

Implementasi *Rest API* Permintaan Data Sektoral Dinas Komunikasi Informatika Statistik dan Persandian Provinsi Sulawesi Selatan berbasis *Android*

¹Rudy Donny Liklikwatil, ²Nur Salman, ³Sunardi

⁴Muhammad Satrya Tangdilingtin, ⁵Muhammad Nur Ichsan B
Universitas Dipa Makassar

Jalan Perintis Kemerdekaan KM. 9 Makassar, Telp. (0411) 587194 – Fax. (0411) 588284

e-mail: ¹rudy.donny@dipanegara.ac.id, ²nursalman.halim@dipanegara.ac.id,

³sunardi@dipanegara.ac.id, ⁴satrya.nugraha12@gmail.com, ⁵muhnurichsan2499@gmail.com

Abstrak

Sistem permintaan data sektoral pada Dinas Komunikasi Informatika Statistik dan Persandian masih dilakukan dengan membawa surat permohonan permintaan data secara langsung yang membutuhkan waktu dalam menunggu hasil validasi dan persetujuan terhadap jenis data yang diminta. Perancangan aplikasi permintaan data berbasis *android* dapat membantu dalam proses permintaan data menjadi lebih cepat dan akurat untuk pengguna data karena semua proses dapat dilakukan secara *online*. Dalam perancangan aplikasi permintaan data ini menggunakan metode *Rest API* yang berfungsi untuk menghubungkan *platform android* dan *website* dalam pertukaran data. Metode perancangan aplikasi yang digunakan adalah *Unified Modeling Language* dengan metode pengujian *Black Box*. Berdasarkan hasil pengujian, aplikasi sudah berjalan dengan baik sesuai tujuan perancangan sehingga dapat diimplementasikan pada Dinas Komunikasi Informatika Statistik dan Persandian.

Kata Kunci : **Permintaan Data, Rest Api, Android.**

Abstract

The sectoral data request system at the Statistical and Encoding Communication and Informatics Service is still carried out by bringing a request letter for direct data requests which requires time to wait for the results of validation and approval of the type of data requested. The design of an Android-based data request application can help in the process of requesting data to be faster and more accurate for data users because all processes can be done online. In designing this data request application, the Rest API method is used which functions to connect the android platform and the website in exchanging data. The application design method used is the Unified Modeling Language with the Black Box testing method. Based on the test results, the application has been running well according to the design objectives so that it can be implemented at the Statistics and Encoding Information and Communications Office.

Keywords: Data Request, Rest API, Android.

1. Pendahuluan

Dinas Komunikasi, Informatika, Statistik dan Persandian (DISKOMINFO-SP) Provinsi Sulawesi Selatan adalah badan yang membantu gubernur dalam penyelenggaraan pemerintahan bidang Komunikasi, Informatika, Statistik, dan Persandian. Terdapat beberapa bidang dalam DISKOMINFO-SP, salah satunya adalah Bidang Statistik Sektoral yang bertugas untuk membantu Kepala Dinas dalam mengkoordinasikan, merumuskan dan melaksanakan kebijakan teknis statistik sektoral. Data sektoral atau Statistik Sektoral adalah statistik yang pemanfaatannya ditujukan untuk memenuhi kebutuhan instansi pemerintah tertentu dalam rangka penyelenggaraan tugas-tugas pemerintah dan tugas pembangunan yang merupakan tugas pokok instansi pemerintah yang bersangkutan. Manfaat lain dengan adanya data sektoral adalah data tersebut dapat digunakan sebagai bahan penelitian.

Penggunaan data sektoral sebagai bahan penelitian, masyarakat atau instansi harus melakukan permintaan data sektoral melalui proses pengiriman surat permohonan permintaan data sektoral secara

langsung dan menunggu persetujuan dari pihak DISKOMINFO-SP sesuai dengan jenis data yang mereka inginkan. Proses permintaan data sektoral tersebut cukup menyita waktu dikarenakan harus melalui proses pengiriman surat permohonan permintaan data, validasi terhadap jenis data yang diminta, hingga persetujuan permohonan permintaan data tersebut.

Perubahan budaya kerja dari *offline* menjadi *online* dengan memanfaatkan teknologi digital sudah waktunya untuk diimplementasikan terkait masalah yang ditimbulkan dalam proses permintaan data sektoral tersebut. Berdasarkan permasalahan tersebut sehingga dirancang sebuah aplikasi berbasis *website* dan *android* yang akan digunakan dalam proses permintaan data sektoral secara *online* sehingga proses tersebut tidak dilakukan lagi dengan harus mendatangi kantor DISKOMINFO-SP secara langsung.

Pada Penelitian ini aplikasi di kembangkan menggunakan *Rest API* yang berfungsi sebagai penghubung antara *website* dan *android*, dimana aplikasi *website* akan bekerja di sisi *server* yang digunakan untuk memvalidasi dan mengirim data sektoral yang diminta oleh pemohon, sedang aplikasi *android* akan bekerja di sisi *client* yang mengirim permohonan permintaan data dan menerima data, di mana *website* dan *android* merupakan *platform* yang berbeda dalam mengakses basisdata yang sama. Metode *Rest* juga memiliki keunggulan dalam hal kecepatan *transfer* data dan performa daripada metode lainnya. *Rest API* juga membantu developer dalam pengembangan aplikasi selanjutnya dan mengintegrasikan dengan *platform* berbeda dengan bahasa yang berbeda.

Yosef Koko Kurniawan, Yetli Oslan, Harianto Kristanto [1] dalam penelitiannya tentang Implementasi *Rest-Api* untuk Portal Akademik UKDW berbasis *Android*. Pada penelitian ini dibuat portal akademik UKDW berbasis *mobile*. Aplikasi harus terkoneksi dengan basis data SITMPT UKDW agar dapat terintegrasi dengan sistem informasi yang lain. Untuk itu sistem yang dibangun membutuhkan API yang menjadi sarana penghubung antara aplikasi dengan basisdata. Dalam penelitian ini API dibangun dengan konsep *Rest* dan menerapkan otorisasi milik *Amazon Web Service* untuk sistem keamanannya. Hasil dari perancangan adalah sebuah aplikasi Portal UKDW berbasis *Android* yang berjalan pada sistem operasi *Android* mulai dari versi 2.2 (*Froyo*) sampai 4.2 (*Jelly Bean*). Selain itu perancangan juga menghasilkan sebuah *Rest API* lengkap dengan sistem keamanannya.

I Gede Tika Permana [2] melakukan penelitian tentang Pengembangan Sistem Presensi Berbasis Lokasi menggunakan *Geofence Wifi* Dan *Rest Api* pada Fakultas Ilmu Komputer Universitas Brawijaya. Penelitian ini berfokus pada fungsi *Rest API* untuk mendapatkan rekapitulasi dari hasil presensi civitas dalam pengajuan cuti dan lembur.

Shofiyullah Al Ghozaly [3] dalam penelitiannya yang berjudul "Implementasi *Rest API* Pada Pusat Informasi Mahasiswa Universitas Teknologi Yogyakarta". Penelitian ini terfokus pada penggunaan metode *Rest API*, fungsi *Rest API* yang digunakan dalam penelitian ini digunakan dalam pengembangan oleh pengembang lainnya untuk peningkatan aplikasi yang lebih kompleks nantinya.

2. Metode Penelitian

2.1. Metod Pengumpulan data

Metode pengumpulan data yang digunakan adalah metode wawancara yang dilakukan langsung terhadap staf pengelola data sektoral untuk memperoleh prosedur permintaan data sektoral. Penelitian pustaka juga dilakukan pada penelitian ini untuk mendapatkan dokumen resmi yang berasal dari data sektoral berupa dokumen *external* yang berisi tentang informasi jenis dan format dari data sektoral yang tersedia pada DISKOMINFO-SP.

2.2. Jenis Data

Adapun data yang akan digunakan dalam penelitian ini adalah :

- a. Data Primer, yaitu data yang diperoleh melalui wawancara langsung dengan staf pengelola data sektoral. Dimana data yang di maksud adalah data prosedural, yaitu data yang akan digunakan untuk analisis sistem berjalan, berisi data tentang bagaimana proses dan prosedur permintaan data sektoral yang dilakukan saat ini.
- b. Data Sekunder, yaitu data yang diperoleh dari staf pengelola data sektoral dalam bentuk laporan atau dokumen-dokumen yang berhubungan dengan penelitian ini. Salah satu contoh jenis data yang diperoleh adalah data sektoral politik, hukum, keamanan dan ketertiban masyarakat, berupa: data Badan Kepegawaian Daerah, data Badan Penghubung Daerah, data Badan Perancangan Pembangunan, data Biro Administrasi Pimpinan, data Biro Hukum dan HAM, data Biro Organisasi dan Tata Laksana, data Biro Pemerintahan dan Otonomi Daerah, data Dinas Energi dan Sumber Daya Mineral, data Dinas Kependudukan dan Pencatatan Sipil, data Dinas Komunikasi Informatika, Statistik dan Persandian, data Dinas Pemberdayaan Perempuan, Anak Pengendalian Penduduk dan

Keluarga Berencana, data Penelitian dan Pembangunan Daerah, data Satuan Polisi Pamong Praja dan data Sekretariat DPRD.

2.3. Tahapan Penelitian

Adapun tahapan penelitian yang dilakukan yaitu :

- a. Pengumpulan data, dengan melakukan wawancara untuk mengetahui prosedur dari proses permintaan dan pengambilan data sektoral secara langsung pada DISKOMINFO-SP.
- b. Analisis sistem berjalan, pada permohonan permintaan data sektoral di DISKOMINFO-SP ini yang masih menggunakan cara manual yaitu dengan pengantaran langsung surat permohonan permintaan data ke DISKOMINFO-SP, datang kembali untuk mengecek validasi dan persetujuan permintaan data tersebut hingga data sektoral yang diminta dapat diperoleh. Analisis ini bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang di hadapi.
- c. Perancangan sistem, dilakukan untuk mendapatkan pemecahan masalah dengan menggunakan perangkat pemodelan sistem seperti UML, meliputi : *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*. *Unified Modeling Language (UML)* adalah notasi grafis untuk membantu pendeskripsian dan mendesain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.
- d. Pengkodean atau pembuatan program aplikasi, dilakukan untuk membangun aplikasi *web* pada sisi *server* sebagai penyedia data sektoral dan aplikasi *android* pada sisi *client* sebagai pemohon data sektoral.
- e. Pengujian sistem, dilakukan dengan menggunakan metode *black box* terhadap aplikasi yang dihasilkan. Metode ini akan menguji kerja *Rest API* terhadap aplikasi yang dirancang secara lengkap dan terstruktur sesuai dengan spesifikasi dan juga menilai apakah tiap-tiap fungsi dari aplikasi telah berjalan sesuai dengan fungsinya masing-masing.
- f. Implementasi sistem, pada tahap ini sistem *Rest API* yang telah di rancang pada sisi *server* yang akan digunakan pada sisi *client* untuk melakukan pertukaran data diterapkan, sehingga dapat mencapai tujuan yang diinginkan.

2.4. Metode Pengujian

Metode pengujian perangkat lunak menggunakan *blackbox testing* untuk menguji terjadinya kesalahan pada *interface*, basisdata, performansi, inisialisasi dan terminasi serta apakah setiap fungsi dari *interface* yang dirancang sudah dinyatakan *valid*.

2.5. Tinjauan Pustaka

2.5.1. API (Application Programming Interface)

Muhammad Amril Sya'ban [4] berpendapat bahwa: *API (Application Programming Interface)* memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developer* untuk membuat aplikasi. Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*. Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, *library* dan *web*.

Application Programming Interface (API) berisi tentang dokumen antarmuka, fungsi, kelas, struktur, dan lain-lain. Untuk membangun perangkat lunak menggunakan *API* ini semakin memudahkan programmer membongkar perangkat lunak, kemudian dapat mengembangkan atau mengintegrasikan dengan perangkat lunak lainnya. *API* bisa dikatakan sebagai penghubung dengan aplikasi-aplikasi lain yang boleh digunakan oleh pemrogram menggunakan fungsi sistem. Prosesnya bisa diatur melalui sistem operasi. Keuntungan dari *API* ini yaitu memungkinkan aplikasi lain dapat saling berhubungan dan berinteraksi. *API* juga merupakan sekumpulan perintah, fungsi dan protokol. Pemrogram saat ini dapat digunakan untuk membangun perangkat lunak untuk sistem operasi. *API* memungkinkan pemrogram berinteraksi menggunakan fungsi standar dengan sistem operasi lainnya.

2.5.2. REST (Representational State Transfer)

Rully Yulian MF [5] berpendapat bahwa : *Rest* merupakan sebuah arsitektur *web services* yang berbasis *http verbs* dalam mengolah data. Pada bentuk sebelumnya kita sudah mengenal arsitektur *web service* lainnya seperti *XML Web Service* yang berbasis *SOAP*. *Http verbs* yang digunakan pada *Rest* dapat berupa *GET*, *POST*, *PUT*, *DELETE* dan lainnya. Berikut asosiasi masing-masing *http verbs* dengan perintah *SQL*: *GET* : *Select*, *POST* : *Insert*, *PUT* : *Update*, *DELETE* : *Delete*.

Data yang dikembalikan dari *Rest* berupa data dalam format *Json* atau *XML*, namun pada umumnya menggunakan *Json* karena dari sisi penulisan dan pembacaannya lebih sederhana selain dari sisi ukuran datanya. Setiap data yang dikembangkan dari *Rest* memiliki *http status code* yang disesuaikan dengan proses yang terjadi di dalam setiap *http verbs*. *Status code* dapat memberikan informasi sukses atau *error*. Informasi status *http code* pada *client* misalnya 200(*OK*), 201(*Created*), 404(*Not Found*), 401(*Unauthorized*), 500(*Internal Server Error*), atau yang lainnya tergantung dari hasil pengolahan data. *Status code* tersebut diberikan oleh *object response*. *Object response* juga bertanggung jawab untuk memberikan *output* data pada *body*.

Rest service dibangun dengan tujuan untuk menjembatani penggunaan data antar *platform* dan antar *device* yang berbeda sehingga bisa saling bertukar, karena data yang dipertukarkan disimpan dalam format standar yang dapat diproses oleh berbagai *platform*.

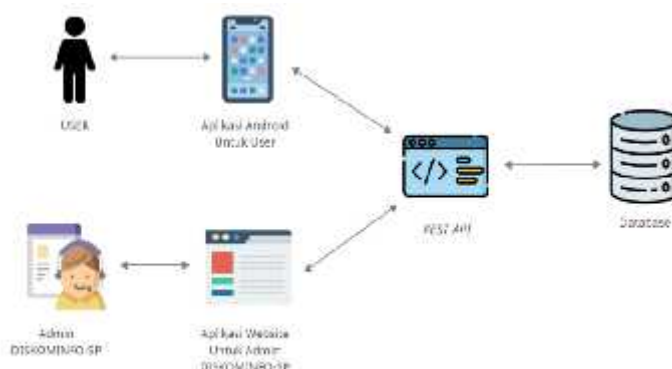
Dari penjelasan diatas, berikut beberapa keunggulan yang dimiliki oleh *Rest* yaitu :

- a. *Rest* menyediakan infrastruktur yang bagus dalam proses *caching* melalui metode *http get*. Hal ini berguna untuk meningkatkan performa jika informasi tidak diubah dan tidak dinamis.
- b. *Rest* membagi perspektif *server* dan *client* melalui interaksi yang menggunakan *http*.
- c. *Rest* dapat mengembalikan respon dalam berbagai macam format dan sesuai dengan permintaan *client*.
- d. *Rest* dapat disempurnakan menggunakan bahasa pemrograman manapun selama bahasa tersebut dapat membuat permintaan berbasis *web* melalui *http*.
- e. *Rest* baik untuk digunakan pada aplikasi perangkat bergerak.

3. Hasil dan Pembahasan

3.1. Arsitektur Aplikasi

Adapun Arsitektur aplikasi yang akan dirancang ditunjukkan pada gambar 1.

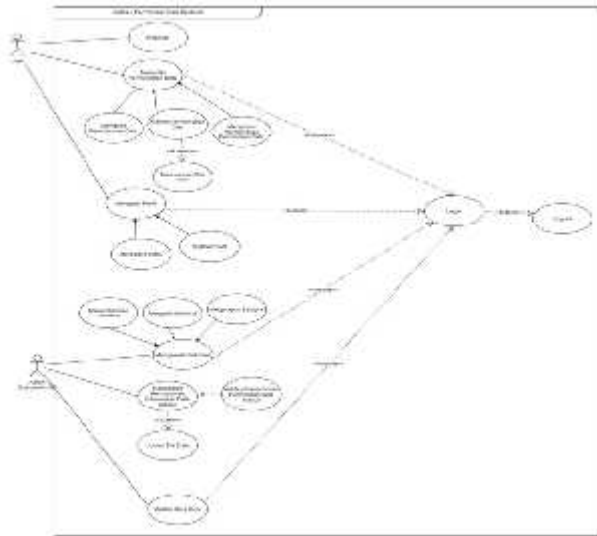


Gambar 1. Arsitektur Aplikasi

Aplikasi yang dirancang menggunakan *Rest API* yang berfungsi sebagai penghubung antara *website* dan *android*, dimana aplikasi *website* akan bekerja di sisi *server* (*admin DISKOMINFO-SP*) yang digunakan untuk memvalidasi dan mengirim data sektoral yang diminta oleh pemohon (*user*), sedang aplikasi *android* akan bekerja di sisi *client* (*user*) yang mengirim permohonan permintaan data dan menerima data yang telah divalidasi dan disetujui oleh *DISKOMINFO-SP*.

3.2. Use Case Diagram

Gambar 2 berikut ini memberikan pemodelan interaksi antara aktor dengan sistem. Terdapat dua aktor yaitu *admin DISKOMINFO-SP* dan *user* sebagai pihak pemohon permintaan data sektoral.



Gambar 2. Use Case Diagram

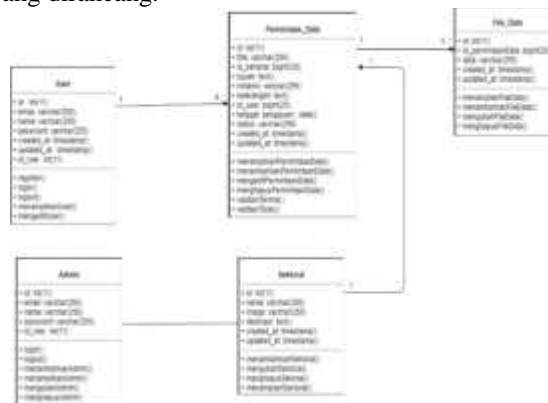
Adapun spesifikasi diagram *Use Case* diatas yang menjelaskan fungsi aktor admin dan *user* dalam sistem ditunjukkan pada tabel 1 berikut ini:

Tabel 1. Spesifikasi aktor dalam *Use Case Diagram*

Aktor	Deskripsi
admin	Menambah, mengedit, menghapus data Sektoral
	Memvalidasi data masuk dan mengupload data
	Melihat data user
user	Registrasi, login, melihat dan mengedit profil
	Menambah, mengedit, menghapus, mendownload permintaan data

3.3. Class Diagram

Class Diagram merupakan hubungan antar kelas dan penjelasan terinci tiap-tiap kelas di dalam model desain dari sistem yang dirancang.



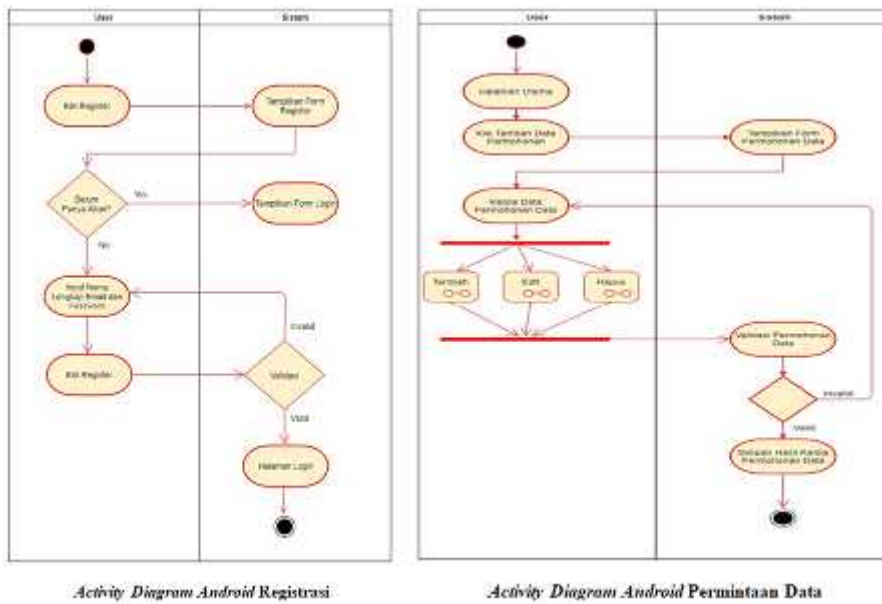
Gambar 2. Class Diagram

Pada *class diagram* sistem ini, memperlihatkan hubungan antara tabel-tabel yang saling berelasi, dimana *admin* terhubung pada semua *class* kecuali *class user* dan mempunyai hak akses untuk mengelola semua *class* kecuali *class user*. Kemudian *user* terhubung ke semua *class* kecuali *class sektoral* dan mempunyai hak akses ke semua *class* kecuali *class sektoral* dan *class admin*.

3.4. Activity Diagram

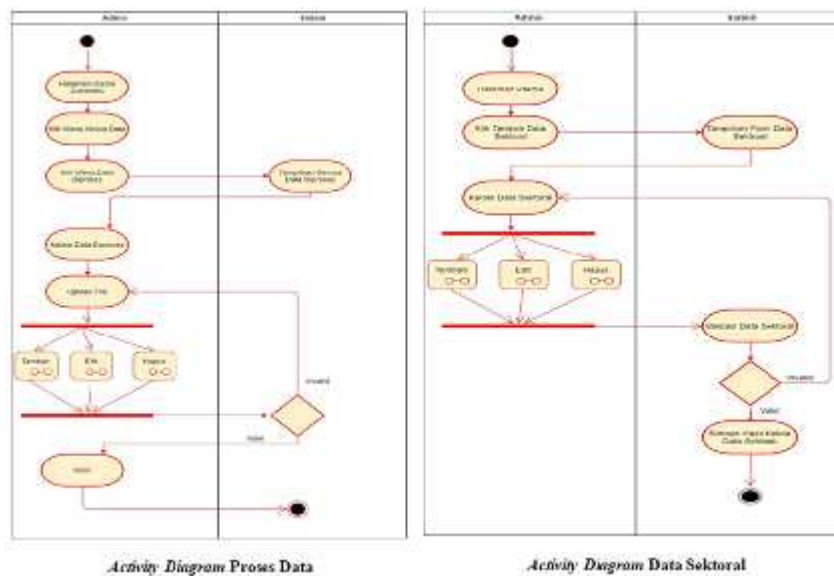
Gambar 3 dan 4 berikut ini menunjukkan aliran logika yang terjadi dalam proses registrasi dan permintaan data oleh user serta proses validasi dan kelola data sektoral oleh admin DISKOMINFO-SP.

a. Activity Diagram user



Gambar 3. Activity Diagram user

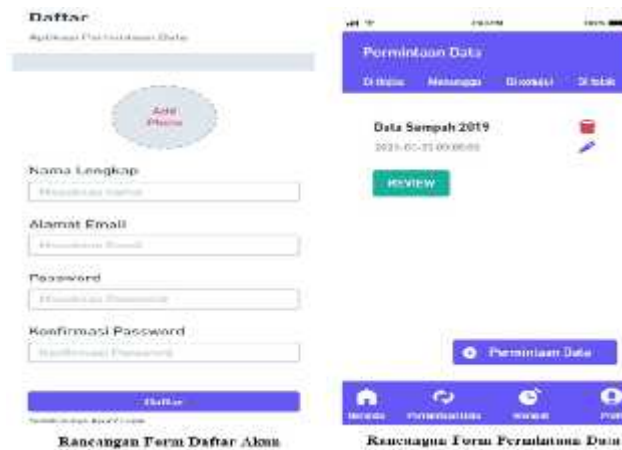
b. Activity Diagram admin



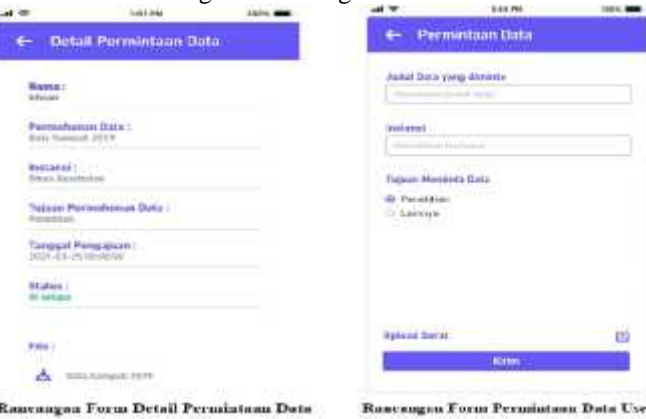
Gambar 4. Activity Diagram Admin

3.5. Rancangan interface Aplikasi

a. Rancangan form untuk client (user pemohon permintaan data sektoral)

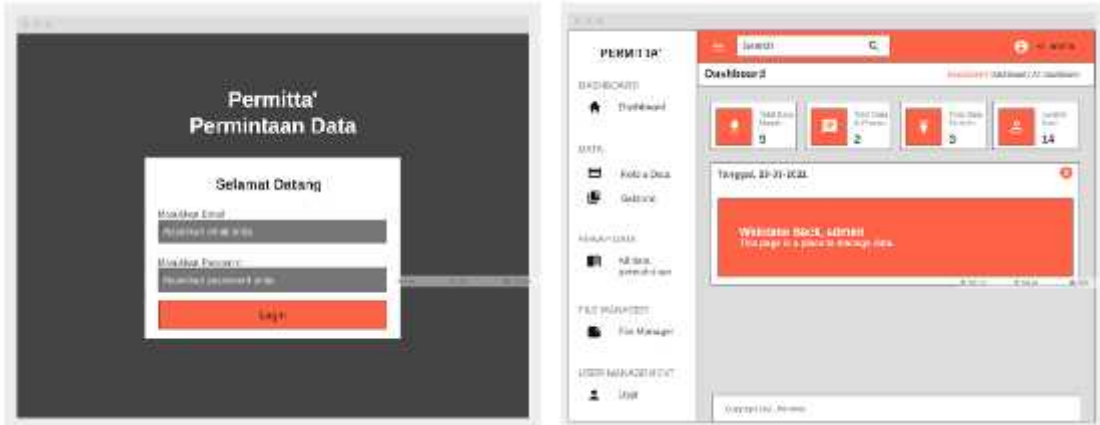


Rancangan Form Registrasi dan Permintaan Data

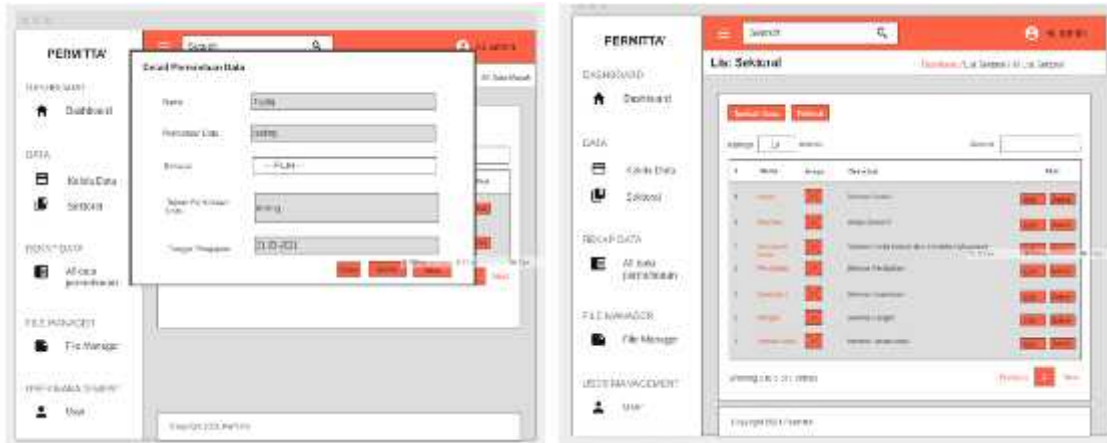


Rancangan Form Detail Permintaan Data dan User

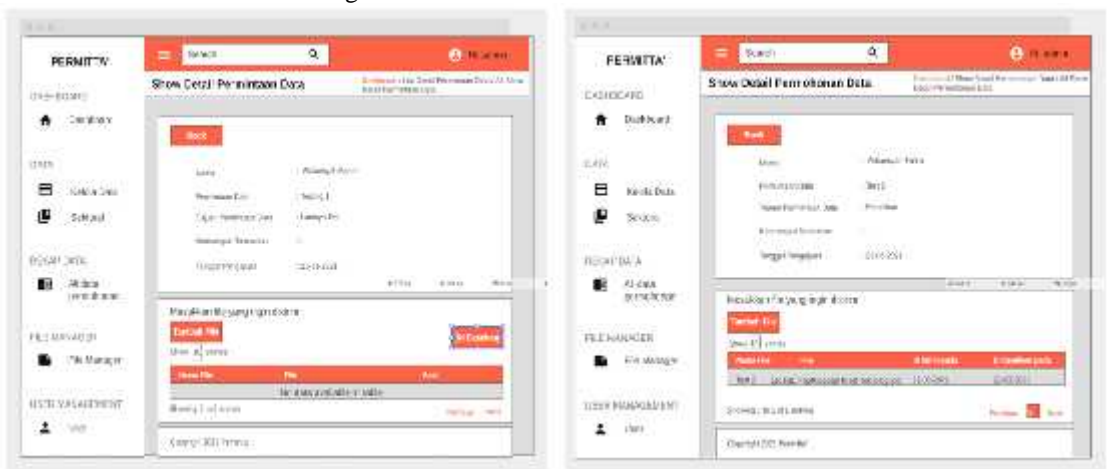
b. Rancangan *website* Pengelolaan Data Sektoral (*admin* penyedia data)



Gambar 7. Rancangan Form Login admin dan tampilan form dashboard



Gambar 7. Rancangan Form Detail Permintaan Data dan List Data Sektoral



Gambar 8. Rancangan Form Detail Permintaan dan Permohonan Data

c. Rancangan Rest API

Rancangan Request dan Response Rest API memuat aturan permintaan sumber data dan hasil kembalian dari layanan. Adapun rancangan request dan response Rest API yang dibuat adalah sebagai berikut:

Tabel 2. Rancangan request dan response Rest API kode Register

Method	POST
URL	http://permitta.nekoding.com/api/register
Headers	-
Body	- Name - Email - image - Password
Keterangan	Pada request api ini memerlukan body yaitu name,email,dan password dan akan mengembalikan response berupa data user dan access token.
Response	{ "data": { "name": "ichsan", "email": "ichsan24@gmail.com", "id_role": 2, "image": null, "created_at": "2021-04-03T05:38:52.000000Z", "updated_at": "2021-04-03T05:38:52.000000Z", "id": 35 }, "token":

<pre>"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9wZXJtaXR0 YS5uZWVtZGlucy5jb21lL2FwaVwvcmlnaXN0ZXIiLCJpYXQiOiJlMTc0MjgzMzIsImZl 5iZiI6MTYxNzQyODMzMiwianRpIjoiMzBXY0hkdUtlQnJpZjRlYiIsInN1YiI6MzUsIn BydiI6Ijg3ZTBhZjZlZjlmZDE1ODEyZmRlYzk3MTUzYTE0ZTBiMDQ3NTQ2YWEifQ. CmS1zAO-MM4T3GP1VtHbHNSfchALVISK3P9vivD489s"</pre>
--

Tabel 3. Rancangan *request* dan *response Rest API* Sektoral

Method	GET
URL	http://permitta.nekoding.com/api/sektorals
Headers	Authorization
Body	-
Keterangan	Pada request api ini memerlukan headers authorization access token untuk mendapatkan list dari data sektorals
Response	<pre>[{ "id": 1, "name": "Sosial", "image": "http://permitta.nekoding.com/uploads/image/sektoral/1615694991sosial.png", "deskripsi": "Sektoral Sosial", "created_at": "2021-03-14 04:09:51", "updated_at": "2021-03-14 04:09:51" }, { },]</pre>

Tabel 4. Rancangan *request* dan *response Rest API* permintaan data

Method	GET
URL	http://permitta.nekoding.com/api/permintaan-data
Headers	Authorization
Body	-
Keterangan	Pada request api ini memerlukan headers authorization access token untuk mendapatkan list dari data permintaan data
Response	<pre>[{ "id": "a84d0587-817e-40be-8aff-d9c600ec6364", "title": "Data Lingkungan 2019", "id_sektoral": null, "tujuan": "Penelitian", "instansi": "Universitas Dipa Makassar", "keterangan": null, "surat": "1617430593ichsan-Permintaan Data Sektoral.pdf", "id_user": 35, "tanggal_pengajuan": "2021-04-03 00:00:00", "status": "Di tinjau", "created_at": "2021-04-03 06:16:33", "updated_at": "2021-04-03 06:16:33", "file": "1617430593ichsan-Permintaan Data Sektoral.pdf", "data": [] }, { },]</pre>

Tabel 5. Rancangan *request* dan *response Rest API* melihat detail permintaan data

Method	GET
---------------	-----

URL	http://permitta.nekoding.com/api/permintaan-data/show/{id}
Headers	Authorization
Body	-
Keterangan	Pada request api ini memerlukan headers authorization access token untuk mendapatkan data detail permintaan data berdasarkan id dari permintaan data.
Response	<pre>{ "id": "a84d0587-817e-40be-8aff-d9c600ec6364", "title": "Data Lingkungan 2019", "id_sektoral": null, "tujuan": "Penelitian", "instansi": "Universitas Dipa Makassar", "keterangan": null, "surat": "1617430593ichsan-Permintaan Data Sektoral.pdf", "id_user": 35, "tanggal_pengajuan": "2021-04-03 00:00:00", "status": "Di tinjau", "created_at": "2021-04-03 06:16:33", "updated_at": "2021-04-03 06:16:33", "data": [] }</pre>

Tabel 6. Rancangan *request* dan *response Rest API* menghapus permintaan data

Method	DELETE
URL	http://permitta.nekoding.com/api/permintaan-data/delete/{id}
Headers	Authorization
Body	-
Keterangan	Pada request api ini memerlukan headers authorization access token untuk menghapus permintaan data berdasarkan id.
Response	<pre>{ "message": "Berhasil di delete" }</pre>

Tabel 7. Rancangan *request* dan *response Rest API* mengubah permintaan data

Method	POST
URL	http://permitta.nekoding.com/api/permintaan-data/update/{id}
Headers	Authorization
Body	<ul style="list-style-type: none"> - Title - Instansi - Tujuan - Surat
Keterangan	Pada request api ini memerlukan headers authorization access token untuk mengubah permintaan data berdasarkan id dan memerlukan body request title,instansi,tujuan,surat
Response	<pre>{ "message": "Berhasil diupdate" }</pre>

d. Pengujian Aplikasi

Pengujian dilakukan terhadap fungsionalitas beberapa form dan komponen pada aplikasi yang dirancang, menggunakan metode *blackbox* untuk memastikan semua fungsi telah berjalan sesuai dengan tujuan perancangannya.

Tabel 8. Pengujian Aplikasi

Komponen	Test Factor	Hasil	Keterangan
Register User	Menginput foto, nama lengkap, alamat email, password dan konfirmasi password. Kemudian menekan tombol daftar	Berhasil	Berhasil membuat akun dan menampilkan <i>alert</i> “Registrasi Sukses. Silahkan login untuk melanjutkan”.

Komponen	Test Factor	Hasil	Keterangan
Permintaan Data	Menginput judul data yang diminta, instansi/organisasi, tujuan meminta data, serta mengupload surat permohonan. Kemudian menekan tombol kirim.	Berhasil	Menampilkan <i>alert</i> "Permintaan data berhasil dikirim".
Melihat Detail Permintaan Data	Menekan salah satu permintan data.	Berhasil	Menampilkan detail permintaan data
Mengunduh Permintaan Data	Mendownload file data yang diminta.	Berhasil	Menampilkan <i>alert</i> "Download selesai. Silahkan cek di unduhan".
Login Admin	Memasukkan alamat email dan password. Kemudian tekan tombol <i>login</i> .	Berhasil	Menampilkan halaman <i>dashboard</i> .
Melihat Surat Permohonan	Menekan tombol "lihat surat permohonan" pada salah satu permintaan data yang masuk.	Berhasil	Menampilkan surat permohonan <i>user</i>
Terima Permintaan Data	Menekan tombol "lihat" pada kolom aksi dan memilih sektoral sesuai data yang diminta. Kemudian klik "terima".	Berhasil	Menampilkan <i>alert</i> "Sukses. Data diterima". Dimana data secara otomatis akan pindah ke halaman "Data Diproses"
Upload File Permintaan	Menekan tombol "upload file" pada kolom aksi dan masukkan nama data serta file yang ingin dikirim dengan menekan tombol "tambah file". Kemudian klik "upload".	Berhasil	Menampilkan <i>alert</i> "Data di tambahkan"
Tambah List Sektoral	Menekan tombol "Tambah Data" dan masukkan nama, gambar serta deskripsi sektoral yang ingin ditambah. Kemudian klik "Submit"	Berhasil	Menampilkan <i>alert</i> "Data Terkirim" dan data secara otomatis akan pindah ke menu "Data Terkirim".
Request dan Response Register User	Mengetes API register dengan memasukkan name,email,image,password	Berhasil	Menampilkan response data user dan token user
Request dan Response Sektoral	Mengetes API sektoral dengan menggunakan Authorization bearer token user	Berhasil	Menampilkan response list data sektoral
Request dan Response Permintaan Data	Mengetes API permintaan data dengan menggunakan Authorization bearer token user	Berhasil	Menampilkan response list data permintaan data user
Request dan Response Melihat Detail Permintaan Data	Mengetes API detail permintaan data menggunakan Authorization bearer token user	Berhasil	Menampilkan response data permintaan data berdasarkan id
Request dan Response Menghapus Permintaan Data	Mengetes API menghapus permintaan data dengan menggunakan Authorization bearer token user	Berhasil	Menampilkan response data message "berhasil di delete"
Request dan Response Mengubah Permintaan Data	Mengetes API mengubah permintaan data dengan memasukkan data title,tujuan,instansi, dan surat serta menggunakan Authorization bearer token user	Berhasil	Menampilkan response data message "berhasil diupdate"

Berdasarkan tabel rekapitulasi pengujian di atas dapat disimpulkan bahwa sistem sudah memenuhi kriteria untuk kerja yang telah ditetapkan. Pengujian terhadap fungsionalitas setiap *form*, menunjukkan hasil yang baik/diterima. Semua *form* yang dibuat dapat bekerja sesuai dengan perencanaan dan perancangan dengan demikian dapat dikatakan bahwa aplikasi ini bekerja dengan benar dan menunjukkan performa yang baik.

4. Kesimpulan

Berdasarkan uraian pembahasan dan pengujian yang telah dilakukan, maka dapat ditarik kesimpulan sebagai berikut :

- a. Aplikasi permintaan data yang dirancang dapat memberikan kemudahan kepada masyarakat ataupun instansi dan lembaga terkait yang ingin meminta data sektoral.
- b. Penggunaan *Rest API* dalam pembuatan aplikasi permintaan data sektoral memudahkan dalam proses menghubungkan *web* pada sisi *server (admin)* dan *user (android)* pada sisi *client*.

DAFTAR PUSTAKA

- [1] Yosef Koko Kurniawan, Yetli Oslan, Harianto Kristanto, “Implementasi *Rest - Api* untuk Portal Akademik UKDW berbasis *Android*”, Jurnal Eksis, Volume 6, Nomor 02, 2013.
- [2] Permana, I. G. T., Rusdianto, D. S., & Fanani, “Pengembangan Sistem Presensi berbasis Lokasi menggunakan Geofence WiFi dan REST API pada Fakultas Ilmu Komputer Universitas Brawijaya”, Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN, 2548, 2019.
- [3] Ghozaly, S, “Implementasi *Rest Api* Pada Pusat Informasi Mahasiswa Universitas Teknologi Yogyakarta (Doctoral dissertation, University of Technology Yogyakarta)”, 2019
- [4] Sya’ban Muhammad Amril, “Rancang Bangun Sistem Jual Beli Mobil Bekas Online Berbasis Website Menggunakan Framework Vue”, Js. Teknik Informatika dan Komputer. Politeknik Negeri Jakarta, 2020
- [5] Rully Yulian MF, “Membangun REST API Services dengan Deno dan PostgreSQL”, 2020.
- [6] Gianty, T, “Black Box Testing and Implementation”, Singapore, Smart Book, 2012
- [7] Satyaputra, A., Aritonang, E. M., & Kom, S. (2016). Lets Build Your Android Apps with Android Studio. Elex Media Komputindo.