LARAVEL 8 REST SERVER API

Pertemuan 1

1. Install Laravel 8
2. Install Jetstream yang livewire , dan sudah termasuk Laravel Sanctum
3. Pergi ke file `app/Http/Kernel.php` tambahkan.

```
'api' => [
    \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
    'throttle:api',
    \Illuminate\Routing\Middleware\SubstituteBindings::class,
],
```

4. Pergi ke file routes/api.php dan masukan

```
use App\Http\Controllers\API\AuthController;
```

```
Route::post('/login',[AuthController::class,'login']);
```

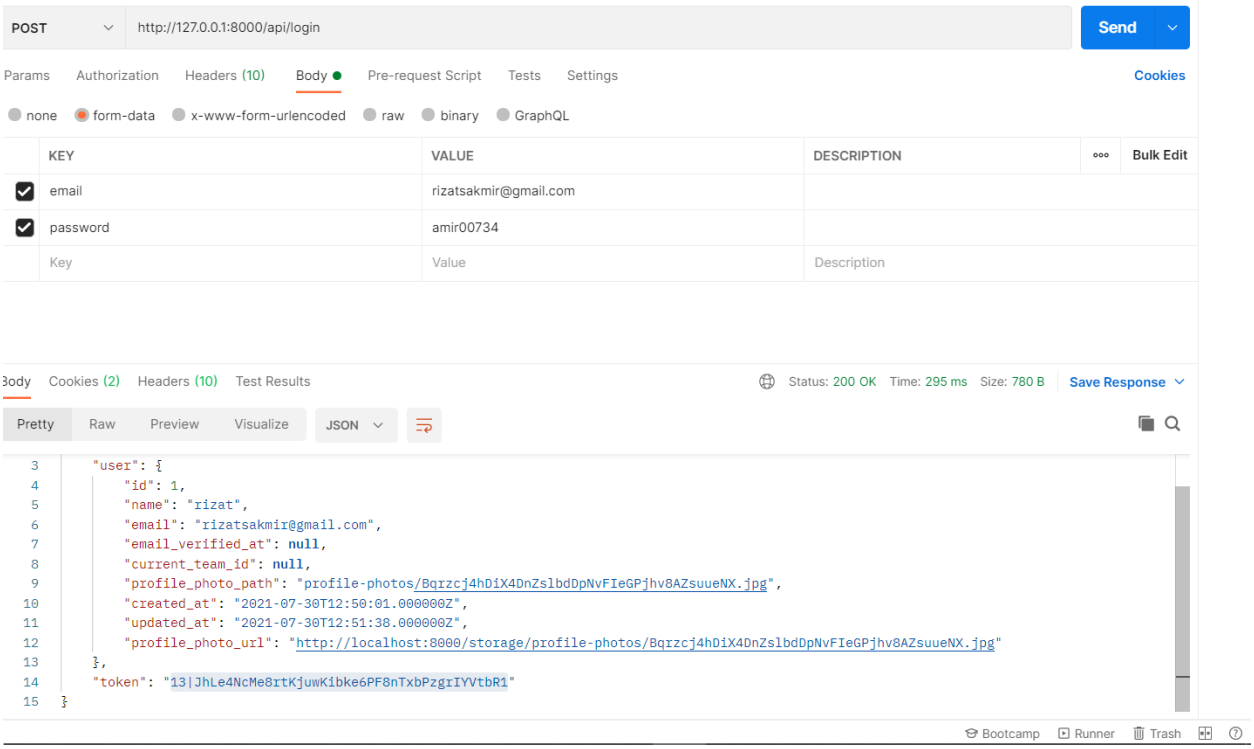
5. Lalu kita buat controllernya

```
php artisan make:controller API/AuthController
```

6. Pergi ke file App/Http/Controllers/API/AuthController.php dan masukan codingan.

```php
namespace App\Http\Controllers\API;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\User;
use Illuminate\Support\Facades\Hash;

class AuthController extends Controller
{
    public function login(Request $request)
    {
        $user = User::where('email',$request->email)->first();
        if (!$user || !Hash::check($request->password, $user->password)){
            return response()->json([
                'message' => 'UnAuthorized'
            ],401);
        }
        $token = $user->createToken('token-name')->plainTextToken;
            return response()->json([
                'message' => 'success',
                'user' => $user,
                'token' => $token
            ],200);
    }
}
```

7. Coba di postman



Pertemuan 2

1. Pergi ke file routes/api.php kita tambahkan.

```
use App\Http\Controllers\API\AuthController;

use App\Http\Controllers\API\FormController;
```

```
Route::group(['middleware' => 'auth:sanctum'],function(){
    Route::get('/form',[FormController::class,'index']);
    Route::get('/logout',[AuthController::class,'logout']);
});
```

2. Kita membuat controller .

```
php artisan make:controller API/FormController
```

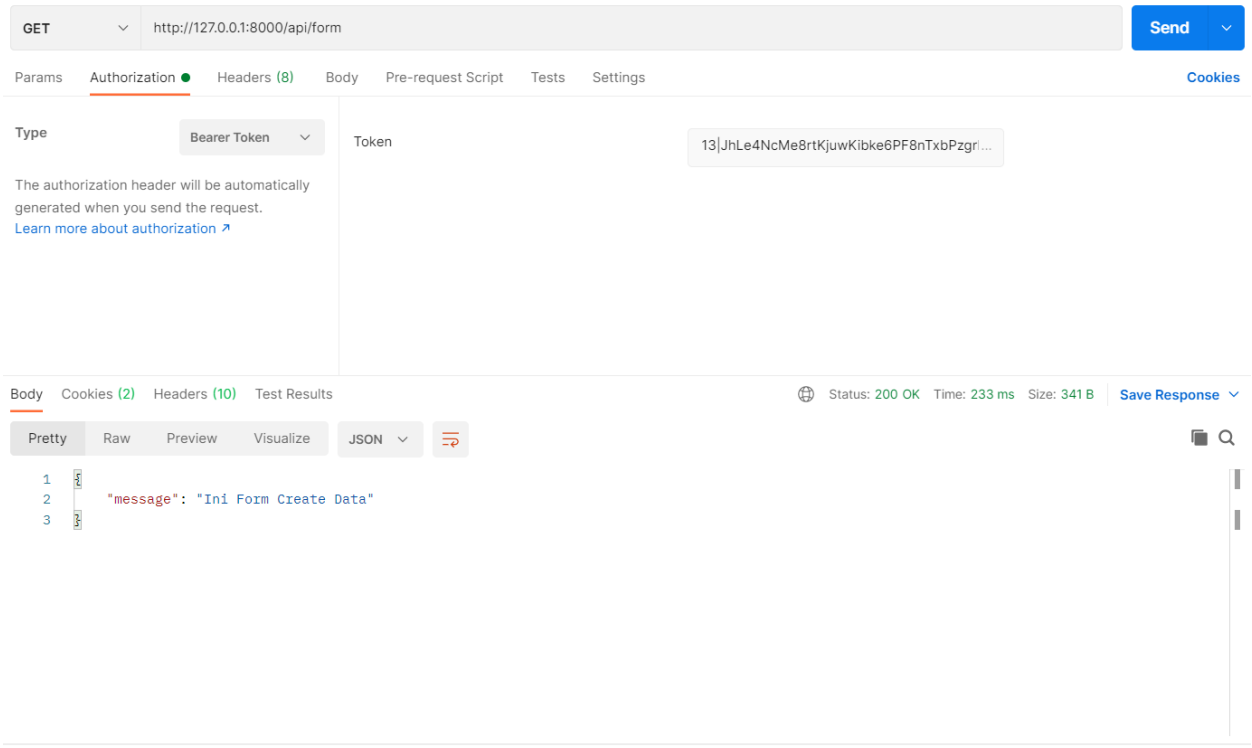
3. Pergi ke file App/Http/Controller/API/FormController.php kita tambahkan.
   Untuk bisa ke Form Create Data

```php
public function index()
{
    return response()->json([
        'message' => 'Ini Form Create Data'
    ],200);
}
```

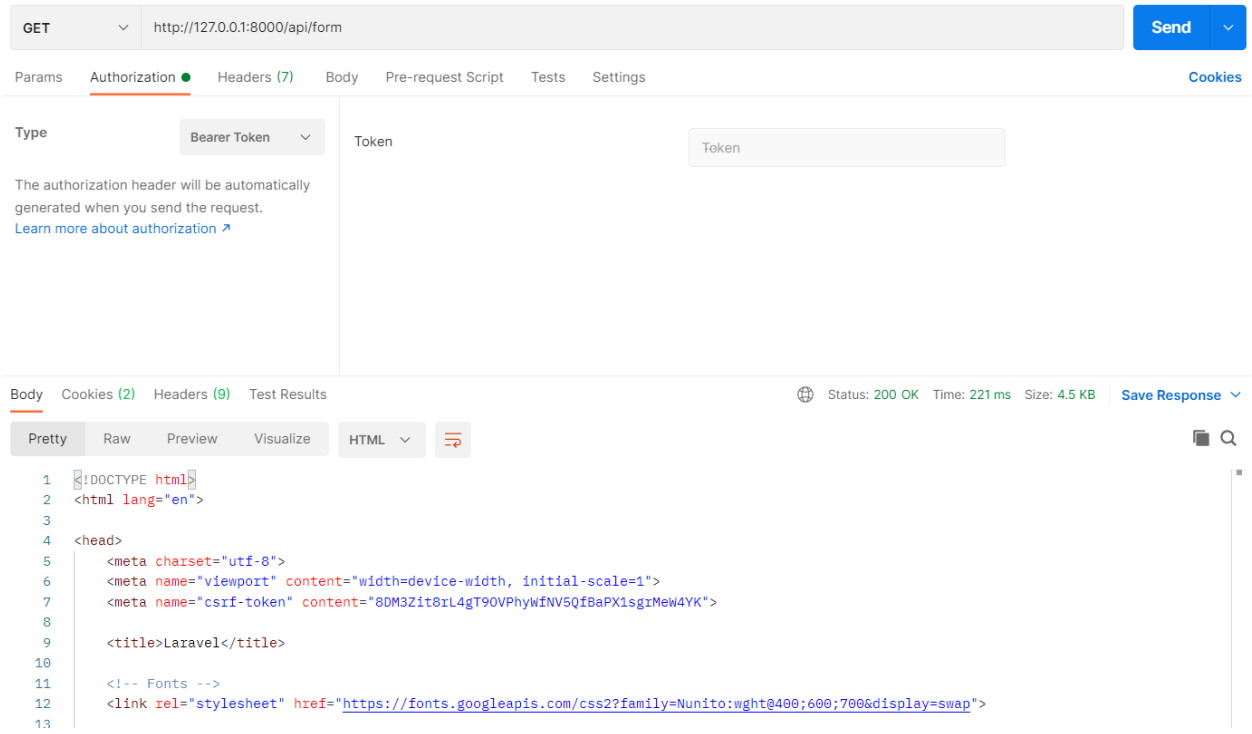
4. Pergi ke file App/Http/Controller/API/AuthController.php kita tambahkan. Untuk membuat API Logoutnya ,menghapus accsess tokenya .

```php
public function logout(Request $request)
{
    $user = $request->user();
    $user->currentAccessToken()->delete();
    return response()->json([
        'message' => 'Berhasil Logout'
    ],200);
}
```

Nah ini untuk bisa mengakses nya harus menggunakan token yang ada pada saat berhasil login.



Kalau tidak masukan token tidak bisa mengakses, hanya tampil halaman form login.



Pertemuan 3

Create Data

1. Pergi ke halaman routes/api.php dan tambahkan didalam Route Group.

```php
Route::post('/create',[FormController::class,'create']);
```

2. Pergi ke file App/Http/Controller/API/FormController.php kita tambahkan.

```php
public function create(Request $request)
{
    $request->validate([
        'nama' => 'required|min:5',
        'nim' => 'required|digits:8|numeric',
        'jurusan' => 'required'
    ]);

    $student = new Student;
    $student->nama = $request->nama;
    $student->nim = $request->nim;
    $student->jurusan = $request->jurusan;
    $student->save();

    return response()->json([
        'message' => 'Data Berhasil Disimpan',
        'data' => $student
    ],200);
}
```

3. Membuat model sekalian migration Student.

```
php artisan make:model Student -m
```

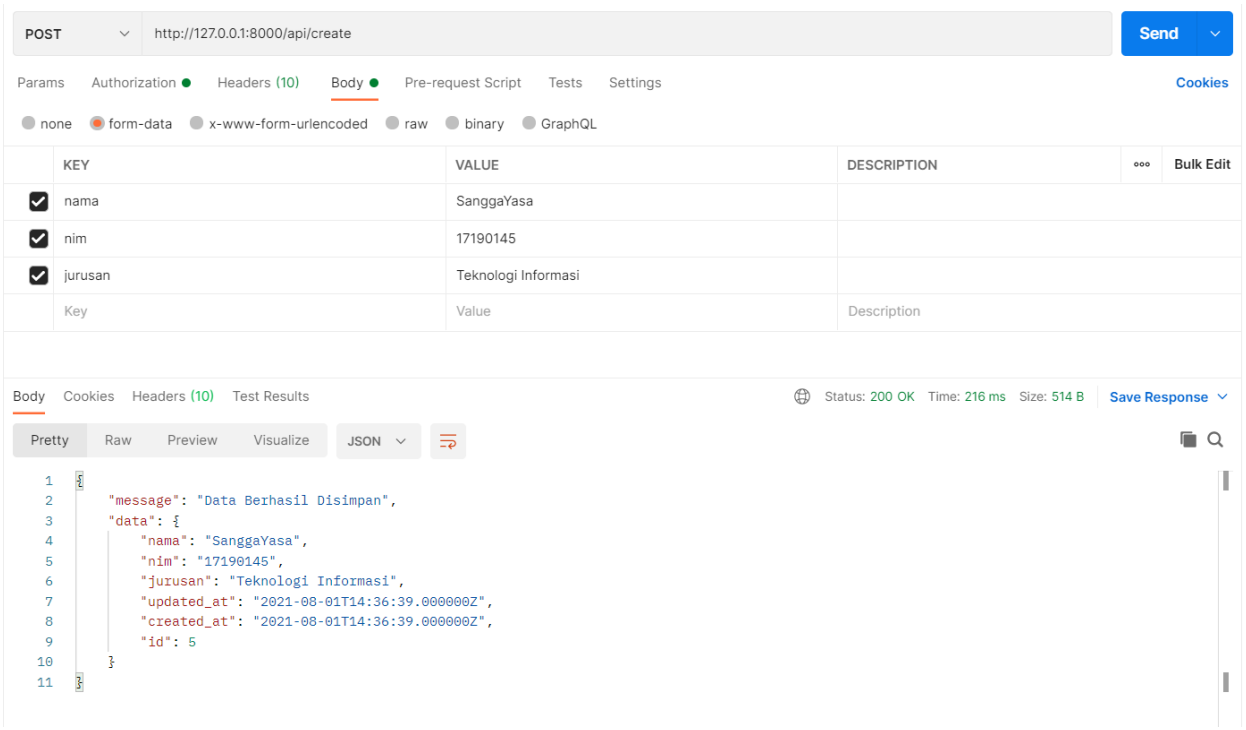
4. Pergi ke file database/migrations/create_students_table.php dan tambahkan .

```
public function up()
{
    Schema::create('students', function (Blueprint $table) {
        $table->id();
        $table->string('nama',40);
        $table->integer('nim'); //minimal 11 caracter
        $table->string('jurusan',40);
        $table->timestamps();
    });
}
```

5. Pergi ke file app/Models/Student.php dan tambahkan .

```
class Student extends Model
{
    use HasFactory;
    protected $guarded = [];
}
```

6. Coba di postman untuk create data , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



Pertemuan 3

Update Data

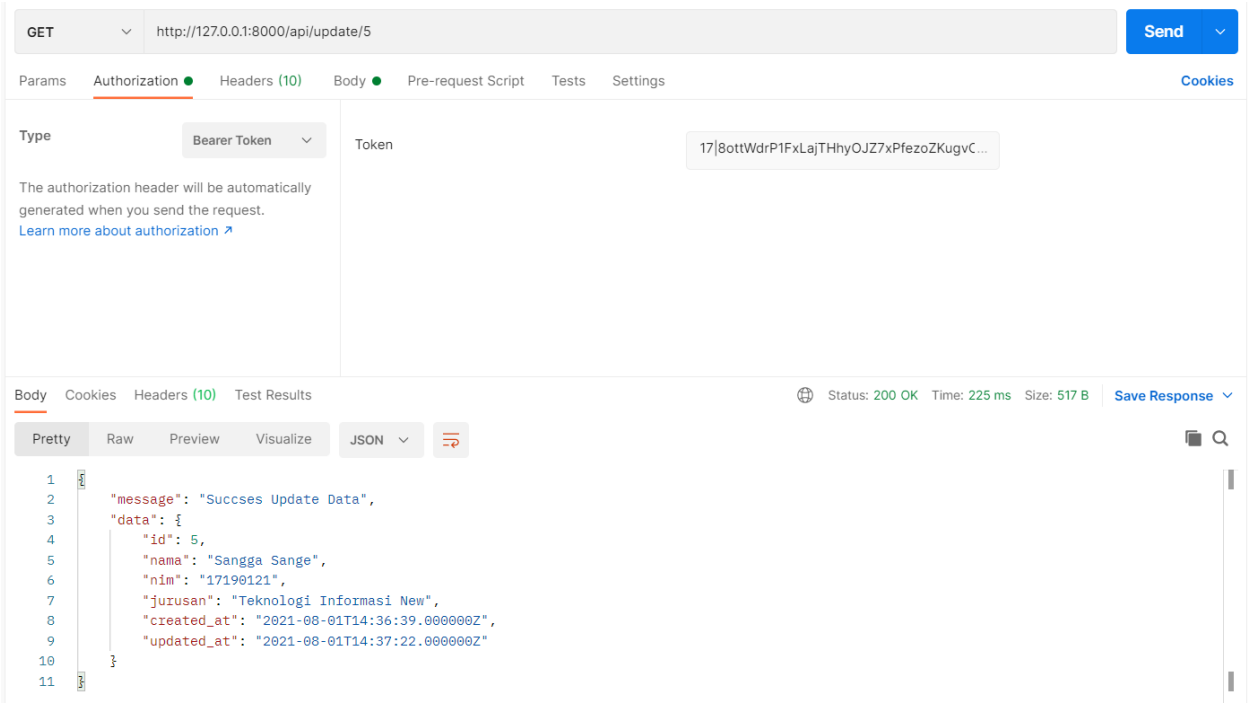
1. Pergi ke halaman routes/api.php dan tambahkan didalam Route Group.

```
Route::get('/update/{id}',[FormController::class,'getUpdate']);
```

2. Pergi ke file App/Http/Controller/API/FormController.php kita tambahkan.

```php
public function getUpdate($id)
{
    $student = Student::find($id);
    if($student != null){
        return response()->json([
            'message' => 'Succses',
            'data' => $student
        ],200);
    }else{
        return response()->json([
            'message' => 'Data Not Found !',
            'data' => $student
        ],404);
    }
}
```

3. Coba di postman untuk get data update , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



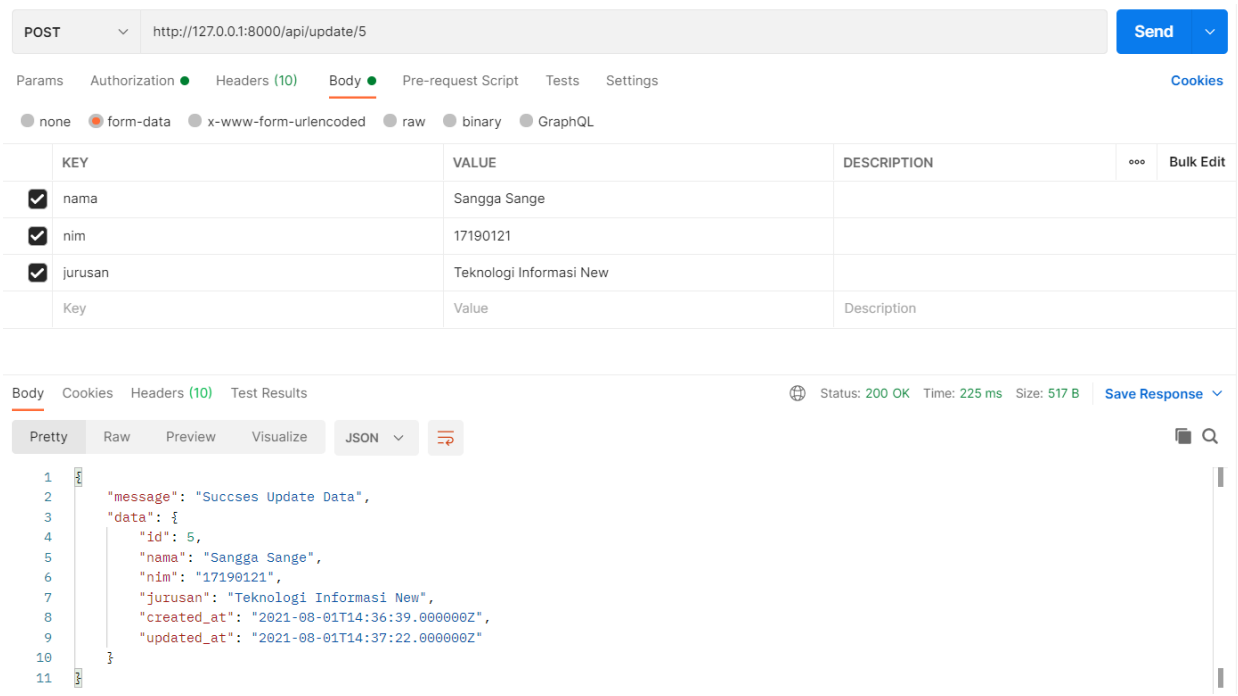
4. Pergi ke halaman routes/api.php dan tambahkan didalam Route Group.

```
Route::post('/update/{id}',[FormController::class,'Update']);
```

5. Pergi ke file App/Http/Controller/API/FormController.php kita tambahkan.

```php
public function update(Request $request,$id)
{
    $student = Student::find($id);

    $request->validate([
        'nama' => 'required|min:5',
        'nim' => 'required|digits:8|numeric',
        'jurusan' => 'required'
    ]);

    $student->update([
        'nama' => $request->nama,
        'nim' => $request->nim,
        'jurusan' => $request->jurusan
    ]);

    return response()->json([
        'message' => 'Succses Update Data',
        'data' => $student
    ],200);
}
```

6. Coba di postman untuk update data, masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



Pertemuan 4

Delete Data

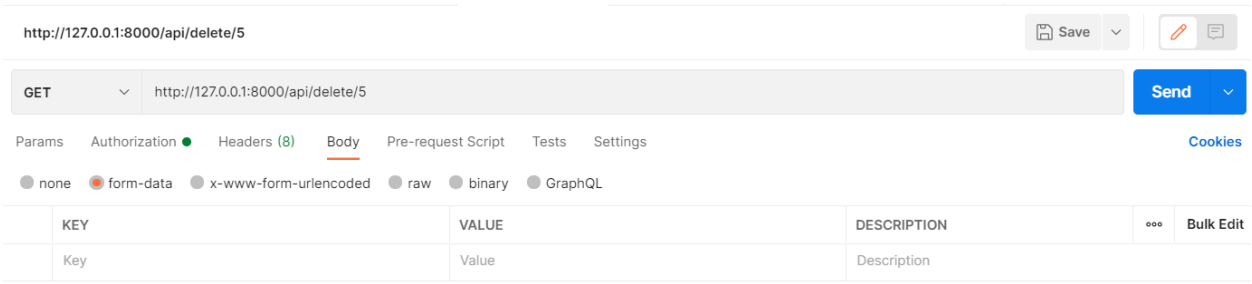
1. Pergi ke halaman routes/api.php dan tambahkan didalam Route Group.

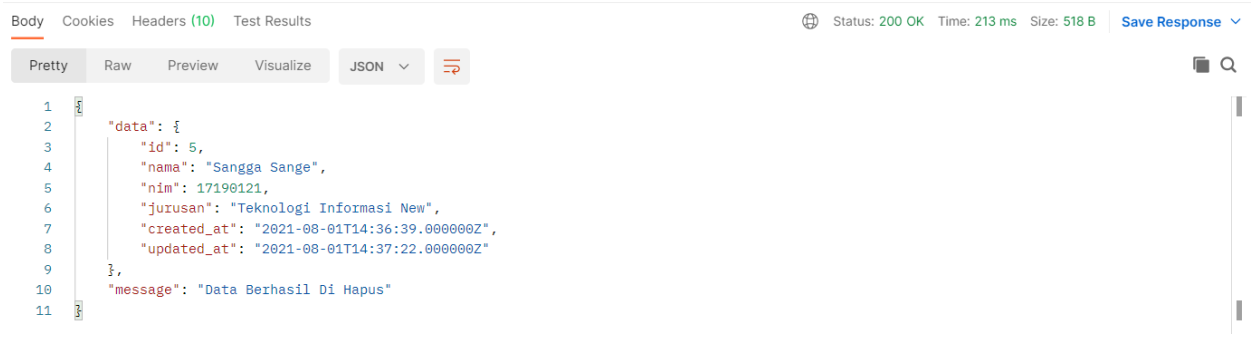
```
Route::get('/delete/{id}',[FormController::class,'delete']);
```

2. Pergi ke file App/Http/Controller/API/FormController.php kita tambahkan.

```php
public function delete($id)
{
    $student = Student::find($id);
    if($student != null){
        Student::find($id)->delete();
        return response()->json([
            'data' => $student,
            'message' => 'Data Berhasil Di Hapus'
        ],200);
    }else{
        return response()->json([
            'message' => 'Data Not Found'
        ],404);
    }
}
```

3. Coba di postman untuk delete data , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .





Pertemuan 5

Create Data Multiple

1. Kita membuat model dan migration dengan nama score .

```
php artisan make:model score -m
```

2. Pergi ke file database/migrations/create_students_table.php dan tambahkan .

```php
public function up()
{
    Schema::create('scores', function (Blueprint $table) {
        $table->id();
        $table->integer('student_id');
        $table->string('mata_pelajaran',50);
        $table->integer('nilai');
        $table->timestamps();
    });
}
```

3. Pergi ke file app/Models/Student.php dan tambahkan .

```php
class Score extends Model
{
    use HasFactory;
    protected $guarded = [];
}
```

4. Kita migrate , “php artisan migrate”.

5. Membuat ScoreController .

```
php artisan make:controller API/ScoreController
```

6. Pergi ke file routes/api.php dan tambahkan ini.

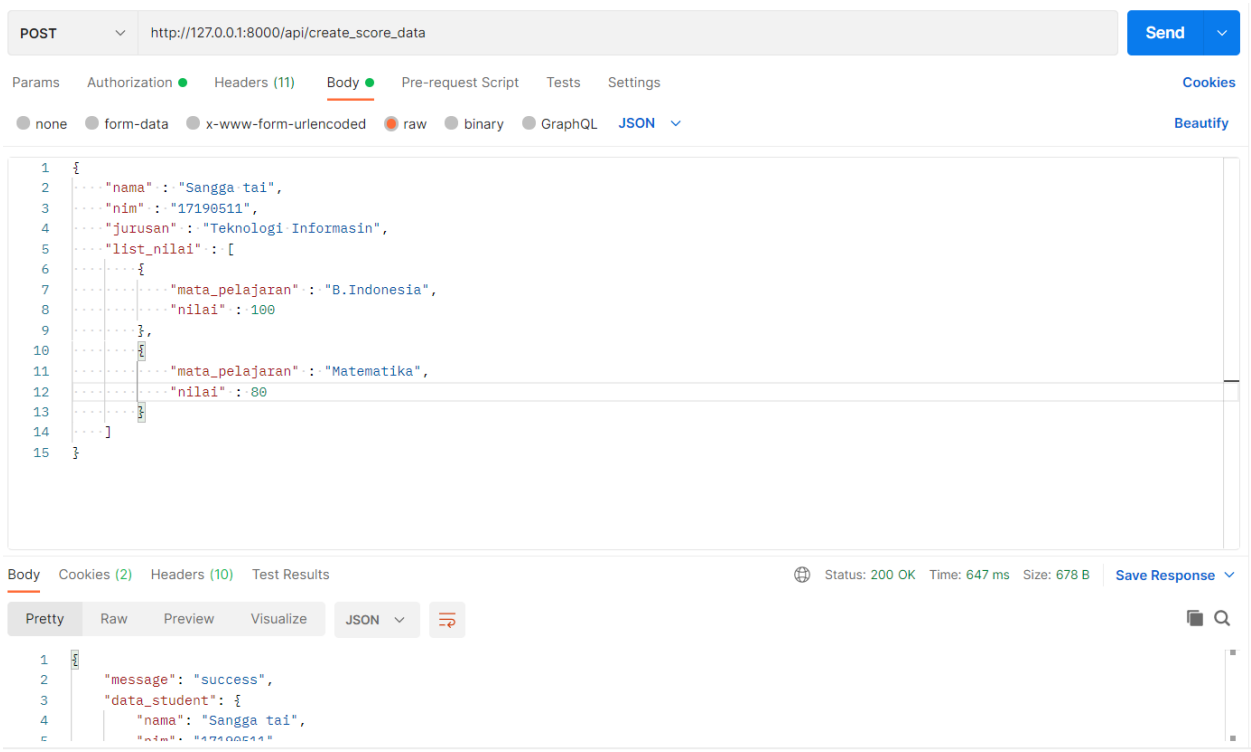
```php
use App\Http\Controllers\API\ScoreController;
```

```php
// CRUD MultiTable

Route::post('/create_score_data',[ScoreController::class,'create']);
```

7. Pergi ke file app/Http/Controllers/API/ScoreController.php dan tambahkan.

```php
public function create(Request $request)
{

    $request->validate([
        'nama' => 'required|min:5',
        'nim' => 'required|digits:8|numeric',
        'jurusan' => 'required'
    ]);

    $student = new Student;
    $student->nama = $request->nama;
    $student->nim = $request->nim;
    $student->jurusan = $request->jurusan;
    $student->save();

    foreach($request->list_nilai as $nilai => $value){
        $score = array(
            'student_id' => $student->id,
            'mata_pelajaran' => $value['mata_pelajaran'],
            'nilai' => $value['nilai']
        );
        $score = Score::create($score);
    }

    return response()->json([
        'message' => 'success',
        'data_student' => $student,
        'nilai_student'=> $score
    ],200);
}
```

4. Coba di postman untuk create data multiple , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



PERTEMUAN 6

UPDATE DATA RELASIONSHIP

1. Pergi ke file routes/api.php tambahkan ini .

```
Route::get('/update_score_dataStudent/{id}',[ScoreController::class,'getUpdate']);
```

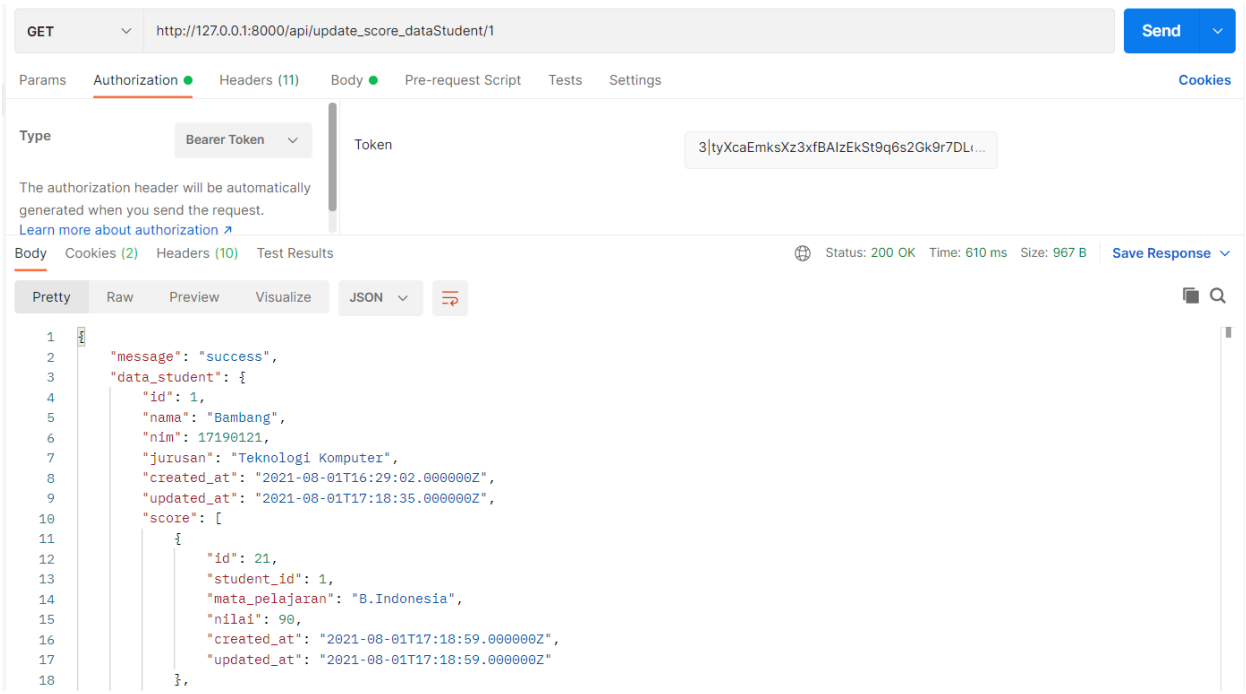
2. Pergi ke app/Http/Models/Student.php tambahkan ini . untuk relasi join , hasMany = OneToMany.

```php
class Student extends Model
{

    use HasFactory;

    protected $guarded = [];


    public function score()
    {
        return $this->hasMany('App\Models\Score', 'student_id');
    }
}
```

3. Pergi ke file app/Http/Controllers/API/ScoreController.php dan tambahkan ini .

```php
public function getUpdate($id)
{
    $student = Student::with('score')->where('id',$id)->first();
    return response()->json([
        'message' => 'success',
        'data_student' => $student
    ],200);
}
```

4. Coba di postman untuk get data update multiple , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



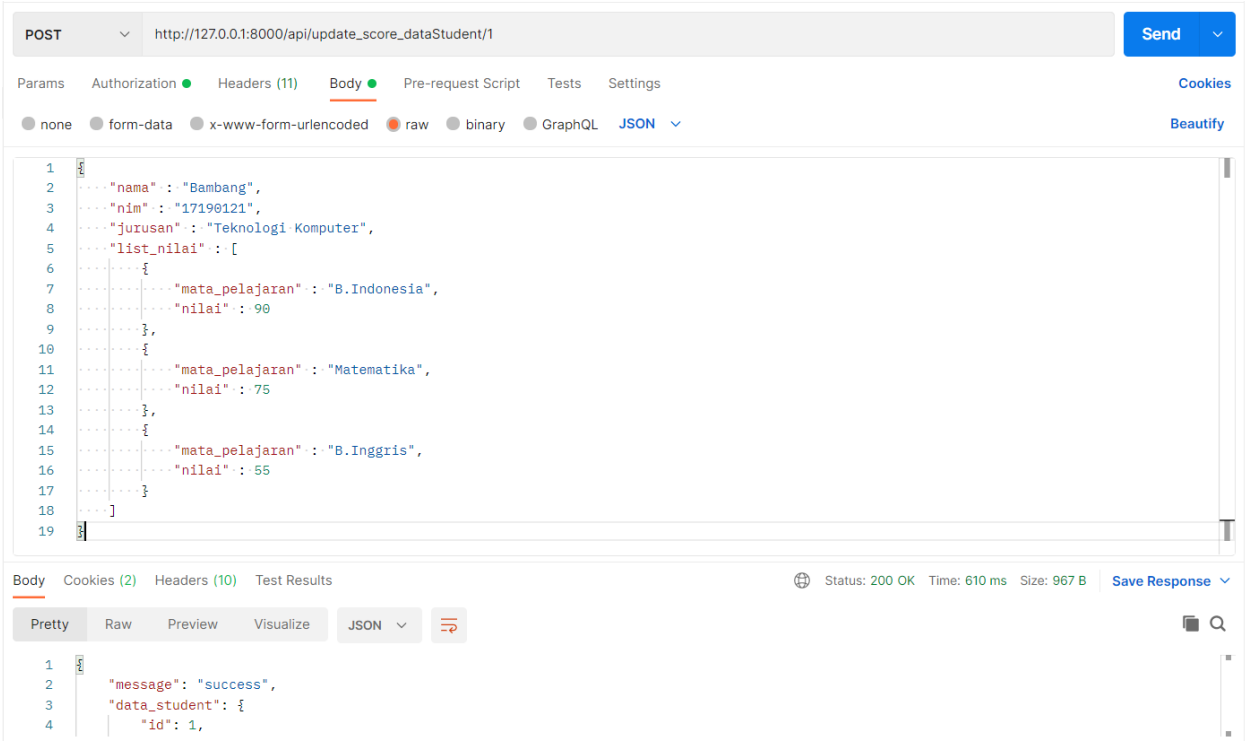
5. Pergi ke file routes/api.php tambahkan ini .

```
Route::post('/update_score_dataStudent/{id}',[ScoreController::class,'update']);
```

6. Pergi ke file app/Http/Controllers/API/ScoreController.php dan tambahkan ini .

```php
public function update(Request $request, $id)
{
    $student = Student::find($id);
    $student->update([
        'nama' => $request->nama,
        'nim' => $request->nim,
        'jurusan' => $request->jurusan
    ]);

    Score::where('student_id',$id)->delete();

    foreach($request->list_nilai as $nilai => $value){
        $score = array(
            'student_id' => $id,
            'mata_pelajaran' => $value['mata_pelajaran'],
            'nilai' => $value['nilai']
        );
        $score = Score::create($score);
    }

    return response()->json([
        'message' => 'success',
        'data_student' => $student,
        'nilai_student'=> $score
    ],200);

}
```

7. Coba di postman untuk update data multiple , masukan tokenya di Authorization dengan type Bearer Token , dan masukan di Headers yaitu ‘Accept : Aplication/json’ .



SELESAI