

Information system of Rumah Makan Sinar Minang Balige

Created by Kelompok 02



Sinar Minang Restaurant

[Patunjuk](#) [Save](#)

4.1 ★★★★☆ 83 ulasan Google

Restoran

Address: Jl. Siborong Borong - Parapat No 2, Sibola Hotangsa, Balige, Kabupaten Toba Samosir, Sumatera Utara

Jam: **Closed** · Opens 09:00 ▾

Telepon: 0812-9464-6807

[Suggest an edit](#)

[Add missing information](#)

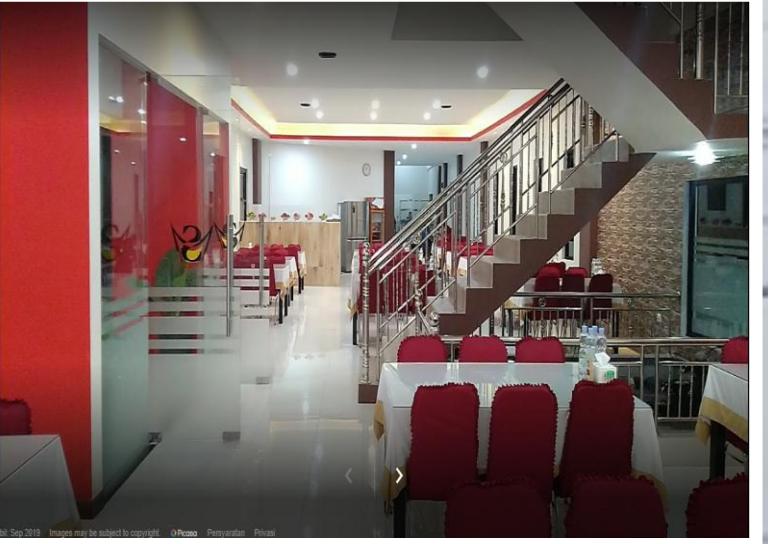
[Add website](#)

Know this place? [Answer quick questions](#)

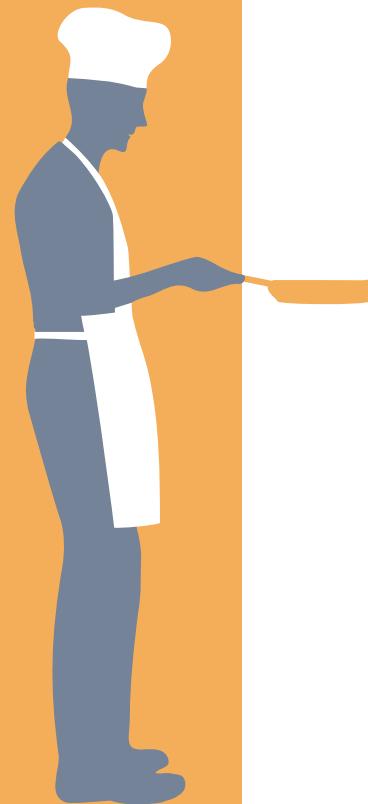
Questions & answers

[See all questions \(1\)](#)

[Ask a question](#)



Current System



1 Customer used to spend long time to order food

2 Food took long to be prepare

3 Staff difficult to handle a lot of customer

4 Food is wasted



Cooking...

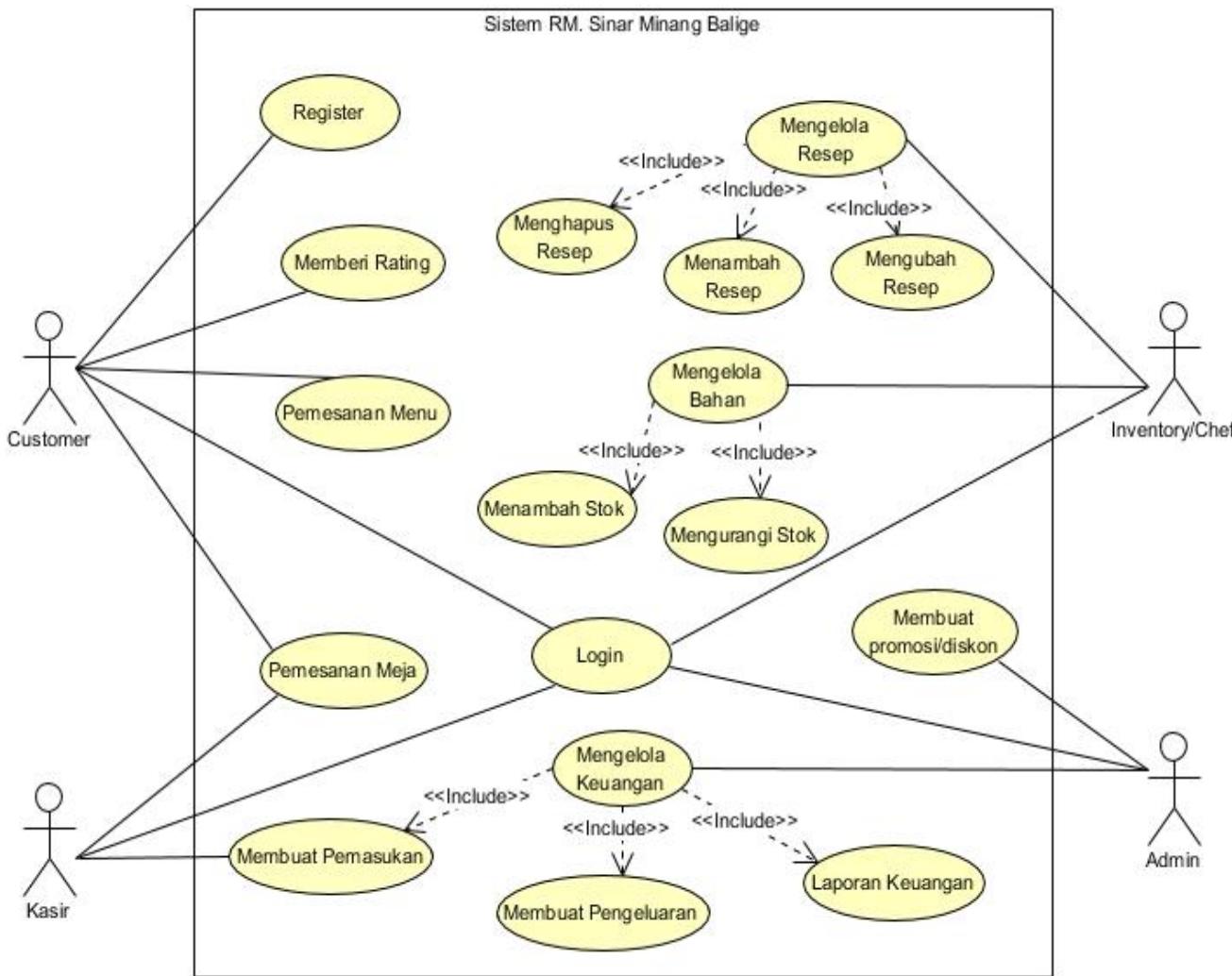


Benefit of **N**e w **S**ystem

- 1** | Business process much easier and faster
- 2** | Enable and courage online order
- 3** | Make it easy to make reservation
- 4** | Grow up the business
- 5** | Recognize inventory system

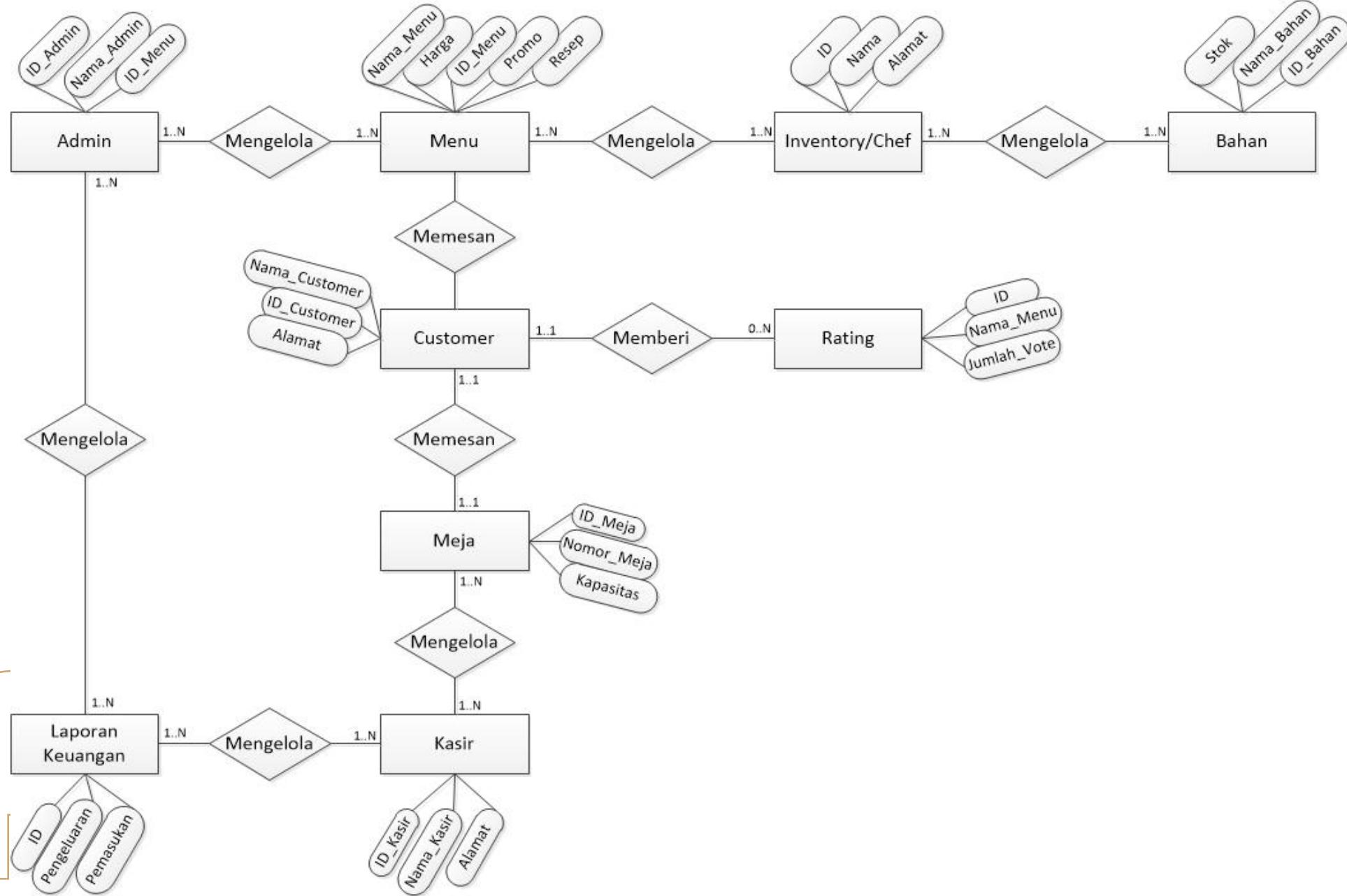


Use Case Diagram Sistem Informasi Rumah Makan Sinar Minang





Entity Relationship Diagram





The Succedd and Fail
Function

Succedd Function

General:

- Login & Logout
- Menu

Customer:

- Register
- Gallery
- Add to cart
- Table Reservation

Inventory/Chef:

- Recipe Create, View, Delete
- Show Ingredients
- Edit Inventory Stock

Cashier:

- Show Booked Table
- Show Income

Admin

- Promo Edit, View
- Register employees



Fail Function

Customer:

Cancel reservation
Total Cart

Inventory/Chef:

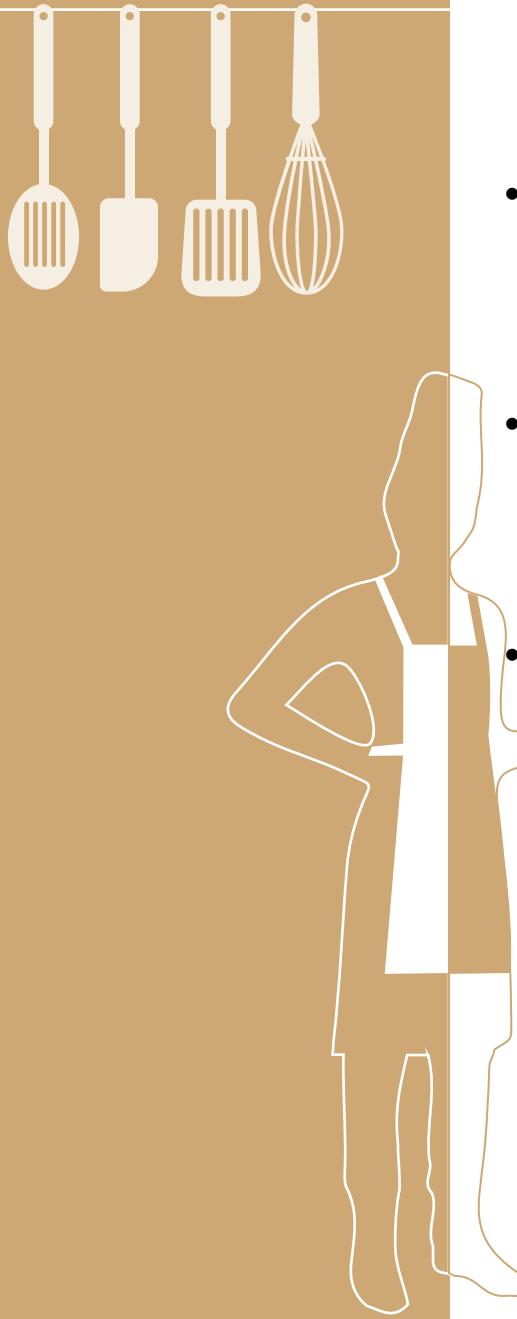
Upload Recipe Picture

Cashier:

Expenses

Admin:

Report



Application Architecture

- In Information system, application architecture is one of several architecture domains that form the pillars of an enterprise architecture.
- Application architecture describes the behavior of application used in a business focused on how they interact with each other and with users.
- In laravel, there are several components that support application architecture.
 1. Routes
 2. Controllers
 3. Views
 4. Models

Website Rumah Makan Minang Balige

This website designed primarily for use in the restaurant industry. This system allows users to quickly and easily manage an online menu which customers can browse and use to place orders with just a few clicks. Restaurant employees then use these orders through an easy-to-navigate graphical interface for efficient processing.





Routes

- Route is a function which manage the mapping between URL and controller.
- In Laravel, route written in a file called web.php
- The directory location is in:

Route > Web.php

The screenshot shows a code editor interface with two main panes. The left pane is titled 'EXPLORER' and shows a tree view of project files under 'OPEN EDITORS'. The 'web.php' file is listed under the 'routes' folder and is currently selected, indicated by a blue background. Other files visible include 'api.php', 'channels.php', 'console.php', 'storage', 'tests', and 'vendor'. The right pane displays the contents of the 'web.php' file, which contains PHP code for defining routes. The code includes various route definitions using the Route::get() and Route::post() methods, along with some Route::resource() calls for managing resources like 'menu', 'promo', 'reservation', and 'billing'.

```
<?php  
Auth::routes();  
Route::get('/', 'ContentsController@index');  
Route::get('/home', 'Auth\UsersController@index');  
Route::resource('menu', 'MenuController');  
Route::get('/login', 'Auth\UsersController@login_index');  
Route::get('/register', 'Auth\UsersController@register_index');  
Route::get('/register-pegawai', 'Auth\UsersController@regisWorker');  
Route::post('/registerPost', 'Auth\UsersController@register');  
Route::post('/registerPost-pegawai', 'Auth\UsersController@postWorker');  
Route::post('/loginPost', 'Auth\UsersController@login');  
Route::get('/logout', 'Auth\UsersController@logout');  
Route::resource('promo', 'Admin\PromoController');  
Route::resource('reservation', 'Kasir\ReservationController');  
Route::resource(['billing'], 'Kasir\BillingController');
```



Controllers

- Controller contains methods for commands that must be performed on the method
- After the route connects controller to the method, then method will return the value for destination URL to go to
- In laravel, the controller directory is in:
app > Http > Controllers > FileName.php

The screenshot shows a code editor interface with two main panes. The left pane is the 'EXPLORER' view, which displays a tree structure of project files. The right pane is the 'MenuController.php X' file, showing its PHP code.

EXPLORER View:

- OPEN EDITORS: MenuController.php
- SINAR
 - app
 - Console
 - Exceptions
 - Http
 - Controllers
 - Admin
 - Auth
 - Customer
 - InventoryCh...
 - Kasir
 - Auth.rar
 - ContentsCo...
 - Controller.php
 - HomeContr...
 - MenuContro... (highlighted)
 - Middleware
 - Kernel.php
 - Providers

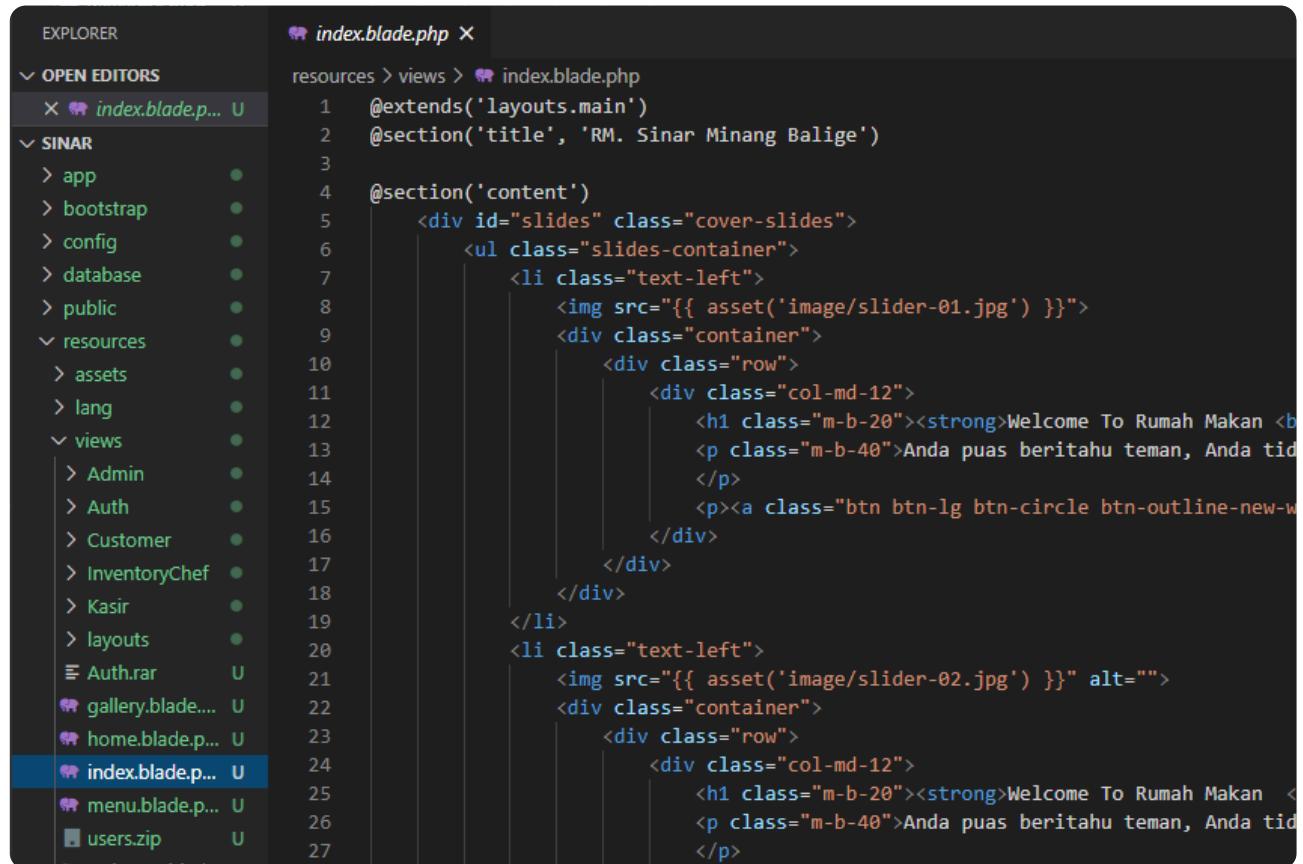
MenuController.php Content:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use DB;
6 use App\Menu;
7 use App\Pesanan;
8 use Illuminate\Http\Request;
9 use App\Http\Controllers\Controller;
10 use Illuminate\Support\Facades\Session;
11
12 class MenuController extends Controller
13 {
14     /**
15      * Display a listing of the resource.
16      *
17      * @return \Illuminate\Http\Response
18     */
19     public function index()
20     {
21         $data = Menu::all();
22         return view('menu',compact('data'));
23     }
24 }
```



Views

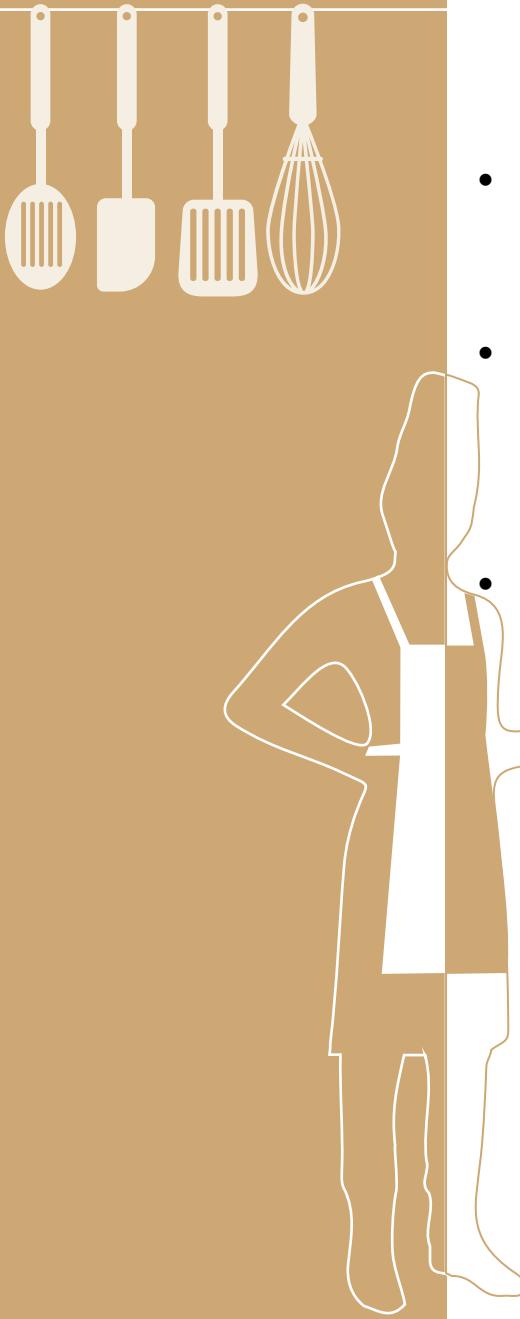
- View is a file contains code that will display the web design
- In laravel, the view directory is inside:
Resources > Views > ViewName.php
- View can use blade function to make it easier.
Resources > Views > View.blade.php



The screenshot shows a code editor with the following details:

- EXPLORER:** Shows the project structure under "SINAR".
- OPEN EDITORS:** Shows multiple files, with "index.blade.php" currently open.
- index.blade.php:** The content of the file is as follows:

```
resources > views > index.blade.php
1 @extends('layouts.main')
2 @section('title', 'RM. Sinar Minang Balige')
3
4 @section('content')
5     <div id="slides" class="cover-slides">
6         <ul class="slides-container">
7             <li class="text-left">
8                 
9                 <div class="container">
10                    <div class="row">
11                        <div class="col-md-12">
12                            <h1 class="m-b-20"><strong>Welcome To Rumah Makan <b>RM. Sinar Minang Balige</b></strong></h1>
13                            <p class="m-b-40">Anda puas beritahu teman, Anda tidak perlu membawa uang tunai</p>
14                            <p><a href="#" class="btn btn-lg btn-circle btn-outline-new-wallet">BELI</a></p>
15                        </div>
16                    </div>
17                </div>
18            </li>
19            <li class="text-left">
20                
21                <div class="container">
22                    <div class="row">
23                        <div class="col-md-12">
24                            <h1 class="m-b-20"><strong>Welcome To Rumah Makan <b>RM. Sinar Minang Balige</b></strong></h1>
25                            <p class="m-b-40">Anda puas beritahu teman, Anda tidak perlu membawa uang tunai</p>
26                        </div>
27                    </div>
28                </div>
29            </li>
30        </ul>
31    </div>
32
```



Models

- Model is one of the component that is directly related to the database
- Model is presented tables and contains attributes that become columns in the database table
- In laravel, model directory is in:

**app > Http >
ModelName.php**

The screenshot shows the VS Code interface with the Explorer and Editor panes. The Explorer pane shows a project structure under 'SINAR' with a 'app' folder containing various models like Console, Exceptions, Http, Providers, Bahan.php, Customer.php, Laporan.php, Meja.php, Menu.php, Pegawai.php, Pesanan.php, and User.php. The User.php file is currently selected in the Explorer. The Editor pane shows the code for the User.php model:

```
<?php  
namespace App;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Notifications\Notifiable;  
use Illuminate\Contracts\Auth\MustVerifyEmail;  
use Illuminate\Foundation\Auth\User as Authenticatable;  
  
class User extends Authenticatable implements MustVerifyEmail  
{  
    use Notifiable;  
  
    protected $table = 'users';  
    protected $primaryKey = 'id';  
  
    /**  
     * The attributes that are mass assignable.  
     *  
     * @var array  
     */  
    protected $fillable = [  
        'email', 'password'  
    ];
```



Thank you

Link presentasi : <https://youtu.be/1TzgjfjSS-g>

Link demo : <https://youtu.be/jD5zw9EzYpQ>