

Tugas Praktikum VIII

(Kelompok TUBES)

Ahmad Azwar Annas (1301180185)

Muhammad Rafif Ghani (1301184052)

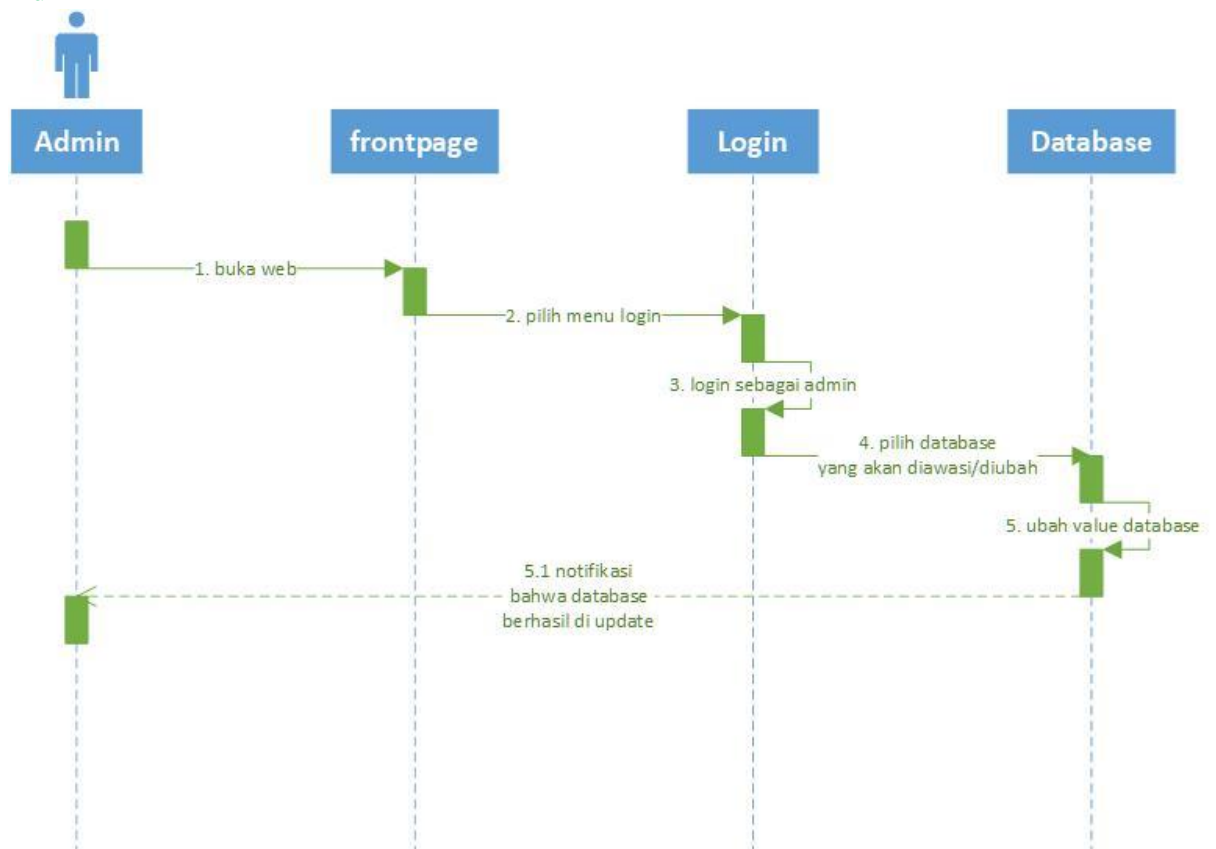
Gilang Ramadhan (1301184376)

TUGAS PENDAHULUAN - Membuat *SEQUENCE DIAGRAM*

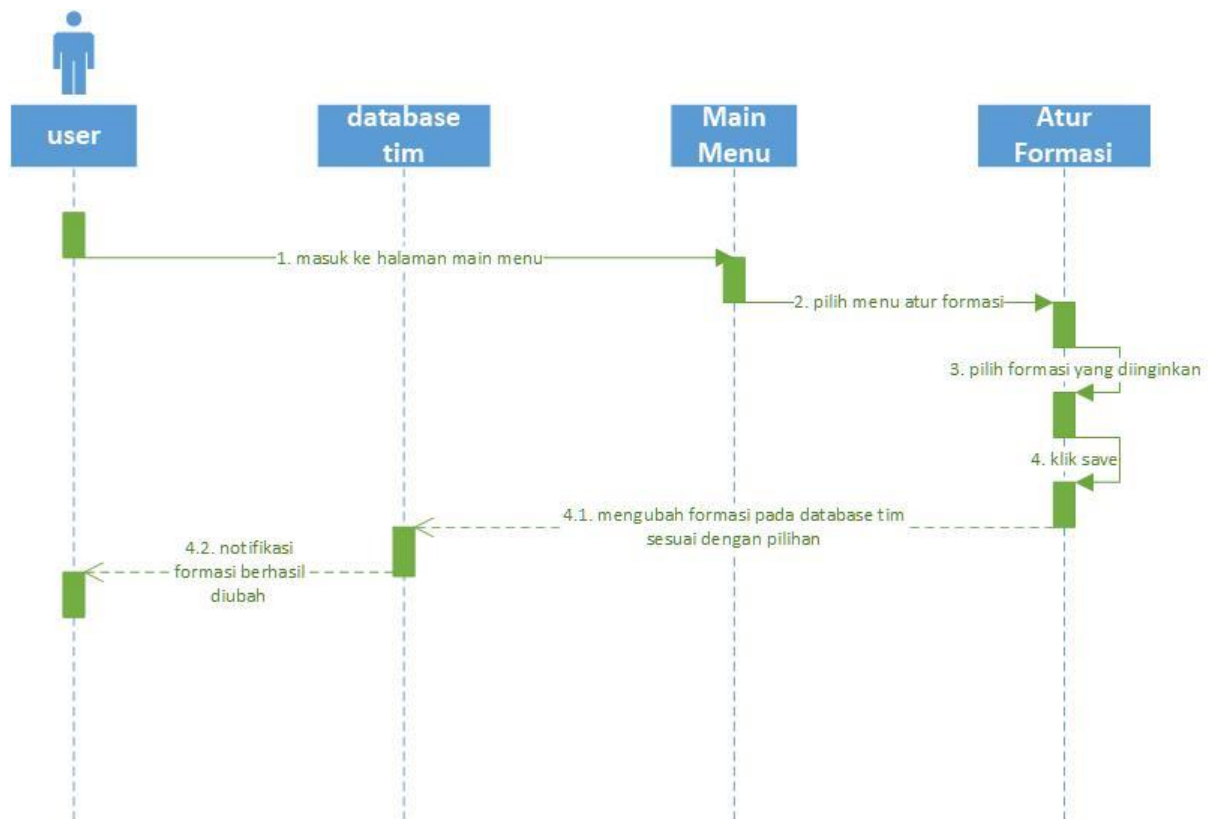
1. Periksa kembali apakah di DPPL Kelompok Anda terdapat *Sequence Diagram* untuk setiap *USE CASE* yang ada di *USE CASE DIAGRAM*. (Misal ada 5 *use case*, maka perlu dibuat 5 *Sequence Diagram*).
2. Jika belum ada, buatlah *SEQUENCE DIAGRAM* di DPPL Anda.
3. Jika sudah ada, periksa kembali kelengkapan dan keakuratannya. Laporkan dokumen DPPL dengan *Sequence Diagram* yang sudah lengkap pada Tugas Pendahuluan.

Jawab : pada DPPL sudah terdapat 4 *Sequence Diagram* yaitu : Jual Pemain, Beli Pemain, Find Match, TopUp. Maka masih ada 5 *use case* yang belum ada *Sequence Diagram*nya yaitu : Admin, Atur Formasi, Login, Register, Wishlist. Berikut *sequence diagram* yang telah dibuat

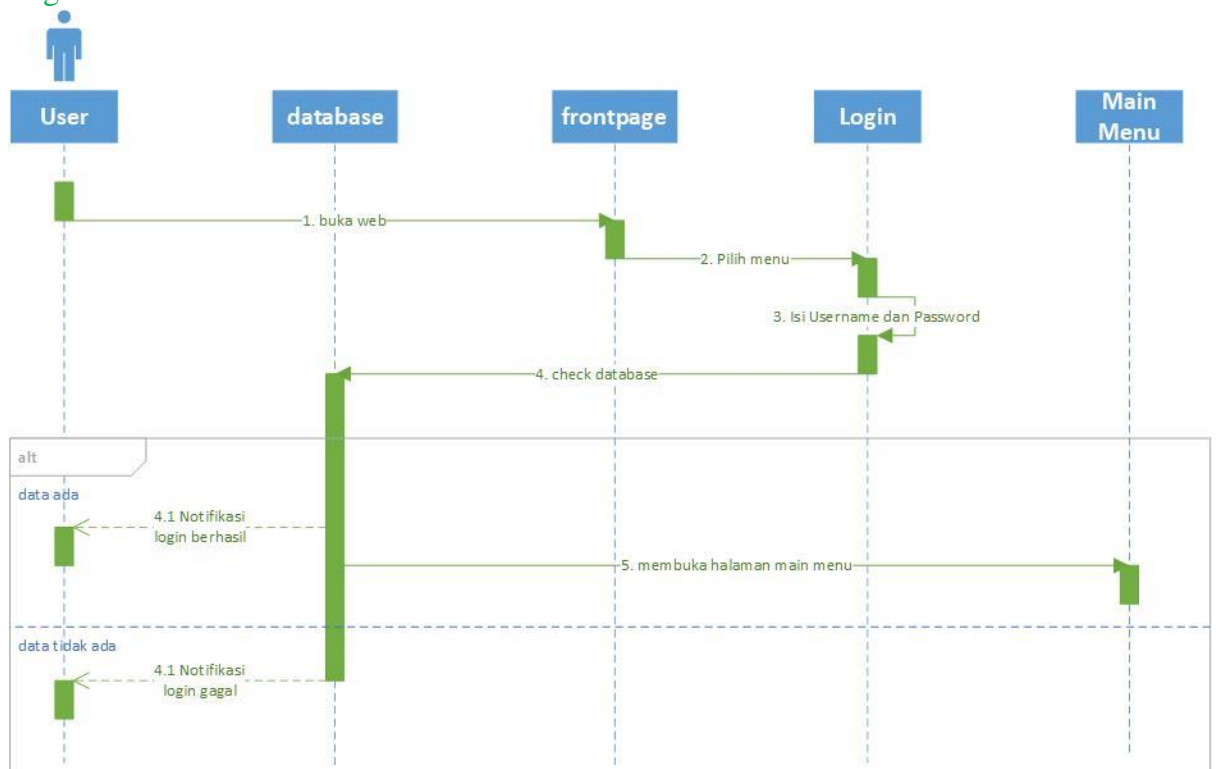
- Admin



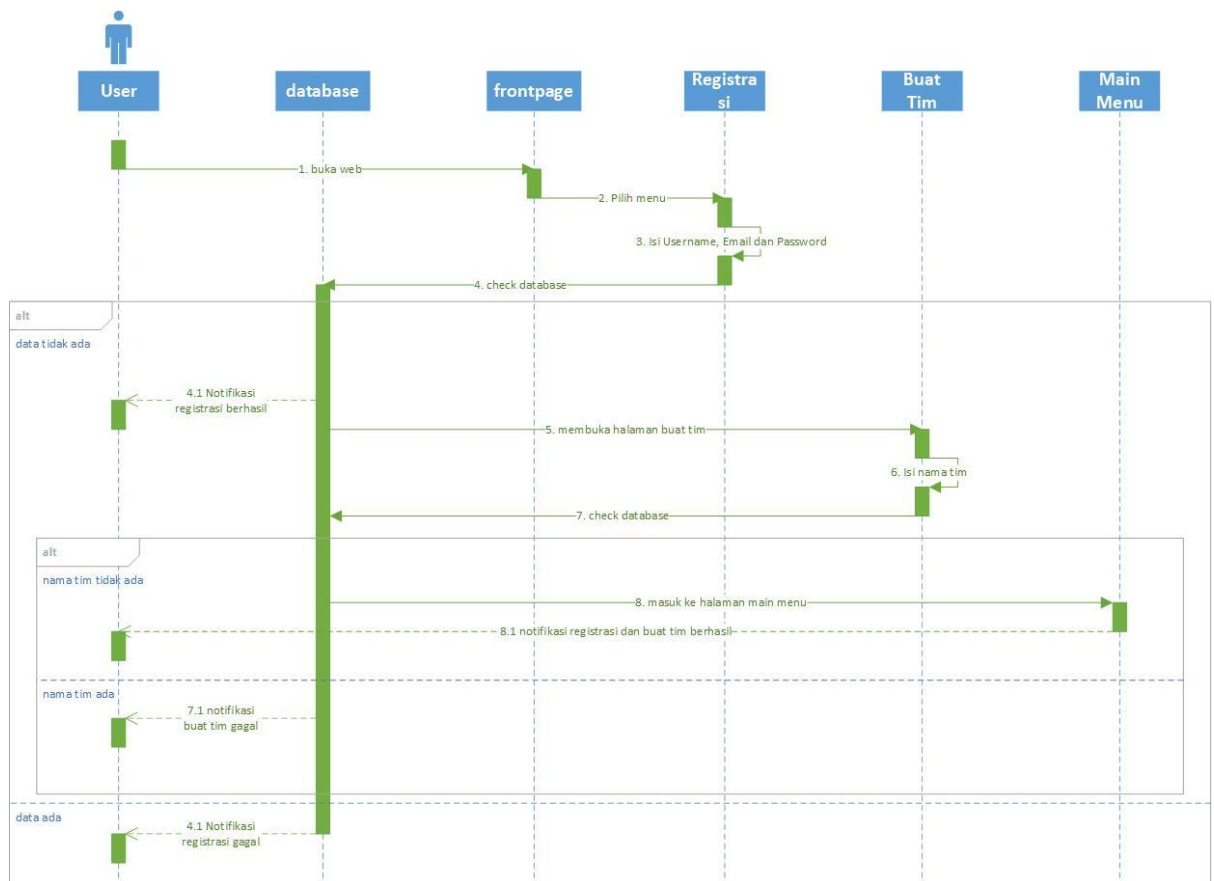
- Atur Formasi



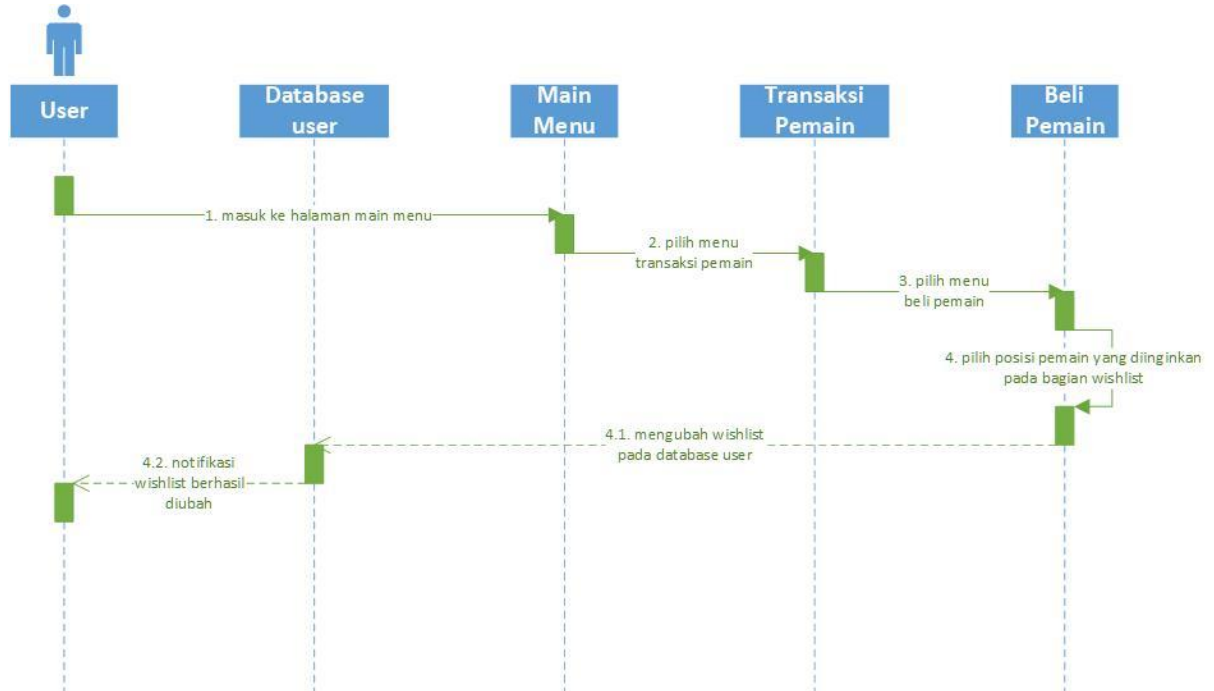
- Login



- Register



- Wishlist



TUGAS LAB

Sequence Diagram pada umumnya menggambarkan interaksi *Actor* dengan sistem dalam satu rangkaian masukan dan keluaran. Masukan dari *Actor* via *IMK*, akan direspon oleh aplikasi melalui pemanggilan *method-method* dari *class-class* yang saling berinteraksi, yang jika

diperlukan sampai perlu mengakses tabel basis data yang relevan dengan SQL yang di-"*embed*" di bahasa pemrograman, lalu memberikan hasil pengaksesannya ke *Actor* via IMK lagi.

Implementasikan setidaknya satu *Sequence Diagram* yang dapat bekerja sesuai deskripsi di atas.

Jawab : Sequence Diagram yang di implementasikan yaitu Login, Register dan Buat Tim

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use App\Models\User;
use Illuminate\Foundation\Auth\RegistersUsers;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;

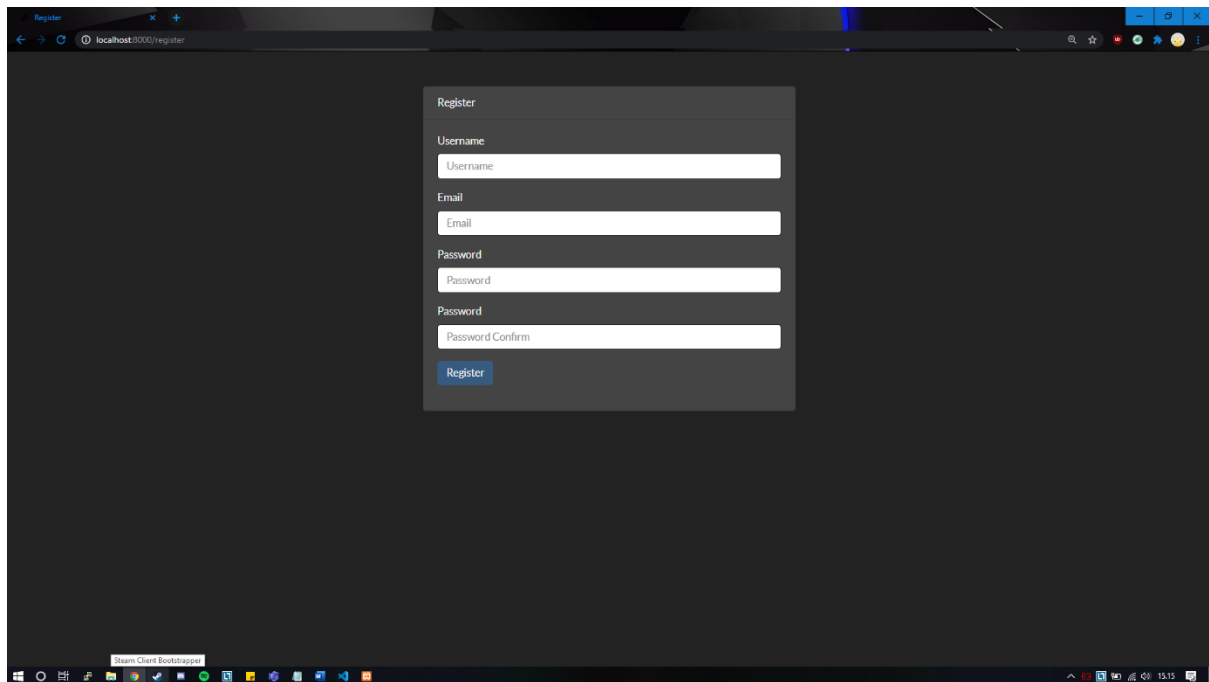
class RegisterController extends Controller
{
    use RegistersUsers;

    protected $redirectTo = RouteServiceProvider::HOME;

    public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => ['required', 'string', 'max:255'],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'password' => ['required', 'string', 'min:8', 'confirmed'],
        ]);
    }

    protected function create(array $data)
    {
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]);
    }
}
```



```
<?php

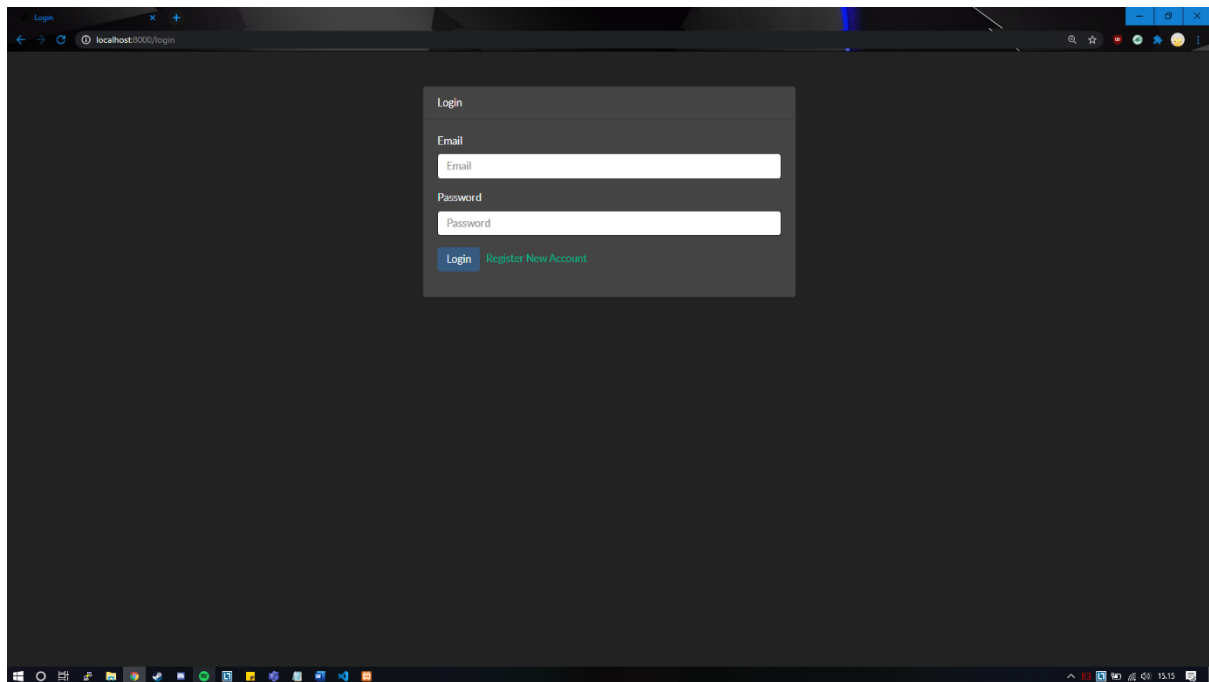
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;
use Auth;

class LoginController extends Controller
{
    use AuthenticatesUsers;
    protected $redirectTo = RouteServiceProvider::HOME;

    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }

    public function logout(Request $request) {
        Auth::logout();
        return redirect('/login');
    }
}
```



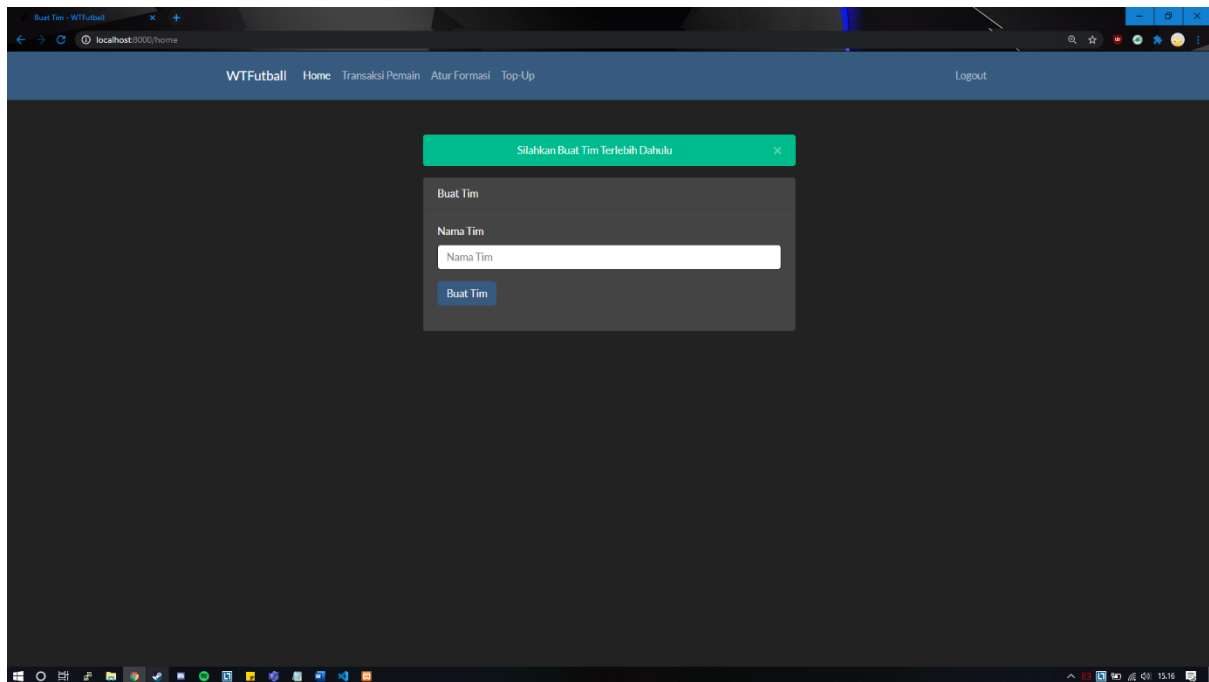
```
class TimController extends Controller
{
    public function index()
    {
        return view('buatTim');
    }

    public function store(Request $request)
    {
        Tim::insert([
            'nama' => $request->namaTim,
        ]);

        $timId = Tim::orderBy('id', 'desc')->select('id')->get();

        foreach($timId as $tim);
        DB::table('users_tim')->insert([
            'userId' => Auth::id(),
            'timId' => $tim->id,
        ]);

        return redirect('home');
    }
}
```



Link Github : <https://github.com/Aerysh/rpl-desain-dan-implementasi>

Selamat bekerja