Nama : Andy Burhanuddin
NIM : L0122021

LAPORAN RESPONSI PEMWEB

1. **PegawaiModel**

app > Models > Pegawai.php > ...

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

16 references | 0 implementations
class Pegawai extends Model
{
    use HasFactory;

    0 references
    protected $table = 'pegawais';

    0 references
    protected $fillable = ['nama', 'age', 'email'];
}
```

code diatas merupakan pegawai model yang memiliki atribut nama, age, email.

2. **UserModel**

```php
<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use HasFactory, Notifiable;

    protected $table = 'users';

    /**
     * The attributes that are mass assignable.
```

```
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
     *
     * @return array<string, string>
     */
    protected function casts(): array
    {
        return [
            'password' => 'hashed',
        ];
    }
}
```

Code di atas merupakan user model dengan credentials email dan password yang dapat digunakan untuk login ke akun dan membuka page mahasiswa.

3. **AuthController**

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use App\Models\User;
```

```php
class UserController extends Controller
{
    public function showRegisterForm()
    {
        return view('register');
    }

    // Handle user registration
    public function register(Request $request)
    {
        $request->validate([
            'username' => 'required|max:255',
            'password' => 'required',
        ]);

        $user = new User();
        $user->username = $request->username;
        $user->password = Hash::make($request->password);
        $user->save();

        return redirect()->route('formlogin')->with('success',
'Registration successful. Please login.');
    }

    public function showLoginForm()
    {
        return view('login');
    }

    public function login(Request $request)
    {
        $request->validate([
            'username' => 'required',
            'password' => 'required',
        ], [
            'username.required' => 'Username wajib diisi!',
            'password.required' => 'Password wajib diisi!',
        ]);

        $credentials = $request->only('username', 'password');

        if (Auth::attempt($credentials)) {
            return redirect()->intended(route('index'));
```

```
        }

        return back()->withErrors([
            'credentials' => 'Username atau password salah!'
        ]);
    }

    public function logout(Request $request)
    {
        Auth::logout();
        $request->session()->invalidate();
        $request->session()->regenerateToken();
        return redirect('/');
    }
}
```

Code di atas adalah AuthController yang digunakan untuk otentikasi user sebelum masuk ke dalam page Pegawai.

4. **PegawaiController**

```
<?php

namespace App\Http\Controllers;

use App\Models\Pegawai;
use Illuminate\Http\Request;

class PegawaiController extends Controller
{
    public function index()
    {
        $pegawai = Pegawai::all();
        return view('index', compact('pegawai'));
    }

    public function tambah()
    {
        return view('forminsert');
    }

    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'nama' => 'required|string|max:255',
```

```
            'age' => 'required|integer',
            'email' => 'required|string|email|max:255',
        ]);

        Pegawai::create([
            'nama' => $validatedData['nama'],
            'age' => $validatedData['age'],
            'email' => $validatedData['email'],
        ]);

        return redirect()->route('index')->with('success', 'Data
inserted successfully');
    }

    public function edit($id)
    {
        $pegawai = Pegawai::findOrFail($id);
        return view('formedit', compact('pegawai'));
    }

    public function update(Request $request, $id)
    {
        $validatedData = $request->validate([
            'nama' => 'required|string|max:255',
            'age' => 'required|integer',
            'email' => 'required|string|email|max:255',
        ]);

        $pegawai = Pegawai::findOrFail($id);
        $pegawai->update($validatedData);

        return redirect()->route('index')->with('success', 'Data
updated successfully');
    }

    public function hapus($id)
    {
        $pegawai = Pegawai::findOrFail($id);
        return view('formdelete', compact('pegawai'));
    }

    public function destroy(Request $request)
    {
```

```
        $validatedData = $request->validate([
            'id' => 'required|integer|exists:pegawais,id',
        ]);

        $pegawai = Pegawai::findOrFail($request->input('id'));
        $pegawai->delete();

        return redirect()->route('index')->with('success', 'Data
deleted successfully');
    }
}
```

Code di atas merupakan mahasiswacontroller yang digunakan untuk membuat data mahasiswa baru, mengedit, dan menghapus data Pegawai.

5. **Views index.blade.php**

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Week 11</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boots
trap.min.css" rel="stylesheet">
    <style>
        body {
            background-color: #f0f8ff; /* Light blue background
*/
            font-family: 'Arial', sans-serif;
        }
        .container {
            background-color: #ffffff; /* White background for
the container */
            padding: 2cm 1cm;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            border-radius: 10px;
            margin-top: 2cm;
        }
        h1, h2 {
            color: #333;
        }
        table {
```

```
            margin-top: 1cm;
        }
        .btn-primary {
            background-color: #28a745;
            border-color: #28a745;
        }
        .btn-primary:hover {
            background-color: #218838;
            border-color: #218838;
        }
        .btn-success {
            background-color: #007bff;
            border-color: #007bff;
        }
        .btn-success:hover {
            background-color: #0056b3;
            border-color: #0056b3;
        }
        .btn-danger {
            background-color: #dc3545;
            border-color: #dc3545;
        }
        .btn-danger:hover {
            background-color: #c82333;
            border-color: #c82333;
        }
    </style>
</head>
<body>
    <div class="container text-center">
        <div class="d-flex justify-content-between mb-3">
            <a href="{{ route('tambah')}}" class="btn
btn-primary">Tambah Data</a>
            <form action="{{ route('logout') }}" method="POST">
                @csrf
                <button class="btn btn-danger">Logout</button>
            </form>
        </div>
        <table class="table table-striped">
            <thead class="table-dark">
                <tr>
                    <th>ID</th>
                    <th>Nama</th>
```

```
                    <th>Usia</th>
                    <th>Email</th>
                    <th>Aksi</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($pegawai as $p)
                    <tr>
                        <td>{{ $p->id }}</td>
                        <td>{{ $p->nama }}</td>
                        <td>{{ $p->age }}</td>
                        <td>{{ $p->email }}</td>
                        <td>
                            <a href="{{ route('edit', ['id' =>
$p->id]) }}" class="btn btn-success btn-sm">Edit</a>
                            <a href="{{ route('hapus', ['id' =>
$p->id]) }}" class="btn btn-danger btn-sm">Hapus</a>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstr
ap.bundle.min.js"></script>
</body>
</html>
```

Code di atas merupakan code untuk page utama (index.blade.php) yang digunakan untuk menambah data pegawai, dan menghapusnya, serta tombol untuk masuk ke dalam page edit untuk mengedit data pegawai.

6. **View Login Page**

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Login</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/boots
trap.min.css" rel="stylesheet">
```

```
</head>
<body>
<div class="container mt-5">
    <h1 class="text-center">Login</h1>

    @if ($errors->any())
        <div class="alert alert-danger">
            <ul class="mb-0">
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    @if (session('success'))
        <div class="alert alert-success">
            {{ session('success') }}
        </div>
    @endif

    <form action="{{ route('login') }}" method="post"
class="border p-4 rounded">
        @csrf
        <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" name="username" id="username"
class="form-control" value="{{ old('username') }}" required>
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
            <input type="password" name="password" id="password"
class="form-control" required>
        </div>
        <button type="submit" class="btn btn-primary
btn-block">Login</button>
    </form>

    <p class="text-center mt-3">Belum punya akun? <a href="{{
route('formregister') }}">Register disini</a></p>
</div>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Code di atas merupakan views bagian login yang digunakan untuk login user.

7. **Register Blade Php**

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Register</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<div class="container mt-5">
    <h1 class="text-center">Register</h1>

    @if ($errors->any())
        <div class="alert alert-danger">
            <ul class="mb-0">
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('register') }}" method="post"
class="border p-4 rounded">
        @csrf
        <div class="form-group">
            <label for="username">Username:</label>
            <input type="text" name="username" id="username"
class="form-control" value="{{ old('username') }}" required>
        </div>
        <div class="form-group">
            <label for="password">Password:</label>
```

```
            <input type="password" name="password" id="password"
class="form-control" required>
        </div>
        <button type="submit" class="btn btn-primary
btn-block">Register</button>
    </form>

    <p class="text-center mt-3">Sudah punya akun? <a href="{{
route('formlogin') }}">Login disini</a></p>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstr
ap.bundle.min.js"></script>
</body>
</html>
```

Code di atas adalah page register yang dapat digunakan user untuk membuat user baru supaya dapat masuk ke page pegawai.

8. **Routes**

```
<?php

use App\Http\Controllers\PegawaiController;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;

Route::get('/', function () {
    return view('welcome');
});

Route::get('register', [UserController::class,
'showRegisterForm'])->name('formregister');
Route::post('register', [UserController::class,
'register'])->name('register');

Route::get('login', [UserController::class,
'showLoginForm'])->name('formlogin');
Route::post('login', [UserController::class,
'login'])->name('login');

Route::post('logout', [UserController::class,
'logout'])->name('logout');
```

```php
Route::middleware(['middleware' => 'auth'])->group(function () {
    Route::get('/view', [PegawaiController::class,
'index'])->name('index');

    Route::get('/add', [PegawaiController::class,
'tambah'])->name('tambah');
    Route::post('/add/store', [PegawaiController::class,
'store'])->name('store');

    Route::get('/edit{id}', [PegawaiController::class,
'edit'])->name('edit');
    Route::post('/edit/update/{id}', [PegawaiController::class,
'update'])->name('update');

    Route::get('/hapus/{id}', [PegawaiController::class,
'hapus'])->name('hapus');
    Route::post('/hapus/destroy', [PegawaiController::class,
'destroy'])->name('destroy');
});
```

Routes yang digunakan untuk mengatur ke mana page akan dialihkan dan controller apa yang akan digunakan berdasarkan aktivitas pengguna.

9. **Migration**

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('pegawais', function (Blueprint $table) {
            $table->id();
            $table->string('nama');
            $table->integer('age');
            $table->string('email')->unique();
            $table->timestamps();
```

```
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('pegawais');
    }
};
```

Code di atas adalah bagian migration yang berfungsi untuk membuat schema database dengan isi atribut nama (string), age(integer), email(string), dan timestamp.

10. **Seeder**

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\User;

class UserSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        $userData = [
            'username' => 'andy',
            'password' => bcrypt('andy'),
        ];

        User::create($userData);
    }
}
```

Code di atas adalah UserSeeder yang nantinya dapat digunakan untuk login. User yang di atas adalah admin dengan usernameandy, serta password andy.

11. **Factory**

```
<?php

namespace Database\Factories;
```

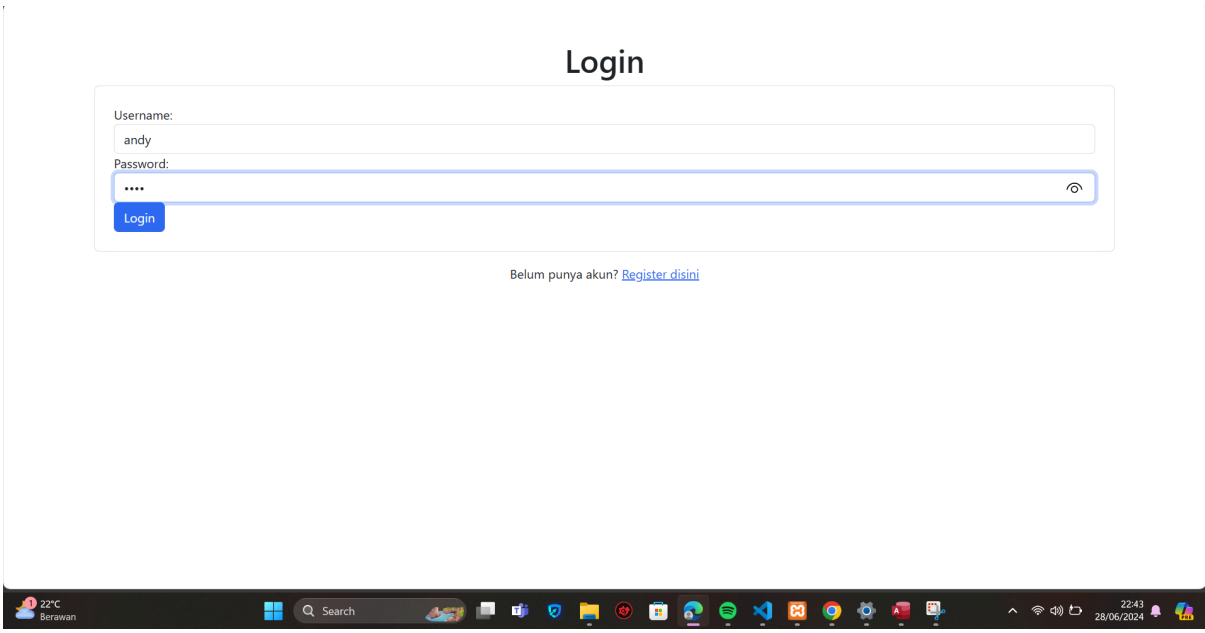
```php
use Illuminate\Database\Eloquent\Factories\Factory;

/**
 * @extends
\Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Pegaw
ai>
 */
class PegawaiFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'nama' => fake()->name(),
            'age' => fake()->numberBetween(18, 60),
            'email' => fake()->safeEmail(),
        ];
    }
}
```

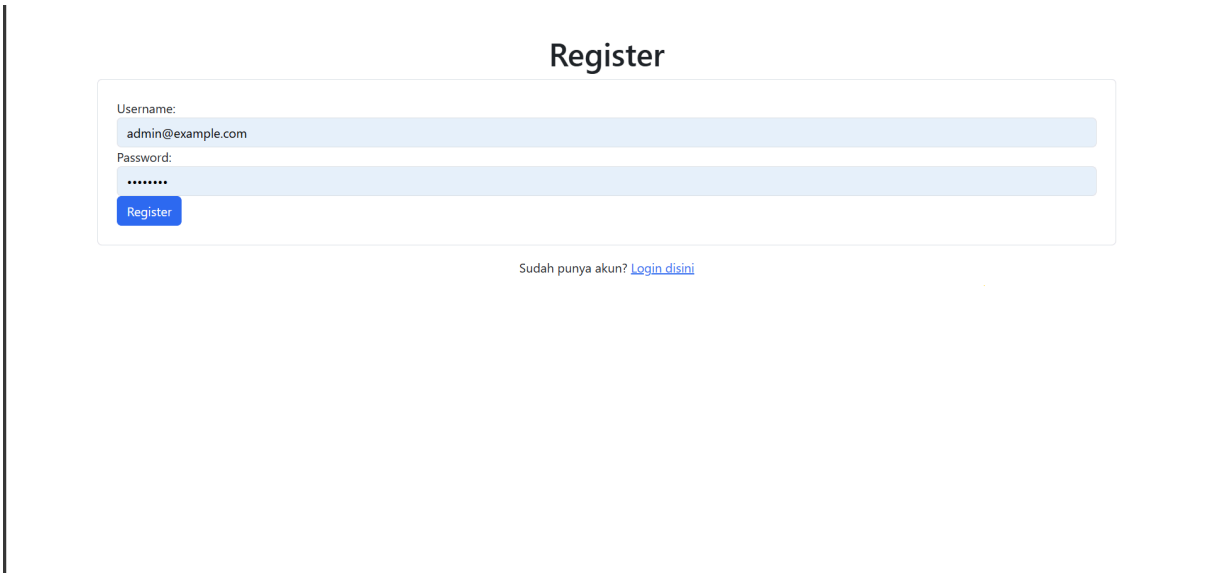
Code di atas adalah PegawaiFactory yang berfungsi untuk membuat data dummy untuk user dengan random. Nantinya factory ini dapat digunakan untuk membuat contoh user dengan usernam beserta passwordnya.

OUTPUT

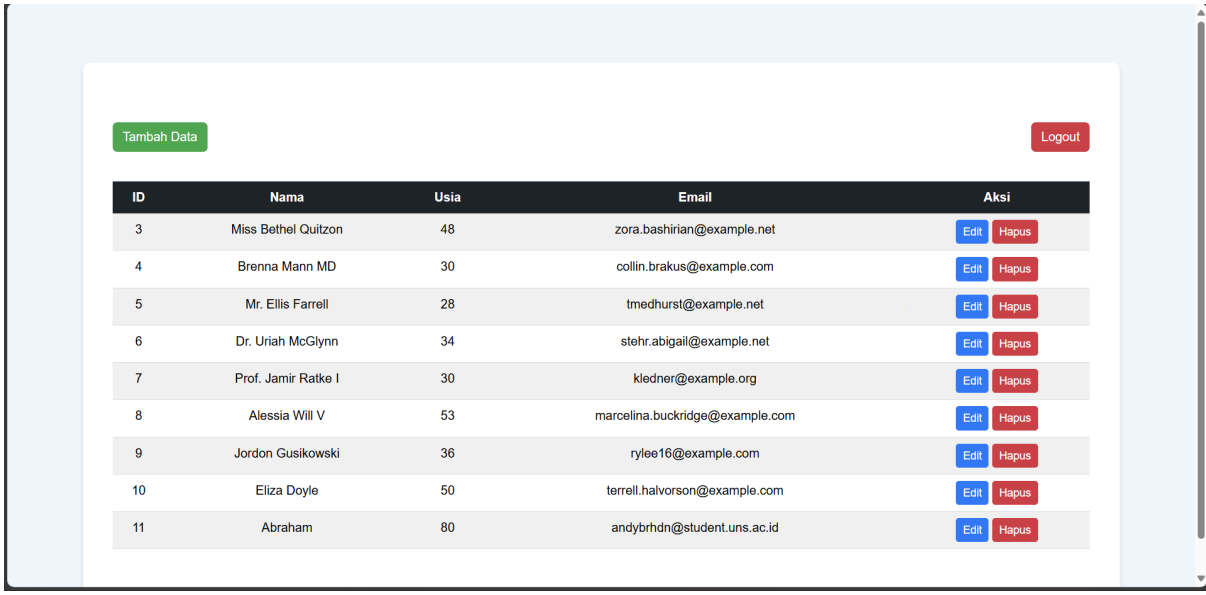
1. **Login**



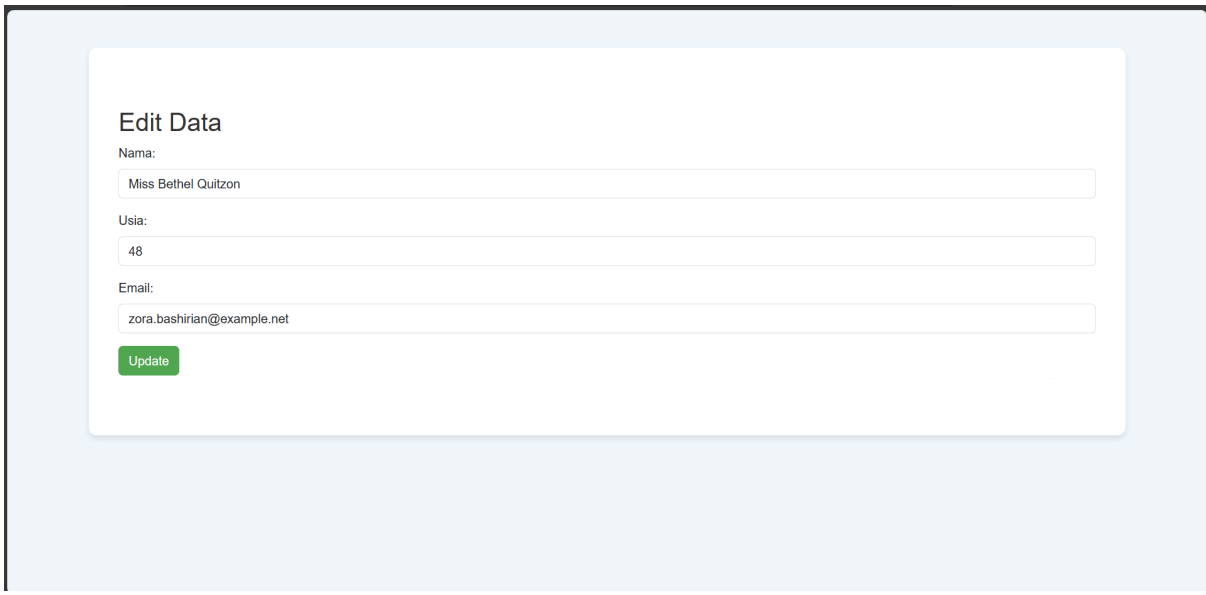
2. Register



3. Index


Tambah Data
Logout
ID
Nama
Usia
Email
Aksi
3
Miss Bethel Quitzon
48
zora.bashirian@example.net
Edit
Hapus
4
Brenna Mann MD
30
collin.brakus@example.com
Edit
Hapus
5
Mr. Ellis Farrell
28
tmedhurst@example.net
Edit
Hapus
6
Dr. Uriah McGlynn
34
stehr.abigail@example.net
Edit
Hapus
7
Prof. Jamir Ratke I
30
kledner@example.org
Edit
Hapus
8
Alessia Will V
53
marcelina.buckridge@example.com
Edit
Hapus
9
Jordon Gusikowski
36
rylee16@example.com
Edit
Hapus
10
Eliza Doyle
50
terrell.halvorson@example.com
Edit
Hapus
11
Abraham
80
andybrhdn@student.uns.ac.id
Edit
Hapus

4. Edit


Edit Data
Nama:
Miss Bethel Quitzon
Usia:
48
Email:
zora.bashirian@example.net
Update