

Penerapan JAVA Spring DAO Pada Pengembangan Sistem Informasi Administrasi Rukun Tetangga (SIA-RT) (Studi kasus: Perum Nuansa Hijau)

Febri Damatraseta Fairuz, Septian Cahyadi, Deni Muharam, Victor Ilyas Sugara
Program Pasca Ilmu Komputer Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Pesanggrahan, Jakarta Selatan, telp: 021-
5853753/fax: 021-5853752
e-mail: fairuz@gmail.com

Abstrak

Salah satu sumber informatika yang dapat diakses melalui websitenya secara online dan realtime. Dengan arsitektur client server, aplikasi web kini dapat menyajikan informasi yang dinamis. Salah satunya adalah untuk meningkatkan kinerja ketua lingkungan (RT) untuk pengolahan data penduduk dan layanan surat-menyurat bagi masyarakat di RT nuansa hijau. Peningkatan populasi membutuhkan pembaharuan dari system yang lebih baik untuk membantu pengolahan data. sehingga pengolahan data dan layanan korespondensi dapat dilakukan secara berkala dan terus menerus, akurat dan efektif. Dalam jurnal ini menggunakan framework spring sebagai dasar platformnya dan untuk arsitektur kerangka ini menggunakan data akses objek (DAO). Kerangka yang memiliki keunggulan yang berbeda dari kerangka kerja yang lain, container ringan, inversi control, aspek berorientasi, dan penggunaan modul yang diperlukan. DAO mampu memberikan portabilitas untuk aplikasi beralih keberbagai jenis mesin database.

Kata kunci: Framework Spring, Data Akses Objek (DAO), Administrasi kependudukan, Mesin database, Administrasi korespondensi.

1. Pendahuluan

Di zaman globalisasi saat ini berbagai lembaga pemerintahan mulai memanfaatkan perkembangan teknologi tersebut untuk memperbaiki system dan meningkatkan kualitas kinerja dari lembaga pemerintah tersebut. Peningkatan kinerja terhadap pelayanan dan pengolahan data penduduk masyarakat Perumahan Nuansa Hijau menjadi suatu tuntutan pokok bagi ketua Rukun Tetangga (RT). Populasi penduduk yang semakin meningkat mengharuskan adanya pembaharuan system yang lebih baik untuk membantu proses pengolahan data. Sehingga proses pengolahan data penduduk dapat dilakukan secara berkala maupun terus-menerus, berkesinambungan, akurat dan efektif.

Selain tuntutan pokok dalam hal mengelola data kependudukan, RT juga memiliki tuntutan pokok lainnya dalam hal penyediaan layanan permintaan Surat pengantar dan Surat Keterangan. Surat Pengantar biasanya ditujukan kepada kantor Kelurahan setempat untuk dapat memberikan persetujuan seperti, memperpanjang masa berlaku KTP (Kartu Tanda Penduduk), membuat SIM (Surat Izin Mengemudi), membuat SKCS (Surat Keterangan Catatan Sipil), ataupun untuk perizinan menikah. Sedangkan Surat Keterangan biasanya diperuntukan untuk perihal Surat Keterangan Domisili Penduduk, Surat Keterangan Kehilangan KK (Kartu Keluarga), Surat Keterangan Pindah Agama, Surat Keterangan Kematian, Surat Keterangan Kelahiran,

Surat Keterangan Tidak Mampu, Surat Keterangan Duda/Janda, Surat Keterangan Pindah, Surat Keterangan Usaha (SKU), Surat Keterangan Domisili Perusahaan, dan lain sebagainya.

Web merupakan salah satu sumber informasi yang banyak digunakan oleh masyarakat. Salah satu produk yang membantu kegiatan dan pekerjaan secara *online* dan *realtime* adalah suatu aplikasi web. Aplikasi tersebut dibuat dengan tujuan agar pemakai dapat berinteraksi dengan penyedia informasi secara mudah dan cepat.

Aplikasi web kini tidak terbatas lagi hanya sebagai penyedia informasi yang statis. Dengan adanya arsitektur *client-server*, aplikasi web juga mampu memberikan informasi yang dinamis, dengan cara melakukan koneksi terhadap *database*. Seorang *developer* web akan mengalami kesulitan jika harus membangun kode-kode program dalam setiap kelas *business logic* untuk koneksi aplikasi web yang dibuat ke *database*. Hal ini memudahkan *developer* sehingga tidak perlu membuat kode program yang bermacam-macam untuk koneksi ke *database*. Penerapan dari pemaparan diatas dalam ilmu komputer dikenal sebagai *Data Access Object (DAO)* [Noe Lopez-Benitez et al: 2006]. Untuk memfasilitasi penerapan DAO maka digunakanlah sebuah platform *Framework JAVA Spring*, *framework spring* adalah salah satu *framework* milik JAVA yang ringan, kinerja yang baik, mudah diuji, dan kode dapat digunakan kembali.

Tujuan dari penelitian ini adalah membangun sebuah sistem untuk membantu dalam pengelolaan pelayanan administrasi kepada masyarakat di lingkungan organisasi RT. Pengelolaan administrasi yang akan ada di SIA-RT ini bersifat sistemik, sehingga akan memberikan kemudahan dan keuntungan baik bagi Ketua RT ataupun warganya.

2. Metode Penelitian

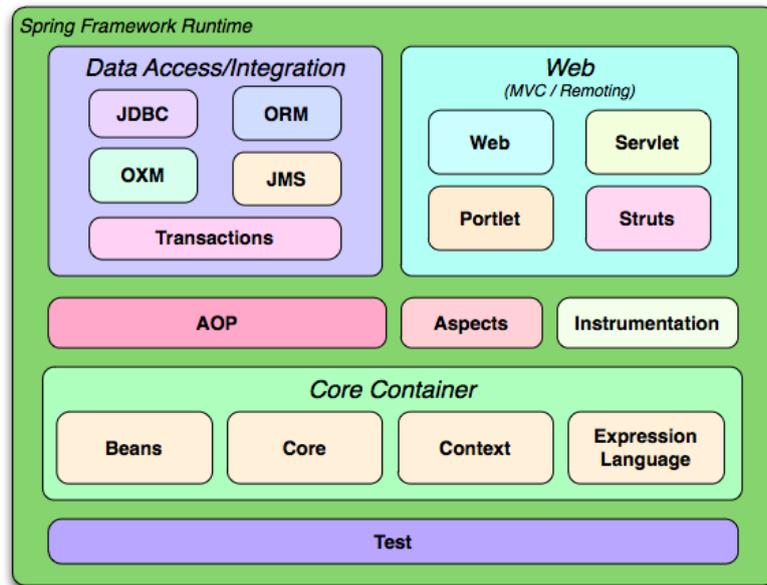
Dalam penelitian ini kami melakukan 2 metode pendekatan untuk mengumpulkan informasi yang terkait:

1) Metode Pengamatan dengan Wawancara

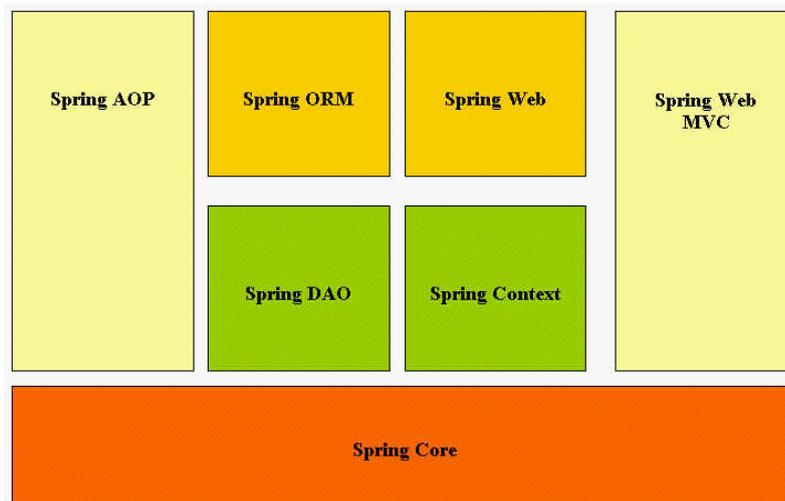
Tahap ini meliputi pengumpulan informasi terkait sistem yang sedang berjalan saat ini, serta mendapatkan gambaran secara langsung mengenai aplikasi yang akan dibangun dari top manajemen yang dalam proyek ini adalah ketua RT. Dalam tahap ini juga dilakukan pengumpulan dokumen-dokumen surat yang akan masuk dalam pelayanan RT.

2) Metode Prototipe

Pada tahapan ini analisis memberikan beberapa sampel aplikasi pelayanan RT yang telah ada dan umum digunakan dalam pelayanan administrasi tingkat RT. Dari proses tersebut, maka diperoleh gambaran aplikasi yang akan dikembangkan untuk menangani masalah yang ada pada Komplek Nuansa Hijau.



Gambar 1. Module Spring Framework



Gambar 2. Arsitektur Spring Web

Salah satu komponen utama Spring adalah AOP Framework, AOP framework digunakan untuk menyediakan layanan Enterprise, terutama sebagai pengganti EJB. Layanan terpenting dalam layanan ini adalah untuk mendekralif manajemen transaksi, yang telah disediakan dalam abstraksi spring transaction. Untuk memungkinkan pengguna dalam menerapkan AOP dalam penggunaan OOP.

Dalam membuat DAO, seorang developer harus mempertimbangkan apakah kelas yang dibuat berupa singleton atau tidak. Bahasa Object Oriented Programming (OOP) mengunggulkan konsep pewarisan (inheritance) suatu kelas sebagai cara utama untuk memperluas fungsionalitasnya. *Singleton* secara efektif dapat mencegah konsep pewarisan tersebut.

DAO dibuat menjadi singleton ketika developer tidak ingin terjadi multiple instances dari kelas DAO. Jika developer ingin mendesain suatu single-user application dan tidak menghiraukan akses konkuren, DAO singleton sangat cocok dan

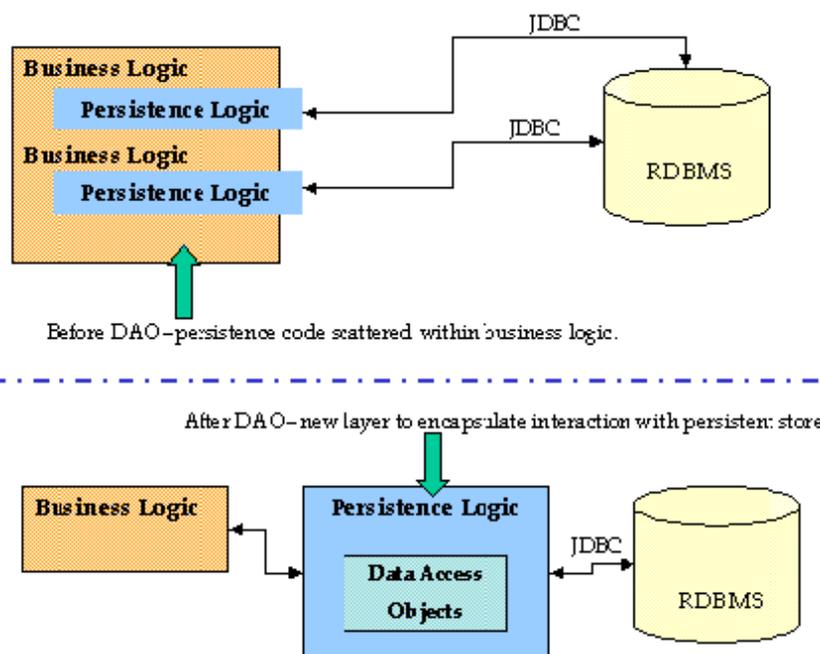
mudah untuk diimplementasikan. Jika terdapat beberapa user dengan akses konkuren, developer dapat menggunakan konsep DAO thread-safe singleton. Pada DAO singleton, developer harus memastikan bahwa setiap method di dalamnya bersifat atomik dan tidak menampung state pada setiap pemanggilannya.

Hal yang mencirikan bahwa suatu DAO singleton adalah hanya ada satu instansiasi (satu objek) yang dibuat dari kelasnya. Contoh kode program yang mencirikannya adalah sebagai berikut:

```
@Repository
public class LetterDao {

    @Autowired private DataSource dataSource;
    @Autowired private UserDao userDao;
```

Gambar 3. Kode program satu objek



Gambar 4. Struktur aplikasi sesudah dan sebelum menggunakan DAO

Data Access Object merupakan integrasi pola desain lapis sebagai katalog dalam inti *J2EE Desain Pattern*. Hal ini menyimpan akses encapsulasi dan memanipulasi kode kedalam lapisan terpisah. Penyimpanan yang kokoh dalam konteks penulisan ini adalah RDBMS. Pola ini memperkenalkan lapisan abstraksi antara tingkat logika bisnis dan tingkat penyimpanan yang tetap, seperti yang ditunjukkan pada gambar 4. Business Logic mengakses RDBMS melalui akses data. Layer dipisahkan dengan arah pada kode aplikasi dan memasukkannya pada *flexibility*. Idealnya, perubahan yang dibuat untuk sumber data, seperti *switching vendor* database atau golongan, akan memerlukan perubahan pada object saja dan harus memiliki dampak pada *business-objects*.

Metode setter digunakan untuk memberikan set nilai pada suatu property. Fungsi ini sering digunakan untuk memvalidasi data yang masuk sebelum data tersebut diisikan pada suatu property. Sedangkan getter digunakan untuk menghasilkan suatu nilai dari hasil perhitungan. Berikut ini adalah contoh penerapan setter dan getter pada SIA-RT:

```
public class Letter{
```

```
private Integer letter_id;
private String number;
private Integer type;
private String content;
private User user;
private String status;
private Timestamp created_at;
private String created_by;

public String getNumber() {
    return number;
}

public void setNumber(String number) {
    this.number = number;
}

public String getStatus() {
    return status;
}

public void setStatus(String status) {
    this.status = status;
}

public Integer getLetter_id() {
    return letter_id;
}

public void setLetter_id(Integer letter_id)
{
    this.letter_id = letter_id;
}

public Integer getType() {
    return type;
}

public void setType(Integer type) {
    this.type = type;
}

public String getContent() {
    return content;
}

public void setContent(String content) {
    this.content = content;
}

public User getUser() {
    return user;
}

public void setUser(User user) {
    this.user = user;
}

public Timestamp getCreated_at() {
    return created_at;
}

public void setCreated_at(Timestamp
created_at) {
    this.created_at = created_at;
}

public String getCreated_by() {
    return created_by;
}

public void setCreated_by(String created_by)
{
    this.created_by = created_by;
}
}
```

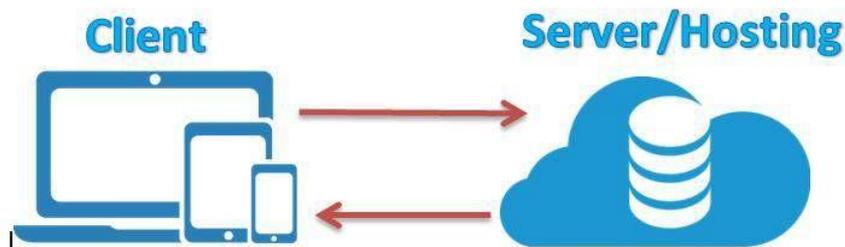
Pada method `executeQuery()` digunakan kata kunci, yaitu `ResultSet`. Suatu objek `ResultSet` menentukan penunjukkan kursor pada baris data yang sedang dipilih. Secara default, objek `ResultSet` tidak bisa di-update dan hanya dapat bergerak ke depan saja. Kelas interface dari `ResultSet` menyediakan *method getter* seperti `getBoolean`, `getLong`, dsb. Untuk mengambil nilai kolom pada baris yang sedang ditunjuk. Kode program dari method `executeQuery` dapat dilihat seperti di bawah ini.

```
@Autowired private DataSource dataSource;
@Autowired private UserDao userDao;

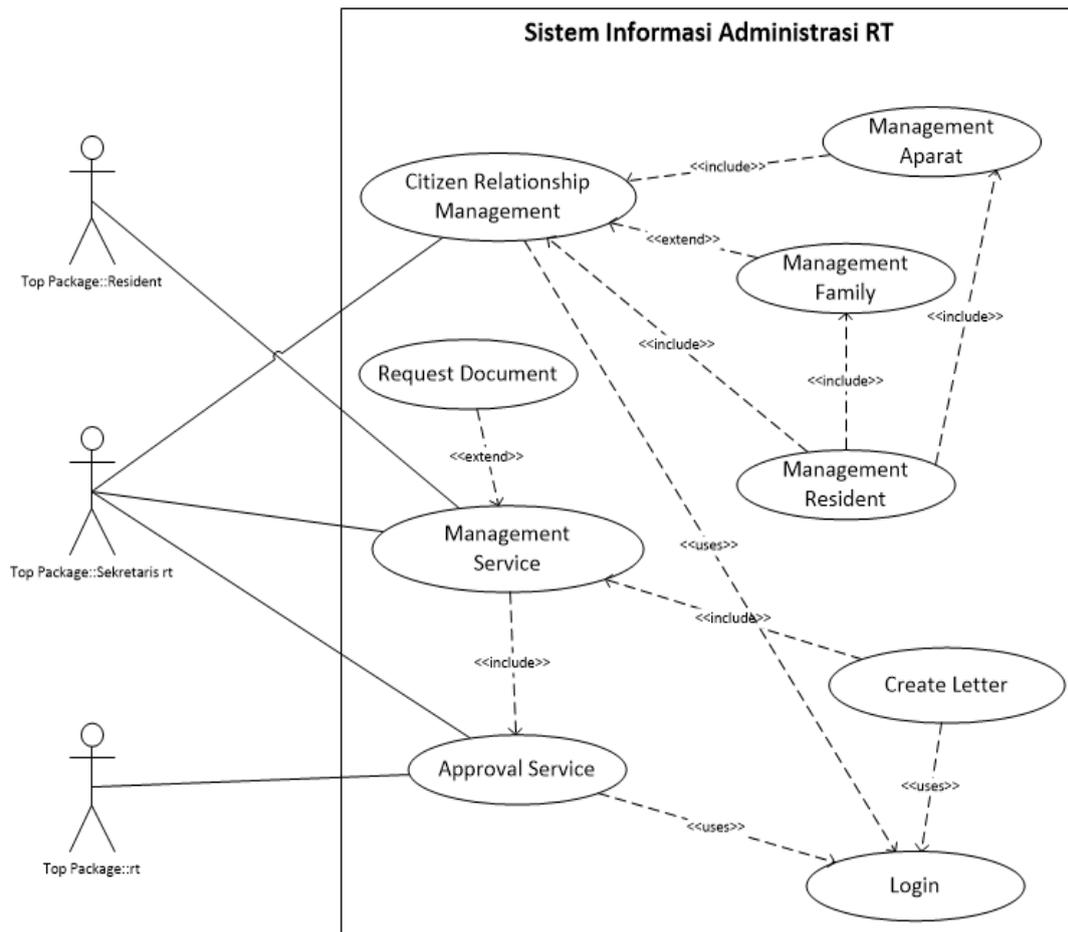
public List<Letter> showAll() throws
SQLException, Exception{
    List<Letter> values = new
    ArrayList<Letter>();
    Connection c = dataSource.getConnection();
    String sql_show_all = "select * from letters
    order by created_at desc";
    PreparedStatement ps =
    c.prepareStatement(sql_show_all);
    ResultSet rs = ps.executeQuery();
    while(rs.next()){
        Letter letter = konversiResultSet(rs);
        values.add(letter);
    }
    c.close();
    return values;
}
```

3. Hasil dan Analisis

Pada perancangan sistem SIA-RT didukung oleh Arsitektur Sistem, Prototyping dan beberapa pemodelan UML sehingga memudahkan proses pembuatannya. Pemodelan tersebut diantaranya *use case diagram*, *Class Responsibility Collaborator (CRC) Models*, *Activity Diagram*, *Class Diagram*, *Package Diagram*, dan *Sequence Diagram*.



Gambar 5. Arsitektur Sistem



Gambar 6. Use Case SIA-RT

Pada gambar 6 menjelaskan bahwa pengguna dari aplikasi sebanyak 3 aktor yaitu, *Resident*, *Secretaris RT* dan *RT*. *Resident* dapat melakukan proses *Management Service* untuk dapat melakukan *Request Document* dimana *Resident* dapat melakukan permintaan terhadap pembuatan dokumen seperti permintaan pembuatan Surat Pengantar atau Surat Keterangan. *Secretaris RT* dapat melakukan proses *Citizen Relationship Management* yang meliputi *case Management Aparat*, *Management Family* dan *Management Resident*. Dan juga *Seckretaris RT* dapat melakukan proses *Management Service* termasuk *case Create Letter* untuk membuat permintaan surat dan melakukan proses persetujuan surat yang nantinya akan diberikan kepada *RT* untuk diberikan persetujuan terakhir. Dimana proses tersebut berada pada proses *Approval Service*. Pada proses *Citizen Relationship Management*, *Create Letter*, dan *Approval Service* para actor diharuskan melakukan proses *Login* agar bisa mengakses case tersebut.

Tabel 1. Pengujian dengan *BlackBoxTesting*

No	Nama Pengujian	Kondisi Pengujian	Hasil Pengujian
1	Pembuatan Surat secara <i>public</i>	1. Mengosongkan nama warga, jenis surat, email, isi surat dan tidak mencentak persetujuan langsung klik tombol	1. Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar

		2. <i>send</i> . Mengisi semua <i>field</i>	2. Sistem akan menampilkan “ <i>Your request has been successfully</i> ”
2	Pengecekan kode unik untuk mengetahui status permintaan secara <i>public</i>	1. Mengosongkan <i>codenumber</i> langsung klik tombol <i>search</i> 2. Mengisi <i>codenumber</i>	1. Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar 2. benar Sistem akan menampilkan info status dari <i>codenumber</i>
3	Login Portal	1. Mengosongkan <i>username</i> dan <i>password</i> atau hanya mengisi salah satu dari 2. <i>field</i> yang disediakan Mengisi semua <i>field</i>	1. Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar 2. benar Berhasil masuk kedalam portal Admin
4	Data Aparat	1. Menambahkan aparat baru dengan tidak mengisi <i>user</i> , <i>password</i> , <i>role</i> dan periode langsung klik 2. tombol <i>Save changes</i> 3. Mengisi semua <i>field</i> Klik tombol <i>Non Active</i> milik Aparat yang statusnya <i>Active</i> 4. Klik tombol <i>Active</i> milik Aparat yang statusnya <i>Non Active</i>	1. Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar 2. benar 3. Berhasil menambahkan Aparat 4. Sistem merubah status Aparat tersebut menjadi “ <i>Non Active</i> ” Sistem merubah status Aparat tersebut menjadi “ <i>Active</i> ”
5.	Data Resident	1. Menambahkan data warga dengan tidak mengisi semua field yang disediakan. Mengisi semua field data warga 3. Memilih warga dan menghapusnya dengan klik tombol “ <i>Delete</i> ” 4. Memilih warga dan klik tombol <i>Preview</i> lalu muncul pop-up data warga klik tombol <i>Edit</i> untuk mengubah data warga sesuai field yang ingin diubah, lalu klik tombol <i>Save Changes</i>	1. Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar 2. benar Berhasil menambahkan data warga 3. data warga Sistem akan menampilkan pop-up “ <i>Are you sure want to delete user ‘currentwarga’ ?</i> ”. Klik <i>Delete</i> dan berhasil terhapus Berhasil diubah
6.	Data Kepala Keluarga	1. Menambahkan keluarga baru dengan tidak	1. Sistem akan menolak permintaan tersebut dan

		<p>mengisi field yang disediakan</p> <ol style="list-style-type: none"> Mengisi semua field untuk menambahkan keluarga Menghapus keluarga dengan klik tombol <i>Delete</i> Memilih keluarga dan klik tombol Preview lalu muncul pop-up data warga klik tombol Edit untuk mengubah data warga sesuai field yang ingin diubah, lalu klik tombol Save Changes Memilih keluarga dan klik tombol Preview lalu muncul pop-up data warga klik <i>Print</i> untuk mencetak Formulir Isian Warga Baru 	<p>meminta untuk mengisi <i>field-filed</i> tersebut dengan benar</p> <ol style="list-style-type: none"> Berhasil menambahkan keluarga baru Berhasil menghapus data keluarga Berhasil diubah Sistem akan mencetak Formulir Isian Warga Baru yang telah terhubung dengan printer
7.	Laporan Kependudukan	<ol style="list-style-type: none"> Pada filter laporan mencoba untuk tidak mengisi <i>startdate</i> dan / <i>enddate</i> lalu klik tombol <i>Search</i> Mengisi semua field Mencetak laporan dengan klik tombol <i>Printer</i> 	<ol style="list-style-type: none"> Sistem akan menolak permintaan tersebut dan meminta untuk mengisi <i>field-filed</i> tersebut dengan benar Sistem menampilkan jumlah warga yang telah terdaftar sebagai warga dimulai dari <i>startdate</i> sampai <i>enddate</i> yang telah diinputkan Sistem akan mencetak Laporan Data Kependudukan yang telah terhubung dengan printer
8.	Pelayanan Masyarakat	<ol style="list-style-type: none"> Mencetak permintaan Surat Keterangan/Surat Pengantar Merubah Status Permintaan dengan mengklik tombol <i>Preview</i> dan memilih status 'CANCEL'/'DRAFT' 	<ol style="list-style-type: none"> Sistem akan mencetak Surat Keterangan/Pengantar yang telah terhubung dengan printer Berhasil diubah

4. Kesimpulan

Sistem Informasi Administrasi Rukun Tetangga (SIA-RT) dengan menerapkan JAVA Spring dapat disimpulkan Layanan administrasi surat menyurat ditingkat RT memberikan dampak yang sangat positif karena mengurangi aktifitas warga yang harus datang ke rumah RT untuk meminta layanan yang ditawarkan. Dengan adanya

sistem ini pengelolaan dokumen menjadi lebih efisien karena tersimpan di dalam database. Pemilihan Spring DAO sebagai arsitekturnya dikarenakan *developer* dapat focus pada pengorganisasian program sebagai kumpulan kelas yang terdefinisi dengan bagus. DAO juga berhubungan dengan pengabstraksian, masing-masing kelas berkorespondensi dengan satu abstraksi dan hanya satu-satunya. Dengan menggunakan DAO dapat membantu *developer* dalam membangun aplikasi yang komplek menjadi sederhana seperti yang dikatakan oleh Myers: "Suatu Aksi mempartisi program menjadi komponen-komponen individual dapat mereduksi kompleksitas."

Daftar Pustaka

- [1] Sri Dharwiyanti. 2003. Pengantar Unified Modeling Language (UML).
- [2] Beck Kent, Cunningham Ward. 2012. "A laboratory for teaching object oriented thinking". ACM SIGPLAN Notices (india, NY, USA: Abg) 24 (10): 1–6.
- [3] Whitten Jeffrey L et al. 2004. Metode Desain & Analisis Sistem, Edisi 6, Edisi International, Mc GrawHill, ANDI, Yogyakarta.
- [4] OMG UML. 2007. OMG Unified Modeling Language (OMG UML). Infrastructure, V2.1.2] p. 149. 2007.
- [5] Noe Lopez-Benitez et al. 2006. Noe Lopez-Benitez; Shin, Michael; Anderson Per; Silenus-A Federated Service-Oriented Approach To Distributed File Systems. Dean of the Graduate School.
- [6] Undang Undang Dasar Tahun 2013. <https://spring.io/guides/>. Di akses pada tanggal 08 Juni 2016 11:50 AM

