

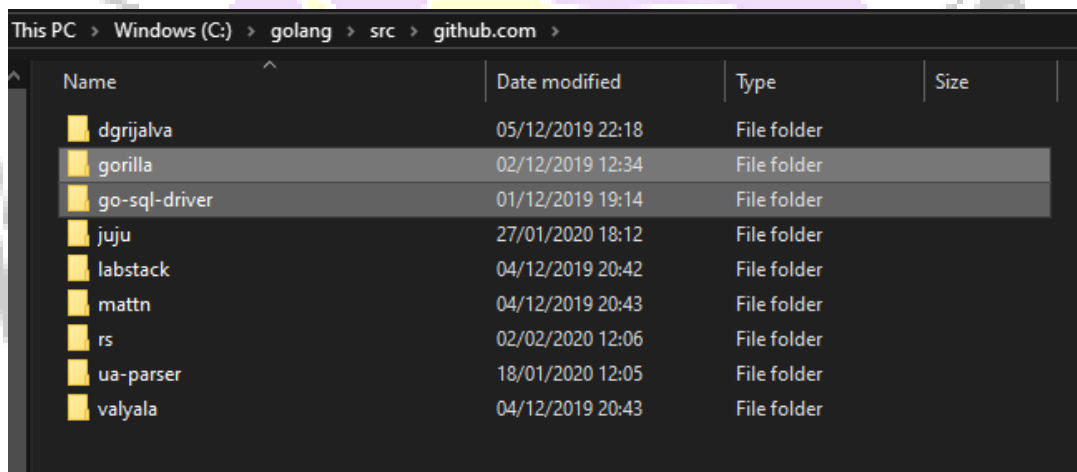
1. Apa yang dimaksud dengan crud? (10 poin)
2. Sebutkan struktur crud pada Go Language! (10 poin)
3. Membuat dan menginput data pada table database! (20 poin)

id	npm	nama	kelas	profile
1	51416277	Azman Agung Nugraha	4IA17	gambar1.jpg
2	15117348	Rizky Imam Nugroho	3KA01	gambar1.jpg
3	52417067	Fahmi Ardhiansyah	3IA03	gambar1.jpg

4. Buatlah output seperti gambar dibawah! (60 poin)

Langkah 1

Persiapkan folder project dan import paket **go-sql-driver** dan **gorilla** pada folder **golang/src/github.com**



Setelah itu buat database mysql, kalian dapat menggunakan mysql shell atau akses ke phpmyadmin.com untuk membuat database, berikut adalah querynya:

```
CREATE DATABASE `nama_npm` ;

CREATE TABLE `mahasiswa` (
  `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
  `npm` CHAR(8) NOT NULL,
  `nama` VARCHAR(30) NOT NULL,
  `kelas` CHAR(5) NOT NULL,
  `profile` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`id`));
```

## Langkah 2

Buatlah struct handler dan handler function pada main.go

```
package main

import (
    "database/sql"
    "encoding/json"
    "fmt"
    "log"
    "mime"
    "net/http"
    "os"
    "path/filepath"
    "time"

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

type spaHandler struct {
    staticPath string
    indexPath string
}

type Mahasiswa struct {
    Id      int    `json:"id"`
    Npm     string `json:"npm"`
    Nama    string `json:"nama"`
    Kelas   string `json:"kelas"`
    Profile string `json:"profile"`
}

type ResponseAllData struct {
    Status bool    `json:"status"`
    Data   []Mahasiswa `json:"data"`
}

type ResponseData struct {
    Status bool    `json:"status"`
    Data   Mahasiswa `json:"data"`
}
```

```

type ResponseMessage struct {
    Status bool `json:"status"`
    Message string `json:"message"`
}

type ResponseError struct {
    Status bool `json:"status"`
    Error string `json:"error"`
}

func dbConn() (db *sql.DB) {
    dbDriver := "mysql"
    dbName := "nama_npm"
    dbUser := "CPC[X]"
    dbPass := "lepkom@123"
    db, err := sql.Open(dbDriver, dbUser+": "+dbPass+"@tcp(dbms.lepkom.f4.com)/"+dbName)

    if err != nil {
        panic(err.Error())
    }

    return db
}

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

    var response ResponseAllData
    var mahasiswa Mahasiswa
    var mhs []Mahasiswa

    db := dbConn()
    rows, err := db.Query("SELECT * FROM mahasiswa")
    defer db.Close()

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

    for rows.Next() {
        err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.Kelas, &mahasiswa.Profile)

        if err != nil {

```

```

        log.Print(err.Error())
    } else {
        mhs = append(mhs, mahasiswa)
    }
}

response.Status = true
response.Data = mhs

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

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

    var response ResponseData
    var responseErr ResponseError
    var mahasiswa Mahasiswa

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

    params := mux.Vars(r)

    rows := db.QueryRow("SELECT * FROM mahasiswa WHERE id=?", params["id"])
    err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.K
elas, &mahasiswa.Profile)

    if err != nil && err == sql.ErrNoRows {
        responseErr.Status = false
        responseErr.Error = "Mahasiswa tidak ditemukan"
        w.WriteHeader(http.StatusNotFound)
        json.NewEncoder(w).Encode(responseErr)
        return
    }

    response.Status = true
    response.Data = mahasiswa

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

func getMahasiswaByName(w http.ResponseWriter, r *http.Request) {

```

```

w.Header().Set("Content-type", "application/json")

var mahasiswa Mahasiswa
var mhs []Mahasiswa
var response ResponseAllData

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

params := mux.Vars(r)
query := fmt.Sprintf("SELECT * FROM mahasiswa WHERE nama LIKE '%s%%'", params["keyword"])

rows, err := db.Query(query)

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

for rows.Next() {
    if err := rows.Scan(&mahasiswa.Id, &mahasiswa.Npm, &mahasiswa>Nama, &mahasiswa.Kelas, &mahasiswa.Profile); err != nil {
        log.Print(err.Error())
    }

    mhs = append(mhs, mahasiswa)
}

response.Status = true
response.Data = mhs

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

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)

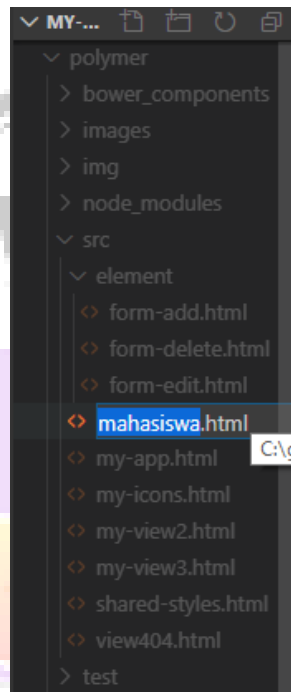
srv := &http.Server{
    Handler:      r,
    Addr:         "localhost:8081",
    WriteTimeout: 15 * time.Second,
    ReadTimeout:  15 * time.Second,
}

log.Print("Server berjalan di http://localhost:8081")
srv.ListenAndServe()
}
```



### Langkah 3

Masuklah ke dalam folder polymer/src, lalu ubah **my-view1.html** menjadi **mahasiswa.html**



Selanjutnya bukalah file **my-app.html**, ubahlah scriptnya menjadi seperti di bawah

```
<link rel="import" href="../../bower_components/polymer/polymer-element.html">
<link rel="import" href="../../bower_components/app-layout/app-drawer/app-drawer.html">
<link rel="import" href="../../bower_components/app-layout/app-drawer-layout/app-drawer-layout.html">
<link rel="import" href="../../bower_components/app-layout/app-header/app-header.html">
<link rel="import" href="../../bower_components/app-layout/app-header-layout/app-header-layout.html">
<link rel="import" href="../../bower_components/app-layout/app-scroll-effects/app-scroll-effects.html">
<link rel="import" href="../../bower_components/app-layout/app-toolbar/app-toolbar.html">
<link rel="import" href="../../bower_components/app-route/app-location.html">
<link rel="import" href="../../bower_components/app-route/app-route.html">
<link rel="import" href="../../bower_components/iron-pages/iron-pages.html">
<link rel="import" href="../../bower_components/iron-selector/iron-selector.html">
<link rel="import" href="../../bower_components/paper-icon-button/paper-icon-button.html">
```

```
<link rel="import" href="my-icons.html">

<link rel="lazy-import" href="mahasiswa.html">
<link rel="lazy-import" href="view404.html">

<dom-module id="my-app">
  <template>
    <style>
      :host {
        --app-primary-color: #4285f4;
        --app-secondary-color: black;

        display: block;
      }

      app-drawer-layout:not([narrow]) [drawer-toggle] {
        display: none;
      }

      app-header {
        color: #fff;
        background-color: var(--app-primary-color);
      }

      app-header paper-icon-button {
        --paper-icon-button-ink-color: white;
      }

      .drawer-list {
        margin: 0 20px;
      }

      .drawer-list a {
        display: block;
        padding: 0 16px;
        text-decoration: none;
        color: var(--app-secondary-color);
        line-height: 40px;
      }

      .drawer-list a.iron-selected {
        color: black;
        font-weight: bold;
      }
    </style>
```

```

<app-location route="{{route}}" url-space-regex="^[[rootPath]]">
</app-location>

<app-
route route="{{route}}" pattern="[[rootPath]]:page" data="{{routeData}}" tail="{{
subroute}}">
</app-route>

<app-drawer-layout fullbleed narrow="{{narrow}}">
  <!-- Drawer content -->
  <app-drawer id="drawer" slot="drawer" swipe-open="[[narrow]]">
    <app-toolbar>Menu</app-toolbar>
    <iron-selector selected="[[page]]" attr-for-
selected="name" class="drawer-list" role="navigation">
      <a name="mahasiswa" href="[[rootPath]]mahasiswa">Mahasiswa</a>
      <!-- <a name="view2" href="[[rootPath]]view2">View Two</a>
      <a name="view3" href="[[rootPath]]view3">View Three</a> -->
    </iron-selector>
  </app-drawer>

  <!-- Main content -->
  <app-header-layout has-scrolling-region>

    <app-header slot="header" condenses reveals effects="waterfall">
      <app-toolbar>
        <paper-icon-button icon="my-icons:menu" drawer-toggle></paper-icon-
button>

        <div main-title>My App</div>
      </app-toolbar>
    </app-header>

    <iron-pages selected="[[page]]" attr-for-selected="name" fallback-
selection="view404" role="main">
      <mahasiswa-page name="mahasiswa"></mahasiswa-page>
      <!-- <my-view2 name="view2"></my-view2>
      <my-view3 name="view3"></my-view3> -->
      <my-view404 name="view404"></my-view404>
    </iron-pages>
  </app-header-layout>
</app-drawer-layout>
</template>

<script>
  // Gesture events like tap and track generated from touch will not be

```

```

// preventable, allowing for better scrolling performance.
Polymer.setPassiveTouchGestures(true);

class MyApp extends Polymer.Element {
  static get is() { return 'my-app'; }

  static get properties() {
    return {
      page: {
        type: String,
        reflectToAttribute: true,
        observer: '_pageChanged',
      },
      routeData: Object,
      subroute: Object,
      // This shouldn't be necessary, but the Analyzer isn't picking up
      // Polymer.Element#rootPath
      rootPath: String,
    };
  }

  static get observers() {
    return [
      '_routePageChanged(routeData.page)',
    ];
  }

  _routePageChanged(page) {
    // If no page was found in the route data, page will be an empty string.
    // Default to 'view1' in that case.
    this.page = page || 'mahasiswa';

    // Close a non-persistent drawer when the page & route are changed.
    if (!this.$.drawer.persistent) {
      this.$.drawer.close();
    }
  }

  _pageChanged(page) {
    // Load page import on demand. Show 404 page if fails
    const resolvedPageUrl = this.resolveUrl(page + '.html');
    Polymer.importHref(
      resolvedPageUrl,
      null,
      this._showPage404.bind(this),

```

```

        true);
    }

    _showPage404() {
        this.page = 'view404';
    }
}

window.customElements.define(MyApp.is, MyApp);
</script>
</dom-module>

```

Buka file **mahasiswa.html**, lalu ubah scriptnya menjadi seperti di bawah

```

<link rel="import" href="../../bower_components/polymer/polymer-element.html">
<link rel="import" href="../../bower_components/polymer/lib/elements/dom-if.html">

<link rel="import" href="../../bower_components/vaadin-grid/theme/material/vaadin-grid.html">

<link rel="import" href="../../bower_components/paper-input/paper-input.html">
<link rel="import" href="../../bower_components/paper-button/paper-button.html">
<link rel="import" href="../../bower_components/paper-dialog/paper-dialog.html">

<link rel="import" href="../../bower_components/iron-input/iron-input.html">
<link rel="import" href="../../bower_components/iron-ajax/iron-ajax.html">

<link rel="import" href="shared-styles.html">

<link rel="import" href="element/form-add.html">
<link rel="import" href="element/form-edit.html">
<link rel="import" href="element/form-delete.html">

<dom-module id="mahasiswa-page">
  <template>
    <style include="shared-styles">
      :host {
        display: block;

        padding: 10px;
      }
    </style>
    <custom-style>
      <style is="custom-style">
        paper-input-container {

```

```

        width: 30%;
        float: right;
    }

    paper-dialog#deleteModal {
        width: 250px;
        max-width: 250px;
    }

    .warning,
    .danger {
        margin-left: -20px;
    }
</style>
</custom-style>

<div class="card">
    <h1>Data Mahasiswa</h1>
    <p>Data Mahasiswa yang terdapat pada database</p>

    <template is="dom-if" if="[[success]]">
        <p class="alert-success"><strong>Message:</strong> [[success]]</p>
    </template>

    <!-- Get data from rest api -->
    <iron-ajax auto url="{{url}}" method="{{method}}" handle-as="json" content-
type="application-json"
        last-response="{{response}}"></iron-ajax>

    <paper-button raised class="link" style="margin-bottom: 20px;" on-
click="openAddModal">Tambah Data</paper-button>

    <paper-input-container no-label-float>
        <label slot="label">Search..</label>
        <iron-input slot="input">
            <input type="text" value="{{keyword::input}}" width="30%">
        </iron-input>
    </paper-input-container>

    <!-- Datatable -->
    <vaadin-grid items="{{response.data}}">
        <vaadin-grid-column width="50px">
            <template class="header">#</template>
            <template>
                <div>[[index]]</div>
            </template>
        </vaadin-grid-column>
    </vaadin-grid>

```

```

        </template>
    </vaadin-grid-column>
    <vaadin-grid-column width="80px">
        <template class="header">Profile</template>
        <template>
            
        </template>
    </vaadin-grid-column>
    <vaadin-grid-column>
        <template class="header">Npm</template>
        <template>
            <div>[[item.npm]]</div>
        </template>
    </vaadin-grid-column>
    <vaadin-grid-column width="150px">
        <template class="header">Nama</template>
        <template>
            <div>[[item.nama]]</div>
        </template>
    </vaadin-grid-column>
    <vaadin-grid-column>
        <template class="header">Kelas</template>
        <template>
            <div>[[item.kelas]]</div>
        </template>
    </vaadin-grid-column>
    <vaadin-grid-column width="200px">
        <template class="header">Action</template>
        <template>
            <paper-button id="[[item.id]]" class="warning" on-
click="openEditModal">Edit
            </paper-button>
            <paper-button id="[[item.id]]" class="danger" on-
click="openDeleteModal">Delete
            </paper-button>
        </template>
    </vaadin-grid-column>
</vaadin-grid>
<!-- End of datatable -->

<!-- Crud view modal -->
<paper-dialog id="addModal" with-backdrop>
    <form-add success="{{success}}" response="{{response}}"></form-add>
</paper-dialog>
<paper-dialog id="editModal" with-backdrop>

```



```

        <form-
edit uid="{{uid}}" success="{{success}}" response="{{response}}"></form-edit>
        </paper-dialog>
        <paper-dialog id="deleteModal" with-backdrop>
        <form-
delete uid="{{uid}}" success="{{success}}" response="{{response}}"></form-delete>
        </paper-dialog>
        <!-- End of modal -->

    </div>
</template>

<script>
class MahasiswaPage extends Polymer.Element {
    static get is() { return 'mahasiswa-page'; }

    static get properties() {
        return {
            response: {
                type: Object,
            },
            keyword: {
                type: String,
                value: ''
            },
            url: {
                computed: '_computeUrl(keyword)'
            },
            method: {
                type: String,
                value: "GET"
            },
            success: String,
            uid: {
                type: Number,
                notify: true
            }
        }
    }

    _computeUrl(keyword) {
        if (keyword) {
            return ['/mahasiswa/search', keyword].join('/');
        } else {
            return '/api/mahasiswa';
        }
    }
}

```

```

    }
  }

  openAddModal() {
    this.$.addModal.open();
  }

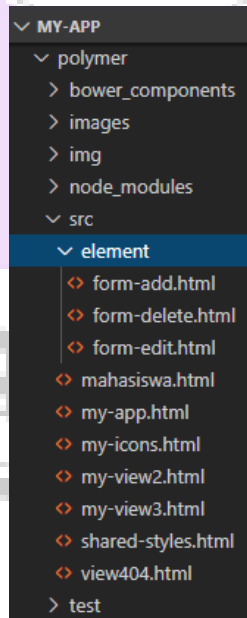
  openEditModal(e) {
    this.uid = e.target.id;
    this.$.editModal.open();
  }

  openDeleteModal(e) {
    this.uid = e.target.id;
    this.$.deleteModal.open();
  }
}

window.customElements.define(MahasiswaPage.is, MahasiswaPage);
</script>
</dom-module>

```

Buatlah folder baru dengan nama **element** di dalam direktori **polymer/src/**, jika sudah buatlah file **form-add.html**, **form-edit.html**, **form-delete.html**.



Berikut adalah script dari file-file tersebut:

**form-add.html**

```

<!--
@license
Copyright (c) 2016 The Polymer Project Authors. All rights reserved.
This code may only be used under the BSD style license found at http://polymer.gi
thub.io/LICENSE.txt
The complete set of authors may be found at http://polymer.github.io/AUTHORS.txt
The complete set of contributors may be found at http://polymer.github.io/CONTRIB
UTORS.txt
Code distributed by Google as part of the polymer project is also
subject to an additional IP rights grant found at http://polymer.github.io/PATENT
S.txt
-->

<link rel="import" href="../../../bower_components/polymer/polymer-element.html">

<link rel="import" href="../../../bower_components/paper-button/paper-button.html">
<link rel="import" href="../../../bower_components/paper-input/paper-input.html">

<link rel="import" href="../../../bower_components/iron-ajax/iron-ajax.html">
<link rel="import" href="../../../bower_components/iron-input/iron-input.html">

<link rel="import" href="../../shared-styles.html">

<dom-module id="form-add">
  <template>
    <style include="shared-styles">
      :host {
        display: block;

        padding: 10px;
      }
    </style>

    <h2>Tambah Data User</h2>
    <paper-input-container>
      <iron-input slot="input">
        <input type="text" value="{{formData.npm::input}}" placeholder="N
pm">
      </iron-input>
    </paper-input-container>
    <paper-input-container>
      <iron-input slot="input">
        <input type="text" value="{{formData.nama::input}}" placeholder="
Nama">
      </iron-input>
    </paper-input-container>
  </template>
</dom-module>

```

```

    </paper-input-container>
    <paper-input-container>
      <iron-input slot="input">
        <input type="text" value="{{formData.kelas::input}}" placeholder=
"Kelas">
      </iron-input>
    </paper-input-container>

    <div class="buttons">
      <paper-button raised class="link" on-
click="addMahasiswa" style="margin-top: 20px;" dialog-confirm>Tambah
      Data
    </paper-button>
    </div>

    <iron-
ajax auto id="mahasiswaAjax" url="{{url}}" method="{{method}}" handle-as="text"
      content-type="application/json" on-response="_handleInsert">
    </iron-ajax>
  </template>

  <script>
    class FormAdd extends Polymer.Element {
      static get is() { return 'form-add'; }

      static get properties() {
        return {
          formData: {
            type: Object,
            value: function () {
              return { id: 0, npm: '', nama: '', kelas: '' }
            }
          },
          url: {
            type: String,
            value: '/api/mahasiswa'
          },
          method: {
            type: String,
            value: 'GET'
          },
          success: {
            type: String,
            notify: true
          },

```

```

        response: {
            type: Object,
            notify: true
        }
    }
}

addMahasiswa() {
    this.url = '/api/mahasiswa';
    this.method = 'POST';
    this.success = 'Data mahasiswa berhasil ditambahkan';
    this.$.mahasiswaAjax.params = this.formData;
    this.formData = {};
}

_handleInsert(e) {
    this.method = 'GET';
    this.set('response', JSON.parse(e.detail.response));
}

}

window.customElements.define(FormAdd.is, FormAdd);
</script>
</dom-module>

```

#### form-edit.html

```

<link rel="import" href="../../bower_components/polymer/polymer-element.html">

<link rel="import" href="../../bower_components/paper-button/paper-button.html">
<link rel="import" href="../../bower_components/paper-input/paper-input.html">

<link rel="import" href="../../bower_components/iron-ajax/iron-ajax.html">
<link rel="import" href="../../bower_components/iron-input/iron-input.html">

<link rel="import" href="../../shared-styles.html">

<dom-module id="form-edit">
    <template>
        <style include="shared-styles">
            :host {
                display: block;

                padding: 10px;
            }

```

```

</style>

<h2>Ubah Data Mahasiswa</h2>

<input type="hidden" value="{{formData.id::input}}">
<paper-input-container>
  <iron-input slot="input">
    <input type="text" value="{{formData.npm::input}}" placeholder="N
pm">
  </iron-input>
</paper-input-container>
<paper-input-container>
  <iron-input slot="input">
    <input type="text" value="{{formData.nama::input}}" placeholder="N
Nama">
  </iron-input>
</paper-input-container>
<paper-input-container>
  <iron-input slot="input">
    <input type="text" value="{{formData.kelas::input}}" placeholder="
"Kelas">
  </iron-input>
</paper-input-container>

<div class="buttons">
  <paper-button raised class="link" on-
click="editMahasiswa" style="margin-top: 20px;" dialog-confirm>Ubah
    Data
  </paper-button>
</div>

<iron-
ajax auto id="getMahasiswaAjax" url="{{mhsurl}}" method="GET" handle-as="json"
  content-type="application/json" on-response="_handleGet">
</iron-ajax>
<iron-
ajax auto id="editMahasiswaAjax" url="{{url}}" method="{{method}}" handle-
as="text"
  content-type="application/json" on-response="_handleUpdate">
</iron-ajax>
</template>

<script>
  class FormEdit extends Polymer.Element {
    static get is() { return 'form-edit'; }

```

```

static get properties() {
    return {
        formData: {
            type: Object,
            value: function () {
                return { id: 0, npm: '', nama: '', kelas: '' }
            }
        },
        url: String,
        mhsurl: {
            computed: '_computeUrl(uid)'
        },
        method: String,
        success: {
            type: String,
            notify: true
        },
        response: {
            type: Object,
            notify: true
        },
        uid: Number,
    }
}

_computeUrl(uid) {
    return ['/api/mahasiswa', uid].join('/');
}

editMahasiswa() {
    this.url = '/api/mahasiswa';
    this.method = 'PUT';
    this.success = 'Data mahasiswa berhasil diubah';
    this.$.editMahasiswaAjax.params = this.formData;
    this.formData = {};
}

_handleUpdate(e) {
    this.method = 'GET';
    this.set('response', JSON.parse(e.detail.response));
}

_handleGet(e) {
    this.set('formData', e.detail.response.data);
}

```

```

    }

    }

    window.customElements.define(FormEdit.is, FormEdit);
  </script>
</dom-module>

```

## form-delete.html

```

<link rel="import" href="../../bower_components/polymer/polymer-element.html">
<link rel="import" href="../../bower_components/paper-button/paper-button.html">
<link rel="import" href="../../bower_components/iron-ajax/iron-ajax.html">

<link rel="import" href="../../shared-styles.html">

<dom-module id="form-delete">
  <template>
    <style include="shared-styles">
      :host {
        display: block;

        padding: 10px;
      }
    </style>

    <h2>Yakin hapus data?</h2>

    <div class="buttons">
      <paper-button class="link" dialog-dismiss>Batal</paper-button>
      <paper-button class="danger" on-
click="deleteMahasiswa" raised dialog-confirm>Hapus</paper-button>
    </div>

    <iron-ajax auto url="{{url}}" method="{{method}}" handle-
as="json" content-type="application/json"
      on-response="_handleDelete"></iron-ajax>
    </template>

    <script>
      class FormDelete extends Polymer.Element {
        static get is() { return 'form-delete'; }

        static get properties() {

```



```
        return {
            uid: Number,
            url: String,
            method: String,
            success: {
                type: String,
                notify: true
            },
            response: {
                type: Object,
                notify: true
            }
        }
    }

    deleteMahasiswa() {
        this.method = 'DELETE';
        this.url = '/api/mahasiswa/' + this.uid;
        this.success = 'Data mahasiswa berhasil dihapus';
    }

    _handleDelete(e) {
        this.method = 'GET';
        this.url = '/api/mahasiswa';
        this.set('response', e.detail.response);
    }
}

window.customElements.define(FormDelete.is, FormDelete);
</script>
</dom-module>
```



Buka terminal pada VSCode dengan menekan **CTRL + `**, setelah itu jalankan program dengan mengetikkan **go run main.go**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\golang\src\my-app> go run main.go
2020/02/26 14:09:30 Server berjalan di http://localhost:8081
|
```

Buka Halaman <http://localhost:8081>

