

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Penerapan REST Web Service

Pada bagian *controller* harus mendefinisikan metode yang akan digunakan untuk pemanggilan *web service* tersebut. *Controller* disini berperan untuk menentukan informasi apa yang akan dikirimkan ketika pengguna mengakses *web services* yang kita buat. yang dibuat menggunakan arsitektur REST maka *method* yang dapat digunakan adalah *method* yang didukung oleh protokol HTTP seperti *method GET, POST, DELETE, dan PUT.*

Berikut adalah contoh potongan kode untuk REST dan pembuatan *service.*

*Listing Kode 4.1 REST Login*

```
private void login() {
    EditText nim = (EditText) view.findViewById(R.id.nim);
    EditText pass = (EditText) view.findViewById(R.id.pass);
    if (nim.getText().toString().isEmpty()) {
        Toast.makeText(activity, "NIM tidak boleh kosong!",
        Toast.LENGTH_LONG).show();
        nim.requestFocus();
        return;
    }
    if (nim.getText().toString().length() != 11) {
        Toast.makeText(activity, "Format NIM salah!",
        Toast.LENGTH_LONG).show();
        nim.requestFocus();
        return;
    }
    if (pass.getText().toString().isEmpty()) {
        Toast.makeText(activity, "Password tidak boleh kosong!",
        Toast.LENGTH_LONG).show();
        pass.requestFocus();
        return;
    }

    User u = new User();
    u.setNim(Long.valueOf(nim.getText().toString()));
    u.setPassword(pass.getText().toString());
    activity.showProgress("Loading ...");
    Restful.login(u, new Callback<Restful.UserCallback>() {
        @Override
        public void onResponse(Call<Restful.UserCallback> call,
        Response<Restful.UserCallback> response) {
            activity.dismissProgress();
            if (response.body() != null) {
                if (response.body().error) {
                    Toast.makeText(activity, "NIM atau password
                    salah!", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(activity, "Login berhasil!",
```

```

Toast.LENGTH_LONG).show();

App.instance().setUser(response.body().data);
    App.instance().save();
    App.instance().loadData();
    Intent intent = new Intent(activity,
MainActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    activity.startActivity(intent);
    activity.finish();
    }
}
}

```

Pada *listing* kode 4.1 diatas dapat dilihat kode untuk meminta data *user*. Kode tersebut memanggil *service* untuk menerjemahkan data dalam bentuk JSON. Kode untuk *service* dapat dilihat pada *listing* kode 4.2.

#### Listing Kode 4.2 Service Login

```

public static void login(User user, Callback<UserCallback> c) {
    HttpLoggingInterceptor interceptor = new
HttpLoggingInterceptor();
    interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
    OkHttpClient client = new
OkHttpClient.Builder().addInterceptor(interceptor).build();
    Retrofit retrofit = new Retrofit.Builder()

.baseUrl(API).addConverterFactory(GsonConverterFactory.create(new
Gson()))
    .client(client)
    .build();
    Network register = retrofit.create(Network.class);

    Call<UserCallback> cal = register.login(user);
    cal.enqueue(c);
}

```

Pada *listing* kode 4.2 pemanggilan data *user* dari web yang kemudian akan ditampilkan di android.

#### Listing Kode 4.3 REST Jadwal Acara

```

public void init() {
    if (App.instance().getJadwalAcara().isEmpty())
        activity.showProgress("Loading ...");
    initSpinner();
    initView();

    Restful.getAcara(new Callback<Restful.JadwalAcaraCallback>()

```

```

{
    @Override
    public void onResponse(Call<Restful.JadwalAcaraCallback>
call, Response<Restful.JadwalAcaraCallback> response) {
        activity.dismissProgress();
        if (swipeRefreshLayout.isRefreshing())
            swipeRefreshLayout.setRefreshing(false);
        if (response.body() != null) {
            if (response.body().data.isEmpty()) {
                initView();
                return;
            }
        }
        App.instance().setJadwalAcara(response.body().data);
        App.instance().save();
        initSpinner();
        initView();
    } else {
        Toast.makeText(activity, "ERROR SERVER NOT
FOUND", Toast.LENGTH_LONG).show();
    }
}
}

```

Pada *listing* kode 4.3 diatas dapat dilihat kode untuk meminta data jadwal acara. Kode tersebut memanggil *service* untuk menerjemahkan data dalam bentuk JSON. Hasil pemanggilan *service* dapat dilihat pada *listing* kode 4.4.

#### Listing Kode 4.4 Service Jadwal Acara

```

public static void getAcara(Callback<JadwalAcaraCallback> c) {
    HttpLoggingInterceptor interceptor = new
HttpLoggingInterceptor();
    interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
    OkHttpClient client = new
OkHttpClient.Builder().addInterceptor(interceptor).build();
    Retrofit retrofit = new Retrofit.Builder()

.baseUrl(API).addConverterFactory(GsonConverterFactory.create(new
Gson())).client(client)
        .build();
    Network register = retrofit.create(Network.class);
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd
HH:mm");
    DataModel<String, String> model = new DataModel<>();
    Calendar calendar = Calendar.getInstance();
    calendar.add(Calendar.MONTH, -1);
    model.header = df.format(calendar.getTime());
    Call<JadwalAcaraCallback> cal = register.getAcara(model);
    cal.enqueue(c);
}

```

Pada *listing* kode 4.4 pemanggilan data jadwal acara dari web yang kemudian akan ditampilkan di android.

## 4.2 Implementasi *User Interface*

Setelah perancangan aplikasi penjadwalan acara selesai dirancang dan dibangun, maka aplikasi akan di implementasikan dengan cara diuji terlebih dahulu untuk melihat apakah sudah sesuai dengan yang diharapkan atau tidak, Pada tahap ini aplikasi dijalankan dan digunakan oleh *user*. Berikut adalah hasil implementasi pengujian terhadap *interface* yang terdapat pada perancangan aplikasi penjadwalan acara berbasis android.

### 4.2.1 *Activity Splash Screen*

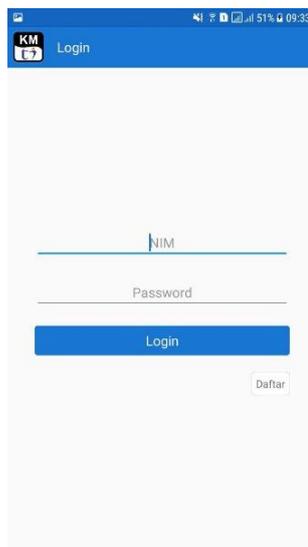
Pada saat pertama kali *user* menjalankan aplikasi penjadwalan acara maka akan muncul *activity splashscreen* yang dapat dilihat pada gambar 4.1.



**Gambar 4.1** *Splash Screen*

### 4.2.2 *Activity Login*

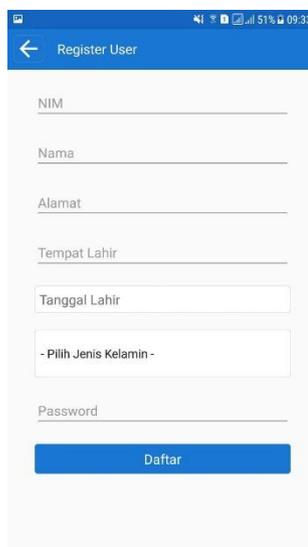
Pada *activity login*, *user* harus mengisi kolom NIM dan *password*. Kemudian *user* harus menekan tombol *login* agar dapat mengakses jadwal acara. Pastikan *user* telah melakukan registrasi sebelumnya. Jika *user* belum registrasi maka *user* harus registrasi terlebih dahulu dengan menekan tombol daftar. Berikut *activity login* dapat dilihat pada gambar 4.2.



**Gambar 4.2** Activity Login

### 4.2.3 Activity Registrasi

Pada *Activity* Registrasi, *user* harus mengisi NIM, nama, alamat, tempat lahir, tanggal lahir, jenis kelamin dan password. *User* memilih tombol register untuk daftar. Berikut *activity* registrasi yang dapat dilihat pada gambar 4.3.



**Gambar 4.3** Activity Registrasi

### 4.2.4 Activity Menu Utama

Pada *activity* menu utama menampilkan NIM dan nama user serta 4 tombol yaitu tombol *logout*, tombol menu jadwal acara, tombol menu acara saya dan

tombol menu tentang aplikasi. Berikut *activity* menu utama yang dapat dilihat pada gambar 4.4.



**Gambar 4.4** *Activity* Menu Utama

#### 4.2.5 *Activity* Menu Jadwal Acara

*Activity* menu jadwal acara merupakan halaman yang menampilkan informasi jadwal acara seperti gambar acara, nama acara, tanggal acara, lokasi acara dan terdapat pilihan kategori acara. Berikut *activity* menu jadwal acara yang dapat dilihat pada gambar 4.5.



**Gambar 4.5** *Activity* Menu Jadwal Acara

#### 4.2.6 *Activity* Detail Acara

*Activity* detail acara merupakan halaman yang menampilkan informasi detail dari jadwal acara seperti gambar acara, nama acara, deskripsi acara, fasilitas

acara, lokasi acara, waktu acara, dan lokasi acara. Pada halaman ini juga terdapat *button* ikuti untuk mengikuti acara. Berikut *activity* jadwal acara yang dapat dilihat pada gambar 4.6.



**Gambar 4.6** *Activity* Detail Acara

#### 4.2.7 *Activity* Notifikasi Acara

*Activity* notifikasi acara merupakan halaman yang menampilkan pemberitahuan 30 menit sebelum acara akan dimulai jika *user* sudah mengikuti acara pada halaman detail acara. Pada halaman ini menampilkan gambar acara, nama acara, lokasi acara, waktu acara serta *button* hadir untuk menghadiri acara dan *button* tidak jika tidak bisa menghadiri acara. Berikut *activity* notifikasi acara yang dapat dilihat pada gambar 4.7.



**Gambar 4.7** Activity Notifikasi Acara

#### 4.2.8 Activity Acara Saya

*Activity* acara saya merupakan halaman yang menampilkan informasi acara-acara yang telah diikuti pada halaman jadwal acara. Pada halaman ini menampilkan nama acara, kategori acara waktu acara dan tempat acara. Berikut *activity* acara saya yang dapat dilihat pada gambar 4.8.



**Gambar 4.8** Activity Acara Saya

#### 4.2.9 Activity Detail Acara Saya

*Activity* detail acara saya merupakan halaman yang menampilkan informasi detail dari *activity* acara saya seperti gambar acara, nama acara, deskripsi acara, fasilitas acara, lokasi acara, waktu acara dan pendaftaran acara. Pada halaman ini juga terdapat kolom komentar untuk mengomentari acara. Berikut *activity* detail acara saya yang dapat dilihat pada gambar 4.9.



**Gambar 4.9** Activity Detail Acara

#### 4.2.10 Activity Tentang Aplikasi

*Activity* tentang aplikasi merupakan halaman yang menampilkan informasi dari aplikasi. Berikut *activity* tentang aplikasi dapat dilihat pada gambar



**Gambar 4.10** Activity Tentang Aplikasi

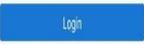
### 4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi apakah seluruh fungsi yang ada pada aplikasi sudah berjalan sesuai dengan rancangan.

#### 4.3.1 Pengujian Activity Login

Pengujian *activity login* yaitu pengujian dimana *user* membuka aplikasi maka *user* harus melakukan login terlebih dahulu agar bisa menggunakan aplikasi. Pengujian ini bertujuan untuk mengetahui fungsionalitas dan validasi yang terdapat pada *activity login*. Hasil pengujian dapat dilihat pada tabel 4.1.

**Tabel 4.1** Pengujian Login

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
1	Pada saat <i>user login</i> tetapi <i>user</i> lupa atau tidak memasukan NIM		Sistem menampilkan keterangan NIM tidak boleh kosong		Berhasil
2	Pada saat <i>user login</i> tetapi <i>user</i> lupa atau tidak memasukan password		Sistem menampilkan keterangan <i>password</i> tidak boleh kosong		Berhasil

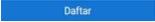
No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
3	Pada saat <i>user login</i> tetapi <i>user</i> lupa atau salah memasukan NIM dan password		Sistem menampilkan keterangan NIM atau <i>password</i> salah		Berhasil

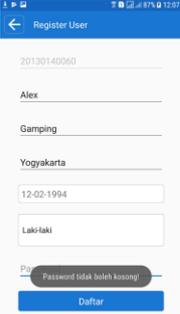
#### 4.3.2 Pengujian Activity Register

Pengujian *activity register* yaitu pengujian dimana *user* membuka aplikasi maka *user* harus melakukan login terlebih dahulu tetapi *user* tidak bisa melakukan *login* jika belum melakukan registrasi akun. Oleh karena itu *user* harus melakukan registrasi dan mengisi *form* registrasi agar bisa *login* aplikasi. Pengujian ini bertujuan untuk mengetahui fungsionalitas dan validasi yang terdapat pada *activity register*. Hasil pengujian dapat dilihat pada tabel 4.2.

Tabel 4.2 Pengujian Register

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
1	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi NIM		Sistem menampilkan keterangan NIM tidak boleh kosong		Berhasil
2	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> memasukan NIM tidak sesuai dengan format NIM di KMTI		Sistem menampilkan keterangan format NIM salah		Berhasil
3	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi nama		Sistem menampilkan keterangan nama tidak boleh kosong		Berhasil

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
4	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi alamat		Sistem menampilkan keterangan alamat tidak boleh kosong		Berhasil
5	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi tempat lahir		Sistem menampilkan keterangan tempat lahir tidak boleh kosong		Berhasil
6	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi tanggal lahir		Sistem menampilkan keterangan tanggal lahir tidak boleh kosong		Berhasil

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
7	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> memasukan tahun lahir yang tahun itu belum terjadi		Sistem menampilkan keterangan bahwa tahun lahir salah		Berhasil
8	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum memilih jenis kelamin		Sistem menampilkan keterangan jenis kelamin harus dipilih		Berhasil
9	Pada saat <i>user</i> mengisi <i>form</i> registrasi tetapi <i>user</i> lupa atau belum mengisi <i>password</i>		Sistem menampilkan keterangan <i>password</i> tidak boleh kosong		Berhasil

### 4.3.3 Pengujian *Activity* Detail Acara

Pengujian *activity* detail acara yaitu pengujian ketika *user* sudah melakukan *login*. Kemudian *user* memilih informasi jadwal acara dan melihat informasi detail

acara. Pengujian ini bertujuan untuk mengetahui apakah *user* bisa mengikuti acara, mendapatkan notifikasi acara, dan memberikan komentar acara. Hasil pengujian dapat dilihat pada tabel 4.3

**Tabel 4.3** Pengujian Detail Acara

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
1	Pada saat <i>user</i> melihat detail jadwal acara dan akan mengikuti acara dengan menekan <i>button</i> ikuti acara.		Sistem menampilkan keterangan acara telah diikuti dan akan mendapatkan notifikasi menjelang acara dimulai.		Berhasil
2	Pada saat <i>user</i> telah mengikuti acara dan mendapatkan notifikasi lalu <i>user</i> akan menghadiri acara maka dengan cara menekan <i>button</i> hadir		Sistem menampilkan informasi bahwa <i>user</i> menghadiri acara dan dapat mengisi komentar di detail acara		Berhasil

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapat	Status
3	Pada saat <i>user</i> telah mengikuti acara dan mendapatkan notifikasi lalu <i>user</i> tidak bisa menghadiri acara maka dengan cara menekan <i>button</i> tidak		Sistem akan menampilkan informasi bahwa <i>user</i> tidak bisa mengikuti acara dan sistem akan menghapus dari acara <i>user</i>		Berhasil
4	Menghapus acara dengan menekan <i>button</i> hapus dari detail acara.		Sistem menampilkan keterangan acara berhasil dihapus.		Berhasil

#### **4.4 Pembahasan**

Setelah melakukan pengujian pada aplikasi, dapat dilihat bahwa semua fungsi dari aplikasi berjalan sesuai dengan rancangan yang telah dibuat. Dengan begitu aplikasi sudah bisa menampilkan informasi penjadwalan acara Keluarga Mahasiswa Teknik Informatika (KMTI) di Universitas Muhammadiyah . Aplikasi juga dapat menampilkan notifikasi ketika acara akan dimulai dan *user* bisa memberikan komentar ketika telah menghadiri acara. Dengan hasil tersebut, dapat dikatakan bahwa tujuan dari penelitian sudah tercapai, yaitu mahasiswa mendapatkan informasi penjadwalan acara Keluarga Mahasiswa Teknik Informatika di Universitas Muhammadiyah Yogyakarta melalui aplikasi android tanpa menyatu dengan informasi lainnya.