|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran Lumen Video ke 11 -20  (Tugas semester 4 #7) | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Update data menggunakan perintah update() pada function update.

Cth :

public function update(Request $request, $id)

    {

        Kategori::where('idkategori', $id)->update($request->all());

*return* response()->json("Data sudah diupdate");

    }

Artinya, pada function update terdapat parameter yang menampung semua nilai data yang akan diupdate ke dalam $request dan terdapat $id. Di dalam function update berisi memanggil Model Kategori dan menjalankan function where yang memilih kolom idkategori yang nilainya sama dengan $id yang dimasukkan, setelah itu menjalankan perintah update dengan menjalankan $request mengupdate semua yang disimpan ke dalam $request. Apabila selesai mengupdate, maka nilai akan dikeluarkan dan menampilkan response() dengan mengirimkan data berbentuk json dengan tulisan Data sudah diupdate.

1. Form-data digunakan untuk mengirimkan data non-ASCII atau biner besar data.
2. X-www-form-urlencoded digunakan untuk mengirimkan data sederhana / data ASCII.
3. Melakukan validasi untuk mengetahui data – data yang sudah dimasukkan apakah sesuai dengan peraturan yang dibuat atau tidak. Jika tidak sesuai, maka akan menampilkan pesan bahwa data yang dimasukkan salah, tetapi jika sesuai, maka akan menjalankan perintah berikutnya.
4. Melakukan validasi dengan menggunakan perintah validate().

Cth :

$this->validate($request, [

            'kategori' => 'required | unique:kategoris',

            'keterangan' => 'required'

        ]);

        $kategoris = Kategori::create($request->all());

*return* response()->json($kategoris);

Artinya, terdapat perintah validate yang membaca semua data pada $request, sebelum data itu dikirimkan, data tersebut harus sesuai dengan peraturan yang dibuat. Ada 2 peraturan pada tabel kategoris, yaitu pada kolom kategori harus diisi dan seitap karakter pada tabel kategori harus berbeda, tidak boleh ada yang sama, jika ada yang sama maka harus mengganti, dan untuk mengisi unique harus ada patokan untuk mengecek data, patokan dari kolom kategori adalah pada tabel kategoris, sementara pada kolom keterangan, kolom harus diisi. Lalu $kategori berisi Model Kategori yang menjalankan perintah create dengan mengirimkan semua data yang ada pada $request. Setelah itu nilai akan dikeluarkan dan menampilkan response() dengan mengirimkan data berbentuk json yang menjalankan $kategoris.

1. getClientOriginalName() adalah fungsi yang digunakan untuk mengambil nama dari file yang dimasukkan.
2. Memasukkan file yang sudah dipilih ke dalam folder menggunakan perintah move().

Cth :

$gambar = $request->file('gambar')->getClientOriginalName();

$request->file('gambar')->move('upload', $gambar);

*return* response()->json($gambar);

Artinya, $gambar berisi $request yang menjalankan perintah file() dengan key gambar yang digunakan untuk mengambil file berupa gambar, setelah file dipilih, maka getClientOriginalName akan mengambil nama file yang sudah dimasukkan. Setelah itu, $request menjalankan perintah file() dengan key yang sama yaitu gambar, dan menjalankan perintah move(), untuk menjalankan perintah move, maka harus mengisi nama folder untuk menyimpan gambar dan nama variabel yang digunakan untuk mengisi file gambar yang akan dikirim, untuk memindahkan file yang dimasukkan ke dalam folder upload yang berada di folder public dan $gambar adalah variabel yang digunakan untuk menampung file yang akan dikirim. Lalu akan menampilkan response() dengan data yang sudah dikirimkan berbentuk json yang menjalankan $gambar.

Jadi, $gambar berisi $request yang menampung file yang akan dikirimkan. Lalu file yang dikirimkan akan dimasukkan ke dalam folder upload menggunakan perintah move(). Setelah itu, akan menampilkan $gambar.

1. Input() adalah perintah yang digunakan untuk mengirimkan data.

Cth :

$data = [

            'idkategori' => $request->input('idkategori'),

            'menu' => $request->input('menu'),

            'gambar' => $gambar,

            'harga' => $request->input('harga')

        ];

Artinya, di dalam $data terdapat key idkategori berisi $request yang menjalankan perintah input ke dalam idkategori, jadi $request yang menampung nilai yang akan dimasukkan ke dalam idkategori menjalankan perintah input() lalu data akan dikirimkan ke dalam kolom idkategori. Key menu berisi $request yang menjalankan perintah input ke dalam kolom menu, jadi $request yang menampung nilai yang akan dimasukkan ke dalam menu menjalankan perintah input(), lalu data akan dimasukkan ke dalam kolom menu. Key gambar berisi $gambar, dimana $gambar berisi $request yang menampung file yang akan dikirimkan dan memindahkannya ke dalam folder. Key harga berisi $request yang menjalankan perintah input ke dalam harga, jadi $request yang menampung nilai yang akan dimasukkan ke dalam harga menjalankan perintah input(), lalu data akan dikirimkan ke dalam kolom harga.

1. Membuat url pada gambar yang dimasukkan dengan menggunakan url().

Cth :

 'gambar' => url('upload/' . $gambar),

Artinya, pada key gambar berisi perintah url(), yang digunakan untuk membuat url. Jadi ketika gambar dimasukkan, maka url yang dibuat adalah upload/nama gambar. Di dalam kolom gambar, maka akan berbentuk url yang akan mengarahkan ke dalam tempat gambar disimpan.

1. Menambahkan ekstensi json view untuk merapikan tampilan data pada web.
2. Kode status respon [HTTP](https://developer.mozilla.org/en-US/docs/Web/HTTP) menunjukkan apakah permintaan HTTP tertentu telah berhasil diselesaikan. Tanggapan dikelompokkan dalam lima kelas:

* Tanggapan informasi.
* Tanggapan yang sukses.
* Pengalihan.
* Kesalahan klien.
* Server kesalahan.

1. Membuat status response ketika data berhasil dikirimkan.

Cth :

*if* ($menus) {

            $result = [

                'status' => 200,

                'pesan' => 'Data berhasil dikirimkan',

                'data' => $data

            ];

        } *else* {

            $result = [

                'status' => 400,

                'pesan' => 'Data gagal dikirimkan',

                'data' => ''

            ];

        }

Artinya, apabila $menus berhasil dikirimkan, maka akan menjalankan $result yang berisi key status dengan value 200 (Status Response OK), key pesan dengan value Data berhasil dikirimkan, key data dengan value $data. Dan jika data gagal dikirimkan, maka akan menjalankan $result yang berisi key status 400 (Status Response Bad Response), key pesan dengan value Data gagal dikirimkan, dan key data dengan value kosong.

1. Di dalam laravel, middleware adalah class khusus yang berperan sebagai penengah antara request yang masuk dengan controller yang dituju. Dan digunakan untuk memfilter siapa saja yang boleh masuk ke dalam controller yang diinginkan, tetapi jika gagal maka akan diarahkan ke dalam controller lain.
2. Cara kerja dari web, yang pertama adalah request, jadi ketika kita mengetik sesuatu di dalam url, maka itu adalah kita sedang menjalankan request. Di dalam lumen, request akan ditangani pertama kali oleh route. Lalu, middleware berada di antara route dan controller. Jadi setelah url masuk ke route, selanjutnya akan masuk ke dalam middleware, apakah permintaan akan dilanjutkan atau tidak. Kemudian masuk ke dalam controller.
3. Mengisi file UserMiddleware dengan if.

Cth :

*if* ($request->user <> 'admin') {

*return* redirect('login');

        }

*return* $next($request);

Artinya, apabila $request dengan variabel user tidak sama dengan admin, maka akan dikembalikan ke dalam login. Tetapi, jika $request variabel user sama dengan admin, maka $request akan dilanjutkan.

1. Mendaftarkan middleware ke dalam file app.php pada folder bootstrap. Setelah itu, membuat route untuk login yang diarahkan ke file LoginController.php apabila variabel user bukan admin. Lalu, menambahkan ‘middleware’ => ‘user’ ke dalam group route, untuk menyeleksi, jika variabel user bukan admin, maka akan diarahkan ke dalam file login, tetapi, jika variabel user = admin, maka bisa untuk membuka file – file yang berada di dalam grou route.

Cth pemanggilan :

<http://localhost:8000/api/menu> => akan diarahkan ke dalam file login, karena user tidak sama dengan admin.

<http://localhost:8000/api/pelanggan?user=user> => akan diarahkan ke dalam file login, karena user tidak sama dengan admin.

<http://localhost:8000/api/pelanggan?user=admin> => akan diarahkan ke dalam file pelanggan dan menampilkan semua data dari file pelanggan, karena user sama dengan admin.

1. Token adalah salah satu penggunaan dari middleware yang melakukan autentifikasi bahwa yang melakukan login / yang bisa mengakses file tersebut hanya yang mempunyai token saja dan token adalah perangkat security dari aplikasi yang dibuat untuk keamanan aplikasi.

Cth :

$token = "1234567890";

            $header = $request->header('Api-Token');

*if* ($header && $header == $token) {

*return* new User();

            }

*return* null;

Artinya, $token berisi kode token yang harus dimasukkan ketika ingin membuka file yang sudah diatur tokennya, yaitu 1234567890. $header berisi $request yang akan dijalankan ketika header dimasukkan kode Api-Token. Lalu terdapat pengujian, yaitu apabila nilai dari $header && $header sama dengan nilai dari $token, maka akan menjalankan user baru. Tetapi jika nilai nya tidak sama maka akan menampilkan nilai null / unauthorized.

1. First() digunakan untuk mengambil data yang paling atas setelah melakukan seleksi dan data tersebut sesuai dengan apa yang diseleksi. Jadi ketika menyeleksi data, tetapi ada 2 atau 3 data yang sama, maka yang diambil adalah data pertama yang cocok.
2. Str::random() adalah fungsi yang digunakan untuk membuat string acak otomatis.
3. Melakukan login user :

public function login(Request $request)

    {

        $email = $request->input('email');

        $password = $request->input('password');

        $user = User::where('email', $email)->first();

*if* ($user->password === $password) {

            $token = Str::random(40);

            $user->update([

                'api\_token' => $token

            ]);

*return* response()->json([

                'pesan' => 'Login anda sukses',

                'token' => $token,

                'data' => $user

            ]);

        } *else* {

*return* response()->json([

                'pesan' => 'Login anda gagal',

                'data' => ''

            ]);

        }

    }

Artinya, di dalam function login terdapat Request yang akan menerima semua data yang akan dikirimkan dan disimpan ke dalam $request. Function login terdapat $email berisi $request yang menjalankan perintah input ke dalam kolom email, jadi $request yang menampung nilai yang akan dimasukkan ke dalam kolom email dan menjalankan perintah input() lalu data akan dikirimkan ke dalam kolom email. Lalu $password berisi $request yang menjalankan perintah input ke dalam kolom password, jadi $request yang menampung nilai yang akan dimasukkan ke dalam kolom password dan menjalankan perintah input(), lalu data akan dikirimkan ke dalam kolom password.

$user berisi Model User yang memilih data sesuai dengan kolom email dan dari $email dengan data yang sama, dan pengambilan data menggunakan first(), yaitu mengambil data yang paling atas jika terdapat email yang sama.

Lalu terdapat pengujian menggunakan if, yaitu apabila $user yang membaca kolom password nilainya sama dengan $password, maka akan menjalankan $token yang berisi perintah untuk membuat string acak sebanyak 40 karakter. Setelah itu, $user akan menjalankan perintah update() dengan mengupdate kolom api\_token sesuai dengan $token. Jadi setiap melakukan login, maka nilai dari $token juga akan berubah karena setiap login, maka token akan diupdate. Lalu nilai akan dikeluarkan dan diberi response dengan data berbentuk json, yaitu key pesan berisi value Login anda sukses, key token berisi value sesuai dengan $token, dan key data berisi value $user. Tetapi jika login tidak berhasil, maka nilai akan dikeluarkan dan menampilkan response dengan data berbentuk json, yaitu key pesan Login anda gagal, dan key data kosong.

1. Autentikasi adalah proses untuk memastikan bahwa login adalah orang yang berhak, sesuai dengan email dan password.

**Saya Belum Mengerti**