

Rifqi Munawar Ridwan

41037006211013

Teknik Informatika A1

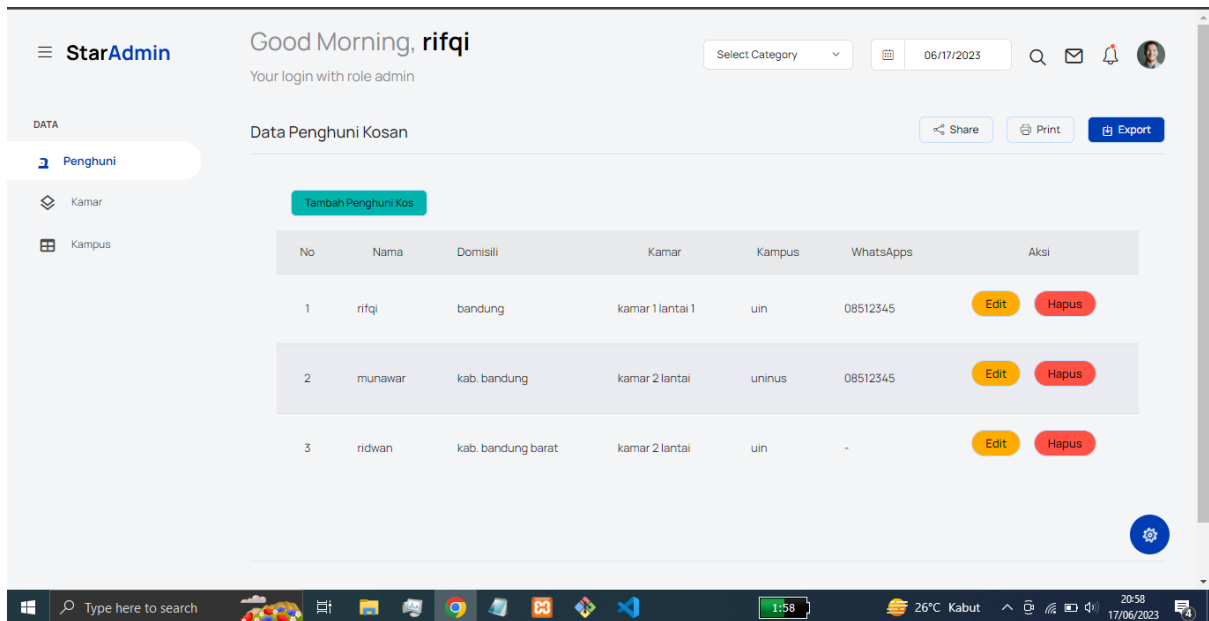
Tugas Mata Kuliah Pemrograman Berbasis Kerangka Kerja: Database Relationship

Github : <https://github.com/rifqimunawar/laravel-pbkk>

Disini saya menggunakan template dari star admin, studi kasusnya data penghuni kosan

Database yang berelasi yaitu

- penghuni berelasi dengan kamar (one to many),
- penghuni berelasi dengan no whatsapp (one to one)
- penghuni berelasi dengan kampus (many to many)



The screenshot displays the StarAdmin dashboard interface. On the left, a sidebar menu lists 'DATA' categories: 'Penghuni' (selected), 'Kamar', and 'Kampus'. The main content area is titled 'Good Morning, rifqi' and 'Your login with role admin'. Below this, the 'Data Penghuni Kosan' section features a 'Tambah Penghuni Kos' button and a table with the following data:

No	Nama	Domisili	Kamar	Kampus	WhatsApps	Aksi
1	rifqi	bandung	kamar 1 lantai 1	uin	08512345	<button>Edit</button> <button>Hapus</button>
2	munawar	kab. bandung	kamar 2 lantai	uninus	08512345	<button>Edit</button> <button>Hapus</button>
3	ridwan	kab. bandung barat	kamar 2 lantai	uin	-	<button>Edit</button> <button>Hapus</button>

The bottom of the image shows a Windows taskbar with the time 1:58, date 17/06/2023, and weather information (26°C Kabut).

Model Penghuni yang berelasi dengan model kamar, model phone, model kampus,

```
1 class Penghuni extends Model
2 {
3     use HasFactory;
4     protected $guarded=[];
5     protected $table='penghuni';
6
7     public function kamars()
8     {
9         return $this->belongsTo(Kamar::class);
10    }
11
12    public function phone()
13    {
14        return $this->hasOne(Phone::class);
15    }
16    public function univ()
17    {
18        return $this->belongsTo(Univ::class)->withDefault();
19    }
20 }
```

Model kamar yang berelasi dengan model penghuni

Model phone yang berelasi dengan model penghuni

```
1 class Kamar extends Model
2 {
3     use HasFactory;
4     protected $guarded=[];
5     public function penghuni()
6     {
7         return $this->hasMany(Penghuni::class);
8     }
9 }
```

```
1 class Phone extends Model
2 {
3     use HasFactory;
4     protected $guarded=[];
5
6     public function penghuni()
7     {
8         return $this->belongsTo(Penghuni::class);
9     }
10 }
```

Model kampus yang berelasi dengan model penghuni →

```
1 class Univ extends Model
2 {
3     use HasFactory;
4     protected $guarded=[];
5     public function Penghuni()
6     {
7         return $this->belongsTo(Penghuni::class);
8     }
9 }
```

Controller Penghuni yang memanggil data penghuni yang sudah berelasi dengan kamar, phone, dan kampus,

```
1 class PenghuniController extends Controller
2 {
3     public function index()
4     {
5         $user=Auth::user();
6         $penghunies = Penghuni::with('kamars', 'phone', 'univ')->latest()->get();
7         return view('admin.index', compact('penghunies', 'user'));
8     }
9 }
```

Output dari function index tersebut sebagai berikut:

StarAdmin Good Morning, rifqi
Your login with role admin

Select Category 06/17/2023

DATA

Penghuni

Kamar

Kampus

Data Penghuni Kosan

Tambah Penghuni Kos

No	Nama	Domisili	Kamar	Kampus	WhatsApps	Aksi
1	rifqi	bandung	kamar 1 lantai 1	uin	08512345	Edit Hapus
2	munawar	kab. bandung	kamar 2 lantai	uninus	08512345	Edit Hapus
3	ridwan	kab. bandung barat	kamar 2 lantai	uin	-	Edit Hapus

20:58 17/06/2023

Data kamar dan kampus diambil dari pilihan data kamar dan kampus pada table berikut:

The screenshot shows the StarAdmin dashboard with the user 'rifqi' logged in as an admin. The left sidebar has a 'DATA' section with 'Kamar' selected. The main content area displays a table titled 'Data Kamar Kosan' with columns 'No', 'Kamar', and 'Aksi'. The table contains four rows of data. Above the table is a 'Tambah Kamar' button. To the right of the table are 'Share', 'Print', and 'Export' buttons.

No	Kamar	Aksi
1	kamar 1 lantai 1	<button>Edit</button> <button>Hapus</button>
2	kamar 2 lantai 1	<button>Edit</button> <button>Hapus</button>
3	kamar 1 lantai 2	<button>Edit</button> <button>Hapus</button>
4	kamar 2 lantai	<button>Edit</button> <button>Hapus</button>

The screenshot shows the StarAdmin dashboard with the user 'rifqi' logged in as an admin. The left sidebar has a 'DATA' section with 'Kampus' selected. The main content area displays a table titled 'Data Kampus Penghuni' with columns 'No', 'Kampus', and 'Aksi'. The table contains two rows of data. Above the table is a 'Tambah Kampus' button. To the right of the table are 'Share', 'Print', and 'Export' buttons.

No	Kampus	Aksi
1	uninus	<button>Edit</button> <button>Hapus</button>
2	uin	<button>Edit</button> <button>Hapus</button>

Create Kamar dan Create Kampus :

The form is titled 'Data Kamar Kosan' and includes 'Share', 'Print', and 'Export' buttons. Below the title is a text input field labeled 'Nama Kamar Baru'. To the right of the input field are two buttons: 'Kembali' (yellow) and 'Simpan' (blue).

Nama Kampus Baru

[Kembali](#) [Simpan](#)

Dari data kamar dan kampus tersebut akan dipanggil di create penghuni kamar:

StarAdmin Good Morning, **rifqi**
Your login with role admin

Select Category

06/17/2023

DATA

- Penghuni
- Kamar
- Kampus

Data Penghuni Kosan

[Share](#) [Print](#) [Export](#)

Nama Penghuni

Domisili Penghuni

Kampus Penghuni

Pilih Kampus

Kamar Penghuni

Pilih Kamar

WhatsApps

[Kembali](#) [Simpan](#) [Settings](#)

Controller dari ke 3 model tersebut:

```

1 class PenghunController extends Controller
2 {
3     public function index()
4     {
5         $user=Auth::user();
6         $penghunies = Penghuni::with('kamars', 'phone', 'univ')->latest()->get();
7         return view('admin.index', compact('penghunies', 'user'));
8     }
9     public function create(Request $request)
10    {
11        $universitas = Univ ::all();
12        $kamars = Kamar ::all();
13        $user = Auth::user();
14        return view('admin.penghuni.create', compact(
15            'universitas', 'user', 'kamars'
16        ));
17    }
18    public function store(Request $request)
19    {
20        // Mengambil data dari permintaan
21        $penghuniData = [
22            'name' => $request->input('name'),
23            'domisili' => $request->input('domisili'),
24            'kamars_id' => $request->input('kamars_id'),
25            'univ_id' => $request->input('univ_id'),
26        ];
27
28        // Buat objek Penghuni dan simpan data
29        $penghuni = Penghuni::create($penghuniData);
30
31        // Jika nomor telepon disertakan dalam permintaan,
32        // buat objek Phone dan hubungkan dengan Penghuni
33        if ($request->has('phone')) {
34            $phoneData = ['phone' => $request->input('phone')];
35            $phone = $penghuni->phone()->create($phoneData);
36        }
37        return redirect('/dashboard');
38    }
39
40    public function edit($id, Request $request)
41    {
42        $user = Auth::user();
43        $penghunies = Penghuni ::find($id);
44        $kamars=Kamar ::all();
45        $universitas = Univ ::all();
46        // dd($penghunies);
47        return view('admin.penghuni.edit', compact('penghunies',
48            'kamars', 'user', 'universitas'));
49    }
50    public function update($id, Request $request)
51    {
52        $penghunies = Penghuni::findOrFail($id);
53
54        $penghuniesData = $request->all();
55        $penghunies->update($penghuniesData);
56
57        return redirect('/dashboard');
58    }
59    public function destroy($id)
60    {
61        $penghuni = Penghuni::findOrFail($id);
62        $penghuni->delete();
63        return redirect('/dashboard');
64    }
65 }
66

```

```
1 class KamarController extends Controller
2 {
3   public function index(Request $request)
4   {
5     $kamars = Kamar::latest()->get();
6     $user = Auth::user();
7     return view('admin.kamar.index', compact('kamars', 'user'));
8   }
9
10  public function create()
11  {
12    $user = Auth::user();
13    return view('admin.kamar.create', compact('user'));
14  }
15
16  public function store(Request $request)
17  {
18    $kamar = $request -> all();
19    $kamar = kamar::create($kamar);
20    return redirect('/kamar');
21  }
22  public function edit($id, Request $request)
23  {
24    $user=Auth::user();
25    $kamars = Kamar ::find($id);
26    return view('admin.kamar.edit', compact('kamars', 'user'));
27  }
28  public function update($id, Request $request)
29  {
30    $kamar = Kamar::findOrFail($id);
31
32    $kamarData = $request->all();
33    $kamar->update($kamarData);
34
35    return redirect('/kamar');
36  }
37  public function destroy($id)
38  {
39    $kamar = Kamar::findOrFail($id);
40    $kamar->delete();
41    return redirect('/kamar');
42  }
43 }
```

```
1 class UnivController extends Controller
2 {
3     public function index(Request $request)
4     {
5         $universitas = Univ::latest()->get();
6         $user = Auth::user();
7         return view('admin.univ.index', compact('universitas', 'user'));
8     }
9
10    public function create()
11    {
12        $user = Auth::user();
13        return view('admin.univ.create', compact('user'));
14    }
15
16    public function store(Request $request)
17    {
18        $universitas = $request -> all();
19        $universitas = Univ::create($universitas);
20        return redirect('/univ');
21    }
22    public function edit($id, Request $request)
23    {
24        $user=Auth::user();
25        $universitas = Univ ::find($id);
26        return view('admin.univ.edit', compact('universitas', 'user'));
27    }
28    public function update($id, Request $request)
29    {
30        $universitas = Univ::findOrFail($id);
31
32        $universitasData = $request->all();
33        $universitas->update($universitasData);
34
35        return redirect('/univ');
36    }
37    public function destroy($id)
38    {
39        $universitas = Univ::findOrFail($id);
40        $universitas->delete();
41        return redirect('/univ');
42    }
43 }
```


Strukture View:

