

LAMPIRAN

1. Source Code Aplikasi Web

- Pembuatan Web Service API (Application Programming Interface)

```
Api.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Api extends MY_Controller {
5
6     private $user;
7
8     public function __construct()
9     {
10         parent::__construct();
11         $this->load->library('auth_library', NULL, 'auth');
12     }
13
14     private function verify_token()
15     {
16         $authorization = $this->input->get_request_header('Authorization');
17         if(preg_match('/^Basic([a-zA-Z0-9+\/-]*)$/i', $authorization, $matches)) {
18             return ['status' => FALSE, 'message' => 'Require basic authorization'];
19         }
20
21         $decoded = base64_decode($matches[1]);
22         if(FALSE === $decoded) {
23             return ['status' => FALSE, 'message' => 'Invalid basic authorization'];
24         }
25
26         [$username, $password] = explode(':', $decoded);
27
28         $user = $this->auth->get_user_by_username($username);
29         if(empty($user) || !$this->auth->verify_password($password, $user->password_hash)) {
30             return ['status' => FALSE, 'message' => 'Username atau password salah'];
31         }
32
33         if($user->role != ROLE_MAHASISWA) {
34             return ['status' => FALSE, 'message' => 'Invalid login method for this account'];
35         }
36
37         $this->user = $user;
38
39         return TRUE;
40     }
41
42     public function token_get()
43     {
44         $result = $this->verify_token();
```

- Source Code form Configurasi Login Web

```
Auth.php
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Auth extends MY_Controller {
5
6     protected $layout = 'layouts/auth/index';
7
8     public function __construct()
9     {
10         parent::__construct();
11         $this->load->library('auth_library', NULL, 'auth');
12     }
13
14     public function index()
15     {
16         return redirect('auth/login');
17     }
18
19     public function login_get()
20     {
21         $this->require_guest();
22         $this->render('login/index');
23     }
24
25     public function login_post()
26     {
27         $this->require_guest();
28         $username = $this->input->post('username');
29         $password = $this->input->post('password');
30
31         $this->auth->login($username, $password);
32         redirect('dashboard');
33     }
34
35     public function logout_post()
36     {
37         $this->require_auth();
38         $this->auth->logout();
39         redirect('auth/login');
40     }
41
42     public function ubah_password_get()
43     {
44         $this->require_auth();
```

- Source Code Tabel Data Aplikasi Web

```
Datatable.php x
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Datatable extends MY_Controller {
5
6     public function master_fakultas_post()
7     {
8         $this->require_auth(ROLE_ADMIN);
9
10        $this->load->model('fakultas_model', 'model');
11        $result = $this->model->datatable();
12
13        $this->render_json($result);
14    }
15
16    public function master_jurusan_post()
17    {
18        $this->require_auth(ROLE_ADMIN);
19
20        $this->load->model('jurusan_model', 'model');
21        $result = $this->model->datatable();
22
23        $this->render_json($result);
24    }
25
26    public function master_user_post()
27    {
28        $this->require_auth(ROLE_ADMIN);
29
30        $this->load->model('user_model', 'model');
31        $result = $this->model->datatable();
32
33        $this->render_json($result);
34    }
35
36    public function master_mahasiswa_post()
37    {
38        $this->require_auth(ROLE_ADMIN);
39
40        $this->load->model('mahasiswa_model', 'model');
41        $result = $this->model->datatable();
42
43        $this->render_json($result);
44    }
45 }
```

- Source Code Master User

```
4 class Master extends MY_Controller {
5
6     public function user_get()
7     {
8         $this->require_auth(ROLE_ADMIN);
9         $this->payload['title'] = 'Master User';
10        $this->payload['bottom_scripts'] = 'master/scripts/user';
11        $this->render('master/user');
12    }
13
14    public function user_post()
15    {
16        $this->require_auth(ROLE_ADMIN);
17        $this->load->model('user_model', 'model');
18
19        $type = $this->input->post('type');
20        switch($type)
21        {
22            case 'insert':
23                $data = $this->input->post(['username']);
24                $this->model->insert_data($data);
25                break;
26            case 'update':
27                $id = $this->input->post('id');
28                $data = $this->input->post(['username']);
29                $this->model->update_data($id, $data);
30                break;
31            case 'delete':
32                $id = $this->input->post('id');
33                $this->model->delete_data($id);
34                break;
35            default:
36                throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
37                break;
38        }
39    }
40 }
```

- **Source Code Master Mahasiswa**

```
    $this->render_json(['code' => 1, 'message' => 'OK']);
}

public function mahasiswa_get()
{
    $this->require_auth(ROLE_ADMIN);
    $this->payload['title'] = 'Master Mahasiswa';
    $this->payload['bottom_scripts'] = 'master/scripts/mahasiswa';

    $this->load->model('jurusan_model');
    $this->payload['jurusan_list'] = $this->jurusan_model->get_all();

    $this->load->model('semester_model');
    $this->payload['semester_list'] = $this->semester_model->get_all();

    $this->load->model('jadwal_pelajaran_model');
    $this->payload['jadwal_list'] = $this->jadwal_pelajaran_model->get_all();

    $this->render('master/mahasiswa');
}

public function mahasiswa_post()
{
    $this->require_auth(ROLE_ADMIN);
    $this->load->model('mahasiswa_model', 'model');

    $type = $this->input->post('type');
    switch($type)
    {
        case 'insert':
            $data = $this->input->post(['npm', 'nama', 'id_jurusan', 'id_semester']);
            $this->model->insert_data($data);
            break;
        case 'update':
            $id = $this->input->post('id');
            $data = $this->input->post(['npm', 'nama', 'id_jurusan', 'id_semester']);
            $this->model->update_data($id, $data);
            break;
        case 'add_jadwal':
            $id_mahasiswa = $this->input->post('id_mahasiswa');
            $id_jadwal = $this->input->post('id_jadwal');
            $this->model->add_jadwal($id_mahasiswa, $id_jadwal);
            break;
    }
}
```

- **Source Code Master Dosen**

```
    $this->render_json(['code' => 1, 'message' => 'OK']);
}

public function dosen_get()
{
    $this->require_auth(ROLE_ADMIN);
    $this->payload['title'] = 'Master Dosen';
    $this->payload['bottom_scripts'] = 'master/scripts/dosen';
    $this->render('master/dosen');
}

public function dosen_post()
{
    $this->require_auth(ROLE_ADMIN);
    $this->load->model('dosen_model', 'model');

    $type = $this->input->post('type');
    switch($type)
    {
        case 'insert':
            $data = $this->input->post(['nip', 'nama']);
            $this->model->insert_data($data);
            break;
        case 'update':
            $id = $this->input->post('id');
            $data = $this->input->post(['nip', 'nama']);
            $this->model->update_data($id, $data);
            break;
        case 'delete':
            $id = $this->input->post('id');
            $this->model->delete_data($id);
            break;
        default:
            throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
            break;
    }
}

    $this->render_json(['code' => 1, 'message' => 'OK']);
}
```

- **Sourcode Master Mata Kuliah**

```
        $this->render_json(['code' => 1, 'message' => 'OK']);
    }

    public function mata_kuliah_get()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->payload['title'] = 'Master Mata Kuliah';
        $this->payload['bottom_scripts'] = 'master/scripts/mata_kuliah';
        $this->render('master/mata_kuliah');
    }

    public function mata_kuliah_post()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->load->model('mata_kuliah_model', 'model');

        $type = $this->input->post('type');
        switch($type)
        {
            case 'insert':
                $data = $this->input->post(['kode', 'nama']);
                $this->model->insert_data($data);
                break;
            case 'update':
                $id = $this->input->post('id');
                $data = $this->input->post(['kode', 'nama']);
                $this->model->update_data($id, $data);
                break;
            case 'delete':
                $id = $this->input->post('id');
                $this->model->delete_data($id);
                break;
            default:
                throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
                break;
        }

        $this->render_json(['code' => 1, 'message' => 'OK']);
    }
}
```

- **Sourcode Master Jadwal Pelajaran**

```
}
$this->render_json(['code' => 1, 'message' => 'OK']);
}

public function jadwal_pelajaran_get()
{
    $this->require_auth(ROLE_ADMIN);
    $this->payload['title'] = 'Master Jadwal Pelajaran';
    $this->payload['bottom_scripts'] = 'master/scripts/jadwal_pelajaran';

    $this->load->model('mata_kuliah_model');
    $this->payload['mata_kuliah_list'] = $this->mata_kuliah_model->get_all();

    $this->load->model('semester_model');
    $this->payload['semester_list'] = $this->semester_model->get_all();

    $this->load->model('dosen_model');
    $this->payload['dosen_list'] = $this->dosen_model->get_all();

    $this->render('master/jadwal_pelajaran');
}

public function jadwal_pelajaran_post()
{
    $this->require_auth(ROLE_ADMIN);
    $this->load->model('jadwal_pelajaran_model', 'model');

    $type = $this->input->post('type');
    switch($type)
    {
        case 'insert':
            $data = $this->input->post(['kode', 'id_matkul', 'hari', 'jam_mulai', 'jam_selesai', 'total_sks', 'id_dosen', 'id_semester']);
            $this->model->insert_data($data);
            break;
        case 'update':
    
```

- **Sourcode Master Fakultas**

```
public function fakultas_get()
{
    $this->require_auth(ROLE_ADMIN);
    $this->payload['title'] = 'Master Fakultas';
    $this->payload['bottom_scripts'] = 'master/scripts/fakultas';
    $this->render('master/fakultas');
}

public function fakultas_post()
{
    $this->require_auth(ROLE_ADMIN);
    $this->load->model('fakultas_model', 'model');

    $type = $this->input->post('type');
    switch($type)
    {
        case 'insert':
            $data = $this->input->post(['kode', 'nama']);
            $this->model->insert_data($data);
            break;
        case 'update':
            $id = $this->input->post('id');
            $data = $this->input->post(['kode', 'nama']);
            $this->model->update_data($id, $data);
            break;
        case 'delete':
            $id = $this->input->post('id');
            $this->model->delete_data($id);
            break;
        default:
            throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
            break;
    }

    $this->render_json(['code' => 1, 'message' => 'OK']);
}
}
```

- **Sourcode Master Jurusan**

```
        $this->render_json(['code' => 1, 'message' => 'OK']);
    }

    public function jurusan_get()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->payload['title'] = 'Master Jurusan';
        $this->payload['bottom_scripts'] = 'master/scripts/jurusan';

        $this->load->model('fakultas_model');
        $this->payload['fakultas_list'] = $this->fakultas_model->get_all();

        $this->render('master/jurusan');
    }

    public function jurusan_post()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->load->model('jurusan_model', 'model');

        $type = $this->input->post('type');
        switch($type)
        {
            case 'insert':
                $data = $this->input->post(['kode', 'nama', 'id_fakultas']);
                $this->model->insert_data($data);
                break;
            case 'update':
                $id = $this->input->post('id');
                $data = $this->input->post(['kode', 'nama', 'id_fakultas']);
                $this->model->update_data($id, $data);
                break;
            case 'delete':
                $id = $this->input->post('id');
                $this->model->delete_data($id);
                break;
            default:
                throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
                break;
        }
    }
}
```

- **Sourcode Master Semester**

```
        $this->render_json(['code' => 1, 'message' => 'OK']);
    }

    public function semester_get()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->payload['title'] = 'Master Semester';
        $this->payload['bottom_scripts'] = 'master/scripts/semester';
        $this->render('master/semester');
    }

    public function semester_post()
    {
        $this->require_auth(ROLE_ADMIN);
        $this->load->model('semester_model', 'model');

        $type = $this->input->post('type');
        switch($type)
        {
            case 'insert':
                $data = $this->input->post(['tahun_ajaran', 'semester']);
                $this->model->insert_data($data);
                break;
            case 'update':
                $id = $this->input->post('id');
                $data = $this->input->post(['tahun_ajaran', 'semester']);
                $this->model->update_data($id, $data);
                break;
            case 'delete':
                $id = $this->input->post('id');
                $this->model->delete_data($id);
                break;
            default:
                throw new Exception('Type not found', ERR_INVALID_FORM_DATA);
                break;
        }
    }
}
```

2. Source Code Aplikasi Android

- Code Activity Xml 1

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context=".LoginActivity">
    <ScrollView
        android:id="@+id/login_form"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:gravity="center_horizontal"
                android:layout_marginTop="16dp">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:textColor="@android:color/tertiary_text_dark"
                    android:text="Absensi"
                    android:textAppearance="?android:attr/textAppearanceLarge"/>

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="match_parent"
                    android:textColor="@color/colorPrimary"
                    android:textStyle="bold"
                    android:text="QR Code"
                    android:textAppearance="?android:attr/textAppearanceLarge"/>

            </LinearLayout>

        </LinearLayout>
    </androidx.cardview.widget.CardView>
```

- Code Activity Xml 2

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    app:cardElevation="4dp"
    app:cardCornerRadius="4dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="16dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:textAppearance="?android:attr/textAppearanceLarge"
            android:layout_gravity="center_horizontal"
            android:textColor="@android:color/secondary_text_light"
            android:text="Login"/>

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp">

            <EditText
                android:id="@+id/text_username"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Username"
                android:inputType="text"
                android:maxLines="1"
                android:singleLine="true" />

        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
```

- Code Activity Xml 3

```

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/text_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:imeActionId="6"
        android:imeActionLabel="Sign in"
        android:imeOptions="actionUnspecified"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true" />
    </com.google.android.material.textfield.TextInputLayout>
    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ProgressBar
            android:id="@+id/login_progress"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_gravity="start"
            android:visibility="gone"/>
        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/button_sign_in"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_gravity="end"
            android:text="Sign in"
            android:textStyle="bold"
            android:textColor="@android:color/white"
            app:backgroundTint="@color/colorAccent"/>
    </FrameLayout>

```

- Code Activity Main

```

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />
    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />
</androidx.constraintlayout.widget.ConstraintLayout>

```


- Code Login Activity

```
public class LoginActivity extends AppCompatActivity {  
  
    // UI references.  
    private EditText textUsername;  
    private EditText textPassword;  
    private View progressBar;  
    private AppCompatActivity buttonSignIn;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_login);  
        // Set up the userLoginTask form.  
        textUsername = findViewById(R.id.text_username);  
        textPassword = findViewById(R.id.text_password);  
        textPassword.setOnEditorActionListener((textView, id, keyEvent) -> {  
            if (id == EditorInfo.IME_ACTION_DONE || id == EditorInfo.IME_NULL) {  
                attemptLogin();  
                return true;  
            }  
            return false;  
        });  
  
        buttonSignIn = findViewById(R.id.button_sign_in);  
        buttonSignIn.setOnClickListener((view) -> { attemptLogin(); });  
  
        progressBar = findViewById(R.id.login_progress);  
    }  
  
    /**  
     * Attempts to sign in or register the account specified by the userLoginTask form.  
     * If there are form errors (invalid email, missing fields, etc.), the  
     * errors are presented and no actual userLoginTask attempt is made.  
     */  
    private void attemptLogin() {  
  
        // Reset errors.  
        textUsername.setError(null);  
        textPassword.setError(null);  
  
        // Store values at the time of the userLoginTask attempt.  
        String username = textUsername.getText().toString();  
        String password = textPassword.getText().toString();  
  
        // ...  
    }  
}
```

- Code Activity 2

```
private void attemptLogin() {  
  
    // Reset errors.  
    textUsername.setError(null);  
    textPassword.setError(null);  
  
    // Store values at the time of the userLoginTask attempt.  
    String username = textUsername.getText().toString();  
    String password = textPassword.getText().toString();  
  
    boolean cancel = false;  
    View focusView = null;  
  
    // Check for a valid username field.  
    if (TextUtils.isEmpty(username)) {  
        textUsername.setError("Tidak boleh kosong");  
        focusView = textUsername;  
        cancel = true;  
    }  
  
    // Check for a valid password field.  
    if (TextUtils.isEmpty(password)) {  
        textPassword.setError("Tidak boleh kosong");  
        focusView = textPassword;  
        cancel = true;  
    }  
}  
  
if (cancel) {  
    // There was an error; don't attempt userLoginTask and focus the first  
    // form field with an error.  
    focusView.requestFocus();  
} else {  
    // Show a progress spinner, and kick off a background task to  
    // perform the user userLoginTask attempt.  
    showProgress(true);  
  
    Intent intent = new Intent(packageContext, LoginActivity.this, MainActivity.class);  
    startActivity(intent);  
    finish();  
    ApiService.authGet(username, password).enqueue(new Callback<AuthResponse>() {  
        @Override  
        public void onResponse(@NonNull Call<AuthResponse> call, @NonNull Response<AuthResponse> response) {  
            // ...  
        }  
    });  
}
```

- Code Activity 3

```
        startActivity(intent);
        finish();
        return;
    }
}

showProgress(false);
textPassword.setError("Username atau password salah");
textPassword.requestFocus();
}

@Override
public void onFailure(@NonNull Call<AuthResponse> call, @NonNull Throwable t) {
    t.printStackTrace();
    String message = t.getLocalizedMessage();
    Log.d("tag: exception", message != null ? message : "No message.");
    Toast.makeText(context LoginActivity.this, text "Terjadi kesalahan", Toast.LENGTH_SHORT).show();
    showProgress(false);
}
});
}
}

/**
 * Shows the progress UI and disables form
 */
private void showProgress(final boolean show) {

    int shortAnimTime = getResources().getInteger(android.R.integer.config_shortAnimTime);

    textUsername.setEnabled(!show);
    textPassword.setEnabled(!show);
    buttonSignIn.setEnabled(!show);

    progressBar.setVisibility(show ? View.VISIBLE : View.GONE);
    progressBar.animate().setDuration(shortAnimTime).alpha(
        show ? 1 : 0).setListener((AnimatorListenerAdapter) (animation) -> {
        progressBar.setVisibility(show ? View.VISIBLE : View.GONE);
    });
}
}
}
```

- Code Activity Scanner

```
public class ScannerActivity extends AppCompatActivity implements ZXingScannerView.ResultHandler, View.OnClickListener {

    private ZXingScannerView mScannerView;
    private FrameLayout frameLayoutCamera;
    private Button buttonReset;
    private TextView textViewQRCodeValue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scanner);

        frameLayoutCamera = findViewById(R.id.frame_layout_camera);
        initScannerView();
        initDefaultView();
    }

    @Override
    protected void onStart() {
        mScannerView.startCamera();
        doRequestPermission();
        super.onStart();
    }

    @Override
    protected void onPause() {
        mScannerView.stopCamera();
        super.onPause();
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        if(requestCode == 100) {
            initScannerView();
        }
    }

    private void doRequestPermission() {
        if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if(checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                requestPermissions(new String[] {Manifest.permission.CAMERA}, requestCode: 100);
            }
        }
    }
}
```

- Code Activity Scanner 2

```
private void doRequestPermission() {
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if(checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[] {Manifest.permission.CAMERA}, requestCode: 100);
        }
    }
}

private void initView() {
    mScannerView = new ZXingScannerView( context: this);
    mScannerView.setAutoFocus(true);
    mScannerView.setResultHandler(this);
    frameLayoutCamera.addView(mScannerView);
}

private void initViewDefaultView() {
    buttonReset = findViewById(R.id.button_reset);
    textViewQrCodeValue = findViewById(R.id.text_view_qr_code_value);

    buttonReset.setVisibility(View.GONE);
    buttonReset.setOnClickListener(this);
    textViewQrCodeValue.setText("QR Code Value");
}

@Override
public void onClick(View v) {
    if(v != null && v.getId() == R.id.button_reset) {
        mScannerView.resumeCameraPreview( resultHandler: this);
        initViewDefaultView();
    }
}

@Override
public void handleResult(Result rawResult) {
    textViewQrCodeValue.setText(rawResult.getText());
    buttonReset.setVisibility(View.VISIBLE);
}
}
```

- Code Activity Scanner 3

```
private void doRequestPermission() {
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if(checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            requestPermissions(new String[] {Manifest.permission.CAMERA}, requestCode: 100);
        }
    }
}

private void initView() {
    mScannerView = new ZXingScannerView( context: this);
    mScannerView.setAutoFocus(true);
    mScannerView.setResultHandler(this);
    frameLayoutCamera.addView(mScannerView);
}

private void initViewDefaultView() {
    buttonReset = findViewById(R.id.button_reset);
    textViewQrCodeValue = findViewById(R.id.text_view_qr_code_value);

    buttonReset.setVisibility(View.GONE);
    buttonReset.setOnClickListener(this);
    textViewQrCodeValue.setText("QR Code Value");
}

@Override
public void onClick(View v) {
    if(v != null && v.getId() == R.id.button_reset) {
        mScannerView.resumeCameraPreview( resultHandler: this);
        initViewDefaultView();
    }
}

@Override
public void handleResult(Result rawResult) {
    textViewQrCodeValue.setText(rawResult.getText());
    buttonReset.setVisibility(View.VISIBLE);
}
}
```