

## Perancangan Notifikasi Pesan Disposisi Pada Sistem Pengelolaan Surat Dinas Menggunakan Layanan *API Media Social* (Studi Kasus: Dinas Pertanian Kota Salatiga)

Billy Gerardhy Lainsamputty<sup>1</sup>, Suprihadi<sup>2</sup>

<sup>1,2</sup>Fakultas Teknologi Informasi Universitas Kristen Satya Wacana  
Jl. Dr. O. Notohamidjojo 1-10, Salatiga 50711, Indonesia  
Email : <sup>1</sup>672013094@student.uksw.edu, <sup>2</sup> suprihadi@staff.uksw.edu

### Abstract

*Based on the Department of Agriculture which still does not have the tools or media to manage the letter of service, both the incoming and outgoing mail which impact on the weak supervision follow-up of the service letter due to ignorance that the letter should be processed. This study was made as a visual and text notification of incoming and outgoing mail data with more detailed archive storage. This website application is supported by using social media telegram API service that is used as notification in the form of notification to the employee so that the Service Letter can be followed up immediately.*

**Keywords:** *Notifications, APIs, Social Media, Mail Management, Website*

### Abstrak

Didasarkan pada Dinas Pertanian yang masih belum memiliki alat atau media untuk mengelola surat dinas, baik surat masuk maupun surat keluar yang berdampak pada lemahnya pengawasan tindak lanjut dari surat dinas tersebut yang dikarenakan ketidaktahuan bahwa adanya surat yang harus segera diproses. Penelitian ini dibuat sebagai pemberitahuan visual dan teks tentang data surat yang masuk dan keluar juga dengan penyimpanan arsip yang lebih terperinci. Aplikasi *website* ini didukung dengan menggunakan layanan *API Media Social Telegram* yang digunakan sebagai pemberitahuan dalam bentuk notifikasi kepada pegawai agar Surat Dinas agar dapat segera ditindak lanjuti.

**Kata kunci:** *Notifikasi, API, Media Social, Pengelolaan Surat, Website*

## 1. Pendahuluan

Perkembangan teknologi informasi khususnya teknologi *website* dan pemanfaatan *Application Programming Interface* (API) sebagai sarana notifikasi semakin pesat dan bervariasi. Saat ini, *website* memiliki fungsi tidak lagi sebagai sumber informasi dan bersifat pasif, tetapi telah dikembangkan dapat bersifat dinamis. Oleh karena itu, peran *website* menjadi media komunikasi dua arah dan memiliki proses timbal balik dengan pengguna. Perkembangan teknologi *website* tersebut sejalan dengan Inpres No. 3 tahun 2003 tentang kebijakan dan strategi

nasional pengembangan elektronik government. Inpres No. 3 tahun 2003 tersebut juga berisi tentang pertimbangan pemanfaatan teknologi informasi dalam alur layanan pemerintahan dalam upaya peningkatan efisiensi, efektivitas, transparansi serta penyelenggaraan pemerintahan yang akuntabel. Dengan teknologi *website* saat ini, diharapkan juga diperoleh aliran data dan informasi yang lebih optimum antara lembaga pemerintahan dengan masyarakat, dan sebaliknya, sehingga peran *website* bagi publik dapat dipergunakan sebagai penyambung aspirasi. Dengan demikian, efisiensi dan efektivitas dalam penyelenggaraan pemerintahan dapat ditingkatkan melalui perkembangan teknologi *website* yang dinamis.

Berdasarkan wawancara dengan Kepala Dinas, dan juga kepada para pegawai di lingkungan Dinas Pertanian Kota Salatiga, diketahui bahwa sebagian besar para pegawai sulit dalam mengakses dan membutuhkan waktu cukup lama dalam memproses surat masuk maupun surat keluar, yang disebabkan karena mekanisme pemberitahuan informasi terkait surat yang harus diproses belum dilakukan secara *realtime*. Kepala Dinas pun tidak dapat memonitoring aktivitas laporan kemajuan tindak lanjut surat hasil proses yang ditujukan bagi para pegawai karena tidak adanya *record* yang menyimpan terkait surat tersebut.

Penyampaian pesan atau informasi pada era globalisasi dapat melalui berbagai media antara lain media sosial, media elektronik dan media cetak. Pemilihan media dalam menyampaikan pesan harus didasarkan pada isi pesan yang akan dikirim maupun diterima. Sistem yang diusulkan dalam penelitian ini yaitu dengan memanfaatkan teknologi notifikasi menggunakan notifikasi media sosial, karena dianggap efisien dalam penggunaan waktu dan media sosial cukup dikenal oleh para pegawai dinas pertanian Kota Salatiga. Selain itu, sarana mengakses juga cukup menggunakan *smartphone* sederhana sehingga dapat dibawa kemana-mana, diharapkan pesan dapat diterima secara tepat dan cepat.

Berdasarkan latar belakang tersebut, maka penelitian ini bertujuan merancang sistem pada sebuah aplikasi *website* yang menggunakan API yang digunakan sebagai media memproses surat dan media pengingat tentang adanya surat yang harus segera diproses. Pembangunan sistem berbasis *web* karena *website* lebih mudah digunakan oleh semua kalangan masyarakat, sedangkan pemanfaatan teknologi API digunakan sebagai media pemberitahuan atau notifikasi kepada para pegawai Dinas Pertanian Kota Salatiga secara *uptodate* agar para pegawai dapat segera melakukan proses tindak lanjut pada surat yang diterima. Notifikasi akan diterima para pegawai melalui sosial media yang telah ditentukan pada penelitian ini.

Adapun batasan masalah pada penelitian ini adalah media notifikasi ini dibangun menggunakan *platform website*, API yang digunakan adalah API yang disediakan oleh aplikasi media sosial yang dipilih, pesan yang dikirim adalah info

sederhana dalam bentuk teks, tidak membahas keamanan data, dan tidak membahas kecepatan notifikasi yang disebabkan faktor *bandwith* internet.

## 2. Tinjauan Pustaka

Penelitian yang dilakukan Maulana (2015) di Kabupaten Garut menggunakan *framework* dan bahasa pemrograman PHP telah menghasilkan suatu aplikasi yang mampu mengelola surat masuk dan surat keluar secara lebih efektif, baik dalam segi waktu, tenaga dan biaya. Hal tersebut disebabkan karena aplikasi mampu memberi pengaruh pada prosedur kerja, antara lain proses penyimpanan dan pencarian dokumen surat. Pengembangan sistem informasi ini yang diperlukan adalah menyediakan layanan pembuatan laporan dokumen surat kepada pegawai berdasarkan periode yang diinginkan [1].

Pada penelitian Fitriana (2015), telah dikembangkan sistem pengolahan yang dapat membantu pegawai dalam mengelola proses *input* surat masuk dan surat keluar. Meningkatkan sistem pengendalian surat dari sistem manual menjadi sistem terkomputerisasi yang membuat proses pengarsipan surat terintegrasi antara bagian dan adanya backup surat apabila surat asli hilang atau rusak. Dengan pengolahan *database* berbasis elektronik menjadikan data mitra kerja lebih cepat diakses [2].

Pada penelitian Hasan (2016), didapatkan pada penelitian ini penggunaan *Application Programming Interface* (API) yang mampu memberikan informasi kepada mahasiswa dalam melihat jadwal bimbingan agar para mahasiswa dapat menerima informasi secara *realtime* dan *up to date*. Aplikasi tersebut dapat memberi informasi kepada mahasiswa mengenai jadwal bimbingan yang diberikan dosen, agar para mahasiswa dapat melakukan bimbingan tepat waktu sesuai jadwal dengan melakukan *request*. Aplikasi telah berguna khususnya bagi mahasiswa dalam melakukan proses bimbingan agar sesuai dengan ketentuan waktu yang telah diinformasikan oleh dosen pembimbing [3].

Pada penelitian ini telah dilakukan Perancangan Sistem Notifikasi Pesan Disposisi Pada Pengelolaan Surat Menggunakan Layanan API *Media Social* (Studi Kasus: Dinas Pertanian Kota Salatiga). Sistem dirancang berbasis *website application* yang mampu memberikan notifikasi pada sosial media dengan memanfaatkan *Application Programming Interface* (API) yang disediakan media sosial tersebut, sehingga dapat memberikan informasi pesan adanya surat secara *realtime* kepada para pegawai, dan membantu memudahkan pengelolaan dokumen surat di lingkungan Dinas Pertanian Kota Salatiga.

*Application Programming Interface* (API) adalah sebuah teknologi untuk memfasilitasi pertukaran informasi atau data antar aplikasi perangkat lunak. API adalah antarmuka virtual antara dua fungsi aplikasi yang saling bekerja sama. API dapat berisi teknik *programmer* dalam memanfaatkan suatu fitur tertentu dari

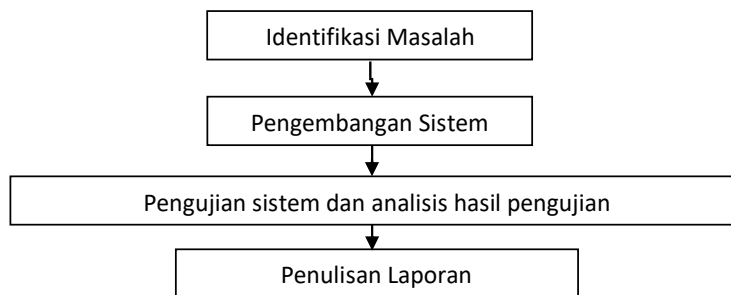
sebuah komputer. Dengan demikian, API dapat tersedia dalam sistem *windowing*, sistem *file*, sistem basis data dan sistem jaringan [4].

Sebuah situs *Website* dapat didefinisikan sebagai kumpulan halaman-halaman elektronik atau *web page* yang berisi informasi atau data. Sebuah *website* pada umumnya memiliki alamat penunjuk yang unik, yaitu *domain*, biasanya memiliki konten terdiri dari tipe format teks, gambar, tabel, grafik, kutipan, video, audio, dan format visual lainnya. *Website* diakses publik melalui *internet protocol* (IP) dalam sebuah jaringan internet, atau diakses secara *offline* melalui jaringan LAN. *Website* dilihat dari sisi pemilik, dapat dikelompokkan antara lain *website* pribadi, komersial, pemerintahan, dan lainnya bahkan untuk kepentingan laba maupun nirlaba yang dipublikasikan secara umum. Selain itu, *website* dilihat dari sisi fungsi, dapat digolongkan tujuan khusus antara lain untuk hiburan, pendidikan, dan sosial [5].

*Telegram* adalah sebuah aplikasi servis pengiriman *message* instan yang tidak dibatasi *platform* berbasis *cloud* yang dapat diakses secara gratis dan non profit. Klien *Telegram* dapat dari berbagai *platform* perangkat seluler dan sistem operasi komputer. Pesan pada *Telegram* dapat berisi *image*, video, audio, stiker, dan semua tipe berkas, dan disediakan layanan pengiriman pesan terenkripsi secara opsional. Layanan *Telegram* ini berupa API yang dapat dipergunakan pengembang independen. Pada 2016, aplikasi *Telegram* telah diakses 100 juta pengguna aktif di tiap bula, serta telah melakukan transaksi pengiriman 15 miliar pesan per hari [6].

### 3. Metode Penelitian

Penelitian yang telah dilakukan melalui tahapan sebagai berikut: 1) Identifikasi Masalah. 2) Pengembangan Sistem. 3) Pengujian Sistem dan Analisis Hasil Pengujian. 4) Penulisan Laporan.



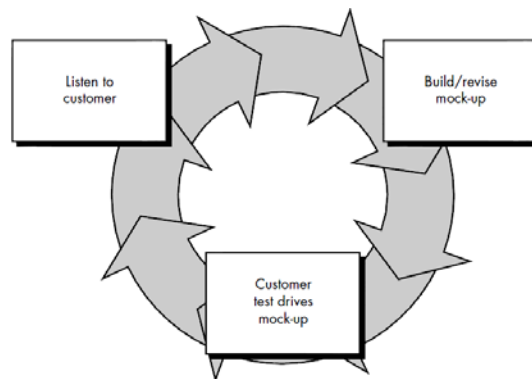
**Gambar 1** Tahap Penelitian

Gambar 1 menjelaskan bahwa tahap awal dilakukan identifikasi masalah, yaitu melakukan pengamatan tentang proses bisnis yang dijalankan, dan menganalisis masalah-masalah yang terjadi pada Dinas Pertanian Kota Salatiga terkait pengelolaan surat masuk dan keluar, dan proses disposisi surat tugas, serta proses progress report pegawai terhadap surat tugas yang diberikan. Pada tahap ini

juga dilakukan wawancara kepada Kepala Dinas dan juga beberapa pegawai tentang faktor apa saja yang menjadi penghambat selama proses bisnis terjadi.

Tahap pengembangan sistem, yaitu kegiatan pembangunan sistem dengan diawali proses perancangan sistem menggunakan metode *Unified Modeling Language* (UML). Tahap ini juga dilakukan perancangan arsitektur sistem, yaitu untuk mengetahui kebutuhan *platform* aplikasi yang digunakan, khususnya posisi *Application Programming Interface Telegram* yang berfungsi memberikan layanan notifikasi secara *realtime*. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *prototype*, karena aplikasi yang dihasilkan supaya dapat lebih sesuai dan tepat seperti yang diharapkan para aktor pengguna di Dinas Pertanian Kota Salatiga.

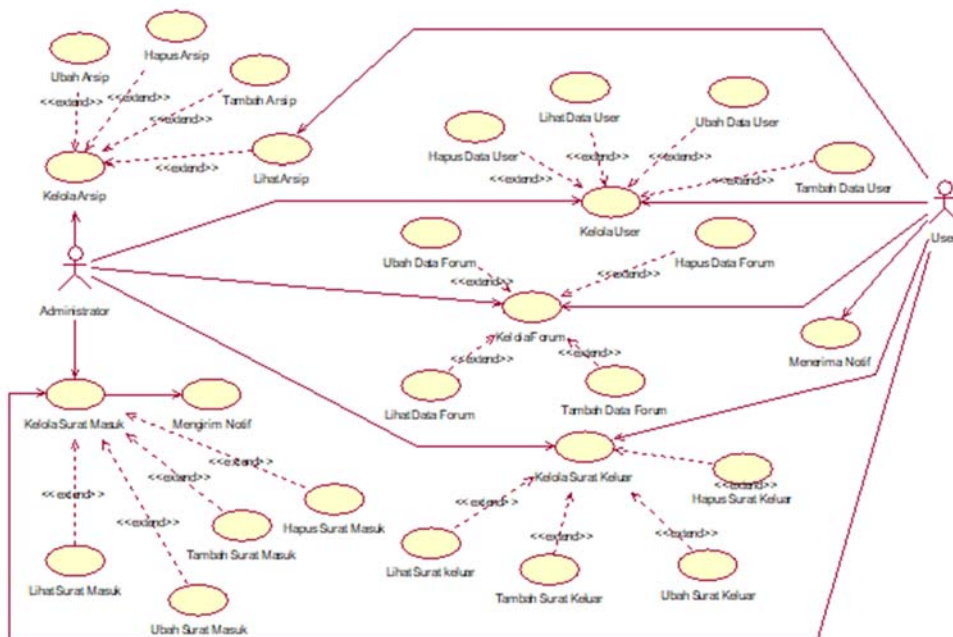
Tahap pengujian sistem dan analisis hasil pengujian dilakukan pada sistem aplikasi pengolahan surat pada Dinas Pertanian Kota Salatiga tersebut melalui media kuisisioner. Dilanjutkan tahap akhir yaitu penulisan laporan yaitu mendokumentasikan hasil rancangan, pengembangan sistem, dan hasil pengujian kedalam bentuk laporan tugas akhir dan artikel ilmiah guna publikasi ke jurnal ilmiah yang bereputasi.



**Gambar 2** Bagan Metode *Prototype* [7]

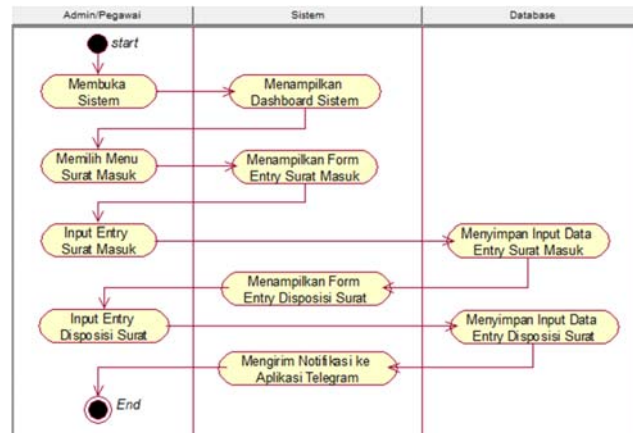
Gambar 2 tahapan metode *prototype*, diawali dengan *listen-to-customer* bertujuan untuk memperoleh informasi kebutuhan pengguna secara lengkap, antara lain alur proses, dan desain antarmuka. Desain alur proses dapat berupa *usecase*, *activity diagram*, *class diagram*, dan *deployment diagram*. Tahapan berikutnya adalah *build/revise mock-up* yaitu membangun aplikasi berdasarkan hasil tahap pertama secara cepat, dan aplikasi tersebut disebut sebagai prototip sampai menjadi aplikasi final. Tahap akhir metode *prototype* adalah *customer test drives mock-up* yaitu melakukan uji dan evaluasi prototip guna memperoleh info kekurangan ada pada prototip, jika dilakukan *testing* masih ada kekurangan, maka kembali pada tahap awal dan akan terus berlanjut hingga prototip menjadi aplikasi sesuai dengan kebutuhan pengguna.

Desain sistem atau aplikasi penelitian ini menghasilkan sebuah *usecase diagram* yang berisi rancangan interaksi antara aktor dengan komponen atau modul sistem yang dibangun [8], hal itu ditunjukkan pada Gambar 3. Pada *usecase diagram* terlihat ada dua aktor aplikasi yaitu *administrator* dan *user* dari Dinas Pertanian Kota Salatiga. Aktor *administrator* dapat mengakses modul kelola *user*, kelola surat masuk, kelola surat keluar, kelola arsip, dan kelola forum. *Administrator* juga dapat mengirimkan notifikasi setelah melakukan pencatatan terhadap adanya surat masuk. Pegawai adalah aktor *user*, diijinkan untuk mengakses kelola profil pada aplikasi, melihat data arsip, dapat melakukan diskusi melalui forum yang tersedia dan mengelola data surat masuk dan keluar lewat fitur yang sudah disediakan.



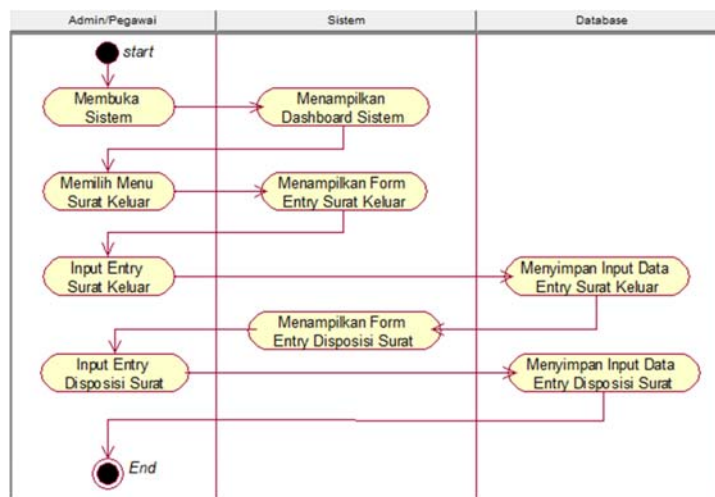
**Gambar 3** Use Case Diagram Sistem Pengelolaan Surat

*Activity diagram* merupakan rancangan sistem atau aplikasi yang berisi desain alur proses aktivitas yang merupakan modul-modul yang terdapat pada desain *usecase diagram*, yang dilakukan oleh aktor aplikasi. Gambar 4 adalah *activity diagram administrator* dalam melakukan pengelolaan surat masuk, dimana diawali dengan membuka sistem yang dilanjutkan dengan sistem menampilkan halaman dimana pada halaman tersebut menampilkan *dashboard* dari aplikasi dan melakukan pemilihan menu untuk pengelolaan surat masuk, diteruskan dengan sistem menampilkan *form entry* surat masuk dimana pegawai meng-*input*-kan data keperluan surat yang akan tersimpan dalam *database* dan dilanjutkan dengan pegawai meng-*input*-kan data disposisi surat yang akan tersimpan dalam *database*. Setelah proses penyimpanan pada *database* selesai, sistem akan melakukan pengiriman notifikasi, alur pengelolaan surat tergambar seperti pada Gambar 4.



Gambar 4 Activity Diagram Administrator Kelola Surat Masuk

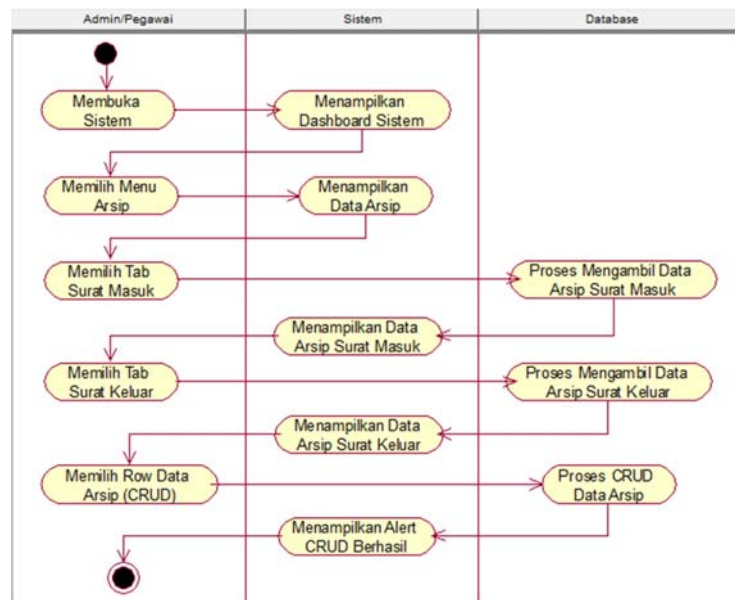
Pada *activity diagram administrator* kelola surat keluar menjelaskan proses alur sistem yang dirancang, pergerakan pada *administrator* Dinas Pertanian Kota Salatiga dalam mengirim atau melakukan pengelolaan terhadap surat keluar, diawali *administrator* membuka sistem yang dilanjutkan dengan sistem menampilkan halaman dimana pada halaman tersebut menampilkan *dashboard* dari aplikasi dan melakukan pemilihan menu untuk pengelolaan surat keluar, diteruskan dengan sistem menampilkan *form entry* surat keluar dimana pegawai meng-*input*-kan data keperluan surat yang akan tersimpan dalam *database* dan dilanjutkan dengan pegawai meng-*input*-kan data disposisi surat yang akan tersimpan dalam *database*. Alur pengelolaan surat tergambar seperti pada Gambar 5.



Gambar 5 Activity Diagram Administrator Kelola Surat Keluar

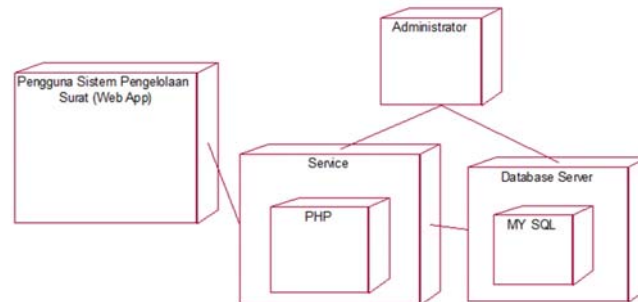
*Activity diagram administrator* kelola data arsip merupakan proses alur sistem pada aktor *administrator* Dinas Pertanian Kota Salatiga dalam melakukan pengelolaan data arsip, *administrator* membuka sistem yang dilanjutkan dengan sistem menampilkan halaman dimana pada halaman tersebut menampilkan *dashboard* dari aplikasi dan melakukan pemilihan menu untuk pengelolaan data

arsip, diteruskan dengan sistem menampilkan seluruh data arsip dimana pegawai dapat melihat seluruh arsip yang ada, baik arsip surat masuk maupun surat keluar. Pada tampilan data arsip dimana admin memilih tab surat masuk ataupun tab surat keluar dapat dilakukan pengelolaan lebih lanjut seperti menambahkan arsip, menghapus, melihat dan memperbaharui arsip. Setelah *admin/pegawai* Dinas Pertanian dan Perikanan Kota Salatiga meng-*input*-kan data keperluan arsip yang akan dikelola lalu akan tersimpan dalam *database* dan dilanjutkan dengan sistem memberikan *alert* bahwa pengelolaan lebih lanjut pada arsip telah berhasil dilakukan. Alur pengelolaan data arsip surat masuk dan surat keluar tergambar seperti pada Gambar 6.



**Gambar 6** Activity Diagram Administrator Kelola Data Arsip

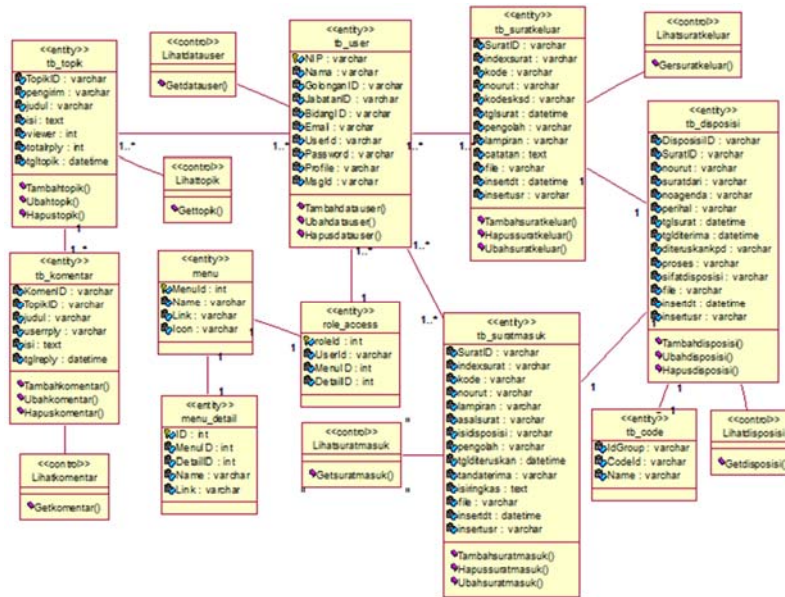
Gambar 7 adalah *deployment diagram*, merupakan desain pemetaan dari bagian-bagian aplikasi yang akan dibangun, baik perangkat lunak maupun perangkat keras yang digunakan supaya aplikasi dapat *running* secara baik.



**Gambar 7** Deployment Diagram Aplikasi

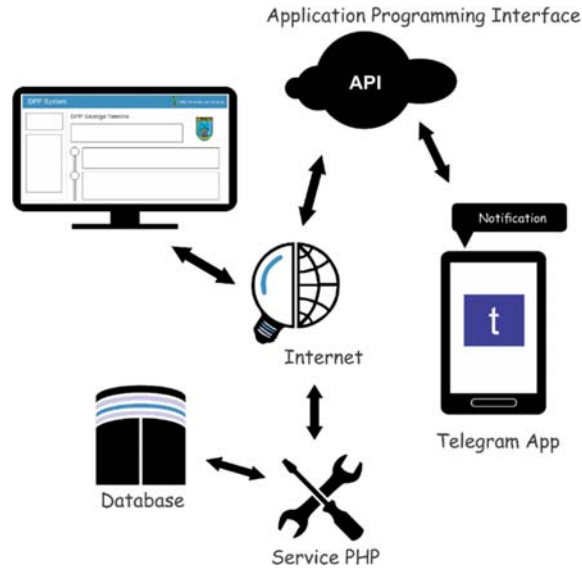


Sedangkan *class diagram* pada Gambar 8, merupakan desain yang menjelaskan interaksi antar entitas-entitas yang ada pada aplikasi, khususnya dalam basis data. Entitas yang berupa *class* dapat mewakili tabel yang beserta atribut yang menjelaskan tipe data tiap komponen-komponennya, serta beisi *method* yang dibutuhkan untuk mengakses atribut pada aplikasi yang dibangun.



Gambar 8 Class Diagram

Pada Gambar 9 menggambarkan arsitektur dari sistem aplikasi yang dibangun. Sistem plikasi tersebut terdiri dari aplikasi *website* yang nantinya dapat digunakan oleh para pegawai Dinas Pertanian dan Perikanan Kota Salatiga. Sistem aplikasi tersebut harus terhubung dengan koneksi internet untuk dapat diakses melalui *website*. Internet akan dipergunakan untuk menjembatani aplikasi *wbsite* untuk mengirim data ke basis data yang disediakan oleh *server* melalui *service PHP* dan *Application Programming Interface (API)* dari aplikasi Telegram dalam *platform mobile android* sehingga semua pegawai Dinas dapat mengetahui tentang adanya surat yang masuk dan mengetahui adanya forum yang dibuat untuk segera didiskusikan dengan cara yang nantinya akan diberikan notifikasi ketika ada salah seorang pegawai yang melakukan pencatatan surat masuk atau pembuatan forum diskusi pada sistem atau aplikasi.



**Gambar 9** Arsitektur Sistem

#### 4. Hasil dan Pembahasan

Penelitian ini menghasilkan satu aplikasi yaitu aplikasi *website*. Aplikasi *website* ini ditujukan kepada pegawai Dinas Pertanian Kota Salatiga. Pegawai yg sebelumnya masih memakai sistem manual pada pengelolaan surat masuk dan keluar serta tidak memiliki *pemberitahuan* tentang surat yang harus segera diproses untuk mengingatkan kapan waktunya tiba melakukan tindak lanjut pada surat yang masuk dan juga Kepala Dinas dapat melihat kinerja dari semua pegawai yang bekerja pada Dinas Pertanian dan Perikanan Kota Salatiga. Pada aplikasi ini saat pegawai membuka *website* akan ditujukan halaman login agar dapat mengetahui siapa yang melakukan pekerjaan dengan baik atau tidak. Gambar 10 menunjukkan *form* login digunakan sebagai keamanan dasar pada sistem dari penyalahgunaan hak akses, sehingga keamanan data dapat terjamin. Pada proses login user diminta untuk memasukkan *username* dan *password* untuk dapat mengakses data selanjutnya.



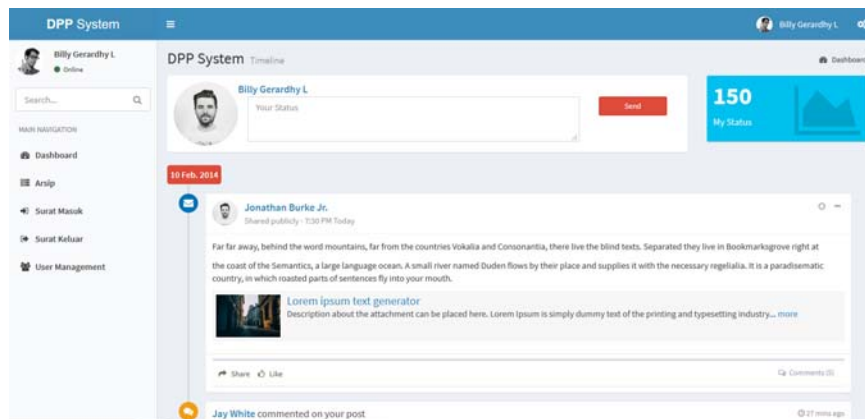
**Gambar 10** Tampilan Form Login

### Kode Program 1. Proses Login Sistem

```
01. function login(){  
02. $user = $this->input->post('user');  
03. $pass = $this->input->post('pass');  
04. if ($this->DataModel->ceklogin($user,$pass)) {  
05. redirect('Admin/index','refresh');}else{  
06. redirect('MainController/index','refresh');}}
```

Kode program 1 merupakan struktur *source code*/pemrograman yang dilakukan untuk melakukan login pada sistem pengelolaan surat. Ketika kode program dari baris 2 dan 3 di eksekusi maka akan mengambil data input dari *form* yang ada pada halaman login. Setelah proses 2 dan 3 terpenuhi maka akan diteruskan pada baris 4 yang melakukan eksekusi pada *function* cek login yang berada pada data model dengan melemparkan *value* berupa *username* dan *password*. Proses terakhir pada login sistem dilakukan pada proses baris 5 dan 6 yang melakukan pemanggilan *page index* bila dalam pengecekan *username* dan *password* berhasil.

Gambar 11 merupakan tampilan utama dari sistem Dinas Pertanian Kota Salatiga yang isinya dashboard, arsip, surat masuk, surat keluar dan user *management*. Pengguna sistem aplikasi tinggal klik saja icon menu yang diinginkan untuk melihat apa saja yang ada di dalam file tersebut.



Gambar 11 Tampilan Dashboard Sistem

**Gambar 12** Tampilan Input Surat Masuk

Pada Gambar 12 merupakan menu surat masuk yang merupakan halaman dimana user dapat menginput data surat masuk. Dimana didalamnya berisikan index surat, kode surat, nomor urut surat, lampiran, asal surat, isi disposisi surat, pengolah, tanggal diteruskan surat, tanda terima surat, isi rangkas dan *attachment* surat dan setelah selesai penginputan klik tombol *save* dan akan diteruskan pada pengisian disposisi surat masuk.

Pada Gambar 13 merupakan Menu input disposisi surat masuk yang merupakan halaman dimana user dapat menginput data disposisi dari surat masuk. Dimana didalamnya berisikan nomor surat, surat dari, no. agenda, tanggal surat, diterima tanggal, perihal, diteruskan kepada, dengan hormat harap, sifat disposisi dan *attachment* surat, setelah selesai penginputan klik tombol *save* maka data akan tersimpan pada *database* lebih jelasnya seperti tampilan dibawah ini.

**Gambar 13** Tampilan Input Disposisi Surat Masuk

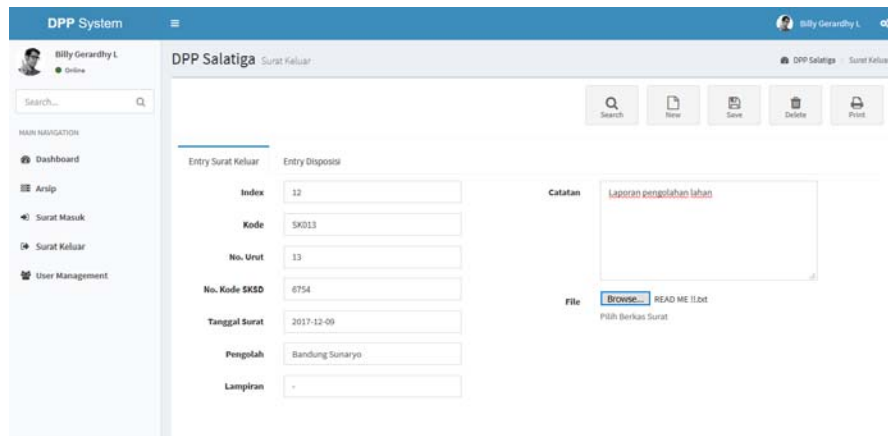
### Kode Program 2. Proses Simpan Surat Masuk dan Disposisi

```
01. function savedata( $index = $this->input->post('index');$kode = $this->input->post('kode');
02. $nourut = $this->input->post('nourut');
03. $lampiran = $this->input->post('lampiran');
```

```
04. $asal = $this->input->post('asal'); ...){
05. $surat = array("SuratID" => $SuratID , "indexsurat" =>
06. $index , "kode" => $kode , "nourut" => $nourut ,
07. "lampiran" => $lampiran , "asalsurat" => $asal ,
08. "isidisposisi" => $isidisposisi , ...)
09. $this->SMModel->insertsurat($surat);
10. $this->SMModel->insertdispo($disposisi);
11. $this->notif($SuratID);}
```

Kode program 2 merupakan struktur *source code* / pemrograman yang dilakukan untuk melakukan proses penginputan surat masuk dan disposisi surat masuk pada sistem pengelolaan surat. Ketika kode program dari baris 2 - 9 di eksekusi maka akan mengambil data input dari *form* yang ada pada halaman surat masuk dan juga disposisi surat. Setelah proses 2 - 9 terpenuhi maka akan diteruskan pada baris 10 yang melakukan eksekusi pada *function* insertsurat yang berada pada data model dengan melemparkan *value* berupa *data array* surat. Pada baris 11 melakukan eksekusi pada *function* insertdispo yang berada pada data model dengan melemparkan *value* berupa *data array* disposisi. Proses terakhir pada proses *insert* data surat masuk dan disposisi surat masuk terdapat pada baris 12 yang melakukan pemanggilan pada *function* notif, dimana *function* ini melakukan pengiriman sebuah notifikasi kepada pegawai melalui aplikasi *telegram*.

Pada Gambar 14 merupakan tampilan menu surat keluar yang merupakan halaman dimana user dapat menginput data surat yang akan keluar. Dimana didalamnya berisikan index surat, kode surat, nomor urut surat, no. kode SKSD, tanggal surat, pengolah, lampiran dan *attachment* surat dan setelah selesai penginputan klik tombol *save* dan akan diteruskan pada pengisian disposisi surat keluar.



Gambar 14 Tampilan Input Surat Keluar

### Kode Program 3. Proses Simpan Surat Keluar

```
01. function tbSave(x){if (x == 1) {$( "#upload1"
02. ).submit();var file1 =$("#file1").val().replace(
03. /C:\\\\fakepath\\\\i, ' ');var index = $('#index').val();
04. var kode = $('#kode').val();var kodesksd =
05. $('#kodesksd').val();var nourut = $('#nourut').val();
06. $.post('<?php echo
```

```

07. site_url('SK/savedata') ?>',{file1:file1, index:index,
08. kode:kode,kodesksd:kodesksd,...},
09. function(data){alert(data);});}else if (x == 2) {
10. no_event();}else{
11. no_event();}
12. }

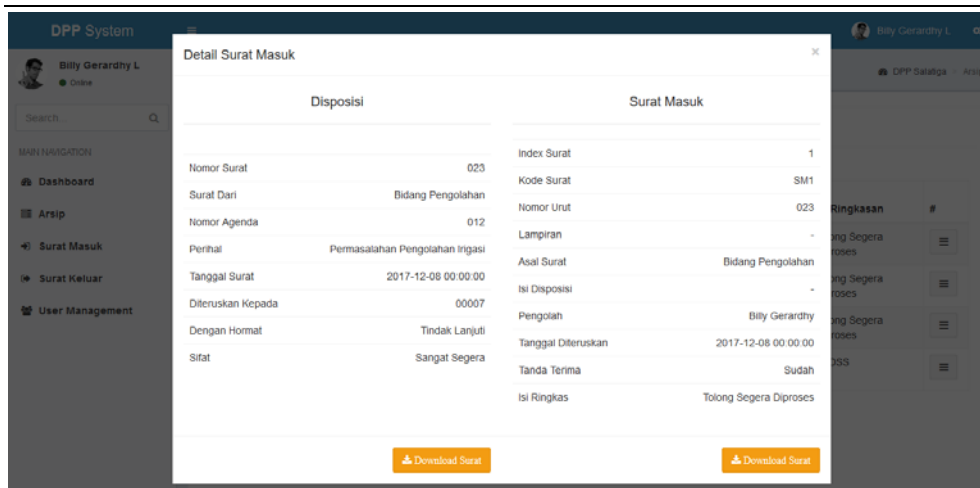
```

Kode program 3 merupakan struktur *source code* / pemrograman yang dilakukan untuk melakukan proses penginputan surat keluar pada sistem pengelolaan surat. Ketika kode program dari baris 2 - 5 di eksekusi maka akan mengambil data input dari *form* yang ada pada halaman surat keluar. Setelah proses 2 - 5 terpenuhi maka akan diteruskan pada baris 6 - 8 yang melakukan eksekusi pada *function* *insertsurat* yang berada pada data model dengan melemparkan *value* berupa *data array* surat. Proses terakhir pada proses *insert* data surat keluar terdapat pada baris 9 yang melakukan pemanggilan pada dimana *function* ini memberikan sebuah *alert* bahwa dan menampilkan *pop-up* detail surat kepada pegawai.

Pada Gambar 15 dan 16 merupakan tampilan menu arsip yang merupakan halaman dimana user dapat melihat data surat masuk yang telah diinputkan oleh user. Dimana didalamnya berisikan data dari surat yang telah selesai dilakukan penginputan. Pada sebelah kolom isi ringkasan terdapat sebuah tombol yang akan menampilkan detail dari surat masuk.

Index	Kode	No Urut	Lampiran	Asal Surat	Isi Disposisi	Pengolah	Tanggal Diteruskan	Tanda Terima	Isi Ringkasan	#
1	SM1	023	-	Bidang Pengolahan	-	Billy Gerardhy	2017-12-08 00:00:00	Sudah	Tolong Segera Diproses	
1	SM1	023	-	Bidang Pengolahan	-	Billy Gerardhy	2017-12-08 00:00:00	Sudah	Tolong Segera Diproses	
1	SM1	023	-	Bidang Pengolahan	-	Billy Gerardhy	2017-12-08 00:00:00	Sudah	Tolong Segera Diproses	
235	2365	23325	-	Sini	Disposisi	ASD	2017-11-30 00:00:00	Sudah	SADSS	

**Gambar 15** Tampilan Arsip Surat Masuk



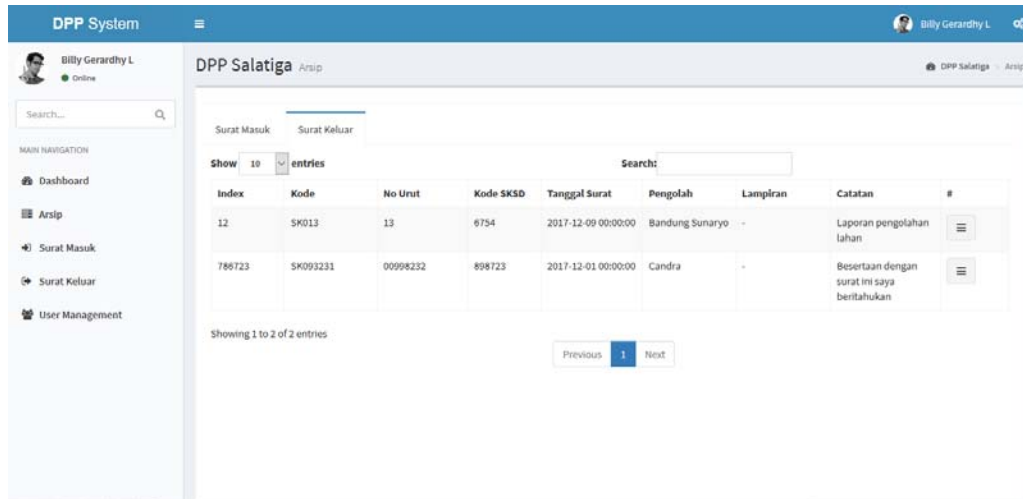
Gambar 16 Tampilan Detail Arsip Surat Masuk

#### Kode Program 4. Proses Detail Surat Masuk dan Download File

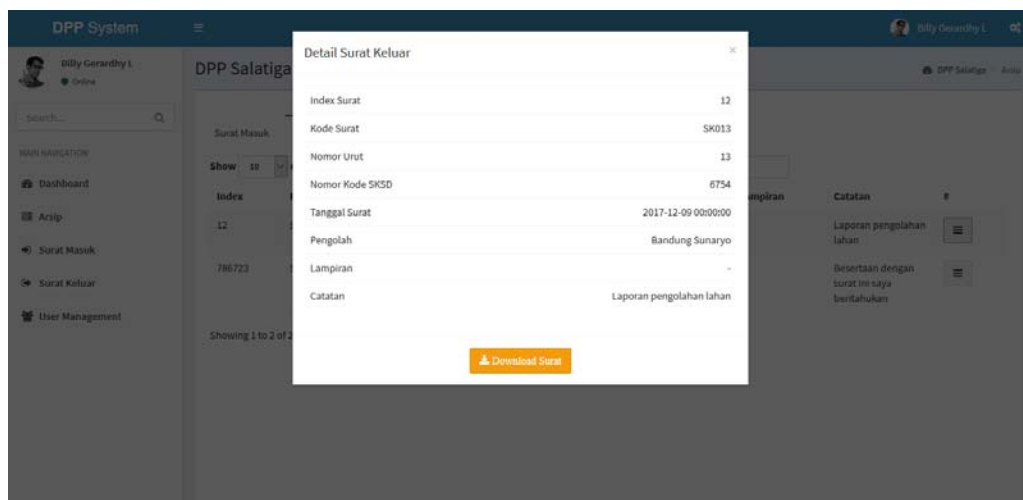
```
01. function showsm(id){$.ajax({type: "GET",url: "<?php
02. echo site_url('Arsip/getdetsm') ?>",dataType: "json",
03. data: {SuratID:id},success: function(data) {
04. $('#nosurat').html(data.nourut);$('#srdari').html(data
05. .suratdari);$('#noagenda').html(data.noagenda);$('#per
06. ihal').html(data.perihal);$('#tgl surat').html(data.tgl
07. surat);$('#diteruskan').html(data.diteruskankpd);$('#d
08. ghormat').html(data.proses);...
09. $('#ddis').attr("href", "<?php echo
10. base_url().\"user_upload/Disposisi/\" ?>"+data.file)
11. $('#dsm').attr("href", "<?php echo
12. base_url().\"user_upload/Surat/\" ?>"+data.file)},
13. error: function(response) {console.log(response);}
14. });$('#smModal').modal('show');}
```

Kode program 4 merupakan struktur *source code* / pemrograman yang dilakukan untuk melakukan proses pemanggilan data surat masuk dan disposisi surat masuk pada sistem pengelolaan surat. Ketika kode program dari baris 1 - 3 di eksekusi maka akan mengambil data input dari *database* dengan melakukan pemanggilan *function getdetsm* untuk surat masuk dan juga disposisi surat. Setelah proses 1 - 3 terpenuhi maka akan diteruskan pada baris 4 - 8 yang melakukan penampungan data dari *database* surat masuk dan disposisi surat. Pada baris 9 dan 11 melakukan eksekusi untuk melakukan proses *download* file surat masuk dan file disposisi surat. Proses terakhir pada proses lihat detail data surat masuk dan disposisi surat masuk terdapat pada baris 13 dan 14 yang melakukan pemanggilan pada *function respon*, dimana *function* ini memberikan sebuah *alert* bahwa dan menampilkan *pop-up* detail surat kepada pegawai.

Pada Gambar 17 dan 18 merupakan tampilan menu arsip surat keluar yang merupakan halaman dimana user dapat melihat data surat keluar yang telah diinputkan oleh user. Dimana didalamnya berisikan data dari surat yang telah selesai dilakukan penginputan. Pada sebelah kolom isi ringkasan terdapat sebuah tombol yang akan menampilkan detail dari surat keluar.



Gambar 17 Tampilan Arsip Surat Keluar



Gambar 18 Tampilan Detail Arsip Surat Keluar

### Kode Program 5. Proses Detail Surat Keluar dan Download File

```

01. function showsk(id){$.ajax({type: "GET", url: "<?php
02. echo site_url('Arsip/getdetsk') ?>",dataType: "json",
03. data: {SuratID:id},success: function(data) {
04. $('#indexsuratk').html(data.indexsurat);$('#kodesuratk
05. ').html(data.kode);$('#nourutk').html(data.nourut);
06. $('#nosksdk').html(data.kodesksd);$('#tglsuratk').html
07. (data.tglsurat);$('#pengolahk').html(data.pengolah);$(
08. '#lampirank').html(data.lampiran);$('#catatank').html(
09. data.catatan);
10. $("#dsk").attr("href", "<?php echo
11. base_url().<user_upload/Surat Keluar/" ?>"+data.file)
12. },error: function(response) {console.log(response);}
13. });$('#skModal').modal('show');}

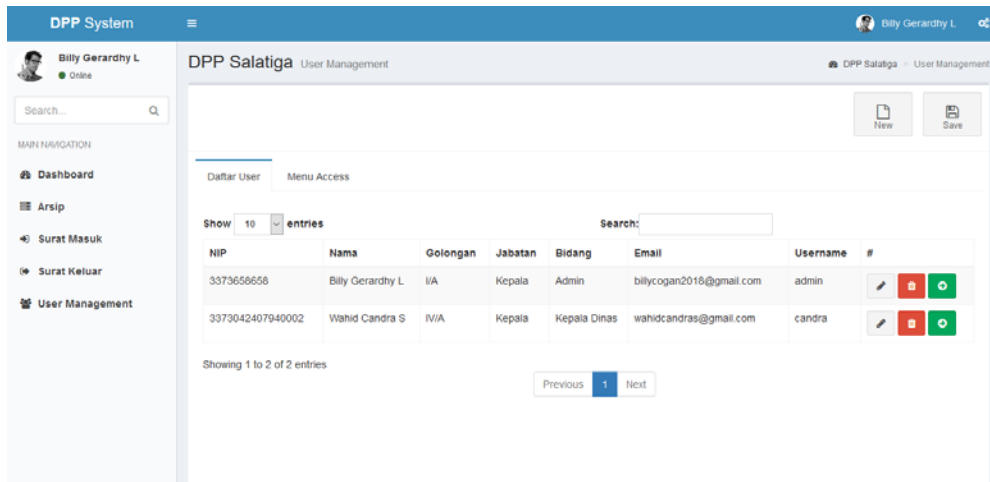
```

Kode program 5 merupakan struktur *source code* / pemrograman yang dilakukan untuk melakukan proses pemanggilan data surat masuk dan disposisi surat masuk pada sistem pengelolaan surat. Ketika kode program dari baris 1 - 3 di eksekusi maka akan mengambil data input dari *database* dengan melakukan

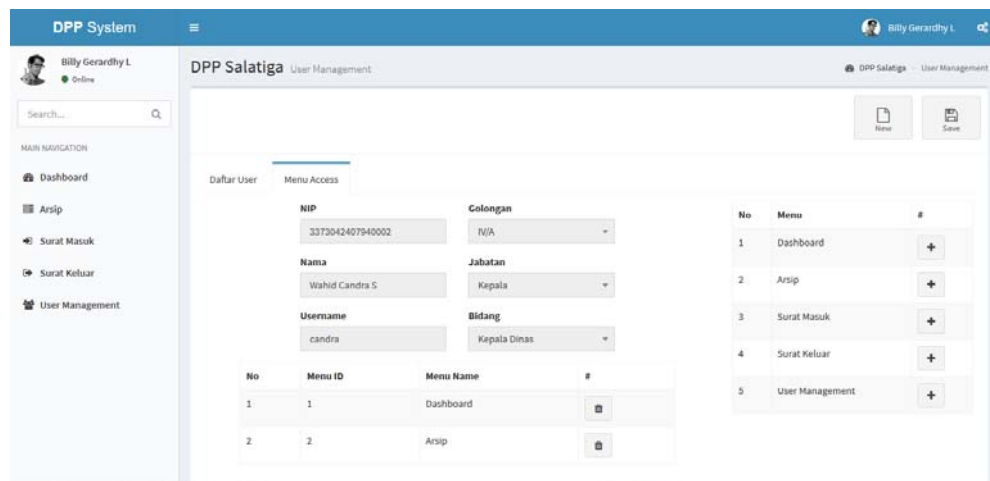


pemanggilan *function getdetsk* untuk surat keluar dan juga disposisi surat. Setelah proses 1 - 3 terpenuhi maka akan diteruskan pada baris 4 - 9 yang melakukan penampungan data dari *database* surat keluar dan disposisi surat. Pada baris 10 melakukan eksekusi untuk melakukan proses *download* file surat masuk dan file disposisi surat. Proses terakhir pada proses lihat detail data surat masuk dan disposisi surat masuk terdapat pada baris 12 dan 13 yang melakukan pemanggilan pada *function respon*, dimana *function* ini memberikan sebuah *alert* bahwa dan menampilkan *pop-up* detail surat kepada pegawai.

Pada Gambar 19 dan 20 merupakan tampilan menu *user management* yang merupakan halaman dimana user dapat melakukan manajemen terhadap user yang akan memakai sistem aplikasi pengolahan surat. Dimana didalamnya berisikan data dari pegawai dan juga hak akses dari pegawai tersebut. Pada sebelah kolom *username* terdapat beberapa tombol yang akan digunakan untuk melakukan pengolahan terhadap data user.

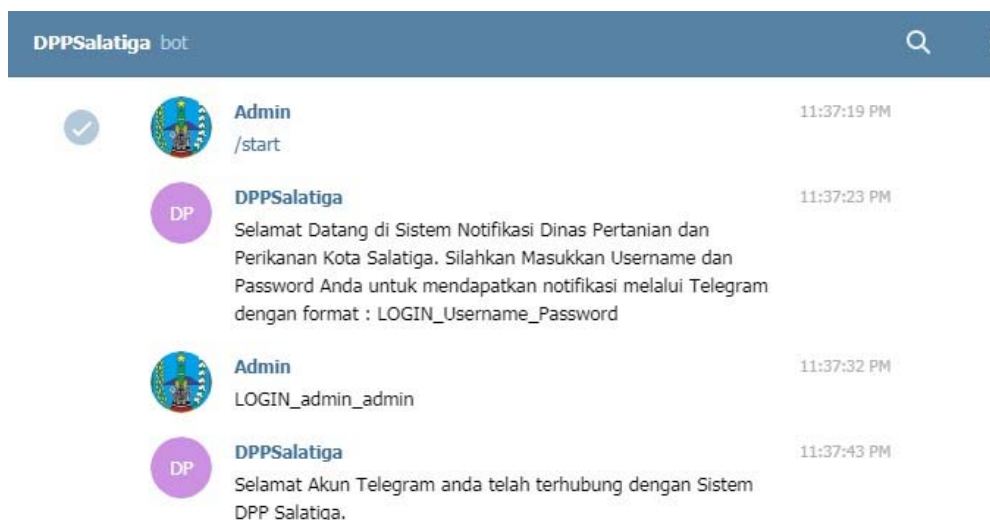


Gambar 19 Tampilan User Management



Gambar 20 Tampilan User Management

Pada aplikasi telegram yang terdapat pada *smartphone*, pegawai diwajibkan melakukan login pada DPPSalatiga *Bot system*, agar dapat mendapatkan notifikasi *realtime* dari proses pengelolaan surat. Gambar 21 menunjukkan *form* login digunakan sebagai keamanan dasar pada sistem dari penyalahgunaan hak akses, sehingga keamanan data dapat terjamin. Pada proses login user diminta untuk memasukkan *username* dan *password* untuk dapat mengakses data selanjutnya.



**Gambar 21** Tampilan DPPSalatiga *Bot system*

#### Kode Program 6. Proses Login pada Telegram App

```

01. function reply(){$data = $this->TelegramModel-
02. >getMessage();foreach ($data as $row){if ($row->text
03. == '/start') {$i = $this->pushMessage($row-
04. >message_id, 'Selamat Datang di Sistem Notifikasi
05. Dinas Pertanian dan Perikanan Kota Salatiga...
06. );if ($i == 1) {$this->TelegramModel-
07. >updateMessage($row->update_id);}
08. }else{$x = explode('_', $row->text);if
09. (strtoupper($x[0]) == 'LOGIN') {if ($this-
10. >TelegramModel->cekUser($x[1],$x[2]) > 0) {$data =
11. array('MsgId' => $row->message_id);$this
12. >TelegramModel->clearmsgid($row->message_id);if($this-
13. >TelegramModel->updateMsgId($data,$x[1]) {$i = $this-
14. >pushMessage($row->message_id, 'Selamat Akun Telegram
15. anda telah terhubung dengan Sistem DPP Salatiga.')}if
16. ($i == 1) {$this->TelegramModel->updateMessage($row-
17. >update_id);}}else{$i = $this->pushMessage($row-
18. >message_id, 'User telah terdaftar.')} if ($i == 1)
19. {$this->TelegramModel->updateMessage($row->update_id);
20. }}

```

Kode program 6 merupakan struktur *source code* / pemrograman yang dilakukan untuk melakukan proses login pada sistem pengelolaan surat yang ada pada aplikasi telegram. Ketika kode program dari baris 1 - 7 di eksekusi maka sistem akan melakukan perulangan dan memulai proses *start* dari layanan yang diberikan oleh API dari telegram dan menampilkan ucapan selamat datang dan permintaan melakukan login kepada pegawai. Setelah proses 1 - 7 terpenuhi maka

akan diteruskan pada baris 8 - 17 yang melakukan pengecekan terhadap *username* dan *password* yang diinputkan oleh pegawai dari *database* user. Pada baris 18 – 20 mengeksekusi bila ternyata user benar sudah terdaftar pada DPPSalatiga Bot System.

Pada Gambar 22 merupakan tampilan aplikasi telegram yang terdapat pada *smartphone*, pegawai diwajibkan mendapatkan sebuah notifikasi secara *realtime* pada DPPSalatiga Bot system bila terdapat surat masuk dan disposisi surat setelah pegawai selesai melakukan penginputan., agar pegawai yang ditujukan dapat segera melakukan proses pada surat masuk tersebut.



**Gambar 22** Tampilan DPPSalatiga Bot system

### Kode Program 7. Proses Detail Notifikasi Telegram App

```
01. public function pushMessage($chat_id, $text){$key =
02. "470154184:AAFMDQnDhUNChYOGx4MSvnj4fqkkHei3-bg";$host
03. = "https://api.telegram.org/bot".$key.'/' ;
04. $send =file_get_contents($host."sendmessage?chat_id="
05. ".$chat_id."&&text=".$text);
06. $result = json_decode($send);return $result->ok;}
07. function notif($suratID,$diteruskan){$surat = $this-
08. >SMMModel->getsurat($suratID);$msgid = $this-
09. >TelegramModel->getmsgid($diteruskan);foreach ($msgid
10. as $row) {echo $row->MsgId;$message = "SURAT
11. MASUK\n\nIndex : ".$surat->indexsurat."\nKode :
12. ".$surat->kode."\nNomor Urut : ".$surat-
13. >nourut."\nLampiran : ".$surat->lampiran."\nSurat Dari
14. : ".$surat->asalsurat."\nDisposisi : ".$surat-
15. >isidisposisi."\nPengolah : ".$surat-
16. >pengolah."\nTanggal Diteruskan : ".date('d-m-
17. Y',strtotime($surat->tgl diteruskan))."\nTanda Terima :
18. ".$surat->tandaterima."\nIsi Ringkas : ".$surat-
19. >isiringkas."\n";$this->pushMessage($row-
20. >MsgId,urlencode($message));}}
```

Kode program 7 merupakan struktur *source code* / pemrograman untuk menampilkan isi dari notifikasi yang diterima oleh pegawai yang ditujukan bila telah dilakukan penginputan surat masuk dan disposisi surat pada sistem pengelolaan surat yang ada pada aplikasi telegram. Kode program dari baris 1 - 6 di eksekusi maka sistem akan melakukan pengecekan terhadap key dari layanan yang diberikan oleh API dari telegram agar dapat mendapatkan notifikasi pada aplikasi telegram. Setelah proses 1 - 6 terpenuhi maka akan diteruskan pada baris 7 - 20 yang melakukan pemanggilan *function* getsurat dan melakukan penampungan data surat yang akan ditampilkan pada aplikasi telegram DPPSalatiga Bot System.

Pengujian aplikasi adalah tahapan pada penelitian yang bertujuan untuk mendapatkan bukti bahwa aplikasi yang dihasilkan telah sesuai yang diharapkan dan memenuhi kebutuhan dari pengguna aplikasi. Pengujian aplikasi menggunakan metode *blackbox* yaitu dilakukan dengan menguji seluruh fungsi dan fitur yang dimiliki aplikasi. Pengujian ini dilakukan guna memastikan bahwa sudah tidak ada kesalahan/*bug* pada sistem dan segala modul proses dan fungsi sistem yang dibuat telah berjalan sesuai rancangan. Tabel 1 merupakan hasil pengujian *blackbox*.

**Tabel 1** Hasil Pengujian *Blackbox*

Fungsi yang diuji	Kondisi	Output yang diharapkan	Output yang dihasilkan sistem	Status Pengujian
<i>Login</i>	Mengisi <i>form</i> dan <i>Password</i> dengan benar	Berhasil <i>Login</i>	Sukses <i>Login</i>	<i>Valid</i>
<i>Input data surat masuk</i>	<i>Form</i> diisi dengan benar	Sukses data <i>insert</i>	Sukses data <i>insert</i>	<i>Valid</i>
<i>Input data surat keluar</i>	<i>Form</i> diisi dengan benar	Sukses data <i>insert</i>	Sukses data <i>insert</i>	<i>Valid</i>
	<i>Form</i> diisi beberapa atau kosong	Gagal tambah data	Gagal tambah data	<i>Valid</i>
<i>Managemen akses user</i>	Pemberian hak akses <i>page</i> sistem pengelolaan surat	Sukses memeberikan hak akses	Sukses memberikan hak akses	<i>Valid</i>
Notifikasi API Telegram	Surat masuk dan Disposisi surat berhasil tersimpan pada <i>database</i>	Sukses kirim pesan dan tampilkan notifikasi	Sukses kirim pesan dan tampilkan notifikasi	<i>Valid</i>
	Surat masuk dan Disposisi surat tanpa diteruskan kepada berhasil tersimpan pada <i>database</i>	Gagal kirim pesan dan notifikasi	Gagal kirim pesan dan notifikasi	<i>Valid</i>

Pengujian lain yang dilakukan pada aplikasi yaitu *acceptance test* yang dilakukan oleh pengguna di Dinas Peranian Kota Salatiga. Pada pengujian ini menggunakan media kuisisioner guna mengetahui dampak langsung dari penggunaan aplikasi terhadap proses pengelolaan surat dan efektivitas kerja, dan hasilnya dapat terlihat pada Tabel 2.

**Tabel 2** Perbedaan Proses Bisnis via *Website*

No	SOP Lama	SOP Baru	Perbedaan
1.	Pegawai tidak memiliki notifikasi adanya surat masuk	Diberikan notifikasi pada aplikasi Telegram bahwa adanya surat yang masuk	Tepat waktu dalam melakukan penindakan terhadap surat yang masuk, sehingga pekerjaan menjadi lebih efisien dan optimal.
2.	Pengelolaan masih bersifat manual	Pengelolaan sudah menggunakan sebuah sistem aplikasi	Memudahkan pekerjaan para pegawai karena dengan menggunakan sistem aplikasi pengelolaan
3.	Penyimpanan Arsip masih secara manual	Penyimpanan semua arsip dan kegiatan pegawai tersimpan pada sebuah <i>database</i>	Mudahnya pencarian arsip ketika dibutuhkan kembali, dan meminimalisir hilangnya arsip
4.	Kepala Dinas mengecek satu persatu pekerjaan para pegawai setiap ada keterlambatan proses surat	Dapat langsung memonitoring pekerjaan pegawai menggunakan aplikasi pengelolaan	Memberikan informasi lebih terpercaya kepada Kepala Dinas tentang kinerja para pegawainya jika terjadi keterlambatan proses surat

## 5. Simpulan

Sistem notifikasi pesan disposisi pada pengelolaan surat menggunakan layana *API media social Telegram* pada Dinas Pertanian Kota Salatiga dibangun berbasis *web* telah dapat berfungsi untuk mengirimkan notifikasi ketika adanya surat masuk yang selesai disimpan dan ditujukan kepada pegawai yang ditentukan untuk segera dapat ditindaklanjuti. Dengan menggunakan Sistem Notifikasi Pesan Disposisi Pada Pengelolaan Surat Menggunakan Layana *API Media Social Telegram* Pada Dinas Pertanian Kota Salatiga sangat membantu dalam menindaklanjuti Surat Dinas agar dapat segera ditindaklanjuti oleh para pegawai yang menerima notifikasi dari aplikasi. Saran pengembangan penelitian selanjutnya adalah pengembangan layanan notifikasi menggunakan layanan *API media social* yang lebih lengkap atau lebih dari satu layanan notifikasi, dan yang lebih populer.

**Daftar Pustaka**

- [1] Maulana, Rizki., 2015. “Pengembangan Sistem Informasi Pengelolaan Surat Masuk dan Surat Keluar Berbasis Web di Dinas Sosial Tenaga Kerja dan Transmigrasi Kabupaten Garut Menggunakan *framework* PHP”. Garut: Sekolah Tinggi Teknologi Garut.
- [2] Fitrianda, Haris., 2015. “Sistem Informasi Pengelolaan Surat Masuk Dan Surat Keluar Paa KPP XYZ Palembang”. Palembang: STMIK GI MDP Palembang.
- [3] Hasan, L.O.M, Sholeh, M., Iswahyudi, C., Pemanfaatan Twitter API Untuk Mengakses Jadwal Bimbingan Dosen Pada IST AKPRIND Yogyakarta, Jurnal SCRIPT: Vol. 3, No. 2. 2016. ISSN:2338-6313: hal 173-181.
- [4] Indrapermana, Yusa, 2017. “Pengertian *Application Programming Interface* (API)”. <http://www.yusaindera.com/2017/03/pengertian-application-programming.html>. Diakses pada tanggal 21 November 2017.
- [5] Nesaba Media. “Kenali Pengertian *Website* Beserta Manfaat dan Jenis-Jenis *Website*”. <http://www.nesabamedia.com/pengertian-website/>. Diakses pada tanggal 21 November 2017.
- [6] Telegram (Aplikasi), diperoleh dari [https://id.wikipedia.org/wiki/Telegram\\_\(aplikasi\)/](https://id.wikipedia.org/wiki/Telegram_(aplikasi)). Diakses pada tanggal 21 November 2017.
- [7] Pressman, R.S. 2001. Software Engineering : A Practitioner's Approach. Amerika Serikat: R.S. Pressman and Associates
- [8] Sulistyorini, Prastuti. “Pemodelan Visual dengan Menggunakan UML dan *Rational Rose*”. STMIK Widya Pratama : Pekalongan.