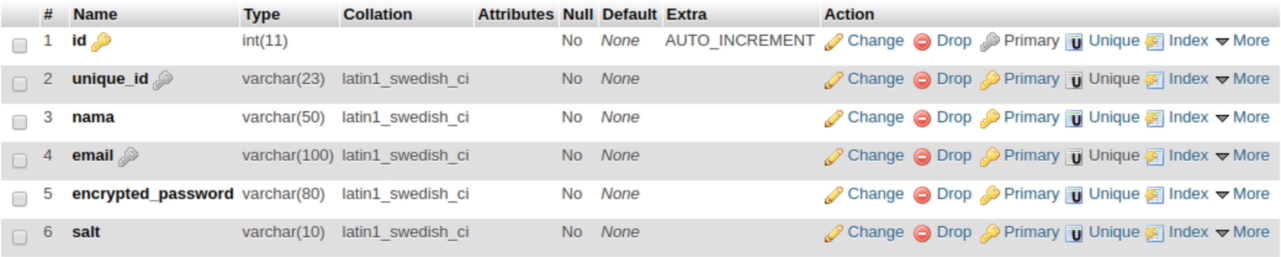
Modul Tutorial
Android Register Login with PHP & MySQL
(Part 1 membuat API menggunakan PHP)

1. Pertama buatlah database dengan nama **db_siswa** !

2. Selanjutnya buatlah tabel dengan nama **tbl_user** dan berisi field-field seperti dibawah ini !



	#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
□	1	**id**	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary Unique Index More
□	2	**unique_id**	varchar(23)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index More
□	3	**nama**	varchar(50)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index More
□	4	**email**	varchar(100)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index More
□	5	**encrypted_password**	varchar(80)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index More
□	6	**salt**	varchar(10)	latin1_swedish_ci		No	*None*		Change Drop Primary Unique Index More

```
create table tbl_user(
   id int(11) primary key auto_increment,
   unique_id varchar(23) not null unique,
   nama varchar(50) not null,
   email varchar(100) not null unique,
   encrypted_password varchar(80) not null,
   salt varchar(10) not null
);
```

3. Buatlah folder dengan nama **androidlogin** didalam *htdocs* atau kalau di linux di */var/www/html* !

4. Didalam folder androidlogin buatlah beberapa file **Config.php**, **DB_Connect.php**, **DB_Functions.php**, **login.php**, dan **register.php**, lalu tambahkan codingan seperti dibawah ini !

Config.php

```
<?php
define("DB_HOST", "localhost");
define("DB_USER", "root");
define("DB_PASSWORD", "admin");
define("DB_DATABASE", "db_siswa");
?>
```

DB_Connect.php

```
<?php
class DB_Connect {
    private $conn;

    // koneksi ke database
    public function connect() {
        require_once 'Config.php';

        // koneksi ke mysql database
        $this->conn = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_DATABASE);

        // return database handler
        return $this->conn;
    }
}
?
```

DB_Functions.php

```php
<?php

class DB_Functions {

    private $conn;

    // constructor
    function __construct() {
        require_once 'DB_Connect.php';

        // koneksi ke database
        $db = new Db_Connect();
        $this->conn = $db->connect();
    }

    // destructor
    function __destruct() {

    }

    public function simpanUser($nama, $email, $password) {
        $uuid = uniqid('', true);
        $hash = $this->hashSSHA($password);
        $encrypted_password = $hash["encrypted"]; // encrypted password
        $salt = $hash["salt"]; // salt

        $stmt = $this->conn->prepare("INSERT INTO tbl_user(unique_id, nama, 
email, encrypted_password, salt) VALUES(?, ?, ?, ?, ?)");
        $stmt->bind_param("sssss", $uuid, $nama, $email, $encrypted_password, 
$salt);
        $result = $stmt->execute();
        $stmt->close();

        // cek jika sudah sukses
        if ($result) {
            $stmt = $this->conn->prepare("SELECT * FROM tbl_user WHERE email 
= ?");
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $user = $stmt->get_result()->fetch_assoc();
            $stmt->close();

            return $user;
        } else {
            return false;
        }
    }

    /**
     * Get user berdasarkan email dan password
     */
    public function getUserByEmailAndPassword($email, $password) {

        $stmt = $this->conn->prepare("SELECT * FROM tbl_user WHERE email = ?");

        $stmt->bind_param("s", $email);

        if ($stmt->execute()) {
            $user = $stmt->get_result()->fetch_assoc();
            $stmt->close();

            // verifikasi password user
            $salt = $user['salt'];
            $encrypted_password = $user['encrypted_password'];
            $hash = $this->checkhashSSHA($salt, $password);
            // cek password jika sesuai
            if ($encrypted_password == $hash) {
                // autentikasi user berhasil
                return $user;
            }
        } else {
            return NULL;
        }
    }

    /**
     * Cek User ada atau tidak
     */
    public function isUserExisted($email) {
        $stmt = $this->conn->prepare("SELECT email from tbl_user WHERE email
= ?");

        $stmt->bind_param("s", $email);

        $stmt->execute();

        $stmt->store_result();

        if ($stmt->num_rows > 0) {
            // user telah ada
            $stmt->close();
            return true;
        } else {
            // user belum ada
            $stmt->close();
            return false;
        }
    }

    /**
     * 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;
    }

}

?>
```

login.php

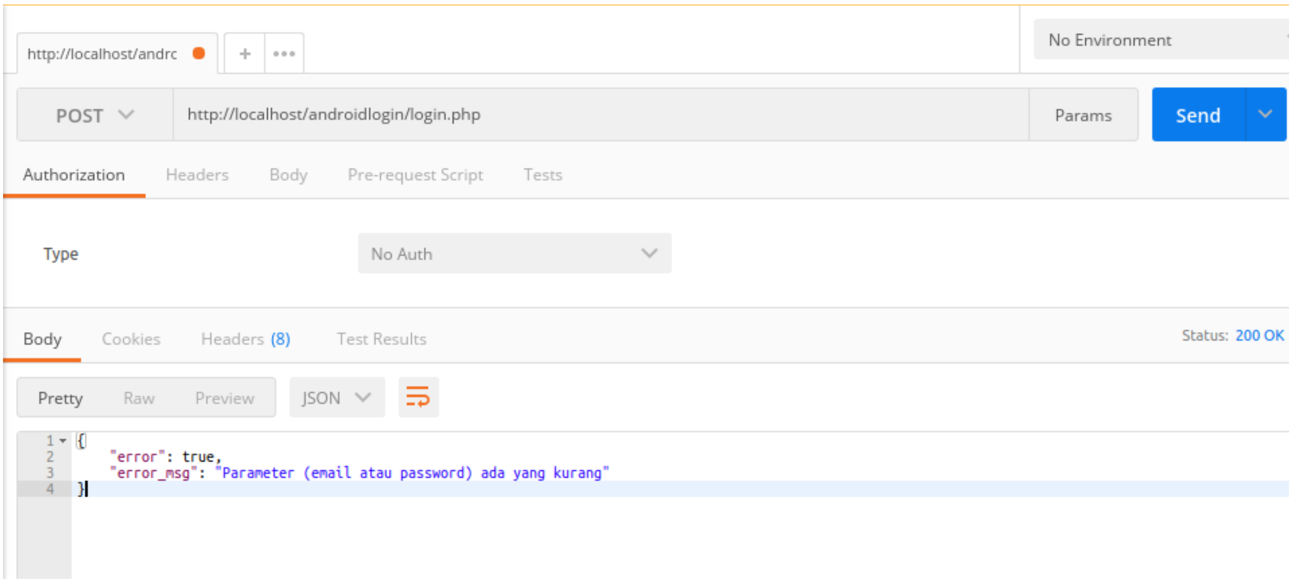
```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

// json response array
$response = array("error" => FALSE);

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

    // menerima parameter POST ( email dan password )
    $email = $_POST['email'];
    $password = $_POST['password'];

    // get the user by email and password
    // get user berdasarkan email dan password
    $user = $db->getUserByEmailAndPassword($email, $password);

    if ($user != false) {

        // user ditemukan
        $response["error"] = FALSE;
        $response["uid"] = $user["unique_id"];
        $response["user"]["nama"] = $user["nama"];
        $response["user"]["email"] = $user["email"];
        echo json_encode($response);
    } else {

        // user tidak ditemukan password/email salah
        $response["error"] = TRUE;
        $response["error_msg"] = "Login gagal. Password/Email salah";
        echo json_encode($response);
    }
} else {
    $response["error"] = TRUE;
    $response["error_msg"] = "Parameter (email atau password) ada yang kurang";
    echo json_encode($response);
}
?>
```

register.php

```
<?php

require_once 'DB_Functions.php';
$db = new DB_Functions();

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

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

    // menerima parameter POST ( nama, email, password )
    $nama = $_POST['nama'];
    $email = $_POST['email'];
    $password = $_POST['password'];

    // Cek jika user ada dengan email yang sama
    if ($db->isUserExisted($email)) {

        // user telah ada
        $response["error"] = TRUE;
        $response["error_msg"] = "User telah ada dengan email " . $email;
        echo json_encode($response);
    } else {

        // buat user baru
        $user = $db->simpanUser($nama, $email, $password);
        if ($user) {

            // simpan user berhasil
            $response["error"] = FALSE;
            $response["uid"] = $user["unique_id"];
            $response["user"]["nama"] = $user["nama"];
            $response["user"]["email"] = $user["email"];
            echo json_encode($response);
        } else {

            // gagal menyimpan user
            $response["error"] = TRUE;
            $response["error_msg"] = "Terjadi kesalahan saat melakukan
registrasi";
            echo json_encode($response);
        }
    }
} else {
    $response["error"] = TRUE;
    $response["error_msg"] = "Parameter (nama, email, atau password) ada yang
kurang";
    echo json_encode($response);
}
?>
```

5. Setelah semua file .php sudah dibuat, selanjutnya test API login dan register menggunakan aplikasi POSTMAN atau bisa menggunakan aplikasi REST Client lainnya. Jika berhasil muncul tampilan seperti dibawah ini !

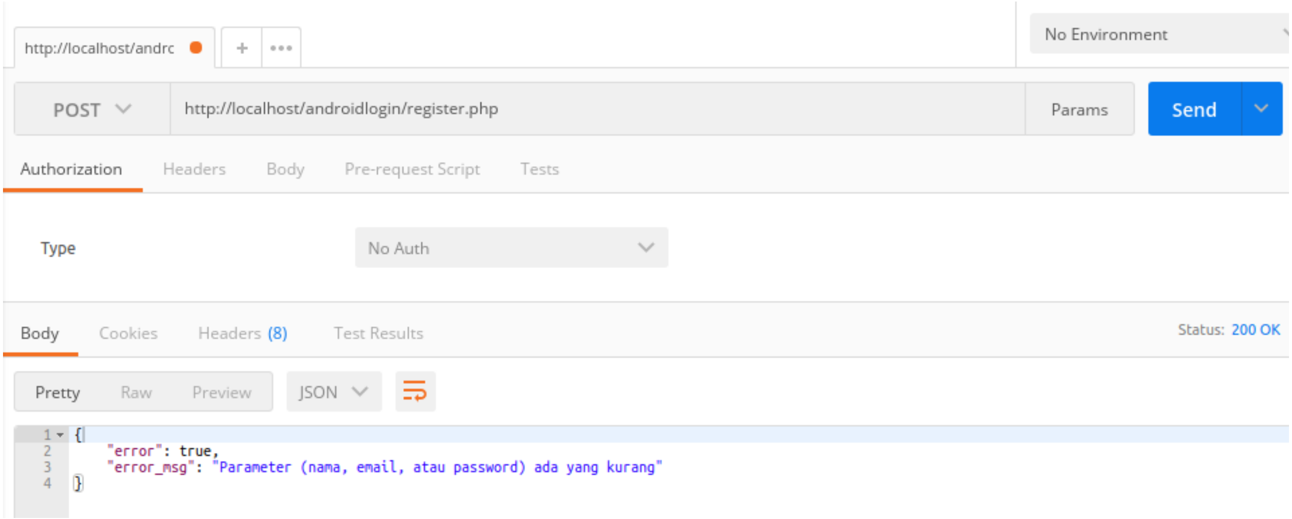


POST http://localhost/androidlogin/login.php

Type: No Auth

Status: 200 OK

```
{
    "error": true,
    "error_msg": "Parameter (email atau password) ada yang kurang"
}
```



POST http://localhost/androidlogin/register.php

Type: No Auth

Status: 200 OK

```
{
    "error": true,
    "error_msg": "Parameter (nama, email, atau password) ada yang kurang"
}
```

6. Tutorial diatas adalah API yang di simpan di server local. Tugas kalian adalah hostingkan API yang sudah kalian buat, lalu kirimkan url API login dan register yang sudah kalian buat ke email saya dadangiswan@gmail.com dengan :

    - Subject : (nama siswa) – XI RPL – Part 1
    - Isi:
        - URL login : (url login yang sudah di hosting)
        - URL register : (url register yang sudah di hosting)