer db.Close()

    params := mux.Vars(r)

    rows := db.QueryRow("SELECT id FROM mahasiswa WHERE id=?", params["id"])
    if err := rows.Scan(&mahasiswa.Id); err != nil && err == sql.ErrNoRows {
        responseErr.Status = false
        responseErr.Error = "Mahasiswa tidak ditemukan"
        w.WriteHeader(http.StatusNotFound)
        json.NewEncoder(w).Encode(responseErr)
        return
    }

    delete, err := db.Prepare("DELETE FROM mahasiswa WHERE id=?")

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

    delete.Exec(params["id"])

    response.Status = true
    response.Message = "Data mahasiswa berhasil dihapus"

    log.Print(response.Message)

    json.NewEncoder(w).Encode(response)
    return
}

func (h spaHandler) ServeHTTP(w http.ResponseWriter, r *http.Request) {
    path := filepath.Join(h.staticPath, r.URL.Path)

    _, err := os.Stat(path)
    if os.IsNotExist(err) {
        http.ServeFile(w, r, filepath.Join(h.staticPath, h.indexPath))
        return
    } else if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }

    http.FileServer(http.Dir(h.staticPath)).ServeHTTP(w, r)
}

func main() {
    r := mux.NewRouter().StrictSlash(true)
    mime.AddExtensionType(".js", "application/javascript")

    r.HandleFunc("/api/mahasiswa", getAllMahasiswa).Methods("GET")
    r.HandleFunc("/api/mahasiswa/{id}", getMahasiswa).Methods("GET")
    r.HandleFunc("/api/mahasiswa", createMahasiswa).Methods("POST")
    r.HandleFunc("/api/mahasiswa", updateMahasiswa).Methods("PUT")
    r.HandleFunc("/api/mahasiswa/{id}", deleteMahasiswa).Methods("DELETE")

    r.HandleFunc("/mahasiswa/search/{keyword}", getMahasiswaByName).Methods("GET")

    spa := spaHandler{staticPath: "polymer", indexPath: "index.html"}
    r.PathPrefix("/").Handler(spa)

    srv := &http.Server{
        Handler: r,
        Addr: "127.0.0.1:8080", // 2 angka belakang port diganti menjadi 2 angka dibelakang npm masing-masing
        WriteTimeout: 15 * time.Second,
        ReadTimeout: 15 * time.Second,
    }

    log.Print("Server berjalan di http://127.0.0.1:8080") // 2 angka belakang port diganti menjadi 2 angka dibelakang npm masing-masing
    srv.ListenAndServe()
}

```


Mahasiswa.html

[illegible]

B. OUPUT PROGRAM

Go run main.go

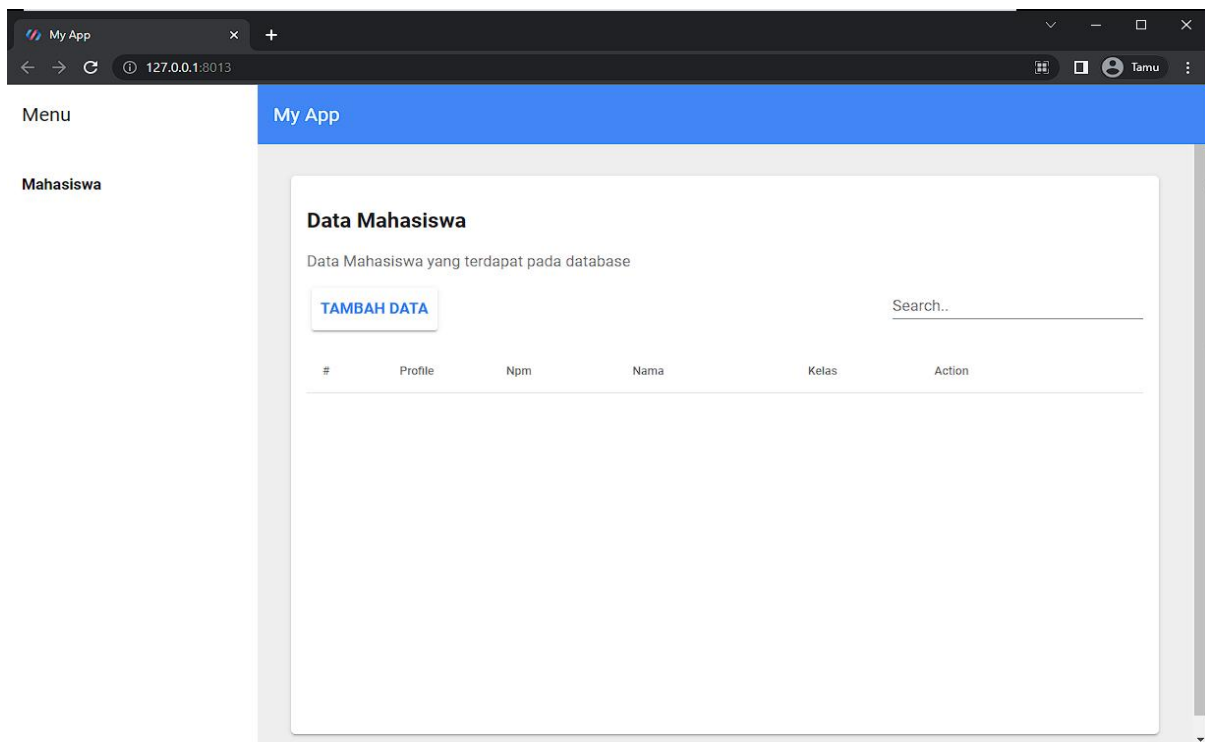
```
reza_50420900_pert7 - main.go

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

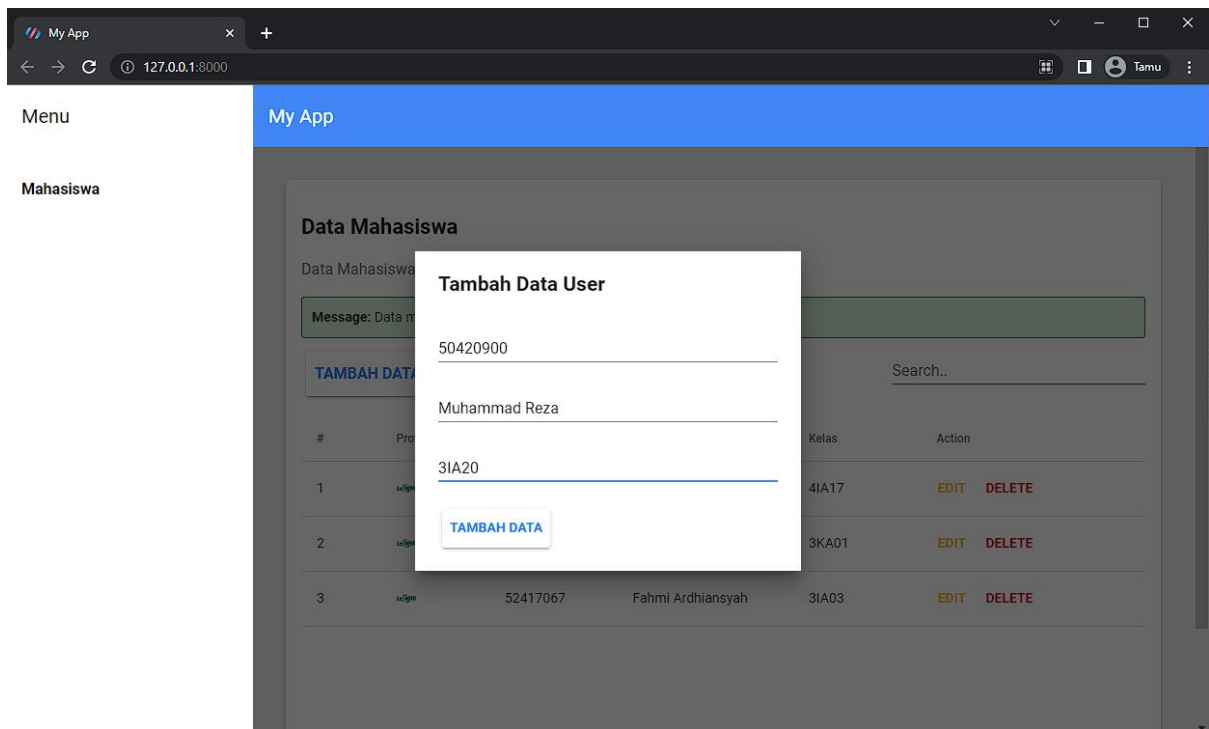
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Program Files\Go\src\reza_50420900_pert7> go run main.go
2022/11/04 18:11:06 Server berjalan di http://127.0.0.1:8000
2022/11/04 18:12:48 Mahasiswa berhasil ditambahkan
2022/11/04 18:13:11 Mahasiswa berhasil ditambahkan
2022/11/04 18:13:29 Mahasiswa berhasil ditambahkan
2022/11/04 18:14:22 Mahasiswa berhasil ditambahkan
2022/11/04 18:15:21 Data mahasiswa berhasil diubah
2022/11/04 18:16:48 Data mahasiswa berhasil dihapus
```

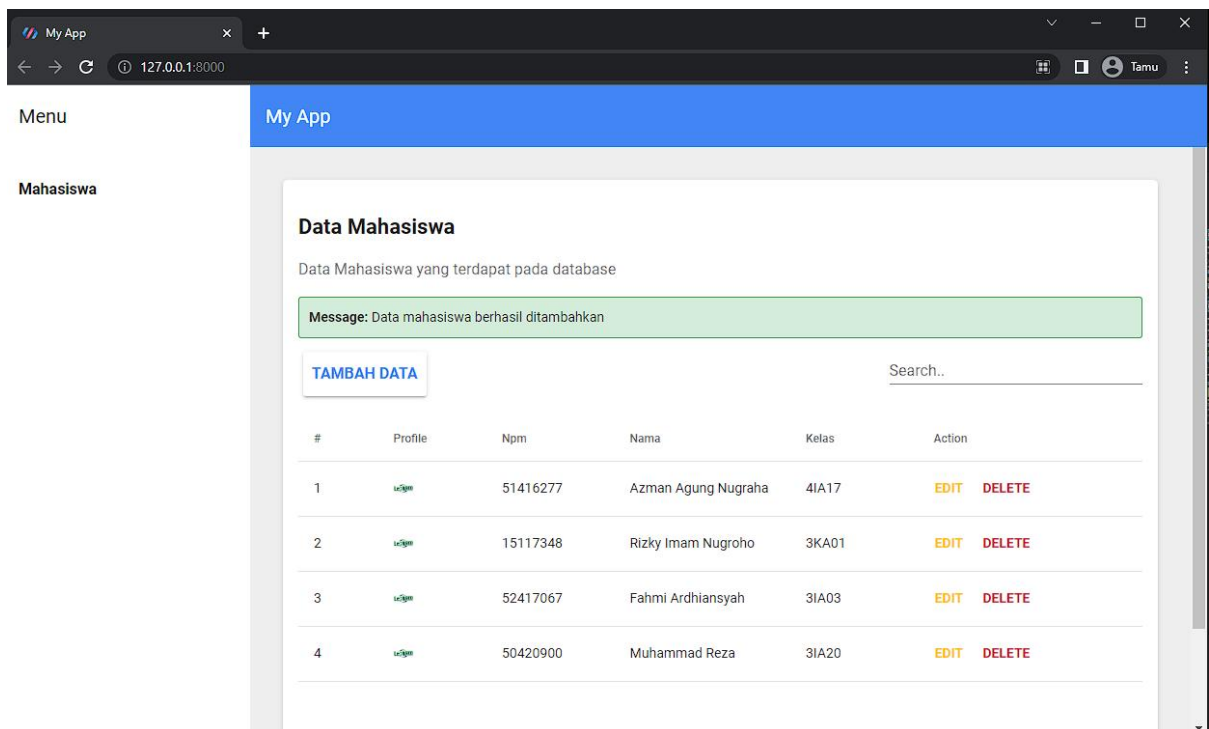
<http://127.0.0.1:8000/>



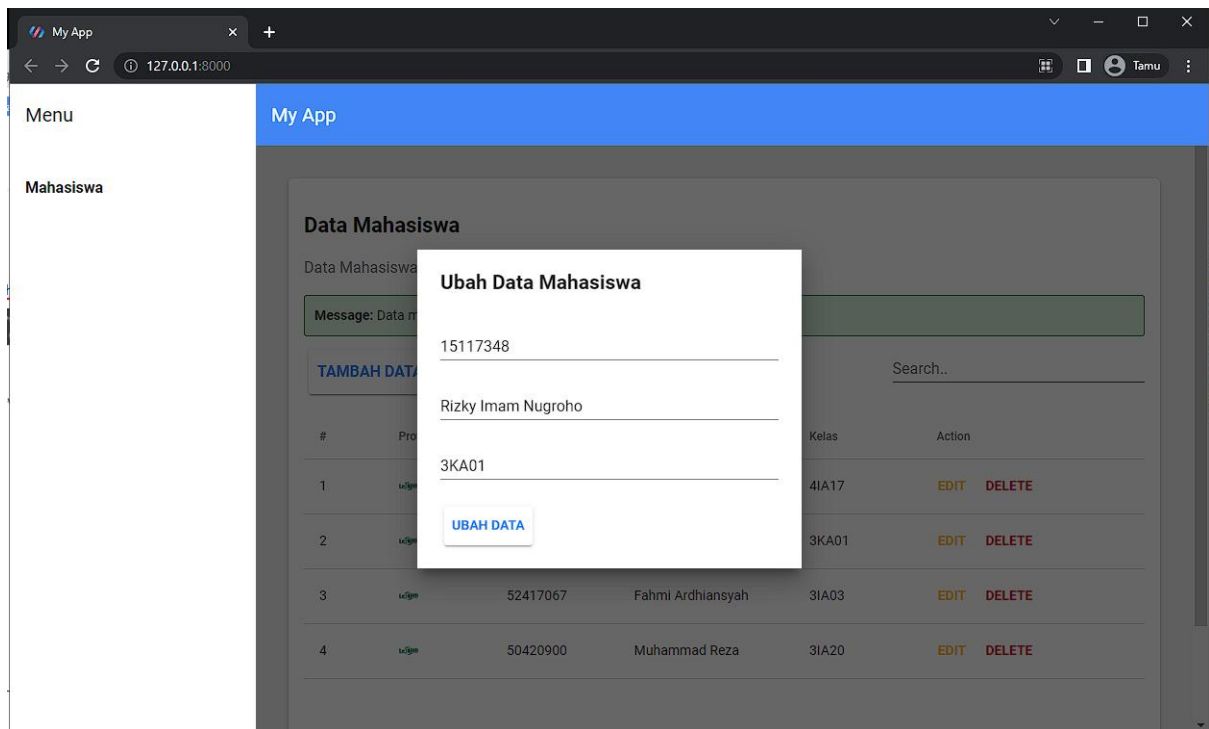
Input data



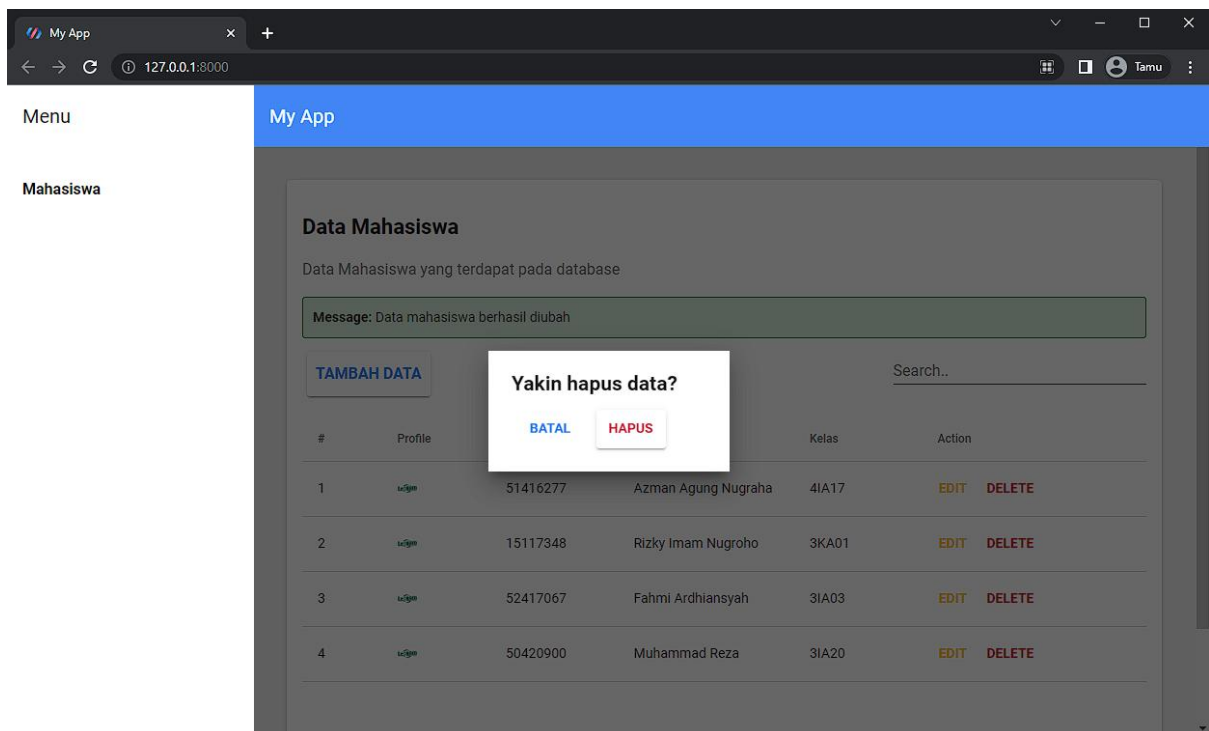
Data yang di input



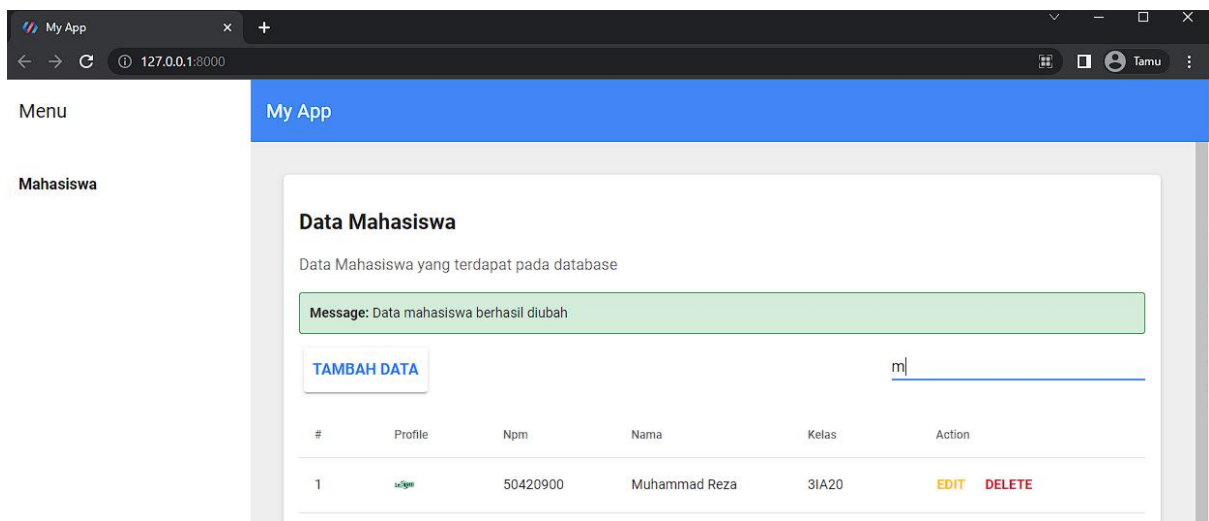
Edit data



Hapus data



Search



DATABASE

```
undefined - pertemuan7.sql

Setting environment for using XAMPP for Windows.
R@DESKTOP-Q3HDCLA c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.4.25-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE reza_50420900_pert7;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use reza_50420900_pert7;
Database changed
MariaDB [reza_50420900_pert7]> CREATE TABLE mahasiswa (
  -> id int(6) unsigned AUTO_INCREMENT primary key,
  -> npm char(8) NOT NULL,
  -> nama varchar(30) NOT NULL,
  -> kelas char(5) NOT NULL,
  -> profile varchar(30) NOT NULL);
Query OK, 0 rows affected (0.018 sec)

MariaDB [reza_50420900_pert7]> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(6) unsigned | NO   | PRI | NULL    | auto_increment |
| npm   | char(8)         | NO   |     | NULL    |                |
| nama  | varchar(30)     | NO   |     | NULL    |                |
| kelas | char(5)         | NO   |     | NULL    |                |
| profile | varchar(30)    | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.007 sec)

MariaDB [reza_50420900_pert7]>
```



undefined - pertemuan7.sql

Setting environment for using XAMPP for Windows.

R@DESKTOP-Q3HDCLA c:\xampp

mysql -u root -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 61

Server version: 10.4.25-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use reza_50420900_pert7;

Database changed

MariaDB [reza_50420900_pert7]> select * from mahasiswa;

| id | npm | nama | kelas | profile |
|----|----------|---------------------|-------|-------------|
| 1 | 51416277 | Azman Agung Nugraha | 4IA17 | gambar1.jpg |
| 3 | 52417067 | Fahmi Ardhiansyah | 3IA03 | gambar1.jpg |
| 4 | 50420900 | Muhammad Reza | 3IA20 | gambar1.jpg |

3 rows in set (0.000 sec)

MariaDB [reza_50420900_pert7]>