JEITECH – Journal of Electrical Engineering and Information technology
Department of Electrical Engineering University of Mataram
Vol. 1, No. 1, June 2023, pp. 1-13 ; E-ISSN: 2987-7830

Aplikasi Sistem Administrasi Inventarisasi dan Peminjaman Barang Berbasis Android

Rangga Satria Wirabangsa[1], Dwi Ratnasari [1] , Giri Wahyu Wiriasto [1]
[1] Jurusan Teknik Elektro – Universitas Mataram, 83127 – Lombok, Indonesia

ARTICLE INFO

Article history :

Received May 16, 2023
Revised May 16, 2023
Accepted June 29, 2023

Keywords :

Information System
Android Application
REST API
Kotlin
Inventarisasi dan Peminjaman barang

ABSTRACT

Management information systems (MIS) have an important role in managing data. This article discusses the development of MIS applications that are used to facilitate the management of goods rental transaction data in government institutions. Usually the rental process is written in the tenant's register book, besides that the condition of the goods being rented cannot be monitored in real time. This MIS was developed in the form of a web-based server and Android mobile application. Webserver base application, used as the main database system and as an admin page when doing data updating work. Meanwhile, the mobile-based application provides a rental report data resume page that is recorded regularly and will be needed by executives. The advantage of this mobile basis is that the lessee can easily apply for a practical rental of goods. If approved, the tenant comes directly to the office to pick up the goods and complete the rental documents. Webserver-based applications are developed using PHP and mobile applications are developed using Kotlin. While the API interface uses the REST method. Tests carried out are in the form of alpha-testing to test the running application system and beta-testing for user satisfaction. The percentage of alpha and beta testing results was 100% and 88.60%.

Corresponding Author:

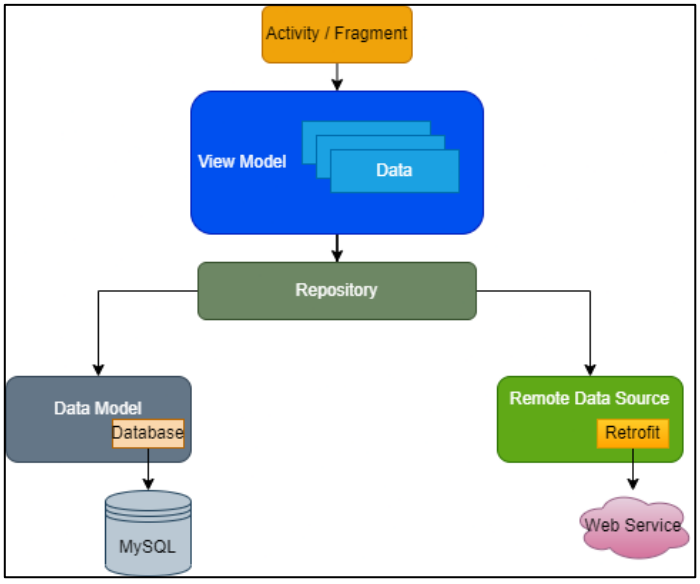
Rangga Satria Wirabangsa, Jurusan Teknik Elektro Universitas Mataram, Jalan Majapahit 63 Kota Mataram, 83127 - Lombok, Indonesia
Email: ranggasatria59@gmail.com

1. INTRODUCTION

Teknologi merupakan produk hasil pemikiran yang digunakan untuk mempermudah pekerjaan manusia. Ini juga berkaitan dengan kemajuan dan inovasi dalam mengelola dan mendapatkan informasi melalui media internet. Saat ini, hampir semua aktifitas operasional organisasi publik dan institusi pemerintah menyediakan layanan secara *online.* Seperti menggunakan sistem informasi manajemen, website dan aplikasi *mobile* [1]. Salah satu kemajuan teknologi berupa adanya sistem informasi yang dapat berjalan pada lintas platform atau bisa disebut *multi-platform. Multi-platform* merupakan kondisi dimana aplikasi yang dikembangkan dapat dibekerja pada beberapa kondisi karaktekteristik sistem yang berbeda [2]. Istilah multi-platform ini tidak dapat dipisahkan dari perkembangan teknologi *Application Programming Interface* (*API*). *API* merupakan teknologi yang mendukung komunikasi pertukaran data antara dua atau lebih aplikasi. W*eb service* merupakan layanan yang mengekspos data melalui internet dan memfasilitasi pertukaran data antara beberapa aplikasi yang berbeda [3][4][14]. *API* berfungsi untuk membuat rancangan *interface* terprogram melakukan *request* dan mendapatkan *response* dari basis data berdasarkan prinsip *REST* [5]. *REST* atau *Representasional State Transfer* merupakan prinsip yang memungkinkan seseorang untuk dapat mengakses data melalui *Hyper-Text Transfer Protocol (HTTP)*, dan menyediakan cara terbaik bagi pengguna

berkomunikasi dengan berbagai lingkungan sistem pengembangan yang berbeda tadi [6][15]. Dalam artikel ini dibahas pengembangan sistem informasi manajemen administrasi pelayanan berbasis web server dan mobile. Rancangan sistem informasi administrasi institusi pemerintah yang berjalan pada platform *web* memiliki fungsi untuk mengelola peminjaman barang dan menyajikan berbagai informasi peminjaman. Informasi peminjaman yang diberikan pada sistem tersebut berupa laporan permintaan, peminjaman, pengembalian barang, jenis barang, dan data pengguna. Dimulai dari user mengajukan peminjaman barang, admin melakukan penerimaan atau penolakan peminjaman, user mengajukan pengembalian barang, admin melakukan penerimaan barang kembali dan admin dapat mencetak laporan pengembalian barang.

Sistem informasi administrasi untuk peminjaman barang ini dirancang dengan menyediakan *web service* dimana nantinya *web service* ini digunakan untuk mengintegrasikan data dengan aplikasi versi *mobile* berbasis *android*. *Web* service ini menggunakan teknologi *web REST* yang melakukan pertukaran data menggunakan protocol http melalui *retrofit*2 client dan gson converter pada *android*. *Retrofit*2 adalah library klien http yang digunakan sebagai *rest* client, dengan adanya *retrofit* ini tidak perlu membuat metod sendiri untuk terhubung ke layanan *web rest*. Sedangkan gson converter adalah library untuk mengubah objek java ke dalam format *json* dan mengubah *json* ke java. *Retrofit*2 ini melakukan proses pertukaran data dari format *json* atau *xml*, kemudian format tersebut diubah ke bentuk plain old java objects *(pojos)*. Adapun *request* yang tersedia berupa *method post*, *get*, *put*, *delete*, dan *patch*. *Retrofit* ini dibangun dari beberapa library memanfaatkan protokol *http* yang dikembangkan oleh developer seperti https://squareup.com/ untuk menangani proses transmisi data (permintaan) melalui jaringan [8]. API ini berpusat pada siklus *request* dan *response*. *Request* yang dimaksud adalah proses mengirim permintaan data dari client ke server. Sedangkan *response* merupakan data yang dikirim server sesuai permintaan client [7]. Dibawah ini merupakan contoh aplikasi yang mengakses *api* menggunakan *retrofit*.



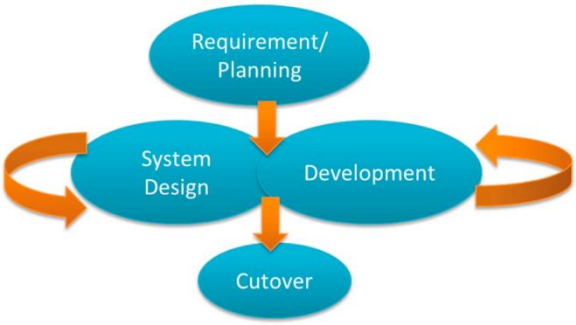
Gambar 1. Arsitektur aplikasi android dengan library retrofit [10]

Dari uraian dan deskripsi singkat tentang rancangan sistem informasi dan *retrofit* yang akan dibuat maka penelitian ini bermaksud untuk membuat suatu aplikasi sistem informasi peminjaman administrasi kantor berbasis multiplatform yang akan dijalankan pada *web* dan *android*. Aplikasi ini akan mengintegrasikan data menggunakan *api* dan *retrofit* agar data pada aplikasi akan bernilai sama dengan *website*.

2. METODE

A. Metode Pengembangan

Gambar 2 menunjukkan metode pengembangan yang menggambarkan desain dari Rapid Application Development yang digunakan dalam pengembangan aplikasi.

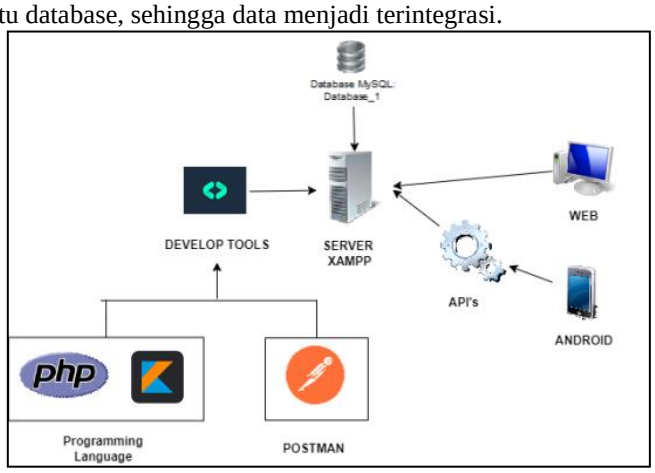


Gambar 2. Rapid *Application Development [11]*

Gambar 2 menunjukkan skema pengembangan Rapid Application Development yang terdiri dari *Requirement planning, RAD design workshop, dan implementation*. Aplikasi yang akan dibuat akan menerapkan pengembangan menggunakan metode RAD skema RAD lebih jelas dapat dilihat dibawah ini :
Tahap pertama yaitu *Requirement planning* dalam Tahapan ini dilakukan pengumpulan data dengan Teknik wawancara dengan *stakeholder* terkait kebutuhan sistem aplikasi, mengumpulkan serta menganalisis alat dan bahan yang digunakan peneliti untuk menunjang pembuatan aplikasi.
Tahap kedua merupakan tahap RAD yakni *design workshop* yang terdiri dari dua bagian yaitu '*work with user'* dan '*build the system'* pada tahapan ini merupakan proses desain sistem yang dilakukan antara peneliti dengan pegawai dan user berdasarkan hasil yang sudah diperoleh dari proses Requirement Planning, Tahapan ini merupakan proses pembangunan aplikasi berdasarkan desain sistem yang telah dibuat. Tahapan ini dilakukan antara peneliti, pegawai dan user.

B. Arsitektur Sistem

Tahap ini merupakan tahap perancangan arsitektur sistem dimana, data-data yang ada pada sistem seluruhnya akan disimpan dalam satu database, sehingga data menjadi terintegrasi.



Gambar 3. Rancangan Arsitektur Aplikasi Web dan Mobile

Gambar 3 menunjukkan arsitektur Sistem yang diusulkan, terdiri dari *develop tools* dan *web server*. Tahap *develop tools* merupakan tahap pengembangan sistem menggunakan bahasa pemrograman PHP untuk sisi *backend web*, *bootstrap* untuk sisi *frontend* web dan *kotlin Object Oriented Programming* dari sisi android. Sistem yang dikembangkan untuk Android menggunakan perangkat lunak *postman* untuk melakukan request *GET* dalam pembuatan API yang berfungsi untuk menghubungkan Android ke database dalam bahasa PHP.

C. Perancangan *API*

Pembuatan API merupakan proses yang dilakukan sebagai penghubung antara aplikasi yang berbeda supaya dapat terintegrasi dengan web service.

```
<?php
$HOST = 'localhost';
$USER = 'root';
$PASS = '';
$DB = 'db_pinjam_barang';
$CON =
mysqli_connect($HOST,$USER,$PASS,$DB)
;
?>
```

Gambar 4. API *connection-Android*

Gambar 4 merupakan API koneksi yang berfungsi sebagai penghubung ke database MySQL dengan nama database *db_pinjam_barang*.

```
<?php
require_once('connection-
android.php');
$result = array();
$query = mysqli_query($CON,"SELECT *
FROM tbl_barang ORDER BY id_barang
DESC");
while($row =
mysqli_fetch_assoc($query)){
  $result[] = $row;
}
echo
json_encode(array('result'=>$result));
?>
```

Gambar 5. API *Read Barang*

Gambar 5 merupakan API yang berfungsi untuk membaca data barang pada *tbl_barang*.

```
<?php
require_once('connection-
android.php');
$result = array();
$query = mysqli_query($CON,"SELECT *
FROM tbl_request
INNER JOIN tbl_barang ON
tbl_request.id_barang =
tbl_barang.id_barang
INNER JOIN user ON tbl_request.id_user
= user.id_user ORDER BY id DESC ");
while($row =
mysqli_fetch_assoc($query)){
  $result[] = $row;
}
echo
json_encode(array('result'=>$result))
;
?>
```

Gambar 6.API Read Permintaan

Gambar 6 merupakan API yang berfungsi untuk membaca data permintaan peminjaman dengan syarat '*id_barang'* pada *tbl_Request* harus memiliki nilai yang sama pada '*id_barang'* pada *'tbl_barang'* dan *'id_user'* pada *tbl_Request* harus sama dengan tabel *user*.

```
<?php
require_once('connection-
android.php');
$nama_barang  = $_POST['nama_barang'];
$stok_barang  = $_POST['stok_barang'];
if(!$nama_barang){
  echo
json_encode(array('message'=>'require
d field is empty.'));
}else{
$query  =  mysqli_query($CON,  "INSERT
INTO     tbl_barang     (nama_barang,
stok_barang)                  VALUES
('$nama_barang','$stok_barang')");
if($query){
    echo
json_encode(array('message'=>'barang
data successfully added.'));
  }else{
    echo
json_encode(array('message'=>'barang
data failed to add.'));
  }
}
?>
```

Gambar 7. API - *tambah barang*

Gambar 7 merupakan API yang berfungsi untuk menambahkan data pada '*tbl_barang'*.

```
<?php
require_once('connection-
android.php');
$id = $_POST['id_barang'];
$nama_barang  = $_POST['nama_barang'];
$stok_barang  = $_POST['stok_barang'];
if(!$id    ||    !$nama_barang    ||
!$stok_barang){
  echo
json_encode(array('message'=>'require
d field is empty.'));
}else{
$query  =  mysqli_query($CON,  "UPDATE
tbl_barang                        SET
nama_barang='$nama_barang',
stok_barang='$stok_barang'       WHERE
id='$id_barang'");
if($query){
    echo
json_encode(array('message'=>'barang
data successfully updated.'));
  }else{
    echo
json_encode(array('message'=>'barang
data failed to update.'));
  }
}
?>
```

Gambar 8. API - *update barang*

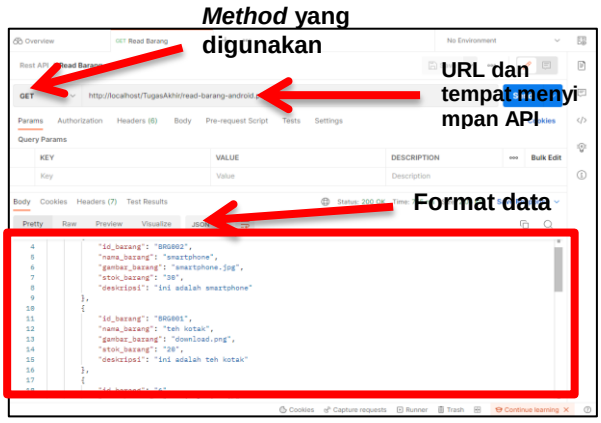
Gambar 8 merupakan API yang berfungsi untuk melakukan update data pada '*tbl_barang'* baik berupa *nama barang* maupun *stok barang*.

```
<?php
require_once('connection-android.php');
$id = $_POST['id'];
$search_request = mysqli_query($CON,"SELECT * FROM tbl_request
INNER JOIN tbl_barang ON tbl_request.id_barang = tbl_barang.id_barang
INNER JOIN user ON tbl_request.id_user = user.id_user WHERE id='$id'");
$data_request= mysqli_fetch_array($search_request);
$id_request = $data_request['id'];
$peminjam_request= $data_request['peminjam'];
$id_barang_request = $data_request['id_barang'];
$id_user_request = $data_request['id_user'];
$nama_barang = $data_request['nama_barang'];
$telpon_request = $data_request['telpon'];
$jml_barang_request = $data_request['jml_barang'];
$tgl_pinjam_request = $data_request['tgl_pinjam'];
$tgl_kembali_request = $data_request['tgl_kembali'];
if($id){
$query_search_barang = mysqli_query($CON,"SELECT * FROM tbl_barang WHERE id_barang = '$id_barang_request'");
$data_search_barang = mysqli_fetch_array($query_search_barang);
$stok_barang = $data_search_barang['stok_barang'] - $jml_barang_request;
if($data_search_barang){
$update_stok = mysqli_query($CON,"UPDATE tbl_barang SET stok_barang = '$stok_barang' WHERE id_barang = '$id_barang_request'");
if($update_stok){
if(mysqli_query($CON,"INSERT INTO tbl_pinjam (id_barang, id_user, peminjam, telpon, jml_barang, tgl_pinjam, tgl_kembali) VALUES ('$id_barang_request', '$id_user_request', '$peminjam_request', '$telpon_request', '$jml_barang_request', '$tgl_pinjam_request', '$tgl_kembali_request')")){
if(mysqli_query($CON,"DELETE FROM tbl_request WHERE id = '$id_request'")){
$konten = "Permintaan Peminjaman Barang Anda Telah di Terima. ".$jml_barang_request." buah dengan nama
```

```
barang: ".$nama_barang.". Username:
".$peminjam_request.". Silahkan ke
bagian Sarpras untuk mengampil barang";
if(mysqli_query($CON, "INSERT INTO
pemberitahuan (id_user, id_barang,
username, konten, status) VALUES
('$id_user_request',
'$id_barang_request',
'$peminjam_request', '$konten',
'terima')")){
echo
json_encode(array('message'=>'Berhasil
Menerima Permintaan.'));
}else{
echo
json_encode(array('message'=>'Berhasil
Menambah Pemberitahuan'));
echo "Gagal Menambah Pemberitahuan";
}else{
echo json_encode(array('message'=>'Gagal
Menghapus dari tbl_pinjam'));
}}else{
echo json_encode(array('message'=>'Gagal
Menambah ke tbl_pinjam'));
 echo "Gagal menambah ke tbl_pinjam";
}else{
echo json_encode(array('message'=>'Tidak
Bisa Update barang'));
 echo "Tidak bisa update data barang";
}}else{
echo json_encode(array('message'=>'Tidak
Bisa Mencari barang'));
}}
?>
```

Gambar 9. API - *proses pinjam*

Gambar 9 merupakan *API* yang berfungsi untuk menerima proses peminjaman yang dilakukan oleh *user* dan akan diterima oleh *admin*.

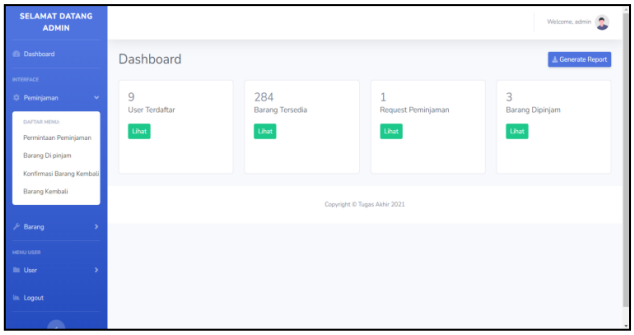


Gambar 10. Antarmuka Postman

Gambar 10 merupakan tampilan antarmuka Postman yang terdiri dari *URL*, *method*, *format data*, dan hasil dari API yang di *request*. dari gambar diatas dapat dilihat API yang akan diakses dengan url *http://localhost/TugasAkhir/read-barang-Android.php* dan *method* yang digunakan adalah *GET* dengan format data *JSON*, hasilnya status '200 OK' data berhasil ditampilkan sesuai dengan url yang dimasukan.

D. User interface website

User Interface atau tampilan antarmuka pengguna merupakan daya tarik utama dalam sebuah aplikasi semakin mudah digunakan aplikasi tentunya antarmuka akan menjadi lebih menarik [10]. Pada gambar dibawah merupakan antarmuka web memiliki beberapa menu halaman yaitu *dashboard*, *peminjaman*, *barang*, *user* dan *logout*.



Gambar 11. Antarmuka website halaman *admin*

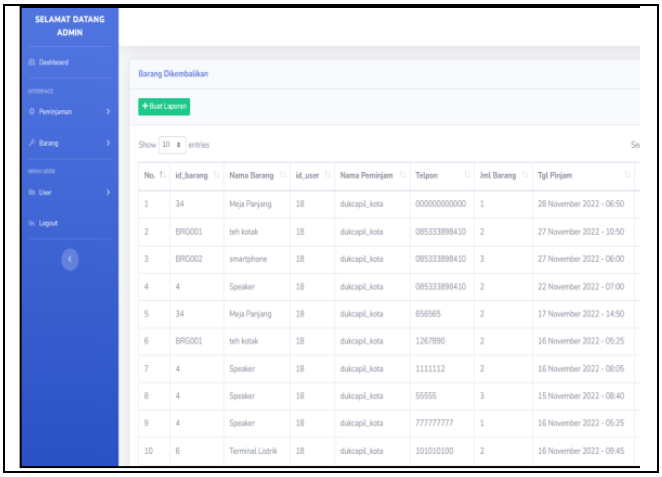
Gambar 11 menunjukkan antarmuka *admin* dari salah satu halaman *web*, yaitu halaman *admin* yang memiliki fitur untuk mengitung jumlah *user*, barang tersedia, *request* peminjaman, dan barang yang sedang dipinjam pada halaman ini juga terdapat fitur untuk melakukan *generate* laporan dalam bentuk *pdf*, *excel*, *csv*. Untuk *copy* laporan ini nantinya berfungsi sebagai data untuk mengevaluasi pelayanan peminjaman.

3. HASIL DAN PEMBAHASAN

Pada hasil dan pembahasan akan dibahas hasil dari perancangan aplikasi yang telah dibuat yang meliputi pembahasan tentang bagaimana cara sistem bekerja dan hasil pengujian serta analisis dari setiap komponen pendukung pada saat sistem tersebut beroperasi.

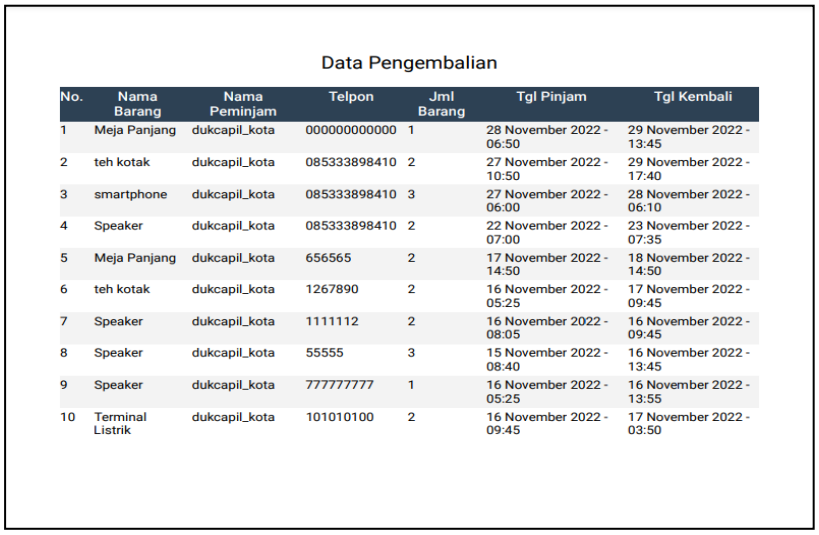
A. Hasil uji data transaksi

Uji data transaksi adalah pengujian terhadap cetak data dan data transaksi berdasarkan akun-akun yang tersedia pada *web*. Terdapat akun yang telah tersedia, dan dari masing-masing akun tersebut telah melakukan peminjaman barang.



Gambar 12. hasil transaksi Peminjaman

Gambar 12 menunjukkan akun hasil transaksi yang telah terjadi, terdapat beberapa akun yaitu *'dukcapil_kota'*.



Data Pengembalian

No.	Nama Barang	Nama Peminjam	Telpon	Jml Barang	Tgl Pinjam	Tgl Kembali
1	Meja Panjang	dukcapil_kota	000000000000	1	28 November 2022 - 06:50	29 November 2022 - 13:45
2	teh kotak	dukcapil_kota	085333898410	2	27 November 2022 - 10:50	29 November 2022 - 17:40
3	smartphone	dukcapil_kota	085333898410	3	27 November 2022 - 06:00	28 November 2022 - 06:10
4	Speaker	dukcapil_kota	085333898410	2	22 November 2022 - 07:00	23 November 2022 - 07:35
5	Meja Panjang	dukcapil_kota	656565	2	17 November 2022 - 14:50	18 November 2022 - 14:50
6	teh kotak	dukcapil_kota	1267890	2	16 November 2022 - 05:25	17 November 2022 - 09:45
7	Speaker	dukcapil_kota	1111112	2	16 November 2022 - 08:05	16 November 2022 - 09:45
8	Speaker	dukcapil_kota	55555	3	15 November 2022 - 08:40	16 November 2022 - 13:45
9	Speaker	dukcapil_kota	777777777	1	16 November 2022 - 05:25	16 November 2022 - 13:55
10	Terminal Listrik	dukcapil_kota	101010100	2	16 November 2022 - 09:45	17 November 2022 - 03:50

Gambar 13. Laporan Data

Akun tersebut sudah melakukan peminjaman barang dan pengembalian barang sesuai dengan tanggal yang sudah ditentukan. Gambar 13 merupakan data laporan tersebut berisi 7 atribut, antara lain, ‘No’, ‘nama barang’, ‘peminjam’, ‘telpon’, ‘jumlah barang’, ‘tgl pinjam’, dan ‘tgl kembali’ dengan total record 10 data dan disimpan dalam format *pdf.*

B. Hasil Implementasi kode program

Dalam tahapan pemanggilan API ini penulisan program menggunakan bahasa pemrograman kotlin dengan *IDE Android studio*. *Kotlin* merupakan bahasa pemrograman pengembangan dari java yang berbasis *java virtual machine (JVM). Kotlin* ini juga merupakan pemrograman *interoperabilitas* yang artinya dapat dikombinasikan dalam satu *project* dengan bahasa pemrograman java, bahasa ini digunakan untuk mengembangkan aplikasi berbasis desktop, web dan bahkan untuk backend.

Hasil Integrasi aplikasi dengan API

Integrasi aplikasi dengan API berfungsi untuk menghubungkan dua atau lebih platform yang berbeda, sehingga data dari dua platform tersebut saling terhubung. Proses integrasi dimulai dari penentuan *endpoint* API serta *method* yang akan digunakan untuk mengakses API. Selanjutnya endpoint tersebut dimasukkan ke dalam fungsi pada kode program sehingga dapat terhubung dengan API. Berikut dibawah ini merupakan beberapa hasil integrasi aplikasi dengan API yang ditampilkan dalam kode program. Beberapa kode program yang ditampilkan yaitu, kode program integrasi A*PI-neraca saldo, API-buku besar, API-laba rugi*, dan *API-penjualan*.

- Integrasi aplikasi dengan API menggunakan Retrofit2

```
package com.rangga.aplikasita

import okhttp3.OkHttpClient
import okhttp3.logging.HttpLoggingInterceptor
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

class ApiRetrofit {
    val endpoint : ApiEndpoint
    get() {
        val interceptor = HttpLoggingInterceptor()
        interceptor.level= HttpLoggingInterceptor.Level.BODY

        val client = OkHttpClient.Builder()

            .addInterceptor(interceptor)
            .build()

        val retrofit = Retrofit.Builder()
            .baseUrl("http://192.168.100.5/TugasAkhir/")
            .client( client )
            .addConverterFactory(GsonConverterFactory.create())
            .build()

        return retrofit.create(ApiEndpoint::class.java)
    }
}
```

IP address server atau localhost

Gambar 14. Kode program integrasi dengan API menggunakan *Retrofit2*

Gambar 14 menunjukkan kode program untuk proses integrasi dengan API menggunakan *Retrofit2*. Pada kode program terdapat *'baseUrl'* tempat menyimpan API yang digunakan untuk *request* data ke server. Jika Request berhasil, maka server akan mengirimkan Response berupa data dengan format *JSON*, dan respon tersebut dapat dilihat pada '*Okhttp'* berupa code *'200 OK'* jika berhasil mendapatkan data. Data tersebut akan disimpan ke dalam sebuah model kelas yang kemudian akan ditampilkan pada sesuai dengan permintaan.

- Integrasi API dengan API-*endpoint* pada aplikasi

```
package com.rangga.aplikasita

import retrofit2.Call
import retrofit2.http.Field
import retrofit2.http.FormUrlEncoded
import retrofit2.http.GET
import retrofit2.http.POST


interface ApiEndpoint {
    @GET("read-barang-android.php")
    fun data() : Call<BarangModel>

    @GET("read-permintaan-android.php")
    fun permintaan() : Call<PinjamModel>

    @GET("read-barang-dipinjam.php")
    fun barangdipinjam() : Call<PinjamModel>
    @FormUrlEncoded
    @POST("tambah-barang-android.php")
    fun create(
        @Field("nama_barang") nama_barang: String,
        @Field("stok_barang") stok_barang: String
    ) : Call<SubmitModel>
```

Gambar 15. Kode program memanggil seluruh API yang telah dibuat

Gambar 15 menunjukkan *class interface* API *Endpoint* kode program untuk proses pemanggilan API yang telah dibuat disertai dengan *method* digunakan kemudian membuat parameter penting yang dibutuhkan oleh API, terdapat beberapa *library* yang dipanggil yaitu *'call'* disini berfungsi untuk memanggil model yang akan digunakan oleh API, field berfungsi sebagai variabel yang akan diberikan nilai, '*FormUrlEncoded'* berfungsi sebagai bahwa API tersebut merupakan form dan akan diupload, *GET* merupakan *method* yang digunakan untuk mendapatkan data, dan *POST* merupakan *method* yang digunakan untuk mengupload nilai dari *field*.

- Pembuatan Data Model

```
package com.rangga.aplikasita

import java.io.Serializable

data class BarangModel (
    val result: List<Data>

    ){
    data class Data(val id_barang: String?,val nama_barang:
String?, val stok_barang: String?) : Serializable
}
```

Gambar 16. Kode program *'BarangModel'*

Gambar 16. Menunjukkan kode program untuk proses pembuatan *'BarangModel'*. Kode diatas berfungsi untuk menampung nilai data yang di *request* oleh pengguna seperti *'nama_barang'* dan *'stok_barang'*. Data tersebut disimpan dalam bentuk *JSON* kemudian ditampilkan pada halaman Data Barang.

- Integrasi aplikasi dengan API-penjualan

```
private fun getBarang(){

api.data().enqueue(object : Callback<BarangModel>{ override fun
onResponse(call: Call<BarangModel>, response:
Response<BarangModel>){ if(response.isSuccessful){
  val listData = response.body()!!.result
               barangAdapter.setData( listData )
  //   listData.forEach({
    //  Log.e("MainActivity", "barang ${it.nama_barang}")
       //             })
                }
            }
            override fun onFailure(call: Call<BarangModel>, t:
Throwable) {
                Log.e("MainActivity", t.toString())
            }
        })
    }
```

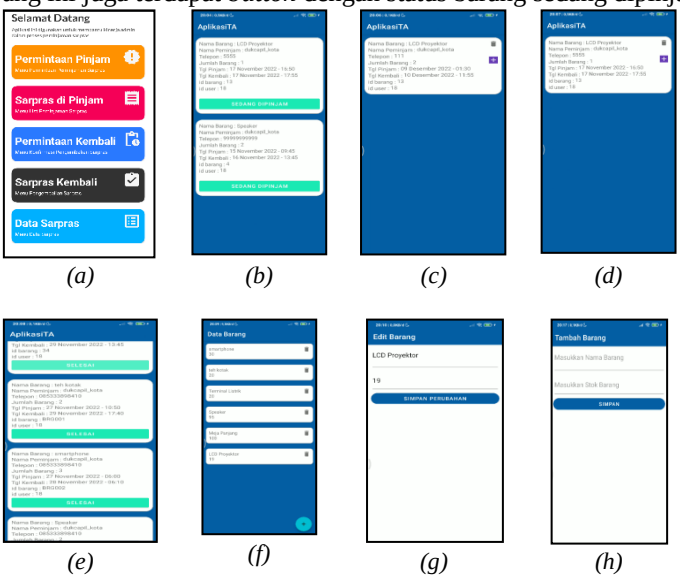
Gambar 17. Kode program mengirim Request data barang ke server

Pada gambar 17 merupakan kode program untuk mengirim *request* data barang ke server pada gambar diatas terdapat fungsi *GET* Barang yang berisi API. Data merupakan nama *endpoint* untuk mengakses API data barang kemudian terdapat fungsi jika berhasil maka data tersebut akan ditampilkan pada barangAdapter dan jika gagal maka akan dikirimkan *result* ke *"MainActivity".*

C. Hasil Tampilan Antarmuka Pada Android

Berturut-turut penjelasan gambar sebagai berikut : gambar 18 menunjukkan halaman dashboard yang berisi informasi terkait Permintaan pinjam, barang dipinjam, permintaan kembali, barang kembali, dan data barang. Gambar 19 menunjukkan halaman barang dipinjam pada halaman ini berisi tentang informasi peminjam

seperti *nama barang, nama peminjam, telepon, jumlah barang, tgl pinjam, tgl kembali, id barang* dan *id user*. Pada halaman barang ini juga terdapat *button* dengan status barang sedang dipinjam.



(a) (b) (c) (d) (e) (f) (g) (h)

Gambar 18. (a) Dashoard, (b) Barang Dipinjam, (c) Permintaan, (d) Permintaan, (e) Barang Kembali, (f) Data Barang (g) Edit Barang pinjam, (h) Tambah Barang Kembali

Gambar 18(c) menunjukkan halaman permintaan pinjam pada halaman ini berisi tentang informasi peminjam seperti *nama barang, nama peminjam, telepon, jumlah barang, tgl pinjam, tgl kembali, id barang* dan *id user*. Pada halaman ini juga terdapat tombol untuk menerima peminjaman dan juga untuk menolak peminjaman. Gambar 18(d) menunjukkan halaman permintaan kembali pada halaman ini berisi tentang informasi peminjam seperti *nama barang, nama peminjam, telepon, jumlah barang, tgl pinjam, tgl kembali, id barang* dan *id user*. Pada halaman ini juga terdapat button untuk menerima peminjaman dan juga terdapat button untuk menolak pengembalian. Gambar 18(e) menunjukkan halaman barang kembali pada halaman ini berisi tentang informasi peminjam seperti *nama barang, nama peminjam, telepon, jumlah barang, tgl pinjam, tgl kembali, id barang* dan *id user*. Pada halaman barang ini juga terdapat button dengan status selesai. Gambar 18(f) menunjukkan halaman data barang pada halaman ini menunjukan informasi terkait *nama barang* dan *stok barang*, pada halaman ini juga terdapat tombol untuk menambah dan menghapus barang. Data barang tersebut dapat di ubah dengan cara menekan pada data barang yang ingin di ubah. Gambar 18(g) menunjukkan halaman edit barang terdapat nilai dari nama barang dan stok barang yang dapat di edit pada halaman ini juga terdapat button simpan perubahan berfungsi untuk menyimpan data yang sudah diubah. Gambar 18(h) menunjukkan halaman tambah barang terdapat kolom nama barang dan stok barang yang dapat diisi. Pada halaman ini juga terdapat tombol simpan yang berfungsi untuk menyimpan data yang telah diisi.

4. KESIMPULAN

Berdasarkan hasil analisis dan pengujian sistem dihasilkan kesimpulan bahwa metode REST API telah dapat mengintegrasikan data antara aplikasi basis web dengan basis *mobile*. Pengintegrasian ini juga menggunakan *library Retrofit2* dan bahasa pemrograman kotlin. Dari hasil yang didapat keseluruhan pemanggilan API berjalan dengan baik dan didapat data mengenai *peminjaman barang* dan *user* yang melakukan peminjaman. Pada pengujian sistem, menggunakan metode testing yaitu *alpha* dan *beta testing*. *Alpha-testing* ditujukan untuk pengguna admin dengan melakukan pengujian langsung terhadap sistem yang telah diimplementasikan dengan tujuan untuk dapat menemukan apakah terdapat kesalahan pada sistem, sehingga nantinya sistem yang dibangun dapat berjalan sesuai dengan harapan pengguna. Kemudian *beta-testing* akan dilakukan dengan menggunakan pendekatan *MOS (mean opinion score)* dimana ini dapat digunakan untuk mengukur berjalannya fungsionalitas sistem yang diharapkan [12][13]. Dalam penelitian disebarkan kuisioner kepada 30 responden untuk menggunakan aplikasi. Proses pengujian dilakukan dengan menunjukan bagaimana sistem berjalan kepada responden, selanjutnya responden akan diminta untuk menjawab beberapa pertanyaan berdasarkan hasil pengujian aplikasi yang sudah dilakukan. Tujuan dari pengujian ini adalah untuk mengetahui bagaimana kualitas dari sistem yang dibangun dari sisi user. Pengujian yang dilakukan berupa

alpha-testing untuk menguji sistem aplikasi yang sedang berjalan dan pengujian beta-testing untuk kepuasan pengguna. Didapatkan persentase hasil pengujian alpha dan beta 100% dan 88,60%.

5. DAFTAR PUSTAKA

[1] Wijaya, Yahya Dwi, and Muna Wardah Astuti. "Sistem informasi penjualan tiket wisata berbasis web menggunakan metode waterfall." Prosiding Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK). Vol. 2. No. 1. 2019.

[2] Alfianti Oktavia, Chaulina, Rosandi Fila Setiawan, and Andrew Christianto. "Perancangan Aplikasi Augmented Reality Untuk Pengenalan Ruangan Menggunakan Marker 3D Objects Tracking." Jurnal Ilmiah Teknologi Informasi 13.1 (2019): 53-60.

[3] L. Li and W. Chou, "Design and Describe REST API without Violating REST: A Petri Net Based Approach," *2011 IEEE International Conference on Web Services*, 2011, pp. 508-515, doi: 10.1109/ICWS.2011.54.

[4] Surendra, Martinus Raditia Sigit. "Implementasi PHP Web Service Sebagai Penyedia Data Aplikasi Mobile." Ultimatics: Jurnal Teknik Informatika 6.2 (2014): 85-93.

[5] Ong SP, Cholia S, Jain A, Brafman M, Gunter D, Ceder G, et al. The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. ComPUT Mater Sci. 2015;97.

[6] Arcuri A. RESTful API automated test case generation with Evomaster. ACM Trans Softw Eng Methodol. 2019;28(1).

[7] Singh A, Jeyanthi N. MVP Architecture Model with Single Endpoint Access for Displaying COVID 19 Patients Information Dynamically. In: Proceedings - 2020 12th International Conference on Computational Intelligence and Communication Networks, CICN 2020. 2020.

[8] Lumba, Ester. "Pertukaran Data Pada Aplikasi Android Menggunakan Java Script Object Notation (Json) Dan Rest Api Dengan Retrofit 2." Prosiding Snast (2021): 118-127.

[9] Ghiffary, Muhammad Nauval El. Analisis komponen desain layout, warna, dan kontrol pada antarmuka pengguna aplikasi mobile berdasarkan kemudahan penggunaan (studi kasus: aplikasi olride). Diss. Institut Teknologi Sepuluh Nopember, 2018.

[10] Team, G. D. (2016). Android Developer Fundamental Course Learn to Develop Android Application: Concept Reference.

[11] H. Qodim, Busro and R. Rahim, "Islamic Calendar: Prototype of Hijri Calendar Application using Rapid Application Development Method," 2019 7th International Conference on Cyber and IT Service Management (CITSM), Jakarta, Indonesia, 2019, pp. 1-4, https://doi.org/10.1109/CITSM47753.2019.8965410

[12] B. Naderi and S. Möller, "Transformation of Mean Opinion Scores to Avoid Misleading of Ranked Based Statistical Techniques," 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), Athlone, Ireland, 2020, pp. 1-4, https://doi.org/10.1109/QoMEX48832.2020.9123078

[13] Giri Wahyu Wiriasto, Misbahuddin, Muhamad Syamsu Iqbal, Djul Fikry Budiman, Sjamsjiar Rachman, Lalu Ahmad Syamsul Irfan Akbar. "Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science) (MIMSE-I-C-2022,Atlantis Press, 367-378, 2352-538X, 2022, https://doi.org/10.2991/978-94-6463-084-8_31 .

[14] Bayu Hermawan, Giri Wahyu Wiriasto, L.Syamsul Irfan A, "*AKSYAA* - ACCOUNTING INFORMATION SYSTEM (AIS) : MOBILE APPLICATION WITH RESTFUL API", Electrical engineering Departement – University of Mataram (submitted draft journal), 2023.

[15] Ong, S. P., Cholia, S., Jain, A., Brafman, M., Gunter, D., Ceder, G., & Persson, K. A. (2015). The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Computational Materials Science*, *97*. https://doi.org/10.1016/j.commatsci.2014.10.037