DAFTAR PUSTAKA

[1] E. Nurmiati, U. Islam, N. Syarif, and H. Jakarta, “Analisis Dan Perancangan Web Server Pada Handphone,” vol. 5, no. 2, pp. 1–17, 2012.

[2] Mauluddin Amras, “Aplikasi Web Browser Menggunakan Metode Url (Universal Resource Locator) Pada Sistem Operasi Window”, Vol. 4, No. 2, 2011.

[3] S.R. Anggiani, R. Eko, D. Asep, “Perancangan Sistem Informasi Berbasisi Website Subsistem Guru di Sekolah Pesantren Persatuan Islam 99 Rancangbango”, Garut, 2012.

[4] Syarif Aryana K.M, “Pengembangan Perangkat Lunak dengan Menggunakan PHP”, Jurnal Computech & Bisnis, Vol. 2, No.1, Bandung, 2008.

[5] A. Solichin, “Pemrograman Web dengan PHP dan MySQL,” Jakarta, 2005.

[6] R. Sovia and J. Febio, “Membangun Aplikasi E-Library Menggunakan Html, Php Script, Dan Mysql Database,” *J. Teknol. Inf. Pendidik.*, vol. 3, no. 1, pp. 86–101, 2011.

[7] P.Amelia, Halilulloh, N.Yunisca, “Pengaruh Pelaksanaan Kuliah Kerja ata (KKN) Terhadap Keterampilan Sosial Mahasiswa Program Studi kn Universitas Lampung, Lampung, 2013.

[8] Octavian Tri, “Web Multi E-Commerce Berbasis Framework Codeigniter”, Jurnal Teknologi dan Informatika, vol. 5, no. 1, 2015.

LAMPIRAN

```
<?php


namespace App\Http\Controllers\Auth;


use App\Http\Controllers\Controller;

use Illuminate\Foundation\Auth\SendsPasswordResetEmails;


class ForgotPasswordController extends Controller

{

    /*

    |--------------------------------------------------------------------------

    | Password Reset Controller

    |--------------------------------------------------------------------------

    |

    | Pengontrol ini bertanggung jawab untuk menangani email atur ulang kata
sandi dan

     | termasuk sifat yang membantu mengirimkan pemberitahuan ini dari

     | aplikasi Anda kepada pengguna Anda. Jangan ragu untuk menjelajahi sifat
ini.    |

    */


    use SendsPasswordResetEmails;




              te a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }
}
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
use Illuminate\Http\Request;
use Illuminate\Validation\ValidationException;

class LoginController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Login Controller
              --------------------------------------------------------------------

              ontrol ini menangani otentikasi pengguna untuk aplikasi dan
    | mengarahkan mereka ke layar beranda Anda. Pengontrol menggunakan suatu
sifat
    | untuk memberikan fungsionalitasnya dengan mudah ke aplikasi Anda.    |
    */

    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = '/';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }
```

```
     * Check either username or email.
     * @return string
     */

    public function username()
    {
        $identity  = request()->get('identity');
        $fieldName = filter_var($identity, FILTER_VALIDATE_EMAIL) ? 'email' :
'no_induk';
        request()->merge([$fieldName => $identity]);
        return $fieldName;
    }

    /**
     * Validate the user login.
     * @param Request $request
     */
    protected function validateLogin(Request $request)
    {
        $this->validate(
            $request,
            [
                'identity' => 'required|string',
                'password' => 'required|string',
            [
                'identity.required' => 'Masukkan Email / No. Induk',
                'password.required' => 'Masukkan Password',
            ]
        );
    }
    /**
     * @param Request $request
     * @throws ValidationException
     */
    protected function sendFailedLoginResponse(Request $request)
    {
        $request->session()->put('login_error', trans('auth.failed'));
        throw ValidationException::withMessages(
            [
                'error' => [trans('auth.failed')],
            ]
        );
    }
}
<?php

ce App\Http\Controllers\Auth;

User;
```

```
use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

use Illuminate\Foundation\Auth\RegistersUsers;


class RegisterController extends Controller

{

    /*

    |--------------------------------------------------------------------------

    | Register Controller

    |--------------------------------------------------------------------------

    |

    | Pengontrol ini menangani pendaftaran pengguna baru dan juga mereka

     | validasi dan pembuatan. Secara default pengontrol ini menggunakan sifat untuk

     | menyediakan fungsionalitas ini tanpa memerlukan kode tambahan apa pun.    |

    */


    use RegistersUsers;


    /**

     * Where to redirect users after registration.

     *

          r string
```

```
protected $redirectTo = '/home';

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest');
}

/**
 * Get a validator for an incoming registration request.
 *
 * @param  array  $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => 'required|string|max:255',
           mail' => 'required|string|email|max:255|unique:users',
           assword' => 'required|string|min:6|confirmed',
    }

    /**
     * Create a new user instance after a valid registration.
     *
     * @param  array  $data
     * @return \App\User
     */
    protected function create(array $data)
    {
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]);
    }
    public function showRegistrationForm() {

        return redirect(route('getRegister'));
    }
}
<?php

ce App\Http\Controllers\Auth;
```

```
use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

class ResetPasswordController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Password Reset Controller
    |--------------------------------------------------------------------------
    |
    | Pengontrol ini bertanggung jawab untuk menangani permintaan pengaturan ulang kata sandi
     | dan menggunakan sifat sederhana untuk memasukkan perilaku ini. Anda bebas untuk
     | jelajahi sifat ini dan timpa metode apa pun yang ingin Anda ubah.    |
    */

    use ResetsPasswords;

    /**
     * Where to redirect users after resetting their password.
     *
     * @var string

             ted $redirectTo = '/home';
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }
}
<?php

namespace App\Http\Middleware;

use Closure;

class Admin
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
            ram  \Closure  $next
            turn mixed
    public function handle($request, Closure $next) {

        if ($request->user()->role == 1) {
            return $next($request);
        }else {
            return redirect(route('unauthorized'));
        }
    }
}
<?php

namespace App\Http\Middleware;

use Illuminate\Cookie\Middleware\EncryptCookies as Middleware;

class EncryptCookies extends Middleware
{
    /**
     * The names of the cookies that should not be encrypted.
     *
     * @var array
     */
          ed $except = [
}
<?php

namespace App\Http\Middleware;

use Closure;

class Mahasiswa
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next) {

        if ($request->user()->role == 2) {
            return $next($request);
        }else {
            return redirect(route('unauthorized'));
             dd('Route Untuk Mahasiswa');
```

```
}
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Auth;

class RedirectIfAuthenticated
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @param  string|null  $guard
     * @return mixed
     */
    public function handle($request, Closure $next, $guard = null)
    {
        if (Auth::guard($guard)->check()) {
            return redirect('/home');



              rn $next($request);
    }
}
<?php

namespace App\Http\Middleware;

use Closure;

class Supervisor
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if ($request->user()->role == 3) {
            return $next($request);
        }else {
              turn redirect(route('unauthorized'));
}
<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\TrimStrings as Middleware;

class TrimStrings extends Middleware
{
    /**
     * The names of the attributes that should not be trimmed.
     *
     * @var array
     */
    protected $except = [
        'password',
        'password_confirmation',
    ];
}
<?php

namespace App\Http\Middleware;

inate\Http\Request;
oper\Proxy\TrustProxies as Middleware;
```

```
class TrustProxies extends Middleware
{
    /**
     * The trusted proxies for this application.
     *
     * @var array
     */
    protected $proxies;

    /**
     * The headers that should be used to detect proxies.
     *
     * @var int
     */
    protected $headers = Request::HEADER_X_FORWARDED_ALL;
}
<?php

namespace App\Http\Middleware;

use Closure;
```

```
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next) {

        if ($request->user()->role == 1) {
            return $next($request);
        }else {
            return redirect(route('unauthorized'));
        }
    }
}
<?php

namespace App\Http\Middleware;

use Illuminate\Foundation\Http\Middleware\VerifyCsrfToken as Middleware;

              ifyCsrfToken extends Middleware
     * The URIs that should be excluded from CSRF verification.
     *
     * @var array
     */
    protected $except = [
        //
    ];
}
<?php

namespace App\Http;

use App\Http\Middleware\Mahasiswa;
use App\Http\Middleware\Supervisor;
use App\Http\Middleware\Upt;
use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    /**
     * The application's global HTTP middleware stack.
     *
              se middleware are run during every request to your application.

              r array
```

```
 */
protected $middleware = [
    \Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
    \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
    \App\Http\Middleware\TrustProxies::class,
];

/**
 * The application's route middleware groups.
 *
 * @var array
 */
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        // \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
            lluminate\Routing\Middleware\SubstituteBindings::class,
    'api' => [
        'throttle:60,1',
        'bindings',
    ],
];

/**
 * The application's route middleware.
 *
 * These middleware may be assigned to groups or used individually.
 *
 * @var array
 */
protected $routeMiddleware = [
    'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
/*!
* Start Bootstrap - SB Admin 2 v3.3.7+1 (http://startbootstrap.com/template-overviews/sb-admin-2)
* Copyright 2013-2016 Start Bootstrap
* Licensed under MIT (https://github.com/BlackrockDigital/startbootstrap/blob/gh-pages/LICENSE)
*/
$(function() {
$('#side-menu').metisMenu();
});

// Memuat sidebar yang benar pada beban jendela,
// mengecilkan ukuran sidebar pada window.
// Atur tinggi minimum # pembungkus halaman ke ukuran jendela$(function() {
$(window).bind("load resize", function() {
    var topOffset = 50;
    var width = (this.window.innerWidth > 0) ? this.window.innerWidth : this.screen.width;
    if (width < 768) {
        $('div.navbar-collapse').addClass('collapse');
        topOffset = 100; // 2-row-menu
    } else {
        $('div.navbar-collapse').removeClass('collapse');
    }
    var height = ((this.window.innerHeight > 0) ? this.window.innerHeight :
this.screen.height) - 1;

    height = height - topOffset;

    if (height < 1) height = 1;

    if (height > topOffset) {

        $("#page-wrapper").css("min-height", (height) + "px");

    }

});


var url = window.location;

// var element = $('ul.nav a').filter(function() {

//     return this.href == url;

// }).addClass('active').parent().parent().addClass('in').parent();

var element = $('ul.nav a').filter(function() {

    return this.href == url;

}).addClass('active').parent();


while (true) {

    if (element.is('li')) {

        element = element.parent().addClass('in').parent();

    } else {

        break;

    }
```

```
function forceNumber(e) {

  var keyCode = e.keyCode ? e.keyCode : e.which;

  if((keyCode < 48 || keyCode > 58)  && keyCode != 188 && keyCode != 8 &&
keyCode != 9 && keyCode != 46) {

      return false;

      event.preventDefault();

  }

  return true;

}

<?php


return [


    /*

    |--------------------------------------------------------------------------

    | Default Database Connection Name

    |--------------------------------------------------------------------------

    |

    | Di sini Anda dapat menentukan koneksi database mana di bawah yang Anda
inginkan

     | untuk digunakan sebagai koneksi default Anda untuk semua pekerjaan basis
data. Tentu saja

     | Anda dapat menggunakan banyak koneksi sekaligus menggunakan
[illegible]kaan Database.
```

```
    'default' => env('DB_CONNECTION', 'mysql'),


    /*
    |--------------------------------------------------------------------------
    | Database Connections
    |--------------------------------------------------------------------------
    |
   | Berikut adalah masing-masing pengaturan koneksi database untuk aplikasi
Anda.
     | Tentu saja, contoh konfigurasi setiap platform basis data yang ada
     | didukung oleh Laravel ditunjukkan di bawah ini untuk mempermudah
pengembangan.
     |   |
    | Semua pekerjaan basis data di Laravel dilakukan melalui fasilitas PHP PDO
     | jadi pastikan Anda memiliki driver untuk basis data khusus Anda
     | pilihan diinstal pada mesin Anda sebelum Anda memulai pengembangan.
     |   */


    'connections' => [


        'sqlite' => [
            'driver' => 'sqlite',
                     atabase' => env('DB_DATABASE', database_path('database.sqlite')),
                     refix' => '',
        'mysql' => [
            'driver' => 'mysql',
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'strict' => true,
            'engine' => null,
        ],

        'pgsql' => [
            'driver' => 'pgsql',
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '5432'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
              assword' => env('DB_PASSWORD', ''),
              harset' => 'utf8',
              refix' => '',
            'schema' => 'public',
            'sslmode' => 'prefer',
        ],

        'sqlsrv' => [
            'driver' => 'sqlsrv',
            'host' => env('DB_HOST', 'localhost'),
            'port' => env('DB_PORT', '1433'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'charset' => 'utf8',
            'prefix' => '',
        ],

    ],

    /*
    |--------------------------------------------------------------------------
    | Migration Repository Table
    |--------------------------------------------------------------------------
    |
               i melacak semua migrasi yang sudah berjalan untuk
               asi Anda. Dengan menggunakan informasi ini, kita dapat menentukan
```

```
     | migrasi pada disk belum benar-benar dijalankan di database.

     |   */


    'migrations' => 'migrations',


    /*

    |--------------------------------------------------------------------------

    | Redis Databases

    |--------------------------------------------------------------------------

    |

Redis adalah toko nilai kunci open source, cepat, dan canggih juga

     | memberikan seperangkat perintah yang lebih kaya daripada sistem nilai kunci
yang khas

     | seperti APC atau Memcached. Laravel membuatnya mudah digali.

     |   */


    'redis' => [


        'client' => 'predis',


        'default' => [

            'host' => env('REDIS_HOST', '127.0.0.1'),

            'password' => env('REDIS_PASSWORD', null),

              ort' => env('REDIS_PORT', 6379),

              atabase' => 0,
        ],

    ],

];
<?php

return [

    /*
    |--------------------------------------------------------------------------
    | Third Party Services
    |--------------------------------------------------------------------------
        | File ini untuk menyimpan kredensial untuk layanan pihak ketiga tersebut
     | seperti Stripe, Mailgun, SparkPost dan lainnya. File ini memberikan waras
     | lokasi default untuk jenis informasi ini, yang memungkinkan paket
     | untuk memiliki tempat konvensional untuk menemukan berbagai kredensial
Anda.
     |   */

    'mailgun' => [
        'domain' => env('MAILGUN_DOMAIN'),
        'secret' => env('MAILGUN_SECRET'),
    'ses' => [
        'key' => env('SES_KEY'),
        'secret' => env('SES_SECRET'),
        'region' => env('SES_REGION', 'us-east-1'),
    ],

    'sparkpost' => [
        'secret' => env('SPARKPOST_SECRET'),
    ],

    'stripe' => [
        'model' => App\User::class,
        'key' => env('STRIPE_KEY'),
        'secret' => env('STRIPE_SECRET'),
    ],

];
```