**­**

**PROYEK AKHIR**

|  |
| --- |
|  |

SISTEM PENGELOLAAN KEBUTUHAN AIR PADA TANAMAN MENGGUNAKAN WIRELESS SENSOR NETWORK

BINTANG REFANI MAULUDI

NRP. 2103161052

Dosen Pembimbing :

Isbat Uzzin Nadhori S.Kom, MT

NIP. 197405052003121002

M. Udin Harun Al Rasyid, S.Kom, Ph.D

NIP. 198108082005011001

PROGRAM STUDI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2019

**PROYEK AKHIR**



SISTEM PENGELOLAAN KEBUTUHAN AIR PADA TANAMAN MENGGUNAKAN WIRELESS SENSOR NETWORK

# BINTANG REFANI MAULUDI

NRP. 2103161052

Dosen Pembimbing :

Isbat Uzzin Nadhori S.Kom, MT

NIP. 197405052003121002

M. Udin Harun Al Rasyid, S.Kom, Ph.D

NIP. 198108082005011001

PROGRAM STUDI D3 TEKNIK INFORMATIKA

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2019

# **ABSTRAK**

Saat ini perkembangan teknologi informasi pada pemerintah Jawa Timur semakin pesat. Menyadari hal tersebut pemerintah berupaya untuk melakukan pengembangan dalam mengelola informasi. Salah satu upaya pemerintahan adalah pengembangan e-Government (e-Gov), dimana hal tersebut diharapkan bisa membantu meningkatkan komunikasi antar pemerintah, masyarakat dan golongan pembisnis sehingga bisa mendorong perkembangan politik dan ekonomi. Dalam perkembangan e-Government tersebut terdapat kendala yang dihadapi. Salah satu kendalanya adalah adanya masalah interoperabilitas antar sistem, dimana antar sistem tidak bisa berkomunikasi dan tidak bisa bertukar data satu sama lain. Keadaan saat ini adalah setiap database pada masing masing sistem masih bersifat sektoral. Hal tersebut dapat mengakibatkan resiko terjadinya redudansi data dan terjadi kehilangan data saat perpindahan data. Untuk menyelesaikan masalah tersebut, penulis membuat aplikasi integrasi yang dapat membuat sistem berkomunikasi dan bertukar data satu dengan lainnya. Aplikasi ini menggunakan teknologi API dalam mengkomunikasikan data. Aplikasi ini dibangun menggunakan arsitektur RESTful API dan menggunakan OAuth2 sebagai keamananya. Aplikasi ini mampu untuk mengkomunikasikan antar sistem dan untuk bertukar data antara satu sistem dan sistem lain.

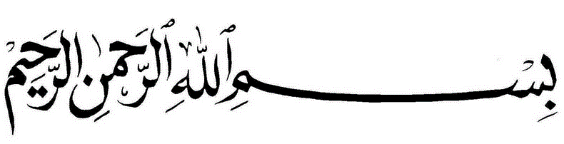
***Kata kunci: e-Government, REST, Oauth2, Interoperabilitas***

# ***ABSTRACT***

*Currently the development of information technology in the government of East Java is increasing rapidly. Realizing that the government is trying to do development in managing information. One of the government's efforts is the development of e-Government (e-Gov), which is expected to help improve communication between government, society and business class so as to encourage political and economic development.* *In the development of e-Government there are obstacles encountered. One of the obstacles is the interoperability between systems, where inter-system can not communicate and can not exchange data with each other. The current state is that each database in each system is still sectoral. This condition may cause redudancy of data and loss of data during migration data. To solve this problem, the author create an integraton application that can make the system communicate and exchange data with each other. This application is built using the RESTful API architecture and uses OAuth2 as its security. This application is able to communicate between system and to exchange data between one system and another system.*

***Keywords: e-Government, REST, OAuth2, Interoperability.***

# **KATA PENGANTAR**



**Assalamu’alaikum Wr. Wb.**

Syukur alhamdulilah atas kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis bisa menyelesaikan proyek akhir ini yang berjudul:

**SISTEM PENGELOLAAN KEBUTUHAN AIR PADA TANAMAN MENGGUNAKAN WIRELESS SENSOR NETWORK**

Buku proyek akhir ini disusun sebagai salah satu syarat guna menyelesaikan studi pada program Diploma III pada jurusan Teknik Informatika di Politeknik Elektronika Negeri Surabaya. Proses penyelesaian proyek akhir ini berdasar pada teori-teori yang telah diperoleh dalam perkuliahan, study literature dan bimbingan dari dosen pembimbing.

Penulis menyadari bahwasanya masih banyak kekurangan dan kelemahan dalam buku ini. Oleh karena itu kritik dan saran yang membangun serta koreksi yang konstruktif sangat diharapkan untuk perkembangan lebih lanjut. Semoga dengan adanya buku ini memberikan manfaat bagi pengembangan ilmu pengetahuan dan bagi semua pihak pada umumnya serta bagi penulis sendiri pada khususnya.

***Wassalamu’alaikum Wr. Wb.***

Surabaya, 20 Juli 2018

**Bintang Refani Mauludi**

# **UCAPAN TERIMA KASIH**

Segala puji dan syukur bagi Allah SWT yang telah memberikan rahmat, karunia, dan petunjuk sehingga proyek akhir ini dapat terselesaikan. Penyelesaian proyek akhir ini tidak akan terwujud tanpa adanya pihak lain. Oleh karena itu dalam kesempatan ini penulis menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu dan membimbing pelaksanaan dan penyelesaian proyek akhir ini, yaitu kepada :

Allah SWT yang telah mencurahkan segenap kenikmatan baik berupa nikmat iman dan islam serta barokah, hidayah dan inayah nya kepada penulis.

1. Ibu , Ayah dan Adik saya yang telah memberikan support dan perhatian kepada penulis dalam menyelesaikan proyek akhir ini.
2. Bapak Dr. Zainal Arief, ST.,MT. selaku Direktur Politeknik Elektonika Negeri Surabaya
3. Tri Harsono., S.Si., M.Kom., Ph.D. selaku Ketua Departemen Teknik Informatika dan Teknik Komputer Politeknik Elektronika Negeri Surabaya
4. Bapak Arif Basofi S.Kom, MT selaku Ketua Program Studi D3 Teknik Informatika PENS Politeknik Elektronika Negeri Surabaya.
5. Bapak Isbat Uzzin Nadhori S.Kom, MT dan Bapak M. Udin Harun Al Rasyid, S.Kom, Ph.D selaku dosen pembimbing penyelesaian proyek akhir ini.

Semoga ucapan terima kasih dari penulis tentunya belum cukup, semoga Allah SWT membalas segala kebaikan yang telah diberikan tersebut.

# **DAFTAR ISI**

[Halaman judul putih ii](#_Toc519698089)

[ABSTRAK iii](#_Toc519698090)

[*ABSTRACT* iv](#_Toc519698091)

[KATA PENGANTAR v](#_Toc519698092)

[UCAPAN TERIMA KASIH vi](#_Toc519698093)

[DAFTAR ISI vii](#_Toc519698094)

[DAFTAR GAMBAR x](#_Toc519698095)

[DAFTAR TABEL xii](#_Toc519698096)

[BAB I Pendahuluan 1](#_Toc519698097)

[1.1 LATAR BELAKANG 1](#_Toc519698098)

[1.2 PERUMUSAN MASALAH 2](#_Toc519698099)

[1.3 BATASAN MASALAH 3](#_Toc519698100)

[1.4 TUJUAN 3](#_Toc519698101)

[1.5 MANFAAT 3](#_Toc519698102)

[1.6 METODOLOGI 4](#_Toc519698103)

[1.7 SISTEMATIKA STUDI 5](#_Toc519698104)

[BAB II teori penunjang 7](#_Toc519698105)

[2.1 TINJAUAN PUSTAKA 7](#_Toc519698106)

[2.1.1 Model Interoperabilitas Antar Aplikasi E-Government [2] 7](#_Toc519698107)

[2.1.2 Model Interoperabilitas E-Government Menggunakan Kombinasi Service Oriented Architecture (SOA) dan Event Driven Architecture (EDA) [5] 7](#_Toc519698108)

[2.1.3 Review Interoperability Model of e-Government Service [6] 8](#_Toc519698109)

[2.1.4 E-Government Interoperability Framework based on a Real Time Architecture [7] 8](#_Toc519698110)

[2.1.5 Kinerja Arsitektur Interoperabilitas E-Government Multi Platform [10] 9](#_Toc519698111)

[2.1.6 Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov di Pemkab Bantul Yogyakarta [11] 9](#_Toc519698112)

[2.1.7 A Comparison of SOAP and REST Implementations of a Service Based Intereaction Independence Middleware Framework [12] 10](#_Toc519698113)

[2.1.8 Performance Comparison of Enterprise Applications on Mobile Operating System [13] 10](#_Toc519698114)

[2.1.9 Big Data, Open Government and e-Government: Issues, Policies, and Recomendations [14] 11](#_Toc519698115)

[2.1.10 Big Data and E-Government: A Review [15] 11](#_Toc519698116)

[2.1.11 Keynote : Big Data Integration for E-Government [16] 12](#_Toc519698117)

[2.2 DASAR TEORI 13](#_Toc519698118)

[2.2.1 Ruby on Rails Framework 13](#_Toc519698119)

[2.2.2 REST API Web Service 13](#_Toc519698120)

[2.2.3 Teknologi API dengan Authentikasi OAuth2 13](#_Toc519698121)

[2.2.4 File JSON 14](#_Toc519698122)

[BAB III perancangan dan pembuatan sistem 15](#_Toc519698123)

[3.1 PERANCANGAN SISTEM 15](#_Toc519698124)

[3.1.1 Desain Sistem Secara Umum 15](#_Toc519698125)

[3.1.1.3 Aplikasi Setiap SKPD 16](#_Toc519698126)

[3.1.1.4 Software Development Kit (SDK) 16](#_Toc519698127)

[3.1.2 Konsep Alur Pemanfaatan Data 17](#_Toc519698128)

[3.1.3 Arsitektur Data Warehouse 20](#_Toc519698129)

[3.1.4 Proses ETL 22](#_Toc519698130)

[3.1.5 Arsitektur API Data Service 26](#_Toc519698131)

[3.1.6 Alur Sistem Request 28](#_Toc519698132)

[3.1.7 SOP Alur Pengambilan Data 34](#_Toc519698133)

[3.1.8 SOP Alur Pemanfaatan Data 36](#_Toc519698134)

[3.2 PERBANDINGAN SOAP DAN REST 38](#_Toc519698135)

[3.2.1 SOAP (Simple Object Access Protocol) 38](#_Toc519698136)

[3.2.2 REST (Representational State Transfer) 39](#_Toc519698137)

[3.2.3 Perbandingan SOAP dan REST 39](#_Toc519698138)

[3.3 USER INTERFACE 41](#_Toc519698139)

[3.2.4 Register dan Login 41](#_Toc519698140)

[3.2.5 Dashboard 42](#_Toc519698141)

[3.2.6 Pengaturan user 43](#_Toc519698142)

[3.2.7 Pengaturan Credential 44](#_Toc519698143)

[3.2.8 Pengaturan Agama 45](#_Toc519698144)

[3.2.9 Pengaturan Lainnya 46](#_Toc519698145)

[3.2.10 Profile / Biodata 46](#_Toc519698146)

[BAB IV pengujian dan analisa 47](#_Toc519698147)

[4.1 TUJUAN ANALISA 47](#_Toc519698148)

[4.2 PERANGKAT UJI COBA 47](#_Toc519698149)

[4.3 DATA 47](#_Toc519698150)

[4.4 SKENARIO UJI COBA 48](#_Toc519698151)

[4.4.1 Pendaftaran dan Mendapatkan Application Credentials 48](#_Toc519698152)

[4.4.2 Pengujian Request Token dan Request API Menggunakan Postman 48](#_Toc519698153)

[4.4.3 Penggunaan SDK dan Unit Testing 48](#_Toc519698154)

[4.4.4 Implementasi pada Aplikasi 49](#_Toc519698155)

[4.5 PENGUJIAN SISTEM 49](#_Toc519698156)

[4.5.1 Pendaftaran dan Mendapatkan Token 49](#_Toc519698157)

[4.5.2 Pengujian Request Token dan Request API Menggunakan Postman 50](#_Toc519698158)

[4.5.3 Penggunaan SDK dan Unit Testing 54](#_Toc519698159)

[4.6 IMPLEMENTASI DATA KE DALAM SISTEM 71](#_Toc519698160)

[4.6.1 Implementasi API Untuk Aplikasi Antrian BPJS 71](#_Toc519698161)

[4.6.2 Implementasi Aplikasi Pembuatan SKCK Polda Jawa Timur 78](#_Toc519698162)

[BAB V penutup 87](#_Toc519698163)

[5.1. KESIMPULAN 87](#_Toc519698164)

[5.2. SARAN 87](#_Toc519698165)

[Daftar Pustaka 89](#_Toc519698166)

[BIOGRAFI PENULIS 91](#_Toc519698167)

# **DAFTAR GAMBAR**

[Gambar 2.1 File Format JSON 14](#_Toc519694068)

[Gambar 3.1 Desain sistem secara umum 15](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694175)

[Gambar 3.2 Konsep alur pemanfaatan data 17](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694176)

[Gambar 3.3 Arsitektur data warehouse 20](#_Toc519694177)

[Gambar 3.4 Proses ETL 22](#_Toc519694178)

[Gambar 3.5 Multistage data transformation 23](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694179)

[Gambar 3.6 Arsitektur API data service 26](#_Toc519694180)

[Gambar 3.7 Alur sistem request 28](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694181)

[Gambar 3.8 SOP alur pengambilan data 34](#_Toc519694182)

[Gambar 3.9 SOP alur pemanfaatan data 36](#_Toc519694183)

[Gambar 3.10 Tampilan Registrasi 41](#_Toc519694184)

[Gambar 3.11 Tampilan Login 42](#_Toc519694185)

[Gambar 3.12 Tampilan Dashboard 43](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694186)

[Gambar 3.13 Tampilan list user 43](file:///D:\Kuliah\Tugas%20Akhir\buku\me\PROYEK%20AKHIR.docx#_Toc519694187)

[Gambar 3. 14 Role use 44](#_Toc519694188)

[Gambar 3.15 Kebutuhan Authorization 45](#_Toc519694189)

[Gambar 3.16 Tampilan list agama 45](#_Toc519694190)

[Gambar 3.17 Tampilan profile 46](#_Toc519694191)

[Gambar 4.1 Application Credential untuk uji coba 50](#_Toc519694091)

[Gambar 4.2 Parameter dan url untuk request token 51](#_Toc519694092)

[Gambar 4.3 Response dari request token 51](#_Toc519694093)

[Gambar 4.4 Parameter dan url untuk request API 52](#_Toc519694094)

[Gambar 4.5 Response dari request API 53](#_Toc519694095)

[Gambar 4.6 Direktori SDK 54](#_Toc519694096)

[Gambar 4.7 Fitur access token 55](#_Toc519694097)

[Gambar 4. 8 Fitur get Authorization Url 56](#_Toc519694098)

[Gambar 4.9 get Access Token melalui Authorization Code 57](#_Toc519694099)

[Gambar 4.10 Fitur get Access Token melalui Client Credential 58](#_Toc519694100)

[Gambar 4.11 Inisialisasi get Access Token 59](#_Toc519694101)

[Gambar 4.12 Request menggunakan Client Credential 59](#_Toc519694102)

[Gambar 4.13 Inisialisasi request 60](#_Toc519694103)

[Gambar 4.14 Test Parameter 61](#_Toc519694104)

[Gambar 4.15 Test Access Token 62](#_Toc519694105)

[Gambar 4.16 Inisialisasi testing Client Credential 63](#_Toc519694106)

[Gambar 4.17 Test get Access Token Client Credential 63](#_Toc519694107)

[Gambar 4.18 Inisialisasi testing Authorization Code 64](#_Toc519694108)

[Gambar 4.19 Testing Authorization Url 64](#_Toc519694109)

[Gambar 4.20 Test get Access Token Authorization Code 65](#_Toc519694110)

[Gambar 4.21 Testing Request 66](#_Toc519694111)

[Gambar 4.22 Hasil Unit Testing 67](#_Toc519694112)

[Gambar 4.23 Halaman Awal Aplikasi Client 68](#_Toc519694113)

[Gambar 4.24 Hasil NIK Salah 69](#_Toc519694114)

[Gambar 4.25 Hasil NIK Benar 70](#_Toc519694115)

[Gambar 4.26 Aplikasi BPJS Cek NIK 72](#_Toc519694116)

[Gambar 4.27 Cek NIK Salah 73](#_Toc519694117)

[Gambar 4.28 Validasi Nomor NIK 74](#_Toc519694118)

[Gambar 4.29 Halaman menu 75](#_Toc519694119)

[Gambar 4.30 Generate Nomor Antrian 76](#_Toc519694120)

[Gambar 4.31 Cetak Antrian 77](#_Toc519694121)

[Gambar 4.32 Halaman Utama Sistem Informasi Ditentelkam 79](#_Toc519694122)

[Gambar 4.33 Login Sistem Informasi Ditintelkam 80](#_Toc519694123)

[Gambar 4.34 Registrasi Sistem Informasi Ditintelkam 81](#_Toc519694124)

[Gambar 4.35 Menu Layanan 82](#_Toc519694125)

[Gambar 4. 36 Layanan pendaftaran SKCK baru 83](#_Toc519694126)

[Gambar 4.37 Persiapan Dokumen 84](#_Toc519694127)

[Gambar 4. 38 Pengisian Otomatis pada Formulir 85](#_Toc519694128)

# **DAFTAR TABEL**

[Table 3.1. Parameter Request Token 32](#_Toc519694564)

[Table 3.2. Parameter Untuk Authorization Pemilik Aplikasi 33](#_Toc519694565)

[Table 3.3. Parameter Request Informasi Biodata WNI 33](#_Toc519694566)

[Table 4.1. Spesifikasi Uji Coba 47](#_Toc519694572)

# **BAB I**

**PENDAHULUAN**

## **LATAR BELAKANG**

Negara Indonesia adalah negara yang sangat kaya akan sumber daya alamnya, diantaranya yaitu kekayaan lautnya, mineral sampai dengan hasil bumi. Negara Indonesia mempunyai sumber daya alam dan luas wilayah yang cukup besar, sehingga bidang pertanian memiliki potensi yang sangat besar sebagai pendapatan negara.

Selain itu, sektor pertanian merupakan salah satu sektor paling penting yang meningkatkan pertumbuhan ekonomi masyarakat indonesia. Sehingga sektor pertanian sebagai salah satu pilar besar perekonomian indonesia. Badan Pusat Statistik (BPS) merilis pertumbuhan ekonomi Indonesia / Produk Domestik Bruto (PDB) triwulan II 2018 terhadap triwulan II 2017 meningkat sebesar 4,21 persen quarter to quarter (q-to-q). Dari sisi produksi, pertumbuhan tertinggi terjadi pada lapangan usaha pertanian, kehutanan, dan perikanan sebesar 9,93 persen. Kemudian, perusahaan jasa 3,37 persen dan jasa lainnya 3,30 persen.Hal tersebut membuktikan bahwa sektor pertanian berperan dalam pembangunan ekonomi Indonesia [1].

Salah satu faktor yang paling penting pada bidang pertanian adalah pengelolaan kebutuhan air pada tanaman secara tepat. Pengelolaan kebutuhan air adalah jumlah volume air yang diperlukan untuk memenuhi kebutuhan air untuk tanaman dengan memperhatikan kondisi lingkungan dan jumlah air yang diberikan oleh alam melalui hujan. Sehingga pengelolaan air harusnya dilakukan sesuai dengan kondisi lingkungan dan cuaca.

Peran air pada tanaman sangat lah penting. Seperti halnya manusia, tanaman juga membutuhkan air yang cukup untuk pertumbuhan dan perkembangannya. Tanaman sendiri terdiri dari 90% air. Peran air dalam tanaman adalah sebagai komponen utama dalam proses fotosintesis dan transpirasi. Karena air sebagai komponen utama dalam proses fotosintesis maka air menjadi hal yang penting sekali sebagai sumber energi bagi tanaman tersebut [2].

Permasalahan yang akan diangkat pada penelitian ini adalah kurangnya sistem monitoring pada pertanian yang terintegrasi secara realtime dengan aplikasi mobile dan web serta dapat melakukan pemantauan secara efisien kapanpun untuk mendapatkan volume air yang dibutuhkan tanaman sesuai dengan keadaan sekitar .

Dengan latar belakang diatas, maka pada proyek akhir ini akan dibagun sebuah sistem yang dapat membatu manusia menentukan volume penyiraman tanaman berdasarkan keadaan lingkungan tanaman dan cuaca dengan keluaran berupa bilangan yang merupakan volume air untuk penyiraman

## **PERUMUSAN MASALAH**

Berdasarkan latar belakang di atas, maka di temukan beberapa masalah, yaitu:

1. Kekurangan air menyebabkan tanaman layu dan akhirnya akan menyebabkan kematian pada tanaman sedangkan kelebihan air pada tanaman akan meyebabkan permukaan tanah tempat tanaman hidup menjadi lembab karena kelebihan air, keaadaan lembab tersebut akan memunculkan mikroorganisme jamur yang akan mengakibatkan tumbuhnya penyakit bagi tanaman.
2. Kurang maksimalnya sistem monitoring pengelolaa kebutuhan air pada tanaman.
3. Pemberian volume kebutuhan air yang hanya berdasarkan perkiraan pemilik tanaman.

## **BATASAN MASALAH**

Batasan masalah dalam pembuatan karya ilmiah ini adalah sebagai berikut :

1. Menggunakan Node Js sebagai bahasa pemrograman utama di bagian backend-end Service

## **TUJUAN**

Dari permasalahan di atas maka tujuan dari aplikasi ini adalah

1.Pemberian volume air secara tepat sesuai kebutuhan tanaman agar hasil panen yang didapat maksimal

2) Menyediakan sistem pengelolaan kebutuhan air pada tanaman secara real-time dan tepat dengan memperhatikan peramalan cuaca, dan kondisi linkungan. Keluaran dari sistem ini adalah keterangan volume air yang dibutuhkan oleh tanaman dan dapat dilihat melalui smartphone dan web.

3) Perhitungan volume air yang dibutuhkan tanaman dihitung dengan menggunakan suatu metode, dan menghasilkan keluaran jumlah volume air yang dibutuhkan berdasarkan beberapa parameter.

## **MANFAAT**

Manfaat yang dapat diperoleh dari pembangunan sistem adalah menyediakan sistem monitoring tanaman yang efektif, realtime, serta terintegrasi berbasis mobile sehingga mempermudah pengguna dalam melakukan monitoring kapanpun.

## **METODOLOGI**

Metodologi yang digunakan dalam proyek akhir ini meliputi beberapa tahapan sebagai berikut :

1. **Studi Literatur**

Studi literatur merupakan langkah awal dalam pengerjaan proyek akhir ini. Tahapan ini adalah tahap untuk mencari informasi terkait konsep didalam Internet of Things, penentuan module sensor, dan materi untuk pembuatan aplikasi berbasis android dan website.

1. **Pengumpulan Data**

Pada tahap ini dilakukan kegiatan pengumpulan data pendukung pembuatan aplikasi. Data yang dibutuhkan yaitu data-data yang perlu diintegrasikan antar sistem di Jawa Timur

1. **Perancangan Sistem**

Tahap ini adalah tahap perancangan desain sistem pada alat sensor, sistem pada Back-End/web service (API dan Web-Socket), dan sistem pada aplikasi Android.

1. **Pembuatan Sistem**

Tahap ini adalah pembuatan sistem dari rancangan sistem yang telah ada. Pembuatan sistem dilakukan mulai dari pembuatan Back-End service API dan Web socket dan pembuatan aplikasi android.

1. **Pengujian dan Analisa**

Pengujian dan analisa dimaksud untuk mengetahui sejauh mana aplikasi yang dibuat pada proyek akhir ini bisa berfungsi sesuai dengan tujuan yang diharapkan dalam proyek akhir ini.

1. **Pembuatan Laporan**

Membuat dokumentasi dari semua tahapan berupa laporan yang berisi tentang dasar teori, hasil proyek akhir dan hasil analisa.

1. SISTEMATIKA STUDI

Sistematika pembahasan dari Proyek Akhir ini direncanakan sebagai berikut:

**BAB I PENDAHULUAN**

Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang,perumusan masalah, batasan masalah, tujuan dan sasaran, metodologi, serta sistematika pembahasan dari proyek akhir ini.

**BAB II TEORI PENUNJANG**

Bab ini membahas mengenai teori-teori yang berkaitan dengan penyelesaian Proyek Akhir, yang didapatkan dari berbagai macam buku serta sumber-sumber terkait lainnya yang berhubungan dengan pembuatan Proyek Akhir ini.

**BAB III PERANCANGAN DAN PEMBUATAN SISTEM**

Bab ini membahas mengenai perancangan sistem, meliputi perancangan web untuk transaksi data antar user (superadmin, admin, pegawai, petugas, dan warga), perancangan database dan perancangan penggunakan teknologi API.

**BAB IV PENGUJIAN DAN ANALISA**

Bab ini merangkum tentang hasil pengujian beserta analisa dari sistem yang telah dibuat dalam proyek akhir sehingga bisa didapat kelebihan atau kekurangan sistem yang telah dibuat.

**BAB V PENUTUP**

Bab ini berisi tentang kesimpulan dari proyek akhir dilengkapi dengan saran yang berguna untuk mengembangkan sistem untuk dikembangkan lebih baik lagi.

**DAFTAR PUSTAKA**

Berisi tentang referensi-referensi yang telah digunakan sebagai landasan selama pembuatan proyek akhir ini.

*\*Halaman ini sengaja dikosongkan\**

# **BAB II**

**TEORI PENUNJANG**

1. TINJAUAN PUSTAKA

Berikut ini mengenai tinjauan pustaka yang didapat untuk mendukung aplikasi ini :

1. Model Interoperabilitas Antar Aplikasi E-Government [2]

Penelitian ini dilakukan oleh Jazi Eko Istiyanto dan Edhy Susanta Tahun 2012. Analisis dan Perancangan Sistem ini bertujuan untuk membuktikan bahwa pemodelan intereoperabilitas antar aplikasi e-Gov bisa dilakukan melalui cara pemanfaatan teknologi web service. Penilitian ini menghasilkan rancangan model arsitektur web service menggunakan metode REST antar aplikasi. Penulis mengkomunikasikan dua aplikasi dengan menggunakan request url GET/POST dan file yang dikirim adalah file XML

1. Model Interoperabilitas E-Government Menggunakan Kombinasi Service Oriented Architecture (SOA) dan Event Driven Architecture (EDA) [5]

Penelitian ini dilakukan oleh Aris Puji Widodo tahun 2015. Penelitian ini bertujuan membuat model interoperabilitas dan integrase data aplikasi e-Gov Indonesia berbasis arsitektur konkuren yang dikembangkan dalam menggunakan layer service. Penulis mendapatkan kesimpulan bahwa model interoperabilitas bisa dilakukan menggunakan pendekatan Service Oriented Architecture (SOA) kemudian pendekatan Event Driven Architecture (EDA) digunakan untuk membuat notifikasi sebagai event terhadapat service lainnya. Penulis mengkomunikasikan dua aplikasi dengan menggunakan request url GET/POST dan file yang dikirim adalah file XML. Relasi request tidak lagi bersifat *one-to-one* tetapi bersifat *one-to-many.* Relasi *one-to-many* dilakukan dengan cara membuat orkerstrasi antar service dengan menggunakan BPEL. BPEL digunakan untuk mendefinisikan partner yang dimiliki oleh sebuat service sehingga pada saat terjadi request terhadap sebuah service maka services partner juga dijalankan.

1. Review Interoperability Model of e-Government Service [6]

Penelitian ini dilakukan oleh Wahyudi Kumorotomo, I Wayan Ordiyasa, Lukito Edi Nugroho, Paulus Insap Santosa, dan Ridi Ferdiana pada tahun 2015. Penelitian ini menghasilkan bahwa penerapan interoperability pada E-Gov sangat penting untuk dilakukan. Hal itu harus dilakukan karena kebutuhan pelanggannya akan layanan E-Gov sangat besar dan untuk mewujudkan *one-stop government* *service* dan *integrated government*. Penulis disini melakukan perbandingan antar model interoperabilitas dan menghasilkan beberapa dari model tidak cocok diterapkan untuk menyelesaikan adaptasi dari data. Hal itu dikarenakan sumber data didapatkan di tempat yang berbeda dan dengan format yang bermacam macam.

1. E-Government Interoperability Framework based on a Real Time Architecture [7]

Penelitian ini dilakukan oleh Aris Puji Widodo, Jazi Eko Istiyanto, Retantyo Wardoyo dan Purwo Santoso pada tahun 2013. Penelitian ini bertujuan untuk membuat EAF (Enterprice Architecture Framework) untuk interoperabilitas dan data integrase untuk E-Gov di Indonesia berbasis arsitekture *real time* dikembangkan dengan layer service. Tujuan dari penulis disini adalah membuat distribusi dari skema database di setiap kota, provinsi dan tingkat nasional untuk memenuhi kebutuhan yang berbeda. Penulis menyimpulkan bahwa service model seharusnya didefinisikan dalam EAF dimana berkas dari model yang memiliki respon services dengan satuan output parameter seperti, rekaputulasi penduduk, kelahiran dan kematian. EAF yang dipakai oleh penulis ini di bangun menggunakan model service base pada dokumen text dengan XML format. Oleh karena itu penulis menyarankan menggunakan EAF dikarenakan sumber daya untuk interoperabilitas dari EAF relatif berkurang dibandingkan dengan arsitektur aplikasi E-Gov dimana database yang digunakan adalah database local dan aplikasi terkonek langsung dengan database.

1. Kinerja Arsitektur Interoperabilitas E-Government Multi Platform [10]

Penelitian ini dilakukan oleh Aris Puji Widodo pada tahun 2016. Penilitian ini bertujuan untuk menguji kelayakan penggunaan arsitektur SOA pada aplikasi multi platform. Penulis melakukan simulasi pengujian aplikasi pada pemerintahan. Platform yang diujikan oleh penulis adalah aplikasi web dengan bahasa pemrograman berbeda dan database yang berbeda pula. Pengujian menggunakan arsitektur SOA dengan response kembali berupa file format XML. Data uji yang dipakai oleh penulis adalah data jumlah penduduk nasional. Parameter uji yang ditetapkan penulis adalah waktu kecepatan penerimaan response dengan model service yang memiliki response parameter tunggal. Hasil yang diperoleh penulis adalah arsitektur SOA dapat dinyatakan memenuhi standard kelayakan pakai oleh aplikasi pada pemerintahan.

1. Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov di Pemkab Bantul Yogyakarta [11]

Penelitian ini dilakukan oleh Edi Susanta, dan Kabib Mustofa pada tahun 2012. Penilitian ini bertujuan untuk mengimplementasikan web service ke dalam aplikasi sistem informasi E-Gov di Pemkab Bantul Yogyakarta. Pada aplikasi sistem informasi di Pemkab Bantul Yogyakarta masih berlum menerapkan web service untuk sinkronisasi data. Penulis merancang web service menggunakan arsitektur REST untuk implementasi web service ke dalam aplikasi. Response yang akan dikirim adalah file berformat XML. Penulis mengimplementasikan arsitektur REST pada aplikasi sistem informasi yang terdapat pada Pemkab Bantul karena beberapa aplikasi sistem informasi memerlukan layanan web service untuk sinkronisasi data antar aplikasi. Penulis mendapatkan kesimpulan bahwa model web service bisa menjadi alternative solusi untuk proses pertukaran data antar sistem informasi. Dengan kemampuan web service untuk bertukar data, maka dimungkinkan untuk melakukan proses sinkronisasi data antar sistem informasi termasuk dalam e-Gov.

1. A Comparison of SOAP and REST Implementations of a Service Based Intereaction Independence Middleware Framework [12]

Penelitian ini dilakukan oleh Gavin Muligan, dan Denis Gracanin pada tahun 2009. Penilitian ini bertujuan untuk mengetahui perbandingan performa dari arsitektur SOAP dan arsitektur REST. Uji coba yang dilakukan oleh penulis adalah dengan membuat aplikasi portal Framework sebagai middleware. Penulis melakukan uji coba aplikasi dengan menggunakan SOAP dan REST kemudian penulis membandingkan performa dari kedua arsitektur tersebut. Penulis menetapkan parameter efisiensi penggunaan bandwidth jaringan dan kecepatan akses sebagai pembanding dari kedua arsitektur tersebut. Hasil yang diperoleh oleh penulis adalah implementasi arsitektur REST lebih efisien dalam perbandingan efisiensi bandwidth jaringan dan kecepatan akses daripada arsitektur SOAP. Jadi, penulis menyimpulkan bahwa implementasi dari arsitektur REST lebih cocok diterapkan untuk model web service tersebut daripada perilaku seperti RPC yang diterapkan oleh arsitektur SOAP.

1. Performance Comparison of Enterprise Applications on Mobile Operating System [13]

Penelitian ini dilakukan oleh Adrian Mullally, Migei McKelvey dan Kevin Curran pada tahun 2011. Penilitian ini bertujuan untuk mengetahui perbandingan kinerja dari arsitektur web service SOAP dan REST yang di implementasikan pada sistem operasi mobile. Penulis melakukan uji coba pada beberapa platform yaitu mobile web, android dan windows phone. Penulis menetapkan parameter rata rata lamanya waktu untuk mengakses, dan rata rata penggunaan memory sebagai pembanding kedua arsitektur. Hasil yang diperoleh penulis adalah untuk jumlah waktu yang dibutuhkan client pada mobile phone untuk menerima respon, performa dari REST lebih cepat daripada SOAP, kecuali percobaan text pada windows phone SOAP lebih cepat dari pada REST. Jadi, penulis menyimpulkan jika developer ingin mendesain aplikasi web ataupun mobile, REST menyediakan cara yang efisien untuk mengirim dan menerima request dan response dari web service.

1. Big Data, Open Government and e-Government: Issues, Policies, and Recomendations [14]

Penelitian ini dilakukan oleh John Carlo Bertot, Ursula Gorham, Paul T.Jaeger, Lindsay C. Sarin, dan Heeyoon Choi pada tahun 2014. Penulis mengatakan bahwa open data berguna untuk masyarakat karena bisa dimanfaatkan dan bisa menghasilkan produk baru darinya. Open data adalah ide dasar untuk menjadikan data tersebut tidak dibatasi dengan copyright, paten, sensor, atau parameter lain yang bisa mempengaruhi data. Pada pemerintahan, open data membuat kesempatan kepada individu, lembaga swasta, dan lembaga non-profit untuk menemukan pandangan baru dan bisa membuat produk baru seperti service dari data tersebut. Open data pada pemerintahan tidak akan lepas dari Big data. Big data memiliki 3 kunci infrastruktur yaitu : 1) platform untuk pengaturan, penyimpanan dan membuat data bisa diakses, 2) teknologi computer dan kekuatan untuk memproses dataset ukuran besar, 3) format data yang terstruktur dan bisa dipakai ulang. Kunci masalah dari open data dan big data ini adalah informasi dan kebijakan data yang mengatur manajemen, penggunaan, penggunaan kembali, dan aksesibilitas informasi dan data pemerintah. Isu-isu yang diangkat oleh big dan open data serta ketidakmampuan kerangka kebijakan saat ini untuk mengatasinya, membuktikan bahwa perlunya pengembangan model tata kelola big dan open data. Penulis menjelaskan beberapa model dari e-government yang seharusnya di implementasi salah satunya adalah kebijakan berbagi data lintas sector.

1. Big Data and E-Government: A Review [15]

Penelitian ini dilakukan oleh Zaher Ali Al-Sai, dan Laith Mohammad Abualigah pada tahun 2017. Penulis mengatakan bahwa big data dapat membantu proses pengambilan keputusan juga membantu untuk meningkatkan efisiensi dan efektifitas dari organisasional. Konsep teknologi dibalik big data memungkinkan organisasi untuk mendapatkan tujuan yang diinginkan. Manfaat utama menggunakan analitik Big Data adalah menjanjikan kemampuan analitis untuk mengintegrasikan kebijakan berdasarkan perkiraan seluruh rentang pengaruh dan dampak. Big data akan menjadi salah satu investasi terkuat dalam pemerintahan. Istilah Big Data menawarkan peluang baru untuk penciptaan nilai, penemuan, prediksi dan memberdayakan intelijen bisnis untuk mendukung keputusan di pemerintah. Penulis menyebutkan banyak keuntungan big data dalam e-government salah satunya adalah menyediakan dan mengintegrasikan secara efisien resource dari big data serta mengintegrasikan data tersebut untuk membantu proses pengambilan keputusan. Melihat pentingnya big data dalam e-gov tentu saja terdapat tantangan dalam menjalankannya. Terdapat beberapa tantangan big data dalam e-gov salah satunya adalah dalam bidang teknologi yang menyebutkan bahwa big data susah untuk diterapkan karena memiliki berbeda format, struktur data pada database tradisional, ataupun data tidak terstruktur seperti email dan transaksi keuangan. Oleh karena itu penulis menyebutkan solusi yang memungkinkan dari masalah tersebut yaitu menyediakan dan mengintegrasikan secara efisien resource dari big data kemudian dengan teknologi data management yang tepat data tersebut menjadi berguna.

1. Keynote : Big Data Integration for E-Government [16]

Pembicara menyebutkan bahwa big data memiliki beberapa konteks didalamnya yaitu jumlah banyaknya data, kecepatan data masuk dan keluar dan berbagai jenis dan sumber data. Penulis juga menyebutkan bahwa data dibagi menjadi data terstruktur dan data tidak terstruktur. Penulis menyebutkan contoh dari data tidak terstruktur adalah pencarian di google dan menghasilkan dokumen sedangkan data terstruktur adalah query komplek dari sebuah database management system (DBMS). Dari hal tersebut penulis menyebutkan bahwa integrasi adalah masalah yang besar dan memakan biaya yang paling besar dalam perusahaan. Penulis menyebutkan untuk mengintegrasikan data tidak terstruktur adalah dengan keyword kemudian mengambil dokumen yang berhubungan. Untuk terstruktur data pada saatnya masih menggunakan global skema dengan cara 2 database dan 2 skema digabungkan menjadi 1 database dan 1 skema baru yang lebih besar dari 2 database dan 2 skema tersebut. Penulis menyebutkan integrasi dengan data terstruktur masih sulit dilakukan jika menggunakan cara otomatis. Penulis menyebutkan tantangan utama yang dihadapi adalah bagaimana cara mengintegrasi antara data terstruktur dan data tidak terstruktur.

1. DASAR TEORI

Dalam bab ini akan dijelaskan mengenai teori yang mendukung dalam pembuatan aplikasi ini.

1. *Ruby on Rails Framework*

Ruby on Rails adalah *framework* pengembangan aplikasi pada sisi server yang ditulis dengan bahasa pemrograman ruby dibawah lisensi MIT. Rails adalah *framework* model-view-controller (MVC), yang menyediakan struktur default untuk database, layanan web, dan halaman web. Rails juga mendorong dan menfasilitasi penggunaan standar web seperti JSON atau XML untuk transfer data, dan HTML, CSS dan JavaScript untuk tampilan dan antarmuka pengguna.[9]

Dengan menggunakan framework ini kita tidak perlu lagi membuat program dari awal. Library dan struktur file sudah diatur saat kita install *framework* ini. Jadi kita dapat membuat suatu program dengan cepat.

1. REST API Web Service

Representational State Transfer (REST) adalah arsitektur standard komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Arsitektur REST server ini menyediakan resource dan service, kemudian REST client akan mengakses dan menampilkan resource tersebut. Pada aplikasi pada penelitian ini menggunakan web service REST untuk pertukaran data.

Menurut Frank Maurer, Craig Anslow, dan Martin P. Robillard, bahwa REST API ini digunakan sebagai mekanisme untuk integrasi pada aplikasi utama dengan aplikasi lainnya melalui internet.[8]

1. Teknologi API dengan Authentikasi OAuth2

OAuth adalah suatu protocol terbuka yang memungkinkan pengguna untuk berbagi sumber pribadi yang disimpan di suatu web tanpa menyerahkan nama pengguna dan kata sandi. Proses ini dilakukan dengan memberikan token. Setiap token memberikan akses untuk suatu yang spesifik terhadap suatu sumber yang spesifik pula. Aplikasi ini menggunakan API OAuth2 untuk menghubungkan database aplikasi satu dengan yang lainnya. Jika aplikasi pada pemerintahan Jawa Timur ingin mengakses database pada aplikasi lain, aplikasi tersebut melakukan request disertai token yang sudah diberikan oleh aplikasi induk. Kemudian jika token cocok dengan aplikasi induk maka aplikasi tersebut bisa mengakses database pada aplikasi lain. Aplikasi Saka Nusantara ini berperan sebagai jembatan atau penghubung antara aplikasi satu dengan aplikasi lain.

1. File JSON

JSON adalah syntax untuk menyimpan dan bertukar data. JSON ini biasanya dalam bentuk file untuk pertukaran data. Pada aplikasi ini pertukaran data menggunakan JSON karena mudah untuk diatur datanya. Aplikasi dari client melakukan request ke aplikasi pada proyek akhir ini kemudian aplikasi proyek akhir ini mengirimkan file JSON yang berupa data. Data yang dikirim dengan file JSON ini adalah data scope dari beberapa data pusat yang kemudian dikirim ke client. Contoh dari file JSON ada pada gambar 2.1.



Gambar 2.1 File Format JSON

# **BAB III**

PERANCANGAN DAN PEMBUATAN SISTEM

1. PERANCANGAN SISTEM

Pada perancangan sistem ini akan dijelaskan beberapa desain sistem.

1. Desain Sistem Secara Umum

Gambar 3.1. Desain Sistem Secara Umum

Gambar 3.1 Desain sistem secara umum

Webservice

Data Scope A

Data Scope B

Data Scope C

Database Aplikasi

Aplikasi Induk

Aplikasi SKPD A

Database Local

Aplikasi SKPD B

Database Local

Aplikasi SKPD C

Database Local

Request / Respond

SDK

Bisa dilihat pada Gambar 3.1, aplikasi ini berbasis web dan berjalan di server-site. Aplikasi ini menggunakan webservice dengan metode keamanan OAuth2 dimana jika ingin mengakses harus memiliki akses token terlebih dahulu. Akses token tersebut digunakan untuk melakukan authorization ke aplikasi. Aplikasi ini menggunakan sistem one gate sehingga hanya server yang sudah di authorization saja yang dapat mengakses. Dari tiap akses terdapat beberapa role utama yaitu administrator, developer, dan warga yang dimana nanti administrator mempunyai kewenangan dalam memberikan izin akses kepada developer. Untuk role developer mempunyai tugas dalam mengembangkan aplikasi pada masing masing SKPD dan berhak menggunakan service jika sudah diberi izin akses oleh administrator. Untuk role warga hanya diberikan izin akses tertentu dan sudah ditetapkan dari awal.

1. Aplikasi Induk

Aplikasi induk ini adalah aplikasi utama dalam penelitian ini. Aplikasi ini bertugas untuk mengintegrasikan atau menghubungkan dari satu data scope ke data scope lain menggunakan web service. Aplikasi ini berada pada server-side. Aplikasi ini mengambil data scope atau data yang diperlukan saja dari database pusat sehingga aplikasi pada setiap SKPD bisa mendapatkan data yang di inginkan dan diizinkan saja. Aplikasi ini juga memiliki database sendiri yang digunakan untuk menyimpan data local seperti data user, token, role dan lain lain. terdapat beberapa role yang terdapat pada aplikasi induk ini. Tiap role memiliki izin akses yang berbeda beda.

1. Data Scope

Data scope pada sistem ini adalah data tertentu saja yang diambil dari beberapa database pusat data, sehingga tidak semua data dari database pusat yang diberikan kepada developer setiap SKPD. Data scope yang diambil bukan dari satu database saja namun berbagai sumber database yang berbeda beda juga.

1. Aplikasi Setiap SKPD

Aplikasi setiap SKPD ini adalah aplikasi yang terdapat pada masing masing SKPD. Setiap aplikasi mempunyai database local masing masing. Setiap aplikasi dikelola atau dibuat oleh role developer masing masing SKPD. Aplikasi ini yang akan memakai service yang ada pada aplikasi induk.

1. Software Development Kit (SDK)

SDK merupakan seperangkat alat pengembangan software yang memungkinkan pembuatan aplikasi untuk software package, software framework, hardware platform, computer system, video game console, operation system, atau platform lainnya. Jadi SDK ini adalah seperangkat library yang mempermudah developer untuk memanggil API dan diterapkan pada aplikasinya. SDK ini akan diterapkan pada aplikasi setiap SKPD.

1. Konsep Alur Pemanfaatan Data

Gambar 3.2 Konsep alur pemanfaatan data

Data Layer

OPD

BPS

Other Ministry / National Bureau

Public News & complaints

Social Media

Other related resource

Data Analytic (Big Data Platform)

Data Ingest (ETL)

OLAP

Staging

Collecting

OLTP

API Control Manager

API Service

Access

Apps.

SKPD Monitoring Center

Kabupaten/ Kota Monitoring Center

Executive Information System

Gambar 3.2. Konsep Alur Pemanfaatan Data

1. Data layer

Dalam layer ini terdapat data dari berbagai sumber seperti : social media, berita, data pengisian form setiap SKPD. Semua data tersebut disimpan dalam database masing masing SKPD yang bersangkutan. Data yang dikumpulkan dan diambil berbeda beda setiap SKPD. Konsep tersebut diterapkan karena Jawa Timur menerapkan konsep One Data.

1. OLTP layer

Dalam layer ini terdapat proses pengambilan data kemudian terdapat proses pembersihan data. Semua data setiap SKPD dilakukan proses tersebut. Proses tersebut bertujuan untuk menghilangkan redudansi data, kesalahan input, dan data kosong. Dalam layer ini terdapat juga proses staging. Data dari setiap SKPD dikumpulkan dalam 1 datawarehouse.

1. OLAP layer

Dalam layer ini terdapat proses ETL dan terdapat proses data analytic. Setelah seluruh data dari setiap SKPD dikumpulkan dalam 1 datawarehouse, setiap database akan diambil hanya scope datanya saja sesuai kebutuhan. Proses sebelum masuk ke dalam data master untuk analisa terjadi proses ETL dimana proses ini mengubah data yang diambil dari setiap database SKPD menjadi satu nilai. Kemudian dari data yang sudah melalui proses ETL dianalisa dan mendapatkan hasil sesuai kebutuhan.

1. Access layer

Dalam layer ini terdapat proses menjadikan data analisis menjadi API yang kemudian bisa dimanfaatkan oleh pihak yang diizinkan untuk mengakses data tersebut. Pada layer ini terdapat aplikasi induk. Aplikasi induk pada layer ini bertugas untuk mengatur API yang bisa dimanfaatkan oleh aplikasi lain. Aplikasi ini juga mengatur role user yang boleh diizinkan untuk memanfaatkan data dan mengakses API.

1. Application layer

Dalam layer ini terdapat proses untuk memanfaatkan API yang sudah diatur oleh aplikasi induk. Aplikasi yang bisa mengakses API hanya aplikasi yang di izinkan oleh aplikasi induk untuk mengakses API. Data yang didapatkan dari API tersebut dimanfaatkan oleh aplikasi tersebut. Contoh pemanfaatannya adalah validasi data, penginputan otomatis, analisa data semesteran, dan lain lain.

1. Alur Konsep Data

Proses dimulai dari data layer dimana data dari berbagai sumber didapatkan. Data tersebut didapatkan oleh masing masing SKPD namun setiap SKPD mendapatkan data dan mengumpulkan data yang berbeda beda. Proses selanjutnya adalah masuk pada OLTP layer, dimana pada layer ini data dibersihkan dari redudansi data, input salah, atau data kosong. Setelah itu data tersebut dilakukan proses staging dimana data tersebut di kirimkan dan dikumpulkan dalam 1 datawarehouse. Proses selanjutnya adalah masuk pada OLAP layer, dimana pada layer ini dilakukan proses ETL yaitu data yang sudah melalui proses staging diambil data scope setiap database saja, kemudian dari data scope yang didapatkan dianalisis dan mendapat hasil analisa baru. Proses selanjutnya adalah masuk pada Access layer dimana pada layer ini data yang sudah didapat analisanya dijadikan API. API tersebut digunakan oleh aplikasi lain untuk dimanfaatkan oleh aplikasi tersebut. Jika gubernur atau pimpinan daerah ingin melihat analisa lain dari source tertentu maka proses akan kembali pada data layer. Proses selanjutnya adalah masuk pada layer application, dimana pada layer ini aplikasi lain memanfaatkan data dari API yang sudah diatur pada access layer. Contoh dari pemanfaatan data tersebut seperti validasi data, pengisian data otomatis, atau analisis data semesteran.

1. Arsitektur Data Warehouse

Data Source

Data Source

API Agent

API Agent

Data Source

API Agent

Internet

API Gateway

Streamset

Apache Kafka

Database Staging

Data Warehouse Diskominfo

Gambar 3.3 Arsitektur data warehouse

1. Data Source

Dalam arsitektur ini data source yang dimadsud adalah data sumber atau data wali dari SKPD yang bersangkutan. Data tersebut disimpan oleh database masing masing SKPD. Data yang dimadsud disini adalah data operasional masing masing SKPD sehingga data source ini masih bisa diupdate. Dalam proses pengiriman data pada arsitektur yang ditunjukkan oleh gambar 3.3, data akan diubah menjadi format JSON sehingga mudah untuk dikirimkan melalui API.

1. API Agent

Dalam arsitektur ini API Agent berguna untuk mengirim data dari database sumber ke database staging melalui API. Dalam proses pengiriman data, API Agent ini ditanamkan dalam sistem aplikasi pada masing masing SKPD kemudian dikoneksikan dengan API Gateway. API Gateway merupakan sebuah platform microservice manajemen API yang berdiri diantara client dan service. Dengan menggunakan konsep REST API yaitu data dijadikan dalam format JSON lalu dikirimkan ke database staging. Proses pengiriman data dibantu oleh Streamset atau Apache Kafka sebagai ETL tools untuk membuat data dapat diakses, dan digunakan pada sistem data yang berbeda.

1. Database Staging

Dalam arsitektur ini database staging adalah tempat semua data dari data sumber yang dikumpulkan menjadi satu database besar. Dalam database staging ini terdapat proses transform dan cleaning data. Langkah ini digunakan untuk membersihkan data dari redudansi data dan data kosong (null value). Kemudian dilakukan query yang akan menyimpan data yang sudah dibersihkan ke dalam data warehouse.

1. Data Warehouse Diskominfo

Dalam arsitektur ini data warehouse adalah tempat semua data dari data sumber yang sudah dibersihkan dan ditransformasi pada database staging. Data warehouse ini terdapat pada server yang terletak pada Diskominfo Jawa Timur. Data warehouse ini digunakan untuk proses analitik data dan terdapat bisnis aplikasi OLAP didalamnya.

1. Alur Infrastruktur

Data source yang ada dari sumber data akan terkoneksikan dengan agent melalui API Gateway. API Gateway bertindah sebagai reverse proxy maupun melayani permintaan routing dari client ke services, dimana juga melakukan berbagai pekerjaan yang saling terkait seperti otentikasi, keamanan koneksi, SSL termination dan rate limiting. API Gateway akan meneruskan koneksi data ke Stramset sebagai ETL Tools atau Apache Kafka. Setelah itu data dari API Gateway disimpan dalam sebuah database sebagai proses staging. Kemudian semua data tersebut dilakukan prose cleaning lalu dimasukkan ke dalam Data Warehouse Diskominfo lalu dilakukan proses analitik dan pembuatan API service.

1. Proses ETL

Extract

Transform dan Clean

Load

Data Warehouse

Source Data Area

Staging Area

Data Warehouse

Gambar 3.4 Proses ETL

1. Extract

Proses extract adalah proses dimana data dari berbagai sumber di pindah ke dalam satu database besar. Namun data yang dipindahkan masih memiliki resiko terdapat ke cacatan seperti : terdapat redudansi data, terdapat data kosong (NULL value), data didak sesuai format. Pada sistem yang diterapkan proses extract ini terjadi pada saat perpindahan data dari database sumber ke dalam database staging.

1. Transform dan Cleaning

Proses transform dan cleaning adalah proses dimana data yang sudah ada pada satu database besar diubah sedemikian rupa sehingga tidak terdapat kecacatan didalamnya. Proses ini bisa dilakukan didalam data warehouse ataupun di luar data warehouse.

Pada sistem yang diterapkan proses transform dan cleaning ini terjadi di dalam database staging. Teknik yang digunakan pada proses transform dan cleaning ini adalah teknik multistage data transformation. Proses transform terjadi pada database staging kemudian setelah data tidak terdapat kecacatan, data dimasukkan ke dalam database warehouse. Gambar 3.5 adalah proses transform yang terjadi pada sistem.

Database Staging

Data Sumber

Cleaning

Transform

Query Insert

Data Warehouse Diskominfo

Gambar 3.5 Multistage data transformation

Gambar 3.5. Pipelined Data Transformation

1. **Data Sumber**

Data sumber adalah data sumber atau data wali dari SKPD yang bersangkutan. Data ini dimasukkan ke dalam tabel pada database staging setelah didapatkan dari API Gateway. Data yang terkandung didalamnya masih terdapat banyak kecacatan seperti terdapat null value, kesalahan format, atau data kembar.

1. **Cleaning**

Proses cleaning adalah proses dimana data pada database staging dibersihkan dari null value dan data yang kembar. Data yang sudah melewati proses ini sudah tidak ada null value namun masih terdapat beberapa kecacatan didalamnya seperti format data masih terdapat kesalahan.

1. **Transform**

Proses transform adalah proses dimana data pada database staging diubah formatnya sesuai dengan format yang ada pada data warehouse. Dalam proses ini terdapat proses validasi terhadap data atau pengecekan terhadap data. Setelah melewati proses ini data yang ada pada database staging sudah bersih dari kecacatan.

1. **Query Insert**

Proses query ini adalah proses untuk memindahkan data dari database staging ke data warehouse. Data yang dikirim ini adalah data yang sudah tidak ada kecacatan.

1. **Data Warehouse Diskominfo**

Data warehouse diskominfo ini adalah database dimana tersimpan semua data yang sudah dibersihkan pada database staging dalam proses sebelumnya. Data warehouse ini terdapat pada sistem serven Diskominfo Jawa Timur yang kemudian akan dijadikan API dan proses analitik.

1. **Alur Proses**

Alur proses dari data transformation ini adalah data sumber yang sudah didapatkan dari API Gateway dari berbagai sumber tersebut masuk ke dalam proses cleaning. Dalam proses ini data dibersihkan dari data kembar dan null value. Setelah itu data masuk dalam proses transform dimana dalam proses ini data akan diubah ke dalam format yang sudah ditentukan dan sesuai dengan data warehouse. setelah data bersih selanjutnya memasukkan data ke dalam data warehouse dengan cara query ke dalam data warehouse.

1. Load

Proses load adalah proses dimana data yang sudah tidak terdapat kecacatan dimasukkan ke dalam database warehouse untuk dilakukan analitik atau untuk menyimpan data historical. Pada sistem yang diterapkan proses extract ini terjadi pada saat perpindahan data dari database staging ke dalam database warehouse.

1. Proses ETL

Proses ETL dimulai dari proses extract dimana pada proses ini data pada database sumber dipindahkan ke database staging. Cara yang digunakan adalah pada sistem database sumber ditanamkan API Agent kemudian API Agent mengkoneksikan dengan API Gateway. API Gateway akan mengirimkan data ke database staging dibantu oleh ETL Tools (Streamset atau Apache Kafka). Pada tahap ini proses extract selesei. Selanjutnya proses yang berjalan adalah proses transform dan cleaning. Proses ini terjadi didalam database staging. Pada database staging data dibersihkan sehingga tidak terdapat kecacatan pada data. Proses selanjutnya adalah proses load. Pada proses ini data pada database staging dikirimkan ke dalam database warehouse. Data yang dikirimkan adalah data yang sudah tidak terdapat kecacatannya saja. Teknik transform yang digunakan adalah teknik pipelined data transform dikarenakan proses transform terjadi di luar database warehouse.

1. Arsitektur API Data Service

Data Warehouse Diskominfo

API Service

Internet

SKPD Client

SKPD Client

SKPD Client

Gambar 3.6 Arsitektur API data service

1. Data Warehouse Diskominfo

Data Warehouse ini adalah database dimana didalamnya terdapat data yang sudah tidak terdapat kecacatan data. Data warehouse ini terdapat pada server Diskominfo. Data yang ada di dalam data warehouse ini digunakan untuk proses analitik dan dimanfaatkan oleh aplikasi client pada SKPD untuk kebutuhan tertentu sesuai dengan izin yang disetujui.

1. API Service

API Service ini adalah service yang disediakan oleh pihak Diskominfo untuk pemanfaatan data yang ada pada data warehouse. Terdapat beberapa service yang disediakan oleh pihak Diskominfo untuk pemanfaatan data. Namun service yang boleh digunakan hanya service yang sudah diberikan izin akses. Format file yang akan dijadikan API adalah file format JSON.

1. SKPD Client

Aplikasi Client SKPD ini adalah aplikasi yang nantinya akan memanfaatkan service yang disediakan oleh pihak Diskominfo. Jika ingin mengakses service API, pihak SKPD terlebih dahulu harus meminta izin kepada SKPD yang bersangkutan kemudian jika diberikan izin akses pihak SKPD tersebut mengkonfirmasikan kepada pihak Diskominfo bahwa SKPD yang ingin mengakses service diperbolehkan untuk mengambil data dengan service yang disediakan oleh pihak Diskominfo. Sehingga pihak Diskominfo mengijinkan SKPD tersebut untuk mengakses servicenya.

1. Penjelasan Arsitektur

API Data Service ini akan melayani permintaan dari aplikasi client yang ada di SKPD melalui API Gateway. API Gateway dapat melakukan authentikasi maupun otorisasi terhadap permintan dari aplikasi Client yang mengaksesnya. Apabila proses authentifikasi dan otorisasi dinyatakan sesuai oleh API Gateway, maka permintaan aplikasi client SKPD akan dapat diteruskan ke API Service. Kemudian API Service akan merespon permintaan tersebut dengan mengirimkan data yang diminta. Data yang dikirimkan berupa file format JSON. File format JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta diterjemahkan dan dibuat oleh komputer.

1. Alur Sistem Request

Gambar 3.7 Alur sistem request

Data Scope

Database Local

Web Service

Aplikasi Integrasi

Web Service

Database Local

Aplikasi SKPD

(1) Register and Login

(2) Response (User Object)

(3) Request Authentication

(6) Response (Access\_token)

(5) Request Token

(4) Grant Authentication

(7) Call REST API Method

(8) Response Data Scope

Gambar 3.7. Alur Sistem Request

1. Register and Login

Dalam proses ini Client yaitu developer dari SKPD harus melakukan registrasi terlebih dahulu ke aplikasi proyek akhir ini karena untuk pengaturan role dan keamanan data untuk orang yang dizinkan saja. Kemudian client melakukan login setelah melakukan registrasi. Ada beberapa role yang terdapat pada aplikasi yaitu :

1. Super Admin

Role ini memiliki semua hak yang ada pada aplikasi dan role ini adalah pemegang kendali dari semua role. Role ini bisa melakukan create update delete pada user dan juga role ini bisa menggunakan semua service yang ada pada aplikasi.

1. Admin

Role ini hampir sama dengan role super admin yang membedakannya adalah role ini tidak diperbolehkan melakukan create delete pada user. Role ini bisa merubah role dari user yang baru saja mendaftar. Role ini bisa menggunakan semua service yang ada pada aplikasi.

1. Developer

Role ini hanya diperbolehkan untuk menggunakan service API tertentu saja. Service API yang diperbolehkan adalah service API dengan data operasional saja.

1. Pegawai

Role ini hanya diperbolehkan menggunakan service API tertentu saja sama seperti role Developer. Namun pada role ini Service API yang diperbolehkan adalah data historical atau data pada data warehouse yang akan digunakan untuk analitik data.

1. Warga

Role hanya diperbolehkan menggunakan satu service saja yaitu service untuk mengetahui apakah biodata WNI benar adanya atau tidak. Untuk mengetahui data biodata WNI tersebut menggunakan NIK.

Role default dari client yang pertama kali mendaftar adalah warga sehingga client harus menghubungi admin terlebih dahulu untuk mengubah role tersebut.

1. Response (User Object)

Setelah login client akan menjadi object sendiri di aplikasi proyek akhir ini. Object tersebut berisi informasi dari client seperti username, password, email, dan lain lain. Data dari object tersebut bisa diubah ubah sesuai keinginan dari client. Client yang pertama kali mendaftar akan otomatis diberikan role warga karena aplikasi ini ditujukan oleh pihak pemerintah. Jika client adalah pihak dari pemerintah yang seharusnya di izinkan untuk menggunakan fitur dari aplikasi ini, client tersebut harus menghubungi admin terlebih dahulu untuk mengubah role dari client tersebut.

1. Request Authentication

Jika role dari client sudah bukan warga, maka client tersebut berhak untuk menggunakan fitur dari aplikasi ini. Client akan mendapatkan application id dan secret id. Semua id tersebut digunakan untuk menggunakan fitur dari aplikasi ini. Setiap client memiliki application id dan secret id yang berbeda dan setiap aplikasi yang menggunakan fitur pada aplikasi ini memiliki application id dan secret id yang berbeda.

1. Grant Authentication

Jika client sudah memiliki application id dan secret id, maka client yang membuat aplikasi dengan menggunakan kedua id tersebut akan bisa menggunakan fitur dari aplikasi ini. Sebelum client bisa menggunakan fitur terdapat pengecekan kecocokan terhadap application id dan secret id. Jika application id dan secret cocok maka client tersebut bisa menggunakan fitur dari aplikasi ini, jika tidak cocok maka authentikasi ditolak.

1. Request Token

Jika Authentikasi sudah disetujui maka user bisa menggunakan fitur dari aplikasi ini. Namun, sebelum menggunakan fitur client tersebut harus memiliki access token untuk bisa menggunakan fitur. Client harus melakukan request token terlebih dahulu kemudian token tersebut disimpan dalam session dalam aplikasi client. Setiap menggunakan fitur terdapat pengecekan access token jika cocok maka bisa menggunakannya. Access token ini terdapat masa kadaluarsanya. Oleh karena itu setiap ingin menggunakan fitur client harus request token terlebih dahulu jikalau token tersebut sudah kadaluarsa.

1. Response Token

Jika client meminta untuk menggunakan fitur aplikasi ini selalu melakukan pengecekan jika access token itu cocok. Jika masa aktif token habis maka aplikasi ini akan otomatis mengirim ulang dengan token yang berbeda kepada aplikasi client. Aplikasi client harus selalu menyimpan token tersebut untuk menggunakan fitur.

1. Call REST API Method

Jika access token dari client dan aplikasi ini cocok maka client berhak untuk menggunakan fitur pada aplikasi ini. Aplikasi ini memiliki fitur untuk mengirimkan data scope sesuai permintaan dari client. Pemanggilan API ini dilakukan dengan cara request method dengan menggunakan request POST/GET ke url yang sudah ditentukan oleh aplikasi ini disertai parameter yang sudah ditentukan juga oleh aplikasi ini sesuai dengan permintaan data scope yang diinginkan dari client.

Terdapat beberapa macam method yang bisa dipanggil. Berikut adalah beberapa method yang tersedia:

1. Method Request token

Method ini adalah method yang digunakan oleh client untuk mendapatkan token agar bisa menggunakan fitur yang ada pada aplikasi ini. Untuk melakukan request token terlebih dahulu harus memiliki application id dan secret id. Token yang sudah diberikan kepada client mempunyai waktu expired sehingga jika sudah masuk waktu expired client harus melakukan request kembali. Response dari request ini adalah access\_token,dan waktu expired. Pemanggilan method bisa di lakukan pada url berikut.

POST : <https://apiduk.jatimprov.go.id/oauth/token>

Dengan parameter yang dikirim adalah

Table 3.1. Parameter Request Token

|  |  |  |  |
| --- | --- | --- | --- |
| Bagian | Key | Value | Desciption |
| Header | Content-Type | application/x-www-form-urlencoded |  |
| Body | Grant\_type | Client\_credentials |  |
| Client\_id | <client\_id> |  |
| Client\_secret | <client\_secret |  |

1. Method Mengetahui Authorization Pemilik Aplikasi

Method ini adalah method yang digunakan oleh client untuk mengetahui kepunyaan dari aplikasi yang sudah diauthorization. Untuk mendapatkan kepemilikan aplikasi dan sudah diauthorization, caranya adalah dengan membuat halaman callback sehingga pada aplikasi tersebut memiliki Authorization Code. Response dari request ini adalah pemilik aplikasi, authorization code, dan scope data yang digunakan. Pemanggilan method ini dilakukan pada url berikut.

GET : <https://apiduk.jatimprov.go.id/api/v1/me>

Dengan parameter yang dikirim adalah

Table 3.2. Parameter Untuk Authorization Pemilik Aplikasi

|  |  |  |  |
| --- | --- | --- | --- |
| Bagian | Key | Value | Description |
| Header | Authorization | Bearer <access\_token> |  |
| Body | - | - | - |

1. Method Request Informasi Biodata WNI

Method ini adalah method yang digunakan oleh client untuk mengetahui informasi biodata dari warga negara Indonesia. Untuk memakai method ini client harus mempunyai NIK yang digunakan untuk memeriksa data yang ada pada database Dispenduk Jawa Timur. Response dari request ini adalah biodata wni sesuai dengan NIK. Pemanggilan method ini dilakukan pada url berikut.

GET: [https://apiduk.jatimprov.go.id/api/v1/biodata\_wni/<NIK](https://apiduk.jatimprov.go.id/api/v1/biodata_wni/%3cNIK)>

Dengan parameter yang dikirim adalah

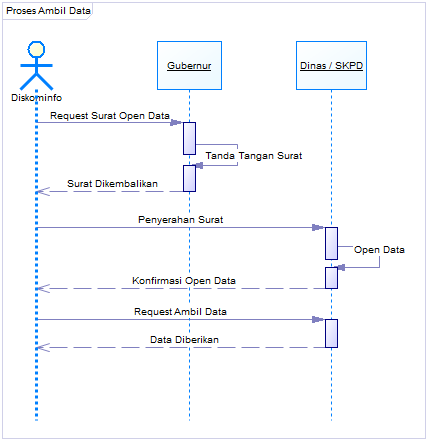
Table 3.3. Parameter Request Informasi Biodata WNI

|  |  |  |  |
| --- | --- | --- | --- |
| Bagian | Key | Value | Description |
| Param Get | Nik | <NIK> |  |
| Header | Authorization | Bearer <access\_token> |  |
| Body | - | - | - |

1. Response Data Scope

Setelah client melakukan request maka aplikasi ini harus menjawab sesuai kebutuhan dari client. Aplikasi ini mengirimkan data scope sesuai yang di inginkan oleh client melalui file JSON. Data yang dikirim hanya sebatas data scope yang sudah di atur oleh aplikasi ini.

1. SOP Alur Pengambilan Data



Gambar 3.8 SOP alur pengambilan data

1. Diskominfo

Diskominfo adalah pihak yang ingin meminta akses ke semua data dari seluruh SKPD. Data tersebut akan digunakan untuk proses analitik dan juga data tesebut bisa dimanfaatkan untuk kebutuhan tertentu oleh SKPD lain. Diskominfo berperan sebagai penyedia data analitik dan sebagai pintu gerbang untuk pemanfaatan data.

1. Gubernur

Gubernur adalah kepala dari semua SKPD. Gubernur disini berperan sebagai pengambil keputusan tertinggi. Pihak Diskominfo menyediakan data beserta analitiknya kemudian Gubernur akan mengambil keputusan dan semua dinas harus mengikuti keputusan tersebut.

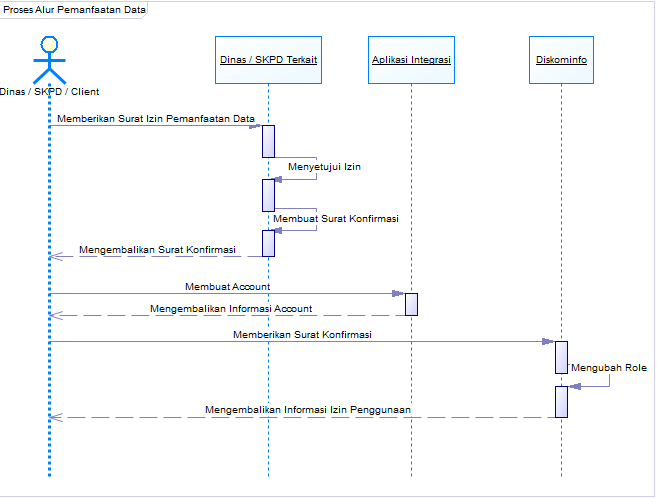
1. Dinas / SKPD

Dinas / SKPD adalah organisasi dibawah kepemimpinan Gubernur yang bertugas untuk mensejahterakan rakyatnya. Dinas / SKPD berperan untuk mewujudkan keputusan Gubernur dengan menyediakan pelayanan kepada masyarakat. Dinas disini memiliki data yang dibutuhkan oleh pihak Diskominfo untuk proses analitic dan akan dimanfaatkan oleh pihak tertentu untuk kebutuhan tertentu.

1. Alur Pengambilan Data

Langkah pertama dari pengambilan data adalah pihak dari Diskominfo mengajukan surat permohonan kepada Gubernur bahwa pihak SKPD diharapkan untuk membuka datanya yang kemudian akan diambil oleh pihak Diskominfo. Jika Gubernur menyetujui surat tersebut maka Gubernur akan menanda tangani surat permohonan tersebut. Setelah itu surat tersebut dikembalikan kepada pihak Diskominfo. Pihak Dikominfo akan menyerahkan surat tersebut kepada seluruh SKPD. Pihak SKPD diharus kan untuk membuka datanya. Setelah itu pihak Diskominfo berhak untuk mengambil data yang sudah dibuka tersebut.

1. SOP Alur Pemanfaatan Data



Gambar 3.9 SOP alur pemanfaatan data

1. Client

Pihak Client ini adalah pihak dari SKPD atau developer lain yang ingin mengakses service dan memanfaatkan data. Service dapat diakses jika client sudah melalui beberapa prosedur dan diizinkan untuk mengakses service. Data yang sudah didapatkan oleh pihak client akan dimasukkan dan dimanfaatkan pada aplikasi pihak client.

1. SKPD / Dinas Terkait

Dinas / SKPD ini adalah pemegang data sumber atau bisa disebut data wali dimana data tersebut adalah data yang terjaga kerahasiaannya. SKPD / Dinas ini berperan sebagai pihak yang mengkonfirmasi dan memastikan bahwa data yang akan dimanfaatkan oleh pihak client tidak digunakan secara sembarangan. Jika pihak client melanggar aturan penggunaan data maka pihak Dinas terkait juga harus bertanggung jawab atas pelanggaran tersebut.

1. Aplikasi Integrasi

Aplikasi Integrasi ini adalah aplikasi utama yang digunakan untuk mengakses service dan penyalur data kepada client yang kemudian oleh client akan dimanfaatkan data tersebut. Aplikasi ini berperan sebagai pintu gerbang untuk mengakses service dan sebagai keamanan data agar data tidak dicuri oleh pihak yang tidak bertanggung jawab.

1. Alur Pemanfaatan Data

Langkah pertama untuk pemanfaatan data adalah pihak Client mengirimkan surat permintaan untuk akses data kepada SKPD yang ingin dimanfaatkan datanya. Jika surat tersebut disetujui oleh pihak SKPD terkait, pihak SKPD tersebut akan membuat surat konfirmasi bahwa pihak SKPD terkait mengizinkan untuk datanya diakses oleh Client. Kemudian Client harus membuat account pada aplikasi integrase. Selanjutnya pihak client menyerahkan surat konfirmasi tersebut kepada pihak Diskominfo. Pihak Diskominfo merubah role status dari client ke role status yang boleh mengakses service dari aplikasi integrasi. Pihak Diskominfo akan menginformasikan kepada pihak client bahwa pihak client dapat mengakses service yang ada pada aplikasi integrasi.

1. PERBANDINGAN SOAP DAN REST

Web service adalah platform standard yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem. Aplikasi yang melakukan pertukaran data tersebut memiliki perbedaan bahasa pemrograman dan juga berjalan pada platform yang berbeda beda. Web service memiliki arsitektur dan protocol untuk pertukaran data itu sendiri. Arsitektur tersebut adalah SOAP dan REST. Berikut adalah penjelasan dan perbandingan dari SOAP dan REST:

1. SOAP (Simple Object Access Protocol)

Arsitektur SOAP adalah salah satu implementasi dari web service pada jaringan komputer. Tujuan dari arsitektur SOAP ini adalah bertukar informasi antara satu sistem dengan sistem lain melalui protocol tertentu. Arsitektur SOAP ini menggunakan format XML untuk informasi yang dikirimnya. Arsitektur SOAP memiliki tiga karakteristik yaitu 1) *extensibility* (bisa diperpanjang), 2) *neutrality* (bisa berjalan pada semua protocol), dan 3) *independence* (bisa memungkinkan untuk menggunakan semua model pemrograman). Arsitektur SOAP ini menggunakan WSDL dan UDDI, keduanya adalah bagian dalam SOAP, memungkinkan developer untuk menemukan layanan web yang tersedia, dan memungkinkan web service untuk ditemukan melalui internet dengan menggunakan UDDI, serta memudahkan web service client untuk menggunakan toolkit dan compiler SOAP melalui WSDL [13]. Arsitektur ini bisa menggunakan banyak bahasa pemrograman, dan juga bisa berjalam diberbagai sistem operasi. Implementasi dari service SOAP semua didefinisikan pada web service definition language (WSDL). File WSDL terdapat tipe skema XML yang mencerminkan model dari server-side data model. Tipe tersebut memetakkan struktur parameter yang dapat dimasukkan ke dalam service request ataupun service response [12]. File WSDL perlu untuk dijadikan public untuk semua aplikasi client yang berhubungan sehingga aplikasi tersebut bisa menjangkau service pada server.

1. REST (Representational State Transfer)

Arsitektur REST adalah salah satu arsitektur implementasi untuk web service yang digunakan untuk bertukar data dari satu sistem ke sistem lain. Arsitektur web service REST ini sebaiknya diimplementasikan ketika web service dalam kondisi stateless yang artinya pengirim informasi tidak perlu mengirimkan informasi untuk disimpan oleh server, sebagai contoh client tidak perlu mengirimkan informasi IP untuk disimpan oleh server [13]. Arsitektur REST ini menggunakan berbagai format untuk mengirim informasi seperti XML, JSON, HTML dan format lain. Arsitektur REST ini mementingkan kecepatan performance, kehandalan dan memiliki kemampuan untuk berkembang karena arsitektur ini menggunakan stateless protocol dan operasi yang standard. Semua resource pada arsitektur REST ditandai dengan URL yang dapat dioperasikan oleh subset dari perintah HTTP yaitu POST, GET, PUT, dan DELETE [12]. Arsitektur ini bisa menggunakan banyak bahasa pemrograman, dan juga bisa berjalam diberbagai sistem operasi. Aplikasi client mengirimkan perintah HTTP pada URL tertentu pada domain server, kemudian server membalasnya dengan file informasi dengan format tertentu langsung ke aplikasi client.

1. Perbandingan SOAP dan REST

Terdapat perbandingan antara arsitektur web service SOAP dengan arsitektur web service REST. Keduanya memiliki kelebihan dan kekurangannya tersendiri. Menurut penelitian yang dilakukan oleh Gavin Mulligan dan Denis Gracanin pada tahun 2009, uji coba dilakukan dengan membuat sebuah aplikasi portal yang bisa digunakan oleh aplikasi client untuk menggunakan service dari server. Aplikasi tersebut menerapkan kedua arsitektur yaitu SOAP dan REST. Hasil dari penelitian tersebut adalah implementasi dengan menggunakan arsitektur REST lebih efisien dalam perbandingan efisiensi bandwidth jaringan dan kecepatan akses daripada arsitektur SOAP.

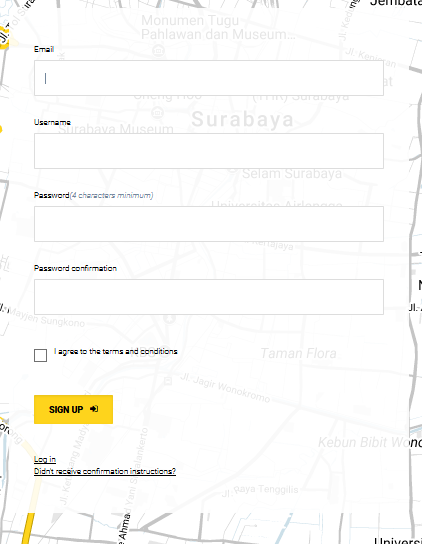
Penelitian lain dilakukan pada tahun 2011 oleh Adrian Mullally, Migei McKelvey dan Kevin Curran. Penelitian yang dilakukan adalah mereka membuat web service yang kemudian akan diakses oleh sistem operasi mobile. Uji coba dilakukan di beberapa platform yaitu mobile web, android, dan windows phone. Hasil dari penelitian tersebut adalah waktu yang dibutuhkan platform mobile untuk menerima respon dengan menggunakan arsitektur REST lebih cepat dan lebih efisien daripada arsitektur SOAP. Dari perbandingan yang didapat oleh penelitian tersebut didapatkan bahwa arsitektur REST lebih cepat daripada arsitektur SOAP. Dengan melihat data dari pemerintahan yang sangat banyak, penelitian ini menggunakan arsitektur REST dalam implementasi, karena dengan menggunakan arsitektur REST ini pengiriman data jadi lebih cepat dan efisien.

1. USER INTERFACE

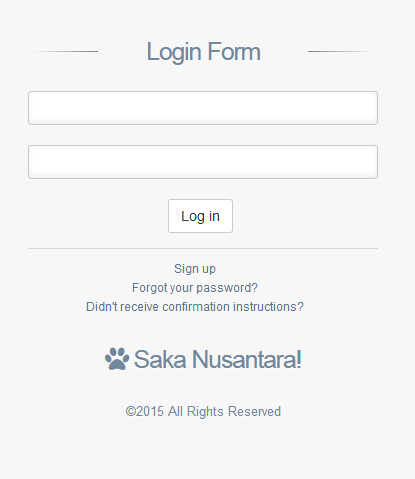
Pada user interface ini akan dijelaskan beberapa bagian dari aplikasi. Role yang digunakan adalah role admin dikarenakan admin bisa mengakses seluruh fitur

1. Register dan Login

Untuk menggunakan fitur dari aplikasi ini terlebih dahulu harus mendaftar dan melakukan login. Untuk mendaftar cukup mengisikan beberapa informasi. Informasi lainnya bisa di tambahkan saat sudah masuk ke aplikasi. Untuk melakukan login cukup masukkan email dan password. Gambar 3.10 adalah user interface dari register dan gambar 3.11 adalah user interface dari login.



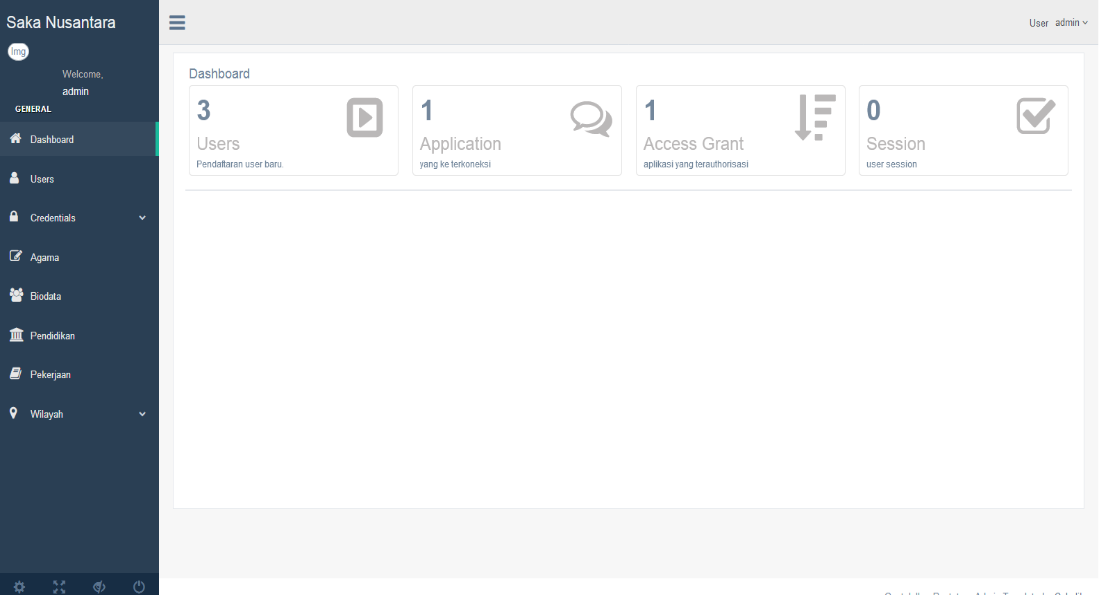
Gambar 3.10 Tampilan Registrasi



Gambar 3.11 Tampilan Login

1. Dashboard

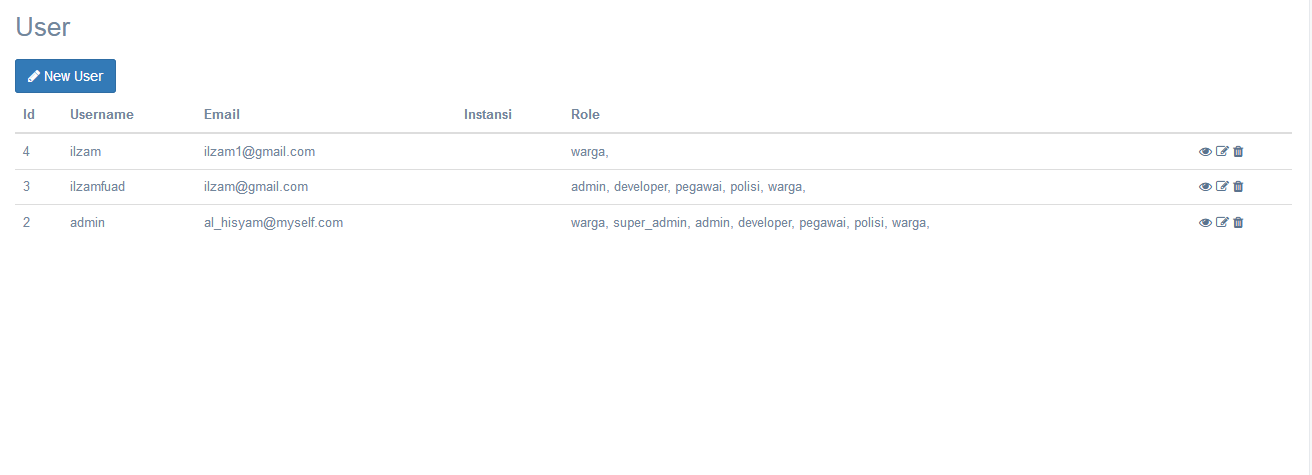
Setelah login akan masuk ke halaman dashboard. Halaman ini berisikan jumlah dari pengguna yang memakai aplikasi ini dan menggunakan fitur dari aplikasi ini. Selain itu untuk role admin bisa mengatur semua pengaturan. Gambar 3.12. Adalah gambar dari tampilan dashboard. Bisa dilihat di samping kirim gambar 3.12 terdapat pengaturan yang bisa diatur oleh admin.



Gambar 3.12. Tampilan Dashboard

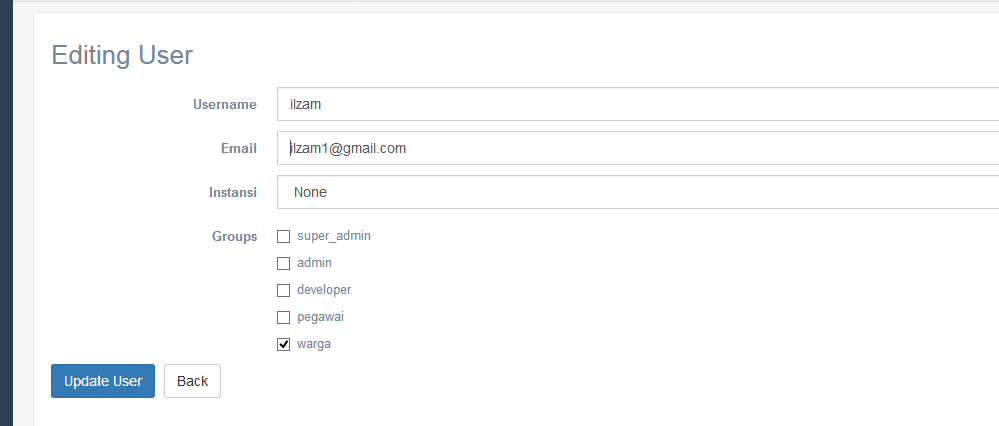
Gambar 3.12 Tampilan Dashboard

1. Pengaturan user

Pada halaman pengaturan user ini admin bisa menambahkan user, mengubah pengaturan user atau menghapus user. Admin juga bisa mengubah role dari user tertentu sehingga user tersebut bisa menggunakan fitur sesuai rolenya. Gambar 3.13. adalah gambar dari list user. Admin bisa mengubah role dari user. Gambar 3.14 adalah cara untuk mengubah role untuk user.

Gambar 3.13 Tampilan list user

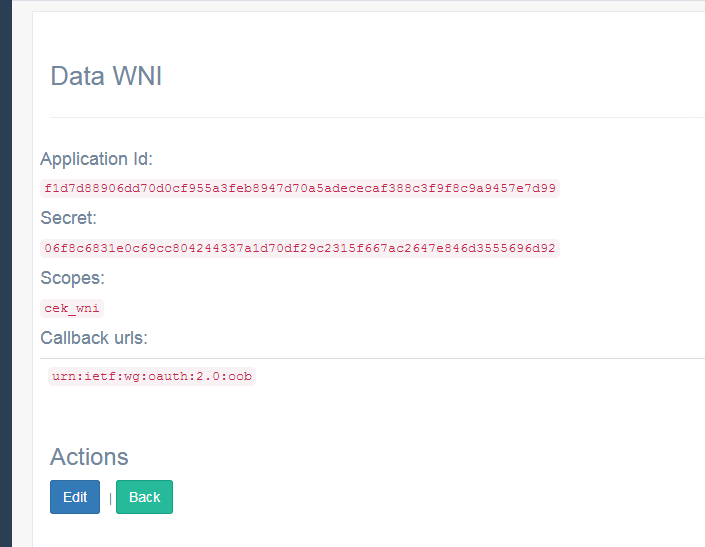
Gambar 3.13. Tampilan list User



Gambar 3. 14 Role use

1. Pengaturan Credential

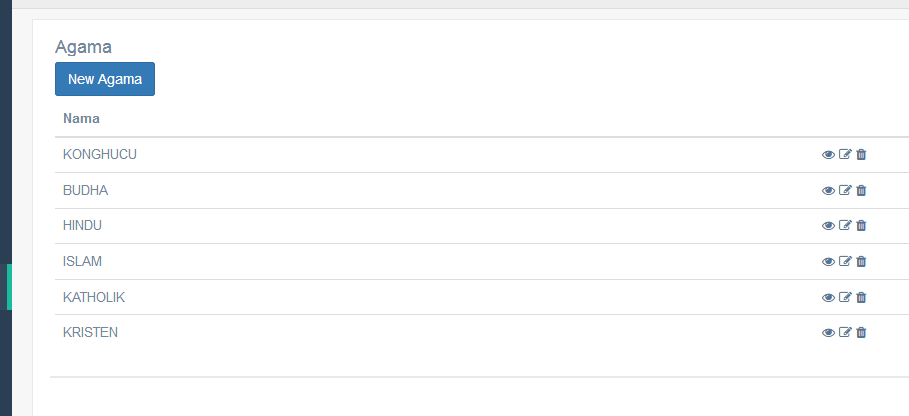
Pada halaman pengaturan credential ini digunakan untuk membuat authorization. Setiap aplikasi akan mendapat application id dan secret id yang digunakan untuk mengaskses fitur yang ada pada aplikasi ini. Gambar 3.15. adalah gambar dari kebutuhan untuk authorization.



Gambar 3.15 Kebutuhan Authorization

1. Pengaturan Agama

Pada halaman pengaturan agama ini digunakan untuk menambah, mengubah atau menghapus database dari agama. Gambar 3.16. adalah tampilan list dari agama.



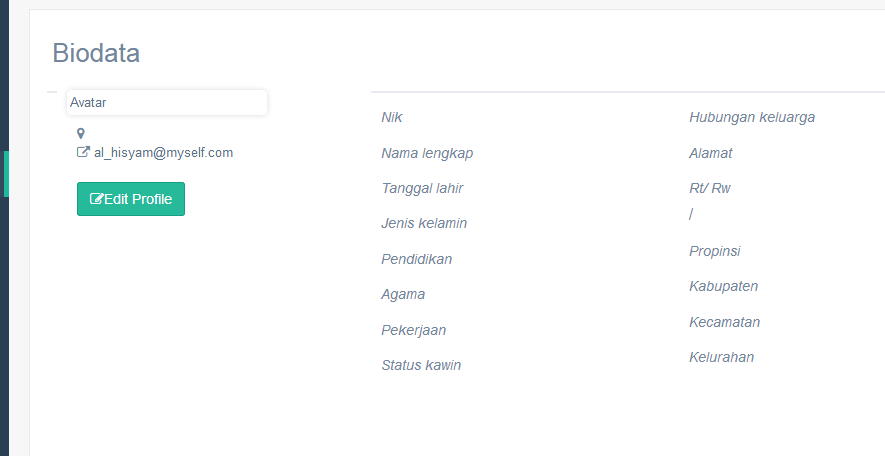
Gambar 3.16 Tampilan list agama

1. Pengaturan Lainnya

Halaman pengaturan lainnya sama dengan halaman pengaturan untuk agama. Admin bisa menambah, mengubah, atau menghapus data dari database aplikasi.

1. Profile / Biodata

Profile adalah informasi diri dari user. User bisa mengisi profile untuk melengkapi data dirinya. Gambar 3.17. adalah tampilan dari halaman profile.



Gambar 3.17 Tampilan profile

# **BAB IV**

UJI COBA DAN ANALISIS

Dalam bab ini akan dibahas mengenai hasil pengujian terhadap aplikasi dan analisa yang diberikan setelah ujicoba. Uji coba dilakukan terhadap sistem secara menyeluruh dengan kondisi yang akan diberikan. Setelah proses uji coba selesai maka hasil yang akan didapatkan akan dianalisa.

1. TUJUAN ANALISA

Tujuan dari analisa ini adalah untuk membuktikan bahwa fitur dari aplikasi yang berupa API bisa digunakan oleh aplikasi client yaitu aplikasi buatan developer dari pemerintah Jawa Timur. Sehingga sistem yang ada pada pemerintah Jawa Timur bisa terintegrasi satu sama lain dan juga data yang digunakan oleh pemerintah Jawa Timur akan bersifat multisektoral.

1. PERANGKAT UJI COBA

Perangkat uji coba untuk proyek akhir ini meliputi :

Table 4.1. Spesifikasi Uji Coba

|  |  |  |
| --- | --- | --- |
| No | Deskripsi | Spesifikasi |
| 1 | Processor | Intel® Core™ i7-2620M CPU @ 2.70Ghz |
| 2 | RAM | 8.00 GB |
| 3 | System Operation | Windows 8.1 Pro |
| 4 | Harddisk | 500 GB |

1. DATA

Data yang digunakan dalam uji coba ini adalah data kependudukan dari Dispenduk (Dinas Kependudukan) Jawa Timur. Data ini didapatkan dari database oracle yang berada pada server Dispenduk Jawa Timur.

1. SKENARIO UJI COBA

Pengujian yang akan dilakukan terhadap aplikasi integrase data ini adalah sebagai berikut.

1. Pendaftaran dan Mendapatkan Application Credentials

Uji coba ini dilakukan dengan mendaftar pada aplikasi integrasi kemudian membuat application credensial. Application credential itu yang kemudian digunakan untuk mengakses fitur pada aplikasi integrase. Application credential tersebut adalah application id dan secret id.

1. Pengujian Request Token dan Request API Menggunakan Postman

Application credential yang sudah didapatkan dari aplikasi integrasi akan dicoba pada aplikasi postman. Aplication credential akan diubah menjadi access token. Kemudian access token tersebut digunakan untuk memanfaatkan fitur yang ada pada aplikasi integrasi. Setelah mendapatkan access token, selanjutnya adalah mencoba memakai fitur dari aplikasi dengan cara request method REST API yang akan menghasilkan data scope yang di inginkan.

1. Penggunaan SDK dan Unit Testing

Setelah mengetahui proses dari pengambilan data API. Selanjutnya adalah implementasi proses pengambilan data tersebut ke dalam sebuah aplikasi. Untuk mempermudah proses pengambilan data, aplikasi tersebut ditambahkan dengan SDK. Sebelum SDK tersebut diterapkan ke dalam aplikasi terlebih dahulu dilakukan unit testing. Unit testing adalah proses pengecekan fitur SDK apakah semua fitur sudah berjalan atau belum. Proses unit test sudah disediakan oleh SDK sehingga developer bisa mencoba semua fitur dari SDK dengan parameter yang disesuaikan.

1. Implementasi pada Aplikasi

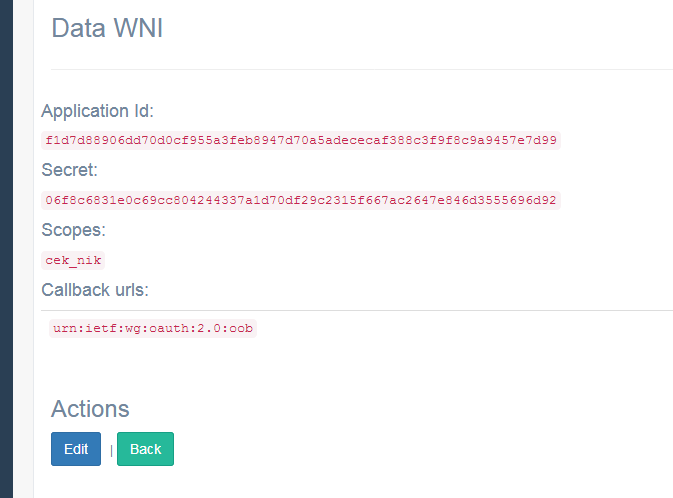
Setelah semua fitur dalam SDK sudah berhasil semua. Langkah selanjutnya adalah memasukkan SDK tersebut kedalam aplikasi untuk mengambil data API. SDK tersebut disesuaikan dengan program yang sudah ada. Tugas dari SDK adalah membantu developer untuk mengambil data API dari aplikasi integrase. Setelah melalui proses pengambilan data, kemudian data yang berhasil didapatkan tersebut diproses dalam sistem dan dimanfaatkan oleh sistem. Data yang dikirim oleh aplikasi integrase ini berupa JSON. Data JSON ini bisa diproses menggunakan semua Bahasa pemrograman.

1. PENGUJIAN SISTEM

Pengujian yang sudah dilakukan adalah sebagai berikut :

1. Pendaftaran dan Mendapatkan Token

Pengujian sudah dilakukan dengan mendaftar pada aplikasi integrase dan mendapatkan application id dan secret id. Gambar 4.1. adalah application credential yang dipakai uji coba.



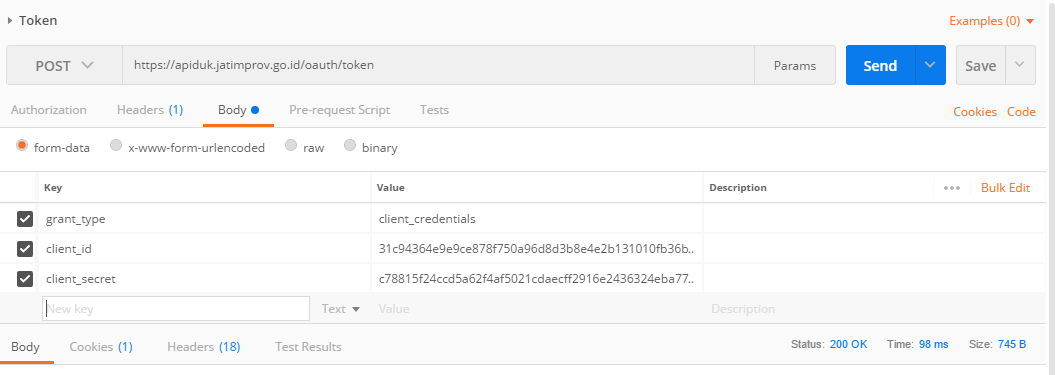
Gambar 4.1 Application Credential untuk uji coba

1. Pengujian Request Token dan Request API Menggunakan Postman

Pengujian dilakukan dengan aplikasi postman untuk pengambilan request token dan request pengambilan data API. Terdapat 2 proses untuk mendapatkan data tersebut. Proses tersebut akan dijelaskan sebagai berikut.

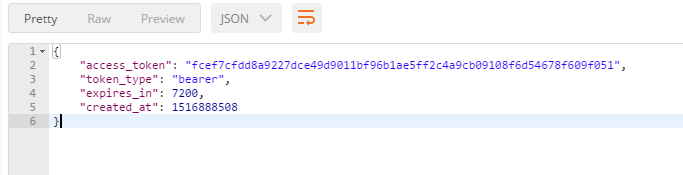
* + - 1. Proses Request Access Token

Proses pertama ini adalah proses untuk mendapatkan access token. Pada proses ini aplikasi client akan memanggil method dan membutuhkan beberapa parameter yang dibutuhkan. Request dalam proses ini adalah request POST sehingga aplikasi client memerlukan body untuk mengirimkan parameternya.



Gambar 4.2 Parameter dan url untuk request token

Pada Gambar 4.2 adalah url dan parameter yang dibutuhkan untuk melakukan request kepada server. End point pada url untuk mendapatkan access token adalah /oauth/token dan parameter yang dikirimkan bersama url adalah client\_id, client\_secret dan grant\_type.

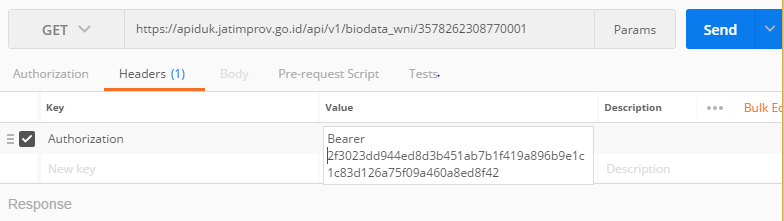


Gambar 4.3 Response dari request token

Pada Gambar 4.3 adalah response dari server berupa access token. Access token tersebut digunakan untuk mengakses service pada server. Proses request dan response tersebut membutuhkan waktu 98 ms.

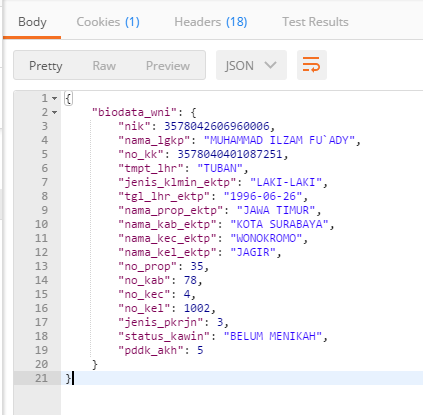
* + - 1. Proses Pemanggilan API dan Mendapatkan Data

Pada proses ini access token yang sebelumnya didapatkan akan digunakan untuk menggunakan fitur pada aplikasi integrasi. Pada proses ini aplikasi client akan memanggil method dan membutuhkan beberapa parameter yang dibutuhkan. Pada percobaan ini, data yang di request adalah data scope dari warga negara Indonesia yang berada pada server DispendukJawa Timur. Request dalam proses ini adalah request GET sehingga parameter yang dibutuhkan disukkan ke dalam url.



Gambar 4.4 Parameter dan url untuk request API

Gambar 4.4 adalah url dan parameter yang diperlukan untuk proses request pada server. End point pada url untuk mendapatkan menggunakan service adalah /api/v1/biodata\_wni/ <NIK> dan parameter yang dikirimkan bersama url adalah authorization berupa access token yang sudah didapatkan dari request sebelumnya.



Gambar 4.5 Response dari request API

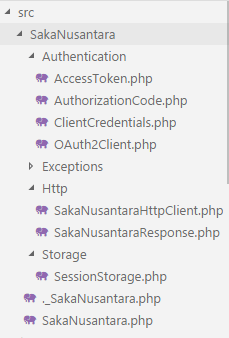
Gambar 4.5 adalah hasil response yang dikirimkan oleh server berupa format JSON. Data JSON tersebut kemudian akan dimanfaatkan oleh aplikasi client untuk kebutuhan tertentu. Proses request dan response tersebut membutuhkan waktu 2216 ms.

1. Penggunaan SDK dan Unit Testing

Penjelasan SDK dan fiturnya beserta proses unit testing adalah sebagai berikut :

1. SDK

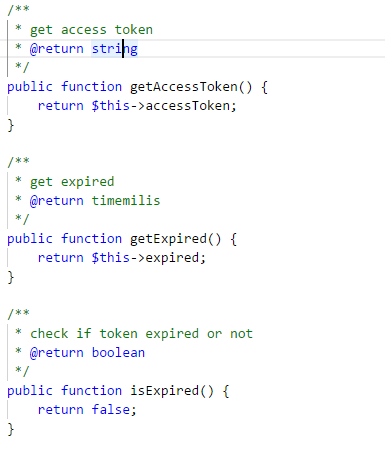
Dalam pemanfaatan data diperlukan cara untuk mendapatkan data dari API. SDK adalah alat bantu untuk developer mendapatkan data. SDK akan dimasukkan dan disesuaikan dengan aplikasi yang sudah ada. SDK membuat proses pengambilan data menjadi lebih mudah. Gambar 4.6 adalah Direktori SDK yang digunakan.



Gambar 4.6 Direktori SDK

Berikut adalah fitur fitur yang terdapat pada SDK :

1. **Access Token**

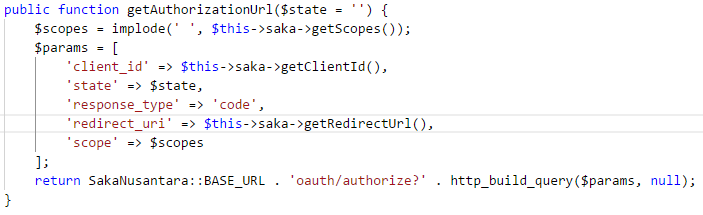


Gambar 4.7 Fitur access token

Pada fitur access token ini bertujuan untuk menyimpan access token yang kemudian akan digunakan untuk mengakses service yang tersedia pada server. Access token ini mempunyai masa waktu tersendiri, jika masa waktu tersebut habis maka access token pada server akan dihapus. Jika access token habis maka sdk akan meminta access token lagi. Pada fitur access token ini menyediakan pengecekan masa aktif access token. Terdapat 2 cara untuk mendapatkan access token yaitu dengan melalui authorization code dan melalui client credential.

1. **Authorization Code**

Fitur authorization code adalah salah satu fitur untuk mendapatkan access token. Fitur ini digunakan untuk mengidentifikasi bahwa aplikasi yang akan menggunakan service pada server benar sudah terdaftar dan mengkonfirmasi kepemilikan dengan menggunakan code. Untuk mendapatkan code harus melalui url tertentu untuk authorization. Authorization Code digunakan untuk mendapatkan access token kemudian bisa menggunakan service pada server.



Gambar 4. 8 Fitur get Authorization Url

Pada gambar 4.8 adalah fitur untuk mendapatkan url dimana pada url tersebut terdapat konfirmasi kepemilikan akun dan konfirmasi bahwa sudah terdaftar. Pada url tersebut terdapat beberapa parameter didalamnya yaitu client id adalah id yang sudah terdaftar dalam aplikasi server, state, response type yang di isi code karena url tersebut adalah untuk mendapatkan code, redirect uri adalah url yang digunakan untuk melakukan callback, dan scope adalah service apa yang di inginkan.



Gambar 4.9 get Access Token melalui Authorization Code

Pada gambar 4.9 adalah fitur untuk mendapatkan access token dengan melalui authorization code. Access token tersebut akan digunakan untuk akses service yang terdapat pada server. Untuk mendapatkan access token diperlukan request berupa post dengan url end point oauth/token dengan beberapa parameter yang disertakan didalamnya. Parameter yang disertakan adalah client id adalah id yang sudah terdaftar dalam aplikasi server, client secret adalah id kedua yang sudah terdaftar dalam server, grant type adalah untuk mendapatkan access code dengan cara tertentu, pada fitur ini cara yang digunakan adalah melalui authorization code, redirect\_uri adalah url yang digunakan untuk melakukan callback, dan code adalah code yang digunakan untuk mendapatkan access token, pada fitur ini code yang digunakan adalah authorization code.

1. **Client Credential**

Fitur client credential adalah salah satu fitur untuk mendapatkan access token. Hal yang memberdakan fitur Client credential dan fitur authorization code adalah fitur ini tidak memerlukan code untuk mendapatkan access token. Fitur ini hanya memerlukan client id dan client secret saja.



Gambar 4.10 Fitur get Access Token melalui Client Credential

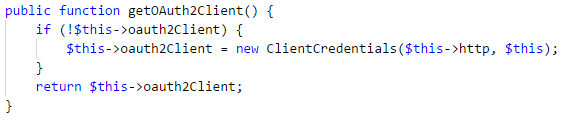
Pada gambar 4.10 adalah fitur untuk mendapatkan access token dengan melalui client credential. . Access token tersebut akan digunakan untuk akses service yang terdapat pada server. Untuk mendapatkan access token diperlukan request berupa post dengan url end point oauth/token dengan beberapa parameter yang disertakan didalamnya. Parameter yang disertakan adalah client id adalah id yang sudah terdaftar dalam aplikasi server, client secret adalah id kedua yang sudah terdaftar dalam server, grant type adalah untuk mendapatkan access code dengan cara tertentu, pada fitur ini cara yang digunakan adalah melalui client credentials, dan scope adalah service apa yang di inginkan.

1. **Fitur yang digunakan**

Fitur utama yang digunakan oleh SDK ini adalah fitur client credential karena dalam fitur ini memiliki kemudahan dalam mengatur parameter dan SDK ini menyimpan semua parameter yang dibutuhkan sehingga client tidak perlu melihat lagi ke dalam aplikasi pada server.

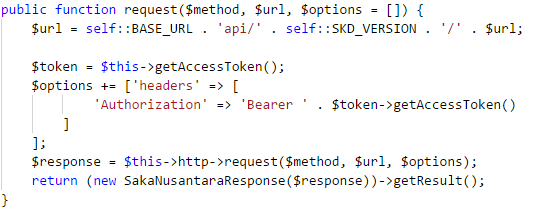


Gambar 4.11 Inisialisasi get Access Token



Gambar 4.12 Request menggunakan Client Credential

Pada gambar 4.11 adalah cara bagaimana mendapatkan access token. Dapat dilihat pada gambar 4.12 bahwa fitur utama yang digunakan oleh SDK ini adalah fitur client credential. Setelah mendapatkan access token, access token tersebut disimpan dalam sebuah storage. Saat terjadi pemanggilan service access token tersebut dilakukan pengecekan pada storage apakah masa aktik dari access token masih berlaku atau sudah mati. Jika access token tersebut masih berlaku maka access token tersebut digunakan kembali untuk pemanfaatan service, jika access token tersebut sudah mati maka access token tersebut dihapus dan SDK akan melakukan request access token yang baru.



Gambar 4.13 Inisialisasi request

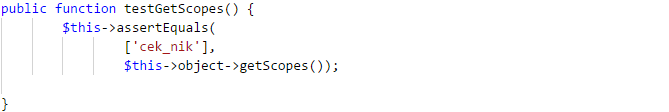
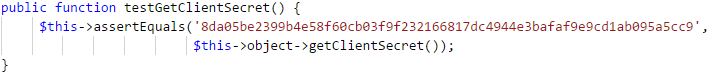
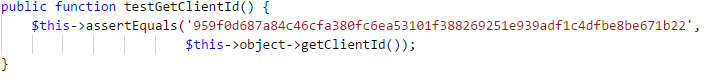
Pada gambar 4.13 adalah cara bagaimana SDK untuk mengambil service dengan menggunakan access token. Service diambil pada url tertentu dengan method tertentu yang sudah disiapkan oleh server. Kemudian request tersebut dikirimkan beserta parameter yang menyertakan access token didalamnya. Setelah melakukan request, server membalas request tersebut dengan response berupa JSON. Lalu JSON tersebut dikembalikan ke aplikasi client yang kemudian data dari JSON tersebut dimanfaatkan oleh aplikasi client.

1. Unit Testing

Unit Testing adalah fase dalam proses development dimana dalam proses ini semua fitur dalam aplikasi dicoba satu persatu. Tujuan dari unit testing ini adalah untuk mendapatkan hasil dan error dalam berbagai percobaan pada setiap fitur yang ada. Unit testing bisa dilakukan secara manual ataupun secara otomatis. Pada penelitian ini dilakukan unit testing pada SDK. Unit testing dilakukan secara otomatis dengan inputan parameter manual. Setiap Fitur dan fungsi dari SDK semua dites satu persatu kemudian dihitung kecepatan untuk mencoba semua fitur.

Berikut adalah fitur fitur yang akan dilakukan unit testing pada SDK :

1. **Parameter Testing**

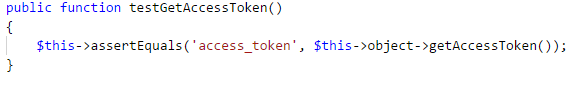


Gambar 4.14 Test Parameter

Pengecekan dilakukan untuk mengetahui apakah parameter yang digunakan sudah tepat. Pengecekan dilakukan pada semua parameter yang digunakan untuk mendapatkan access token. Parameter tersebut adalah client\_id, secret\_id, redirect\_uri, dan scope.

1. **Access Token**

Pengecekan dilakukan pada fitur access token. Pengecekan ini dilakukan untuk mengetahui apakah proses request access token berhasil mendapatkan access token dan tidak kosong. Access token yang didapatkan ini nantinya akan digunakan untuk mengakses salah satu service yang tersedia pada server.

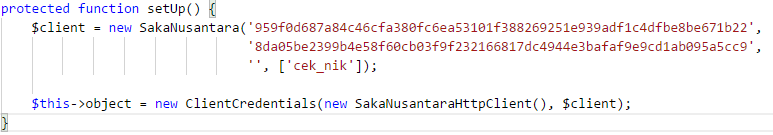


Gambar 4.15 Test Access Token

Pada gambar 4.15 adalah fungsi untuk melakukan pengecekan ada tidaknya access token setelah melakukan request access token. Pengecekan access token melalui dua cara yaitu melalui authorization code dan melalui client credential. Access token yang sudah didapatkan disimpan dalam storage dari SDK yang nantinya akan digunakan untuk mengakses salah satu service yang tersedia dalam server.

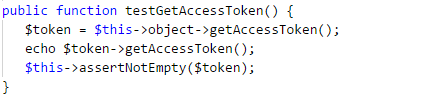
1. **Client Credential**

Pengecekan dilakukan pada fitur request access token menggunakan cara Client Credential. fitur ini digunakan untuk mendapatkan access token dengan hanya menggunakan parameter client\_id, secret\_id, redirect\_uri, dan scope.



Gambar 4.16 Inisialisasi testing Client Credential

Pada gambar 4.16 adalah fungsi untuk mengatur inisialisasi request access token. Kemudian setelah dilakukan inisialisasi testing dimulai untuk request access token. Parameter dalam inisialisasi adalah client\_id, secret\_id, redirect\_uri, dan scope.

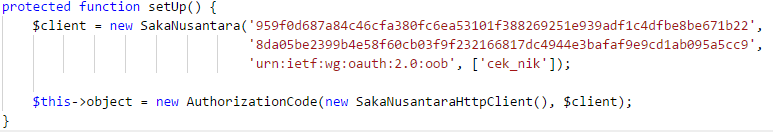


Gambar 4.17 Test get Access Token Client Credential

Pada gambar 4.17 adalah inisialisasi untuk mendapatkan access token. Pengecekan ini dilakukan apakah access token benar benar didapatkan dan tidak kosong.

1. **Authorization Code**

Pengecekan dilakukan pada fitur request access token menggunakan cara Authorization Code. fitur ini digunakan untuk mendapatkan access token dengan hanya menggunakan parameter client\_id, secret\_id, redirect\_uri, dan scope. Namun untuk request access token terdapat parameter tambahan yaitu authorization code. Authorization code bisa didapatkan melalui aplikasi server.



Gambar 4.18 Inisialisasi testing Authorization Code

Pada gambar 4.18 adalah fungsi untuk mengatur inisialisasi request access token. Kemudian setelah dilakukan inisialisasi testing dimulai untuk request access token. Parameter dalam inisialisasi adalah client\_id, secret\_id, redirect\_uri, dan scope.



Gambar 4.19 Testing Authorization Url

Pada gambar 4.19 adalah pengecekan url dimana dalam url tersebut adalah tempat untuk authorization yang kemudian mendapatkan authorization. Authorization code tersebut akan digunakan untuk mendapatkan access token.



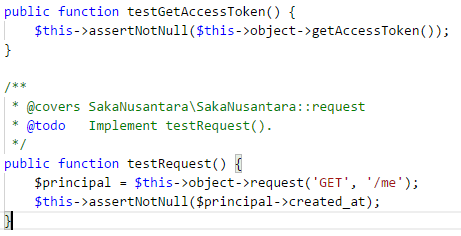
Gambar 4.20 Test get Access Token Authorization Code

Gambar 1

Pada gambar 4.20 adalah untuk request dari access token. Pada pengecekan ini dilakukan apakah access token tersebut bisa didapatkan dan tidak kosong. Untuk mendapatkan access token tersebut dengan menggunakan authorization code. Authorization code tersebut didapatkan dari url yang sudah dilakukan pengecekan sebelumnya. Setelah access token didapatkan, access token tersebut akan digunakan untuk mengakses salah satu service yang terdapat pada server.

1. **Request Service**

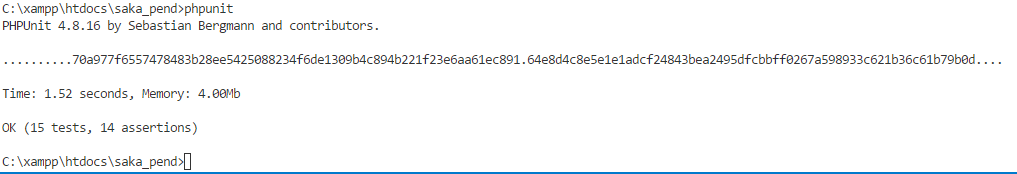
Pengecekan dilakukan pada fitur request salah satu service yang terdapat pada server. Request service tersebut menggunakan access token yang sudah didapatkan sebelumnya. Percobaan service yang akan dilakukan adalah service dengan end point /me dimana pada service tersebut adalah service untuk mengetahui kepemilikan dari client\_id dan secret\_id.



Gambar 4.21 Testing Request

Pada gambar 4.21 adalah fungsi untuk membuat request ke end point /me dengan menggunakan access token. Service yang akan dilakukan tes ini adalah service untuk mengetahui kepemilikan dari client id dan secret id. Access token didapatkan dari storage kemudian dengan menggunakan access token tersebut request dilakukan. Pengecekan dilakukan dengan mengambil salah satu nilai pada response apakah berhasil didapat atau tidak.

1. **Hasil Unit Testing**

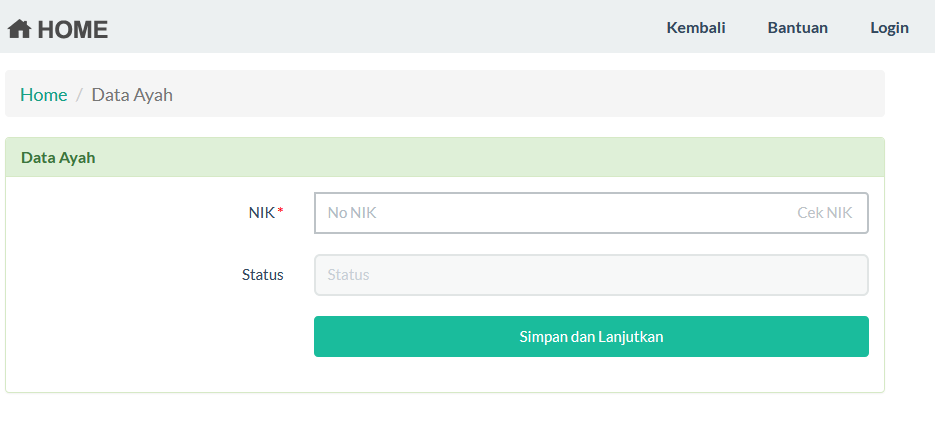


Gambar 4.22 Hasil Unit Testing

Setelah semua persiapan testing sudah dilakukan, langkah selanjutnya adalah melakukan testing tersebut. Testing dilakukan untuk semua fitur, mulai dari pengecekan parameter sampai request service pada server. Gambar 4.22 adalah hasil dari testing jika semua testing dinyatakan berhasil. Dari hasil yang diperoleh oleh unit testing ini menandakan bahwa SDK yang akan digunakan bisa diimplementasikan ke aplikasi client.

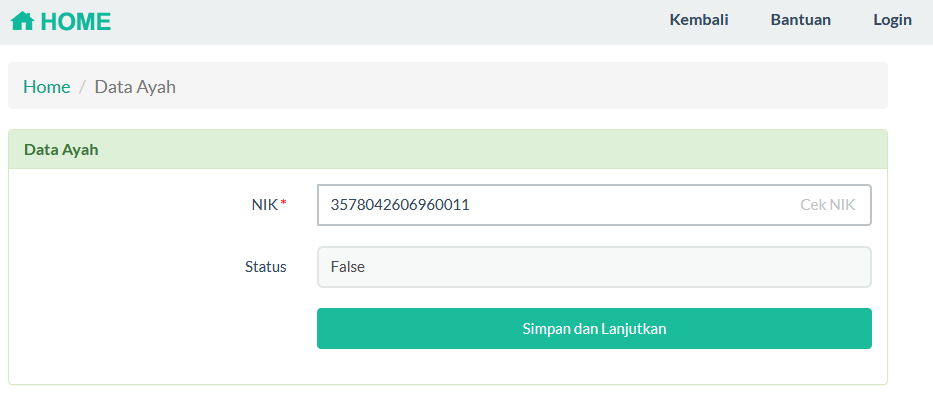
1. Implementasi SDK

Setelah unit testing dilakukan dan menghasilkan bahwa semua fitur tidak terdapat error dan berjalan lancer, selanjutnya adalah mengimplementasikan SDK tersebut ke dalam apikasi client sederhana. Aplikasi sederhana ini hanya untuk membuktikan bahwa SDK yang akan digunakan bisa diimplementasikan ke dalam aplikasi client. Aplikasi sederhana adalah aplikasi untuk mendapatkan data biodata wni dengan hanya mengirimkan parameter NIK.



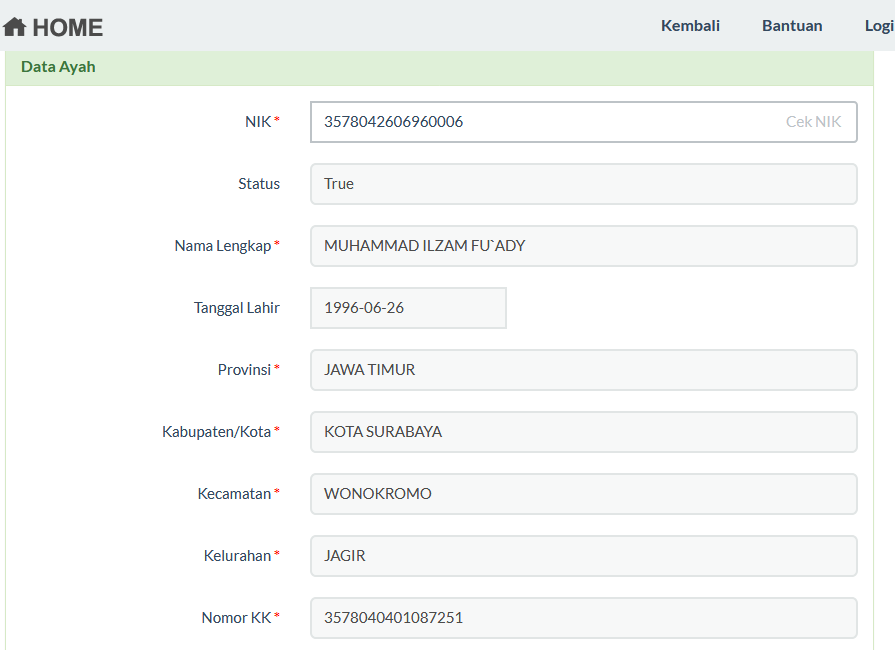
Gambar 4.23 Halaman Awal Aplikasi Client

Pada gambar 4.23 adalah halaman awal aplikasi sederhana ini. Pada aplikasi tersebut user diminta untuk memasukkan NIK. Setelah itu disebelah kanan terdapat tombol cek nik. Pada tombol tersebut SDK yang di implementasikan pada aplikasi ini digunakan. Pada aplikasi ini dilakukan request ke server dengan mengirimkan parameter NIK saja. Kemudian dari server merespon dengan biodata wni sesuai dengan NIK yang dikirim. Semua proses request tersebut dilakukan oleh SDK.



Gambar 4.24 Hasil NIK Salah

Pada gambar 4.24 adalah hasil jika setelah menekan tombol cek nik, data nik pada server tidak ada. Jika client mengirimkan nik yang salah atau nik tersebut tidak ada pada data server, maka server akan merespon bahwa data nik tersebut tidak ada atau data nik tersebut salah. Setelah itu aplikasi client akan mengatur response tersebut bahwa nik yang dikirim tersebut salah atau tidak ada. Pada aplikasi sederhana ini mengatur jika nik yang dikirimkan tidak ada maka status akan berubah menjadi false.



Gambar 4.25 Hasil NIK Benar

Pada gambar 4.25 adalah hasil setelah menekan tombol cek nik jika NIK yang dikirim tersebut ada pada data server. Data yang dikirim oleh server berupa JSON sehingga aplikasi client harus menyesuaikan bagaimana cara untuk memanfaatkan data JSON tersebut. Pada aplikasi sederhana ini jika NIK ada pada data server maka status akan berubah menjadi true, kemudian data JSON yang dikirimkan oleh server dimasukkan ke dalam input form sesuai dengan nilai masing masing.

1. IMPLEMENTASI DATA KE DALAM SISTEM

Dari data yang sudah di dapatkan sebelumnya, akan dimasukkan ke dalam sistem aplikasi. Aplikasi yang akan diimplementasikan adalah sebagai berikut :

1. Implementasi API Untuk Aplikasi Antrian BPJS

Uji Coba selanjutnya adalah penerapan REST API dalam aplikasi BPJS. Pada aplikasi ini data scope dari API digunakan untuk validasi data NIK untuk mendapatkan nomor antrian. Sehingga validasi dan pengecekan data untuk mendapat nomor antrian hanya membutuhkan nomor NIK saja.

Berikut implementasi API pada aplikasi antrian BPJS. Pada gambar 4.26 adalah tampilan utama untuk daftar dan mendapatkan nomor antrian. Untuk mendaftar hanya dibutuhkan nomor NIK dan dilakukan pengecekan dengan menekan tombol cek NIK. Pengecekan NIK dilakukan secara otomatis. Setelah NIK tervalidasi maka pemilik NIK bisa mendapatkan nomor antrian. Gambar 4.28 adalah gambar bahwa NIK sudah tervalidasi.

Setelah mendaftar akan masuk pada halaman menu. Setelah itu menekan tombol pendaftaran baru untuk mendapatkan nomor antrian. Gambar 4.29 adalah gambar halaman menu. Setelah semua proses dilakukan pemegang nomor NIK yang sudah divalidasi sebelumnya mendapatkan nomor antrian kemudian mencetak nomor antrian tersebut. Gambar 4.30 dan gambar 4.31 adalah nomor antrian yang didapat kemudian dilakukan cetak



Gambar 4.26 Aplikasi BPJS Cek NIK

Pada gambar 4.26 adalah halaman utama dari aplikasi BPJS kesehatan dimana pada halaman ini adalah halaman untuk mendaftarkan diri dan mendapatkan nomor urutan antrian. Untuk mendapatkan urutan antrian tersebut user harus memasukkan NIK kemudian tekan tombol cek nik. Secara otomatis informasi data pemilik dari NIK tersebut akan terisi pada kolom dibawahnya.



Gambar 4.27 Cek NIK Salah

Pada gambar 4.27 adalah halaman dimana user memasukkan NIK yang salah pada aplikasi BPJS kesehatan ini. Aplikasi client akan merequest kepada server NIK yang salah kemudian server akan meresponse dengan error bahwa NIK yang dikirimkan tersebut salah. Oleh aplikasi diatur sehingga menunjukkan bahwa NIK yang dimasukkan tersebut salah.



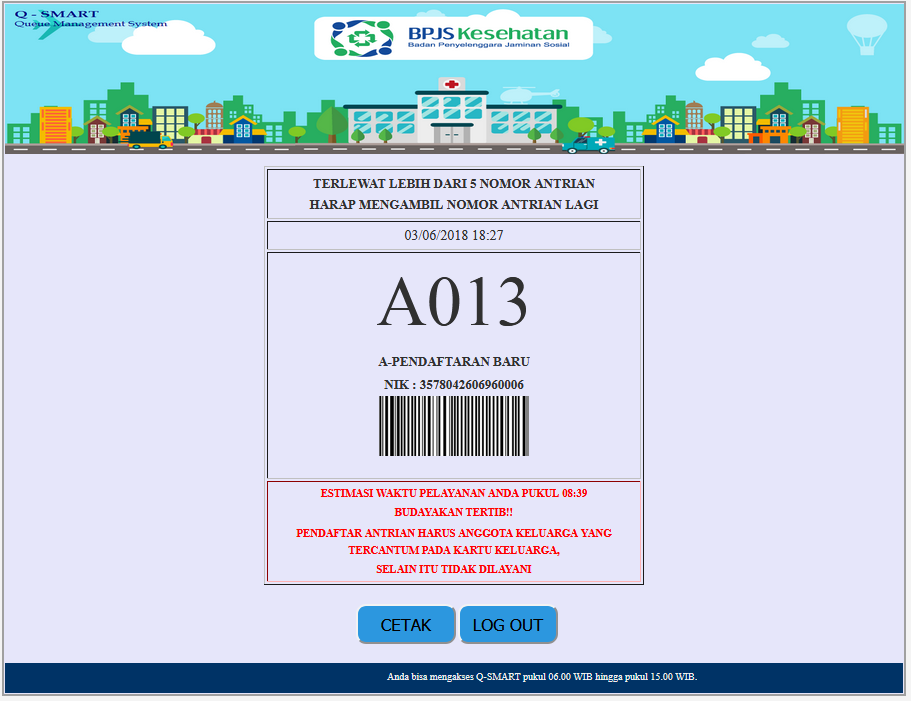
Gambar 4.28 Validasi Nomor NIK

Pada gambar 4.28 adalah halaman dimana user memasukkan NIK yang benar. User memasukkan NIK yang benar kemudian aplikasi akan merequest data dengan NIK yang benar. Server akan meresponse informasi data dari pemelik NIK tersebut. Aplikasi BPJS secara otomatis akan memasukkan data tersebut ke dalam field masing masing. User diminta untuk melakukan captcha selanjutnya user bisa mendaftar pada aplikasi ini.



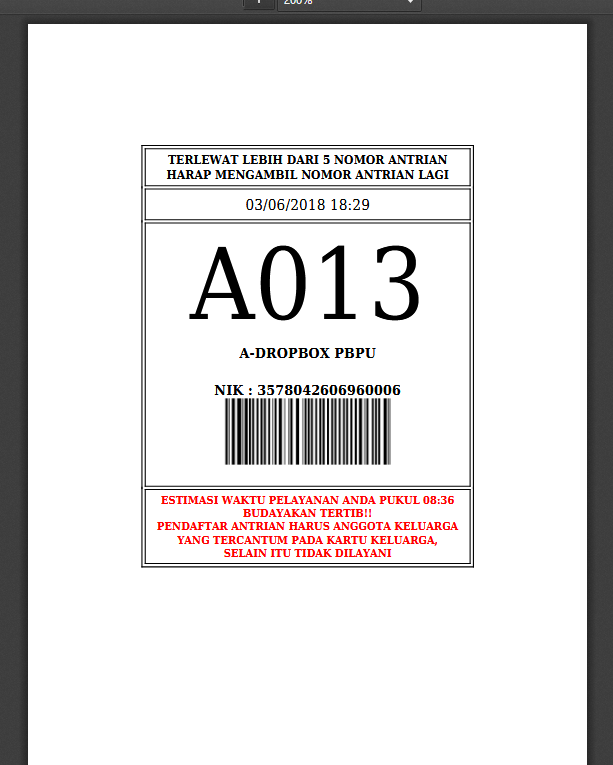
Gambar 4.29 Halaman menu

Pada gambar 4.29 adalah halaman setelah user mendaftar. Pada halaman ini disediakan beberapa menu. Menu untuk mengambil antrian adalah menu A yaitu pendaftaran baru.



Gambar 4.30 Generate Nomor Antrian

Pada gambar 4.30 adalah halaman nomor antrian yang didapat oleh user. Pada halaman ini user mendapatkan nomor antrian sebagai pendaftaran baru. Nomor antrian tersebut bisa dicetak atau user bisa lansung melakukan log out.



Gambar 4.31 Cetak Antrian

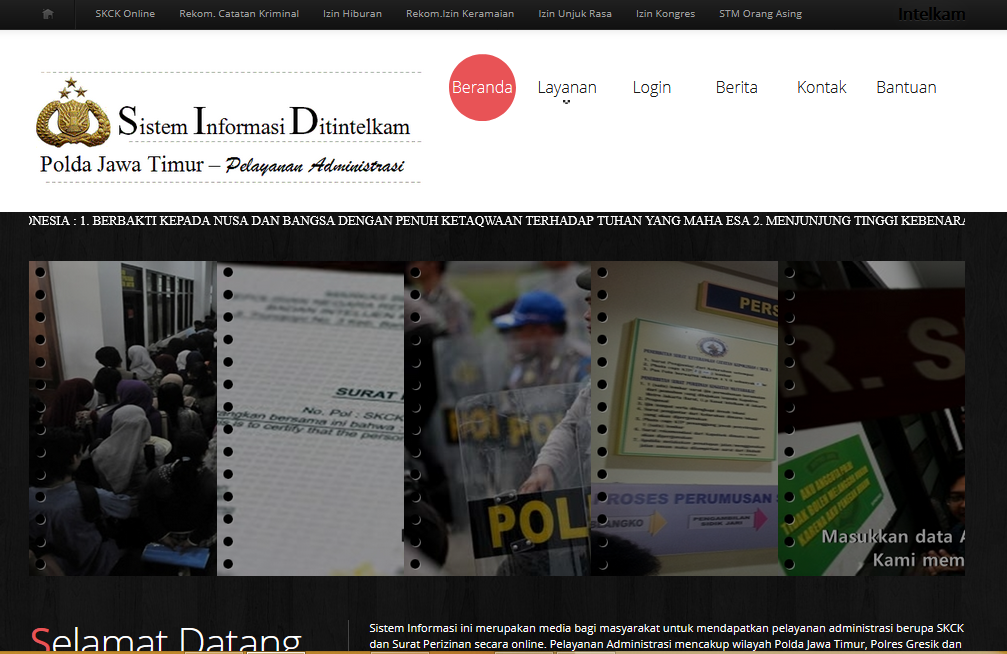
Pada gambar 4.31 adalah halaman untuk mencetak nomor urut. Pada halaman ini user mencetak nomor urutan yang didapatnya.

1. Implementasi Aplikasi Pembuatan SKCK Polda Jawa Timur

Uji Coba selanjutnya adalah penerapan lainnya dari REST API ini pada aplikasi pemerintahan. Aplikasi yang dijadikan uji coba adalah aplikasi untuk pada kepolisian untuk membuat surat SKCK. Aplikasi tersebut otomatis akan mengisikan data hanya dari NIK.

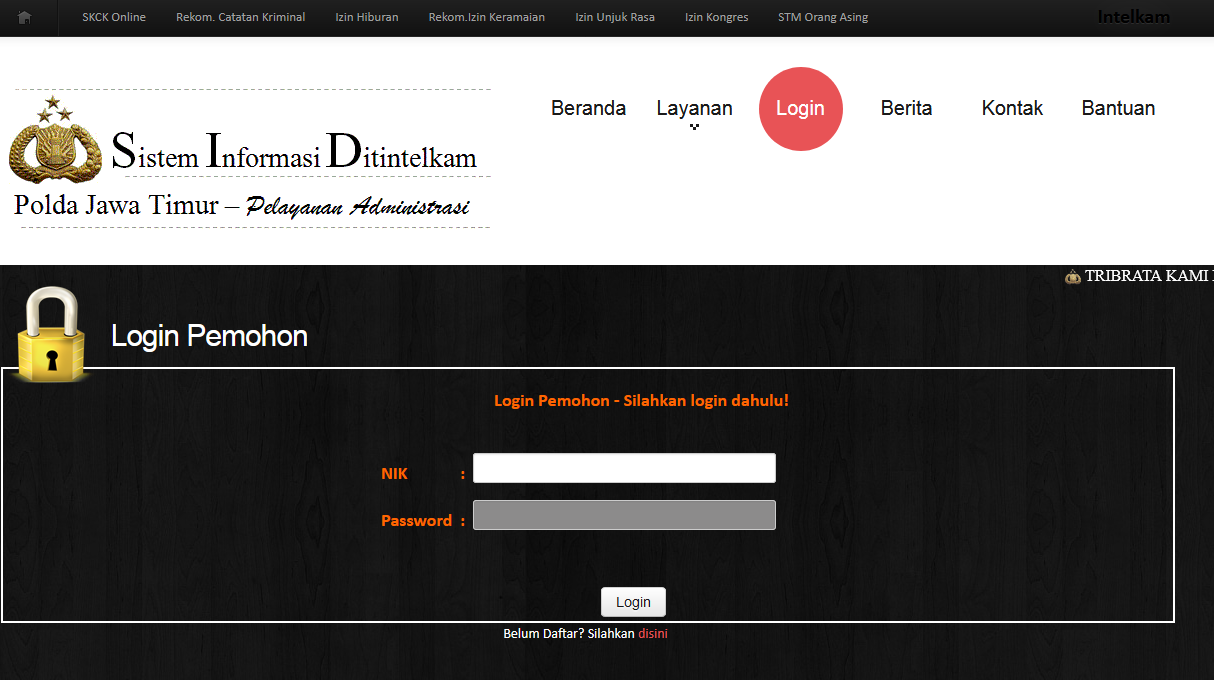
Berikut adalah aplikasi pembuatan SKCK Polda Jawa Timur. Langkah pertama yang dilakukan adalah melakukan login kedalam sistem aplikasi. jika belum memiliki account maka user harus mendaftar terlebih dahulu. User disini mendaftar dengan menggunakan nomor NIK.

Setelah melakukan login kemudian masuk ke layanan tingkat polda jatim. Halaman dashboard ditunjukkan pada gambar 4.32. kemudian pilih SKCK online dan pemohon baru. Layanan pembuatan SKCK baru ditunjukkan oleh gambar 4.36. Langkah selanjutnya adalah persiapan dokumen yang dibutuhkan ditunjukkan oleh gambar 4.37. Lalu kemudian masuk ke dalam pengisian formulir. Pada langkah inilah pemanfaatan API digunakan. Pada langkah ini user tidak perlu lagi memasukkan datanya satu satu. Pengisian formulir ini dilakukan secara otomatis dengan menanfaatkan data dari API. Pengisian formulir secara otomatis ditunjukkan pada gambar 4.38.



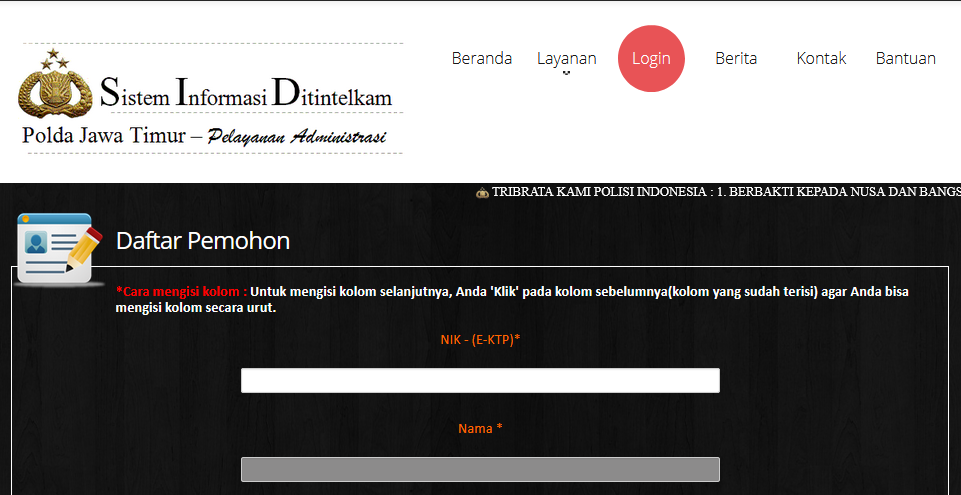
Gambar 4.32 Halaman Utama Sistem Informasi Ditentelkam

Pada gambar 4.32 adalah halaman beranda dari aplikasi Sistem Informasi Ditintelkam. Pada halaman ini terdapat beberapa menu didalamnya. Selain itu pada halaman ini terdapat beberapa informasi dari aplikasi tersebut.



Gambar 4.33 Login Sistem Informasi Ditintelkam

Pada gambar 4.33 adalah halaman login untuk mengurus skck bagi para pemohon. Pada halaman ini user diharuskan untuk memasukkan NIK dan password sesuai dengan data pada saat pendaftaran. Jika pemohon belum memiliki akun pada aplikasi ini, user bisa daftar dengan menekan tombol daftar dibawah tombol login.



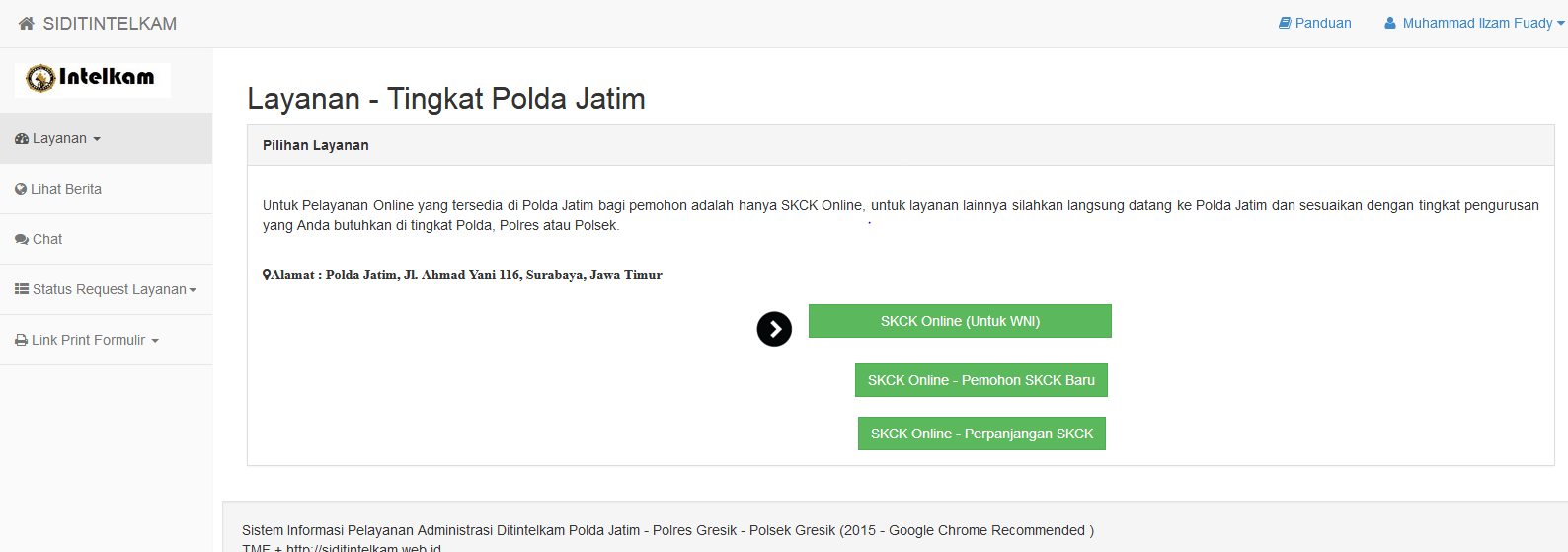
Gambar 4.34 Registrasi Sistem Informasi Ditintelkam

Pada gambar 4.34 adalah halaman pendaftaran bagi pemohon yang belum memiliki akun pada aplikasi ini. Pada halaman ini pemohon diharuskan untuk mengisi identitasnya sesuai dengan field yang sudah ditentukan. Setelah terisi semua tekan tombol daftar. Kemudian pemohon melakukan login dengan menggunakan NIK yang sudah didaftarkan sebelumnya.



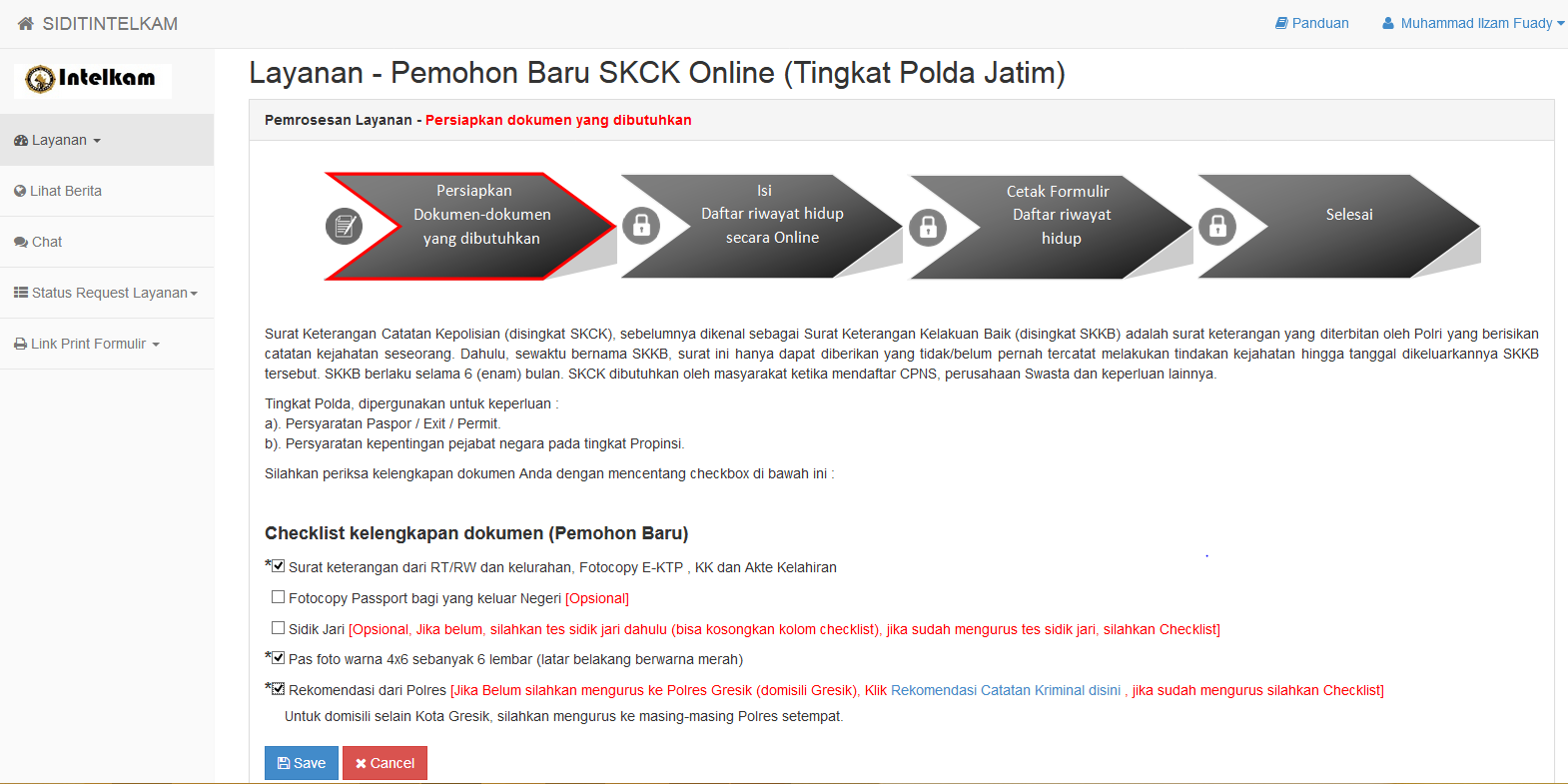
Gambar 4.35 Menu Layanan

Pada gambar 4.35 adalah halaman dashboard dari aplikasi Sistem Informasi Ditintelkam. Pada halaman ini terdapat beberapa menu untuk layanan SKCK dan perizinan. Selain itu pada halaman ini terdapat Visi, Motto, dan Semboyan Lambang didalamnya.



Gambar 4. 36 Layanan pendaftaran SKCK baru

Pada gambar 4.36 adalah halaman layanan SKCK Tingkat Polda Jatim. Pada halaman ini pemohon bisa membuat SKCK baru atau memperpanjang SKCK.



Gambar 4.37 Persiapan Dokumen

Pada gambar 4.37 adalah halaman persyaratan untuk mendaftarkan SKCK baru bagi pemohon. Pada halaman ini terdapat checklist kelengkapan dokumen untuk membuat SKCK baru. Jika semua dokumen yang dibutuhkan sudah lengkap maka pemohon bisa melanjutkan ke langkah selanjutnya.



Gambar 4. 38 Pengisian Otomatis pada Formulir

Pada gambar 4.38 adalah halaman untuk pengisian formulir identitas pemohon. Pada halaman ini informasi identitas pemohon akan di isikan secara otomatis oleh aplikasi sesuai dengan NIK dari pemohon saat mendaftar. Pada saat halaman ini di ditampilkan aplikasi akan melakukan request dengan mengirimkan NIK kepada server. Kemudian server akan merespon dengan mengirimkan data biodata wni kepada client dalam format JSON. Aplikasi akan mengolah data JSON tersebut dan memasukkan data pada setiap field yang bersangkutan.

*\*Halaman ini sengaja dikosongkan\**

# **BAB V**

PENUTUP

1. KESIMPULAN

Dengan uji coba sistem yang telah dilakukan maka dapat ditarik kesimpulan :

1. Aplikasi antar SKPD dapat terintegrasi satu sama lain dengan menggunakan web service REST yang artinya aplikasi satu dengan yang lainnya bisa berkomunikasi satu sama lain.
2. Pengiriman data tidak perlu lagi menggunakan cara manual.
3. Resiko terhadap data seperti redudansi data dapat diminimalisir.
4. Interoperabilitas pada aplikasi E-Government bisa diterapkan dengan metode REST API.
5. Dengan menggunakan Authentikasi OAuth2 data yang akan semakin aman untuk dikirimkan.
6. Dari hasil percobaan yang dilakukan menghasilkan interoperabilitas sistem antar SKPD dapat diimplementasikan dengan metode REST API dan untuk keamanan data dapat dilengkapi dengan metode OAuth2.
7. SARAN

Dengan penelitian yang sudah dilakukan terdapat saran untuk mengembangkan penelitian ini. Beberapa saran adalah sebagai berikut :

1. Aplikasi bisa dikembangkan kearah luar Jawa Timur.
2. Implementasi dapat dilakukan di lain platform seperti mobile atau desktop

*\*Halaman ini sengaja dikosongkan\**

# **Daftar Pustaka**

|  |  |
| --- | --- |
| [1] | "Intruksi Presiden No. 3 tahun 2003, tentang pengembangan E-Government Pemerintah Daerah di Indonesia". |
| [2] | J. E. Istiyanto and E. Susanta, "Model Interoperabilitas Antar Aplikasi e-Government," *ResearchGate,* 2012. |
| [3] | E. Susanta and J. E. Istiyanto, "Kebijakan Standarisasi Data Dan Problem Interoperabilitas Pada Aplikasi e-Government," 2012. |
| [4] | J. Simangunsong, "Pengembangan E-Government Di Indonesia," 2010. |
| [5] | A. P. Widodo, "MODEL INTEROPERABILITAS e-GOVERNMENT MENGGUNAKAN KOMBINASI SERVICE ORIENTED ARCHITECTURE (SOA) DAN EVENT DRIVEN ARCHITECTURE (EDA)," 2015. |
| [6] | W. Kumorotomo, I. W. Ordiyasa, L. E. Nugroho, P. I. Santoso and R. Ferdiana, "Review: Interoperability Model of e-Government Services," in *Information Technology, Computer, and Electrical Engineering (ICITACEE), 2015 2nd International Conference on*, Semarang, 2015. |
| [7] | A. P. Widodo, J. E. Istiyanto, R. Wardoyo and P. Santoso, "E-Government Interoperability Framework based on a Real Time Architecture," *IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 2,* 2013. |
| [8] | F. Maurer, C. Anslow and M. P. Robillard, "A Strudy of the Effectiveness of Usage Example in REST API Documentation," *IEEE Symposium on Visual Languages and Human-Centric Computing,* pp. 53-61, 2017. |
| [9] | Wikipedia, "Ruby on Rails," 18 Maret 2018. [Online]. Available: https://en.wikipedia.org/wiki/Ruby\_on\_Rails. [Accessed 19 Maret 2018]. |
| [10] | A. P. Widodo, "Kinerja Arsitektur Interoperabilitas E-Government Multi Platform," *Jurnal Matematika,* vol. 19, no. 1, pp. 16-28, 2016. |
| [11] | E. Susanta and K. Mustofa, "Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov di Pemkab Bantul Yogyakarta," *Jurtik - STMIK Bandung,* 2012. |
| [12] | G. Mulligan and D. Gracanin, "A Comparison of SOAP and REST Implementations of a Service Based Interaction Independence Middleware Framework," in *Winter Simulation Conference*, 2009. |
| [13] | A. Mulally, N. McKelvey and K. Curran, "Performance Comparison of Enterprise Applications on Mobile Operating Systems," *TELKOMNIKA,* vol. 9, pp. 503-514, 2011. |
| [14] | J. C. Bertot, U. Gorham, P. J. Jaeger, L. C. Sarin and H. Choi, "Big Data, Open Government and E-Government: Issues, Policies and Recommendations," *Information Polity,* vol. 19, pp. 5-16, 2014. |
| [15] | Z. A. Al-Sai and L. M. Abualigah, "Big Data and E-government: A review," in *International Conference on Information Technology*, 2017. |
| [16] | P. Cudre-Mauroux, "Keynote: Big Data Integration foe eGovernment," in *Fourth International Conference on eDemocracy & eGovernment* , Quito, 2017. |

# **BIOGRAFI PENULIS**

****

Nama : Muhammad Ilzam Fuady

TTL : Tuban, 26 Juni 1996

Alamat : Jln Bentul 2 no. 8, Surabaya, Jawa Timur

Email : ilzam.fuad12@gmail.com

No. HP : 085607569909

**Riwayat Pendidikan:**

* 2014 - 2018 : Politeknik Elektronika Negeri Surabaya

D4 Teknik Informatika

* 2011 - 2014 : SMA Negeri 15 Surabaya
* 2008 - 2011 : SMP Negeri 22 Surabaya
* 2002 - 2008 : SD Negeri Margorejo 1 Surabaya