

API LOGIN UNTUK ANDROID

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
date_default_timezone_set("Asia/Jakarta");
class Login extends CI_Controller {

    public function __construct()
    {
        parent::__construct();

        $this->load->model('Siswa_model', 'siswa', TRUE);
        $this->load->helper('form');

        $token = $this->input->post('token', true);
        $versi = $this->config->item('versi');

        /*
        if ($token != $versi || $QR != $qrcode ){
            $result = $this->errToken();
            echo json_encode($result);
        }else {
            $result = $this->AksesJamKehadiran();
            echo json_encode($result);
        }

        if ($token != $versi ){
            $result = $this->errToken();
            echo json_encode($result);
            exit();
        }

        */
    }
}
```

```

public function errToken(){
    $response['error']          = TRUE;
    $response['error_msg']     = "Illegal Operation .... Akses di tolak";
    return $response;
}

/**
 * Encrypting password
 * @param password
 * returns salt and encrypted password
 */
public function hashSSHA($password) {

    $salt = sha1(rand());
    $salt = substr($salt, 0, 10);
    $encrypted = base64_encode(sha1($password . $salt, true) . $salt);
    $hash = array("salt" => $salt, "encrypted" => $encrypted);
    return $hash;
}

/**
 * Decrypting password
 * @param salt, password
 * returns hash string
 */
public function checkhashSSHA($salt, $password) {

    $hash = base64_encode(sha1($password . $salt, true) . $salt);

    return $hash;
}

/**

```

```

* Storing new user
* returns user details
*/
public function storeUser($name, $email, $password, $phone, $nipd) {

    $uuid = uniqid('', true);

    $hash = $this->hashSHA($password);

    $encrypted_password = $hash["encrypted"]; // encrypted password
    $salt = $hash["salt"]; // salt

    $object = array(

        'unique_id' => $uuid,

        'name' => $name,

        'email' => $email,

        'encrypted_password' => $encrypted_password,

        'salt' => $salt,

        'created_at' => date('Y-m-d H:i:s'),

        'phone' => $phone,

        'nipd' => $nipd

    );

    $result = $this->db->insert('m_users', $object);

    // check for successful store

    if ($result) {

        $q = "SELECT * FROM m_users WHERE email ='$email'";

        $res = $this->db->query($q);

        $user = $res->row_array();

        return $user;
    } else {

        return false;
    }
}

```

```
    }  
}
```

```
/**  
 * Get user by email and password  
 */  
public function getUserByEmailAndPassword($email, $password) {  
  
    $q ="SELECT * FROM m_users WHERE email ='$email'";  
    $res = $this->db->query($q);  
  
    if ($res) {  
        $user = $res->row_array();  
        // verifying user password  
        $salt = $user['salt'];  
        $encrypted_password = $user['encrypted_password'];  
        $hash = $this->checkhashSHA($salt, $password);  
        // check for password equality  
        if ($encrypted_password == $hash) {  
            // user authentication details are correct  
            return $user;  
        }  
    } else {  
        return NULL;  
    }  
  
}  
  
/**  
 * Check user is existed or not  
 */
```

```
public function isUserExisted($email) {
```

```
    $q ="SELECT * FROM m_users WHERE email ='$email'";
```

```
    $res = $this->db->query($q)->num_rows();
```

```
    if ($res > 0) {
```

```
        return true;
```

```
    } else {
```

```
        return false;
```

```
    }
```

```
}
```

```
public function register () {
```

```
    // json response array
```

```
    $response = array("error" => FALSE);
```

```
    if (!empty($this->input->post('name')) && !empty($this->input->post('email')) && !empty($this->input->post('password')) && !empty($this->input->post('phone')) && !empty($this->input->post('nipd')) {
```

```
        // receiving the post params
```

```
        $name = $this->input->post('name');
```

```
        $email = $this->input->post('email');
```

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

```
        $phone = $this->input->post('phone');
```

```
        $nipd = $this->input->post('nipd');
```

```
        // check if user is already existed with the same email
```

```
        if ($this->isUserExisted($email)) {
```

```
            // user already existed
```

```
            $response["error"] = TRUE;
```

```
            $response["error_msg"] = "User already existed with " . $email;
```

```
            echo json_encode($response);
```

```
        } else {
```

```
            // create a new user
```

```

        $user = $this->storeUser($name, $email, $password, $phone,
$nipd);

        if ($user) {
            // user stored successfully
            $response["error"] = FALSE;
            $response["uid"] = $user["unique_id"]; //$user->unique_id;
            $response["user"]["name"] = $user["name"]; //$user->name;
            $response["user"]["email"] = $user["email"]; //$user-
>email;
            $response["user"]["created_at"] = $user["created_at"];
//$user->created_at;
            $response["user"]["updated_at"] = $user["updated_at"];
//$user->updated_at;
            $response["user"]["nipd"] = $user["nipd"]; //$user-
>updated_at;

            echo json_encode($response);
        } else {
            // user failed to store
            $response["error"] = TRUE;
            $response["error_msg"] = "Unknown error occurred in
registration!";
            echo json_encode($response);
        }
    }
} else {
    $response["error"] = TRUE;
    $response["error_msg"] = "Required parameters (name, email or
password) is missing!";
    echo json_encode($response);
}
}

```

```

public function cek_user(){
    // json response array
    $response = array("error" => FALSE);

```

```

if (isset($_POST['email']) && isset($_POST['password'])) {

    // receiving the post params

    $email = $_POST['email'];

    $password = $_POST['password'];

    // get the user by email and password

    $user = $this->getUserByEmailAndPassword($email, $password);

    if ($user != false) {
        // use is found

        $response["error"] = FALSE;

        $response["uid"] = $user["unique_id"];

        $response["user"]["name"] = $user["name"];

        $response["user"]["email"] = $user["email"];

        $response["user"]["created_at"] = $user["created_at"];

        $response["user"]["updated_at"] = $user["updated_at"];

        $response["user"]["nipd"] = $user["nipd"]; //$user->updated_at;

        $response["user"]["versi"] = $this->config->item('versi');

        echo json_encode($response);
    } else {
        // user is not found with the credentials

        $response["error"] = TRUE;

        $response["error_msg"] = "Email atau Password anda salah ...!";

        echo json_encode($response);
    }
} else {
    // required post params is missing

    $response["error"] = TRUE;

    $response["error_msg"] = "Required parameters email or password is
missing!";

    echo json_encode($response);
}
}

```

}