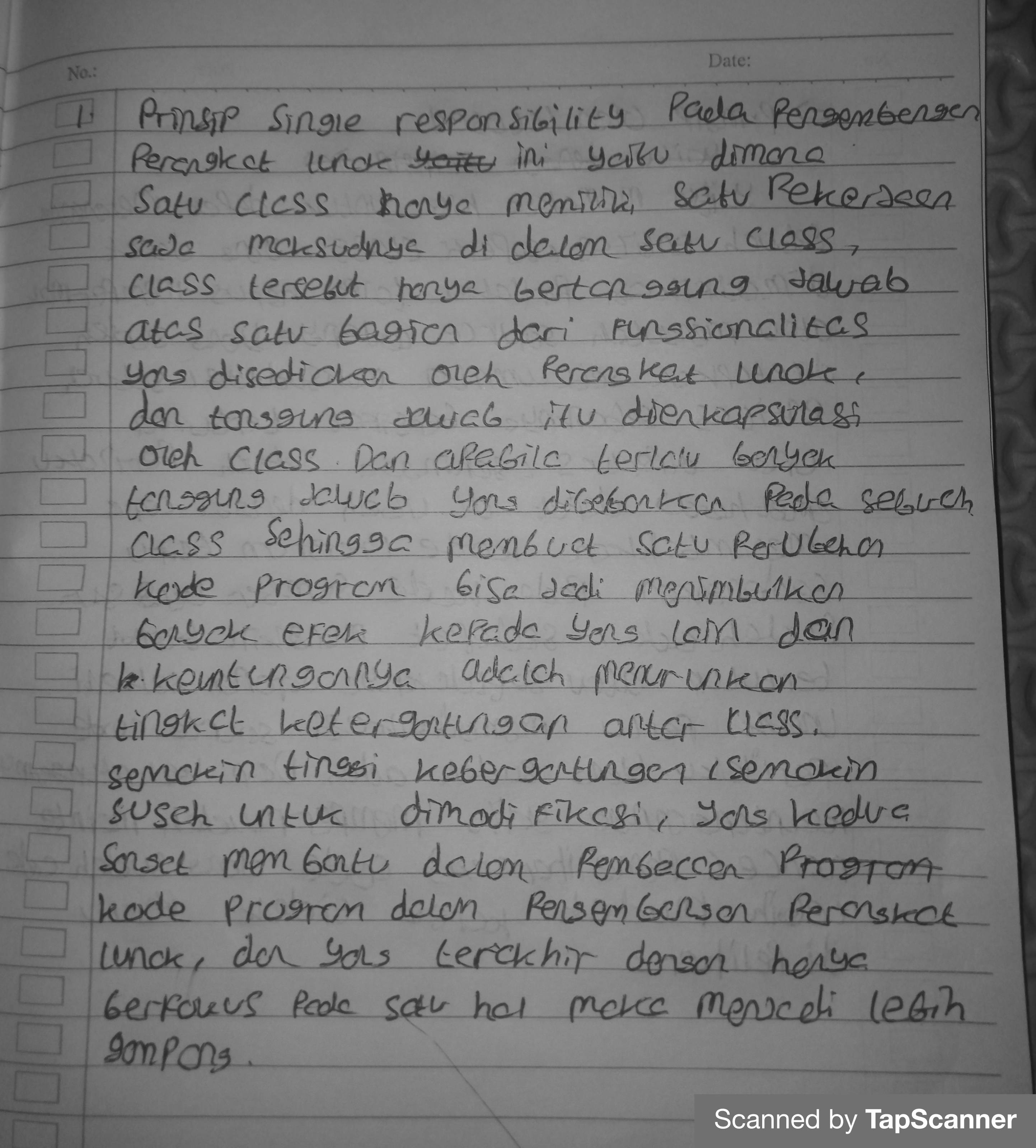
NAMA: HERDIN ASMARA TIMOR

NIM: 1900018212

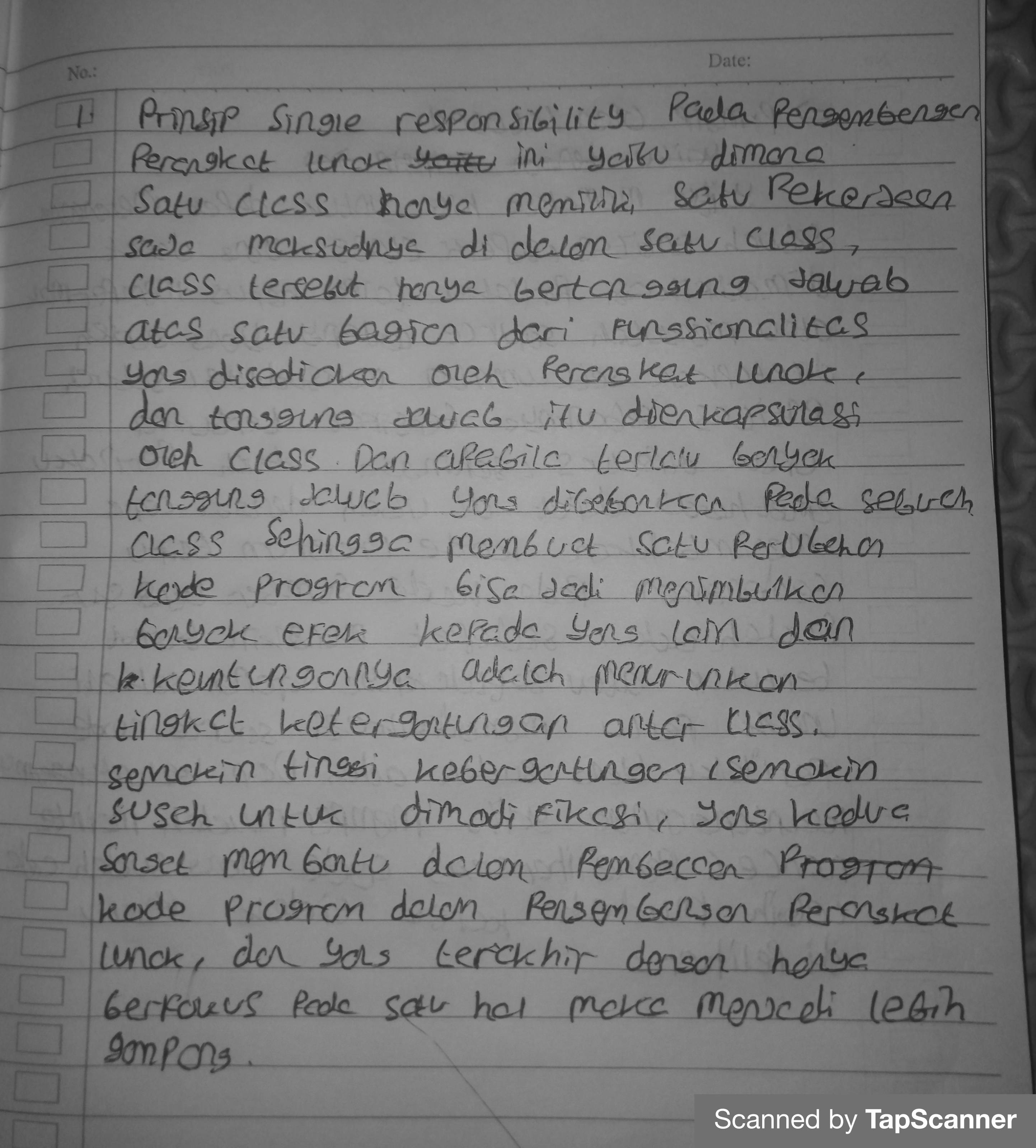
KELAS: D

UAS Pengantar Rekayasa Perangkat Lunak

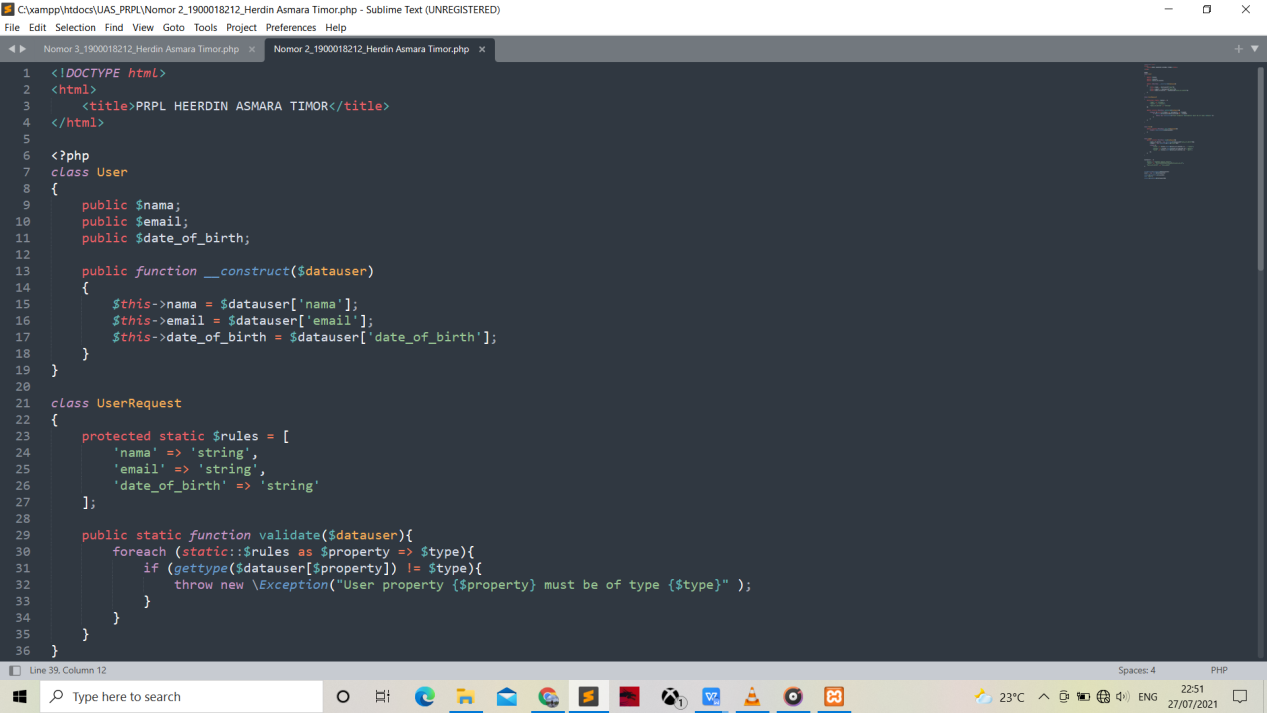
1. **=>Prinsip single responsibility**

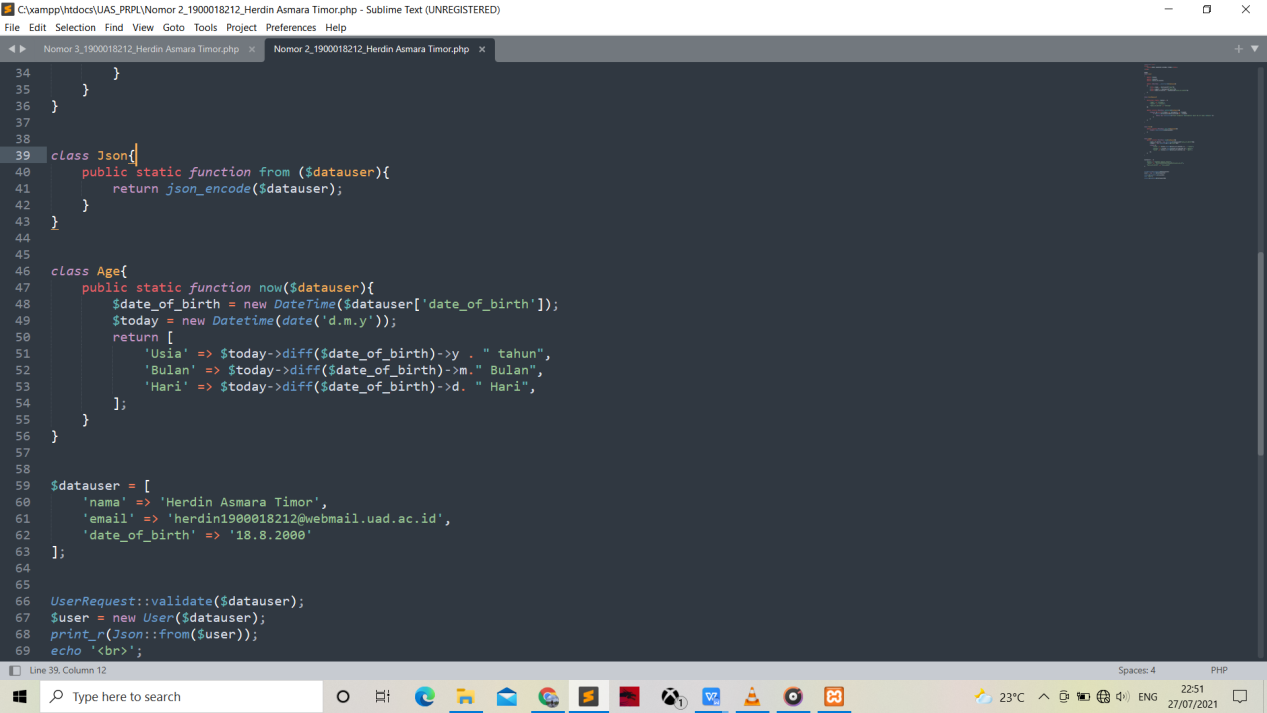


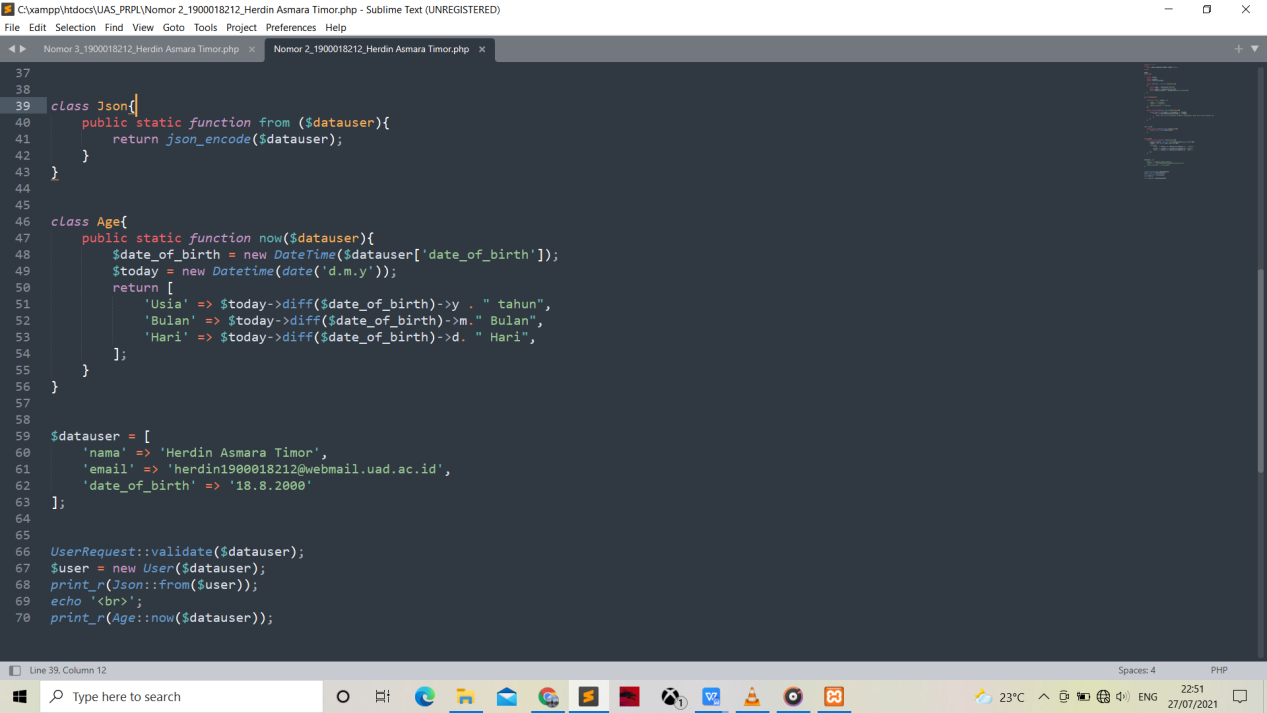
**=>Prinsip open/closed**



2.**Source code**







<!DOCTYPE html>

<html>

<title>PRPL HEERDIN ASMARA TIMOR</title>

</html>

<?php

class User

{

public $nama;

public $email;

public $date\_of\_birth;

public function \_\_construct($datauser)

{

$this->nama = $datauser['nama'];

$this->email = $datauser['email'];

$this->date\_of\_birth = $datauser['date\_of\_birth'];

}

}

class UserRequest

{

protected static $rules = [

'nama' => 'string',

'email' => 'string',

'date\_of\_birth' => 'string'

];

public static function validate($datauser){

foreach (static::$rules as $property => $type){

if (gettype($datauser[$property]) != $type){

throw new \Exception("User property {$property} must be of type {$type}" );

}

}

}

}

class Json{

public static function from ($datauser){

return json\_encode($datauser);

}

}

class Age{

public static function now($datauser){

$date\_of\_birth = new DateTime($datauser['date\_of\_birth']);

$today = new Datetime(date('d.m.y'));

return [

'Usia' => $today->diff($date\_of\_birth)->y . " tahun",

'Bulan' => $today->diff($date\_of\_birth)->m." Bulan",

'Hari' => $today->diff($date\_of\_birth)->d. " Hari",

];

}

}

$datauser = [

'nama' => 'Herdin Asmara Timor',

'email' => 'herdin1900018212@webmail.uad.ac.id',

'date\_of\_birth' => '18.8.2000'

];

UserRequest::validate($datauser);

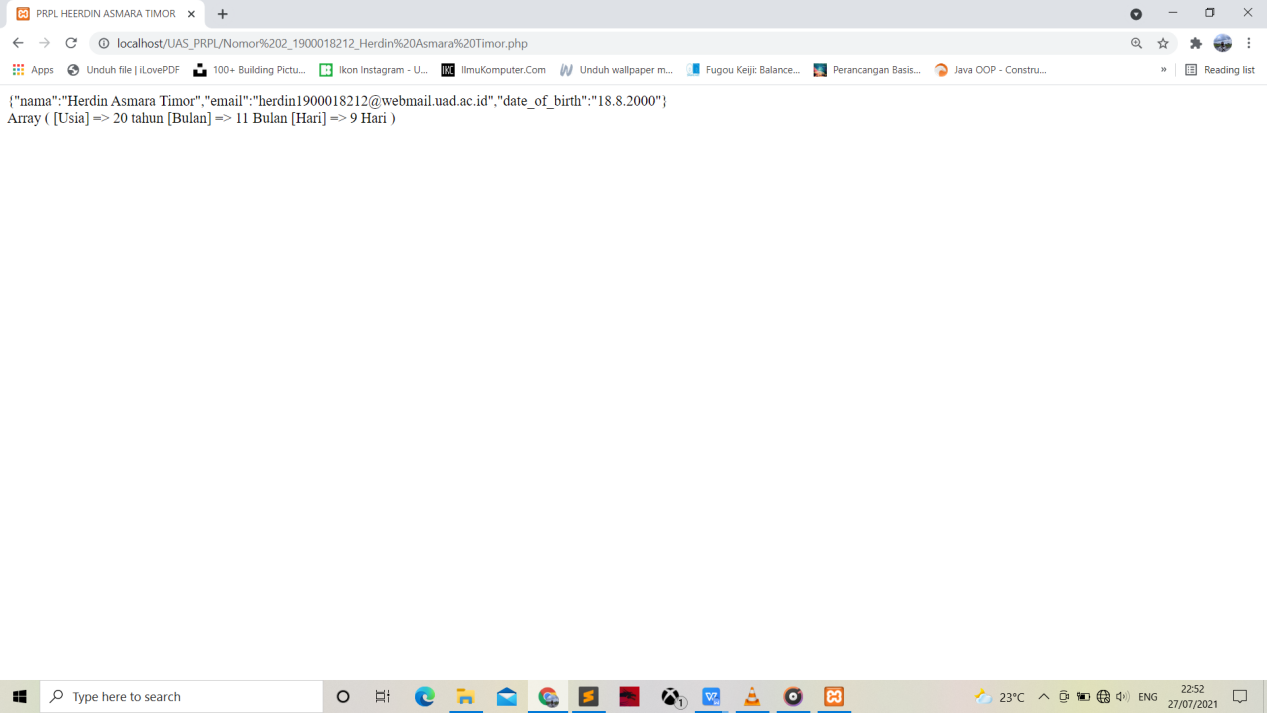
$user = new User($datauser);

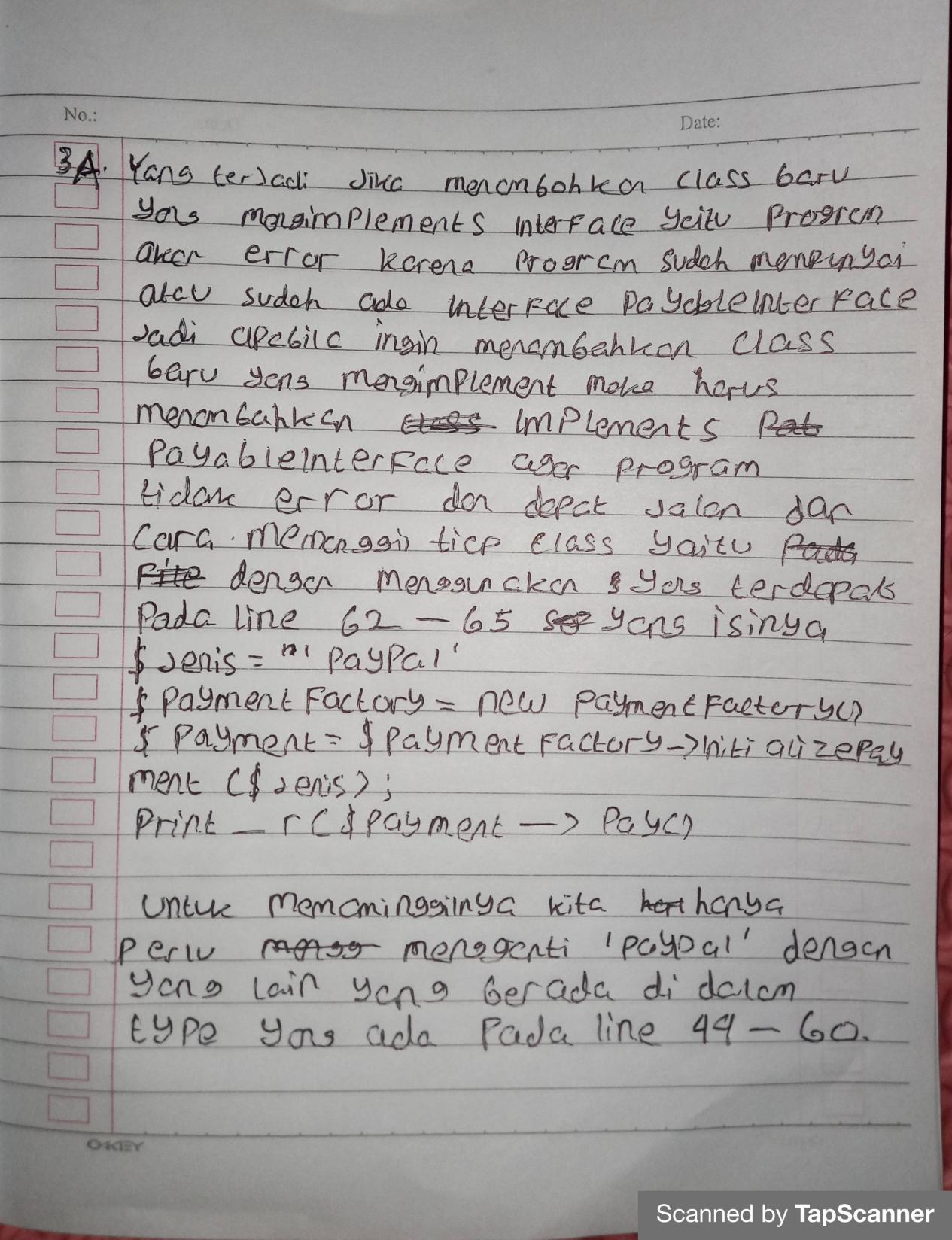
print\_r(Json::from($user));

echo '<br>';

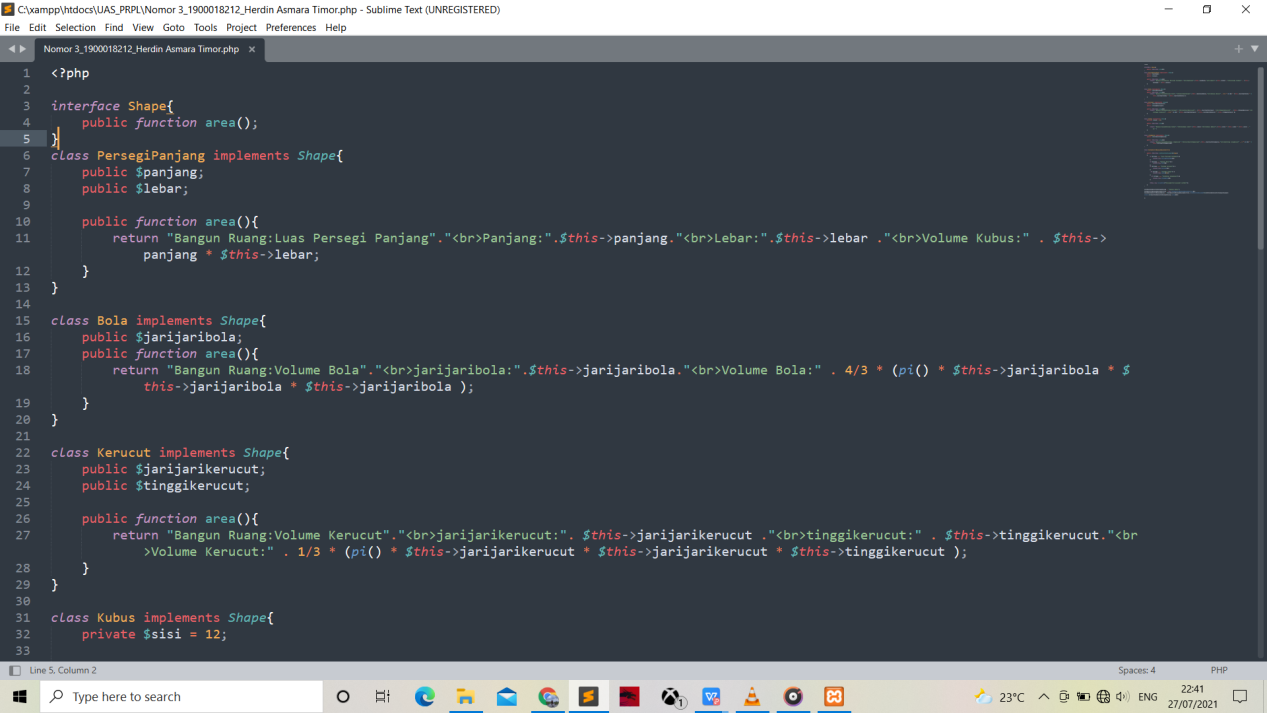
print\_r(Age::now($datauser));

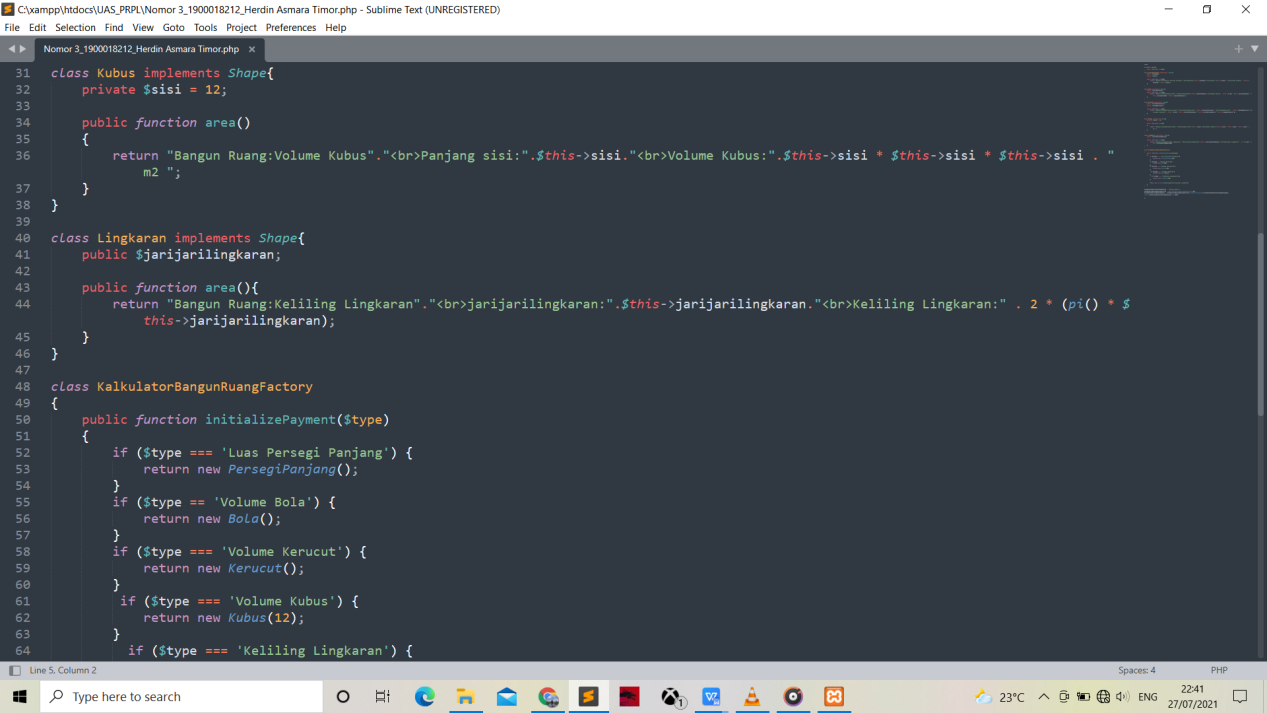
**Hasil**

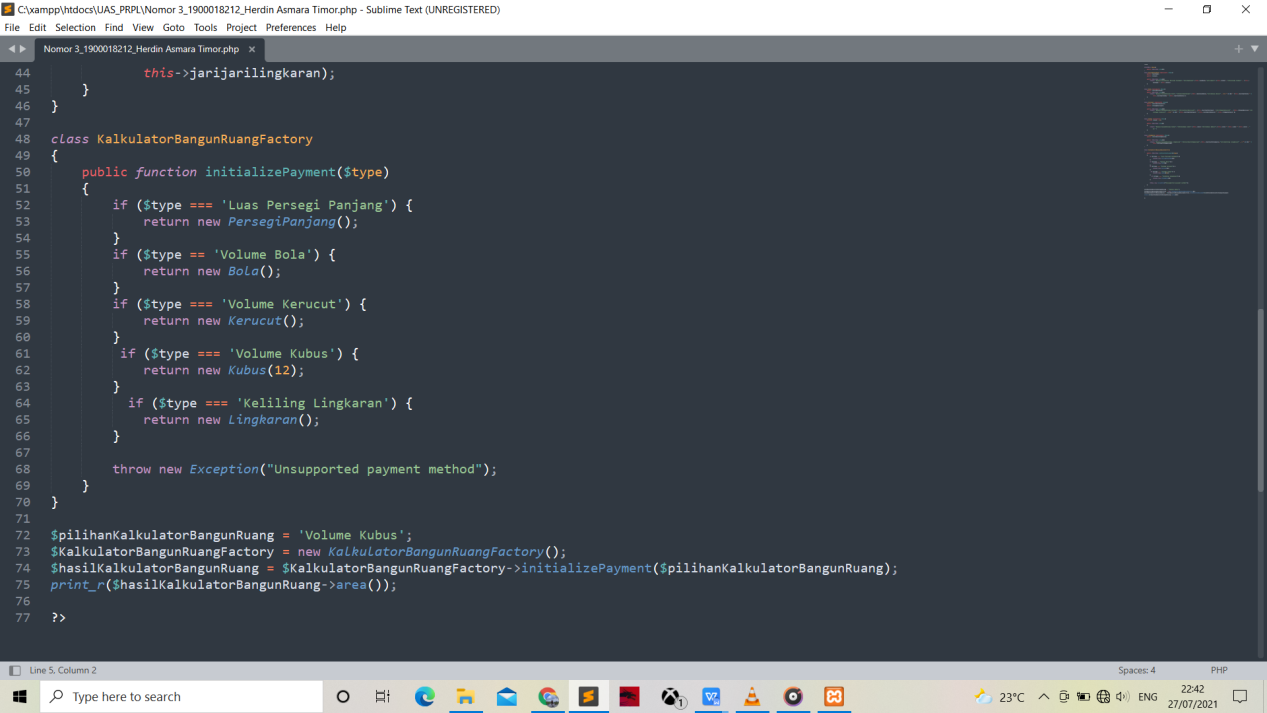


1. **A. **

**B.Source code**







<?php

interface Shape{

public function area();

}

class PersegiPanjang implements Shape{

public $panjang;

public $lebar;

public function area(){

return "Bangun Ruang:Luas Persegi Panjang"."<br>Panjang:".$this->panjang."<br>Lebar:".$this->lebar ."<br>Volume Kubus:" . $this->panjang \* $this->lebar;

}

}

class Bola implements Shape{

public $jarijaribola;

public function area(){

return "Bangun Ruang:Volume Bola"."<br>jarijaribola:".$this->jarijaribola."<br>Volume Bola:" . 4/3 \* (pi() \* $this->jarijaribola \* $this->jarijaribola \* $this->jarijaribola );

}

}

class Kerucut implements Shape{

public $jarijarikerucut;

public $tinggikerucut;

public function area(){

return "Bangun Ruang:Volume Kerucut"."<br>jarijarikerucut:". $this->jarijarikerucut ."<br>tinggikerucut:" . $this->tinggikerucut."<br>Volume Kerucut:" . 1/3 \* (pi() \* $this->jarijarikerucut \* $this->jarijarikerucut \* $this->tinggikerucut );

}

}

class Kubus implements Shape{

private $sisi = 12;

public function area()

{

return "Bangun Ruang:Volume Kubus"."<br>Panjang sisi:".$this->sisi."<br>Volume Kubus:".$this->sisi \* $this->sisi \* $this->sisi . " m2 ";

}

}

class Lingkaran implements Shape{

public $jarijarilingkaran;

public function area(){

return "Bangun Ruang:Keliling Lingkaran"."<br>jarijarilingkaran:".$this->jarijarilingkaran."<br>Keliling Lingkaran:" . 2 \* (pi() \* $this->jarijarilingkaran);

}

}

class KalkulatorBangunRuangFactory

{

public function initializePayment($type)

{

if ($type === 'Luas Persegi Panjang') {

return new PersegiPanjang();

}

if ($type == 'Volume Bola') {

return new Bola();

}

if ($type === 'Volume Kerucut') {

return new Kerucut();

}

if ($type === 'Volume Kubus') {

return new Kubus(12);

}

if ($type === 'Keliling Lingkaran') {

return new Lingkaran();

}

throw new Exception("Unsupported payment method");

}

}

$pilihanKalkulatorBangunRuang = 'Volume Kubus';

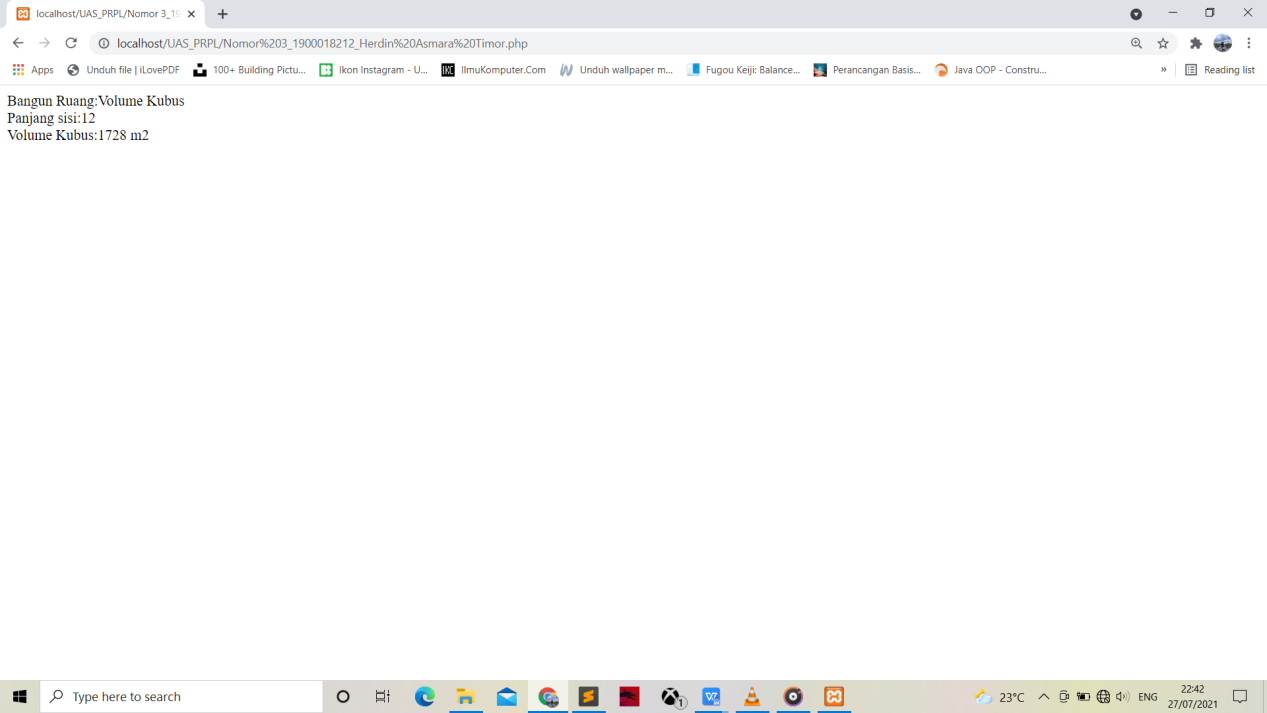
$KalkulatorBangunRuangFactory = new KalkulatorBangunRuangFactory();

$hasilKalkulatorBangunRuang = $KalkulatorBangunRuangFactory->initializePayment($pilihanKalkulatorBangunRuang);

print\_r($hasilKalkulatorBangunRuang->area());

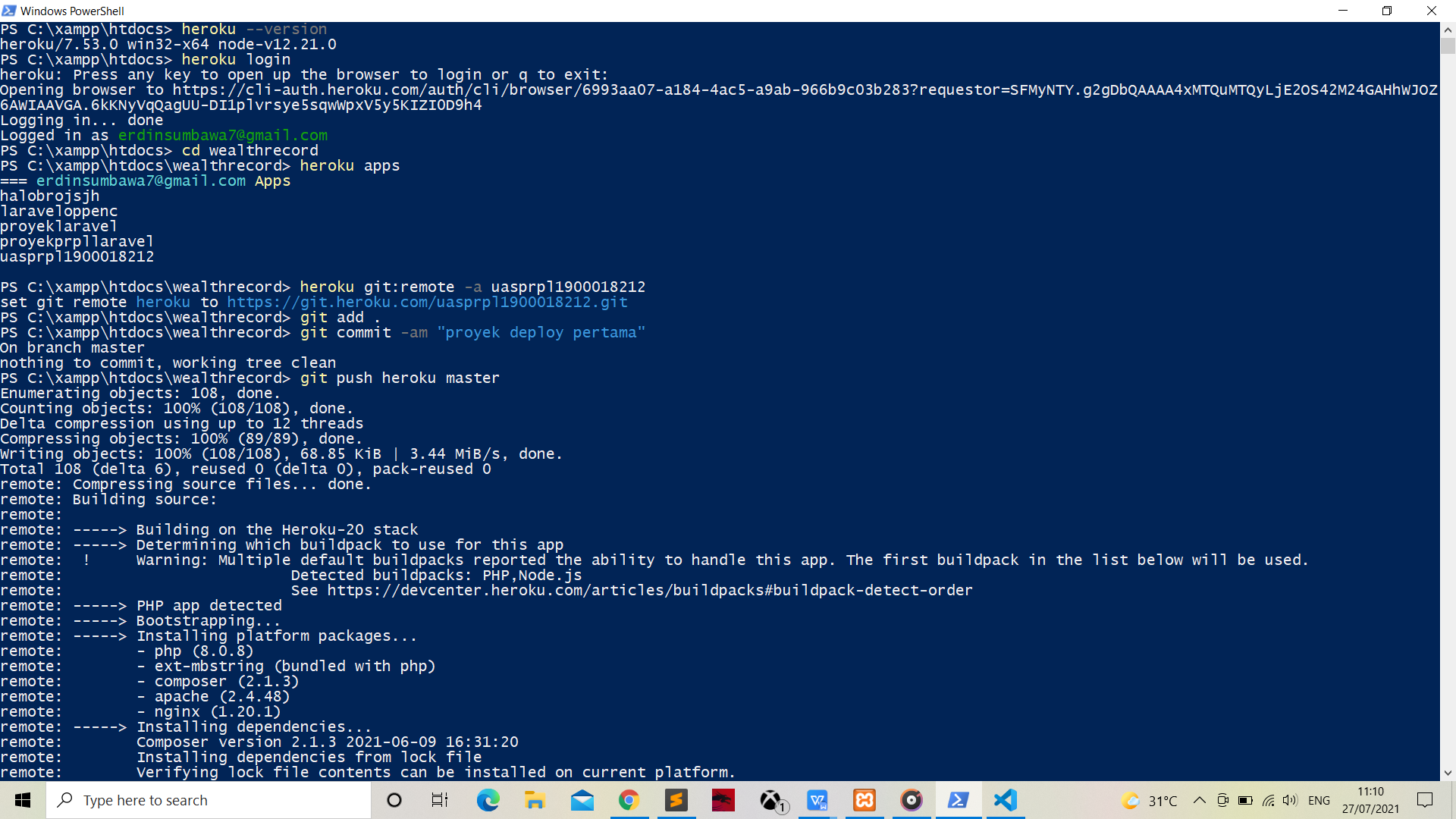
?>

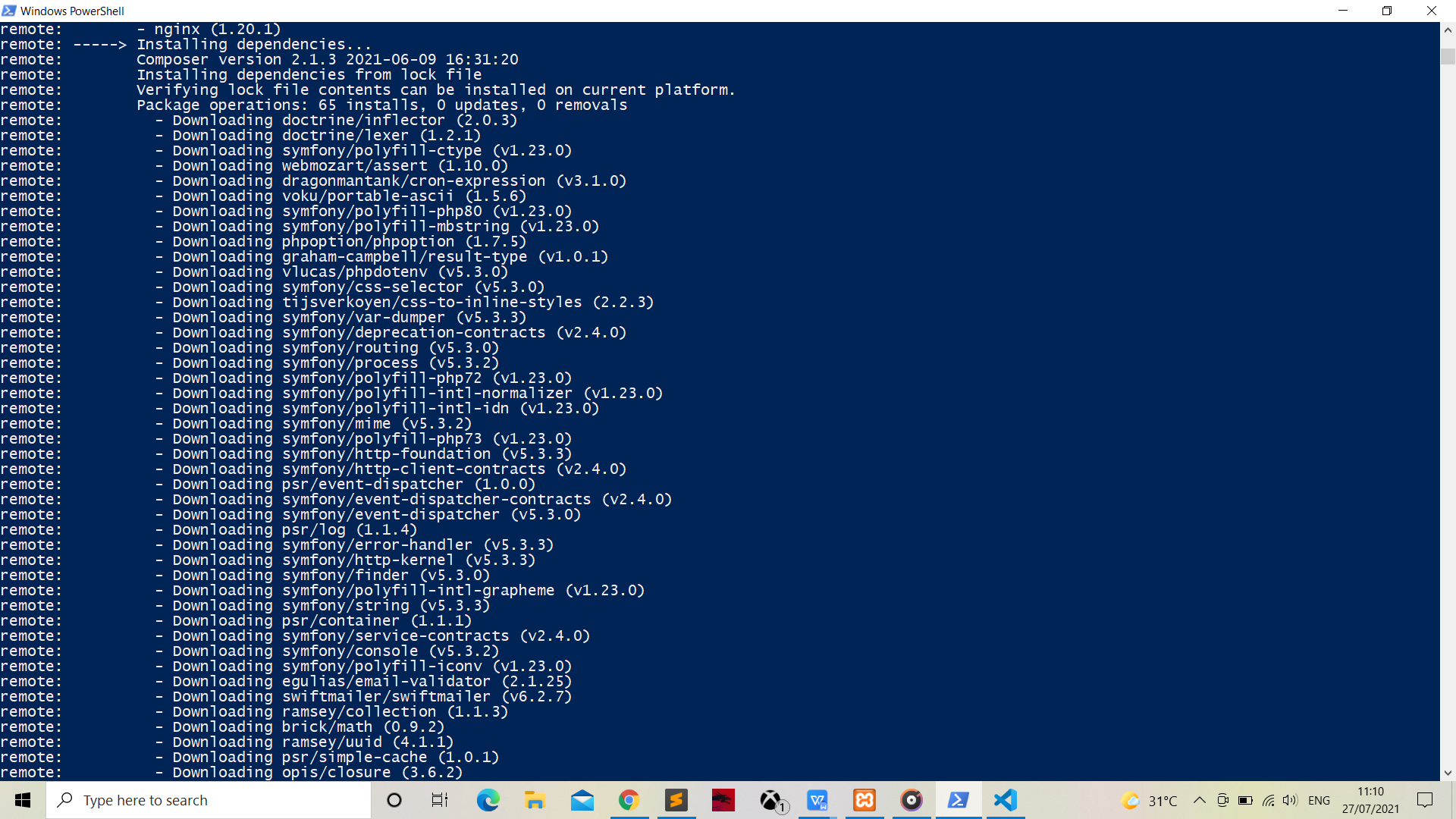
**HASIL SOURCE CODE**

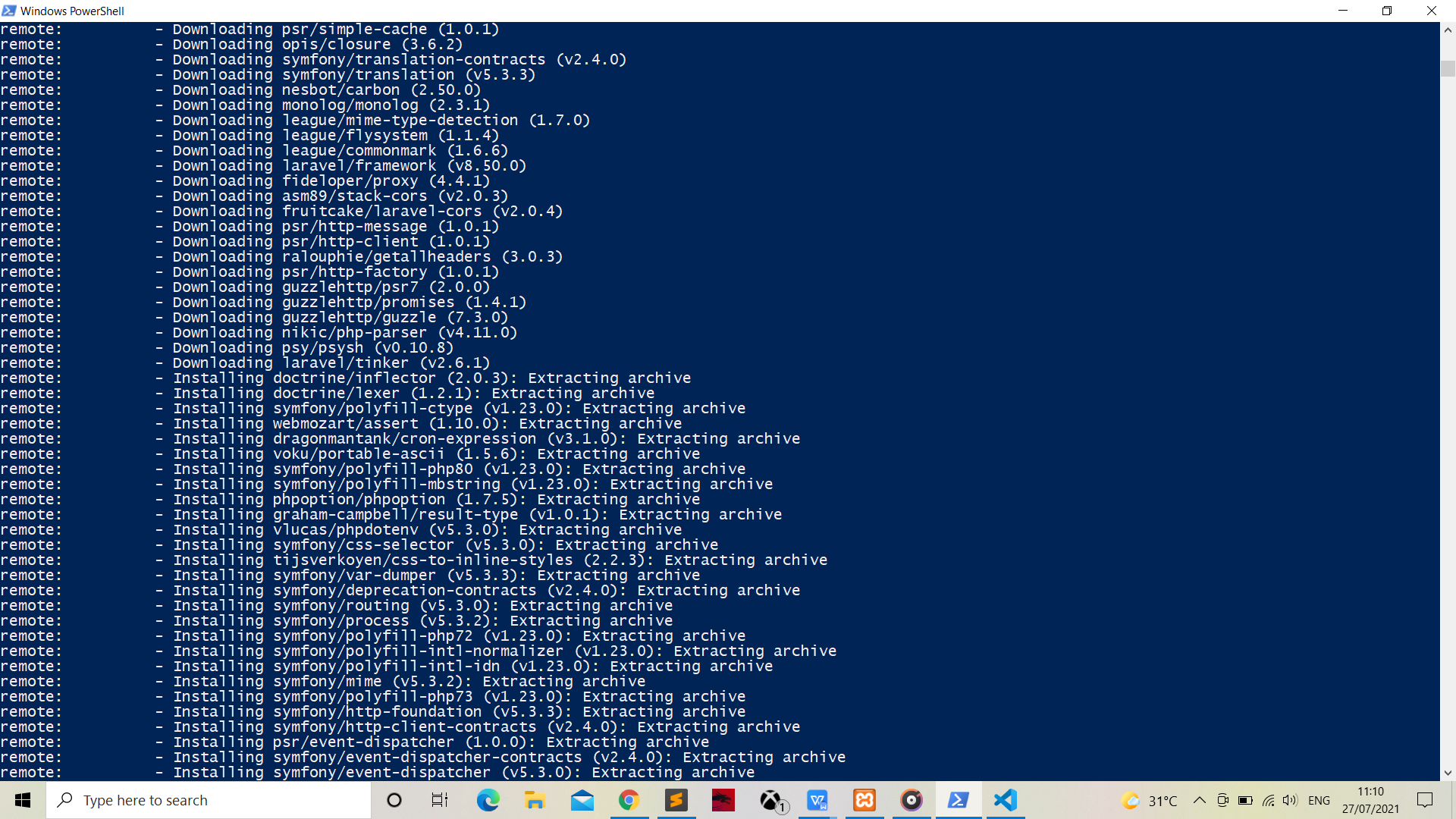


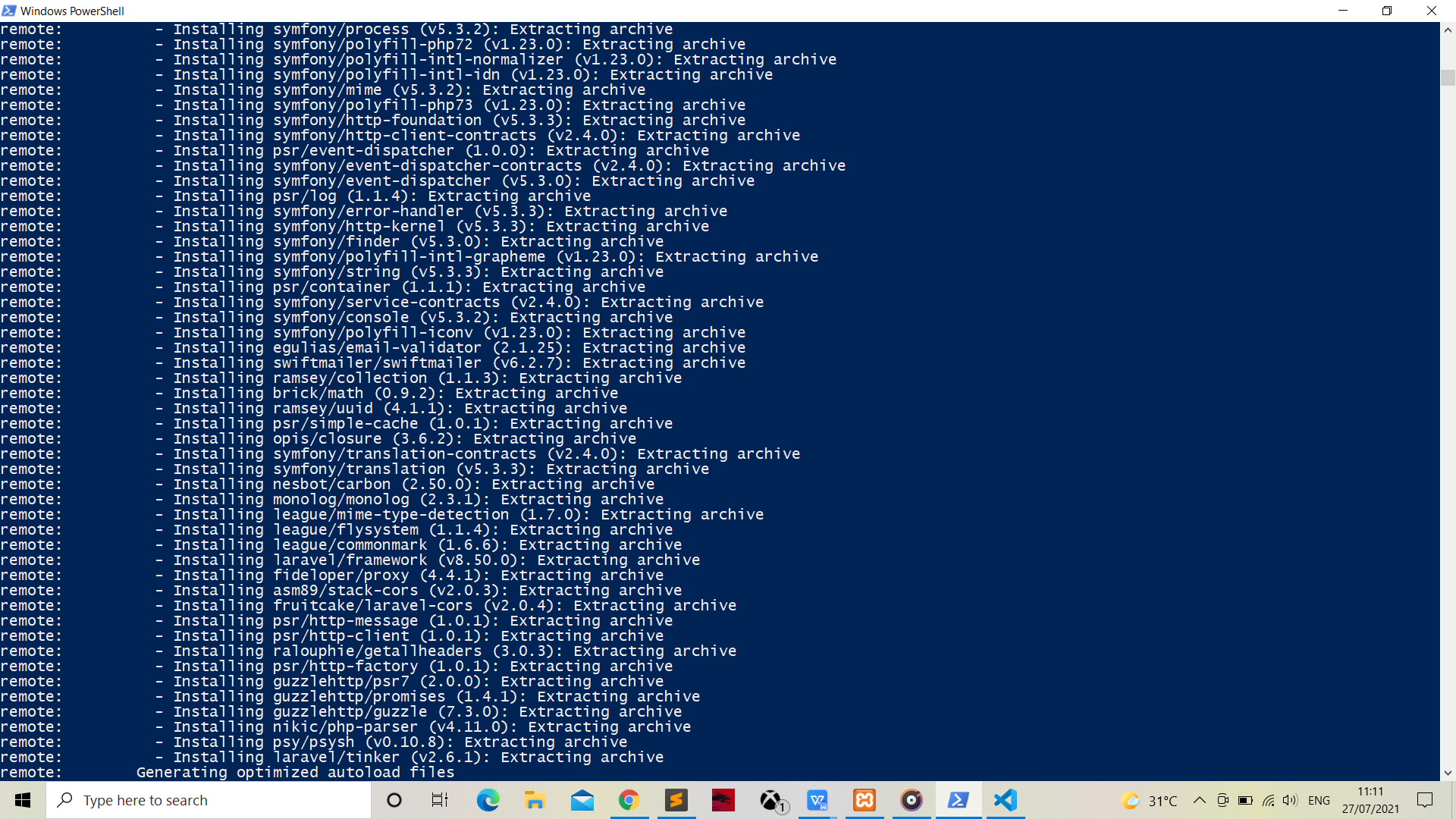
1. **Deployment proyek perangkat lunak berbasis Laravel**

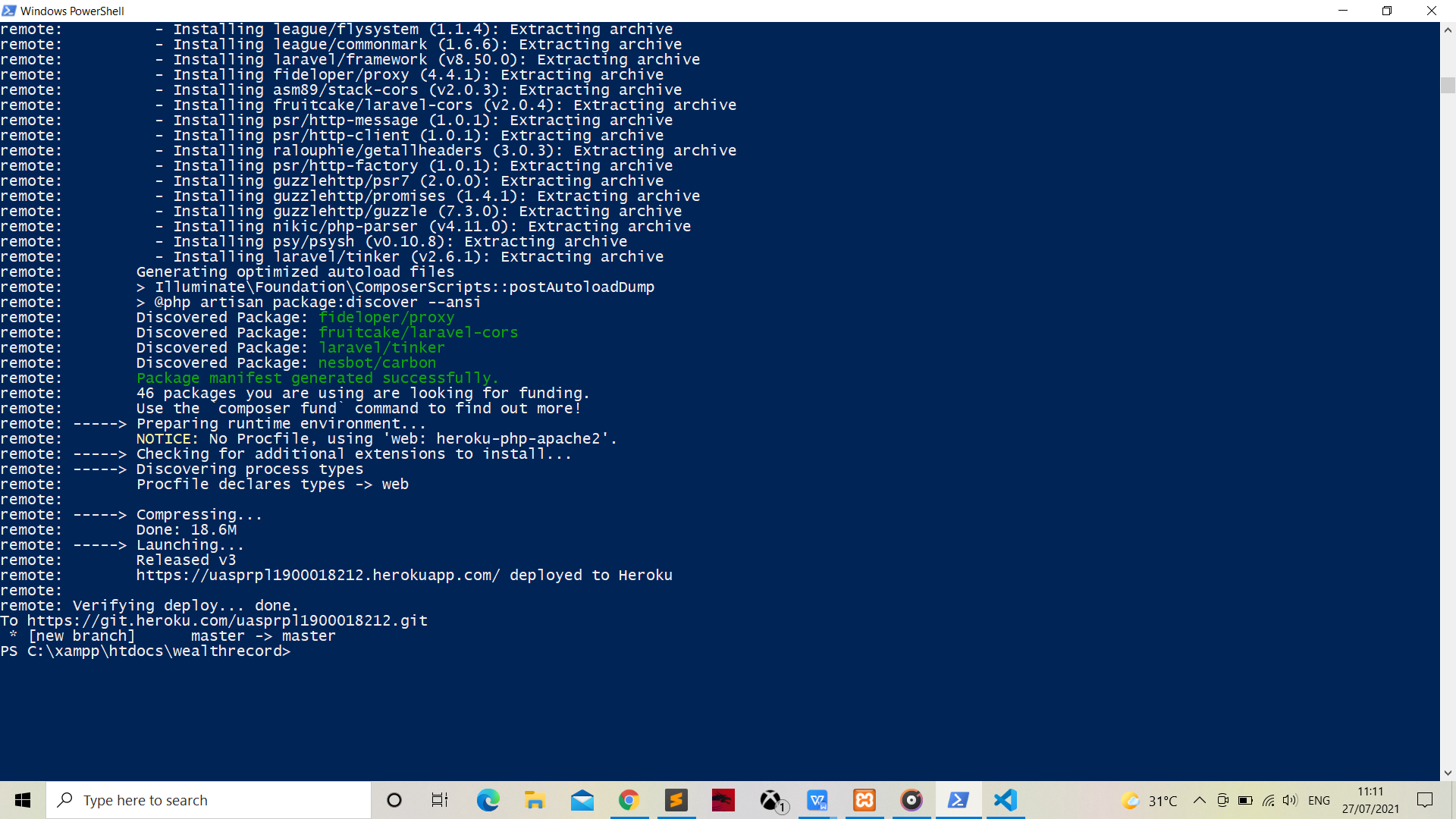
**Git/Github**



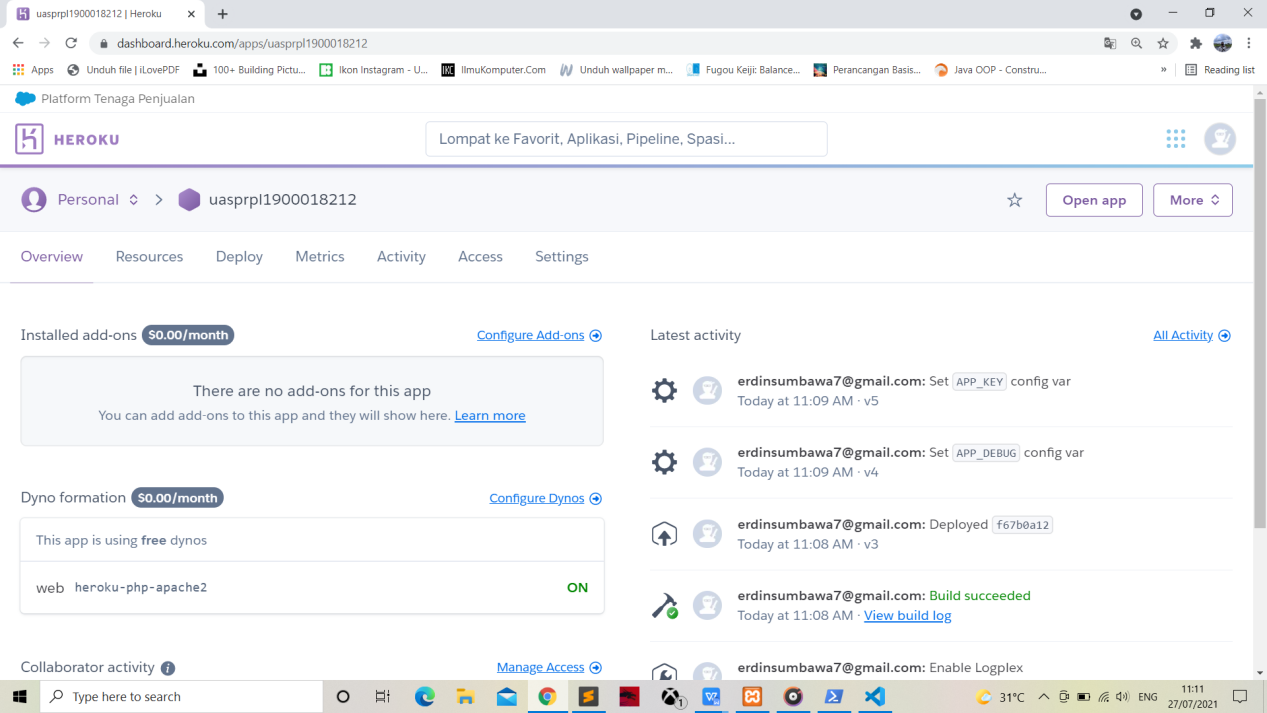


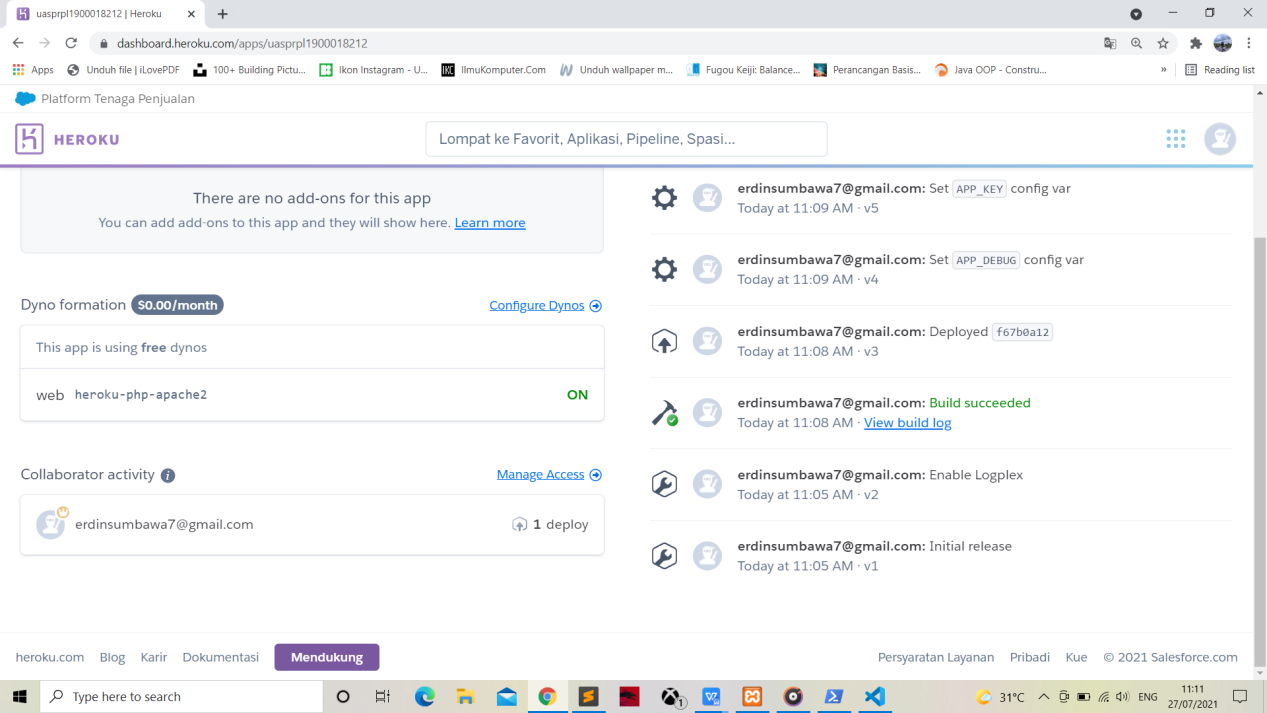




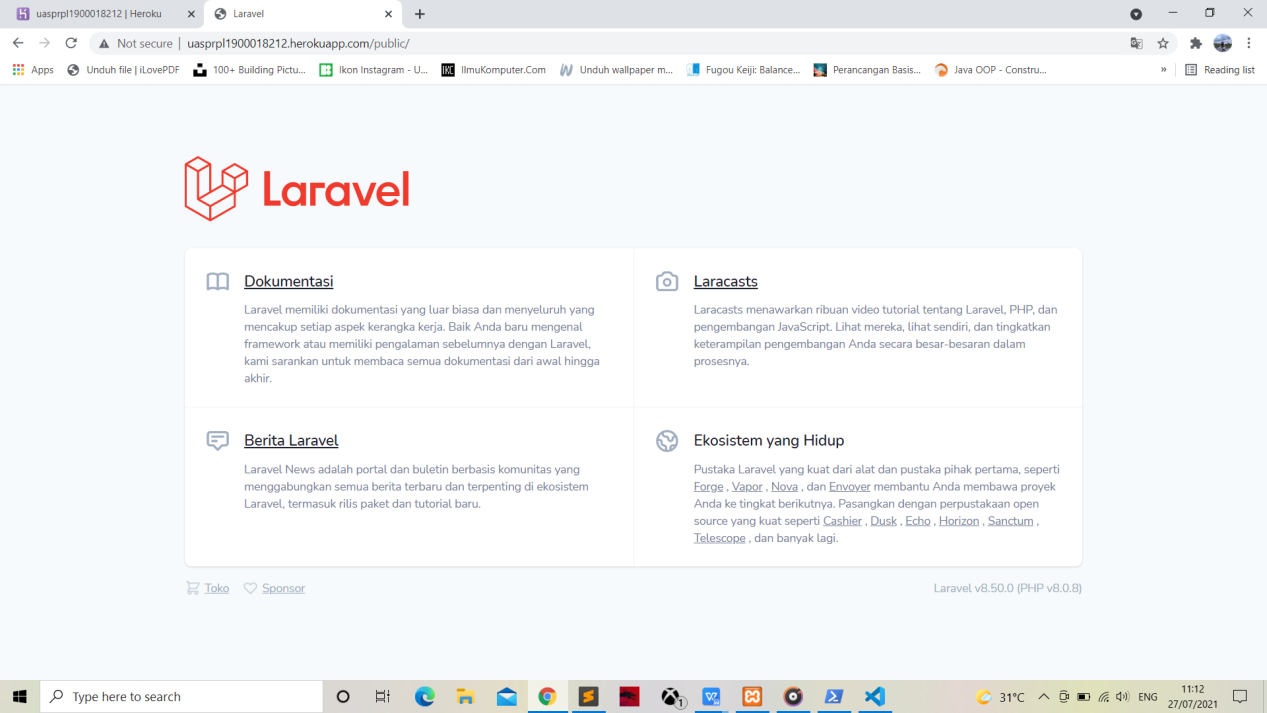


**HEROKU**





LARAVEL



http://uasprpl1900018212.herokuapp.com/public/