

Listing Aplikasi Mobile

C. CONTROLLER

```
<?php
```

```
class Sistem_mobile extends CI_Controller {
```

```
function __construct() {
```

```
    parent::__construct();
```

```
    header('Access-Control-Allow-Origin: *');
```

```
    header('Access-Control-Allow-Method: PUT, GET, POST, DELETE, OPTIONS');
```

```
    header('Access-Control-Allow-Headers: Content-Type, x-xsrf-token');
```

```
}
```

```
public function doLogin() {
```

```
    $postdata = file_get_contents("php://input");
```

```
    if ($postdata != null) {
```

```
        $request = json_decode($postdata);
```

```
        $username = $request->username;
```

```
        $password = $request->password;
```

```
        $alert = $request->alert;
```

```
        $this->login($username, $password, $alert);
```

```
    }
```

```
}
```

```
public function login($username, $password, $alert) {
```

```
    $member = $this->loginMember($username, $password);
```

```
    if ($member != null) {
```

```
        $status = $this->cekstatussignup($username);
```

```
        if ($status != null) {
```

```

    $status = 1;
} else {
    $status = 0;
}
$token = $member[0]['token'];
if ($token == 'ed94289f329664d608d2835c123c992a') {
    $now = date("Y-m-d h:i:s");
    $this->saveToken($username, $now);
    $json[] = array(
        'token' => sha1($username . $now),
        'status' => $status
    );
    echo json_encode($json);
} else {
    if ($alert == 0) {
        $json[] = array(
            'alert' => 'Akun anda sebelumnya telah digunakan pada perangkat lain. Jika anda login
pada perangkat ini, maka akun anda pada perangkat sebelumnya akan logout secara otomatis.
Apakah anda yakin ingin login di perangkat ini?'
        );
        echo json_encode($json);
    } else if ($alert == 1) {
        $now = date("Y-m-d h:i:s");
        $this->saveToken($username, $now);
        $json[] = array(
            'token' => sha1($username . $now), 'status' => $status
        );
        echo json_encode($json);
    }
}
} else {
    $json[] = array(

```

```

        'pesan' => 'Maaf kombinasi Username dan Password yang Anda masukkan Salah!',
    );
    echo json_encode($json);
}
}

```

```

public function dosignup() {
    $postdata = file_get_contents("php://input");
    if ($postdata != NULL) {
        $request = json_decode($postdata);
        $token = $request->npm;
        $b = $this->checkNPMuser($token);
        if ($b != NULL) {
            $json[] = array(
                'pesan' => 'NPM anda sudah digunakan pengguna lain ^^ ',
            );
            echo json_encode($json);
        } else {
            $id_jurusan = substr($token, 6, 1);
            $newpassword = $request->password;
            $data_user = array(
                'username' => $request->npm,
                'password' => $newpassword
            );
            $b = $this->saveuser($data_user);
            $data_peserta = array(
                'npm' => $request->npm,
                'namavoter' => $request->nama,
                'id_jurusan' => $id_jurusan
            );
            $a = $this->saveDaftar($data_peserta);

```

```

if ($a === true && $b === true) {
    $json[] = array(
        'sukses' => 'Pendaftaran Berhasil!'
    );
} else {
    $json[] = array(
        'pesan' => 'Maaf terjadi Kesalahan! Silahkan coba beberapa saat lagi!',
    );
}

echo json_encode($json);
}
}
}

```

```

public function loadData() {
    $postdata = file_get_contents("php://input");
    if ($postdata != NULL) {
        $request = json_decode($postdata);
        $token = $request->token;
        $b = $this->checkMember($token);
        if ($b == NULL) {
            $json[] = array(
                'log_out' => 'Maaf akun anda telah digunakan pada perangkat lain, Silahkan hubungi
admin jika anda tidak mengenali aktivitasnya ^^ ',
            );
            echo json_encode($json);
        } else {
            $id_jurusan = substr($b[0]['username'], 6, 1);
            $dataJurusan = $this->getdataJurusan($id_jurusan);
            $datavoter = $this->getdatavoter($b[0]['username']);

```

```

$namajurusan = $dataJurusan[0]['jurusan'];

$json[] = array(
    'jurusan' => $namajurusan,
    'id_jurusan' => $id_jurusan,
    'datavoter' => $datavoter
);

echo json_encode($json);
}
}
}

public function dogantipassword() {
    $postdata = file_get_contents("php://input");
    if ($postdata != NULL) {
        $request = json_decode($postdata);
        $token = $request->token;
        $b = $this->checkMember($token);
        if ($b == NULL) {
            $json[] = array(
                'log_out' => 'Maaf akun anda telah digunakan pada perangkat lain, Silahkan hubungi
admin jika anda tidak mengenali aktivitasnya ^^ ',
            );
            echo json_encode($json);
        } else {
            $oldPass = $request->password;
            $cekpassword = $this->cekpassworduser($oldPass, $token);
            if ($cekpassword === true) {
                $newPass = $request->newPass;
                $data_password['password'] = $newPass;
                $a = $this->update_password($data_password, $token);
            }
        }
    }
}

```

```
if ($a === true) {  
    $json[] = array(  
        'sukses' => 'Update Password Berhasil',  
    );  
    echo json_encode($json);  
} else {  
    $json[] = array(  
        'pesan' => 'Maaf terjadi kesalahan sistem ',  
    );  
    echo json_encode($json);  
}  
} else {  
    $json[] = array(  
        'pesan' => 'Maaf password lama yang anda masukkan salah ',  
    );  
    echo json_encode($json);  
}  
}  
}  
}
```

```
public function logOut() {  
    $postdata = file_get_contents("php://input");  
    if (isset($postdata)) {  
        $request = json_decode($postdata);  
        $token = $request->token;  
        $b = $this->checkMember($token);  
        if ($b != NULL) {  
            $nullToken = 'ed94289f329664d608d2835c123c992a';  
            $update = $this->log_out($token, $nullToken);  
        } else {
```

```
$json['log_out'] = 'Maaf akun anda telah digunakan diperangkat lain, Segera hubungi  
adoapo jika anda tidak mengenali aktivitas ini!';
```

```
    echo json_encode($json);  
  }  
}  
}
```

```
public function loadVote() {  
    $postdata = file_get_contents("php://input");  
    if ($postdata != NULL) {  
        $request = json_decode($postdata);  
        $token = $request->token;  
        $b = $this->checkMember($token);  
        if ($b == NULL) {  
            $json[] = array(  
                'log_out' => 'Maaf akun anda telah digunakan pada perangkat lain, Silahkan hubungi  
admin jika anda tidak mengenali aktivitasnya ^^ ',  
            );  
            echo json_encode($json);  
        } else {  
            $id_jurusan = $request->id_jurusan;  
            $kategori_calon = $request->kategori_calon;  
            $fungsi='get'.$kategori_calon;  
            $datacalon = $this->$fungsi($id_jurusan);  
            $json[] = array(  
                'data_calon' => $datacalon['data_calon'],  
                'jumlah_calon' => $datacalon['jumlah_calon'],  
                'id_jurusan' => $id_jurusan,  
                'kategori_calon' => strtolower($kategori_calon)  
            );  
            echo json_encode($json);  
        }  
    }  
}
```

```
    }  
  }  
}
```

```
public function dovote() {  
    $postdata = file_get_contents("php://input");  
    if ($postdata != NULL) {  
        $request = json_decode($postdata);  
        $token = $request->token;  
        $b = $this->checkMember($token);  
        if ($b == NULL) {  
            $json[] = array(  
                'log_out' => 'Maaf akun anda telah digunakan pada perangkat lain, Silahkan hubungi  
admin jika anda tidak mengenali aktivitasnya ^^ ',  
            );  
            echo json_encode($json);  
        } else {  
            $id_calon=$request->id_calon;  
            $jumlah_calon=$request->jumlah_calon;  
            $kategori_calon=$request->kategori_calon;  
            $pilihancalon=$request->pilihanvote;  
            $data_calon=$this->getdatacalon($id_calon, $kategori_calon);  
            $username=$b[0]['username'];  
            $id_jurusan = substr($b[0]['username'], 6, 1);  
            if ($jumlah_calon==1){  
                $updatedatacalon=$this->hitungsatucalon($data_calon, $kategori_calon, $pilihancalon);  
            }else{  
                $updatedatacalon=$this->hitungbanyakcalon($data_calon, $kategori_calon,  
$id_jurusan);  
            }  
            $updatevoter=$this->updatevoter($kategori_calon, $username);  
            if ($updatedatacalon===TRUE && $updatevoter===TRUE){
```



```

    $json[] = array(
        'sukses' => 'Voting Berhasil',
        'pilihan' => $pilihancalon
    );
    echo json_encode($json);
} else {
    $json[] = array(
        'pesan' => 'Maaf terjadi kesalahan sistem ',
    );
    echo json_encode($json);
}

}
}
}

private function hitungbanyakcalon ($datacalon, $kategori_calon, $id_jurusan) {
    $suara=$datacalon[0]['suaramasuk'];
    if ($suara==null){
        $suara=0;
    }
    $jumlahsuara=$suara+1;
    $data['suaramasuk']=$jumlahsuara;
    $savesuara=$this->savesuaracalon($data, $datacalon[0]['id'],$kategori_calon);
    $dataseluruhcalon=$this->getseluruhcalon($id_jurusan, $kategori_calon);
    $totalsuara=$this->gettotalsuara($id_jurusan, $kategori_calon);
    foreach ($dataseluruhcalon as $calon){
        $persentase=$calon['suaramasuk']/$totalsuara[0]['TotalSuara']*100;
        $data = null;
        $data['persentase']=$persentase;
        $savepersentase=$this->savesuaracalon($data, $calon['id'],$kategori_calon);
    }
}

```

```

}
if ($savesuara===true && $savepersentase===true){
return TRUE;
} else{
return FALSE;
}
}

```

```

private function hitungsatucalon ($datacalon, $kategori_calon, $pilihancalon) {
    $suara= unserialize($datacalon[0]['suaramasuk']);
    if ($suara==null){
        $suara=[0,0];
    }
    if ($pilihancalon=="setuju"){
        $suara[0]=$suara[0]+1;
    } else {
        $suara[1]=$suara[1]+1;
    }
    $jumlahsuara= serialize($suara);
    $data['suaramasuk']=$jumlahsuara;
    $savesuara=$this->savesuaracalon($data, $datacalon[0]['id'],$kategori_calon);
    $totalsuara=$suara[0]+$suara[1];
    $persentase=$suara[0]/$totalsuara*100;
    $data = null;
    $data['persentase']=$persentase;
    $savepersentase=$this->savesuaracalon($data, $datacalon[0]['id'],$kategori_calon);
    if ($savesuara===true && $savepersentase===true){
        return TRUE;
    } else{
        return FALSE;
    }
}

```

```
}
```

```
private function updatevoter($kategori_calon, $username){
```

```
    $npm=$username;
```

```
    $data['status_'].$kategori_calon]=1;
```

```
    $update=$this->updatestatusvoter($npm, $data);
```

```
    if ($update===true){
```

```
        return TRUE;
```

```
    }else{
```

```
        return FALSE;
```

```
    }
```

```
}
```

```
private function saveToken($username, $now) {
```

```
    $this->load->model('Model_user');
```

```
    return $this->Model_user->saveToken($username, $now);
```

```
}
```

```
private function loginMember($username, $password) {
```

```
    $this->load->model('Model_user');
```

```
    return $this->Model_user->login($username, $password);
```

```
}
```

```
private function cekstatussignup($username) {
```

```
    $this->load->model('Model_user');
```

```
    return $this->Model_user->statussignup($username);
```

```
}
```

```
private function saveDaftar($data_peserta) {
```

```
    $this->load->model('Model_user');
```

```
    return $this->Model_user->daftar($data_peserta);
```

```
}
```

```
private function checkMember($token) {  
    $this->load->model('Model_user');  
    return $this->Model_user->checkMember($token);  
}
```

```
private function checkNPMuser($token) {  
    $this->load->model('Model_user');  
    return $this->Model_user->checkNPMuser($token);  
}
```

```
private function update_password($data_password, $token) {  
    $this->load->model('Model_user');  
    return $this->Model_user->update_password($data_password, $token);  
}
```

```
private function getdataJurusan($id_jurusan) {  
    $this->load->model('Model_user');  
    return $this->Model_user->getdataJurusan($id_jurusan);  
}
```

```
private function getdatavoter($username) {  
    $this->load->model('Model_user');  
    return $this->Model_user->getdatavoter($username);  
}
```

```
private function log_out($token, $nullToken) {  
    $this->load->model('Model_user');  
    return $this->Model_user->log_out($token, $nullToken);  
}
```

```
private function saveuser($data_user) {
    $this->load->model('Model_user');
    return $this->Model_user->saveuser($data_user);
}

private function cekpassworduser($oldPass, $token) {
    $this->load->model('Model_user');
    return $this->Model_user->cekpassworduser($oldPass, $token);
}

private function getPresma($id_jurusan) {
    $this->load->model('Model_user');
    return $this->Model_user->getPresma($id_jurusan);
}

private function getGubma($id_jurusan) {
    $this->load->model('Model_user');
    return $this->Model_user->getGubma($id_jurusan);
}

private function getMPM($id_jurusan) {
    $this->load->model('Model_user');
    return $this->Model_user->getMPM($id_jurusan);
}

private function getdatacalon($id_calon, $kategori_calon) {
    $this->load->model('Model_user');
    return $this->Model_user->getdatacalon($id_calon, $kategori_calon);
}

private function savesuaracalon($data, $id_calon, $kategori_calon) {
```

```
$this->load->model('Model_user');  
return $this->Model_user->savesuaracalon($data, $id_calon, $kategori_calon);  
}  
private function getseluruhcalon($id_jurusan, $kategori_calon) {  
    $this->load->model('Model_user');  
    return $this->Model_user->getseluruhcalon($id_jurusan, $kategori_calon);  
}  
  
private function gettotalsuara($id_jurusan, $kategori_calon) {  
    $this->load->model('Model_user');  
    return $this->Model_user->gettotalsuara($id_jurusan, $kategori_calon);  
}  
private function updatestatusvoter($npm, $data) {  
    $this->load->model('Model_user');  
    return $this->Model_user->updatestatusvoter($npm, $data);  
}  
  
}
```


A. MODEL

```
<?php
```

```
if (!defined('BASEPATH'))
```

```
    exit('No direct script access allowed');
```

```
class Model_user extends CI_Model {
```

```
    private $table;
```

```
    function __construct() {
```

```
        parent::__construct();
```

```
    }
```

```
    private function get_table() {
```

```
        return $this->table = 'tb_user';
```

```
    }
```



```
function login($username, $password) {
    $table = $this->get_table();
    $this->db->select('token');
    $this->db->from($table);
    $this->db->where('username', $username);
    $this->db->where('password', $password);

    $query = $this->db->get();
    $result = $query->result_array();
    if ($result != null) {
        return $result;
    } else {
        return false;
    }
}

function statussignup($username) {
    $table = 'voter';
    $this->db->select('namavoter');
    $this->db->from($table);
    $this->db->where('npm', $username);

    $query = $this->db->get();
    $result = $query->result_array();
    if ($result != null) {
        return $result;
    } else {
        return false;
    }
}
```

```
function saveToken($username, $now) {  
    $data = array('token' => sha1($username . $now));  
    $this->db->where('username', $username);  
    $update = $this->db->update('tb_user', $data);  
    return $update;  
}
```

```
function update_password($data_password,$token ) {  
    $this->db->where('token', $token);  
    $update = $this->db->update('tb_user', $data_password);  
    return $update;  
}
```

```
function log_out($token, $nullToken) {  
    $data = array('token' => $nullToken);  
    $this->db->where('token', $token);  
    $update = $this->db->update('tb_user', $data);  
    return $update;  
}
```

```
function checkStatusMember($username) {  
    $table = $this->get_table();  
  
    $this->db->select('*');  
    $this->db->from($table);  
    $this->db->where('kode', $username);  
  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {  
        return $result;  
    }
```

```
    } else {  
        return NULL;  
    }  
}
```

```
function checkMember($token) {  
    $table = $this->get_table();  
  
    $this->db->select('username');  
    $this->db->from($table);  
    $this->db->where('token', $token);  
  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {  
        return $result;  
    } else {  
        return NULL;  
    }  
}
```

```
function checkNPMuser($token) {  
    $table = $this->get_table();  
  
    $this->db->select('username');  
    $this->db->from($table);  
    $this->db->where('username', $token);  
  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {
```

```
        return $result;
    } else {
        return NULL;
    }
}
```

```
function getPesma($id_jurusan) {
    $table = 'tb_pesma';
    $this->db->select('*');
    $this->db->from($table);
    $query = $this->db->get();
    $result = $query->result_array();
    $jumlah_calon = $query->num_rows();
    if ($result != null) {
        $data['data_calon'] = $result;
        $data['jumlah_calon'] = $jumlah_calon;
        return $data;
    } else {
        return NULL;
    }
}
```

```
function getGubma($id_jurusan) {
    $table = 'tb_gubma';
    $this->db->select('*');
    $this->db->from($table);
    $this->db->where('id_jurusan',$id_jurusan);
    $query = $this->db->get();
    $result = $query->result_array();
    $jumlah_calon = $query->num_rows();
```

```
if ($result != null) {  
    $data['data_calon'] = $result;  
    $data['jumlah_calon'] = $jumlah_calon;  
    return $data;  
} else {  
    return NULL;  
}  
}
```

```
function getMPM($id_jurusan) {  
    $table = 'tb_mpm';  
    $this->db->select('*');  
    $this->db->from($table);  
    $this->db->where('id_jurusan',$id_jurusan);  
    $query = $this->db->get();  
    $result = $query->result_array();  
    $jumlah_calon = $query->num_rows();  
    if ($result != null) {  
        $data['data_calon'] = $result;  
        $data['jumlah_calon'] = $jumlah_calon;  
        return $data;  
    } else {  
        return NULL;  
    }  
}
```

```
function getdatacalon($id_calon, $kategori_calon) {  
    $table = 'tb_'. $kategori_calon;  
    $this->db->select('*');  
    $this->db->from($table);  
    $this->db->where('id', $id_calon);
```

```

$query = $this->db->get();
$result = $query->result_array();
if ($result != null) {
    return $result;
} else {
    return NULL;
}
}

function savesuaracalon($data, $id_calon, $kategori_calon) {
    $this->db->where('id',$id_calon);
    $update = $this->db->update('tb_'.$kategori_calon, $data);
    $error=$this->db->error();
    if ($error['message'] != NULL){
        return FALSE;
    } else{
        return TRUE;
    }
}

function updatestatusvoter($npm, $data) {
    $this->db->where('npm',$npm);
    $update = $this->db->update('voter', $data);
    $error=$this->db->error();
    if ($error['message'] != NULL){
        return FALSE;
    } else{
        return TRUE;
    }
}
}

```

```
function cekpassworduser($oldPass, $token) {  
    $table = $this->get_table();  
  
    $this->db->select('username');  
    $this->db->from($table);  
    $this->db->where('token', $token);  
    $this->db->where('password', $oldPass);  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {  
        return TRUE;  
    } else {  
        return FALSE;  
    }  
}
```

```
function getDataMember($id) {  
    $table = $this->get_table();  
  
    $this->db->select('nama');  
    $this->db->select('email');  
    $this->db->select('tiket_kuis');  
    $this->db->from($table);  
    $this->db->where('id', $id);  
  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {  
        return $result;  
    } else {  
        return NULL;  
    }  
}
```

```
}  
}
```

```
function getseluruhcalon($id_jurusan, $kategori_calon) {  
    $table = 'tb_'.$kategori_calon;  
    $this->db->select('*');  
    $this->db->from($table);  
    if($kategori_calon != 'presma'){  
        $this->db->where('id_jurusan', $id_jurusan);  
    }  
    $query = $this->db->get();  
    $result = $query->result_array();  
    if ($result != null) {  
        return $result;  
    } else {  
        return NULL;  
    }  
}
```

```
function gettotalsuara($id_jurusan, $kategori_calon) {  
    $table = 'tb_'.$kategori_calon;  
    if ($kategori_calon != 'presma'){  
        $where="WHERE id_jurusan = $id_jurusan";  
    }else{  
        $where=null;  
    }  
    $query = $this->db->query("SELECT $table.id, SUM(suaramasuk) AS TotalSuara FROM $table  
$where");  
    $result = $query->result_array();  
    return $result;  
}
```



```

function getdataJurusan($id_jurusan) {
    $table = 'id_jurusan';

    $this->db->select('jurusan');
    $this->db->from($table);
    $this->db->where('id', $id_jurusan);
    $query = $this->db->get();
    $result = $query->result_array();
    if ($result != null) {
        return $result;
    } else {
        return NULL;
    }
}

```

```

function getdatavoter($username) {
    $table = 'voter';
    $this->db->select('namavoter');
    $this->db->select('status_presma');
    $this->db->select('status_gubma');
    $this->db->select('status_mpm');
    $this->db->from($table);
    $this->db->where('npm', $username);
    $query = $this->db->get();
    $result = $query->result_array();
    if ($result != null) {
        return $result;
    } else {
        return NULL;
    }
}

```

B. VIEW

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title>Autentifikasi <i>Login</i></ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<ion-list>
```

```
<p *ngIf="alert!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify; color:maroon; font-size:1.2rem">
```

```
<ion-icon color="danger" name="warning"></ion-icon> {{alert}}
```

```
</p>
```

```
<ion-item>
```

```
<ion-label floating>Username</ion-label>
```

```
<ion-input type="text" [(ngModel)]="username" maxLength="12"></ion-input>
```

```
</ion-item>
```

```
<p *ngIf="alertusername!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify; color:maroon; font-size:1.2rem; padding-left:2rem">
```

```
<ion-icon color="danger" name="warning"></ion-icon> {{alertusername}} </p>
```

```
<ion-item>
```

```
<ion-label floating>Password</ion-label>
```

```
<ion-input type="password" [(ngModel)]="password" maxLength="50"></ion-input>
```

```
</ion-item>
```

```
<p *ngIf="alertpassword!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify; color:maroon; font-size:1.2rem; padding-left:2rem">
```

```
<ion-icon color="danger" name="warning"></ion-icon> {{alertpassword}} </p>
```

```
</ion-list>
```

```
<button ion-button color="deepskyblue" clear (click)="login()">Log In</button>
```

```
<br>
```

```
<br>
```

```
<p ion-text> Belum punya akun?
```

```
<br> Silahkan daftar
```

```
<a (click)="signup()">disini!</a>
```

```
</p>
```

```
</ion-content>
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-angular';
```

```
import { ServiceProvider } from '../providers/service/service';
```

```
import { Storage } from '@ionic/storage';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-login',
```

```
  templateUrl: 'login.html',
```

```
})
```

```
export class LoginPage {
```

```
  username: any;
```

```
  password: any;
```

```
token: any;
alertusername: any;
alertpassword: any;
alert: any;
alertlogin: any = 0;
konfirmasilogin: any;
errorci: any;
newtoken: any;
pesan: any;
loginData: any;
status: any;
```

```
constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,
  private alertCtrl: AlertController, public loadingCtrl: LoadingController, private service:
  ServiceProvider) {

}
```

```
login() {
  this.alert = null;
  this.alertusername = null;
  this.alertpassword = null;
  if (this.username == "" || this.password == "" || this.username == null || this.password == null) {
    this.alert = 'Masukkan username dan password Anda';
  }
  else {
    let filter_username = /^[0-9]*$/; let filter_password = /^[a-z0-9]*$/;
    if (this.username && filter_username.test(this.username)) {
      if (this.password && filter_password.test(this.password)) {
        let alert = this.alertCtrl.create({
```

title: 'Konfirmasi Log In', message: 'Apakah anda yakin telah memasukkan username dan password dengan benar?'

```
    buttons: [{
      text: 'Tidak'
    }, {
      text: 'Ya',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.dologin();
        });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}
else {
  this.alertpassword = 'Password berupa angka dan huruf kecil';
}
} else {
  this.alertusername = 'Username berisi NPM berupa 12 angka';
}

}
}
dologin() {
  let loading = this.loadingCtrl.create({
    content: 'Log in...',
    spinner: 'bubbles',
```

```
duration: 1000000,
dismissOnPageChange: true
});
loading.present();
let data_login = JSON.stringify({
  username: this.username, password: this.password, alert: this.alertlogin
});
this.service.login(data_login).then(data => {
  this.loginData = data;
  console.log(this.loginData);
  if (this.loginData == 'notOnline') {
    let title = 'Error Connection';
    let message = 'Terjadi Kesalahan Koneksi';
    this.show_error(title, message);
  }
  else if (this.loginData == 'error') {
    let title = 'Error System';
    let message = 'Terjadi Kesalahan Sistem';
    this.show_error(title, message);
  }
  else {
    setTimeout(() => {
      this.respon();
    }, 1000);

    setTimeout(() => {
      loading.dismiss();
    }, 5000);
  }
});
}
```

```

show_error(title, message) {
  let alert = this.alertCtrl.create({
    title: title, message: message,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}

respon() {
  for (let data_login of this.loginData) {
    this.newtoken = data_login.token;
    this.pesan = data_login.pesan;
    this.konfirmasilogin = data_login.alert;
    this.errorrci = data_login.error;
    this.status = data_login.status;
  }

  if (this.newtoken) {
    this.storage.set('03710efd46cc30e2f6f3ab06b12f547bb0f1e474',
'da8df333f1244618ceb8f613bbb2236d41dfbd77');
    this.storage.set('token', this.newtoken);
    this.storage.get('token').then((data) => {
      this.token = data;
    });
  }
}

```

```

this.alertlogin = 0;
if (this.status == 1) {
  this.navCtrl.setRoot('HomePage');
}
else {
  this.navCtrl.setRoot('SignUpPage');
}
});

}

else if (this.konfirmasilogin) {
  let alert = this.alertCtrl.create({
    title: 'Information', message: this.konfirmasilogin,
    buttons: [{
      text: 'Tidak'
    }, {
      text: 'Ya',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.alertlogin = 1;
          this.dologin();
        });
        return false;
      }
    }
  ]),
  enableBackdropDismiss: false
});
alert.present();
}

else if (this.pesan) {

```



```
    let alert = this.alertCtrl.create({
      title: 'Peringatan', message: this.pesan
    });
    alert.present();
  }
  else if (this.errorci) {
    let alert = this.alertCtrl.create({
      title: 'Error System', message: this.errorci
    });
    alert.present();
  }
}

signup(){
  this.navCtrl.push('SignUpPage');
}

}
```

<ion-header>

<ion-navbar>

<ion-title><i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>

</ion-navbar>

</ion-header>

<ion-content>

```
<div [hidden]="hidden[0]" style="height: 100%; display: flex; align-items: center; justify-content: center">
<p style="text-align: center;">
<ion-spinner color="primary" name="bubbles" duration="1000"></ion-spinner>
<br>
<i style="color: grey">Loading</i>
</p>
</div>
```

```
<div [hidden]="hidden[1]" style="height: 100%; display: flex; align-items: center; justify-content: center">
<p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
    Maaf terjadi kesalahan koneksi
<br>silahkan coba beberapa saat lagi!
<br>
<button ion-button clear style="font-size: 3rem" (click)="refresh()">
<ion-icon name="md-refresh-circle"></ion-icon>
</button>
</p>
</div>
```

```
<div [hidden]="hidden[3]" style="height: 100%; display: flex; align-items: center; justify-content: center">
<p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
    {{pesan}}
</p>
</div>
```

```
<div [hidden]="hidden[2]">
<p *ngIf="dataVoter != null"
```

style="text-align: center" padding >Selamat Datang di Aplikasi <i>Electronic Voting</i> Pemilihan Umum Raya KM Politeknik Negeri Sriwijaya

voter atas nama {{dataVoter[0]['namavoter']}}

jurusan {{jurusan}}</p>

<ion-row>

<ion-col col-6 style="padding:0">

<button ion-button full color="tan" style="font-size: 24px; min-height:100px"

(click)="getVote()">V O T E</button>

</ion-col>

<ion-col col-6 style="padding:0">

<button ion-button full color="wheat" style="font-size: 20px; min-height: 100px"

(click)="getVoteResult()">VOTE
 RESULT</button>

</ion-col>

</ion-row>

<ion-row padding>

<button ion-button block outline icon-start

(click)="logOutAkun()"><ion-icon name="ios-power-outline"></ion-icon>Log Out</button>

</ion-row>

<p ion-text> Get tired with the code?

 Change your password

<a (click)="GantiPassword()">here

</p>

```
</div>
```

```
</ion-content>
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-angular';
```

```
import { ServiceProvider } from '../providers/service/service';
```

```
import { Storage } from '@ionic/storage';
```

```
import { SignUpPage } from '../sign-up/sign-up';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-home',
```

```
  templateUrl: 'home.html',
```

```
})
```

```
export class HomePage {
```

```
  test: any;
```

```
  a: any;
```

```
  b: any;
```

```
  c: any;
```

```
  d: any;
```

```
  e: any;
```

```
  f: any;
```

```
  id_jurusan: any;
```

```
  token: any;
```

```
  pesan: any;
```

```
  errorci: any;
```

```
  memberData: any;
```

```
  hidden: any;
```

```
  dataVoter: any;
```

```
jurusan: any;
dataLogOut: any;
```

```
constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,
  private alertCtrl: AlertController, public loadingCtrl: LoadingController, private service:
ServiceProvider) {
  this.hidden = [false, true, true, true];
  this.test = 'hai';
  this.a = 1234;
  this.b = ['hai', 'nama', 'saya'];
  this.c = { nama: 'Ditya' };
  this.d = this.c.nama;
  this.e = { nama: 'Manajemen Informatika' };
  this.f = this.e.nama;
  console.log(this.b);

  let npm = this.navParams.data.id_jurusan; console.log(npm);
  this.id_jurusan = '81';//npm.substring(6,7);
  console.log(this.id_jurusan);
}
```

```
ionViewWillLoad() {
  this.hidden = [false, true, true, true];
  this.storage.get('token').then((data) => {
    this.token = data;
    this.loadData();
  });
}
```

```
loadData() {
```

```

let data_member = JSON.stringify({
  token: this.token
});
this.service.load_Data(data_member).then(data => {
  this.memberData = data;
  console.log(this.memberData);
  if (this.memberData == 'notOnline') {
    this.hidden = [true, false, true, true];
  }
  else if (this.memberData == 'error') {
    this.hidden = [true, true, true, true];
    this.pesan = 'Terjadi Kesalahan Sistem';
  }
  else {
    setTimeout(() => {
      this.hidden = [true, true, false, true];
      this.respon();
    }, 1000);
  }
});
}

```

```

show_error(title, message) {
  let alert = this.alertCtrl.create({
    title: title, message: message,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {

```

```
});  
    return false;  
  }  
  },  
  enableBackdropDismiss: false  
});  
alert.present();  
}
```

```
logOutAkun() {  
  let loading = this.loadingCtrl.create({  
    spinner: 'circles',  
    content: 'Harap Tunggu Permintaan Anda Sedang Di Proses',  
    duration: 37500  
  });  
  loading.present();  
  let data_member = JSON.stringify({ token: this.token });  
  this.service.logOut(data_member).then(data => {  
    this.dataLogOut = data;  
    if (this.dataLogOut == "Unknown Error") {  
      let alert = this.alertCtrl.create({  
        title: 'Error Connection',  
        message: 'Maaf terjadi kesalahan koneksi, silahkan coba beberapa saat lagi!',  
        buttons: [{  
          text: 'Ok',  
          handler: () => {  
            let navTransition = alert.dismiss();  
  
            navTransition.then(() => {  
              loading.dismiss();  
            });  
          }  
        }  
      });  
    }  
  });  
}
```

```

        return false;
    }
}],
enableBackdropDismiss: false
});
alert.present();
} else {
setTimeout(() => {
    this.storage.remove('03710efd46cc30e2f6f3ab06b12f547bb0f1e474');
    this.storage.remove('token');
    this.navCtrl.setRoot('LoginPage');
}, 1000);

setTimeout(() => {
    loading.dismiss();
}, 5000);
}
});
}

```

```

respon() {
for (let data_member of this.memberData) {
    this.pesan = data_member.pesan;
    this.errorci = data_member.error;
    this.dataVoter = data_member.datavoter;
    if (this.dataVoter == null) {
        this.navCtrl.setRoot('SignUpPage');
    }
    this.jurusan = data_member.jurusan;

```



```
    this.id_jurusan = data_member.id_jurusan;
}
if (this.pesan) {
    this.hidden = [true, true, true, false];
    let alert = this.alertCtrl.create({
        title: 'Peringatan', message: this.pesan
    });
    alert.present();
}
else if (this.errorci) {
    this.hidden = [true, true, true, true];
    let alert = this.alertCtrl.create({
        title: 'Error System', message: this.errorci
    });
    alert.present();
}
}

getVote() {
    this.navCtrl.push('VotePage', { id_jurusan: this.id_jurusan, jurusan: this.jurusan,
dataVoter:this.dataVoter });
}

getVoteResult() {
    this.navCtrl.push('VoteResultPage', { id_jurusan: this.id_jurusan, jurusan: this.jurusan});
}

GantiPassword() {
    this.navCtrl.push('GantiPasswordPage');
}
}
```

```
}
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title><i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<ion-content padding>
```

```
<ion-list>
```

```
<p *ngIf="alertsSignup!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify; color:maroon; font-size:1.2rem">
```

```
<ion-icon color="danger" name="warning"></ion-icon> {{alertsSignup}}
```

```
</p>
```

```
<ion-item>
```

```
<ion-label floating>NPM</ion-label>
```

```
<ion-input type="number" [(ngModel)]= "npm" maxLength="12"></ion-input>
```

```
</ion-item>
```

```
<p *ngIf="alertNpm!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify; color:maroon; font-size:1.2rem; padding-left:2rem">
```

```
<ion-icon color="danger" name="warning"></ion-icon> {{alertNpm}} </p>
```

```
<ion-item>
```

```
<ion-label floating>Nama</ion-label>
<ion-input type="text" [(ngModel)]="nama" maxLength="50"></ion-input>
</ion-item>
```

```
<p *ngIf="alertnama!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify;
color:maroon; font-size:1.2rem; padding-left:2rem">
<ion-icon color="danger" name="warning"></ion-icon> {{alertnama}} </p>
```

```
<ion-item>
<ion-label floating>Password Baru</ion-label>
<ion-input type="password" [(ngModel)]="password" maxLength="50"></ion-input>
</ion-item>
```

```
<p *ngIf="alertpassword!=null" ion-text style="margin: 0px; padding: 5px; text-align: justify;
color:maroon; font-size:1.2rem; padding-left:2rem">
<ion-icon color="danger" name="warning" ></ion-icon> {{alertpassword}} </p>
```

```
</ion-list>
```

```
<button ion-button color="deepskyblue" clear (click)="signup()">Sign Up</button>
```

```
</ion-content>
```

```
</ion-content>
```

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-
angular';
import { ServiceProvider } from '../providers/service/service';
import { Storage } from '@ionic/storage';
```

```

@IonicPage()
@Component({
  selector: 'page-sign-up',
  templateUrl: 'sign-up.html',
})
export class SignUpPage {
  npm: any;
  nama: any;
  password: any;
  signupData: any;
  alertnpm: any;
  alertpassword: any;
  alertnama: any;
  alertsignup: any;
  pesan: any;
  errorci: any;
  sukses: any;

  constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,
    private alertCtrl: AlertController, public loadingCtrl: LoadingController, private service:
    ServiceProvider) {

  }

  signup() {
    this.alertsignup = null;
    this.alertnpm = null;
    this.alertnama = null;
    this.alertpassword = null;
  }

```

```

if (this.npm == "" || this.nama == "" || this.password == "" || this.npm == null || this.nama == null ||
this.password == null) {
  this.alertsignup = 'Masukkan data Anda';
}
else {
  let filter_npm = /^[0-9]*$/; let filter_password = /^[a-z0-9]*$/; let filter_nama = /^[a-zA-Z]*$/;
  if (this.npm && filter_npm.test(this.npm)) {
    if (this.nama && filter_nama.test(this.nama)) {
      if (this.password && filter_password.test(this.password)) {
        let alert = this.alertCtrl.create({
          title: 'Konfirmasi Sign Up', message: 'Apakah anda yakin telah memasukkan data dengan
benar?',
          buttons: [{
            text: 'Tidak'
          }, {
            text: 'Ya',
            handler: () => {
              let navTransition = alert.dismiss();
              navTransition.then(() => {
                this.dosignup();
              });
              return false;
            }
          }
        ]),
          enableBackdropDismiss: false
        });
        alert.present();
      } else {
        this.alertpassword = 'Password berupa angka dan huruf kecil';
      }
    } else {

```

```
        this.alertnama = 'Nama diisi dengan karakter A-Z atau a-z';
    }
} else {
    this.alertnpm = 'Username berisi NPM berupa 12 angka';
}

}
}
```

```
dosignup() {
    let loading = this.loadingCtrl.create({
        content: 'Sign Up...',
        spinner: 'bubbles',
        duration: 1000000,
        dismissOnPageChange: true
    });
    loading.present();
    let data_signup = JSON.stringify({
        npm: this.npm, nama: this.nama, password: this.password
    });
    this.service.signup(data_signup).then(data => {
        this.signupData = data;
        console.log(this.signupData);
        if (this.signupData == 'notOnline') {
            let title = 'Error Connection';
            let message = 'Terjadi Kesalahan Koneksi';
            this.show_error(title, message);
        }
        else if (this.signupData == 'error') {
            let title = 'Error System';
            let message = 'Terjadi Kesalahan Sistem';
```

```
    this.show_error(title, message);
  }
  else {
    setTimeout(() => {
      this.respon();
    }, 1000);

    setTimeout(() => {
      loading.dismiss();
    }, 2000);
  }
});
}
```

```
show_error(title, message) {
  let alert = this.alertCtrl.create({
    title: title, message: message,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
        });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}
respon() {
```

```
for (let data_signup of this.signupData) {
  this.pesan = data_signup.pesan;
  this.errorci = data_signup.error;
  this.sukses = data_signup.sukses;
}
if (this.sukses) {
  let alert = this.alertCtrl.create({
    title: 'informasi', message: this.sukses,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.navCtrl.setRoot('LoginPage');
        });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}
else if (this.pesan) {
  let alert = this.alertCtrl.create({
    title: 'Peringatan', message: this.pesan
  });
  alert.present();
}
else if (this.errorci) {
  let alert = this.alertCtrl.create({
```



```
    title: 'Error System', message: this.errorci
  });
  alert.present();
}
}
}
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title><i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div style="padding: 15px 0px; display: flex; align-items: center; justify-content: center">
```

```
<img style="height: 100px" src='assets/imgs/{{id_jurusan}}.jpg'>
```

```
</div>
```

```
<button ion-button full color="gray" style="font-size: 16px; min-height:100px"
(click)="getVoteCalon('Presma')"> PRESIDEN MAHASISWA </button>
```

```
<button ion-button full color="lightgray" style="font-size: 16px; min-
height:100px"(click)="getVoteCalon('Gubma')"> GUBERNUR MAHASISWA </button>
```

```
<button ion-button full color="gainsboro" style="font-size: 16x; min-
height:100px"(click)="getVoteCalon('MPM')"> ANGGOTA MPM </button>
```

```
</ion-content>
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, AlertController } from 'ionic-angular';
```

```
@IonicPage()
```

```

@Component({
  selector: 'page-vote',
  templateUrl: 'vote.html',
})
export class VotePage {
  test: any;
  id_jurusan: any;
  jurusan: any;
  dataVoter: any;

  constructor(public navCtrl: NavController, public navParams: NavParams, private alertCtrl:
AlertController) {

    this.id_jurusan = this.navParams.data.id_jurusan;
    this.jurusan = this.navParams.data.jurusan;
    this.dataVoter = this.navParams.data.dataVoter;
  }

  getVoteCalon(kategori_calon) {
    let str = kategori_calon;
    if (this.dataVoter[0]['status_' + str.toLowerCase()] == '0') {
      this.navCtrl.push('VoteCalonPage', { kategori_calon: kategori_calon, id_jurusan: this.id_jurusan,
jurusan: this.jurusan });
    }else{
      let alert = this.alertCtrl.create({
        title: 'Peringatan',
        message: 'Anda sudah melakukan voting',
        buttons: [{
          text: 'Ok',
          handler: () => {
            let navTransition = alert.dismiss();

```

```
        navTransition.then(() => {

            });

            return false;
        }
    }},
    enableBackdropDismiss: false
});
alert.present();
}
}

}

<ion-header>

<ion-navbar>
<ion-title><i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>
</ion-navbar>

</ion-header>

<ion-content>
<div [hidden]="hidden[0]" style="height: 100%; display: flex; align-items: center; justify-content: center">
<p style="text-align: center;">
<ion-spinner color="primary" name="bubbles" duration="1000"></ion-spinner>
<br>
<i style="color: grey">Loading</i>
```

```
</p>
</div>
```

```
<div [hidden]="hidden[1]" style="height: 100%; display: flex; align-items: center; justify-content: center">
```

```
<p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
```

```
  Maaf terjadi kesalahan koneksi
```

```
<br>silahkan coba beberapa saat lagi!
```

```
<br>
```

```
<button ion-button clear style="font-size: 3rem" (click)="refresh()">
```

```
<ion-icon name="md-refresh-circle"></ion-icon>
```

```
</button>
```

```
</p>
```

```
</div>
```

```
<div [hidden]="hidden[3]" style="height: 100%; display: flex; align-items: center; justify-content: center">
```

```
<p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
```

```
  {{pesan}}
```

```
</p>
```

```
</div>
```

```
<div [hidden]="hidden[2]">
```

```
<div style="padding: 15px 0px; display: flex; align-items: center; justify-content: center">
```

```
<img style="height: 100px" src='assets/imgs/{{id_jurusan}}.jpg'>
```

```
</div>
```

```
<div *ngIf="memberData != null">
```

```
<div *ngIf="memberData[0]['jumlah_calon'] > 1">
```

```
<div style="display: flex; flex-flow: row wrap;">
```

```

<ion-card *ngFor="let calon of memberData[0]['data_calon']; let i = index" style="width: 47%;
margin-bottom: 1%; margin-top: 1%; display: flex; align-items: center"
    [ngStyle]="{'margin-left': + ((i+1)%2)+1 + '%','margin-right': + (i%2)+1 + '%'}"
    (click)="Vote(calon)">
<div *ngIf="kategoricalon == 'mpm'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
    {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}{{i*1+1}}.jpg'>
<p style="margin: 10px; text-align: center; min-height: 3rem; font-size: 1.3rem;overflow : hidden;
text-overflow: ellipsis; display: flex; align-items: center;
    display: -webkit-box; -webkit-line-clamp: 2; -webkit-box-orient: vertical; font-weight:
bold">{{calon.nama_mpm}}</p>
</div>
<div *ngIf="kategoricalon == 'gubma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
    {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}{{i*1+1}}.jpg'>
<p style="margin: 10px; text-align: center; font-size: 1.3rem; font-weight:
bolder">{{calon.nama_cagub}}
<br>{{calon.nama_cawagub}}</p>
</div>
</ion-card>

```

```

</div>

```

```

</div>

```

```

<div *ngIf="memberData[0]['jumlah_calon'] == 1">
<ion-card *ngFor="let calon of memberData[0]['data_calon']; let i = index"
    style="width: 96%; margin: 2%;; display: flex; align-items: center">
<div *ngIf="kategoricalon == 'presma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
    {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/presma00.jpg'>

```

```

<p style="margin: 10px; text-align: center; font-size: 2rem; font-weight:
bolder">{{calon.nama_presma}}
<br>{{calon.nama_wapresma}}</p>
</div>
<div *ngIf="kategoricalon == 'gubma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
    {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}1.jpg'>
<p style="margin: 10px; text-align: center; font-size: 2rem; font-weight:
bolder">{{calon.nama_cagub}}
<br>{{calon.nama_cawagub}}</p>
</div>
</ion-card>
<ion-row>
<ion-col col-6>
<button ion-button block
(click)="SetujuVote(memberData[0]['data_calon'][0]['id'])">Setuju</button>
</ion-col>
<ion-col col-6>
<button ion-button block color="danger"
(click)="TidakSetujuVote(memberData[0]['data_calon'][0]['id'])">Tidak Setuju</button>
</ion-col>
</ion-row>
</div>

</div>
</div>
</ion-content>

```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-
angular';
```

```
import { ServiceProvider } from '../providers/service/service';
```

```
import { Storage } from '@ionic/storage';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-vote-calon',
```

```
  templateUrl: 'vote-calon.html',
```

```
})
```

```
export class VoteCalonPage {
```

```
  kategoricalon: any;
```

```
  token: any;
```

```
  hidden: any;
```

```
  memberData: any;
```

```
  pesan: any;
```

```
  errorci: any;
```

```
  id_jurusan: any;
```

```
  jurusan: any;
```

```
  voteData: any;
```

```
  id_calon: any;
```

```
  sukses: any;
```

```
  pilihanvote: any;
```

```
  constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,  
    private alertController: AlertController, public loadingCtrl: LoadingController, private service:  
    ServiceProvider) {
```

```
    this.kategoricalon = this.navParams.data.kategori_calon;
```

```
    this.id_jurusan = this.navParams.data.id_jurusan;
```

```
    this.jurusan = this.navParams.data.jurusan;
```

```
    this.hidden = [false, true, true, true];
```

```
  }
```

```
ionViewWillLoad() {  
  this.hidden = [false, true, true, true];  
  this.storage.get('token').then((data) => {  
    this.token = data;  
    this.loadData();  
  });  
}
```

```
loadData() {  
  
  let data_member = JSON.stringify({  
    token: this.token, id_jurusan: this.id_jurusan, kategori_calon: this.kategoricalon  
  });  
  this.service.loadVote(data_member).then(data => {  
    this.memberData = data;  
    if (this.memberData == 'notOnline') {  
      this.hidden = [true, false, true, true];  
    }  
    else if (this.memberData == 'error') {  
      this.hidden = [true, true, true, true];  
      this.pesan = 'Terjadi Kesalahan Sistem';  
    }  
    else {  
      setTimeout(() => {  
        this.id_jurusan = this.memberData[0]['id_jurusan'];  
        this.kategoricalon = this.memberData[0]['kategori_calon'];  
        this.hidden = [true, true, false, true];  
        for (let data_member of this.memberData) {  
          this.pesan = data_member.pesan;  
          this.errorci = data_member.error;
```



```
    this.jurusan = data_member.jurusan;
    this.id_jurusan = data_member.id_jurusan;
    this.sukses = data_member.sukses;
  }
  this.respon();
}, 1000);
}
});
}
```

```
show_error(title, message) {
  let alert = this.alertCtrl.create({
    title: title, message: message,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}
```

```
respon() {
  if (this.sukses) {
    let alert = this.alertCtrl.create({
      title: 'informasi', message: this.sukses,
```

```
buttons: [{
  text: 'OK',
  handler: () => {
    let navTransition = alert.dismiss();
    navTransition.then(() => {
      this.navCtrl.setRoot('HomePage');
    });
    return false;
  }
}],
enableBackdropDismiss: false
});
alert.present();

}

else if (this.pesan) {
  this.hidden = [true, true, true, false];
  let alert = this.alertCtrl.create({
    title: 'Peringatan', message: this.pesan
  });
  alert.present();
}

else if (this.errorci) {
  this.hidden = [true, true, true, true];
  let alert = this.alertCtrl.create({
    title: 'Error System', message: this.errorci
  });
  alert.present();
}
}
```

```
Vote(calon) {  
  let alert = this.alertCtrl.create({  
    title: "Konfirmasi Pemilihan", message: "Apakah Anda Yakin?",  
    buttons: [{  
      text: 'OK',  
      handler: () => {  
        let navTransition = alert.dismiss();  
        navTransition.then(() => {  
          this.hidden = [false, true, true, true];  
          this.id_calon = calon.id;  
          this.pilihanvote = 'setuju';  
          this.doVote();  
        });  
        return false;  
      }  
    }],  
    enableBackdropDismiss: false  
  });  
  alert.present();  
}
```

```
SetujuVote(id_calon){  
  let alert = this.alertCtrl.create({  
    title: "Konfirmasi Pemilihan", message: "Apakah Anda Yakin?",  
    buttons: [{  
      text: 'OK',  
      handler: () => {  
        let navTransition = alert.dismiss();  
        navTransition.then(() => {  
          this.hidden = [false, true, true, true];  
          this.id_calon = id_calon;  
        });  
      }  
    }],  
    enableBackdropDismiss: false  
  });  
  alert.present();  
}
```

```
    this.pilihanvote = 'setuju';
    this.doVote();
  });
  return false;
}
}},
enableBackdropDismiss: false
});
alert.present();
}
```

```
TidakSetujuVote(id_calon){
  let alert = this.alertCtrl.create({
    title: "Konfirmasi Pemilihan", message: "Apakah Anda Yakin?",
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.hidden = [false, true, true, true];
          this.id_calon = id_calon;
          this.pilihanvote = 'tidak_setuju';
          this.doVote();
        });
        return false;
      }
    }
  ]},
  enableBackdropDismiss: false
});
alert.present();
}
```

```
doVote() {  
  let loading = this.loadingCtrl.create({  
    content: 'Voting...',  
    spinner: 'bubbles',  
    duration: 1000000,  
    dismissOnPageChange: true  
  });  
  loading.present();  
  let data_vote = JSON.stringify({  
    token: this.token, id_calon: this.id_calon, jumlah_calon: this.memberData[0]['jumlah_calon'],  
    kategori_calon: this.kategoricalon, pilihanvote: this.pilihanvote  
  });  
  console.log(data_vote);  
  this.service.vote(data_vote).then(data => {  
    this.voteData = data;  
    console.log(this.voteData);  
    if (this.voteData == 'notOnline') {  
      let title = 'Error Connection';  
      let message = 'Terjadi Kesalahan Koneksi';  
      this.show_error(title, message);  
    }  
    else if (this.voteData == 'error') {  
      let title = 'Error System';  
      let message = 'Terjadi Kesalahan Sistem';  
      this.show_error(title, message);  
    }  
    else {  
      setTimeout(() => {  
        for (let data_member of this.voteData) {  
          this.pesan = data_member.pesan;  
        }  
      }, 1000);  
    }  
  });  
}
```

```
        this.errorci = data_member.error;
        this.jurusan = data_member.jurusan;
        this.id_jurusan = data_member.id_jurusan;
        this.sukses = data_member.sukses;
    }
    this.respon();
}, 1000);

setTimeout(() => {
    loading.dismiss();
}, 2000);
}
});
}
}
}
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title><i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content padding>
```

```
<div style="padding: 15px 0px; display: flex; align-items: center; justify-content: center">
```

```
<img style="height: 100px" src='assets/imgs/{{id_jurusan}}.jpg'>
```

```
</div>
```

```

<button ion-button full color="gray" style="font-size: 16px; min-height:100px"
(click)="getresultcalon('Presma')"> VOTE RESULT <br> PRESIDEN MAHASISWA </button>

<button ion-button full color="lightgray" style="font-size: 16px; min-height:100px"
(click)="getresultcalon('Gubma')"> VOTE RESULT <br>GUBERNUR MAHASISWA </button>

<button ion-button full color="gainsboro" style="font-size: 16x; min-height:100px"
(click)="getresultcalon('MPM')"> VOTE RESULT <br>ANGGOTA MPM </button>

</ion-content>

```

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

```

```
@IonicPage()
```

```
@Component({
  selector: 'page-vote-result',
  templateUrl: 'vote-result.html',
})
```

```
export class VoteResultPage {
```

```
  test: any;
```

```
  id_jurusan: any;
```

```
  jurusan: any;
```

```
  constructor(public navCtrl: NavController, public NavParams: NavParams) {
```

```
    this.id_jurusan = this.NavParams.data.id_jurusan;
```

```
    this.jurusan = this.NavParams.data.jurusan;
```

```
  }
```

```
  getresultcalon(kategori_calon) {
```

```
    this.navCtrl.push('ResultcalonPage', { kategori_calon: kategori_calon, id_jurusan: this.id_jurusan,
jurusan: this.jurusan });
```

```
}  
}
```

```
<ion-header>
```

```
<ion-navbar>
```

```
<ion-title>
```

```
<i>E-Voting</i> Pemilu Raya KM Polsri</ion-title>
```

```
</ion-navbar>
```

```
</ion-header>
```

```
<ion-content>
```

```
<div [hidden]="hidden[0]" style="height: 100%; display: flex; align-items: center; justify-content: center">
```

```
<p style="text-align: center;">
```

```
<ion-spinner color="primary" name="bubbles" duration="1000"></ion-spinner>
```

```
<br>
```

```
<i style="color: grey">Loading</i>
```

```
</p>
```

```
</div>
```

```
<div [hidden]="hidden[1]" style="height: 100%; display: flex; align-items: center; justify-content: center">
```

```
<p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
```

```
    Maaf terjadi kesalahan koneksi
```

```
<br>silahkan coba beberapa saat lagi!
```

```
<br>
```

```
<button ion-button clear style="font-size: 3rem" (click)="refresh()">
```

```
<ion-icon name="md-refresh-circle"></ion-icon>
```

```
</button>
```



```
</p>
</div>
```

```
<div [hidden]="hidden[3]" style="height: 100%; display: flex; align-items: center; justify-content: center">
  <p style="width: 100%; text-align: center; color: grey; font-size: 1.3rem;">
    {{pesan}}
  </p>
</div>
```

```
<div [hidden]="hidden[2]">
  <div style="padding: 15px 0px; display: flex; align-items: center; justify-content: center">
    <img style="height: 100px" src='assets/imgs/{{id_jurusan}}.jpg'>
  </div>
```

```
<div *ngIf="memberData != null">
```

```
<div *ngIf="memberData[0]['jumlah_calon'] > 1">
```

```
<div style="display: flex; flex-flow: row wrap;">
```

```
<ion-card *ngFor="let calon of memberData[0]['data_calon']; let i = index" style="width: 47%; margin-bottom: 1%; margin-top: 1%; display: flex; align-items: center"
```

```
  [ngStyle]="{'margin-left': + ((i+1)%2)+1 + '%', 'margin-right': + (i%2)+1 + '%'}">
```

```
<div *ngIf="kategoricalon == 'mpm'">
```

```
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
```

```
  {{i*1+1}}</p>
```

```
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}{{i*1+1}}.jpg'>
```

```
<p style="margin: 10px; text-align: center; min-height: 3rem; font-size: 1.3rem; overflow : hidden; text-overflow: ellipsis; display: flex; align-items: center;
```

```
  display: -webkit-box; -webkit-line-clamp: 2; -webkit-box-orient: vertical; font-weight: bold">{{calon.nama_mpm}}</p>
```

```
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder; color:blue">
```

```

        {{calon.persentase}}%</p>
<br>
</div>
<div *ngIf="kategoricalon == 'gubma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
        {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}{{i*1+1}}.jpg'>
<p style="margin: 10px; text-align: center; font-size: 1.3rem; font-weight:
bolder">{{calon.nama_cagub}}
<br>{{calon.nama_cawagub}}</p>
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder; color:blue">
        {{calon.persentase}}%</p>
<br>
</div>
</ion-card>

</div>
</div>

<div *ngIf="memberData[0]['jumlah_calon'] == 1">
<ion-card *ngFor="let calon of memberData[0]['data_calon']; let i = index" style="width: 96%;
margin: 2%;; display: flex; align-items: center">
<div *ngIf="kategoricalon == 'presma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
        {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/presma00.jpg'>
<p style="margin: 10px; text-align: center; font-size: 2rem; font-weight:
bolder">{{calon.nama_presma}}
<br>{{calon.nama_wapresma}}</p>
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder; color:blue">
        {{calon.persentase}}%</p>
<br>

```

```

</div>
<div *ngIf="kategoricalon == 'gubma'">
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder;">
    {{i*1+1}}</p>
<img style="padding: 0px 5px" src='assets/imgs/{{kategoricalon}}{{id_jurusan}}1.jpg'>
<p style="margin: 10px; text-align: center; font-size: 2rem; font-weight:
bolder">{{calon.nama_cagub}}
<br>{{calon.nama_cawagub}}</p>
<p style="margin-top: 5px; text-align: center; font-size: 2rem; font-weight: bolder; color:blue">
    {{calon.persentase}}%</p>
<br>
</div>
</ion-card>
</div>

```

```

</div>
</div>
</ion-content>

```

```

import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-
angular';
import { ServiceProvider } from '../providers/service/service';
import { Storage } from '@ionic/storage';

```

```

/**
 * Generated class for the ResultcalonPage page.
 *
 * See https://ionicframework.com/docs/components/#navigation for more info on
 * Ionic pages and navigation.
 */

```

```

@IonicPage()
@Component({
  selector: 'page-resultcalon',
  templateUrl: 'resultcalon.html',
})
export class ResultcalonPage {
  kategoricalon: any;
  token: any;
  hidden: any;
  memberData: any;
  pesan: any;
  errorci: any;
  id_jurusan: any;
  jurusan: any;
  voteData: any;
  id_calon: any;
  sukses: any;
  pilihanvote: any;

  constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,
    private alertCtrl: AlertController, public loadingCtrl: LoadingController, private service:
    ServiceProvider) {
    this.kategoricalon = this.navParams.data.kategori_calon;
    this.id_jurusan = this.navParams.data.id_jurusan;
    this.jurusan = this.navParams.data.jurusan;
    this.hidden = [false, true, true, true];
  }

  ionViewWillLoad() {
    this.hidden = [false, true, true, true];
  }

```

```
this.storage.get('token').then((data) => {  
  this.token = data;  
  this.loadData();  
});  
}
```

```
loadData() {
```

```
  let data_member = JSON.stringify({  
    token: this.token, id_jurusan: this.id_jurusan, kategori_calon: this.kategoricalon  
  });
```

```
  this.service.loadVote(data_member).then(data => {
```

```
    this.memberData = data;
```

```
    if (this.memberData == 'notOnline') {
```

```
      this.hidden = [true, false, true, true];
```

```
    }
```

```
    else if (this.memberData == 'error') {
```

```
      this.hidden = [true, true, true, true];
```

```
      this.pesan = 'Terjadi Kesalahan Sistem';
```

```
    }
```

```
    else {
```

```
      setTimeout(() => {
```

```
        this.id_jurusan = this.memberData[0]['id_jurusan'];
```

```
        this.kategoricalon = this.memberData[0]['kategori_calon'];
```

```
        this.hidden = [true, true, false, true];
```

```
        for (let data_member of this.memberData) {
```

```
          this.pesan = data_member.pesan;
```

```
          this.errorci = data_member.error;
```

```
          this.jurusan = data_member.jurusan;
```

```
          this.id_jurusan = data_member.id_jurusan;
```

```
          this.sukses = data_member.sukses;
```

```
    }  
    this.respon();  
  }, 1000);  
}  
});  
}
```

```
show_error(title, message) {  
  let alert = this.alertCtrl.create({  
    title: title, message: message,  
    buttons: [{  
      text: 'OK',  
      handler: () => {  
        let navTransition = alert.dismiss();  
        navTransition.then(() => {  
          });  
        return false;  
      }  
    }],  
    enableBackdropDismiss: false  
  });  
  alert.present();  
}
```

```
respon() {  
  if (this.sukses) {  
    let alert = this.alertCtrl.create({  
      title: 'informasi', message: this.sukses,  
      buttons: [{  
        text: 'OK',  
        handler: () => {
```

```

    let navTransition = alert.dismiss();
    navTransition.then(() => {
        this.navCtrl.setRoot('HomePage');
    });
    return false;
}
}},
enableBackdropDismiss: false
});
alert.present();

}
else if (this.pesan) {
    this.hidden = [true, true, true, false];
    let alert = this.alertCtrl.create({
        title: 'Peringatan', message: this.pesan
    });
    alert.present();
}
else if (this.errorci) {
    this.hidden = [true, true, true, true];
    let alert = this.alertCtrl.create({
        title: 'Error System', message: this.errorci
    });
    alert.present();
}
}
}
}

```

<p style="text-align: center; margin: 30px 0px 10px; background: white">

</p>

<ion-list no-lines style="margin-bottom: 0px;">

<p ion-text *ngIf="alert[0] != null" style="margin: 0px; padding: 5px; text-align: justify; color: red; font-size: 1.2rem">

<ion-icon color="danger" name="warning"></ion-icon> {{alert[0]}}

</p>

<ion-item>

<ion-label>

<ion-icon name="ios-key"></ion-icon>

</ion-label>

<ion-input type="password" placeholder="Password Lama" value="" [(ngModel)]="oldPass"></ion-input>

</ion-item>

<ion-item style="margin-top: 2rem;">

<ion-label>

<ion-icon name="ios-lock"></ion-icon>

</ion-label>

<ion-input type="password" placeholder="Password Baru" value="" [(ngModel)]="newPass"></ion-input>

</ion-item>

<ion-item style="margin-top: 2rem;">

<ion-label>

<ion-icon name="md-key"></ion-icon>

</ion-label>

<ion-input type="password" placeholder="Konfirmasi Password Baru" value="" [(ngModel)]="confirmPass"></ion-input>

</ion-item>

<p ion-text *ngIf="alert[1] != null" style="margin: 0px; padding: 5px; text-align: justify; color: red; font-size: 1.2rem; padding-left: 2rem">


```
<ion-icon color="danger" name="warning"></ion-icon> {{alert[1]}}
```

```
</p>
```

```
<br>
```

```
</ion-list>
```

```
<button ion-button block (click)="updatePassword()">
```

```
  Update
```

```
</button>
```

```
<ion-fab bottom left>
```

```
<button ion-fab navPop>
```

```
<ion-icon name="md-arrow-back"></ion-icon>
```

```
</button>
```

```
</ion-fab>
```

```
import { Component } from '@angular/core';
```

```
import { IonicPage, NavController, NavParams, LoadingController, AlertController } from 'ionic-angular';
```

```
import { ServiceProvider } from '../providers/service/service';
```

```
import { Storage } from '@ionic/storage';
```

```
@IonicPage()
```

```
@Component({
```

```
  selector: 'page-ganti-password',
```

```
  templateUrl: 'ganti-password.html',
```

```
})
```

```
export class GantiPasswordPage {
```

```
  alert:any;
```

```
  oldPass:any;
```

```

newPass:any;
confirmPass:any;
token: any;
gantipasswordData:any;
sukses:any;
errorci:any;
pesan:any;
dataLogOut:any;
constructor(public navCtrl: NavController, public navParams: NavParams, private storage: Storage,
  private alertCtrl: AlertController, public loadingCtrl: LoadingController, private service:
ServiceProvider) {
  this.alert=[null,null];
  this.storage.get('token').then((data) => {
    this.token = data;
  });
}

updatePassword() {
  this.alert=[null,null];

  if (this.oldPass == "" || this.newPass == "" || this.confirmPass == "" || this.oldPass == null ||
this.newPass == null || this.confirmPass == null) {
    this.alert[0] = 'Masukkan data Anda';
  }
  else {
    let filter_password = /^[a-z0-9]*$/;

    if (this.oldPass && filter_password.test(this.oldPass) && this.newPass &&
filter_password.test(this.newPass) && this.confirmPass && filter_password.test(this.confirmPass)) {
      if (this.newPass==this.confirmPass) {
        let alert = this.alertCtrl.create({
          title: 'Konfirmasi', message: 'Apakah anda yakin telah memasukkan data dengan benar?',
          buttons: [{
            text: 'Tidak'

```

```

    }, {
      text: 'Ya',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.dogantipassword();
        });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
} else {
  this.alert[1] = 'Password baru tidak sama dengan konfirmasi password';
}
} else {
  this.alert[0] = 'Password berupa angka dan huruf kecil';
}
}
}

```

```

dogantipassword() {
  let loading = this.loadingCtrl.create({
    content: 'Sign Up...',
    spinner: 'bubbles',
    duration: 1000000,
    dismissOnPageChange: true
  });
  loading.present();
}

```

```

let data_gantipassword = JSON.stringify({
  password: this.oldPass, newPass:this.newPass, token: this.token
});
this.service.gantipassword(data_gantipassword).then(data => {
  this.gantipasswordData = data;
  console.log(this.gantipasswordData);
  if (this.gantipasswordData == 'notOnline') {
    let title = 'Error Connection';
    let message = 'Terjadi Kesalahan Koneksi';
    this.show_error(title, message);
  }
  else if (this.gantipasswordData == 'error') {
    let title = 'Error System';
    let message = 'Terjadi Kesalahan Sistem';
    this.show_error(title, message);
  }
  else {
    setTimeout(() => {
      this.respon();
    }, 1000);

    setTimeout(() => {
      loading.dismiss();
    }, 2000);
  }
});
}

```

```

show_error(title, message) {
  let alert = this.alertCtrl.create({
    title: title, message: message,

```

```
buttons: [{
  text: 'OK',
  handler: () => {
    let navTransition = alert.dismiss();
    navTransition.then(() => {
    });
    return false;
  }
}],
enableBackdropDismiss: false
});
alert.present();
}
```

```
logOutAkun() {
  let loading = this.loadingCtrl.create({
    spinner: 'circles',
    content: 'Harap Tunggu Permintaan Anda Sedang Di Proses',
    duration: 37500
  });
  loading.present();
  let data_member = JSON.stringify({ token: this.token});
  this.service.logOut(data_member).then(data => {
    this.dataLogOut = data;
    if (this.dataLogOut == "Unknown Error") {
      let alert = this.alertCtrl.create({
        title: 'Error Connection',
        message: 'Maaf terjadi kesalahan koneksi, silahkan coba beberapa saat lagi!',
        buttons: [{
          text: 'Ok',
          handler: () => {
```

```

    let navTransition = alert.dismiss();

    navTransition.then(() => {
      loading.dismiss();
    });

    return false;
  }
  }},
  enableBackdropDismiss: false
});
alert.present();
}else{
  setTimeout(() => {
    this.storage.remove('03710efd46cc30e2f6f3ab06b12f547bb0f1e474');
    this.storage.remove('token');
    this.navCtrl.setRoot('LoginPage');
  }, 1000);

  setTimeout(() => {
    loading.dismiss();
  }, 5000);
}
});
}

respon() {
  for (let data_gantipassword of this.gantipasswordData) {
    this.pesan = data_gantipassword.pesan;
    this.errorci = data_gantipassword.error;
  }
}

```

```
this.sukses = data_gantipassword.sukses;
}
if (this.sukses) {
  let alert = this.alertCtrl.create({
    title: 'informasi', message: this.sukses,
    buttons: [{
      text: 'OK',
      handler: () => {
        let navTransition = alert.dismiss();
        navTransition.then(() => {
          this.logOutAkun();
        });
        return false;
      }
    }],
    enableBackdropDismiss: false
  });
  alert.present();
}
else if (this.pesan) {
  let alert = this.alertCtrl.create({
    title: 'Peringatan', message: this.pesan
  });
  alert.present();
}
else if (this.errorci) {
  let alert = this.alertCtrl.create({
    title: 'Error System', message: this.errorci
  });
  alert.present();
}
```

```
}  
}  
}
```

```
import { HttpClient } from '@angular/common/http';  
import { Injectable } from '@angular/core';
```

```
@Injectable()
```

```
export class ServiceProvider {
```

```
data: any;
```

```
private base_url="http://localhost/cismigel/index.php/api/sistem_mobile/";
```

```
constructor(public http: HttpClient) {
```

```
  this.data=null;
```

```
}
```

```
login(data_login){
```

```
  let link=this.base_url+'doLogin';
```

```
  return new Promise(resolve => {
```

```
    this.http.post(link, data_login).subscribe(data => {
```

```
      this.data = data;
```

```
      resolve(this.data);
```

```
    }, (err) => {
```

```
      if (err.statusText == "Unknown Error"){
```

```
        this.data = 'notOnline';
```

```
      } else {
```

```
        this.data = err;//'error';
```

```
      }
```

```
      resolve(this.data);
```

```
    });
```

```
  });
```



```
}
```

```
signup(data_signup){  
  let link=this.base_url+'dosignup';  
  return new Promise(resolve => {  
    this.http.post(link, data_signup).subscribe(data => {  
      this.data = data;  
      resolve(this.data);  
    }, (err) => {  
      if (err.statusText == "Unknown Error"){  
        this.data = 'notOnline';  
      } else {  
        this.data = err; //'error';  
      }  
      resolve(this.data);  
    });  
  });  
}
```

```
load_Data(data_member){  
  let link=this.base_url+'loadData';  
  return new Promise(resolve => {  
    this.http.post(link, data_member).subscribe(data => {  
      this.data = data;  
      resolve(this.data);  
    }, (err) => {  
      if (err.statusText == "Unknown Error"){  
        this.data = 'notOnline';  
      } else {  
        this.data = err; //'error';  
      }  
    })  
  });  
}
```

```
    resolve(this.data);
  });
});
}
```

```
logOut(data_member){
  let link=this.base_url+'logOut';
  return new Promise(resolve => {
    this.http.post(link, data_member).subscribe(data => {
      this.data = data;
      resolve(this.data);
    }, (err) => {
      if (err.statusText == "Unknown Error"){
        this.data = 'notOnline';
      } else {
        this.data = err;//'error';
      }
      resolve(this.data);
    });
  });
}
```

```
gantipassword(data_gantipassword){
  let link=this.base_url+'dogantipassword';
  return new Promise(resolve => {
    this.http.post(link, data_gantipassword).subscribe(data => {
      this.data = data;
      resolve(this.data);
    }, (err) => {
      if (err.statusText == "Unknown Error"){
        this.data = 'notOnline';
      }
    });
  });
}
```

```
    } else {
      this.data = err;//'error';
    }
    resolve(this.data);
  });
});
}
```

```
loadVote(data_member){
  let link=this.base_url+'loadVote';
  return new Promise(resolve => {
    this.http.post(link, data_member).subscribe(data => {
      this.data = data;
      resolve(this.data);
    }, (err) => {
      if (err.statusText == "Unknown Error"){
        this.data = 'notOnline';
      } else {
        this.data = err;//'error';
      }
      resolve(this.data);
    });
  });
}
```

```
vote(data_vote){
  let link=this.base_url+'dovote';
  return new Promise(resolve => {
    this.http.post(link, data_vote).subscribe(data => {
      this.data = data;
      resolve(this.data);
    });
  });
}
```

```
}, (err) => {  
  if (err.statusText == "Unknown Error"){  
    this.data = 'notOnline';  
  } else {  
    this.data = err; //'error';  
  }  
  resolve(this.data);  
});  
});  
}  
  
}
```