

Integrasi *E-Government* Kabupaten Sidoarjo dengan *Service Oriented Architecture (SOA)*

Rengga Asmara¹, Jauari Akhmad Nur Hasim², Adi Putra Utama³

Departemen Teknik Informatika dan Komputer

Politeknik Elektronika Negeri Surabaya

Kampus PENS, Jalan Raya ITS, Keputih, Sukolilo, Surabaya, 60111, Indonesia

Email: rengga@pens.ac.id¹, jauari@pens.ac.id², adiputrautama17@gmail.com³

Abstrack - Duplication of data, data is out of sync, and the swelling of funds is the main problem of the current government service system (e-government). This is due to the lack of data integration between e-government systems in regional governments. The application of data integration can save up to 7 trillion in manufacturing and operational funds in each local government and can accelerate the process of interaction and communication between agencies or to people in the regional government. This study aims to develop an integrated e-government from various services that exist in government services in Sidoarjo Regency. The method used in developing this integration is using the Service Oriented Architecture (SOA) method. This method was chosen because it has many advantages, such as: reusable, easy to manage, high scalability, and easy to develop. The development of integrated e-government has three main stages, namely: database creation, service creation, and service implementing on the frontend. The results of this study are in the form of conclusions on the ability of the methods used in e-government integration and an application of e-government services.

Keywords - E-government, Integration, Sidoarjo Regency, Service Oriented Architecture (SOA).

Intisari - Duplikasi data, data tidak sinkron, dan pembengkakan dana merupakan masalah utama dari sistem pelayanan pemerintahan (*e-government*) saat ini. Hal ini disebabkan karena belum adanya integrasi antar sistem e-government pada pemerintah daerah (pemda). Penerapan integrasi data dapat menghemat dana pembuatan dan operasional hingga 7 triliun pada setiap pemerintah daerah dan dapat mempercepat proses interaksi dan komunikasi antar dinas pada pemerintah daerah. Pada penelitian ini bertujuan untuk mengembangkan sebuah e-government yang terintegrasi dari berbagai layanan yang ada pada pelayanan pemerintahan Kabupaten Sidoarjo. Metode yang digunakan dalam mengembangkan integrasi ini yaitu menggunakan Metode *Service Oriented Architecture (SOA)*. Metode ini dipilih karena mempunyai banyak kelebihan, seperti : *reusable*, mudah dikelola, skalabilitas yang tinggi, dan mudah dikembangkan. Pada proses pengembangan e-government mempunyai 3 tahap utama, yaitu : pembuatan database, pembuatan *service*, dan penerapan *service* pada *frontend*. Hasil dari penelitian ini yaitu berupa kesimpulan terhadap kemampuan metode yang digunakan dalam integrasi e-government dan sebuah aplikasi pelayanan e-government.

Kata Kunci - E-government, Integrasi, Kabupaten Sidoarjo, Service Oriented Architecture (SOA).

I. PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi (TIK) mengubah kehidupan manusia. Perkembangan ini mengubah hubungan karakteristik antar hubungan manusia, bisnis, dan mengubah hubungan antar pemerintah dan masyarakat. Kebutuhan informasi yang cepat membuat pemerintah harus mengikuti perkembangan teknologi informasi. Seperti memberikan pelayanan yang terbaik untuk masyarakat, internal pemerintahan, partner pemerintah, dan entity pemerintahan. Pelayanan ini biasa disebut dengan *E-Government*.

Menurut Kementerian Komunikasi dan Informasi, *E-Government* adalah aplikasi teknologi informasi yang berbasis internet dan perangkat lainnya yang dikelola oleh pemerintah untuk keperluan penyampaian informasi dari pemerintah kepada masyarakat, mitra bisnisnya, dan lembaga-lembaga lain secara online[1]. Kebutuhan *E-Government* di Indonesia menjadi sangat penting sejak dikeluarkannya Instruksi Presiden Republik Indonesia Nomor 3 Tahun 2003 Tentang Kebijakan dan Strategi Nasional Pengembangan *E-Government*[2], yang mengharuskan setiap wilayah daerah, lembaga-lembaga, instansi pemerintahan, dll. Untuk memanfaatkan kemajuan teknologi komunikasi dan informasi yang pesat. Demi tercapainya efisiensi, efektifitas, transparansi dan akuntabilitas penyelenggaraan pemerintah serta terbukanya peluang bagi pengaksesan, pengelolaan, dan pendayagunaan informasi dalam volume yang besar secara cepat dan akurat.

Saat ini perkembangan *E-Government* di Indonesia belum maksimal. Berdasarkan survei tahun 2016, Perserikatan Bangsa-Bangsa (PBB) mempublikasikan bahwa Indonesia mendapatkan peringkat 116 *E-Government Development Index* (EGDI)[3]. Hal ini dibuktikan masih 85% jumlah website *E-Government* di Indonesia yang dapat diakses. Sedangkan website lainnya masih belum bias diakses yang disebabkan oleh beberapa faktor, seperti sedang dalam perbaikan, server *error*, gagal koneksi ke *database*, dll. Serta masih banyak *E-Government* di Indonesia yang belum menerapkan Sistem Integrasi antar layanan yang mengakibatkan kurangnya efektifitas dan efisiensi layanan *E-Government*.

Banyak daerah di Indonesia yang belum menerapkan Integrasi *E-Government*, salah satunya *E-Government* di Kabupaten Sidoarjo[6]. Di Kab. Sidoarjo hampir di setiap dinas menyediakan layanan *E-Government*-nya masing-masing. Namun, karena belum adanya system integrasi antar layanan *E-Government*, proses pelayanan baik transaksi maupun komunikasi lintas dinas masih menggunakan cara manual sehingga kurang efektif dan efisien dalam segi kecepatan maupun ketepatan[5].

Kabupaten Sidoarjo merupakan salah satu penyangga Ibukota Provinsi Jawa Timur merupakan daerah yang mengalami perkembangan pesat[4]. Oleh karena itu *E-Government* di Kabupaten Sidoarjo diperlukan adanya system integrasi untuk mengatasi permasalahan tersebut. Dengan adanya system ini diharapkan pelayanan pada bidang pemerintahan dapat berjalan lebih efektif dan efisien. Sehingga mampu meningkatkan kepuasan masyarakat terhadap kinerja pemerintah.

Berdasarkan latarbelakang di atas, dapat dirumuskan permasalahan yang ada dalam paper ini adalah sebagai berikut:

- a. Adanya duplikasi data pada server *E-Government* Kab. Sidoarjo yang menyebabkan pembengkakan data dan habisnya ruang storage. Duplikasi data terjadi karena adanya beberapa *table* dengan *field* yang sama antara aplikasi yang satu dengan aplikasi yang lain.
- b. Adanya data yang tidak sinkron antara dinas yang satu dengan dinas yang lain karena kesalahan input user. Karena masih banyaknya data yang belum terintegrasi, data inputan dari user bias mengakibatkan adanya perbedaan ketika menginputkan pada aplikasi yang berbeda yang seharusnya data yang dimasukkan harus sama.
- c. Pelayanan pada system pemerintahan yang masih membutuhkan waktu yang lama. Ini dikarenakan beberapa pelayanan pemerintahan harus berinteraksi dengan beberapa dinas lain, dan masyarakat yang berkepentingan harus mendatangi kedinas tersebut.

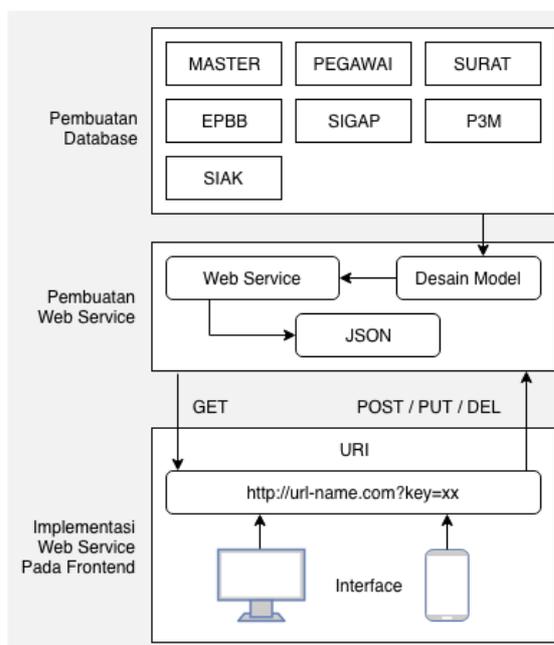
As'ad[11] menjelaskan bahwa integrasi antar *service* adalah salah satu masalah yang paling penting yang mempengaruhi implementasi *E-Government* di seluruh dunia. Menyediakan layanan terpadu untuk warga, bisnis, dan *stakeholder* yang terlibat dalam *E-Government* di "*one stop portal*" dianggap sebagai peluang besar bagi pemerintah untuk meningkatkan efisiensi dan efektivitas layanan mereka. *E-Government* yang sukses didasarkan pada dua faktor. Pertama adalah ketersediaan situs web *E-Government* dan layanan yang tersedia tidak hanya pada waktu jam kerja pemerintah. Kedua adalah aksesibilitas, yang artinya situs *E-Government* dan

layanannya harus dapat diakses oleh pengguna dimanapun. Dennis. dkk.[8] menjelaskan pada pemerintahan terjadi penyebaran data dan informasi pada setiap unit instansi pengelola maupun instansi penyelenggara kegiatan. Maka dengan kasus seperti itu terjadi redundansi data dan kesulitan untuk akses data satu dengan yang lain. Sumber data yang tidak terkoordinasi tersebut adalah salah satu permasalahan yang dapat disolusikan dengan menggunakan pengembangan sistem yang terintegrasi dengan *E-Government*. Integrasi data hanya dapat terlaksana pada ruang lingkup fisik tertentu, misalnya: tersedianya sistem manajemen *database*, aplikasi dan *database* terhubung pada suatu jaringan komputer[9].

II. SIGNIFIKANSI STUDI

Tujuan dari pada paper ini yaitu peneliti akan mengembangkan sistem integrasi pada *E-Government* yang diimplementasikan pada *E-Government* Kabupaten Sidoarjo. Dimana sistem integrasi ini akan menghubungkan berbagai *E-Government* pada dinas pemerintahan yang berbeda. Sehingga nantinya dapat mempercepat dan mempermudah proses transaksi maupun komunikasi data sebagai bentuk pelayanan kepada masyarakat. Selain itu tujuan dari paper ini yaitu memanfaatkan kemajuan teknologi dalam bidang TIK dengan pengelolaan secara maksimal. Dengan tujuan menghasilkan suatu layanan yang bermanfaat bagi pemerintah dan masyarakat Kabupaten Sidoarjo.

Berikut ini pada Gambar 1 di bawah ini merupakan desain sistem paper yang sedang berlangsung. Mulai dari proses pembuatan database, pembuatan *web service* dan yang terakhir penggunaan *web service* tersebut ke dalam *interface*.



Gambar 1. Desain Sistem

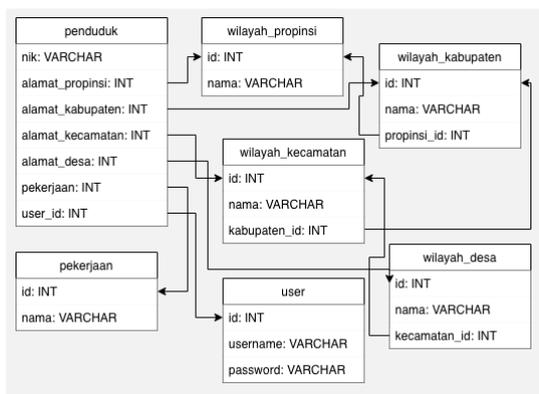
Pembuatan *Web Service* menggunakan metode *Service Oriented Architecture* (SOA)[7], yaitu dengan membuat paket-paket layanan dalam bentuk unit-unit kecil yang dapat dikembangkan secara terus-menerus. Metode SOA ini nantinya juga dapat mengintegrasikan antara berbagai layanan paket lainnya yang telah dibuat.

Langkah pertama yang dilakukan untuk membangun integrasi *web service* ini yaitu membuat database. peneliti menggunakan database MySQL, dikarenakan *database* tersebut merupakan *database* yang *free-to-use*, serta memfasilitasi manajemen *database* yang efektif dengan untuk menghubungkannya ke *database*. Dengan memigrasikan aplikasi basis data saat

ini ke MySQL, banyak perusahaan menikmati penghematan biaya yang signifikan pada aplikasi-aplikasi baru. Ketergantungan dan kemudahan manajemen dapat menghemat waktu pemecahan masalah yang terbuang sia-sia dalam memperbaiki masalah *downtime* dan masalah kinerja. Ada beberapa *database* yang akan dibuat pada proye akhir ini, berikut ini merupakan *database* atau tabel yang akan dibuat beserta kegunaannya:

a) *Database Master*

Database master merupakan *database* yang bersifat *primary*, artinya *database* ini digunakan sebagai pendukung dari beberapa *database*/tabel lainnya. Contohnya tabel penduduk, wilayah, user, dll. Berikut pada Gambar 2 merupakan tabel-tabel dari *database* master



Gambar 2. Database Master

a) *Database* SIAK

Database siak berisi beberapa tabel yang digunakan untuk menyimpan data-data kependudukan, seperti: surat kelahiran, surat kematian, surat permohonan kartu tanda penduduk (KTP), dan surat permohonan kartu keluarga (KK). Berikut merupakan spesifikasi dari *database* siak.

b) *Database* SURAT

Database surat berisi beberapa tabel yang digunakan untuk menyimpan data-data surat menyurat yang bersifat non-kependudukan, seperti: surat domisili perorangan, surat domisili usaha, surat izin keramaian, surat keterangan desa, surat keterangan miskin, surat pengantar, dan surat permohonan SKCK. Berikut merupakan spesifikasi dari *database* surat.

c) Tabel Pegawai

Tabel pegawai digunakan untuk menyimpan data-data penduduk yang bekerja pada dinas pemerintahan. Berisi beberapa *field* yang digunakan untuk menyimpan NIK, NIP, jabatan pegawai, golongan pegawai, status aktif atau tidak, dll. Berikut pada Gambar 3 merupakan tabel pegawai beserta *field* dan tipe datanya.

Pegawai	(lanjutan)
id: INT	asal_dinas: VARCHAR
nik: VARCHAR	pas_foto: VARCHAR
nip: CHAR	pendidikan_terakhir: VARCHAR
jabatan: VARCHAR	status_aktif: INT
golongan: VARCHAR	created_at: TEXT
tmt_cpns: DATE	updated_at: TEXT
tmt_pns: DATE	

Gambar 3. Tabel Pegawai

d) Tabel EPBB

Tabel EPBB digunakan untuk menyimpan data-data perpajakan yang akan dibayar oleh setiap masyarakat. Tabel ini berisi *field* lokasi pembayar pajak, jumlah pajak, masa tenggang, dll. Berikut pada Gambar 4 merupakan tabel epbb.

PBB	(lanjutan)
id: INT	status: INT
propinsi_id: INT	tahun: CHAR(4)
kabupaten_id: INT	updated_at: DATETIME
kecamatan_id: INT	updated_by: INT
desa_id: INT	created_at: DATETIME
jumlah: VARCHAR	created_by: INT
no_urut: INT	
tgl_jatuh_tempo: DATE	

Gambar 4. Tabel EPBB

e) Tabel SIGAP

Tabel sigap digunakan untuk menyimpan data-data pelaporan bencana dari masyarakat. Berisi *field* nik yang berhubungan langsung ke tabel penduduk, lokasi bencana, waktu bencana, lampiran, dll. Berikut pada Gambar 5 merupakan tabel sigap.

SIGAP
id: INT
nik: VARCHAR
tempat: VARCHAR
waktu: DATETIME
catatan: TEXT
jml_warga_terancam: INT
jenis: VARCHAR
lampiran: VARCHAR

Gambar 5. Tabel SIGAP

f) Tabel P3M

Tabel p3m digunakan untuk menyimpan data-data pegaduan masyarakat kepada pihak dinas terkait. Tabel ini berisi *field* judul laporan, isi, gambar, tanggal approve, tanggapan, dll. Berikut pada Gambar 6 merupakan tabel p3m.

P3M	(lanjutan)
id: INT	status: INT
judul: VARCHAR	notif: INT
isi: TEXT	kirim_ke: INT
gambar: VARCHAR	lapor: INT
tanggal: DATE	tanggapan: TEXT
tanggal_approve: DATE	
nik: VARCHAR	

Gambar 6. Tabel P3M

Langkah kedua yaitu pembuatan *web service*. Pada tahap ini mempunyai dua tahapan, yaitu membuat desain model dan membuat *web service* itu sendiri. Selanjutnya dari *web service* menghasilkan keluaran berupa data JSON (*JavaScript Object Notation*). Berikut merupakan penjelasan dari tahapan tersebut:

a) Desain Model

Desain model bertujuan untuk merepresentasikan suatu *database* dalam bentuk kode. Model yang dibangun berfungsi menghubungkan *database* dengan *web service* untuk interaksi dalam pengolahan data.

Atribut yang terdapat pada model hampir sama dengan atribut pada tabel, namun lebih spesifik. Pada Gambar 7 merupakan contoh model dalam bentuk kode yang merepresentasikan tabel. Pada paper ini peneliti menggunakan *library sequelize* untuk mempermudah dalam pembuatan model.

```

'use strict';
module.exports = (sequelize, DataTypes) => {
  const epbb = sequelize.define('epbb', {
    nik: {
      allowNull: false,
      type: DataTypes.STRING,
      validate: {
        notEmpty: true,
      }
    },
    total: {
      allowNull: false,
      type: DataTypes.STRING,
      validate: {
        notEmpty: true,
      }
    },
    jatuh_tempo: {
      allowNull: true,
      type: DataTypes.DATE,
      validate: {
        notEmpty: false,
      }
    }
  },

```

Gambar 7. Desain model dalam bentuk code

Dari contoh kode di atas diinisialisasikan bahwa nama tabel yaitu epbb, dari tabel epbb berisi field nik, total dan jatuh tempo. Setiap field mempunyai atribut masing-masing. Seperti contohnya nik mempunyai atribut *allowNull* yang bernilai *false*, artinya bahwa *field* nik tidak boleh kosong (harus diisi). *Field* nik ini mempunyai tipe data *string* atau bila di *database* biasa disebut dengan *varchar*. Kemudian untuk validasinya berisi *notEmpty* bernilai *true*, artinya jika data yang dimasukkan pada *field* nik bernilai kosong maka secara otomatis model akan memberikan *feedback error* pada *service* yang berkaitan dengan tabel tersebut.

Untuk fungsi-fungsi yang mendukung pembuatan model biasanya bergantung pada *library* yang digunakan. Sehingga perlu disesuaikan dengan kebutuhan untuk proses pembuatan layanan aplikasi.

b) Web Service

Setelah membuat model langkah selanjutnya yaitu masuk pada inti pembuatan web service. web service biasa disebut dengan REST (*REpresentational State Transfer*), yaitu standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web[10]. *Web service* yang dikembangkan mempunyai tiga komponen utama. Pertama *provider* yang menyediakan informasi dan data. Kedua agent/broker yang membangun *web service*. Ketiga adalah *requester* yang dapat menggunakan layanan dari *provider*[14]. *Web service* umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai protokol untuk komunikasi data[13].

Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST *client* mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID[12]. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

Pada penelitian ini ada beberapa layanan yang dibangun dalam bentuk web service. Berikut merupakan *service* yang akan dibangun pada paper ini:

1) SURAT

SURAT merupakan aplikasi yang digunakan untuk pembuatan surat menyurat, seperti pembuatan Surat Domisili Perorangan, Surat Domisili Usaha, Surat Izin Keramaian, Surat Keterangan Desa, Surat Keterangan Miskin, Surat Pengantar, dll

2) SIAK

Sistem Informasi Administrasi Kependudukan, untuk menata sistem administrasi kependudukan di Indonesia, sistem ini meliputi pendataan penduduk dan pencacatan sipil. Data kependudukan antara lain : Nomor Induk Kependudukan (NIK), Kartu Keluarga (KK), Kartu Tanda Penduduk (KTP), Akta Kelahiran, Akta Kematian, dan sebagainya.

3) E-PBB (Pajak Bumi dan Bangunan)

Sistem Manajemen Informasi Objek Pajak, digunakan untuk melihat daftar pajak dan tanggungan pajak. Berikut merupakan hasil dari pengecekan pajak dari seseorang dengan memasukkan sebuah NIK (Nomor Induk Kependudukan) yang nantinya dapat menampilkan jumlah pajak tahun ini dan history pajak pada tahun-tahun sebelumnya.

4) P3M (Pusat Pelayanan Pengaduan Masyarakat)

P3M adalah aplikasi untuk menampung aspirasi masyarakat Kota Delta (Sidoarjo) yang secara real time akan disampaikan melalui email masing-masing anggota Dewan Perwakilan Rakyat Daerah Kabupaten Sidoarjo.

5) SIGAP (Sidoarjo Tanggap)

Aplikasi sistem informasi tentang bencana alam dan laporan bencana alam. Pada pelayan ini masyarakat dapat mengirimkan informasi terkait dengan bencana yang sedang terjadi melalui *website* ataupun mobile. Pelayanan ini dilengkapi dengan platform google map untuk memudahkan masyarakat dalam menentukan lokasi terjadinya bencana.

6) SIMPEG

Simpeg adalah *Data Center* (pusat data) kepegawaian pada Pemerintah Kabupaten Sidoarjo yang menyediakan informasi-informasi tentang kepegawaian. Saat ini ada ribuan data pegawai yang tercatat dalam *database*.

Dalam pembuatan *web service* harus diinisialisasikan bagaimana metode untuk memprosesnya. Ada beberapa metode untuk memproses *web service* yaitu, GET, POST, PUT / PATCH. GET berfungsi untuk mengambil data, POST berfungsi untuk menambah data dan PUT untuk mengubah data.

c) *Output JSON (JavaScript Object Notation)*

Keluaran dari *web service* berupa format JSON sebagai bentuk representasi dari REST itu sendiri. JSON mempunyai komponen dasar, yaitu :

- 1) *Objects*: *Objects* diawali dan diakhiri dengan kurung kurawal ({}).
- 2) *Object Member*: Terdiri dari string dan dipisahkan dengan koma (,).
- 3) *Arrays*: *Arrays* diawali dan diakhiri dengan kurung siku ([]) dan mengandung nilai yang berbeda.
- 4) *Values*: Nilai dapat berupa string, objek, array, atau literal.
- 5) *Strings*: String dikelilingi oleh tanda kutip ganda (") dan berisi titik dua Unicode (:). Anggota dipisahkan oleh koma (,). Nilai dipisahkan oleh koma (,).

Langkah terakhir yaitu implementasi *web service* ke user interface seperti web atau mobile app. Untuk menerapkan *web service* harus melalui mekanisme penggunaan sintak URI (*Uniform Resource Identifier*). Berikut merupakan penjelasan dari URI dan user interface yang akan dibangun.

a) URI

URI yaitu sebuah karakter yang mengidentifikasi sumber daya web berdasarkan nama, lokasi ataupun keduanya. Dalam hal ini sumber daya yang dimaksud adalah *web service*. Dalam URI mempunyai beberapa komponen yaitu: skema, *authority* (informasi *user*, *host*, *port*, *path*), *query*, *fragment*.

Berikut pada Gambar merupakan contoh sintaks URI dalam penggunaan *web service* yang berfungsi untuk mengambil data penduduk berdasarkan nik.

Setiap URI dimulai dengan skema untuk menetapkan identifikasi pada karakter tersebut. Ada banyak jenis skema yang sangat familiar, seperti HTTP, HTTPS, FTP, SFTP, dll. Kemudian untuk *authority* menggunakan localhost. *Authority* ini berfungsi sebagai otoritas penamaan URI. Pada bagian port mempunyai nilai 3000 yang merupakan sub komponen dari *authority*. Biasanya secara default pada skema HTTP mempunyai port 80. Untuk *pathnya* bernama penduduk sesuai dengan data yang akan diambil melalui sintak tersebut. Sedangkan untuk parameter bernilai 12345 yang merupakan nik dari data penduduk dan ditambah key sebagai verifikasi atau ijin untuk mengakses *web service*.

Untuk memudahkan implementasi *web service* ataupun penggunaan *web service* peneliti menyediakan dokumentasi yang sudah lengkap dan siap digunakan seperti contoh pada Gambar di bawah ini yang merupakan dokumentasi penggunaan fungsi login, yaitu dengan menggunakan method POST disertai dengan body username dan password.

b) User Interface

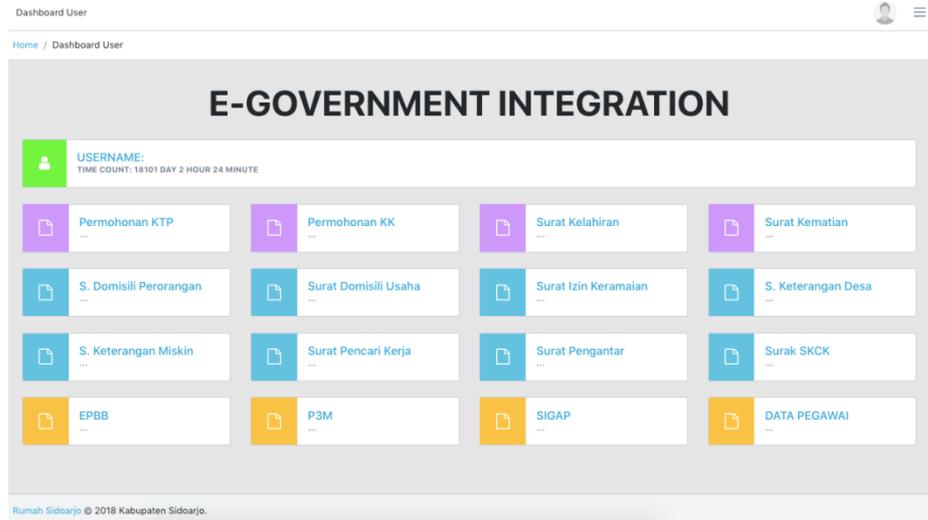
Pada paper ini aplikasi *E-Government* akan dikembangkan dalam 2 platform antar muka (*user interface*), yaitu pada web dan *mobile (android)*. Ini bertujuan untuk tercapainya kemudahan penggunaan pada aplikasi *E-Government* Kabupaten Sidoarjo.

III. HASIL DAN PEMBAHASAN

Dari penelitian yang telah dilakukan peneliti telah membuat *E-Government* yang telah terintegrasi. Berikut merupakan hasil dari aplikasi *E-Government* yang telah terintegrasi :

A. Dashboard

Berisi menu-menu yang nantinya dapat digunakan oleh masyarakat dalam penggunaan pelayanan. Pada Gambar 8 di bawah ini merupakan hasil tampilan dari dashboard user



Gambar 8. Dashboard

B. Surat Domisili Perorangan

Pada Gambar 9 di bawah ini merupakan form untuk mengajukan surat permohonan domisili perorangan untuk keperluan tempat tinggal.

Gambar 9. Form surat domisili perorangan

C. Surat Domisili Usaha

Pada Gambar 10 di bawah ini merupakan form untuk mengajukan surat izin usaha

Gambar 10. Form surat domisili usaha

D. Surat Izin Keramaian

Pada Gambar 11 di bawah ini merupakan *form* untuk mengajukan surat izin keramaian

Gambar 11. Form surat izin keramaian

E. Surat Keterangan Desa

Pada Gambar 12 di bawah ini merupakan *form* untuk mengajukan surat keterangan desa

Gambar 12. Form surat keterangan desa

F. Surat Keterangan 13 Miskin

Pada Gambar di bawah ini merupakan *form* untuk mengajukan surat keterangan miskin

Gambar 13. Form surat keterangan miskin

G. Surat Pengantar

Pada Gambar 14 di bawah ini merupakan *form* untuk mengajukan surat pengantar

Gambar 14. Form surat pengantar

H. Surat Permohonan KTP

Pada Gambar 15 di bawah ini merupakan *form* untuk mengajukan surat permohonan KTP

Gambar 15. Form surat permohonan KTP

I. Surat Permohonan KK

Pada Gambar 16 di bawah ini merupakan *form* untuk mengajukan surat permohonan KK

Gambar 16. Form surat permohonan KK

J. Surat Kelahiran

Pada Gambar 17 di bawah ini merupakan *form* untuk mengajukan surat kelahiran

Gambar 17. Form surat kelahiran

K. Surat Kematian

Pada Gambar 18 di bawah ini merupakan *form* untuk mengajukan surat kematian

Gambar 18. Form surat kematian

L. EPBB

Pada Gambar 19 di bawah ini merupakan tampilan dari aplikasi EPBB yang mempunyai fitur pengecekan pajak berdasarkan *input* nik.

Masukkan NIK anda

12345

Q Search

Biodata

Nama	Jenis Kelamin	Alamat	No. Telp
adiputra17	1	Pucanglaban, T. Agung	085785067866

Pajak anda 2018 : Rp. 100000

History

Tahun	Jatuh Tempo	Pokok PBB	Denda	Jumlah Bayar	Tgl. Pembayaran
2018	01/01/2018	100.000	0	90000	01/02/2018
2017	01/01/2018	50.000	0	50000	01/02/2018
2016	01/01/2018	120.000	0	120000	01/02/2018

Rumah Sidoarjo © 2018 Kabupaten Sidoarjo.

Gambar 19. Form pengecekan pajak

M. SIGAP

Pada Gambar 21 di bawah ini merupakan *form* untuk melaporkan terjadinya bencana.

Lapor Bencana History Laporan Jenis Bencana

Lapor Bencana

Pilih Lokasi

Enter a location Tambah

Map Satellite

Balungbendo Penun Citra Surya Mas Sidoarjo

Keterangan

Tempat Bencana Catatan

Waktu Bencana dd/mm/yyyy

Jml. Warga Terancam

Jenis Bencana Lampiran/Foto Choose File No file chosen

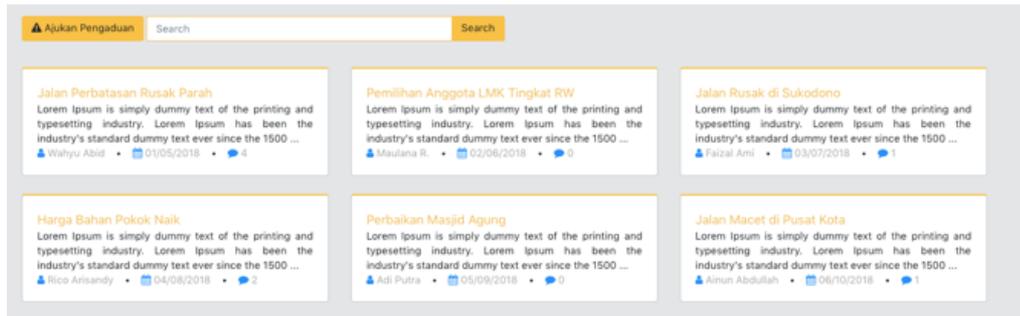
Submit

Rumah Sidoarjo © 2018 Kabupaten Sidoarjo.

Gambar 20. Form SIGAP

N. P3M

Pada Gambar 20 di bawah ini merupakan tampilan dari aplikasi pengaduan masyarakat, di fitur ini masyarakat dapat mengajukan pengaduan kepada pemerintah. Dan masyarakat dapat juga mencari pengaduan yang lainnya.



Gambar 21. Tampilan aplikasi P3M

Selain itu peneliti juga melakukan uji kecepatan di setiap pelayanan, yang membandingkan antara arsitektur SOA dengan tanpa arsitektur SOA. Berikut pada Gambar 21 di bawah ini merupakan hasil dari uji coba kecepatan tersebut.

Service	Hasil Percobaan SOA (s)					Hasil Percobaan Tanpa SOA (s)				
	1	2	3	4	5	1	2	3	4	5
Permohonan Surat KTP	0.02	0.02	0.025	0.02	0.02	0.032	0.034	0.35	0.033	0.033
Permohonan Surat KK	0.025	0.02	0.025	0.02	0.02	0.03	0.03	0.039	0.03	0.03
Surat Kelahiran	0.08	0.1	0.09	0.08	0.1	0.19	0.22	0.19	0.18	0.2
Surat Kematian	0.033	0.035	0.04	0.036	0.036	0.046	0.047	0.048	0.043	0.044
Surat Domisili Perorangan	0.04	0.045	0.04	0.04	0.04	0.05	0.055	0.052	0.054	0.053
Surat Domisili Usaha	0.05	0.05	0.055	0.05	0.05	0.06	0.06	0.065	0.06	0.06
Surat Izin Keramaian	0.05	0.05	0.05	0.05	0.05	0.07	0.08	0.08	0.075	0.08
Surat Keterangan Desa	0.018	0.02	0.019	0.02	0.019	0.018	0.02	0.019	0.02	0.019
Surat Keterangan Miskin	0.02	0.022	0.024	0.02	0.02	0.035	0.032	0.034	0.038	0.037
Surat Pengantar	0.015	0.017	0.017	0.015	0.016	0.025	0.027	0.027	0.025	0.026
EPBB	0.013	0.014	0.013	0.013	0.014	0.023	0.024	0.023	0.023	0.024
P3M	0.012	0.012	0.014	0.013	0.014	0.026	0.023	0.024	0.023	0.024
SIGAP	0.014	0.015	0.015	0.015	0.014	0.024	0.027	0.025	0.025	0.026

Gambar 21. Perbandingan kecepatan pelayanan

IV. KESIMPULAN

E-government menjadi salah satu layanan yang dapat mempercepat proses transaksi dan komunikasi antara petugas pemerintahan dengan masyarakat. Namun karena belum adanya integrasi data proses pelayanan dari *e-government* menjadi kurang maksimal. Karena beberapa layanan juga bergantung dari layanan lainnya. Selain itu karena belum adanya integrasi memungkinkan bahwa data dapat mengalami duplikasi data yang juga memungkinkan ada data yang tidak sinkron.

Dari penelitian ini telah dibuat aplikasi *e-government* yang terintegrasi dengan SOA. Proses pembuatan *e-government* ini mempunyai tiga tahapan utama. Pertama adalah proses pembuatan database. Database harus dibuat dengan memperhatikan skema yang benar agar tidak menimbulkan duplikasi data. Kedua adalah proses pembuatan *web service*. Pembuatan *web service* juga memperhatikan *tool* apakah yang digunakan, seperti bahasa pemrograman yang dipakai, *web server* yang digunakan, dll. Ketiga adalah implementasi *web service* kedalam *user interface*, baik dalam bentuk *web application* ataupun *mobile application*.

Dengan adanya pembangunan aplikasi *e-government* yang terintegrasi menggunakan SOA, proses dapat dijalankan dengan lebih cepat, karena arsitektur ini membuat aplikasi kedalam bentuk *web service*. Sehingga dapat mengkesinambungkan antara layanan yang satu dengan layanan yang lainnya. Selain itu juga adanya duplikasi data dan adanya data yang tidak sinkron dapat di atasi dengan baik. Penghematan dana juga menjadi salah satu penerapan dari integrasi *e-government* baik bagi petugas pemerintah ataupun masyarakat.

REFERENSI

- [1] Dwika Putri, Tania. 2015. Penerapan Electronic Government (E-Gov) Melalui Penyelenggaraan Situs Web Pemerintah Provinsi Riau Untuk Mewujudkan Transparansi Publik. Pekanbaru: Universitas Riau.
- [2] Instruksi Presiden Republik Indonesia Nomor 3 Tahun 2003 Tentang Kebijakan dan Strategi Nasional Pengembangan E-Government.
- [3] Dyah Puspito Dewi Widowati. Inilah Peringkat E-Government Indonesia Berdasarkan Survei PBB 2016. <https://bpptik.kominfo.go.id/2016/09/09/2190/inilah-peringkat-e-government-t-indonesia-berdasarkan-survei-pbb-2016/>. Diakses Tanggal 15 Mei, 2018
- [4] https://id.wikipedia.org/wiki/Kabupaten_Sidoarjo. Diakses pada Tanggal 15 Desember 2018.
- [5] Syamsul Huda, <http://www.unusida.ac.id/artikel/e-government-untuk-peningkatkan-pelayanan>, Diakses tanggal 17 Mei, 2018
- [6] Imam Wahyudiyanta, 28 September 2016. <https://news.detik.com/berita-jawa-timur/d-3308888/40-daerah-aplikasikan-e-government-surabaya-dan-sidoarjo>
- [7] Kayode, Ajibade Ibrahim. *Using SOA with Web Services for effective data integration of Enterprise Pharmaceutical Information Systems*. UK/Ghana. University College.
- [8] Hai, Henry. Et al. 2009. *SaaS and Integration Best Practices*. FUJITSU Sci. Tech. J.
- [9] Dennis, et al. 2014. Integrasi Sistem Informasi Pada Pemerintahan Dengan E-Government. Jakarta. Universitas Bina Nusantara.
- [10] Feridi. Mengenal RESTful Web Service. <https://www.codepolitan.com/mengenal-restful-web-services>. Diakses Tanggal 18 Januari 2019.
- [11] As'ad, et al. 2016. Importance of service integration in e-government implementations. Sheffield Hallam University Research Archive.
- [12] T. Berners-Lee, et al. 2005. Uniform Resource Identifier (URI): Generic Syntax. Network Working Group.
- [13] Sumirah, MuahmmadZohri. 2016. Integrasi Data Dalam Proses Layanan Publik Menuju Percepatan E-Government. Mataram. AMIKOM-ASM.
- [14] Sutanta, Edhy. Mustofa, Khabib. 2012. Kebutuhan Web Service Untuk Sinkronisasi Data Antar Sistem Informasi Dalam E-Gov Di Pemkab Bantul Yogyakarta. Yogyakarta. JURTIK - STMIK BANDUNG.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terimakasih kepada Pemerintah Daerah Kabupaten Sidoarjo beserta masyarakatnya. Serta terima kasih kepada para kerabat yang telah membantu saya menyelesaikan makalah ini.