

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 *Document Management System (DMS)***

*Document Management System (DMS)* merupakan sebuah sistem komputer (atau seperangkat program komputer) yang digunakan untuk menelusuri dan menyimpan dokumen elektronik dan gambar pada dokumen. DMS bermanfaat dalam mengefektifkan dan mengefisienkan proses bisnis atau organisasi. Manfaat yang utama adalah pengguna dapat menemukan informasi yang dibutuhkan dengan cepat, sehingga dapat membantu proses menjadi lebih cepat, baik dan murah. Sistem manajemen dokumen adalah sistem untuk menyederhanakan siklus pengelolaan dokumen bisnis atau organisasi. Sebuah sistem manajemen file guna mendukung otomatisasi dan konsistensi organisasi dan seluruh dokumen kebijakan (Laserfiche, 2007). Beberapa manfaat DMS sebagai berikut :

1. Meningkatkan produktivitas proses bisnis DMS dapat membantu penambahan produktivitas pekerja dimana pekerja dapat mengakses dan beraksi dengan informasi yang lebih cepat dan tepat.
2. Meningkatnya *response time* proses bisnis Pencarian file dokumen, *update* dokumen, dan pendistribusian dokumen digital dapat dilakukan jauh lebih cepat dengan DMS.
3. Mengurangi total biaya dokumen dan meningkatkan efisiensi ruang penyimpanan DMS dapat mengurangi biaya-biaya, karena dimensi penyimpanan file digital yang jauh lebih kecil dan lebih murah daripada

sistem penyimpanan file konvensional. DMS Secara dramatis menurunkan ukuran ruang penyimpanan dokumen, menggantikan gudang dokumen dengan media penyimpanan elektronis yang jauh lebih kecil seperti hard disk, disket dan CD.

4. Menurunkan biaya tambahan Dengan menerapkan DMS, maka biaya-biaya *overhead* untuk penyimpanan dokumen konvensional seperti : kertas, foto-copy, *filing cabinet* dapat ditekan sampai dengan nol persen.
5. Mengurangi resiko kehilangan ataupun kerusakan dokumen Dengan menyimpan dokumen secara digital, gangguan rayap jelas akan dinihilkan 100%. Gangguan yang lain seperti kebakaran dapat diminimalisasi dengan sistem penyimpanan yang lebih canggih daripada sekedar dokumen kertas konvensional.
6. *Document Sharing* Melalui DMS, pemakaian dokumen dapat dilakukan secara bersamaan oleh beberapa *user* sekaligus.

## 2.2 Arsip Elektronik (*E-Archive*)

Arsip elektronik menurut International Council of Archives (ICA), adalah : “*Electronic record is a record that is suitable for manipulation, transmission or processing by a digital computer*”, (arsip elektronik ialah arsip yang dapat dimanipulasi, ditransmisikan, atau diproses oleh komputer digital).

ARMA Standards Program: *Glossary of Records Management Terms*, 1984, mendefinisikan arsip elektronik sebagai “*Machine-Readable Record: Coded information which to be understood, must be translated by a computer*”, (Arsip yang terbacakan oleh mesin: Kode Informasi yang dimana untuk memahaminya

harus diterjemahkan dengan komputer). Menurut NARA (*National Archives and Record Administration*) adalah arsip yang disimpan dan diolah dalam suatu format dimana hanya komputer yang dapat memprosesnya.

Sedangkan menurut *Standards of Electronic Records Filing and Management China* adalah arsip elektronik sebagai dokumen yang diciptakan oleh perangkat digital, dalam lingkup digital dan format digital, membutuhkan perangkat digital seperti komputer untuk membaca dan memproses, dan dapat didistribusikan melalui jaringan komunikasi. Jadi arsip elektronik (*e-archive*) yaitu merupakan dokumen, catatan, dan surat-surat yang dibuat atau disimpan dalam bentuk digital.

### **2.2.1 Arsip**

Arsip menurut kamus besar bahasa indonesia /ar-sip/ dokumen tertulis (surat, akta, dan sebagainya), lisan (pidato, ceramah, dan sebagainya), atau bergambar (foto, film, dan sebagainya) dari waktu yang lampau, disimpan dalam media tulis (kertas), elektronik (pita kaset, pita video, disket komputer, dan sebagainya), biasanya dikeluarkan oleh instansi resmi, disimpan dan dipelihara di tempat khusus untuk referensi.

### **2.3 Hypertext Preprocessing (PHP)**

*PHP (Personal Home Page)* dibuat pertama kali pada musim gugur tahun 1994 oleh Rasmus Lerdoff, awalnya digunakan pada websitenya untuk mencatat siapa saja yang berkunjung dan melihat biodatanya. Versi pertama yang diterbitkan tersedia pada awal tahun 1995, dikenal sebagai aplikasi *Personal Home Page* yang terdiri atas *engine parser* yang sangat sederhana yang hanya mengerti

beberapa makro khusus dan sejumlah utilitas yang sering digunakan pada halaman-halaman web, seperti buku tamu, *counter* pengunjung dan lainnya.

Parser diprogram ulang pada pertengahan tahun 1995 dan diberi nama PHP/FI atau PHP2. FI (*Form Interpreter*) merupakan paket Rasmus lainnya yang ditulis untuk menginterpretasi data dari *form* HTML, yang kemudian dikombinasikan dengan *tool Personal Home Page* dan ditambahkan dukungan untuk *database* mSQL (*miniSQL*).

Pada pertengahan tahun 1997 terjadi perubahan pengembangan PHP. Pengembangan dilakukan oleh tim yang terorganisasi bukan oleh Rasmus sendiri saja. *Parser* dikembangkan oleh Zeev Suraski dan Andi Gutmans yang kemudian menjadi dasar untuk versi 3, dan banyak utilitas tambahan yang diprogram untuk menambah kemampuan dari versi 2. Versi terakhir (PHP 4) menggunakan *engine script* Zend untuk lebih meningkatkan kinerja (*performance*) dan mempunyai dukungan yang banyak berupa ekstensi dan fungsi dari berbagai *library* pihak ketiga (*third party*), dan berjalan seolah modul asli (*native*) dari berbagai server web yang populer.

PHP adalah suatu bahasa pemrograman *open source* yang digunakan secara luas terutama untuk pengembangan web dan dapat disimpan dalam bentuk html. Untuk menghasilkan sebuah HTML, *script* yang ditulis menggunakan PHP mempunyai perintah yang lebih singkat dibandingkan bahasa pemrograman lain seperti Perl atau C. Hanya perlu memasukan kode untuk melakukan sesuatu diantara tag awal dan tag akhir PHP.

Keuntungan utama menggunakan PHP adalah *script* PHP tidak hanya benar – benar sederhana bagi pemula, tetapi juga menyediakan banyak fitur tambahan untuk *programmer* profesional. Jangan ketakutan membaca daftar fitur PHP yang panjang. Anda dapat melewatinya dan mulai menulis *script* sederhana dalam beberapa jam.

*Script* PHP dapat digunakan dalam tiga hal yaitu :

1. Penulisan program *Server Side*. Hal ini adalah target utama PHP. Diperlukan tiga hal agar *script* PHP dapat bekerja antara lain, PHP Parser (CGI atau *server module*), *server web* (misal apache) dan browser web.
2. Penulisan program *Command Line*. *Script* PHP dapat berjalan tanpa server atau browser. Hanya diperlukan PHP parser dalam bentuk *Command Line*.
3. Penulisan program untuk aplikasi desktop.

Sebelum mulai membuat program PHP, diperlukan perangkat sebagai berikut :

1. Editor teks atau web *development tool* seperti Notepad, Sublime Text, PHPed, vi, emacs, Ultra edit, Dreamweaver, dan lainnya.
2. Untuk melihat dan melakukan pengujian pada suatu halaman, diperlukan browser web (IE, Chrome, Firefox, dan lainnya).
3. Server yang mendukung PHP.

Secara singkat, kelebihan PHP meliputi :

1. *Script* PHP sederhana, mudah dibuat, dan mempunyai kecepatan akses tinggi.

2. Dapat berjalan dalam server web yang berbeda dan dalam sistem operasi yang berbeda. PHP dapat berjalan pada sistem operasi Linux/Unix, Windows, dan Macintosh.
3. Bersifat *Open Source* sehingga diterbitkan secara gratis.
4. Dapat berjalan pada server web Microsoft Personal Web Server, Apache, IIS, Xitamid dan sebagainya.
5. Termasuk bahasa yang *embedded* (bisa ditempel atau diletakkan dalam tag HTML).

#### 2.4 Javascript

Javascript Menurut Budiyanto (2013:27) “Bahasa pemrograman java, perintah - perintah ditulis dengan kode yang disebut skrip. Java adalah bahasa pemrograman yang berorientasi objek, sedangkan script adalah serangkaian intruksi program”. Suatu bahasa pemrograman bisa dikembangkan untuk dapat menghasilkan berjalanya pada web browser adalah pengertian javascript (Sibero 2013:150).

Berdasarkan teori diatas, Javascript adalah bahasa pemrograman yang mempunyai kode script untuk intruksi program dan dikembangkan pada web browser. Javascript merupakan suatu bahasa *scripting* yang dikembangkan oleh Netscape. Dengan javascript ini pengguna dapat membuat suatu halaman web menjadi lebih interaktif dan lebih hidup. Sifat javascript antara lain diinterpretasikan oleh *client*, kode terintegrasi dalam dokumen HTML, kemampuannya terbatas pada fungsi browser, mampu mengakses objek dan fungsionalitas browser.

Aplikasi javascript yang umum adalah untuk *rollover image*, validasi form, menampilkan pesan pada status bar, membuat suatu *window* yang diatur propertinya, fungsi untuk menangani *mouse over* dan lain sebagainya.

Contoh program javascript :

```
<HTML>
<HEAD>
<TITLE> Contoh program javascript <TITLE>
<HEAD>
<BODY>
<SCRIPT language = "Javascript">
document . write ("Rifky Setiana")
</SCRIPT>
<BODY>
<HTML>
```

Program diatas akan menampilkan suatu *window* baru yang berisi satu baris teks "Rifky Setiana". Kata *document* pada program diatas adalah contoh objek yang diakses menggunakan metode *write* yang berfungsi untuk menampilkan teks pada browser.

## **2.5 Regular Expression (Regex)**

### **2.5.1 Regular Expression (Automata Theory)**

*Regular expression* pada *automata theory* merupakan suatu notasi yang dapat mendefinisikan semua *language* yang merupakan *regular language*. *Regular expression* dapat dijabarkan secara rekursif, dengan  $L(x)$  menandakan

language dari  $x$ .

**Basis :**

1.  $\varepsilon$  (*empty string*) dan  $\emptyset$  (*empty set*) merupakan *regular expression*, dengan  $L(\varepsilon) = \{\varepsilon\}$ , dan  $L(\emptyset) = \emptyset$ .
2. Jika  $a$  adalah *symbol*, maka  $a$  adalah *regular expression* dengan  $L(a) = \{a\}$ .

**Rekurens :**

1. Jika  $E$  dan  $F$  adalah *regular expression*, maka  $E + F$  adalah *regular expression* yang merupakan gabungan (*union*) dari  $E$  dan  $F$ . Sehingga,  $L(E + F) = L(E) \cup L(F)$ .
2. Jika  $E$  dan  $F$  adalah *regular expression*, maka  $EF$  adalah *regular expression* yang merupakan *concatenation* dari  $E$  dan  $F$ . Sehingga,  $L(EF) = L(E)L(F)$ .
3. Jika  $E$  adalah *regular expression*, maka  $E^*$  adalah *regular expression* yang merupakan *closure* dari  $E$ . Sehingga  $L(E^*) = L((E)^*)$ .
4. Jika  $E$  adalah *regular expression*, maka  $(E)$  juga merupakan *regular expression*. Sehingga,  $L((E)) = L(E)$ .

Tanda *asterisk* (\*) di atas menandakan *Kleene closure* dari sebuah *language*  $L$ . Definisi *Kleene closure* dari  $L$  ( $L^*$ ) adalah “*the set of those strings that can be formed by taking any number of strings from  $L$ , possibly with repetitions, and concatenating all of them*” (Hopcroft, 87). Sebagai

contoh, jika  $L = \{0, 1\}$ , maka  $L^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ .

Pada *regular expression*, terdapat *operator precedence*. Tanda \* mempunyai *precedence* paling tinggi, kemudian diikuti oleh *concatenation* dan terakhir *union*. Tanda kurung (*parentheses*) mempunyai arti seperti ekspresi matematika, jika kita ingin melakukan *grouping* tanpa mengikuti *precedence* sebenarnya. Tanda kurung bebas digunakan jika dibutuhkan, bahkan ketika arti dari *regular expression* tanpa tanda kurung sama dengan arti jika menggunakan tanda kurung. *Regular expression* bisa langsung kita tentukan secara intuitif. Contohnya, *regular expression* yang menerima semua *string* yang berakhir dengan 01 adalah  $(0+1)^*01$ . Bagian  $(0+1)^*$  artinya *closure* dari 0 atau 1, yang mencakup *empty string*, dan semua *string* lain dengan panjang berapapun yang terdiri dari kombinasi 0 dan 1. Bagian **01** secara eksplisit adalah bagian akhir *string*. Jadi, *regular expression* kita akan menerima *string* dengan bagian akhir 01 dengan bagian depannya *string* apapun. Untuk kasus yang lebih kompleks, terdapat cara menyusun dan menyederhanakan *regular expression*.

### 2.5.2 Definisi *Regular Expression (regex)*

Tahun 1956, Stephen Cole Kleene seorang ahli matematika membuat sebuah model *pattern string* menggunakan notasi matematika, yang disebut *regular sets*. Kemudian Ken Thompson mengimplementasikan notasi tersebut ke dalam *text editor* QED buatannya untuk pencarian *string* dengan pola tertentu. Dia juga menambahkan fitur tersebut ke dalam, sebuah *text editor* dalam sistem operasi UNIX yang dikenal dengan ED. Untuk pencarian *string* dalam ED

digunakan pattern : /g /re /p (/g : globally, /re : *regular expression*, /p : print) yang maksudnya adalah pencarian global baris-baris dalam sebuah file yang memiliki pola tertentu, dan ditampilkan atau di cetak. Sejak saat itu *regular expression* mulai banyak digunakan.

Regular Expression (*regex*) adalah konstruksi bahasa untuk mencocokkan teks berdasarkan pola tertentu, terutama untuk kasus-kasus kompleks. Selain itu, *Regex* juga sangat manjur (*powerful*) terutama untuk proses penguraian kata (*text parsing*), Regular expression (*regex*) memungkinkan melakukan pencarian, substitusi, atau pemisahan string dalam kasus yang kompleks (Didik Dwi Prasetyo, 2008).

*Regular Expression* (disebut juga *regex* atau *regexp*) adalah salah satu metode *string matching* yang cukup penting pada dunia informatika sekarang ini. *Regex* sangat banyak aplikasinya, terutama untuk validasi, seperti validasi nama, nomor telepon, *email*, dll. *Regex* sendiri berbeda dari algoritma *string matching* seperti KMP dan Boyer-Moore, karena *regex* memiliki fleksibilitas yaitu tidak selalu *exact match*, tergantung pada *pattern* dari *regex*. *Regex* sendiri bisa berbeda-beda implementasi, *syntax*, dan kemampuannya tergantung *regex engine* yang digunakan. Salah satu jenis *regex* yang paling umum dan paling *powerful* adalah PCRE (*Perl Compatible Regular Expression*).

Alasan dari *regex* ini dibedakan dengan *regular expression* dari *automata theory* bukan hanya karena *regex* secara *practical* memiliki *syntax* yang jauh lebih kompleks, tetapi pada umumnya *regex* yang ada di komputer

kita tidak lagi termasuk dalam *regular language*, tetapi sudah lebih *powerful* dari itu. *Regex* sudah memiliki kemampuan kondisional, rekursi, penamaan, penyimpanan nilai, dll.

Sebagai contoh, *regex* untuk *match* semua *string* yang berakhir dengan 01 adalah `.*01$`. Tanda `.` akan *match* semua karakter sebanyak 0 atau lebih kali, lalu `01` akan *match* 01 secara literal (harafiah). Tanda `$` dimasukkan ke dalam *pattern* untuk memastikan bahwa 01 terdapat di akhir *string*.

Kemampuan *regex* tidak hanya bersumber dari tingkat fleksibilitas dari *pattern regex*, tetapi juga dari *regex engine* yang melakukan *matching regex pattern*. Pada *regex engine*, biasanya terdapat *flag* yang bisa di-*set* untuk memodifikasi pencarian, seperti *ignore case* (besar kecilnya huruf tidak diperhatikan), *global* (mencari semua *match* yang ada), dll. Untuk penjelasan lebih lanjut tentang beberapa implementasi *syntax* dari *regex* akan dijelaskan di bab empat implementasi dan pembahasan.

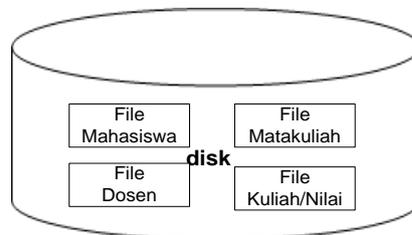
## 2.6 *Web Server*

*Web Server* adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam halaman-halaman web yang umumnya berbentuk dokumen HTML. *Web server* yang dimaksud disini adalah simulasi dari sebuah *web server* secara fisik. *Web server* biasanya juga disebut HTTP *server* karena menggunakan *protocol* HTTP sebagai basisnya. Beberapa *web server* yang sering digunakan diantaranya adalah PWS, IIS, Apache dan sebagainya. (Wikipedia, 2019).

## 2.7 Basis Data

### 2.7.1 Pengertian Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya tersimpan di simpanan luar komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam pertukaran data, karena berfungsi sebagai basis penyedia data bagi para pemakainya.



Gambar 2.1 Basis Pada Media Penyimpanan Perangkat Keras

(Ringgodoni, 2011)

Adapun istilah-istilah dalam basis data diantaranya adalah sebagai berikut:

- 1) *Entity* merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain, misalnya orang, tempat, kejadian, atau konsep yang informasinya direkam.
- 2) *Attribute* merupakan sesuatu atau hal yang mendeskripsikan karaktersistik (*property*) dari *entity*.
- 3) *Data value* (nilai atau isi data) adalah data actual atau informasi yang disimpan pada *attribute*.
- 4) *Record* (tupel) yaitu kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.

- 5) File merupakan kumpulan record-record sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, namun berbeda-beda *data value*-nya.
- 6) *Database management system* (DBMS) merupakan kumpulan *file* yang saling berkaitan bersama dengan program untuk mengolahnya. *Database* merupakan kumpulan datanya, sedangkan program pengelolaannya berdiri sendiri dalam satu paket program yang berfungsi untuk membaca data, mengisi data, menghapus data serta melaporkan data dalam *database*.

### 2.7.2 Kriteria-Kriteria Basis Data

Basis data (*database*) memiliki beberapa kriteria yang harus dipenuhi, yaitu sebagai berikut:

- 1) Bersifat *data oriented* dan bukan *program oriented*.
- 2) Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya.
- 3) Dapat berkembang dengan mudah, baik volume maupun strukturnya.
- 4) Dapat memenuhi sistem-sistem baru secara mudah.
- 5) Dapat digunakan dengan cara-cara berbeda.
- 6) Kerangkapan (redudansi) data dapat diminimalkan.

### 2.7.3 Database

Ketika aplikasi yang akan dibuat membutuhkan informasi yang akan ditampilkan ke user, seperti berita, stok barang, perkiraan cuaca dan lain-lain, maka perlu adanya aplikasi yang digunakan untuk menyimpan berbagai informasi atau data yang dibutuhkan dengan terstruktur. Dengan menggunakan PHP

sebenarnya pengguna dapat saja menyimpan data dengan bentuk teks, untuk aplikasi yang tidak begitu kompleks mungkin hal tersebut dapat saja dilakukan. Namun, ketika aplikasi yang dibuat cukup rumit dan kompleks, maka digunakan *database*. Dengan *database*, aplikasi web yang dibuat akan bersifat dinamis. MySQL adalah salah satu contoh *database* yang banyak digunakan untuk pembuatan aplikasi. Dengan konfigurasi *web server* apache yang baik PHP dan MySQL menjadi kombinasi yang baik. Dengan penggunaan *database*, aplikasi web yang akan dibuat akan menjadi lebih fleksibel dan memudahkan dalam *maintenance*.

#### **2.7.4 MySQL**

MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelola datanya. Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan *database* perusahaan – perusahaan skala menengah – kecil. MySQL merupakan *database* yang digunakan oleh situs - situs terkemuka di Internet untuk menyimpan datanya.

Software *database* MySQL kini dilepas sebagai software manajemen *database* yang *open source*, sebelumnya merupakan software *database* yang *shareware*. *Shareware* adalah suatu software yang dapat didistribusikan secara bebas untuk keperluan penggunaan secara pribadi, tetapi jika digunakan keperluan penggunaan secara pribadi, tetapi jika digunakan secara komersial maka pemakai harus mempunyai lisensi dari pembuatnya.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sehingga istilah seperti table, baris dan kolom tetap digunakan dalam MySQL. Pada MySQL, sebuah *database* mengandung beberapa *table*, satu *table* terdiri dari sejumlah baris dan kolom. Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam tabel – tabel yang secara logik merupakan struktur dua dimensi yang terdiri atas baris – baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom (*column*). Baris pada *table* sering disebut sebagai *instance* dari data sedangkan kolom sering disebut sebagai *attributes* atau *field*.

## 2.8 *State Of The Art*

*State Of The Art* merupakan penelitian lain yang terkait dengan penelitian yang akan dilakukan. Berikut di bawah ini adalah *State Of The Art* yang terkait:

1. Penelitian menurut Deddy Hidayat, Ika Marhaeni, Prasetyo, dan Yenni Dwi Kurniawati yang berjudul "ANALISA DAN PERANCANGAN APLIKASI PENGOLAHAN ARSIP SURAT JALAN (PESAN) PADA PT. SUBUR SENTOSA" menyimpulkan bahwa (Deddy Hidayat, Ika Marhaeni, Prasetyo, dan Yenni Dwi Kurniawati, 2014) analisa dan rancangan terhadap Sistem pengolahan arsip surat jalan mulai dari *supply* barang sampai dengan tanda bukti pengiriman barang (barang telah diterima) yang dilakukan masih konvensional. Sehingga alternatif yang diajukan yaitu mengoptimalisasikan sistem pengolahan arsip surat jalan di PT. Subur Sentosa menggunakan suatu sistem yang berbasis web untuk mendukung proses pengolahan arsip surat jalan yang bertujuan untuk membuat sistem laporan. Laporan data barang yang kembali menggunakan data surat jalan

juga bisa diketahui. Dalam aplikasi ini terdapat menu utama terdiri dari *home*, *About us* (berisi kata-kata sambutan dan organisasi dari PT. Subur Sentosa), *service* (menu transaksi untuk meng-input surat jalan masuk dan keluar). Dalam menu ini juga terdapat pilihan untuk melihat laporan rekap berdasarkan tanggal. Sistem laporan dalam pengolahan data arsip ini dengan langsung memberikan bayangan surat jalan asli dan form bukti penerimaan barang kepada *purchasing* yang selanjutnya digunakan untuk proses penagihan terhadap barang.

2. Penelitian menurut Mohamad Charis yang berjudul "SISTEM INFORMASI PENGELOLAAN SURAT MASUK DAN SURAT KELUAR DENGAN VISUAL FOXPRO 8.0" menyimpulkan bahwa (Mohamad Charis, 2009) aplikasi pengelolaan surat berfungsi untuk membantu meringankan pekerjaan seorang agendaris yang dilakukan setiap hari sehubungan dengan adanya kegiatan di kantor yang bersifat kedinasan. Sistem kerja aplikasi ini terbagi menjadi dua bagian yaitu surat masuk dan surat keluar. Untuk surat masuk agendaris tinggal menginput mulai dari tanggal pencatatan, nomor surat, kode surat, tanggal surat, asal surat dan perihal. Secara otomatis komputer akan menyimpannya. Sedangkan surat keluar disediakan konsepsurat pengantar yang berbentuk kolom setelah jadi, secara otomatis akan masuk ke dalam agenda surat keluar. Dalam aplikasi ini juga dapat melihat laporan data surat. Sedangkan pengembangan sistem aplikasi ini menggunakan software Visual Foxpro 8.0.

3. Penelitian menurut Arie Vironica, Sukadi yang berjudul "RANCANG BANGUN APLIKASI PENGELOLAAN SURAT MASUK DAN SURAT KELUAR PADA SEKOLAH MENENGAH PERTAMA NEGERI 2 NAWANGAN" menyimpulkan bahwa (Arie Sukadi, 2013) Sistem ini dibuat dengan Sistem satu user dengan menggunakan program PHP (*Hypertext Preprocessor*), MySQL sebagai *database* nya, Adobe Photoshop CS 3 sebagai desain grafisnya dan windows 7 sebagai Sistem operasinya. Dalam aplikasi surat ini user dapat mengurangi penggunaan proses yang cukup lama dalam pengarsipan surat, mengurangi penggunaan kertas, mempercepat proses pencarian arsip surat, dan memudahkan pengontrolan disposisi surat. Aplikasi ini juga memudahkan proses komunikasi data antar bagian serta pembuatan laporan yang selalu di update berdasarkan laporan bulanan maupun tahunan.
4. Penelitian menurut Rizky Maulana Syaban, H. Bunyamin yang berjudul "PENGEMBANGAN SISTEM INFORMASI PENGELOLAAN SURAT MASUK DAN SURAT KELUAR BERBASIS WEB DI DINAS SOSIAL TENAGA KERJA DAN TRANSMIGRASI KABUPATEN GARUT MENGGUNAKAN FRAMEWORK PHI" menyimpulkan bahwa (Rizky Maulana Syaban, H. Bunyamin, 2015) tujuan dari pengembangan sistem pengelolaan surat masuk dan surat keluar ini adalah untuk menyediakan sistem pengelolaan surat berbasis komputer yang terdiri dari proses pengelolaan surat masuk dan surat keluar serta pembuatan laporan pengelolaan surat. Metode pengembangan sistem yang digunakan yaitu

USDP (*Unified Software Development Process*) karena USDP merupakan salah satu kerangka kerja pengembangan perangkat lunak berorientasi objek. Dalam pengembangan aplikasi ini menggunakan MySQL sebagai manajemen basis datanya, PHP dan HTML sebagai pengkodean dasar untuk membangun halaman web serta CSS untuk mendesain tampilan dari web dan sublime text digunakan sebagai penyunting kodenya. Sedangkan untuk pengujian sistem informasi ini menggunakan metode *black box testing*. *Black box* itu sendiri merupakan pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari *software*.

5. Penelitian menurut Sapto Aji, Migunani, Fitro Nur Hakim yang berjudul "RANCANG BANGUN SISTEM INFORMASI DISPOSISI SURAT BERBASIS' WEB (STUDI KASUS: KEMENTRIAN PEKERJAAN UMUM) " menyimpulkan bahwa (Sapto Aji, Migunani, Fitro Nur Hakim, 2014) Sistem informasi disposisi surat dirancang untuk memperbaiki kekurangan pada proses pengarsipan disposisi Surat di Kementerian PU. Perancangan dilakukan dengan mempertimbangkan setiap kelemahan dari sistem manual yaitu faktor kemanan, efisiensi waktu, ruang dan biaya serta fleksibilitas proses disposisi surat, dengan tujuan membuat perubahan yang signifikan. Perubahan yang dimaksud yaitu perubahan dari manual menjadi otomatis, perubahan dari analog menjadi digital dan perubahan dari lambat menjadi cepat. Dalam pengembangan aplikasi ini menggunakan metode waterfall. Dari implementasi ini dihasilkan aplikasi yang menyediakan fitur

halaman login, halaman daftar surat, halaman daftar disposisi, dan halaman update disposisi.

6. Penelitian menurut Bibit, Sukadi yang berjudul "SISTEM PENGELOLAAN SURAT MASUK DAN SURAT KELUAR TERKOMPUTERISASI PADA UNIT PELAKSANA TEKNIS (UPT) SEKOLAH MENENGAH PERTAMA (SMP) NEGERI 1 TEGALOMBO" menyimpulkan bahwa (Sukadi, 2015) pengarsipan surat dilakukan secara terkomputerisasi dengan menggunakan macam-macam elemen sistem berupa perangkat lunak (*software*), perangkat keras (*hardware*), operator (*brainware*), *database*, dokumentasi, dan prosedur. Dalam pembuatan aplikasi ini bahasa pemrograman yang digunakan yaitu PHP, MySQL sebagai database nya, dan Adobe Dreamweaver yang bertujuan untuk membuat fitur yang menyediakan penyimpanan berkas surat secara elektronik, sehingga data-data penting tidak rusak. Aplikasi ini menghasilkan pengarsipan surat yang dilakukan secara elektronik yaitu penyimpanan dokumen *softcopy* laporan surat masuk dan surat keluar, baik perhari maupun perperiode. Selain itu dapat melakukan pencarian data surat sebagai sistem yang dapat memberikan kemudahan bagi user.
7. Penelitian menurut Dewi Leyla Rahmah yang berjudul "PERANCANGAN APLIKASI SISTEM PERSUIUTAN BERBASIS WEB PADA PT. DWI PILAR PRATAMA " menyimpulkan bahwa (Dewi Leyla Rahmah, 2014) Dalam pembuatan aplikasi ini menggunakan metode *grounded research* Metode *grounded research* merupakan suatu metode yang berdasarkan fakta

dan menggunakan analisis perbandingan. Sehingga perancangan aplikasi secara terkomputerisasi ini dapat menghasilkan informasi yang tepat dan akurat. Untuk proses penyimpanan arsipnya disimpan ke hardisk komputer dan di rekap file nya setiap bulan. Dari data yang ada di komputer juga, aplikasi ini dapat melakukan pencarian data surat dan pembuatan laporan bulanannya.

8. Penelitian menurut Sri Rahayu, Dara Hana yang berjudul "SISTEM INFORMASI PENDATAAN SURAT MASUK DAN SURAT KELUAR PADA SMK NEGERI 5 TANGERANG" menyimpulkan bahwa (Sri Rahayu, Dara Hana, 2016) dalam pembuatan aplikasi ini, sistem membutuhkan data untuk diolah sehingga menghasilkan informasi yang sesuai dengan yang dibutuhkan. Adapun beberapa informasi yang dibutuhkan bagian Tata Usaha adalah informasi hasil verifikasi yang bersumber dari kartu kendali, informasi lainnya adalah hasil dari proses pencarian data surat yang sudah pernah diproses yang dan tersimpan dalam *database*. Dengan adanya aplikasi yang disediakan ini, Sistem juga memberikan kemudahan bagi user (Tata Usaha) saat akan membuat laporan setiap akhir bulan, karena data yang dimasukkan setiap hari secara otomatis digeneralisasi menjadi laporan. dibuat dengan menggunakan Macromedia Rancangan aplikasi.
9. Penelitian menurut Endang Sugiharti, sulis Eli Triliani yang berjudul "PERANCANGAN APLIKASI SURAT MASUK DAN SURAT KELUAR PADA PT. ANGKASA PURA 1 SEMARANG" menyimpulkan bahwa

(Endang Sugiharti, Sulis Eli Triliani, 2014) perancangan sistem aplikasi ini menghasilkan penguasaan data surat, update surat, pencarian data, dan pembuatan laporan surat. Dalam pengembangan aplikasi ini juga dapat digunakan sistem *client-server*. Sedangkan pemrograman aplikasi ini menggunakan visual basic 6.0, MySQL sebagai database nya dan crystal report. Program visual basic ini dalam pembuatan laporan bisa dikombinasikan dengan CR (crystal report). Secara defacto CR merupakan standar untuk pembuatan laporan pada sistem operasi windows, di mana cetakkan atau template laporan yang dihasilkan dapat disertakan pada banyak bahasa pemrograman.

10. Penelitian menurut Muhammad Khoirul, Tri Irianto Tjendrowasono, Berliana Kusuma Rasti yang berjudul "APLIKASI PENGELOLAAN DATA KEARSIPAN PADA SEKOLAH MENENGAH ATAS NEGERI 1 MLONGGO JEPARA BERBASIS MULTIUSER" menyimpulkan bahwa (Muhammad Khoirul, Tri Irianto Tjendrowasono, Berliana Kusuma Rasti, 2013) dalam merancang aplikasi ini menghasilkan pengelolaan data kearsipan surat dan pengelolaan surat keputusan. Pembuatan aplikasi ini menggunakan bahasa pemrograman basis data berupa Visual Foxpro 9.0 yang bersifat multiuser yang berarti kumpulan beberapa komputer yang ditunjang dengan peralatan lainnya seperti printer, CD-room, dan scanner yang terhubung dalam satu kesatuan. Membangun sistem ini memiliki rancangan desain untuk dijadikan aplikasi yang akan menghasilkan implementasi berupa halaman login, administrator, input data user, input

data lokasi, input data instansi, input data pegawai, input data surat masuk, input data surat keluar, input data surat keputusan, cetak data master, hasil cetak data master, cetak data transaksi, hasil cetak data transaksi, dan terdapat hasil uji implementasi yang menghasilkan grafik.

### 2.8.1 Peta Penelitian

Persamaan dan perbedaan jurnal terkait dengan penelitian yang dilakukan

#### a. Persamaan

1. Bahasa pemrograman yang dipakai PHP
2. Database yang dipakai MySQL
3. Pengujian aplikasi menggunakan *black box*

#### b. Perbedaan

1. Secara umum tidak menggunakan algoritma pencarian *regular expression*.
2. Secara umum tidak melakukan pengujian performa pada proses pencarian data-nya
3. Fitur sistem yang digunakan dalam jurnal terkait hanya input dan lihat, sedangkan penelitian yang dilakukan selain ada input dan lihat, juga terdapat implementasi output berupa *print out*