**NAMA : AZIZ WIBOWO**

**NIM :1900018194**

**KLS : D**

**UTS PRPL**

1. Penerapan Perintah git pada Vsual Studi Codes

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git add .

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: login.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git commit -m "penerapan

perintah git pada penambahan file baru login.php"

[master 9c843db] penerapan perintah git pada penambahan file baru login.php

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 login.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git log --oneline

9c843db (HEAD -> master) penerapan perintah git pada penambahan file baru login.php

4d3d6bf penerapan perintah git log.css

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git add .

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: pendaftaran.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git commit -m "perintah git add file pendaftaran.php"

[master 695cd92] perintah git add file pendaftaran.php

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 pendaftaran.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git log --oneline

695cd92 (HEAD -> master) perintah git add file pendaftaran.php

9c843db penerapan perintah git pada penambahan file baru login.php

4d3d6bf penerapan perintah git log.css

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git add .

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git status

On branch master

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: user.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git commit -m "perintah git add file user.php"

[master 84c78e5] perintah git add file user.php

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 user.php

PS C:\Users\U S E R\Documents\PRPL\Proyek\TA\_PRPL\_190018194\_AZIZ WIBOWO> git log --oneline

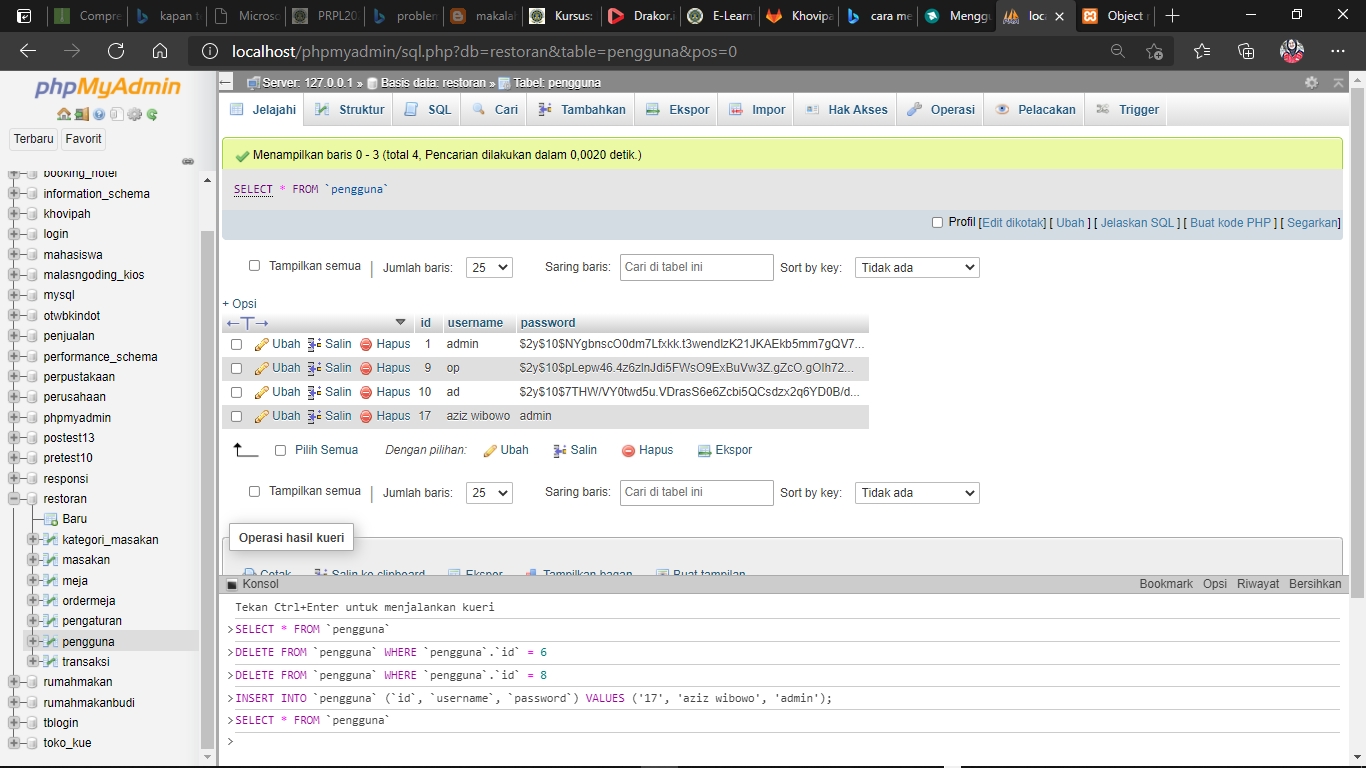
84c78e5 (HEAD -> master) perintah git add file user.php

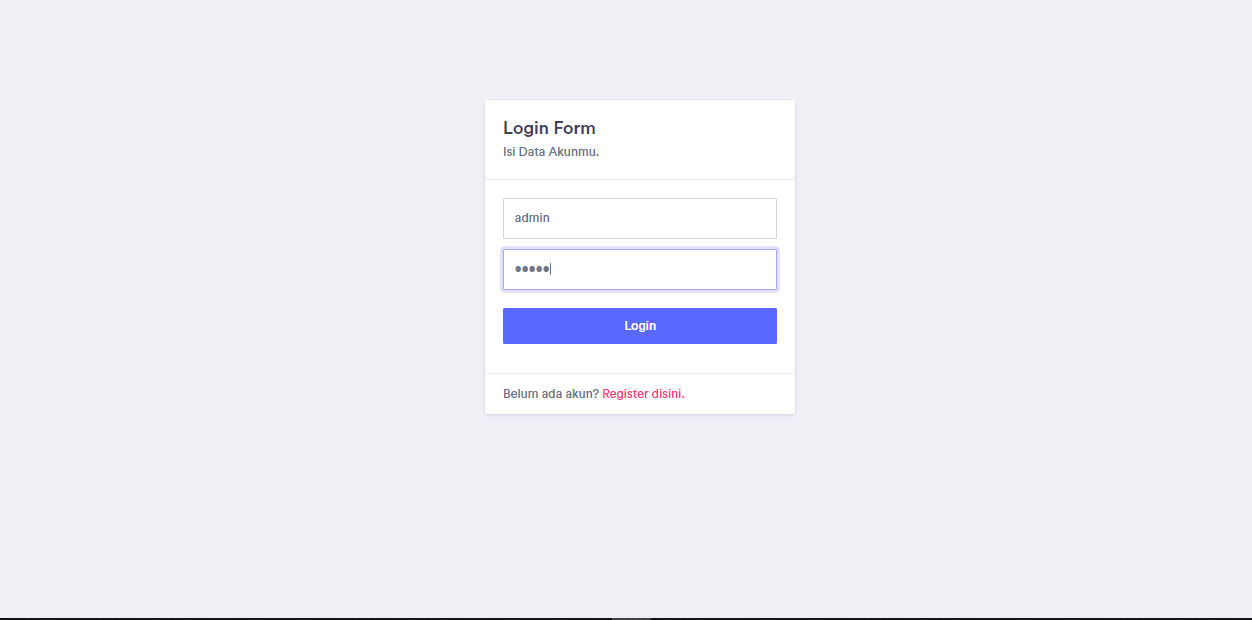
695cd92 perintah git add file pendaftaran.php

9c843db penerapan perintah git pada penambahan file baru login.php

4d3d6bf penerapan perintah git log.css

1. Fitur sign in dan Sign up beserta DATABASE





Login => Sign in

REgister => Sign up

**5.**Analisa Source Code

Disini saya membuat laman login dan Sign up dengan satu program yang dimana menggunakan json untuk pendeteksian Data inputan.  JavaScript Object Notation, sebagai format untuk berbagi data, JSON sendiri di turunkan langsung dari javascript. Sebelumnya saya sudah menghubungkan laman Sign in dan sign up dengan program json yang telah saya buat.

Berikut Analisa sorce code :

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Auth extends CI\_Controller {//class Auth untuk data Login dan Sign up

public function \_\_construct()

{

parent::\_\_construct();

$this->load->model('auth\_model');

}

//fungsi login terdapat percabangan untuk penyeleksian username dan password, jika username dan password memiliki status “ada” di data maka setelah login akan menuju ke laman Dashboard

Dan jika username dan password salah maka laman akan nge stuck dibagian form login

public function login() {

if ($this->session->userdata('status') === 'login') {

redirect('/');

}

if ($username = $this->input->post('username')) {

$password = $this->input->post('password');

echo json\_encode($this->auth\_model->login($username,$password));

} else {

$this->load->view('auth/login');

}

}

//untuk fungsi Register/Sign up itu merupakan tempat untuk membuat /mendaftarkan akun anda kedalam sebuah data. Dalam program yang saya buat ini cukup sederhana.

public function register()

{

if ($this->session->userdata('status') === 'login') {

redirect('/');

}

if ($username = $this->input->post('username')) {//proses penginputan data kemudian jika data yang diinputkan sesuai maka data tersebut akan di UPDATE dan pemilik akun tersebut akan terdata pada suatu aplikasi

$password = $this->input->post('username');

if ($user = $this->auth\_model->register($username,$password)) {

$this->session->set\_userdata($user);

echo json\_encode('sukses');//dihubungkan dengan javascript

} else {

echo json\_encode('gagal');

}

} else {

$this->load->view('auth/register');

}

}

public function logout()

{

$this->session->sess\_destroy();

redirect('/');

}

}

/\* End of file Auth.php \*/

/\* Location: ./application/controllers/Auth.php \*/

Analisa:

Dari aspek kesederhanaan :

Fitur yang saya buat kali ini memiliki tampilann yang sederhana dan sangat masih awam sekali, pada sign up saya menggunakan database MySQL untuk menyimpan data hasil akses yang di gunakan oleh user, laman login terdapaat opsi “Resgister” artinya disitu dapat dii lakkukan Sign up untuk yang belum meilliki akses ke aplikasi tersebut

-Sign in pada sign in kita hanya perlu unutk mengisi username dan Password yang telah kita buat sebelumnya. Disini pada program yang dibuat form login di buat dengan menggunakan database ringan dan pada programnya menggunakan Boostrap4 dan juga Json encode dan dalam desain terlihat lebih simple dalam penggunaanya. Setelah berhasil login maka akan menuju ke tampilan Dashboard

-Kompilasi dengan program di waktu kuliah

Laman sign in dan sign up yang telah saya buat menggunakan konsep yang dibuat pada skrip program yang dibuat sebelumnya hanya saja form yang saya buat kali ini terlihat lebih elegant dan simple dan juga tidak ada batasan untuk penamaan yang dengan kategori huruf kapital ataupun biasa.

**6.** Program API menggunakan RESTFUL API

postControler.php

<?php

namespace App\Http\Controllers\api\v1;

use App\Http\Controllers\Controller;

use App\Post;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

class PostsController extends Controller

{

/\*\*

\* @return \Illuminate\Contracts\Routing\ResponseFactory|\Illuminate\Http\Response

\*/

public function index()

{

$posts = Post::latest()->get();

return response([

'success' => true,

'message' => 'List Semua Posts',

'data' => $posts

], 200);

}

/\*\*

\* @param Request $request

\* @return \Illuminate\Http\JsonResponse

\*/

public function store(Request $request)

{

//validate data

$validator = Validator::make($request->all(), [

'title' => 'required',

'content' => 'required',

],

[

'title.required' => 'Masukkan Title Post !',

'content.required' => 'Masukkan Content Post !',

]

);

if($validator->fails()) {

return response()->json([

'success' => false,

'message' => 'Silahkan Isi Bidang Yang Kosong',

'data' => $validator->errors()

],401);

} else {

$post = Post::create([

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

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

]);

if ($post) {

return response()->json([

'success' => true,

'message' => 'Post Berhasil Disimpan!',

], 200);

} else {

return response()->json([

'success' => false,

'message' => 'Post Gagal Disimpan!',

], 401);

}

}

}

/\*\*

\* @param $id

\* @return \Illuminate\Http\JsonResponse

\*/

public function show($id)

{

$post = Post::whereId($id)->first();

if ($post) {

return response()->json([

'success' => true,

'message' => 'Detail Post!',

'data' => $post

], 200);

} else {

return response()->json([

'success' => false,

'message' => 'Post Tidak Ditemukan!',

'data' => ''

], 401);

}

}

}

Api.php

Route::get('/v1/posts/{id?}', 'api\v1\PostsController@show');