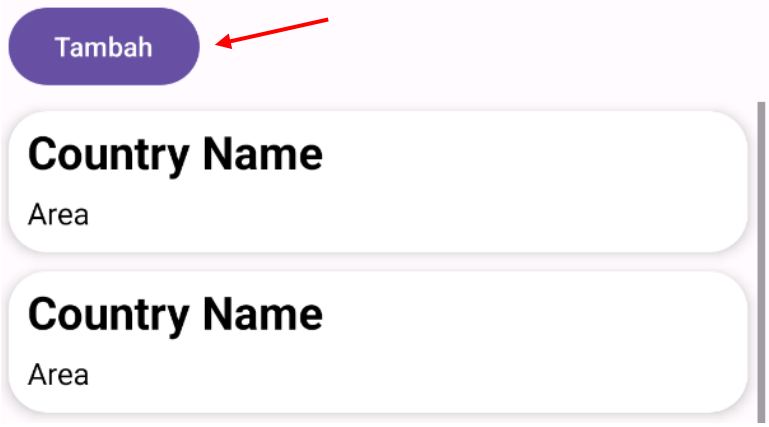
Mobile Programming 9 – Retrofit.3

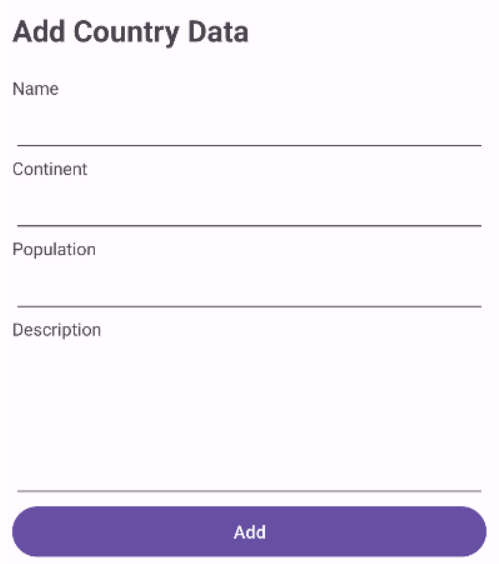
Pada pertemuan Retrofit kali ini, kita akan membahas bagaimana melakukan proses CREATE, UPDATE, dan DELETE pada API.

1. CREATE

Proses create diawali dengan menambah fungsi tombol 'Tambah' pada Main Activity menggunakan OnClick Listener untuk membawa ke Activity Add (perhatikan file layout dan kotlin class Main Activity).



Activity Add berisikan, field untuk mengisi data baru dan sebuah tombol 'Add'



Tombol Add akan diisi OnClick Listener yang menjalankan fungsi 'addCountryData()'. Fungsi ini akan mengambil isi inputan field yang kemudian akan dilakukan validasi pengecekan field kosong atau tidak (lihat kotlin class AddActivity). Jika, field sudah terisi semua, maka akan dilakukan pemanggilan RetrofitClient dengan menjalankan instance fungsi

‘addCountryDetail(input1, input2, input3, input4)’ dengan parameter data inputan sebelumnya.

Pada file Interface API, fungsi ‘addCountryDetail(input1, input2, input3, input4)’ menggunakan Anotasi ‘FormUrlEncoded’ dan menggunakan metode POST, kemudian parameter pada fungsi menggunakan anotasi ‘Field’ untuk kemudian isi parameter ini akan dikirimkan pada endpoint ‘add.php’. Callback yang digunakan adalah Data Class buatan sendiri ‘Response’.

```
@FormUrlEncoded
@POST("add.php")
fun addCountryDetail(
    @Field("country_name") country_name: String?,
    @Field("country_continent") country_continent: String?,
    @Field("country_population") country_population: String?,
    @Field("country_description") country_description: String?
): Call<Response>
```

Pada endpoint add.php, parameter yang dikirim sesuai dengan nama: “country_name”, “country_continent”, “country_population”, dan “country_description” akan ditangkap menggunakan metod POST. Yang kemudian, akan menjalankan Query ke Database, serta response apabila query berhasil atau tidak.

```
$name = $_POST['country_name'];
$continent = $_POST['country_continent'];
$population = $_POST['country_population'];
$description = $_POST['country_description'];

$query = "INSERT INTO country (name, continent, population, description)
          VALUES('$name', '$continent', '$population', '$description');";

$query_run = mysqli_query($CON, $query);

if ($query_run) {
    echo json_encode([
    "error" => false,
    "message" => 'Country has been successfully added!'
    ]);
} else {
    echo json_encode([
    "error" => true,
    "message" => 'Country failed to be added!'
    ]);
}
```

2. UPDATE

(Perhatikan file kotlin class EditActivity!) Untuk Update lihat Activity Detail, kemudian ada sebuah tombol ‘edit’ yang akan membawa ke Activity Edit. Fungsi di Activity Edit kurang lebih sama, yaitu form, bedanya form sudah terisi detail data dan siap diubah jika ingin diubah. Pada activity edit terdapat tombol ‘Update’ yang akan menjalankan OnClick

Listener untuk mengubah data yang hendak di-edit. Event handler akan memanggil fungsi ‘updateCountryDetail(countryId)’. Fungsi ini menerima parameter ID country yang hendak diedit. Fungsi ini mengecek apakah field kosong, jika tidak kosong, maka akan memanggil RetrofitClient dengan instance ‘updateCountryDetail(input1, input2, input3, input4, input5)’. Penambahan satu input lagi adalah nomor ID data yang hendak diedit di database.

Pada file Interface API, fungsi ‘updateCountryDetail(input1, input2, input3, input4, input5)’ menggunakan Anotasi ‘FormUrlEncoded’ dan menggunakan metode POST, kemudian parameter pada fungsi menggunakan anotasi ‘Field’ untuk kemudian isi parameter ini akan dikirimkan pada endpoint ‘update.php’. Callback yang digunakan adalah Data Class buatan sendiri ‘Response’.

```
@FormUrlEncoded
@POST("update.php")
fun updateCountryDetail(
    @Field("country_id") country_id: String?,
    @Field("country_name") country_name: String?,
    @Field("country_continent") country_continent: String?,
    @Field("country_population") country_population: String?,
    @Field("country_description") country_description: String?
): Call<Response>
```

Pada endpoint update.php, parameter yang dikirim sesuai dengan nama: “country_id”, “country_name”, “country_continent”, “country_population”, dan “country_description” akan ditangkap menggunakan metod POST. Yang kemudian, akan menjalankan Query ke Database, serta response apabila query berhasil atau tidak.

```
$id = $_POST['country_id'];
$name = $_POST['country_name'];
$continent = $_POST['country_continent'];
$population = $_POST['country_population'];
$description = $_POST['country_description'];

$query = "UPDATE country SET
name = '$name',
continent = '$continent',
population = '$population',
description = '$description'
WHERE id = '$id';";

$query_run = mysqli_query($CON, $query);

if ($query_run) {
    echo json_encode([
    "error" => false,
    "message" => 'Country has been successfully updated!'
    ]);
} else {
    echo json_encode([
    "error" => true,
    "message" => 'Country failed to be updated!'
    ]);
}
```

3. DELETE

(Perhatikan file kotlin class DetailActivity!) Untuk Delete lihat Activity Detail, kemudian ada sebuah tombol 'Delete' yang akan menjalankan OnClick Listener untuk menghapus data yang hendak dihapus. Event handler akan memanggil fungsi 'deleteCountry (countryId)'. Fungsi ini menerima parameter ID country yang hendak dihapus. Fungsi ini akan memanggil RetrofitClient dengan instance 'deleteCountryDetail (input1)'. Input ini adalah Id data yang hendak dikirim ke endpoint untuk dihapus.

Pada file Interface API, fungsi 'deleteCountryDetail(input1)' menggunakan Anotasi 'FormUrlEncoded' dan menggunakan metode POST, kemudian parameter pada fungsi menggunakan anotasi 'Field' untuk kemudian isi parameter ini akan dikirimkan pada endpoint 'delete.php'. Callback yang digunakan adalah Data Class buatan sendiri 'Response'.

```
@FormUrlEncoded
@POST("delete.php")
fun deleteCountryDetail(
    @Field("country_id") country_id: String?
): Call<Response>
```

Pada endpoint delete.php, parameter yang dikirim sesuai dengan nama: "country_id" akan ditangkap menggunakan metod POST. Yang kemudian, akan menjalankan Query ke Database, serta response apabila query berhasil atau tidak.

```
$id = $_POST['country_id'];

$query = "DELETE FROM country WHERE id = '$id';";

$query_run = mysqli_query($CON, $query);

if ($query_run) {
      echo json_encode([
      "error" => false,
      "message" => 'Country has been successfully deleted!'
      ]);
  } else {
      echo json_encode([
      "error" => true,
      "message" => 'Country failed to be deleted!'
      ]);
  }
```

***NB:**

Pada penerapan REST API sesungguhnya, proses Update dan Delete seharusnya menggunakan metode @PUT dan @DELETE. Namun, dikarenakan PHP hanya memiliki variabel global bawaan $_GET dan $_POST serta tidak memiliki variabel global untuk $_PUT dan $_DELETE, maka untuk mengganti pengiriman data untuk proses Update dan Delete dapat diganti dengan metode POST (untuk pemberian contoh kali ini). Untuk penerapan sesungguhnya tetap harus menggunakan metode masing-masing sesuai fungsinya! Untuk

endpoint menggunakan PHP, dapat dilakukan pembuatan variabel global $_PUT dan $_DELETE sebelum menerapkan REST API nya.