

BUKU TUTORIAL APLIKASI ARSIP SURAT

“SISTEM MANAJEMEN APLIKASI ARSIP SURAT MASUK DAN SURAT KELUAR DI PT. POS INDONESIA (PERSERO) DENGAN SISTEM KEARSIPAN ALPHABETICAL FILING SYSTEM DAN CHRONOLOGY FILING SYSTEM” (STUDI KASUS: DIVISI ASSESSMENT CENTER)

Buku ini dibuat untuk memenuhi persyaratan kelulusan
matakuliah Program Internship I



Dibuat Oleh,

1.16.4.067 Annisa Fathoroni

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA
BANDUNG
2020**

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat, hidayah serta kekuatan sehingga Buku tutorial ini dengan judul “Sistem Manajemen Aplikasi Arsip Surat Masuk dan Surat Keluar Di PT. Pos Indonesia (Persero) dengan Sistem Kearsipan *Alphabetical Filing System* dan *Chronology Filing System* (Studi Kasus: Divisi Assessment Center)” dapat terselesaikan.

Banyak kendala yang dihadapi dalam penyusunan Buku ini dan penulis menyadari bahwa penyusunan laporan ini masih belum sempurna. Ini mengingat keterbatasan pengetahuan, pengalaman serta kemampuan penulis. Penulis megharapkan kritik dan saran yang sifatnya membangun dari pembaca. Untuk itu penulis mengucapkan terima kasih kepada:

1. Dra. Latifah Handayani selaku pembimbing *external* di PT. Pos Indonesia (Persero).
2. Rd. Nuraini Siti Fatonah, S.S., M.Hum. selaku dosen pembimbing Internship I.
3. Nisa Hanum Harani, S.Kom., M.T. selaku Koordinator Internship I.
4. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi DIV Teknik Informatika.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga buku ini membawa manfaat bagi pengembangan ilmu.

Bandung, 20 Januari 2020

Penulis

DAFTAR ISI

Kata Pengantar	2
DAFTAR ISI.....	3
DAFTAR GAMBAR	5
DAFTAR TABEL.....	6
BAB I.....	1
PENGENALAN SISTEM, APLIKASI, MANAJEMEN	1
1.1 Pengantar Sistem	1
1.2 Pengantar Aplikasi.....	13
1.3 Pengantar Manajemen	23
BAB II.....	45
PENGENALAN SURAT, ARSIP, SISTEM KEARSIPAN	
<i>ALPHABETICAL FILING SYSTEM</i> DAN <i>CHRONOLOGY FILING</i>	
<i>SYSTEM</i>	45
2.1. Pengantar Surat.....	45
2.2. Pengantar Arsip	52
2.3. Pengantar Sistem Kearsipan <i>Alphabetical Filling System</i> dan	
<i>Chronology Filling System</i>	68
BAB III	71
PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG	
DIGUNAKAN	71
3.1. Tools Yang Digunakan	71
3.2. Bahasa Pemograman.....	84
BAB IV	91
INSTALASI <i>TOOLS</i> YANG DIGUNAKAN	91
1. <i>Tools</i> Yang Digunakan.....	91
2. Bahasa Pemrograman Yang Digunakan.....	103
BAB V	109
TUTORIAL PEMBUATAN APLIKASI.....	109

5.1.	Mempersiapkan <i>Tools</i>	110
5.2.	Membuat <i>database</i>	110
5.3.	Konfigurasi <i>CodeIgniter</i>	115
5.4.	Konfigurasi <i>Template CSS Bootstrap</i>	120
5.5.	Penjelasan <i>Source Code</i> Sistem Aplikasi	125

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENGENALAN SISTEM, APLIKASI, MANAJEMEN

Sistem, aplikasi dan manajemen adalah beberapa istilah yang digunakan dalam penyusunan buku ini. Penulis akan memaparkan penjelasan mengenai ketiga istilah tersebut.

1.1 Pengantar Sistem

Sistem adalah suatu jaringan dari prosedur-prosedur yang saling berhubungan, berkumpul dan bersama-sama untuk menyelesaikan suatu sasaran tertentu. Kata Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*). Dapat diartikan bahwa sistem adalah kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki penggerak, contoh umum misalnya seperti negara.

1.1.1 Penggunaan Istilah

Penggunaan istilah sistem itu meliputi diantaranya, yaitu:

1. Sistem yang digunakan untuk menunjukkan suatu kumpulan yang disatukan oleh suatu bentuk saling hubungan yang saling ketergantungan yang teratur secara alamiah maupun oleh budi daya manusia sehingga menjadi suatu kesatuan yang hakiki, suatu yang berfungsi bersama-sama mengikuti suatu kontrol tertentu. Salah satu contohnya adalah ekosistem dan sistem tata surya.
2. Sistem yang berhubungan dengan alat-alat atau organ tubuh secara keseluruhan yang secara khusus memiliki hubungan terhadap fungsi organ tubuh tertentu yang rumit namun tetapi sangat vital. Salah satu contohnya adalah sistem syaraf.
3. Sistem yang menunjukkan sekumpulan gagasan, prinsip serta doktrin hukum yang membentuk suatu hubungan yang dikenal

sebagai isi buah pemikiran tentang filsafat, agama atau pemerintahan tertentu. Salah satu contohnya adalah sistem pemerintahan demokratis.

4. Sistem yang digunakan untuk menunjukkan suatu teori atau hipotesis. Salah satu contohnya adalah sistem pendidikan sistematis.
5. Sistem yang digunakan dalam tata acara atau metode. Salah satu contohnya adalah sistem mengetik dengan menggunakan sepuluh jari, sistem modul pembinaan atau pengajaran.
6. Sistem yang digunakan untuk menunjukkan metode pengaturan organisasi atau susunan tata cara dalam metode pengelompokan, dan sebagainya. Salah satu contohnya adalah sistem pengelompokan bahan pustaka.

1.1.2 Pemakaian Sistem

Pemakaian sistem dapat digolongkan menjadi 2 secara garis besarnya, yaitu diantaranya:

1. Sistem sebagai suatu wujud entitas.

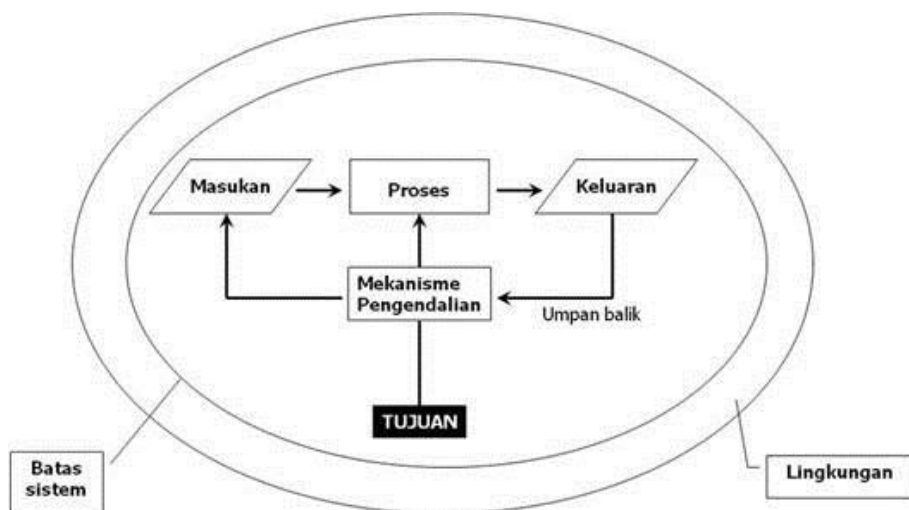
Suatu sistem dianggap suatu bagian yang saling berkaitan membentuk keseluruhan yang komplek. Sistem sendiri merupakan suatu wujud atau entitas sebagai suatu benda yang pada dasarnya bersifat deskriptif. Hal ini memberikan kemungkinan untuk membedakan atau menggambarkan antara benda yang satu dengan yang lainnya untuk kepentingan penganalisaan untuk mempermudah pemecahan masalah.

2. Sistem sebagai suatu metode.

Sistem mempunyai makna metodologi, sistem yang digunakan akan menunjukkan tata cara atau prosedur yang bersifat preskriptif

bukan deskriptif. Salah satu contohnya adalah deskriptif (program investasi) dan preskriptif (program investasi yang meningkatkan deviden). Contoh deskriptif menunjukkan wujud barang dan preskriptif menunjukkan suatu metode atau cara untuk mencapai sesuatu. Konsep ini dikenal dalam pengertian sebagai pendekatan sistem. Pendekatan tersebut ialah penerapan metode ilmiah dalam usaha memecahkan masalah. Menentukan pemahaman bahwa setiap sistem berada dari sistem yang besar atau luas sehingga semua benda dengan suatu cara yang saling berkaitan.

1.1.3 Karakteristik Sistem



Gambar 1.1 Karakteristik Sistem

Karakteristik sistem diantaranya meliputi sebagai berikut:

1. Memiliki komponen.

Suatu sistem terdiri dari sejumlah komponen yang saling berhubungan atau berinteraksi, bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem yang dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli

seberapapun kecilnya selalu mengandung komponen atau subsistem. Setiap subsistem mempunyai sifat dari sistem untuk menjalankan fungsi tertentu dan mempengaruhi cara kerja sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang biasa disebut sebagai supra sistem. Salah satu contohnya adalah perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar sehingga dapat disebut dengan supra sistem. Terlihat dari pandangan industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Jika perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

2. Batas sistem (*boundary*).

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan diluarnya. Batas sistem ini memungkinkan suatu sistem yang dipandang sebagai suatu kesatuan. Batasan suatu sistem untuk menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem (*environment*).

Lingkungan luar sistem merupakan apapun yang berada diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung sistem (*interface*).

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya.

5. Masukkan sistem (*input*).

Masukan sistem merupakan energi yang dimasukkan kedalam sistem. Masukan dapat berupa perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input*

adalah energi yang diproses untuk mendapatkan keluaran. Salah satu contohnya adalah sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputer dan data adalah sebagai signal input untuk diolah menjadi informasi.

1.1.4 Klasifikasi Sistem

Klasifikasi sistem diantaranya meliputi sebagai berikut:

1. Sistem abstrak merupakan sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Salah satu contohnya adalah sistem teologia.
2. Sistem fisik merupakan sistem yang ada secara fisik. Salah satu contohnya adalah sistem komputer, sistem akuntansi, sistem produksi dll.
3. Sistem alamiah merupakan sistem yang terjadi melalui proses alam. Salah satu contohnya adalah sistem matahari, sistem luar angkasa, sistem reproduksi dll.
4. Sistem buatan manusia merupakan sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*. Salah satu contohnya adalah contoh istem informasi.
5. Sistem tertentu (*deterministic system*) merupakan beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Salah satu contohnya adalah sistem komputer.

6. Sistem tak tentu (*probabilistic system*) merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
7. Sistem tertutup (*close system*) merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system*.
8. Sistem terbuka (*open system*) merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

1.1.5 Sistem Terotomasi

Sistem terotomasi mempunyai sejumlah komponen yaitu diantaranya:

1. Perangkat keras, misalnya CPU, *disk*, *printer*, *tape*.
2. Perangkat lunak, misalnya sistem operasi, sistem *database*, program pengontrol komunikasi, program aplikasi.
3. Personil yang mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem.
4. Data yang harus tersimpan dalam sistem selama jangka waktu tertentu).

5. Prosedur instruksi dan kebijakan untuk mengoperasikan sistem.

Sistem terotomasi terbagi dalam sejumlah kategori :

1. *On-line systems.*

Sistem *on-line* adalah sistem yang menerima langsung *input* pada area, dimana *input* tersebut direkam dan menghasilkan *output* yang dapat menghasilkan komputasi pada area. Area dapat dipisah-pisah dalam skala. Salah satu contohnya adalah ratusan kilometer. Biasanya digunakan bagi reservasi angkutan udara, reservasi kereta api, perbankan, dll.

2. *Real-time systems.*

Sistem *real-time* adalah mekanisme pengontrolan, pemrosesan, dan perekaman data yang sangat cepat sehingga output yang dihasilkan dapat diterima dalam waktu yang relatif sama. Perbedaannya dengan sistem *on-line* adalah satuan waktu yang digunakan *real-time* biasanya seperseratus atau seperseribu detik, sedangkan *on-line* masih dalam skala detik atau bahkan kadang memerlukan waktu beberapa menit. Perbedaan lainnya, *on-line* biasanya hanya berinteraksi dengan pemakai, sedangkan *real-time* berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

3. *Decision support system + strategic planning system.*

Sistem yang memproses transaksi organisasi secara harian dan membantu para manajer mengambil keputusan, mengevaluasi dan menganalisa tujuan organisasi. Digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi dan sistem produksi. Biasanya berbentuk paket statistik, paket pemasaran, dll. Sistem ini tidak hanya merekam dan menampilkan data tetapi juga fungsi matematik, data

analisa statistik dan menampilkan informasi dalam bentuk grafik, tabel, chart sebagaimana laporan konvensional.

4. *Knowledge-based system.*

Program komputer yang dibuat mendekati kemampuan dan pengetahuan. Umumnya menggunakan perangkat keras dan perangkat lunak seperti PROLOG dan LISP.

1.1.6 Jenis Sistem

Ada berbagai tipe sistem berdasarkan kategori:

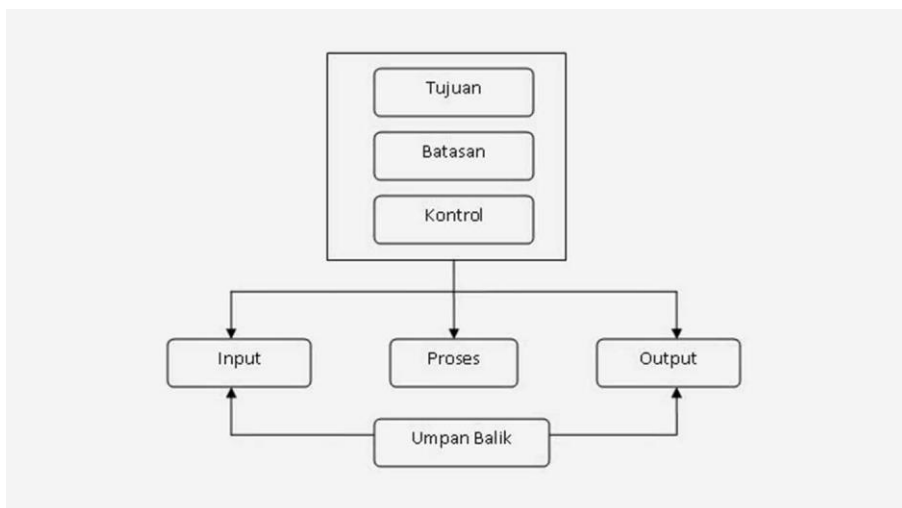
2.2.1. Atas dasar keterbukaan:

- a. Sistem terbuka, di mana pihak luar dapat mempengaruhinya.
- b. Sistem tertutup.

2.2.2. Atas dasar komponen:

- a. Sistem fisik, dengan komponen materi dan energi.
- b. Sistem non-fisik atau konsep, berisikan ide-ide.

1.1.7 Elemen Sistem



Gambar 1.2 Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, berikut penjelasannya sebagai berikut:

1. Tujuan

Setiap sistem memiliki tujuan, entah hanya satu atau lebih. Tujuan inilah yang memotivasi dan mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Masukan

Masukan (*input*) sistem merupakan segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan bisa berupa hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

3. Proses

Proses adalah bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga dapat berupa hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

4. Keluaran

Keluaran (*output*) adalah hasil dari proses. Pada sistem informasi berupa informasi atau laporan, dan lain-lain.

5. Batas

Yang disebut batas sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi,

ruang lingkup, atau kemampuan sistem. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem.

6. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian diwujudkan dengan menggunakan umpan balik, yang mencuplik keluaran. Umpan balik ini digunakan untuk mengontrol baik masukan maupun proses.

7. Lingkungan

Lingkungan bisa berpengaruh terhadap operasi sistem, dalam arti bisa merugikan atau menguntungkan sistem itu sendiri.

1.1.8 Sistem Berdasarkan Prinsip

Sistem berdasarkan prinsip dasar secara umum terbagi dalam beberapa macam, diantaranya yaitu:

1. Sistem terspesialisasi adalah sistem yang sulit diterapkan pada lingkungan yang berbeda, misalnya sistem biologi, ikan yang dipindahkan ke darat.
2. Sistem besar adalah sistem yang sebagian besar sumber dayanya berfungsi melakukan perawatan harian, misalnya dinosaurus sebagai sistem biologi menghabiskan sebagian besar masa hidupnya dengan makan dan makan.
3. Sistem sebagai bagian dari sistem lain adalah sistem selalu merupakan bagian dari sistem yang lebih besar, dan dapat terbagi menjadi sistem yang lebih kecil.
4. Sistem berkembang merupakan sistem yang walaupun tidak berlaku bagi semua sistem tetapi hampir semua sistem selalu berkembang.

1.1.9 Pelaku Sistem

Pelaku sistem terdiri dari 7 kelompok, diantaranya sebagai berikut:

1. Pemakai, pada umumnya 3 ada jenis pemakai, yaitu operasional, pengawas dan eksekutif.
2. Manajemen, pada umumnya terdiri dari 3 jenis manajemen, yaitu manajemen pemakai yang bertugas menangani pemakaian dimana sistem baru diterapkan, manajemen sistem yang terlibat dalam pengembangan sistem itu sendiri dan manajemen umum yang terlibat dalam strategi perencanaan sistem dan sistem pendukung pengambilan keputusan. Kelompok manajemen biasanya terlibat dengan keputusan yang berhubungan dengan orang, waktu dan uang.
3. Pemeriksa, ukuran dan kerumitan sistem yang dikerjakan dan bentuk alami organisasi dimana sistem tersebut diimplementasikan dapat menentukan kesimpulan perlu tidaknya pemeriksa. Pemeriksa biasanya menentukan segala sesuatunya berdasarkan ukuran-ukuran standar yang dikembangkan pada banyak perusahaan sejenis.
4. Penganalisa sistem, fungsi-fungsinya antara lain yaitu:
 - a. Arkeolog yaitu yang menelusuri bagaimana sebenarnya sistem lama berjalan, bagaimana sistem tersebut dijalankan dan segala hal yang menyangkut sistem lama.
 - b. Inovator, yaitu yang membantu mengembangkan dan membuka wawasan pemakai bagi kemungkinan-kemungkinan lain.
 - c. Mediator, yaitu yang menjalankan fungsi komunikasi dari semua level, antara lain: pemakai, manajer, programmer, pemeriksa dan pelaku sistem yang lainnya yang mungkin belum punya sikap dan cara pandang yang sama.

- d. Pimpinan proyek, penganalisa sistem haruslah personil yang lebih berpengalaman dari programmer atau desainer. Selain itu mengingat penganalisa sistem umumnya ditetapkan terlebih dahulu dalam suatu pekerjaan sebelum yang lain bekerja, adalah hal yang wajar jika penanggung jawab pekerjaan menjadi porsi penganalisa sistem.
5. Pendesain sistem merupakan sistem menerima hasil penganalisa sistem berupa kebutuhan pemakai yang tidak berorientasi pada teknologi tertentu, yang kemudian ditransformasikan ke desain arsitektur tingkat tinggi dan dapat diformulasikan oleh programmer.
6. Programmer mengerjakan dalam bentuk program dari hasil desain yang telah diterima dari pendesain.
7. Personel pengoperasian bertugas dan bertanggungjawab di pusat komputer misalnya jaringan, keamanan perangkat keras, keamanan perangkat lunak, pencetakan dan *backup*. Pelaku ini mungkin tidak diperlukan bila sistem yang berjalan tidak besar dan tidak membutuhkan klasifikasi khusus untuk menjalankan sistem.

1.1.10 Analisis Sistem

Penganalisa sistem merupakan bagian dari tim yang berfungsi mengembangkan sistem yang memiliki daya guna tinggi dan memenuhi kebutuhan pemakai akhir. Pengembangan ini dipengaruhi sejumlah hal, yaitu:

1. Produktifitas, saat ini dibutuhkan sistem yang lebih banyak, lebih bagus dan lebih cepat. Hal ini membutuhkan lebih banyak programmer dalin penganalisa sistem yang berkualitas, kondisi kerja ekstra, kemampuan pemakai untuk mengembangkan sendiri, bahasa

pemrograman yang lebih baik, perawatan sistem yang lebih baik (umumnya 50 % sampai 70 % sumber daya digunakan untuk perawatan sistem), disiplin teknis pemakaian perangkat lunak dan perangkat pengembangan sistem yang terotomasi.

2. Realibilitas, waktu yang dihabiskan untuk testing sistem secara umum menghabiskan 50% dari waktu total pengembangan sistem. Dalam kurun waktu 30 tahun sejumlah sistem yang digunakan di berbagai perusahaan mengalami kesalahan dan ironisnya sangat tidak mudah untuk mengubahnya. Jika terjadi kesalahan, ada dua cara yang bisa dilakukan, yaitu melakukan pelacakan sumber kesalahan dan harus menemukan cara untuk mengoreksi kesalahan tersebut dengan mengganti program, menghilangkan sejumlah statement lama atau menambahkan sejumlah statement baru.
3. Maintabilitas, perawatan mencakup:
 - a. Modifikasi sistem sesuai perkembangan perangkat keras untuk meningkatkan kecepatan pemrosesan (yang memegang peranan penting dalam pengoperasian sistem)
 - b. Modifikasi sistem sesuai perkembangan kebutuhan pemakai. Antara 50% sampai 80% pekerjaan yang dilakukan pada kebanyakan pengembangan sistem dilakukan untuk revisi, modifikasi, konversi, peningkatan dan pelacakan kesalahan.

1.2 Pengantar Aplikasi

Aplikasi adalah sebuah program siap pakai yang bisa dipakai untuk menjalankan sejumlah perintah dari pengguna aplikasi itu sendiri. Dengan tujuan untuk memperoleh hasil yang lebih akurat dan sesuai dengan tujuan pembuatan aplikasi tersebut. Aplikasi juga memiliki pengertian sebagai

pemecah masalah yang memakai salah satu teknik pemrosesan data aplikasi yang mengacu pada sebuah komputerisasi atau smartphone yang diinginkan atau diharapkan. Aplikasi berasal dari kata bahasa Inggris “*Application*” yang artinya merupakan bentuk dari kata kerja *to apply* atau dalam bahasa Indonesia artinya pengolah. Secara istilah, aplikasi komputer adalah subkelas perangkat lunak komputer yang memakai kemampuan komputer dengan langsung melaksanakan suatu tugas yang diinginkan pengguna tersebut. Selain itu, pengertian aplikasi adalah suatu perintah yang mengeksekusi dalam memberikan sebuah petunjuk kerja serta fungsi yang diinginkan. Dan pengertian aplikasi secara umum adalah sebagai alat terapan yang berfungsi secara khusus serta terpadu sesuai kemampuan yang dimiliki aplikasi adalah suatu perangkat komputer yang sudah siap dipakai sebagai *user*.

Kesimpulan dari pengertian diatas adalah bahwa aplikasi ini merupakan *software* yang fungsinya untuk melaksanakan beragam bentuk pekerjaan maupun tugas tugas tertentu misalnya seperti penerapan, pemakaian dan juga penambahan data.

1.2.1. Jenis-Jenis Aplikasi Komputer

Jenis-jenis aplikasi komputer yang ada atau telah dipergunakan, yaitu:

1. Aplikasi grafis.

Aplikasi grafis merupakan suatu program untuk mengolah data yang berformat gambar baik dengan membuat gambar baru maupun mengubah gambar yang sudah dibuat sebelumnya. *Software* aplikasi grafis dibagi menjadi tiga macam, yaitu:

a. Aplikasi grafis berbasis *vector*

Contoh: *Adobe Illustrator, Corel Draw, Macromedia Freehand, Micrografxdesigner.*

b. Aplikasi grafis berbasis *pixel/BITMAP*

Contoh: *Adobe Photoshop, Macromedia Fireworks, Corel Photopaint.*

c. Aplikasi grafis berbasis tata letak

Contoh: *Adobe FrameMaker, Adobe In Design, Adobe PageMaker, Corel Ventura, Microsoft Publisher, Quark Xpress.*

2. Aplikasi *Web Browser*

Merupakan bagian dari internet sebagai komunitas jaringan komputer yang memberikan pelayanan http (*world wide web*). Dengan demikian, definisi teknis dari *world wide web* adalah semua sumber daya dan semua pengguna di internet yang menggunakan HTTP (*Hypertext Transfer Protocol*). WWW adalah aplikasi yang paling menarik di Internet dan seperti email, aplikasi ini sangat penting dan banyak digunakan. Aplikasi *web browser* yang sering digunakan antara lain, *Opera, Internet Explorer, Safari, Firefox* dan *Chrome*.

3. Animasi

Merupakan bentuk seni yang tampak secara spontan menimbulkan gerakan kehidupan pada suatu obyek. Sebenarnya terdapat beberapa fungsi yang berbeda untuk menghasilkan animasi berbasis komputer dan satu daripadanya ialah animasi tiga dimensi (3D). Fungsi lain untuk mencipta animasi komputer ialah dengan menggunakan alat pengecatan komputer yang standar untuk mengecat frame-frame tunggal sebelum dilakukan proses penggabungan. Ini kemudian disimpan sebagai sebuah file gambar (*movie*).

4. Aplikasi Pendidikan

a. *Computer Assisted Instruction (CAI)*

Komputer secara langsung digunakan dalam proses belajar, sebagai pengganti pengajar ataupun buku. Beberapa aplikasi CAI antara lain, *Drill and Practice*, Tutorial, Simulasi.

b. *Computer Managed Instruction (CMI)*

Para pengajar memanfaatkan komputer untuk merencanakan kuliah, disesuaikan dengan kondisi para siswa, yang terdiri dari acara belajar dengan bantuan komputer, membaca, dan ujian.

c. *Computer Assisted Testing (CAT)*

Komputer digunakan sebagai media ujian untuk menggali kemampuan siswa dengan cara-cara tanya jawab secara aktif. Bentuknya bermacam-macam, dari mulai yang sederhana dimana komputer (biasanya melalui layar peraga) hingga bentuk yang lebih maju.

5. Aplikasi Multimedia

Multimedia merupakan *software* yang digunakan untuk menghubungkan komputer dengan peralatan multimedia seperti kamera video, kamera digital, video. Salah satu contohnya adalah *Windows Media Player*, *Winamp*, *PowerDVD*, *Klite*, *VLC Media Player*.

6. Antivirus

Antivirus merupakan program yang digunakan untuk mendeteksi dan menghilangkan virus yang tertular pada komputer yang sedang dipakai. Salah satu contohnya adalah *McAfee VirusScan*, *Norton Antivirus*, *AVG*.

7. Aplikasi Komunikasi

Aplikasi yang satu ini merupakan yang saat ini paling banyak digunakan dan merupakan yang paling populer. Aplikasi ini digunakan agar manusia bisa berkomunikasi dengan pengguna komputer, *smartphone* atau gadget lain. Salah satu contohnya adalah *DataFax*, *Carbon Copy*, *CrossTalk*, *Line*, *BBM*, *Whatsapp*, dll.

8. Aplikasi DBSM

Aplikasi DBMS (*Database Management System*) digunakan untuk menyimpan data, mengolah data, serta menghasilkan *output* berupa informasi. Aplikasi seperti ini ada yang tersedia secara gratis, namun ada juga yang berbayar. Salah satu contohnya adalah *MySQL*, *Microsoft Access*, *Oracle*, *Foxpro*, dan lain-lain.

1.2.2. Klasifikasi Aplikasi

Demi mempermudah mengenal aplikasi, maka aplikasi biasanya diklasifikasikan menjadi 7 klasifikasi macam, yaitu:

1. *System software*, perangkat lunak inilah yang akan mengelola serta mengendalikan jalannya operasi internal pada sistem komputer.
2. *Real time software*, yaitu perangkat lunak yang bertugas menganalisa, mengamati, serta mengendalikan kejadian pada dunia nyata saat sedang terjadi.
3. *Business software*, yaitu perangkat lunak yang digunakan untuk mengatur sistem keuangan, dimana pengelolaan uang menjadi hal yang sangat fundamental bagi sebagian masyarakat.
4. *Engineering and scientific software*, perangkat lunak ini berfungsi untuk membantu dalam pengembangan teknologi dan dalam kegiatan penelitian.

5. *Artificial intelegency software*, yakni aplikasi yang digunakan untuk membantu memecahkan sebuah masalah yang bersifat non algoritmik, yang mana tidak sesuai dengan perhitungan dan analisis secara langsung.
6. *Web based software*, merupakan perangkat lunak yang berfungsi untuk mengantarkan pengguna pada akses internet secara langsung.
7. *Personal computer software*, perangkat lunak yang terakhir ini merupakan perangkat pengguna resmi dan juga pribadi, yang telah banyak digunakan sejak dua dekade terakhir ini.

1.2.3. Fungsi Aplikasi

Seperti yang sudah disinggung di atas, aplikasi mempunyai beberapa fungsi. Salah satunya adalah untuk mempermudah kegiatan penggunaannya. Secara spesifik aplikasi mempunyai beragam fungsi, baik di dunia pendidikan, industri dan manufaktur, bisnis dan perbankan, militer hingga kedokteran.

1. Fungsi Aplikasi dalam Pendidikan

Fungsi aplikasi dalam dunia pendidikan tentu saja untuk menambah pengetahuan. Selain itu aplikasi juga bisa bermanfaat untuk mempermudah proses belajar mengajar antara guru dengan siswanya.

2. Tenaga pengajar bisa memanfaatkan aplikasi sebagai bahan untuk pembelajaran. Contohnya mencari bahan untuk belajar, hingga informasi lain seperti melihat data nilai siswa, jadwal pembelajaran, beasiswa dan lain-lain. Dengan aplikasi, maka seluruh kegiatan pembelajaran bisa dilakukan dengan lebih mudah dan maksimal.

3. Industri dan Manufaktur

Penggunaan aplikasi pada komputer ini tentu saja dapat digunakan untuk mengkoordinasi mengenai penggunaan mesin dan mengontrolnya. Hal ini dilakukan agar hasil yang diberikan memiliki kualitas dan kuantitas yang lebih banyak. Dengan demikian akan mendapatkan kenyamanan yang diberikan. Untuk mendapatkan mengenai proses yang demikian tentu saja akan meningkatkan keuntungan yang diberikan. Sehingga akan meningkatkan hasil yang diinginkan dengan memberikan format khusus tanpa mengeluarkan banyak biaya yang besar.

4. Bisnis dan Perbankan

Didalam suatu bisnis, perlu menghitung besarnya keuntungan yang diperoleh. Jika keuntungan dihitung secara manual maka akan membutuhkan waktu yang lama. Oleh karena itu diperlukannya aplikasi untuk menghitung besar keuntungan tersebut. Dalam menjalani bisnis tentu saja aplikasi komputer sangat digunakan untuk menghitung keuntungan, proses produksi dan masih banyak lainnya. Bahkan apabila dalam bidang bisnis tentu saja aplikasi ini akan memberikan lapangan pekerjaan baru yang menghasilkan pundi-pundi keuntungan. Sedangkan untuk perbankan tentu saja akan membantu dalam memudahkan perhitungan prosentase bunga, layanan keuangan dan masih banyak lainnya. Dengan demikian hasilnya menjadi semakin maksimal dirasakan untuk mendapatkan kenyamanan dalam mengatur segala kepentingan yang dimiliki.

5. Militer

Aplikasi juga membantu dalam kemampuan untuk mempertahankan negara dengan digunakannya untuk keperluan militer. Tentu saja

akan memudahkan dalam mengembangkan dalam mengontrol pesawat, perhitungan ketinggian, mengontrol *navigasi*, peluru selam dan masih banyak lainnya. Dengan keamanan dan kecanggihan militer yang demikian. Tentu saja hasilnya yang didapatkan menjadi semakin optimal sesuai dengan keinginan kita. Usahakan dalam menggunakan dalam militer pastikan penggunaannya secara bijaksana. Tujuannya tentu saja agar hasilnya menjadi semakin bermanfaat untuk mendapatkan hasil yang sesuai dengan sasaran dan keinginan.

6. Kedokteran

Siapa sangka adanya aplikasi yang dikembangkan khusus untuk bidang kedokteran. Misalnya saja, adanya aplikasi untuk mendiagnosa penyakit, menawarkan perawatan rutin, bahkan sampai meracik obat. Dengan demikian, kita dapat mengecek diri sendiri melalui aplikasi tersebut.

Hal ini karena dalam bidang kedokteran aplikasi ini akan digunakan dalam mendiagnosa sebuah penyakit, memilih obat yang dapat menyembuhkan dan meracik obat tersebut. Tentu saja dengan adanya bantuan yang demikian hasilnya menjadi semakin maksimal sesuai dengan keinginan kita. Mulailah menggunakan aplikasi komputer yang dimiliki tersebut sesuai dengan keinginan yang diinginkan.

1.2.4. Ciri – Ciri Kualitas Aplikasi

Sejumlah ciri-ciri yang menandakan kualitas pada sebuah aplikasi, yaitu:

1. Aplikasi bisa memenuhi kebutuhan user.
2. Merespon sebuah intruksi seacara cepat.
3. Bisa berjalan di beragam *platform*.
4. Bisa membutuhkan *resource* (prosesor dan media penyimpanan) yang rendah.

1.2.5. Contoh Aplikasi



Gambar 1.3 Contoh Aplikasi

Berikut ini adalah beberapa contoh aplikasi yang banyak digunakan:

1. *Adobe Reader* yaitu aplikasi yang berfungsi untuk membaca dokumen, seperti .doc, .docx, dan .pdf.
2. *CCleaner* yaitu aplikasi yang berfungsi untuk membersihkan *junk file* atau *file* sampah dari perangkat komputer, *handphone* sehingga dapat meningkatkan kinerja perangkat tersebut.

3. *Game* yaitu aplikasi yang digunakan sebagai media hiburan untuk bermain *game*.
4. *Google Chrome, Mozilla Firefox* yaitu aplikasi yang berfungsi untuk menjelajahi internet.
5. *Instagram* yaitu aplikasi yang digunakan untuk membagikan foto atau video kepada orang lain.
6. *Internet Download Manager* yaitu aplikasi yang digunakan untuk mempercepat proses *download*.
7. *Microsoft Access, Paradox* yaitu aplikasi yang digunakan untuk menyimpan, mengolah data dalam jumlah yang besar.
8. *Microsoft Word* yaitu aplikasi yang digunakan untuk mengolah kata, seperti mengetik dokumen, membaca dokumen dan meng-*edit* dokumen.
9. *Microsoft Excel* yaitu aplikasi ini digunakan untuk mengolah angka dan tabel untuk dihitung.
10. *Microsoft Powerpoint* yaitu aplikasi yang digunakan untuk membuat dan menampilkan data yang bersifat presentatif.
11. *Notepad* yaitu dahulu *notepad* berfungsi sebagai catatan kecil. Namun sekarang, *notepad* digunakan para pemrogram untuk menuliskan *script* dan menyimpannya dengan berbagai ekstensi.
12. *Smadav, Avast* yaitu aplikasi yang digunakan untuk melindungi perangkat dari *malware* dan virus.
13. *Shareit* yaitu aplikasi yang dapat digunakan untuk melakukan transfer data ke perangkat lain.
14. *SPSS* yaitu aplikasi yang digunakan untuk mengolah data statistik.
15. *Adobe Photoshop, Corel Draw* yaitu aplikasi yang berfungsi untuk mengolah gambar.

16. *Sound Recorder* yaitu aplikasi yang digunakan untuk merekam suara.
17. *WhatsApp* yaitu aplikasi yang digunakan untuk berinteraksi dengan orang lain.
18. *Winrar* yaitu aplikasi yang digunakan untuk mengompres ukuran *file*.
19. *Winamp, GOM Player, Windows Media Player* yaitu aplikasi yang digunakan memutar lagu atau video dengan berbagai *format*.

1.3 Pengantar Manajemen

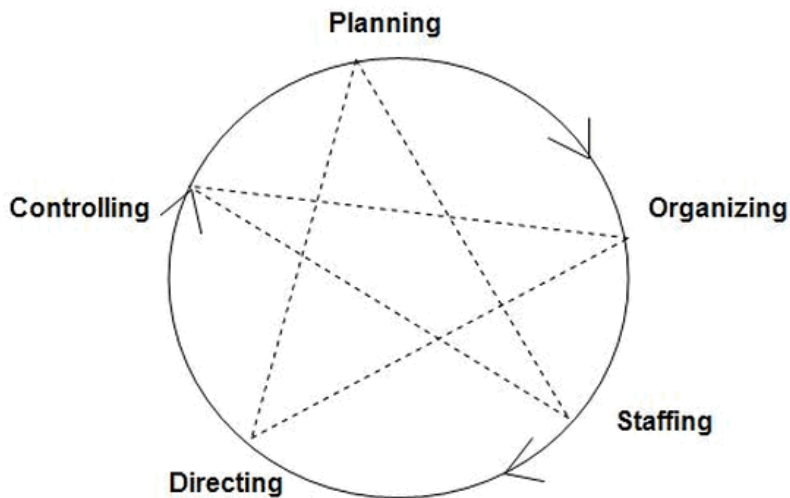
Manajemen adalah sebuah proses untuk mengatur suatu hal yang dilakukan oleh sekelompok organisasi atau orang untuk mencapai tujuannya tersebut dengan cara bekerjasama memanfaatkan sumber daya yang dimilikinya. Secara Etimologi, kata manajemen diambil dari bahasa Perancis kuno, yaitu “*management*” yang artinya berarti seni dalam mengatur dan melaksanakan. Manajemen juga dapat didefinisikan sebagai perencanaan, pengorganisasian, pengkoordinasian serta pengontrolan sumber daya untuk mencapai sasaran yang lebih efektif. Efektif yang dimaksudkan adalah untuk mencapai tujuan sesuai dengan perencanaan yang telah diharapkan.

1.3.2. Tujuan Manajemen

Pada dasarnya manajemen sudah pasti memiliki tujuan, adanya sebuah pola tentunya pasti ada tujuan yang ditujunya. Adapun tujuan manajemen yang paling umum adalah untuk memberi pola dalam menyelesaikan sebuah pekerjaan. Namun tidak hanya itu ada beberapa tujuan manajemen yang lain, nah berikut adalah penjelasannya:

1. Dapat menciptakan sebuah strategi yang efektif dan efisien dalam mencapai tujuan yang sudah ditentukan.
2. Mengkaji dan mengevaluasi sistem kerja, yang mana dengan tujuan menyesuaikan hal-hal yang terjadi diluar strategi yang sudah ditetapkan.
3. Melakukan tinjauan mengenai kekuatan, peluang serta kelemahan dan ancaman yang ada.
4. Mengatur semua sektor perusahaan agar mencapai profit, hal ini mengatur keuangan dan menjaga keadaan personal baik dari kesehatan maupun emosinya.

1.3.3. Fungsi Manajemen



Gambar 1.4 Diagram Fungsi Manajemen

Terdapat beberapa penjelasan mengenai fungsi manajemen, diantaranya sebagai berikut:

1. Perencanaan (*Planning*)

Dalam menjalankan manajemen, perencanaan menjadi hal yang sangat penting. Karena dalam perencanaan akan menentukan target untuk proses kedepannya. Manajemen dalam perusahaan akan dikelola oleh seorang yang bernama manajer. Manajer ini akan mengevaluasi tindakan yang dilakukan dan manajer juga akan menindaklanjuti perencanaan. Pentingnya perencanaan untuk menentukan tujuan serta untuk menentukan upaya yang tepat agar bisa mencapai tujuan.

Perencanaan sangat penting untuk seorang manajer, karena adanya perencanaan manajer dapat menindaklanjuti proses dengan mudah dalam mencapai tujuan. Perencanaan tidak hanya mencakup tujuan jangka panjang, namun mencakup juga untuk jangka menengah dan jangka pendek. Sebuah proses untuk mencapai tujuan tidak akan berkembang jika tanpa adanya perencanaan yang tepat. Dan juga dengan tanpa adanya perencanaan yang tepat, bisa berjalan tidak sesuai yang diinginkan. Ini harus dihindari, karena jika terjadi dalam waktu yang lama maka suatu tujuan tersebut akan melakukan kegagalan.

Manfaat Perencanaan untuk sebuah organisasi saat menentukan. Tanpa perencanaan yang baik, maka operasi organisasi mengalami hambatan. Perencanaan yang baik memberikan manfaat. Manfaat perencanaan adalah sebagai berikut :

- a. Mengidentifikasi peluang masa depan
- b. Mengembangkan langkah-langkah yang strategis

- c. Sebagai pedoman dalam pelaksanaan kegiatan
- d. Mengidentifikasi dan menghindarkan permasalahan yang timbul di masa yang akan datang.
- e. Dengan mudah melakukan pengawasan.

Tahap perencanaan ada empat tahap yang harus dilalui dalam proses perencanaan adalah sebagai berikut:

- a. Menetapkan tujuan yaitu perencanaan dimulai dengan keputusan tentang keinginan kebutuhan organisasi atau kelompok kerja
- b. Merumuskan keadaan saat ini yaitu pemahaman akan posisi perusahaan, maka dapat diperkirakan untuk masa depan.
- c. Mengidentifikasi kemudahan dan hambatan, kemudahan, hambatan, kekuatan, dan kelemahan dari organisasi perlu diidentifikasi untuk mengukur kemampuan organisasi dalam mencapai tujuan.
- d. Mengembangkan rencana untuk pencapaian tujuan, tahap terakhir dari proses perencanaan meliputi pengembangan berbagai alternatif untuk mencapai tujuan, penilaian alternatif, dan pengambilan keputusan untuk menentukan pilihan yang terbaik diantara berbagai alternatif yang ada.

2. Pengorganisasian (*Organizing*)

Selain perencanaan yang sudah dijelaskan diatas, pengorganisasian juga diperlukan dalam sebuah proses manajemen. Pengorganisasian adalah salah satu bagian kerja atau kegiatan menjadi beberapa bagian atau kegiatan. Tujuan dari pengorganisasian ini yaitu untuk mempermudah seseorang dalam

melakukan pengawasan yang lebih efektif. Selain itu, pengorganisasian digunakan untuk menentukan sumber daya yang dibutuhkan untuk setiap kegiatannya. Dalam pelaksanaan pengorganisasian, maka terasa lebih mudah jika terdapat pembagian tugas. Dengan menentukan beberapa hal tersebut, maka tujuan akan dengan mudah dicapai melalui proses yang lebih terorganisasi atau terstruktur.

Fungsi manajemen pengorganisasian ada tiga aspek dari fungsi manajemen mengenai pengorganisasian yaitu sebagai berikut:

- a. Menetapkan struktur organisasi
- b. Mendelegasikan wewenang
- c. Memantapkan hubungan

Ada beberapa bentuk organisasi, diantaranya sebagai berikut:

- a. Organisasi Garis.

Organisasi garis mempunyai struktur organisasi yang wewenang pimpinan langsung ditujukan kepada bawahan. Bawahan yang bertanggung jawab langsung pada atasan. Kelebihan organisasi garis antara lain sebagai berikut:

- Kesatuan komando terjamin berada pada satu tangan
- Proses pengambilan keputusan cepat
- Jumlah karyawan sedikit dan rasa solidaritas tinggi
- Pembagian kerja jelas dan mudah dilaksanakan

Kelemahan organisasi garis antara lain sebagai berikut:

- Maju mundurnya organisasi berada di tangan satu orang
- Kecenderungan pimpinan bertindak otoriter
- Sistem kerja pada tiap individual bersifat individual
- Kesempatan kerja untuk berkembang terbatas

b. Organisasi garis dan *staf*

Organisasi garis dan *staf* memberikan wewenang pada pimpinan untuk memberikan komando pada bawahan. Pimpinan dibantu oleh *staf* dalam pelaksanaan tugas. Bentuk organisasi tersebut banyak ditemukan diberbagai instansi atau perusahaan.

Kelebihan organisasi garis dan *staf* adalah sebagai berikut:

- Cocok diterapkan dalam organisasi yang sifatnya kompleks
- Memungkinkan adanya spesialisasi dengan berpedoman pada prinsip *the right man in the right place*.
- Keputusan yang diambil lebih rasional karena dipikirkan lebih dari satu orang.
- Adanya pembagian tugas secara lebih tegas antara pimpinan, staf, dan bawahan.
- Koordinasi dapat berjalan dengan baik karena setiap bidang telah memiliki tugas masing-masing.

Kelemahan organisasi garis dan staf adalah sebagai berikut:

- Pelaksanaan tugas sering bingung karena dimungkinkan terjadinya perintah lebih dari satu orang
- Karyawan cenderung tidak mengenal tanggung jawab
- Jumlah tenaga kerja yang diperlukan cukup banyak
- Solidaritas karyawan kurang

c. Organisasi Fungsional

Organisasi fungsional disusun berdasarkan sifat dan macam fungsi yang harus dilaksanakan.

Kelebihan organisasi fungsional adalah sebagai berikut:

- Kesimpangsiuran perintah dari atasan dapat dihindari karena ada pembagian tugas yang ada
- Penggunaan tenaga ahli dalam berbagai bidang sesuai dengan fungsinya
- Produktivitas relatif tinggi karena adanya spesialisasi pekerjaan
- Mudah dalam melakukan organisasi.

Kelemahan organisasi fungsional adalah sebagai berikut:

- Sulit mengadakan mutasi kerja tanpa terlebih dahulu mengadakan latihan
- Koordinasi secara menyeluruh sulit dilakukan
- Terjadi pengkotak-kotakan karyawan karena bidang tugas yang berlainan
- Kesimpangsiuran tugas masih mungkin terjadi karena perintah bisa datang lebih dari satu orang.

3. Penempatan (*Staffing*)

Penempatan atau *staffing* hamper mirip dengan pengorganisasian atau *organizing*. Namun, *staffing* dalam penggunaannya lebih luas dan lebih berfokus pada manajemen sumber daya manusia. Dalam *staffing* ini meliputi sumber daya perlengkapan, peralatan, dan inventaris yang ada pada sebuah organisasi.

Fungsi *staffing* mencakup kegiatan sebagai berikut:

- a. Perencanaan sumber daya manusia
- b. Rekrutmen karyawan
- c. Seleksi
- d. Pengenalan dan orientasi

- e. Penilaian dalam pelaksanaan kerja
- f. Pemberian balas jasa dan penghargaan
- g. Perencanaan dan pengembangan karier.

4. Pengarahan (*Directing*)

Pengarahan berguna untuk mengarahkan setiap anggota agar bisa mencapai tujuan yang diinginkan dengan prosedur manajerial yang sudah direncanakan. Selain itu, penggunaan pengarahan ini untuk mengarahkan setiap anggota yang mengalami masalah atau pekerjaan yang tidak sesuai dengan apa yang telah diinginkan. Dalam proses kegiatannya, seringkali terdapat sebuah rencana tidak bisa tercapai. Hal ini dikarenakan banyak kejadian tak terduga dan tidak bisa dihindari. Fungsi pengarahan ini juga dapat membantu setiap anggotanya agar bisa bekerja sesuai dengan jalur yang semestinya.

5. Pengawasan (*Controlling*)

Fungsi dari manajemen yang terakhir akan dibahas yaitu pengawasan atau *controlling*. Pengawasan berguna untuk mengevaluasi kinerja sumber daya perusahaan secara keseluruhan. Dalam pengawasan berfungsi untuk memastikan pekerjaan dikerjakan sesuai dengan yang telah direncanakan. Dalam proses pengawasan ini, akan ada orang yang mengawasi setiap adanya penyimpangan atau kesalahan dalam menjalankan tugas. Selain itu akan terus dilakukan evaluasi sehingga menjadi pembelajaran dalam perencanaan pada tahap berikutnya. Manajemen dilakukan dengan kerjasama secara tim untuk lebih mudah dalam proses pengerjaannya. Beberapa hal yang dilakukan dalam pengawasan manajemen diantaranya meliputi:

- a. Jalur (*routing*) yaitu seorang bisa memperkecil resiko kesalahan yang terjadi dengan menetapkan jalur.
- b. Dapat menetapkan waktu untuk melakukan pengawasan dalam kurun waktu tertentu.
- c. Dengan memerintah dan mendorong SDM yang ada akan lebih mudah dan cepat terselesaikan pekerjaan sesuai target yang diinginkan.
- d. Tindak lanjut (*Follow up*) bisa dilakukan untuk mengatasi permasalahan yang terjadi. Dalam melakukan evaluasi ini, harus bisa memberikan solusi terbaik sehingga dapat mencapai tujuan dan meminimalisir terjadinya kesalahan yang sama.

Langkah –langkah proses pengawasan sebagai berikut:

- a. Menetapkan standar dan metode untuk mengukur prestasi
- b. Mengukur prestasi kerja
- c. Menentukan apakah prestasi kerja sudah sesuai dengan standar atau belum
- d. Pengambilan tindakan koreksi bila pelaksanaannya menyimpang dari standar.

1.3.1. Macam-macam Manajemen

Terdapat banyak contoh manajemen yang bisa ditemukan dalam kehidupan sehari-hari. Berikut beberapa macam-macam manajemen, diantaranya:

- a. Manajemen Strategi
- b. Manajemen Administrasi Perkantoran
- c. Manajemen Biaya

- d. Manajemen Organisasi
- e. Manajemen Personalia dan Administrasi
- f. Manajemen Perusahaan
- g. Manajemen Pemasaran
- h. Manajemen Produksi
- i. Manajemen Keuangan
- j. Manajemen Waktu
- k. Manajemen Organisasi
- l. Manajemen Komunikasi
- m. Manajemen Pendidikan
- n. Manajemen Konstruksi
- o. Manajemen Agribisnis
- p. Manajemen Stress
- q. Manajemen Sumber Daya Manusia
- r. Manajemen Risiko
- s. Manajemen Rantai Pasokan
- t. Manajemen Keselamatan dan Kesehatan Kerja
- u. Manajemen Hubungan Masyarakat (Humas)

1.3.2. Unsur-Unsur Manajemen

Unsur-unsur manajemen sangat dibutuhkan dalam membentuk sistem manajerial. Unsur-unsur manajemen meliputi manusia, uang, bahan, mesin, metode dan pasar. Semua unsur tersebut harus saling melengkapi agar bisa mencapai tujuan. Lebih jelasnya berikut ini unsur-unsur dalam manajemen, yaitu:

- a. Manusia (*Human*)

Dalam manajemen, faktor yang paling penting yaitu manusia. Karena manusia yang paling berperan penting dalam membuat perencanaan dan melakukan proses untuk mencapai tujuan. Sehingga, tanpa adanya manusia, maka proses kerja tidak akan terjadi.

b. Uang (*Money*)

Selain manusia, uang juga sangat dibutuhkan dalam kegiatan di sebuah perusahaan. Uang bisa menjadi alat dalam mencapai tujuan setiap pekerjaan. Namun, penggunaannya harus diperhitungkan secara rasional agar tidak terjadi kerugian. Unsur manajemen yang satu ini digunakan sebagai biaya operasional. Dengan kata lain, uang ini digunakan untuk menggaji pegawai, pembelian peralatan kantor dan kebutuhan lainnya.

c. Bahan (*Materials*)

Faktor penting lainnya dalam manajemen yaitu material atau bahan. Dikatakan penting, karena dengan adanya bahan, maka tujuan akan bisa tercapai dengan mudah.

d. Mesin (*Machines*)

Selain bahan, mesin juga tak kalah penting. Dengan adanya mesin, tujuan perusahaan akan mudah dicapai, karena pekerjaan yang sulit akan terasa lebih mudah dengan menggunakan mesin.

e. Metode (*Methods*)

Metode yang tepat menjadikan proses pelaksanaan kerja berjalan secara efektif dan efisien. Meski begitu, metode yang digunakan tetap harus memperhatikan sasaran, fasilitas, waktu, uang dan kegiatan lainnya. Metode yang digunakan dalam perusahaan harus bisa dipahami oleh setiap anggotanya.

f. Pasar (*Market*)

Unsur terakhir dalam manajemen yaitu pasar. Unsur ini juga sangat penting bagi perusahaan, karena jika barang yang diproduksi tidak laku di pasar, maka akan mengalami kerugian. Maka dari itu, penting sekali menawarkan produk yang sesuai dengan minat konsumen dengan kualitas terbaiknya. Sehingga, kualitas barang yang dipasarkan sangat erat hubungannya dengan proses pemasaran.

1.3.3. Tingkatan Manajemen



Gambar 1.5 Tingkatan Manajemen

Manajer dapat diartikan sebagai anggota organisasi yang melaksanakan fungsi manajemen, dengan cara mengkoordinasikan dan mengawasi pekerjaan pekerja-pekerja lainnya sehingga dapat mencapai tujuan organisasi yang diharapkan. Secara tradisional, tingkatan manajemen atau manajer dapat dibagi dalam tiga klasifikasi, yaitu:

1. *First-line* Manajer

Merupakan tingkatan paling bawah dalam klasifikasi manajemen. Pada tingkatan manajemen ini, manajer memiliki tugas untuk

mengelola pekerjaan dari pekerja-pekerja diluar tingkatan manajerial, yang meliputi orang-orang yang melakukan pekerjaan di bidang produksi dan pelayanan pada pelanggan. Selain itu *first-line* manajer juga harus mempertanggungjawabkan pekerjaan karyawan-karyawan yang dipimpinnya kepada *middle-line* manajer.

2. *Middle-line* Manajer

Merupakan tingkatan dalam klasifikasi manajemen yang menjembatani antara tingkatan *first-line* manajer dengan top manajer. Selain itu, *middle-line* manajer bertugas mengelola pekerjaan *first-line* manajer dan mempertanggungjawabkannya kepada top manajer.

3. Top Manajer

Merupakan tingkatan manajemen tertinggi dalam struktur organisasi, dan memiliki tanggung jawab dalam membuat keputusan organisasi serta menyusun rencana dan tujuan yang akan dicapai oleh perusahaan.

1.3.4. Keahlian Manajemen

Semakin tinggi tingkatan manajer, maka *output* atau hasil yang diharapkan dari peran yang dimilikinya lebih berupa ide, konsep dan perencanaan, sementara semakin ke bawah, *output* yang diharapkan umumnya lebih berupa tindakan atau hal-hal yang bersifat teknis yang diperlukan untuk memastikan jalannya kegiatan operasional organisasi atau perusahaan. Secara lebih lengkap, keahlian yang dibutuhkan terbagi menjadi tiga, yaitu:

1. *Technical skills* adalah pengetahuan mengenai teknik dan *skill* yang dibutuhkan untuk melaksanakan pekerjaan.

2. *Human skills* adalah pengetahuan dan kemampuan untuk bekerja dengan baik dengan semua orang, serta membina hubungan dengan semua lapisan organisasi.
3. *Conceptual skills*, adalah kemampuan untuk mengungkapkan gagasan dan konsep mengenai situasi abstrak dan kompleks yang harus dihadapi oleh organisasi sebagai satu kesatuan.

1.3.5. Peran Manajemen

Peran manajemen pada dasarnya dapat diartikan sebagai tindakan atau kegiatan yang harus dilakukan oleh manajer dalam melaksanakan tugasnya untuk memastikan bahwa tujuan perusahaan atau organisasi dapat tercapai. Secara umum peran manajemen dapat dibagi menjadi tiga, yaitu:

1. Peran interpersonal

Peran interpersonal mencakup hubungan antara manajer dengan orang-orang disekitarnya dalam tujuan untuk melakukan pengawasan dan pengarahan tugas. Peran ini dibagi lagi menjadi tiga yaitu:

- a. Pemimpin simbol atau *figure head* yaitu peran manajer sebagai simbol dalam acara-acara yang berhubungan dengan perusahaan atau organisasi
- b. Pemimpin atau *leader* yaitu peran manajer sebagai pemimpin yang dapat memberikan contoh serta motivasi kepada orang-orang di sekitarnya, terutama ketika terjadi suatu masalah yang memerlukan penyelesaian secara segera.

- c. Penghubung atau *liaison* yaitu peran manajer sebagai penghubung antara pihak internal perusahaan atau organisasi dengan pihak eksternal yang berkepentingan.

2. Peran informasi

Peran informasi mencakup kemampuan dan ketersediaan akses bagi manajer untuk memperoleh informasi baik dari dalam maupun luar organisasi atau perusahaan, untuk kemudian diolah dan disalurkan pada orang-orang disekitarnya untuk mendukung kinerja masing-masing. Peran ini dapat dibagi menjadi tiga, yaitu:

- a. *Monitor* atau pemantau yaitu manajer berperan dalam melakukan pengawasan, pemantauan, serta pengumpulan kejadian atau informasi yang terjadi, baik secara langsung maupun tidak langsung
- b. Penyebar atau *disseminator* yaitu manajer berperan dalam menyebarkan informasi yang tepat kepada orang-orang di organisasi yang membutuhkan informasi tersebut untuk kepentingan pekerjaannya
- c. Juru bicara atau *spokesperson* yaitu manajer berperan sebagai juru bicara yang secara khusus mewakili bagian atau divisi yang dipimpinnya dan secara umum mewakili organisasi atau perusahaan tempatnya bekerja

3. Peran pengambil keputusan

Peran pengambil keputusan mencakup kemampuan manajer dalam mengambil keputusan, baik ditentukan secara individu maupun berdasarkan keputusan bersama, dengan mempertimbangkan tujuan organisasi atau perusahaan, strategi serta sumber daya yang dimiliki. Peran ini dapat dibagi menjadi empat, yaitu:

a. Kewirausahaan

Peran manajer dalam mencari peluang dalam organisasi dan lingkungan serta berinisiatif memprakarsai proyek-proyek yang dibutuhkan agar tercipta perubahan ke arah yang lebih baik

b. Penyelesaian masalah

Peran manajer untuk bertanggung jawab terhadap tindakan korektif yang diperlukan oleh organisasi atau perusahaan ketika terjadi gangguan atau kesalahan yang penting dan tidak terduga yang berpengaruh terhadap kinerja perusahaan atau organisasi secara keseluruhan

c. Pengalokasian sumber daya

Peran manajer dalam membuat atau menyetujui keputusan-keputusan organisasi atau perusahaan terhadap pengalokasian sumber daya yang dimiliki agar dapat mencapai tujuan yang diharapkan secara optimal, dengan mempertimbangkan prinsip pengalokasian secara efektif dan efisien

d. Negosiator

Peran manajer dalam melakukan negosiasi yang diperlukan untuk memastikan rencana dan strategi yang ditentukan dalam proses perencanaan dapat dilaksanakan dengan mempertimbangkan kepentingan pihak-pihak yang bersangkutan.

1.3.6. Prinsip Manajemen Menurut Henri Fayol



Gambar 1.6 Henri Fayol

Prinsip manajemen pada dasarnya merupakan inti dari keberhasilan manajemen untuk mencapai tujuan yang diharapkan. Henry Fayol mengungkapkan bahwa terdapat 14 prinsip manajemen, yaitu:

1. Pembagian kerja (*division of work*), disesuaikan dengan keahlian yang dimiliki secara objektif sehingga pelaksanaan tugas dapat berjalan dengan efektif. Dalam hal ini penempatan karyawan sebaiknya menggunakan prinsip *right man in the right place*.
2. Wewenang dan tanggung jawab (*authority and responsibility*), tiap karyawan memiliki wewenang tertentu dalam melaksanakan pekerjaannya, yang juga diimbangi dengan tanggung jawab dari setiap pekerjaan yang dilakukan
3. Disiplin, tiap karyawan harus memiliki disiplin diri dalam melaksanakan pekerjaannya dengan penuh tanggung jawab

4. Kesatuan perintah (*unity of command*) , perintah yang diterima harus memiliki tujuan yang sama meski berasal dari orang yang berbeda, agar tidak terjadi kebingungan dalam pelaksanaan pekerjaan
5. Kesatuan pengarahan (*unity of direction*), karyawan memerlukan pengarahan yang tepat agar dapat melaksanakan tugasnya dengan baik dan tepat sasaran
6. Mengutamakan kepentingan organisasi di atas kepentingan sendiri, karyawan harus menyadari bahwa keberhasilan organisasi atau perusahaan akan berpengaruh pula terhadap dirinya sehingga mengusahakan dan memberikan yang terbaik untuk kepentingan organisasi
7. Penggajian pegawai, kompensasi yang sesuai akan memberikan kedisiplinan dan semangat bekerja yang tinggi bagi masing-masing karyawan
8. Pemusatan (*centralization*), tanggung jawab akhir dari suatu organisasi atau perusahaan berada pada tingkatan manajer puncak (top manajer) yang memiliki wewenang tertinggi.
9. Hirarki, adanya pembagian kerja akan menimbulkan hirarki dalam suatu susunan organisasi atau perusahaan
10. Ketertiban (*order*) , ketertiban dalam pekerjaan sangat penting untuk menghindari kekacauan yang dapat menghambat pelaksanaan operasional perusahaan atau organisasi
11. Keadilan dan kejujuran , terkait dengan moral karyawan yang harus dimiliki oleh semua lapisan organisasi atau perusahaan

12. Stabilitas kondisi karyawan, kestabilan kondisi perlu dijaga agar tidak mengganggu performa karyawan yang pada akhirnya mengambat pencapaian tujuan
13. Prakarsa (*initiative*), prakarsa timbul ketika seseorang berusaha menyelesaikan hal-hal yang semestinya dilakukan agar memberikan hasil yang maksimal
14. Semangat kesatuan dan semangat korps (*espirts de corps*), rasa senasib sepenanggungan akan memberikan semangat untuk bekerja sama dalam mencapai tujuan yang diharapkan.

1.3.7. Bidang-Bidang Manajemen

1. Manajemen Produksi

Manajemen produksi merupakan kegiatan untuk mengoordinasikan penggunaan sumber daya (sumber daya, alat, sumber daya manusia, sumber daya dana) secara efisien dan efektif untuk menambah kegunaan suatu barang dan jasa. Tujuan utama dari manajemen produksi adalah menciptakan nilai tambah pada perusahaan demi kepuasan konsumen. Kegiatan manajemen produksi adalah sebagai berikut:

a. Perencanaan

Perencanaan merupakan langkah awal sebelum memproduksi suatu barang perencanaan produksi meliputi keputusan yang mencakup jenis barang yang diproduksi, jumlah barang yang akan diproduksi, desain produksi, bahan baku yang dibutuhkan dan cara pengolahan.

b. Pengendalian Produksi

Pengendalian produksi merupakan rangkaian prosedur yang diarahkan pada semua elemen dalam proses produksi (pekerja, materi, peralatan, dan material) sehingga memberikan hasil dengan ongkos terendah dalam waktu tercepat

c. Pengawasan Produksi

Fungsi yang digunakan untuk menjamin terlaksananya kegiatan sesuai dengan yang direncanakan, yaitu memproduksi dengan cara yang terbaik dan biaya serendah-rendahnya, serta tepat waktu.

2. Manajemen Pemasaran

Manajemen pemasaran mencakup kegiatan perpindahan barang atau jasa dari produsen ke konsumen atau semua kegiatan yang berhubungan dengan arus barang dan atau jasa dari produsen ke konsumen. Ada delapan fungsi pemasaran, yaitu penjualan, pembelian, pengangkutan, penyimpanan, pembelian, penanggungan resiko, standardisasi, dan grading serta pengumpulan informasi pasar. Kegiatan manajemen pemasaran yang mencakup sebagai berikut:

a. Riset Pasar

Merupakan penelitian yang dilakukan perusahaan untuk mengidentifikasi kehendak dan keinginan dari konsumen, sehingga perusahaan dapat menentukan produk yang dapat memenuhi kebutuhannya.

b. Segmen Pasar

Segmen pasar merupakan kegiatan membagi suatu pasar ke dalam kelompok-kelompok yang berbeda, dan masing-masing kelompok terdiri dari kelompok yang mempunyai ciri atau

sifat yang hampir sama. Dengan melaksanakan segmentasi pasar, kegiatan pemasaran dapat lebih terarah, efektif, dan efisien.

c. Mempromosikan Produk

Ada beberapa langkah kegiatan promosi yang dapat dilakukan, yaitu periklanan (promosi melalui media massa, reklame, atau *billboard*), *personal selling* (promosi secara lisan oleh perusahaan), promosi penjualan (kegiatan pemasaran yang merangsang pembelian produk oleh konsumen), dan publisitas (merupakan rangsangan untuk meningkatkan permintaan terhadap suatu produk melalui media publisitas seperti radio, televisi, dan pertunjukan).

3. Manajemen Keuangan

Manajemen keuangan merupakan pengelolaan aspek keuangan yang digunakan untuk berbagai penggunaan bisnis, dan berhubungan dengan kombinasi jenis-jenis pembiayaan yang terbaik agar dicapai efisiensi dalam perusahaan. Aspek-aspek termasuk kegiatan manajemen keuangan adalah sebagai berikut:

- a. Merencanakan dan melaksanakan kerja sama dengan pihak terkait dalam mencari dana
- b. Mengoordinasikan keputusan keuangan menyangkut investasi baik sumber maupun penggunaan
- c. Berintegrasi dengan pihak lain agar perusahaan lebih efektif dan efisien dalam beroperasi
- d. Mengawasi keuangan dengan membuat laporan perusahaan.

5. Manajemen Personalia

Manajemen personalia merupakan suatu ilmu dan seni perencanaan, pengorganisasian, dan pengawasan terhadap personalia sehingga efektivitas dan efisien personalia dapat ditingkatkan secara maksimal. Kegiatan manajemen personalia adalah sebagai berikut:

- a. Pengadaan tenaga kerja (penentuan jumlah dan kualifikasi tenaga kerja, penarikan tenaga kerja dan penempatan tenaga kerja.
- b. Pengembangan tenaga kerja (pendidikan dan pelatihan, mutasi jabatan, dan promosi jabatan)
- c. Pemanfaatan tenaga kerja (pemberhentian dan pemberian motivasi)

6. Manajemen Perkantoran

Kegiatan manajemen perkantoran adalah mengumpulkan, mencatat, menganalisis, dan melaporkan keuangan perusahaan sebagai bahan pengambilan keputusan. Tahapan dalam manajemen perkantoran adalah sebagai berikut:

- a. Pengumpulan data
- b. Pencatatan data
- c. Pengelompokan data
- d. Pelaporan
- e. Penafsiran data untuk memprediksi keadaan yang akan datang serta mengambil langkah yang perlu ditempuh.

BAB II

PENGENALAN SURAT, ARSIP, SISTEM KEARSIPAN *ALPHABETICAL FILING SYSTEM* DAN *CHRONOLOGY FILING SYSTEM*

Surat, arsip dan sistem kearsipan *Alphabetical Filling System* dan *Chronology Filling System* adalah beberapa istilah yang digunakan dalam penyusunan buku ini. Penulis akan memaparkan penjelasan mengenai ketiga istilah tersebut.

2.1. Pengantar Surat

Surat adalah media komunikasi tulis yang digunakan untuk menyampaikan informasi. Terdapat bagian-bagian surat yang biasanya ada dalam format penulisan surat terutama untuk surat resmi misalnya seperti kop surat, tanggal surat, nomor surat, alamat tujuan, isi surat, nama pengirim surat serta tembusan. Pengertian surat menurut KBBI adalah kertas dan sebagainya yang bertulis. Sementara definisi surat secara umum diartikan sebagai sebuah kertas yang berisikan informasi atau berita dari pengirim surat yang ditujukan kepada pembaca yang dituju. Surat digunakan untuk berkirim pesan dengan tujuan memberi informasi atau berita dari pengirim surat kepada pembaca surat yang dituju. Fungsi surat juga untuk menyampaikan pesan atau maksud tertentu dari pengirim pada yang dituju.

Ada banyak jenis-jenis surat dari mulai surat pribadi, surat dinas, surat lembaga, surat niaga, surat pemberitahuan atau bahkan surat kaleng. Surat menjadi media komunikasi formal yang biasa dilakukan oleh sebuah lembaga, instansi, organisasi dan perusahaan. Adapun surat elektronik disebut sebagai *email* atau *electronic mail*.

2.1.1. Fungsi Surat

Fungsi surat secara umum adalah sebagai alat komunikasi dalam bentuk tertulis. Berikut ini adalah beberapa fungsi surat diantaranya sebagai berikut:

- a. Sebagai sarana komunikasi untuk menyampaikan pemberitahuan, permohonan atau permintaan, perintah, pernyataan, laporan, dll.
- b. Sebagai alat bukti tertulis yang memiliki kekuatan hukum
- c. Sebagai alat untuk membantu mengingat suatu kegiatan
- d. Sebagai catatan atau dokumentasi untuk keperluan kegiatan lain
- e. Sebagai alat untuk mewakili pribadi atau lembaga pembuat surat yang menyampaikan pesan, informasi, atau misi, yang ingin disampaikan pada penerima

2.1.2. Jenis dan Macam-Macam Surat

Surat dapat dibagi dalam beberapa jenis berdasarkan tujuan, sifat, wujud, dan kriteria lainnya, berikut ini adalah macam-macam surat tersebut:

1. Jenis surat berdasarkan tujuannya:
 - a. Surat pemberitahuan yaitu biasanya berisi pemberitahuan kepada banyak pihak yang merupakan bagian dari suatu instansi
 - b. Surat perintah yaitu surat yang diberikan kepada pihak tertentu agar melakukan sesuatu yang tertera dalam surat tersebut
 - c. Surat permintaan yaitu surat yang isinya meminta informasi lebih detail mengenai barang atau jasa yang ditawarkan
 - d. Surat peringatan yaitu surat ini bertujuan untuk mengingatkan kesalahan seseorang di suatu instansi

- e. Surat panggilan yaitu bertujuan untuk memanggil pihak lain baik secara personal maupun organisasi
 - f. Surat susulan yaitu merupakan penegasan dari surat yang pernah dikirim sebelumnya
 - g. Surat keputusan yaitu surat yang berisi keputusan yang diambil oleh suatu organisasi atas dasar pertimbangan
 - h. Surat laporan yaitu surat yang berisi laporan kerja kepada orang lain
 - i. Surat penawaran yaitu umumnya surat ini berhubungan dengan jual-beli barang atau jasa yang isinya mengenai penawaran spesifikasi dan harga
2. Jenis surat berdasarkan wujudnya:
- a. Kartu pos yaitu surat yang isinya ditulis pada selembar kertas dan dikirim melalui pos. Biasanya berukuran 10×15 cm.
 - b. Warkat pos yaitu surat tertutup yang dibuat dari selembar kertas cetakan dan dapat dilipat seperti amplop.
 - c. Surat bersampul yaitu surat yang dibuat dalam beberapa helaian kertas terpisah dimana untuk mengirimnya harus menggunakan amplop
 - d. Memorandum dan Nota yaitu surat yang digunakan untuk keperluan internal suatu organisasi yang bertujuan untuk memberi/ meminta informasi antar pejabat kantor.
 - e. Telegram yaitu pesan singkat yang dikirim dari jarak jauh dalam waktu cepat dengan bantuan pesawat telegram.

3. Jenis surat menurut sifat isi dan asalnya:
 - a. Surat dinas yaitu digunakan untuk kepentingan tugas dinas oleh instansi-instansi tertentu, baik instansi sipil maupun militer.
 - b. Surat niaga atau bisnis yaitu surat yang dipakai untuk keperluan bisnis oleh lembaga-lembaga usaha.
 - c. Surat pribadi yaitu surat yang sifatnya pribadi yang dikirimkan kepada orang lain atau petugas lembaga.
4. Jenis surat menurut jumlah penerima:
 - a. Surat biasa yaitu umumnya ditujukan kepada satu orang atau petugas sebuah lembaga.
 - b. Surat edaran yaitu ditujukan kepada banyak orang atau banyak lembaga, bahkan kepada seluruh masyarakat.
 - c. Surat pengumuman yaitu umumnya ditujukan kepada sekelompok orang atau lembaga.
5. Jenis surat menurut kerahasiaannya:
 - a. Surat biasa yaitu surat yang isinya tidak terdapat hal-hal yang perlu dirahasiakan.
 - b. Surat rahasia yaitu surat yang isinya hanya boleh diketahui oleh orang tertentu yang berkepentingan saja.
 - c. Surat sangat rahasia yaitu surat yang kerahasiaan isinya dijaga sangat ketat agar tidak diketahui oleh orang lain
6. Jenis surat berdasarkan jangkauannya:
 - a. Surat internal yaitu surat yang asalnya dari dalam suatu kantor atau perusahaan dan ditujukan ke divisi lain di perusahaan yang sama.

- b. Surat eksternal yaitu surat yang asalnya dari luar kantor atau perusahaan yang masuk melalui pengiriman pos surat.

2.1.3. Makna Surat

Surat memiliki makna tertentu. Makna surat adalah jika seseorang sedang membutuhkan sesuatu atau memiliki informasi yang penting yang harus segera disampaikan, maka pihak tersebut akan memberikan tanda dengan mengirimkan maksud pada pihak lain. Hal ini menandakan bahwa ada informasi yang harus diketahui oleh pihak tersebut.

1. Surat Menyurat

Surat menyurat adalah surat yang digunakan untuk kegiatan yang dilakukan oleh satu pihak kepada pihak lain secara terus menerus, yang kegiatannya dilakukan dengan saling berkirim surat. Kegiatan dari surat menyurat ini juga memiliki istilah sendiri yaitu korespondensi. Selain itu ada juga surat masuk dan surat keluar. Apa itu surat masuk dan apa itu surat keluar. Berikut adalah penjelasan mengenai surat masuk dan juga surat keluar.

2. Surat Masuk

Surat masuk adalah surat yang diterima oleh suatu organisasi ataupun perusahaan yang dikirimkan maupun ditulis dari seseorang atau satu organisasi.

12. Surat Masuk

PT USAHA TIRTA JAYA

Jl. Soekarno Hatta No. 11

Palembang

Nomor : 100/PP/III/14

10 Maret 2014

PT Usaha Mandiri

Jl. Soedirman No.18

Palembang

Perihal : Permintaan Penawaran

Dengan hormat,

Berkenaan dengan tahun anggaran 2014 tersedia anggaran untuk pembelian komputer.

Dari salah seorang relasi, kami memperoleh keterangan bahwa perusahaan Saudara bergerak di bidang peralatan dan perlengkapan mesin kantor yang berkualitas baik.

Sehubungan dengan hal tersebut, kami meminta agar Saudara mengirimkan surat Penawaran mengenai peralatan dan perlengkapan komputer, disertai keterangan-keterangan sebagai berikut:

1. Syarat pembayaran.
2. Syarat penyerahan barang.
3. Potongan harga yang diberikan.

Di samping itu, kami minta dikirim leaflet, katalog dan brosur barang tersebut.

Atas perhatian saudara, kami ucapkan terima kasih.

Hormat kami,

Ocky Sulistianingsih

Direktur Utama

Gambar 1.7 Contoh Surat Masuk

3. Surat Keluar

Surat keluar adalah surat-surat yang sudah dikeluarkan oleh seseorang atau dibuat suatu organisasi maupun perusahaan untuk dikirimkan kepada pihak lain, baik dikirimkan oleh perseorangan maupun kelompok.

5. Surat Keluar

PT USAHA MANDIRI

Jl. Soedirman No.18

Palembang

10 April 2014

Nomor: 105/UM/III/2014

Perihal: Penagihan Faktur No.21/III/14

PT Usaha Tirta Jaya

Jl. Soekarno Hatta No 11

Palembang

Dengan hormat,

Menurut catatan kami ternyata Saudara belum melunasi faktur kami No. 21/III/14, tanggal 20 Maret 2014, sebesar Rp.1.500.000 (satu juta lima ratus), yang seharusnya saudara selesaikan tanggal 27 Maret 2014.

Mengingat pembayaran sudah lewat jangka waktu pembayaran. Besar harapan kami, Saudara segera membayar faktur tersebut melalui cabang BRI setempat.

Atas perhatian Saudara, kami ucapkan terima kasih.

Hormat kami,

Ocky Sulistianingsih

Direktur Utama

Gambar 1.8 Contoh Surat Keluar

2.2. Pengantar Arsip

Menurut pasal 1 angka 2 Undang-Undang Nomor 43 Tahun 2009 tentang kearsipan, pengertian mengenai arsip atau *kintaka* adalah rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan komunikasi yang dibuat dan diterima oleh lembaga negara, pemerintahan daerah, lembaga pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, berbangsa, dan bernegara. Arsip sangat berbeda dengan bahan pustaka yang terdapat dalam perpustakaan. Arsip mempunyai ciri khusus yang berbeda dengan bahan pustaka diantaranya adalah arsip harus autentik dan terpercaya sebagai alat bukti yang sah, informasinya utuh, dan berdasarkan asas asal usul (*principle of provenance*) dan aturan asli (*principle of original order*). Kata "arsip" merupakan kata serapan dari bahasa Belanda *archieff* yang pada gilirannya diserap dari bahasa Perancis *archives*. Pengucapan dan cara penulisan dalam bahasa Indonesia ini tampaknya berasal dari pelafalan bahasa Perancis.

Arsip merupakan sebuah “alat” untuk mendukung bisnis organisasi dan efisiensi administrasi. Apa saja yang diberikan dengan adanya arsip? Arsip memberikan:

1. Informasi yang berguna untuk perencanaan dan pengambilan keputusan
2. Bukti transaksi atau kegiatan sebagai bagian dari akuntabilitas organisasi

Untuk organisasi seperti perguruan tinggi, arsip mencatat atau mendokumentasikan apa yang sudah dilakukan (pembangunan, kerjasama, pengajaran, pendirian), mengapa kegiatan tersebut dilakukan, dan siapa saja yang telah terlibat dalam kegiatan universitas (mahasiswa, dosen, pegawai, peneliti). Arsip memberikan informasi berupa bukti dari sebuah proses komunikasi, pengambilan keputusan, dan sebuah kegiatan. Dalam jangka panjang, beberapa arsip akan disimpan sebagai memori universitas yang menyimpan informasi penting baik bagi organisasi maupun luar organisasi.



Gambar 1.9 Contoh Arsip

Arsip memberikan informasi yang spesifik tentang apa yang telah dilakukan, transaksi apa yang telah terjadi, di unit kerja anda. Mengelola arsip berarti mengelola informasi dan dokumen yang anda buat, sehingga apabila informasi atau dokumen tertentu dibutuhkan anda dengan mudah dan cepat menemukannya.

Arsip akan memudahkan dalam:

1. Menemukan *email*, dokumen atau informasi yang dibutuhkan
2. Menggunakan kembali suatu borang dari pekerjaan yang telah anda kerjakan dimasa lalu
3. Menentukan dokumen mana yang paling terbaru dari suatu revisi
4. Menelaah mengapa suatu kebijakan diambil pada masa yang telah lalu
5. Menjaga budaya, sosial, dan nilai sejarah pendidikan tinggi, membantu generasi masa depan mengerti sejarah sosial, budaya, dan personal.

2.2.1. Fungsi Arsip

Secara umum arsip memiliki fungsi untuk penunjang aktivitas administrasi, alat pengambil keputusan, bukti pertanggungjawaban, sumber informasi, dan wahana komunikasi. Selain itu memiliki fungsi primer dan sekunder:

1. Fungsi primer adalah nilai guna arsip yang didasarkan pada kepentingan pencipta arsip tersebut sebagai penunjang saat tugas sedang berlangsung maupun setelah kegiatan selesai, baik itu oleh lembaga atau instansi pemerintah, swasta, maupun perorangan. Nilai guna pada arsip primer meliputi administrasi, hukum, keuangan, ilmiah maupun teknologi.
2. ***Fungsi sekunder*** adalah nilai guna arsip yang didasarkan pada kegunaan bukan untuk pencipta arsip melainkan bagi kepentingan lembaga/instansi pemerintah, swasta, perorangan dan juga kepentingan umum lain sebagai bahan bukti dan bahan pertanggungjawaban. Nilai guna skunder meliputi nilai guna pembuktian dan penginformasian.

2.2.2. Peralatan Penyimpanan Arsip

1. Map Arsip atau *Folder*



Gambar 2.10 Contoh Map Arsip atau Folder

Map arsip atau *folder* adalah lipatan kertas atau plastik tebal untuk menyimpan arsip. Macam-macam map arsip atau *folder* meliputi:

- a. *Stofmap folio* (map berdaun)
- b. *Snelhechter* (map berpenjepit)
- c. *Brief Ordner* (map besar berpenjepit)
- d. *Portapel* (map bertali)
- e. *Hanging Folder* (map gantung)

2. Sekat Petunjuk atau *Guide*



Gambar 2.11 Contoh Sekat Petunjuk atau Guide

Sekat petunjuk atau *guide* adalah alat yang terbuat dari karton atau plastik tebal yang berfungsi sebagai penunjuk, pembatas atau penyangga deretan *folder*.

3. Almari Arsip atau *Filing Cabinet*



Gambar 2.12 Contoh Almari Arsip atau Filing Cabinet

Almari arsip atau *filing cabinet* adalah alat yang digunakan untuk menyimpan arsip dalam bentuk lemari yang terbuat dari kayu, alumunium atau besi baja tahan karat atau api.

4. Rak Arsip



Gambar 2.13 Contoh Rak Arsip

Rak arsip adalah almari tanpa daun pintu atau dinding pembatas untuk menyimpan arsip yang terlebih dahulu dimasukkan dalam *ordner* atau kotak arsip.

5. Kotak atau Almari Kartu atau *Card Cabinet*



Gambar 2.14 Contoh Kotak atau Almari Kartu atau Card Cabinet

Kotak atau almari kartu atau *card cabinet* adalah alat yang digunakan untuk menyimpan kartu kendali, kartu indeks dan kartu-kartu lain yang penyimpanannya tidak boleh sembarangan agar mudah untuk ditemukan kembali.

6. Berkas Peringatan atau *Tickler File*



Gambar 2.15 Contoh Berkas Peringatan atau Trickler File

Berkas peringatan atau *tickler file* adalah alat yang digunakan untuk menyimpan arsip atau kartu-kartu yang memiliki tanggal jatuh tempo.

7. Kotak Arsip atau File Box



Gambar 2.16 Contoh Kotak Arsip atau File Box

Kotak arsip atau *file box* adalah alat yang digunakan untuk menyimpan arsip yang terlebih dahulu dimasukkan ke dalam *folder*/ atau map arsip.

8. Rak Sortir



2.17 Contoh Rak Sortir

Rak sortir adalah alat yang digunakan untuk memisah-misahkan surat yang diterima, diproses, dikirimkan atau untuk menggolong-golongkan arsip sebelum disimpan.

2.2.3. Tata Cara Penyimpanan Arsip

1. Horizontal Filing (Flat Filing)

Penyimpanan arsip dengan cara arsip dimasukkan dalam stofmap atau snelhechter kemudian ditumpuk ke atas dalam alamari arsip (disusun secara mendatar atau *horizontal* dari bawah ke atas).

2. Vertikal Filing

Penyimpanan arsip dengan cara arsip dimasukkan dalam folder atau map arsip kemudian diletakkan berdiri atau tegak memanjang (sisi panjang arsip sejajar dengan lipatan folder atau map) dan disusun berurutan dari depan ke belakang.

3. Lateral Filing

Penyimpanan arsip dengan cara arsip dimasukkan dalam *snelhechter* atau *brief ordner* kemudian diletakkan berdiri dengan punggung di depan.

2.2.4. Prosedur Penyimpanan Arsip

1. Meneliti dulu tanda pada lembar disposisi apakah surat tersebut sudah boleh untuk disimpan (meneliti tanda pelepas surat atau *release mark*). Tanda pelepas surat biasanya berupa disposisi dep. (*deponeren*) yang menunjukkan perintah untuk menyimpan surat.
2. Mengindeks atau memberi kode surat tersebut. Indeks atau kode surat dibuat sesuai sistem penyimpanan arsip yang dipergunakan dan dibuat untuk memudahkan penyimpanan dan penemuan kembali surat.
3. Menyortir atau memisah-misahkan surat sesuai dengan bagian, masalah atau tujuan surat. Kegiatan menyortir atau memisah-

misahkan surat sebelum disimpan biasanya dilakukan dengan menggunakan rak atau kotak sortir.

4. Menyimpan surat ke dalam map (*folder*). Penyimpanan surat ke dalam map atau *folder* dapat menggunakan *stofmap folio*, *snelhechter*, *brief ordner*, *portapel* atau *folder* gantung kemudian dimasukkan ke dalam almari arsip atau *filing cabinet* atau alat penyimpanan arsip yang lain.
5. Menata arsip dengan baik sesuai dengan sistem yang dipergunakan. Penyimpanan arsip dapat menggunakan sistem penyimpanan arsip sebagai berikut :
 - a. Sistem Abjad (*Alphabetic Filing System*)
 - b. Sistem Tanggal (*Chronological Filing System*)
 - c. Sistem Nomor (*Numeric Filing System*)
 - d. Sistem Wilayah (*Geographic Filing System*)
 - e. Sistem Subyek atau Pokok Masalah (*Subject Filing System*)

2.2.5. Sifat dan Karakter Arsip

Arsip memiliki sifat dan karakter untuk membedakan kualitas arsip, antara lain:

1. *Autentik* yaitu informasi melekat pada wujud aslinya seperti informasi mengenai waktu dan tempat arsip dibuat atau diterima, memiliki tujuan dan kegiatan, bukti kebijaksanaan dan organisasi penciptanya.
2. *Legal* yaitu dokumentasi untuk mendukung tugas dan kegiatan, memiliki status sebagai bahan bukti resmi bagi keputusan dan pelaksanaan kegiatan.

3. Unik karena tidak dibuat massal dan memiliki kronologi produk. Jika arsip diduplikasi (dibuat tembusan) akan memiliki arti yang berbeda untuk pelaksanaan kegiatan.
4. Terpercaya sehingga dapat dipergunakan sebagai bukti sah sebagai bahan pendukung pelaksanaan kegiatan

2.2.6. Kearsipan

Kearsipan (*filing*) adalah suatu proses kegiatan pengaturan arsip (*file*) mulai dari penciptaan, penerimaan, pencatatan, penyimpanan. Proses kearsipan menggunakan sistem tertentu dalam penyusunan, pemeliharaan arsip agar dapat ditemukan kembali dengan cepat dan tepat serta untuk pemusnahan arsip berdasarkan kriteria tertentu.

Ruang lingkup kegiatan kearsipan meliputi:

1. Penciptaan, penerimaan, pengumpulan arsip
2. Pengendalian, pemeliharaan dan perawatan arsip
3. Penyimpanan dan pemusnahan arsip

Perkembangan teknologi dan informasi saat ini dapat mengubah proses kearsipan dengan lebih praktis, cepat dan mudah. Arsip-arsip dapat disimpan dalam bentuk digital berupa mikro film, cd, dvd, hard disk dan sebagainya yang dapat menghemat ruang dan biaya. Apalagi telah hadir *cloud computing* yang memanfaatkan teknologi internet untuk penyimpanan *file* atau dokumen.

2.2.7. Manajemen Kearsipan

Manajemen kearsipan diartikan sebagai seni pengendalian dokumen berupa pengendalian penggunaan, pemeliharaan, perlindungan, dan penyimpanan arsip. Manajemen kearsipan juga dapat

diartikan sebagai pekerjaan pengurusan arsip yang meliputi pencatatan, pengendalian dan pendistribusian, penyimpanan, pemeliharaan, pengawasan, pemindahan dan pemusnahan. Manajemen kearsipan dibatasi dalam ruang lingkup aspek POAC (*Planning, Orgaizing, Actuating, dan Controlling*)

1. *Planning* (perencanaan) adalah kegiatan yang sangat penting karena tanpa adanya perencanaan maka kegiatan tidak akan berjalan baik. Kegiatan perencanaan dalam pengelolaan arsip meliputi rencana arsip yang akan dilakukan, alat media yang digunakan, cara pelayanan arsip, sampai bagaimana pemusnahan dan pelestarian arsip.
2. *Organizing* (pengorganisasian) merupakan tindak lanjut dari suatu perencanaan. Rencana yang sudah dibuat dikoordinasikan kepada anggota organisasi dan dibentuk suatu langkah konkrit untuk pelaksanaannya. Langkah-langkah yang perlu dilakukan dalam pengelolaan arsip, yaitu:
 - a. Pegawai cakap sesuai bidangnya
 - b. Keuangan mendukung rencana pengurusan arsip
 - c. Peralatan arsip yang memadai
 - d. Sistem atau metode penyimpanan arsip yang baik
 - e. Sistem penataan berkas yang disesuaikan dengan aktivitas manajemen
3. *Actuating* (pelaksanaan) yang meliputi kegiatan pelaksanaan langkah pengelolaan kearsipan mulai dari awal diterima arsip sampai pemusnahan atau pelestarian. Kegiatan ini juga mencakup pemeliharaan arsip dengan pengawasan.

4. *Controlling* (pengawasan) merupakan seluruh kegiatan pengawasan dari semua komponen manajemen kearsipan sehingga pelaksanaan manajemen kearsipan dapat berjalan sesuai standar tertentu, efektif, dan efisien.

2.2.8. Asas dalam Kearsipan

Berdasarkan Undang-Undang Kearsipan, penyelenggaraan dan pengelolaan kearsipan dilaksanakan dengan asas sebagai berikut:

1. Asas kepastian hukum, pelaksanaan kearsipan dilaksanakan berdasarkan landasan hukum dan selaras dengan aturan Undang-Undang, dan keadilan dalam kebijakan penyelenggara negara.
2. Asas keterpercayaan, kegiatan kearsipan harus asli dan terpercaya agar dapat digunakan sebagai bukti.
3. Asas keutuhan, penyelenggaraan kearsipan harus terhindar dari pengurangan, penambahan, dan pengubahan baik dari segi isi maupun fisik yang dapat mengurangi keterpercayaan arsip.
4. Asas asal-usul, arsip harus dikelola dalam satu kesatuan pencipta arsip dan tidak tercampur dengan pencipta yang lain.
5. Asas aturan asli, arsip harus tertata sesuai dengan pengaturan aslinya atau dengan aturan saat arsip masih digunakan untuk kegiatan pencipta arsip.
6. Asas keamanan, arsip harus terjamin keamanannya dari kemungkinan kebocoran dan penyalahgunaan informasi oleh pengguna yang tidak berkepentingan.
7. Asas keselamatan, arsip harus terjamin keselamatannya dari bahaya baik dari alam maupun perbuatan manusia.

8. Asas keprofesionalan, penyelenggaraan kearsipan harus dilaksanakan oleh pihak yang profesional dan berkompotensi dalam bidang kearsipan.
9. Asas koresponsifan, dalam penyelenggaraannya kearsipan harus tanggap atas masalah yang menyangkut kearsipan atau masalah lain yang berkaitan dengan kearsipan.
10. Asas keantisipatifan, penyelenggaraan kearsipan harus sadar atas perubahan dan perkembangan akan pentingnya arsip.
11. Asas kepartisipatifan, penyelenggaraan kearsipan harus memberikan kesempatan bagi masyarakat untuk berperan dalam bidang kearsipan.
12. Asas akuntabilitas, arsip harus mampu merefleksikan suatu kegiatan dan peristiwa dalam rekaman sebagai bahan akuntabilitas.
13. Asas kemanfaatan, harus memberi manfaat bagi kehidupan masyarakat, bangsa, dan negara.
14. Asas aksesibilitas, penyelenggaraan arsip harus memberi kemudahan bagi masyarakat untuk memanfaatkan arsip.
15. Asas kepentingan umum, penyelenggaraan kearsipan memperhatikan kepentingan umum dan menghindari adanya diskriminasi.

2.2.9. Arsip Elektronik (*E-Archives*)

Arsip elektronik adalah definisi lain dari *Elektronic Archive (e-Archive)* yang artinya yaitu suatu sistem atau tata cara mengumpulkan informasi dalam bentuk dokumen yang direkam dan disimpan memakai teknologi komputer berupa dokumen elektronik (*Document*

Management System atau *e-Documents*) yang bertujuan supaya dokumen mudah dilihat, dikelola, ditemukan dan dipakai kembali.

2.2.10. Komponen Arsip Elektronik

1. Kabinet *Virtual*

Kabinet *virtual* merupakan *database* yang meniru bentuk dari kabinet nyata yang dipakai pada sistem kearsipan konvensional, bedanya apabila dalam kabinet maya kemampuan dalam menampung datanya terbatas, sedangkan kabinet maya ini tidak terbatas, yang membatasi adalah kemampuan fisik hardisk dalam menyimpan data digital.

Ataupun atribut dalam kabinet virtual antara lain yaitu:

- a. Kode kabinet, dengan fungsi untuk pencatatan kode kabinet sesuai dengan aturan penulisan kode dalam perusahaan
- b. Nama kabinet, yang berfungsi sebagai pencatat nama kabinet seperti Surat Masuk, Surat Keluar dan sebagainya.
- c. Fungsi kabinet, yang berfungsi sebagai pencatat keterangan fungsi kabinet
- d. Lokasi, yang berfungsi sebagai pencatat lokasi kabinet

2. Map *Virtual*

Map *virtual* merupakan *database* yang atributnya seperti map yang sesungguhnya dalam sistem kearsipan konvensional. Tetapi tidak seperti pada map konvensional yang mempunyai kemampuan terbatas untuk penyimpanan dokumen, map virtual ini mempunyai kemampuan tidak terbatas dalam penyimpanan dokumen. Adapun atribut map virtual antara lain yaitu:

- a. Kode Map, mempunyai fungsi sebagai pencatat kode map sesuai dengan aturan penulisan kode dalam perusahaan
- b. Nama map, mempunyai fungsi sebagai pencatat nama map seperti contohnya bagian keuangan, bagian pemasaran dan lain sebagainya.
- c. Lokasi map
- d. Keterangan dan lain sebagainya.

2.2.11. Lembaran Arsip

Lembaran arsip yang tersimpan dalam map virtual, bisa dalam bentuk *file* dokumen atau gambar. *File* dokumen merupakan *file* yang dibuat dari *microsoft word*, *excel*, *power point* dan sebagainya. Sedangkan *file* gambar merupakan *file* dalam bentuk gambar sebagai hasil scanner atau import bitmap dari media lain. Beberapa atribut yang dicatat di dalam *database*-nya antara lain yaitu:

1. Kode arsip, mempunyai fungsi sebagai pencatat kode arsip sesuai dengan aturan penulisan kode arsip dalam perusahaan
2. Nama arsip, mempunyai fungsi sebagai pencatat nama yang menunjukkan isi detail dari arsip yang disimpan
3. Klasifikasi, mempunyai fungsi sebagai pencatat klasifikasi map seperti penawaran khusus, rahasia dan lain sebagainya.
4. Tanggal Arsip, mempunyai fungsi sebagai pencatat tanggal arsip tersebut dibuat
5. Tanggal diterima, mempunyai fungsi sebagai pencatat tanggal arsip tersebut diterima
6. Pengirim, mempunyai fungsi untuk mencatat pengirim arsip

7. Penerima, mempunyai fungsi sebagai pencatat bagian yang menerima arsip (tujuan arsip)
8. Gambar, mempunyai fungsi sebagai pencatat *file* arsip yang telah di *scanner* apabila ada.

2.3. Pengantar Sistem Kearsipan *Alphabetical Filling System* dan *Chronology Filling System*

Filing sistem adalah suatu rangkaian kerja yang teratur yang dapat dijadikan pedoman untuk penyimpanan arsip sehingga saat diperlukan dapat ditemukan kembali dengan cepat dan tepat. Kearsipan adalah semua kegiatan pengurusan arsip yang dimulai dari kegiatan penciptaan arsip, penyimpanan (*filling*) dan penemuan kembali (*finding*). Penyelamatan arsip (pengamanan, pemeliharaan, dan perawatan) dan penyusutan arsip (pemindahan, pemusnahan, dan penyerahan).

Tujuan *filing system*:

1. Menghemat Waktu

Dengan menggunakan *filing* sistem yang tepat, penyimpanan dan penemuan kembali arsip dapat dilakukan dengan mudah.

2. Menghemat Tenaga

Dalam kegiatan penyimpanan (*filling*) dan penemuan kembali (*finding*) arsip tidak terlalu banyak menimbulkan tenaga.

3. Menghemat Tempat

Dengan menggunakan sistem penyimpanan yang tepat penyimpanan arsip tidak membutuhkan ruangan yang luas dan peralatan yang banyak, karena arsip yang disimpan hanyalah arsip-arsip yang bernilai guna saja.

Ciri-Ciri *Filing System* yang baik:

1. Tidak memakan tempat, letak (*lay out*) dibuat seefektif dan seefisien mungkin.
2. Sederhana dan praktis, mudah dilaksanakan dan tidak berbelit-belit.
3. Mudah dicapai, penyimpanan surat atau warkat harus dapat dengan mudah diambil dan digapai.
4. Ekonomis, tidak berlebihan dalam pengeluaran biaya, perlengkapan, tenaga, dan cara pengerjaan.
5. Cocok dan tepat guna, disesuaikan dengan tujuan dan kepentingan.
6. Fleksibel, mudah dikembalikan apabila ada perluasan kerja dan mudah dilaksanakan.
7. Klasifikasi yang khusus, keanekaragaman arsip dapat menimbulkan kesulitan.
8. Aman, bebas kerusakan karena penyimpanan terpelihara dari gangguan serangga, rayap, air, debu, dan sebagainya.

2.3.1. *Alphabetical Filing System*

Alphabetical filing system adalah penyimpanan surat dengan menggunakan abjad sebagai kode dalam penyimpanannya. Dalam penyimpanannya dengan mengindeks (membagi nama-nama menjadi beberapa unit untuk dijadikan kode) nama pengirim atau juga nama instansi yang bersangkutan. Mengindeks merupakan hal yang sangat dibutuhkan dalam sistem ini, karena dari mengindeks tersebut suatu nama seseorang atau instansi akan dijadikan sebagai kode dalam penyimpanan maupun penemuan kembali arsip atau warkat.

Prosedur penyimpanan, pertama kita harus mengindeks nama pengirim ataupun nama instansi yang mengirim surat kepada kita, lalu

kita catat dalam kartu indeks dengan memasukan kode. Setelah itu masukan arsip atau surat tersebut sesuai dengan kode yang tertera dalam kartu indeks tersebut. Prosedur penemuan kembali, untuk penemuan surat yang telah di arsipkan setelah kita mengetahui surat jenis apa yang dibutuhkan, maka kita bisa langsung menuju kartu indeks untuk melihat kode surat tersebut. Setelah mengetahui kodenya kita bisa menuju laci, *guide* dan *folder* yang sesuai dengan kode tersebut.

2.3.2. Chronology Filing System

Chronology filing system adalah suatu sistem penyimpanan atau pencarian arsip dimana arsip-arsip disusun berdasarkan urutan waktu (dari tanggal, bulan, tahun yang tertera di surat) yang dijadikan pokok atau judul surat adalah tanggal, bulan, tahun dibuatnya surat tersebut. Syarat dokumen yang diarsip dengan sistem *chronology* adalah tanggalnya harus jelas.

Kelebihan menggunakan sistem *chronological system*:

1. Memberikan keterangan yang *real* untuk penyelesaian masalah
2. Tidak memerlukan banyak indeks atau *cross reference*
3. Pencarian arsip akan lebih terarah
4. Arsip atau data suatu kegiatan tidak terpisah-pisah

Kekurangan menggunakan sistem *chronology system*:

1. Tidak bisa memisahkan hubungan antara subjek umum dan khusus
2. Harus membuat *file* umum dan khusus yang berhubungan dengan penyelesaian masalah
3. Harus memiliki aturan khusus dalam menatanya
4. Memerlukan indeks gabungan

BAB III

PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN

3.1. Tools Yang Digunakan

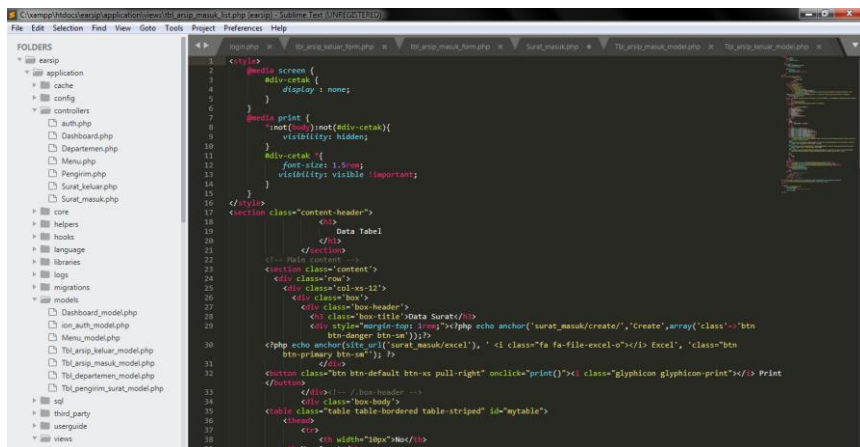
3.1.1. *Sublime*



Gambar 3.18 Logo Sublime

Sublime Text adalah aplikasi *editor* untuk kode dan teks yang dapat berjalan diberbagai *platform operating system* dengan menggunakan teknologi *Phyton API*. Terciptanya aplikasi ini terinspirasi dari aplikasi *Vim*, Aplikasi ini sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *opensource* dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, akan tetapi beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis. *Sublime Text* mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir

di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara *default* dapat lebih dimaksimalkan atau didukung dengan menggunakan *add-ons* yang bisa di *download* sesuai kebutuhan *user*.



Gambar 3.19 Contoh Tampilan Sublime Text

Berikut beberapa fitur yang diunggulkan dari aplikasi Sublime Text:

1. Go to Anything

Fitur yang sangat membantu dalam membuka *file* ataupun menjelajahi isi dari *file* hanya dengan beberapa *keystrokes*.

2. Multiple Selections

Fitur ini memungkinkan user untuk mengubah secara interaktif banyak baris sekaligus, mengubah nama *variabel* dengan mudah, dan memanipulasi *file* lebih cepat dari sebelumnya.

3. *Command Pallette*

Dengan hanya beberapa *keystrokes*, *user* dapat dengan cepat mencari fungsi yang diinginkan, tanpa harus *me-navigasi* melalui menu.

4. *Distraction Free Mode*

Bila *user* memerlukan fokus penuh pada aplikasi ini, fitur ini dapat membantu *user* dengan memberikan tampilan layar penuh.

5. *Split Editing*

Dapatkan hasil yang maksimal dari *monitor* layar lebar dengan dukungan *editing* perpecahan. Mengedit sisi *file* dengan sisi, atau meng-*edit* dua lokasi di satu *file*. Anda dapat meng-*edit* dengan banyak baris dan kolom yang *user* inginkan.

6. *Instant Project Switch*

Menangkap semua *file* yang dimasukkan kedalam *project* pada aplikasi ini. Terintegrasi dengan fitur *Goto Anything* untuk menjelajahi semua *file* yang ada ataupun untuk beralih ke *file* dalam *project* lainnya dengan cepat.

7. *Plugin API*

Dilengkapi dengan *plugin API* berbasis *Phyton* sehingga membuat aplikasi ini sangat tangguh

8. *Customize Anything*

Aplikasi ini memberikan *user* fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini

9. *Cross Platform*

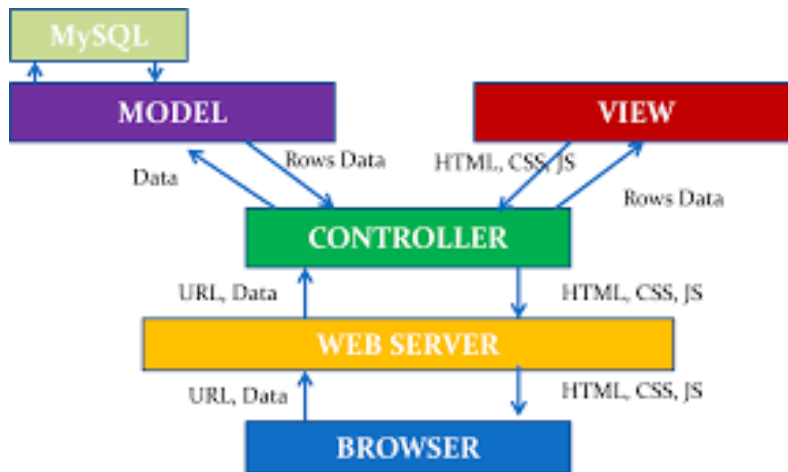
Aplikasi ini dapat berjalan hampir disemua *operating system modern* seperti *Windows*, *OS X*, dan *Linux based operating system*.

3.1.2. *Framework Code Igniter*

Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi atau prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal.

Alasan menggunakan *Framework*:

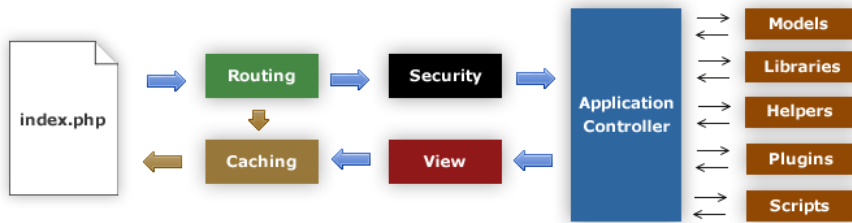
1. Mempercepat dan mempermudah pembangunan sebuah aplikasi *web*.
2. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
3. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya *validasi*, *ORM*, *pagination*, *multiple database*, *scaffolding*, *pengaturan session*, *error handling*, dll)
4. Lebih bebas dalam pengembangan jika dibandingkan *CMS*



Gambar 3.20 Konsep CodeIgniter

CodeIgniter adalah sebuah *web application network* yang bersifat *open source* yang digunakan untuk membangun aplikasi *PHP* dinamis. *CodeIgniter* menjadi sebuah *framework PHP* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan *PHP* yang dapat mempercepat pengembang untuk membuat sebuah aplikasi *web*. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *CodeIgniter* sebagai *framework* pilihannya. *CodeIgniter* pertamakali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, *CodeIgniter* dengan cepat “membakar” semangat para *web developer* untuk mengembangkan *web* dinamis dengan cepat dan mudah menggunakan *framework PHP* yang satu ini.

Alur kerja *framework CodeIgniter*:



Gambar 3.21 Alur Kerja Framework CodeIgniter

1. *Index.php*: *Index.php* disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program.
2. *The Router*: *Router* akan memeriksa *HTTP request* untuk menentukan hal apa yang harus dilakukan oleh program.
3. *Cache File*: Apabila dalam program sudah terdapat “*cache file*” maka *file* tersebut akan langsung dikirim ke *browser*. *File cache* inilah yang dapat membuat sebuah *website* dapat di buka dengan lebih cepat. *Cache file* dapat melewati proses yang sebenarnya harus dilakukan oleh program *codeigniter*.
4. *Security*: Sebelum *file controller* di load keseluruhan, *HTTP request* dan data yang *disubmit* oleh *user* akan disaring terlebih dahulu melalui fasilitas *security* yang dimiliki oleh *codeigniter*.
5. *Controller*: *Controller* akan membuka *file model*, *core libraries*, *helper* dan semua *resources* yang dibutuhkan dalam program tersebut.
6. *View*: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam *view file* dan mengirimkannya ke *browser* supaya dapat dilihat. Apabila *file view* sudah ada yang di “*cache*” maka *file view* baru yang belum ter-*cache* akan mengupdate *file view* yang sudah ada.

Penjelasan MVC:

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi *web*, berawal pada bahasa pemrograman *Small Talk*, *MVC* memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti *manipulasi data*, *user interface*, dan bagian yang menjadi kontrol aplikasi. Dengan menggunakan prinsip *MVC* suatu aplikasi dapat dikembangkan sesuai dengan kemampuan *developer*nya, yaitu programmer yang menangani bagian model dan *controller*, sedangkan *designer* yang menangani bagian *view*, sehingga penggunaan arsitektur *MVC* dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

Terdapat 3 jenis komponen yang membangun suatu *MVC pattern* dalam suatu aplikasi yaitu:

1. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa *file template HTML*, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.

```

1 |<!DOCTYPE html>
2 |<html>
3 |
4 |<head>
5 |<meta charset="utf-8">
6 |<meta http-equiv="X-UA-Compatible" content="IE=edge">
7 |<title>E-Archive | Assessment Center</title>
8 |<!-- Tell the browser to be responsive to screen width -->
9 |<meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
10 |<!-- Bootstrap 3.3.5 -->
11 |<link rel="stylesheet" href="{?php echo base_url()}?template/bootstrap/css/bootstrap.min.css">
12 |<!-- Font Awesome -->
13 |<!--<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">-->
14 |<link rel="stylesheet" href="{?php echo base_url()}?template/font-awesome-4.4.0/css/font-awesome.min.css">
15 |<!-- Ionicons -->
16 |<!--<link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">-->
17 |<link rel="stylesheet" href="{?php echo base_url()}?template/ionicons-2.0.1/css/ionicons.min.css">
18 |<!-- DataTables -->
19 |<link rel="stylesheet" href="{?php echo base_url()}?template/plugins/datatables/dataTables.bootstrap.css">
20 |<!-- Theme style -->
21 |<link rel="stylesheet" href="{?php echo base_url()}?template/dist/css/AdminLTE.min.css">
22 |<!-- AdminLTE Skins. Choose a skin from the css/skins
23 |     folder instead of downloading all of them to reduce the load. -->
24 |<link rel="stylesheet" href="{?php echo base_url()}?template/dist/css/skins/_all-skins.min.css">
25 |<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
26 |<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
27 |<!--[if lt IE 9]>
28 |<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
29 |<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
30 |<endif-->
31 |</head>
32 |<body class="hold-transition skin-yellow sidebar-mini">
33 |<div class="wrapper">
34 |
35 |<header class="main-header">
36 |<!-- Logo -->
37 |<a href="#" class="logo">
38 |<!-- logo for regular state and mobile devices -->
39 |<span class="logo-lg"><b>E-Archive</b></span>
40 |</a>
41 |<!-- Header Navbar: style can be found in header.less -->

```

Gambar 3.22 Contoh Source Code View

2. *Model*, biasanya berhubungan langsung dengan *database* untuk memanipulasi data (*insert*, *update*, *delete*, *search*), menangani *validasi* dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

```

1 |<?php
2 |
3 |if (!defined('BASEPATH'))
4 |    exit('No direct script access allowed');
5 |
6 |class Menu_model extends CI_Model
7 |{
8 |
9 |    public $table = 'menu';
10 |    public $id = 'id';
11 |    public $order = 'DESC';
12 |
13 |    function __construct()
14 |    {
15 |        parent::__construct();
16 |    }
17 |
18 |    // get all
19 |    function get_all()
20 |    {
21 |        $this->db->order_by($this->id, $this->order);
22 |        return $this->db->get($this->table->result();
23 |    }
24 |
25 |    // get data by id
26 |    function get_by_id($id)
27 |    {
28 |        $this->db->where($this->id, $id);
29 |        return $this->db->get($this->table->row();
30 |    }
31 |
32 |    // get total rows
33 |    function total_rows($q = NULL) {
34 |        $this->db->like('id', $q);
35 |        $this->db->or_like('name', $q);
36 |        $this->db->or_like('link', $q);
37 |        $this->db->or_like('icon', $q);
38 |        $this->db->or_like('is_active', $q);
39 |        $this->db->or_like('is_parent', $q);
40 |        $this->db->from($this->table);
41 |        return $this->db->count_all_results();

```

Gambar 3.23 Contoh Source Code Model

3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses oleh aplikasi.

```
1 <?php defined('BASEPATH') OR exit('No direct script access allowed');
2
3 class Auth extends CI_Controller {
4
5     function __construct()
6     {
7         parent::__construct();
8         $this->load->database();
9         $this->load->library(array('ion_auth', 'form_validation'));
10        $this->load->helper(array('url', 'language'));
11
12        $this->form_validation->set_error_delimiters($this->config->item('error_start_delimiter', 'ion_auth'), $this->
13            config->item('error_end_delimiter', 'ion_auth'));
14
15        $this->lang->load('auth');
16    }
17
18    // redirect if needed, otherwise display the user list
19    function index()
20    {
21        if (!$this->ion_auth->logged_in())
22        {
23            // redirect them to the login page
24            redirect('auth/login', 'refresh');
25        }
26        elseif (!$this->ion_auth->is_admin()) // remove this elseif if you want to enable this for non-admins
27        {
28            // redirect them to the home page because they must be an administrator to view this
29            return show_error('You must be an administrator to view this page.');
```

Gambar 3.24 Contoh Source Code Controller

3.1.3. XAMPP



Gambar 3.25 Logo XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. XAMPP merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi

melakukan instalasi dan konfigurasi web server *Apache*, *PHP* dan *MySQL* secara manual. *XAMPP* akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi. *XAMPP* merupakan salah satu paket *instalasi Apache*, *PHP* dan *MySQL instant* yang dapat kita gunakan untuk membantu proses instalasi ketiga produk tersebut. Selain paket instalasi *instant XAMPP* versi 1.6.4 juga memberikan fasilitas pilihan penggunaan *PHP4* atau *PHP5*. Untuk berpindah versi *PHP* yang ingin digunakan juga sangat mudah dilakukan dengan menggunakan bantuan *PHP-Switch* yang telah disertakan oleh *XAMPP*, dan yang terpenting *XAMPP* bersifat *free* atau gratis untuk digunakan. *XAMPP* merupakan *tool* yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan meng-*install XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi web server *Apache*, *PHP* dan *MySQL* secara manual. *XAMPP* akan menginstalasi dan mengkonfigurasikannya secara otomatis. Merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat men-*download* langsung dari web resminya. Fungsi *XAMPP* sendiri adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* sendiri merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. Untuk mendapatkannya *XAMPP* anda dapat men-*download* langsung dari web resminya. Dan

berikut beberapa definisi program lainnya yang terdapat dalam *XAMPP*.

Fitur-fitur *XAMPP*:

1. *Apache*

Apache adalah perangkat lunak sumber terbuka yang menjadi alternatif dari *server web Netscape*. *Server HTTP Apache* atau *Server Web* atau *WWW Apache* merupakan *server web* yang dapat dijalankan di banyak sistem operasi yang berguna untuk melayani dan memfungsikan situs *web*. *Apache* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

2. *MySQL*

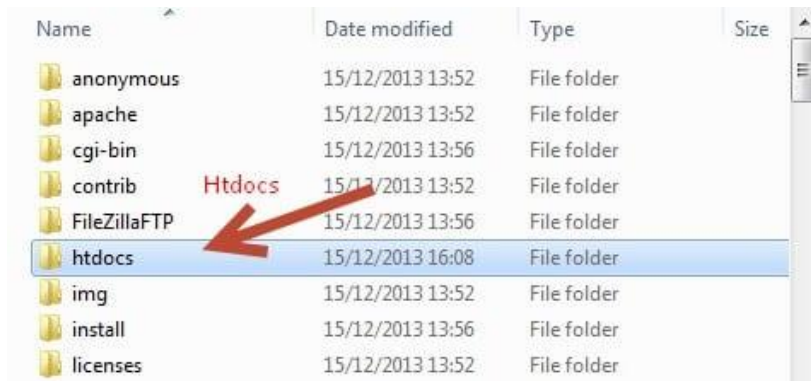
MySQL adalah singkatan “*My Structured Query Language*”. Program ini berjalan sebagai *server* menyediakan *multi-user* mengakses ke sejumlah *database*. *MySQL* umumnya digunakan oleh perangkat lunak bebas yang memerlukan fitur penuh sistem manajemen *database*, seperti *WordPress*, *phpBB* dan perangkat lunak lain yang dibangun pada perangkat lunak *LAMP*. Ia juga digunakan dalam skala sangat tinggi *World Wide Web*, termasuk produk-produk *Google* dan *Facebook*.

3. *PHP*

PHP adalah bahasa pemrograman *script* yang banyak dipakai untuk memrogram situs *web dinamis*, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain. Contoh terkenal dari aplikasi *PHP* adalah *phpBB* dan *MediaWiki* (*software* di belakang *Wikipedia*).

Bagian-Bagian XAMPP:

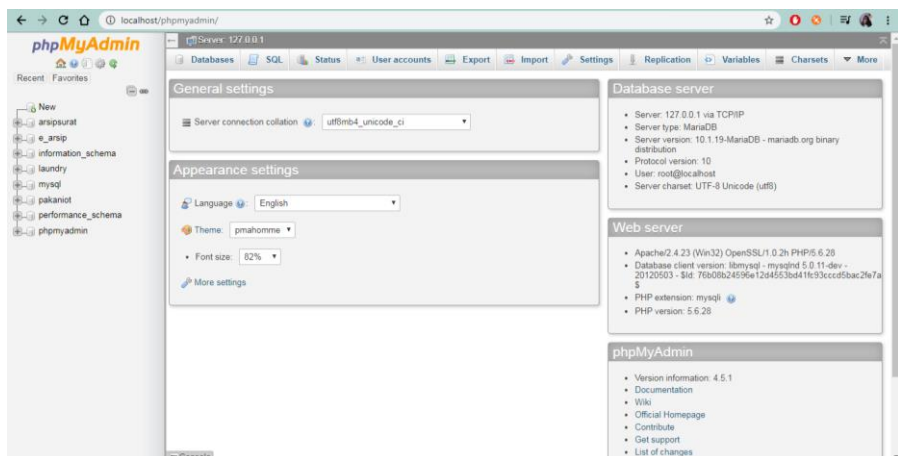
1. *Htdocs*



Gambar 3.26 Contoh *Htdocs*

Htdocs adalah sebuah *folder* yang digunakan sebagai tempat penyimpanan berkas seperti *PHP*, *HTML*, dan *script* lain yang digunakan dalam sebuah halaman *website*. Secara kapasitas penyimpanan, *XAMPP* tergantung dari seberapa besar kapasitas *hardisk* di laptop atau komputer anda. Sedangkan bila menggunakan *hosting online*, maka tergantung pilihan waktu membeli sebuah *hosting*.

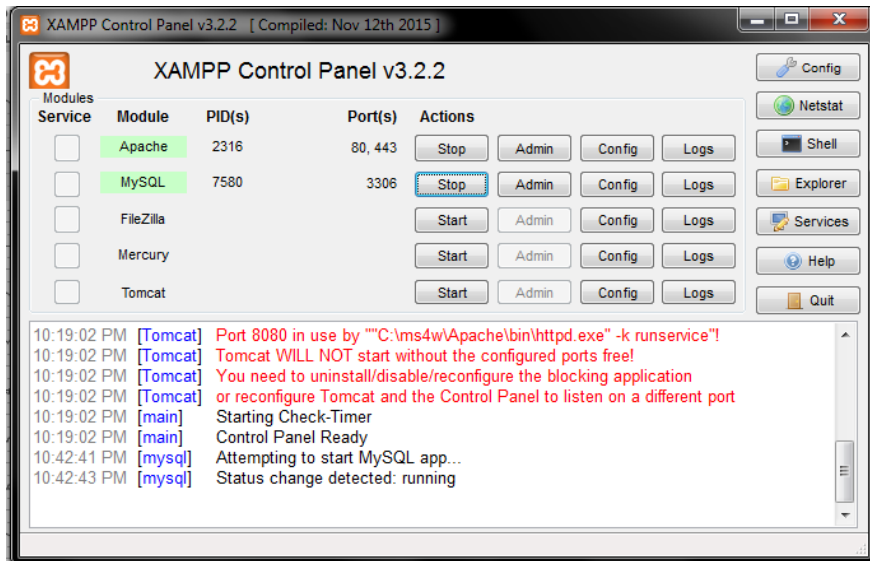
2. *phpMyadmin*



Gambar 3.27 Contoh *phpMyadmin*

phpMyadmin adalah sebuah tempat yang digunakan untuk mengelola *database MySQL* yang berada di komputer atau laptop. Untuk mengakses *phpMyadmin* yakni dengan membuka *browser internet* (*Mozilla* atau *chrome*) lalu ketikkan alamat *http://localhost/phpMyadmin*, maka akan muncul tampilannya.

3. Control Panel



Gambar 3.28 Contoh Control Panel

Control Panel adalah sebuah layanan untuk mengelola *XAMPP* baik itu mengontrol (*start* atau *stop XAMPP*) serta layanan *service* lainnya. Secara *online* di dalam *hosting* atau *VPS* dikenal *CPanel*.

Kelebihan dari *XAMPP*, diantaranya:

1. *Database Storage Engine* ini banyak digunakan oleh *programmer* apalagi oleh *web developer* karena sifatnya yang *free*. Untuk yang *expert* sudah ada yang bayar.

2. Kemampuannya sudah bisa diandalkan, mempunyai kapasitas yang cukup mumpuni sekitar 60.000 tabel dengan jumlah *record* mencapai 5.000.000.000 bahkan untuk yang terbaru sudah lebih.
3. Keamanan datanya cukup aman walaupun tidak sehebat *Postgre* apalagi *Oracle*.
4. *Engine* ini *multiplatform* sehingga mampu diaplikasikan di berbagai sistem operasi. *MySQL* cocok diaplikasikan diaplikasi kelas kecil dan menengah.
5. Kelebihan paling utama *engine* ini adalah kecepatannya.

Kekurangan dari *XAMPP*, diantaranya:

1. Tidak cocok untuk menangani data dengan jumlah yang besar, baik untuk menyimpan data maupun untuk memproses data.
2. Memiliki keterbatasan kemampuan kinerja pada *server* ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung *server* karena tidak menerapkan konsep *Technology Cluste*.

3.2. Bahasa Pemrograman

3.2.1. PHP

PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa pemrograman berbasis *script server-side* yang dikembangkan untuk pembuatan *web*. Selain untuk pengembangan *web*, *PHP* juga dimanfaatkan sebagai bahasa untuk pemrograman umum. *PHP* disebut sebagai bahasa pemrograman *server side* karena ia diproses pada komputer *server* saja. Hal inilah yang membedakan PHP dengan bahasa pemrograman *client-side* seperti *JavaScript*.

Fungsi PHP adalah membuat atau mengembangkan situs *web statis* atau situs *web dinamis* atau aplikasi web. Walaupun sebenarnya bukan hanya *PHP* bahasa pemrograman yang bisa digunakan untuk memuat *website*. PHP digunakan karena untuk membuat *website* dinamis bisa digunakan untuk menyimpan data ke dalam *database*, membuat halaman yang dapat berubah-ubah sesuai dengan *input user*, memproses *form*, dll. Dalam membuat file PHP dapat digabung menggunakan tag html, Dan ketika tanpa menggunakan tag html apa pun disebut *file PHP Murni*. Server menginterpretasikan kode PHP dan mengeluarkan hasilnya sebagai kode HTML ke browser web. Agar server mengidentifikasi kode PHP dari kode HTML, kita harus selalu menyertakan kode PHP dalam tag PHP.

Kelebihan bahasa pemrograman PHP, diantaranya:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan di mana - mana dari mulai *apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya *milis - milis* dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (*Linux*, *Unix*, *Macintosh*, *Windows*) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

PHP mempunyai aturan dasar untuk menuliskannya, diantaranya:

1. Penulisan *case sensitivity*

Dalam hal perbedaan penggunaan *case sensitivity* atau huruf besar dan kecil, *PHP* tidak membedakan hal tersebut untuk nama *class*, penamaan fungsi (*function*), ataupun *keyword* bawaan *PHP* seperti *while* dan *echo*. Jadi, `<?php Echo "contoh"; ECHO "contoh"; ?>` akan dianggap sama dalam *PHP*. Namun, *PHP* akan membedakan *case sensitive* untuk penamaan *variabel*. Jadi, `$Nama` dan `$NAMA` akan dianggap berbeda.

2. Penulisan baris perintah

Baris perintah atau *statement* dalam *PHP* merupakan kumpulan perintah yang menginstruksikan *PHP* untuk melakukan sesuatu. Baris perintah tersebut dapat terdiri dari satu baris singkat saja atau bisa juga sesuatu yang lebih rumit dan terdiri dari sejumlah baris seperti kode perulangan (*loop*). Untuk penulisan yang sederhana misalnya perintah *echo* untuk menampilkan *text* pada layar.

3. Karakter *spasi* dan *tab*

Secara umum, baik karakter *tab* maupun *spasi* diabaikan dalam eksekusi program *PHP*. Dalam hal ini, Anda diperbolehkan memecah suatu *statement* menjadi beberapa baris, maupun menyatukan sejumlah *statement* dalam satu baris yang panjang. Meskipun membuatnya menjadi satu baris panjang dapat menghemat tempat, namun lebih disarankan untuk membuatnya menjadi beberapa baris untuk memudahkan pembacaan kode program.

Berikut adalah perbandingan antara PHP dengan dengan bahasa *scripting server-side* lainnya seperti *ASP – Active Server Pages*, *JSP – Java Server Pages*, *CFML – Cold Markup Language*:

FITUR	PHP	ASP	JSP	CFML
Kurva belajar	pendek	Lebih lama dari PHP	Lebih lama dari PHP	Lebih lama dari PHP
Hosting web	Didukung oleh hampir semua server hosting	Membutuhkan server khusus	Cukup didukung	Membutuhkan server khusus
Sumber terbuka	Iya	Tidak	Iya	Baik komersial maupun open source
Dukungan layanan web	Dibangun	Menggunakan kerangka .NET	Menggunakan add on libraries	Dibangun
Integrasi dengan HTML	Mudah	Cukup rumit	Cukup rumit	Mudah
Dukungan MySQL	Asli	Membutuhkan driver pihak ketiga	Membutuhkan driver pihak ketiga	Versi saat ini memiliki dukungan asli. Versi yang lebih lama menggunakan ODBC
Mudah diperluas dengan bahasa lain	Iya	Tidak	Diperpanjang menggunakan kelas dan perpustakaan Java.	Iya

Gambar 3.29 Perbandingan PHP dengan Scripting Server Side

Tipe data yang PHP memiliki 8 tipe data, yaitu:

1. Boolean
2. Integer
3. Float/ Double
4. String
5. Array
6. Object
7. Resource
8. NULL

3.2.2. CSS

CSS (*Cascading Style Sheet*) adalah salah satu bahasa desain *web* (*style sheet language*) yang mengontrol *format* tampilan sebuah halaman *web* yang ditulis dengan menggunakan penanda (*markup language*). Biasanya CSS digunakan untuk mendesain sebuah halaman *HTML* dan *XHTML*, tetapi sekarang CSS bisa diaplikasikan untuk segala dokumen *XML*, termasuk *SVG* dan *XUL* bahkan *ANDROID*. CSS dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkann daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi darisebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah *formatting* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik *tableless* pada desain *web*.

CSS juga memungkinkan sebuah halaman untuk ditampilkan dalam berbagai *style* dengan menggunakan metode pembawaan yang berbeda pula, seperti *on-screen*, *in-print*, *by voice*, dan lain-lain. Sementaraitu, pemilik konten *web* bisa menentukan link yang menghubungkan konten dengan *file CSS*. Tujuan utama CSS diciptakan untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan itu, pembuatan ataupun pemrograman ulang *web* akan lebih mudah dilakukan. Hal yang termasuk dalam desain *web* diantaranya adalah warna, ukuran dan *formatting*. Dengan adanya CSS, konten dan *desain web* akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu *web*, sehingga akan memudahkan dalam membuat halaman *web* yang

banyak, yang pada akhirnya dapat memangkas waktu pembuatan web. Fungsi utama CSS adalah merancang, merubah, mendisain, membentuk halaman *website* (blog juga *website*). dan isi dari halaman *website* adalah tag-tag html, logikanya CSS itu dapat merubah tag-tag html (yang sederhana) sehingga menjadi lebih fungsional dan menarik.

Untuk cara kerjanya CSS beroperasi melalui tag `<style>` dengan atribut class warna. Dengan adanya CSS pada HTML tersebut maka pengaturan warna teks akan menjadi lebih mudah. Saat kamu ingin mengganti warna teks cukup mengetikkan tag `` tanpa harus menulis ulang perintah. Jadi bisa disimpulkan bahwa CSS akan menghemat waktumu dengan perintah-perintah yang efisien. Hal ini bisa terjadi karena CSS sendiri dikembangkan untuk bisa mengubah tampilan laman website tanpa harus mengganti isi konten. Jika kembali pada perumpamaan manusia dan pakaian di poin sebelumnya, dengan CSS kamu tidak mengubah bentuk manusianya tapi hanya mengganti pakaiannya. Dengan begitu untuk mengubah dan memprogram ulang tampilan website pun bisa dilakukan dalam waktu cepat. peran CSS untuk website sangatlah penting. Tanpa adanya CSS, tampilan website akan membosankan atau bahkan membutuhkan waktu lama untuk loading. Bayangkan saja kamu hanya bisa bergantung pada HTML untuk membuat sebuah situs. Bukan cuma tampilan situs akan “hambar” tapi kamu juga butuh waktu lebih lama karena harus berulang kali mengetikkan perintah.

Macam-macam CSS, diantaranya yaitu:

1. *Inline style sheet*

Pertama ada *inline style sheet*. Sederhananya, CSS model ini adalah CSS dengan perintah pemrograman yang letaknya ada

pada objek. Misalnya kamu ingin mengubah sebuah tulisan pada laman tertentu di website milikmu, inline style sheet CSS harus menempel pada elemen tulisan tersebut. Kamu cukup menambahkan tag `<style>` saja untuk menerapkan CSS ini.

2. *External style sheet*

Selain *inline style sheet* ada *external style sheet*. CSS ini letaknya berbeda dengan laman yang akan diubah. Cara ini lebih praktis daripada inline style sheet karena bisa menghemat ruang dan bisa digunakan berulang-ulang untuk laman web yang berbeda. Kamu bisa mengenali CSS tipe ini lewat tag `<link rel>`. Tag ini akan menghubungkan halaman *coding* pada *external style sheet CSS* yang terpisah.

3. *Embedded style sheet*

Terakhir ada *embedded style sheet*. CSS model ini sama seperti *inline style sheet*, sama-sama berada pada satu laman *coding*. Maka tidak mengherankan jika embedded style sheet terkadang disebut dengan *internal style sheet*. Biasanya CSS ini diapit oleh tag `<head>` `</head>` dan diawali dengan tag `<style>`. *Embedded style sheet* sering digunakan untuk mengatur laman web dengan tampilan yang unik. Misalnya dalam satu paragraf tulisan ada kalimat yang berbeda dan hal tersebut terus berulang.

BAB IV

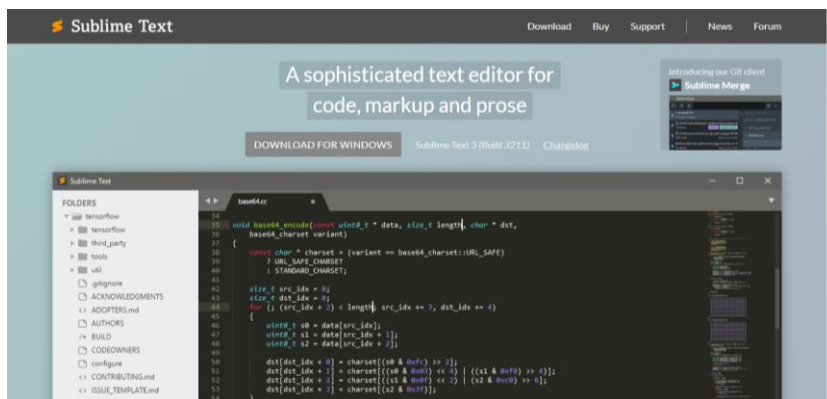
INSTALASI *TOOLS* YANG DIGUNAKAN

1. *Tools* Yang Digunakan

4.1.1. *Sublime Text*

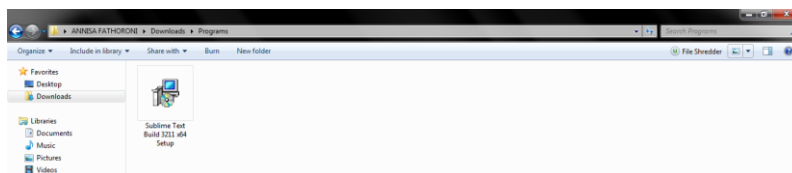
Berikut ini merupakan tahapan-tahapan yang dilakukan untuk instalasi *Sublime Text*, yaitu:

1. Tahapan yang pertama untuk melakukan instalasi maka download dahulu *file .exe* pada <https://www.sublimetext.com/>.



Gambar 4.30 Website Sublime

2. Tahapan yang kedua setelah selesai mendownload maka buka *directory* dimana *file .exe* tersimpan, kemudian Double klik *file .exe* hasil download:



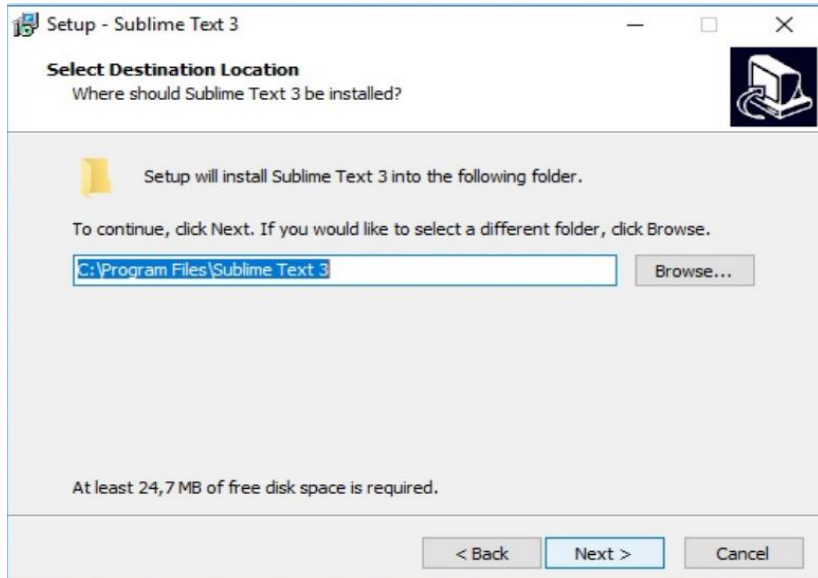
Gambar 4.31 Letak Directory File .exe Tersimpan

3. Tahapan ketiga, jika sudah *double* klik *file* .exe maka akan muncul tampilan berikut, kemudian klik next:



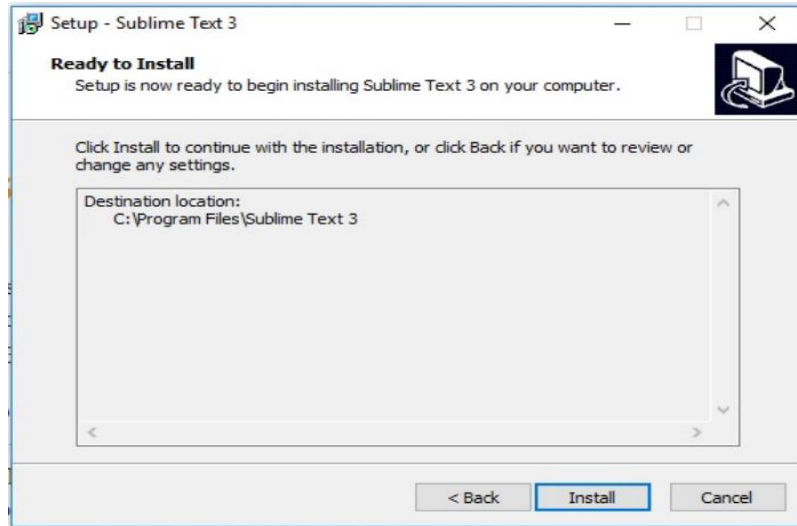
Gambar 4.32 Proses Instalasi Sublime

4. Selanjutnya, klik next:



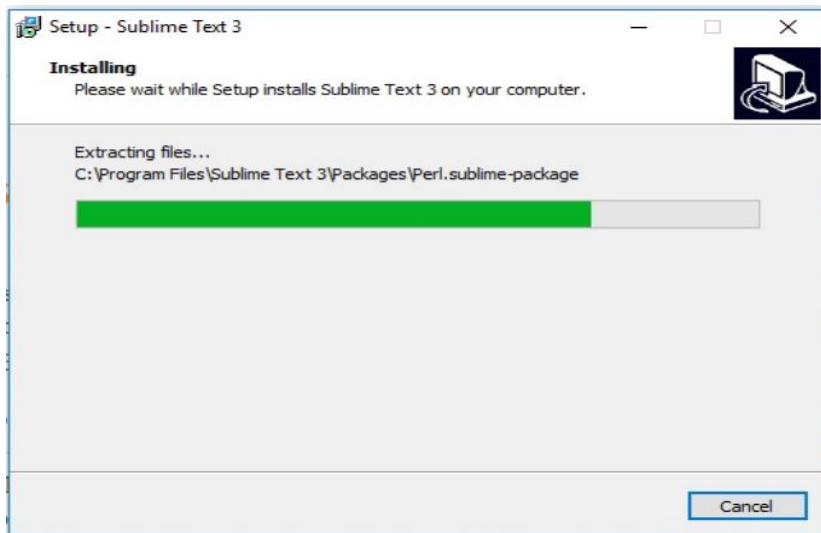
Gambar 4.33 Proses Instalasi Sublime

5. Selanjutnya, klik next:



Gambar 4.34 Proses Instalasi Sublime

6. Tunggu sampai proses instalasi selesai, Setelah proses ini selesai maka sublime telah bisa digunakan.



Gambar 4.35 Proses Instalasi Sublime

4.1.2. XAMPP

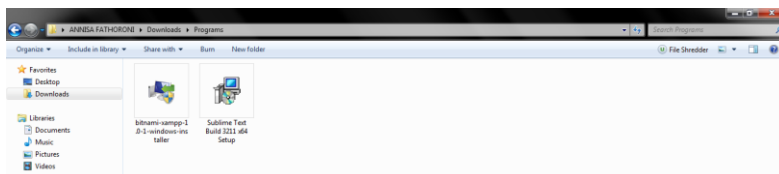
Berikut ini merupakan tahapan-tahapan yang dilakukan untuk instalasi XAMPP, yaitu:

1. *Download Package XAMPP* pada situ resminya, <https://www.apachefriends.org/download.html>



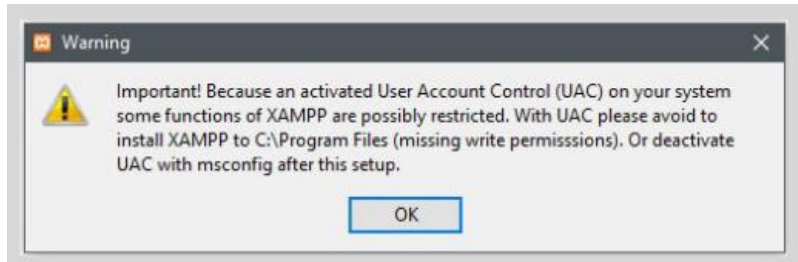
Gambar 4.36 Website XAMPP

2. Tahapan yang kedua setelah selesai mendownload maka buka *directory* dimana *file .exe* tersimpan, kemudian Double klik file *.exe* hasil download:



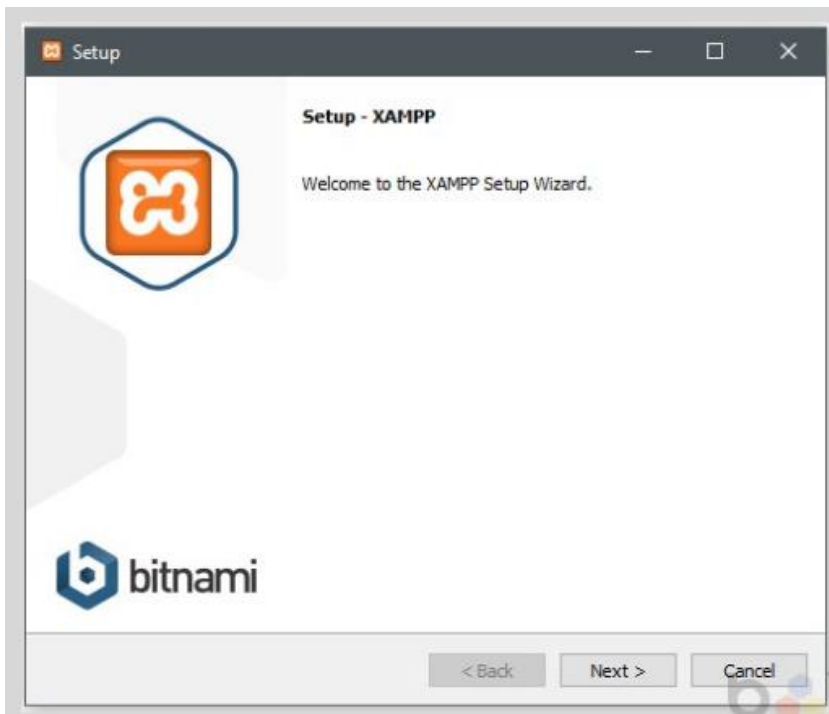
Gambar 4.37 Letak Directory

3. Lalu selanjutnya akan muncul *splash screen bitnami*, setelah itu akan ada peringatan *error*, ini terjadi terkait masalah *permission* atau hak akses, klik OK karena nantinya akan di konfigurasi ke *hard drive* lain.



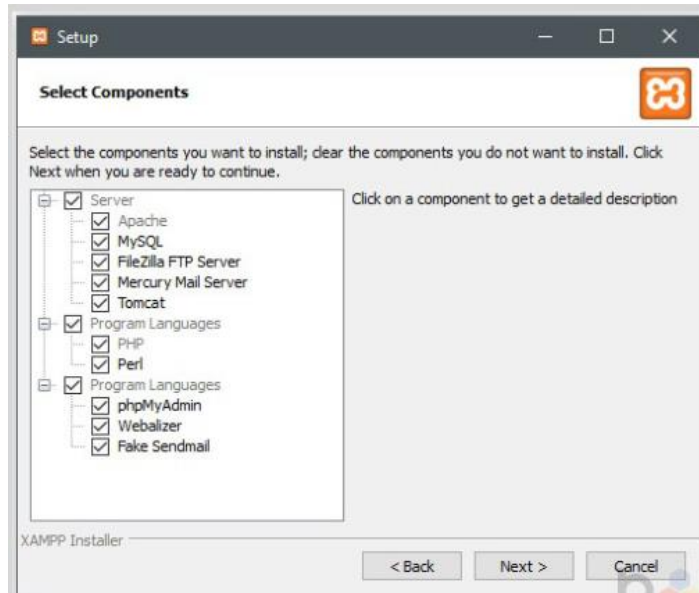
Gambar 4.38 Proses Instalasi XAMPP

4. Setelah itu akan muncul bagian awal *instalasi Setup Wizard xamp*, klik *Next* untuk memulai instalasi



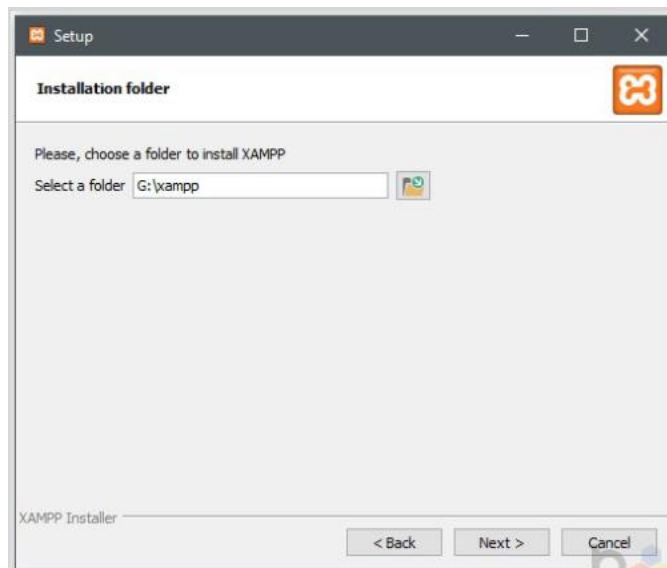
Gambar 4.39 Proses Instalasi XAMPP

5. Setelah itu akan muncul halaman pemilihan komponen apa saja yang ingin di *install* di dalam *web server xampp* ini, silahkan untuk di sesuaikan, anda dapat meminimalisir modul-modul instalasi sesuai kebutuhan, tetapi di dalam tutorial ini saya membiarkan semuanya terinstall. jika sudah klik *Next*.



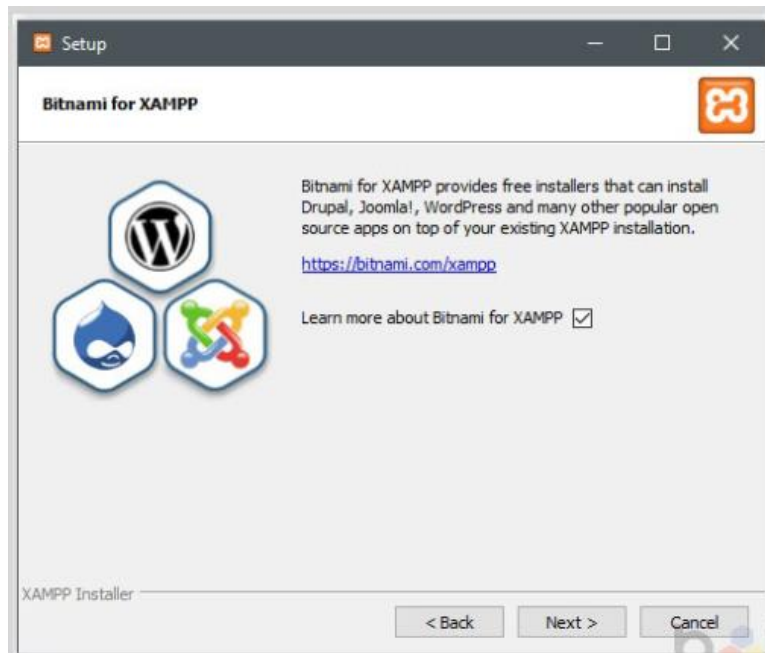
Gambar 4.40 Proses Instalasi XAMPP

6. Selanjutnya akan di arahkan untuk penempatan folder instalasi xampp, jika secara bawaan akan di arahkan ke drive C, karena masalah hak akses yang terjadi di awal tadi sehingga perlu di arahkan ke hard drive lain, setelah itu klik Next.



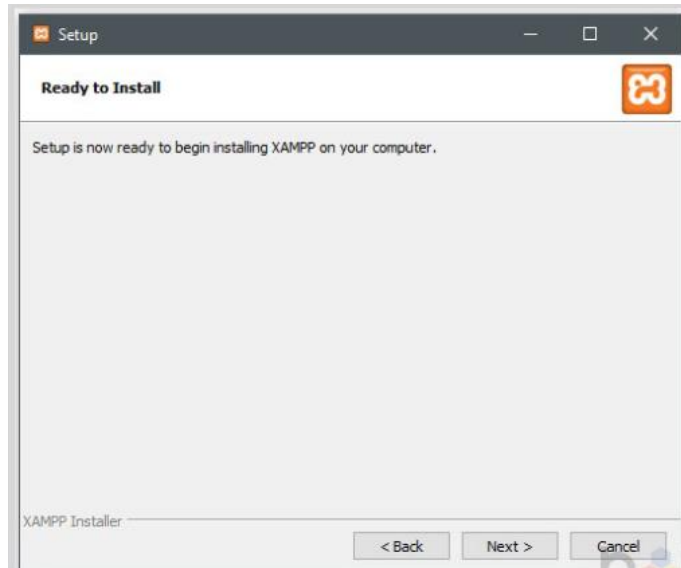
Gambar 4.41 Proses Instalasi XAMPP

7. Langkah berikutnya adalah tampilan promosi bitnami menawarkan cara menginstall cms seperti *WordPress* dan lain-lain. Pada langkah ini ada sebuah *checkbox* pada bagian *Learn more about Bitname for Xampp* silahkan di hilangkan ceklis yang secara bawaan telah ada jika tidak ingin mengikuti petunjuk instalasi *CMS*. jika sudah klik *Next*.



Gambar 4.42 Proses Instalasi XAMPP

8. Pada langkah berikutnya *Ready to Install* ini sebenarnya hanya untuk memastikan apakah anda sudah yakin pada tahap-tahap sebelumnya, jika di rasa anda kurang yakin bisa menekan tombol back, tetapi dalam tutorial ini telah di uji coba langkah-langkah instalasi xampp sebagai *web server* lokal di tutorial ini. Jika sudah klik *Next*.



Gambar 4.43 Proses Instalasi XAMPP

9. Selanjutnya adalah proses *extract file-file xampp* ke dalam *hard drive folder* yang telah di tetapkan sebelumnya, tunggu sampai proses instalasi selesai dimana proses ini tidak memakan waktu lama. Jika sudah klik *Next*.



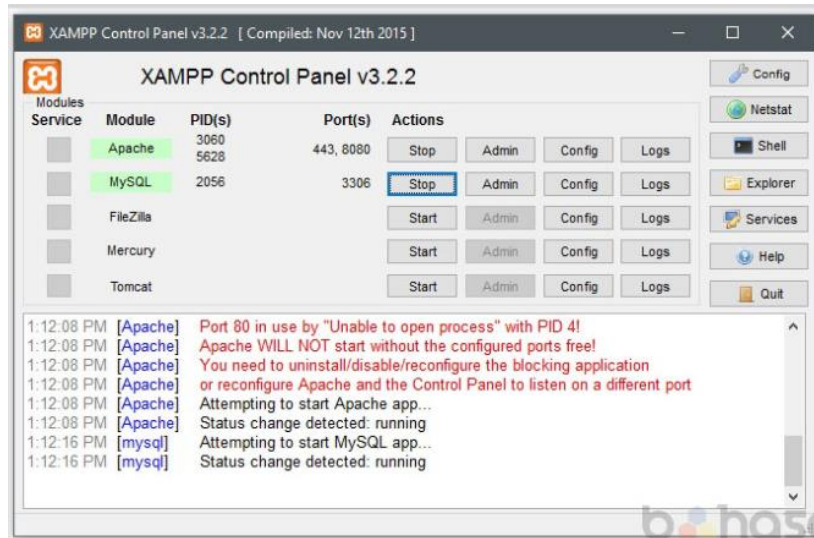
Gambar 4.44 Proses Instalasi XAMPP

10. Langkah berikutnya anda akan di bawa ke bagian *Completing the Xampp Setup Wizard* ini adalah langkah terakhir dalam rangkaian proses instalasi. Terdapat *checkbox Do you want to start the Control Panel now?* biarkan saja ceklis tersebut, atau jika tidak ada ceklis maka silahkan berikan ceklis agar setelah ini *Xampp control panel* secara otomatis berjalan. Jika sudah klik *finish*.



Gambar 4.45 Proses Instalasi XAMPP

11. Sampai disini proses instalasi telah selesai, bisa mengecek langsung ke folder yang sebelumnya telah di arahkan. Karena pada tahap terakhir kita memberikan ceklis agar xampp control panel secara otomatis tampil maka tunggu sampai control panel tersebut muncul. Jika *Control Panel Xampp* sudah muncul silahkan klik *start* pada *Apache* dan *Mysql* hingga tulisan *Apache* dan *Mysql* di *block* warna hijau.



Gambar 4.46 Tampilan XAMPP

12. Untuk menguji jika semua sudah berjalan dengan baik silahkan buka *browser* dan arahkan ke alamat <http://localhost/> atau <http://127.0.0.1/> ini adalah alamat umum komputer lokal. Jika berhasil seharusnya di arahkan ke halaman XAMPP.

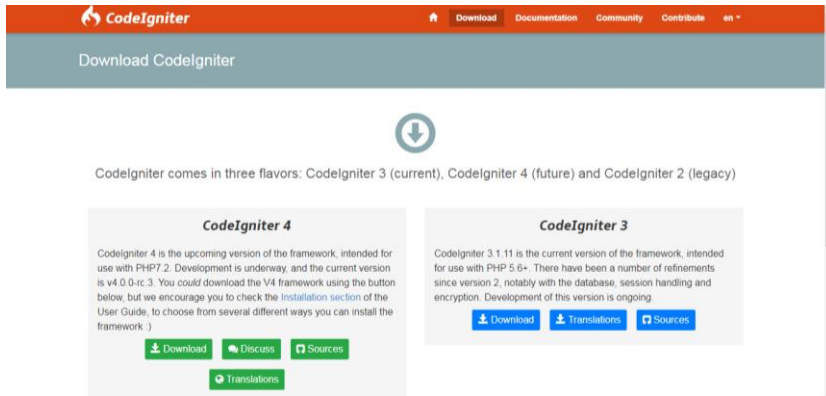


Gambar 4.47 Hasil Akhir XAMPP

4.1.3. Framework CodeIgniter

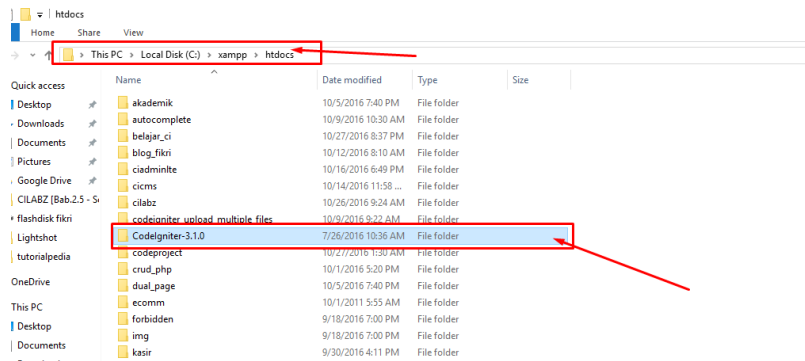
Berikut ini merupakan tahapan-tahapan yang dilakukan untuk instalasi Framework CodeIgniter, yaitu:

1. *Download Package codeigniter* pada situ resminya, <https://codeigniter.com/download>



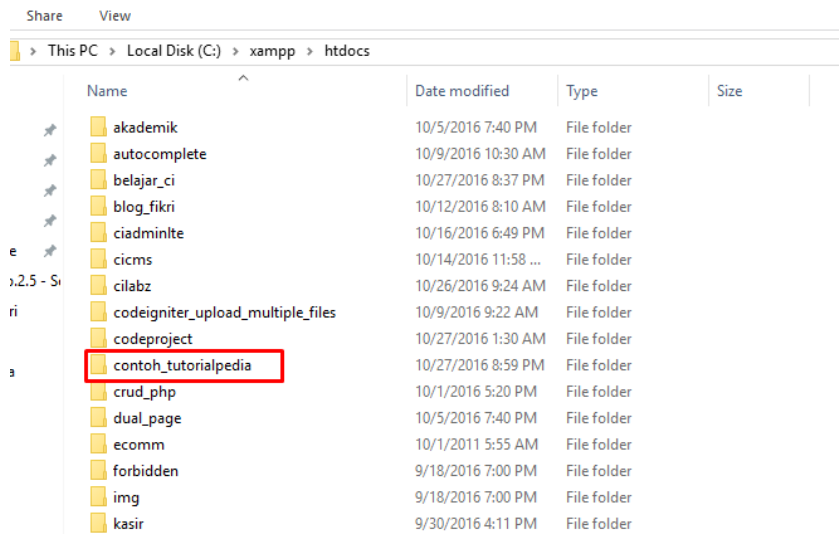
Gambar 4.36 Website CodeIgniter

2. Setelah di *download* ekstrak *file codeigniter* pada direktori C:\xampp\htdocs



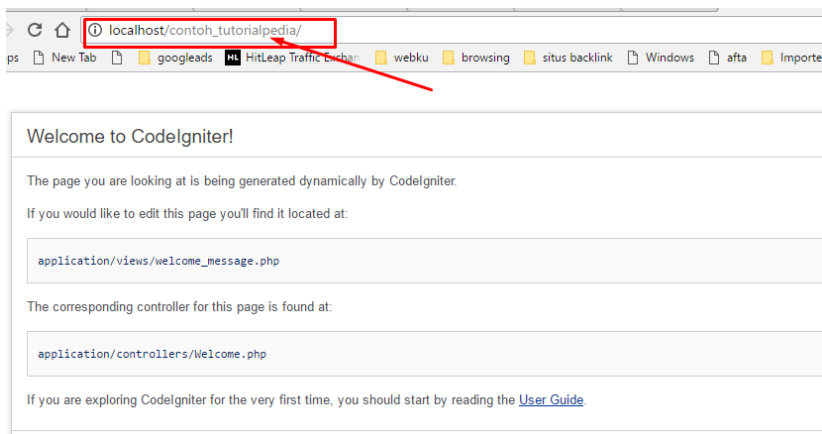
3. *Gambar 4.37 Lokasi Direktori Instalasi CodeIgniter*

4. Lalu ubah folder Codeigniter menjadi sesuai keinginan kita. disini saya akan ubah folder codeigniter tersebut menjadi contoh_tutorialpedia



Gambar 4.38 Lokasi Direktori Instalasi CodeIgniter

- Setelah itu, coba akses folder tersebut melalui browser, jangan lupa untuk menyalakan XAMPP terlebih dahulu. Lalu akses dengan url `http://localhost/nama_folder_project` pada tutorial yang diberi nama dengan `contoh_tutorialpedia`



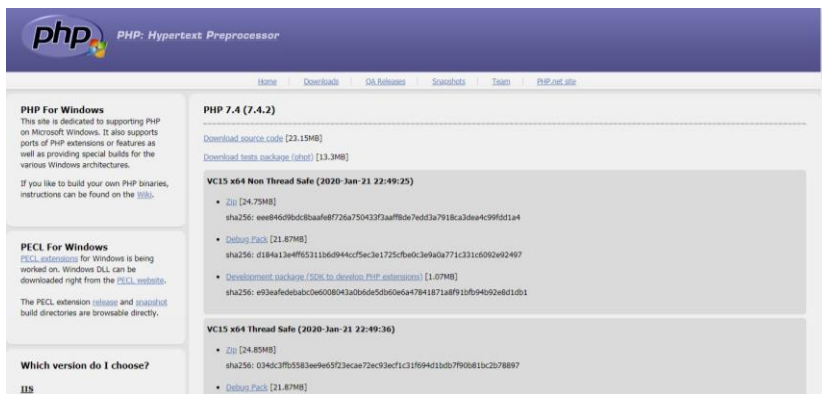
Gambar 4.39 Tampilan Awal CodeIgniter

2. Bahasa Pemrograman Yang Digunakan

4.2.1. PHP

Berikut ini merupakan tahapan-tahapan yang dilakukan untuk instalasi *PHP*, yaitu:

1. *Download Package PHP* pada situ resminya, <http://windows.php.net>.



Gambar 4.40 Website PHP

2. Buat *folder* baru di "C:\Apache" namainya dengan "php" jadi *folder path*nya seperti berikut "C:\Apache\php". *Copy* semua *file* dan *folder* yang telah di *unzip* ke "C:\Apache\php". Dalam *folder* "C:\Apache\php" cari dan *rename* "php.ini-production" ke "php.ini".

Double click php.ini untuk membuka dan cari baris berikut:

```
doc_root =
```

Tambahkan *path* ke *Document Root*, bisa di cari pada *Apache konfigurasi*:

```
doc_root = "D:\My Websites"
```

Kemudian cari baris berikut:

```
; extension_dir = "ext"
```

Buang komen dan tambahkan *path* ke dalam *direktori ekstensi* PHP:

```
extension_dir = "C:\Apache\php\ext"
```

3. Secara *default* PHP akan menggunakan windows *temp* sebagai *folder temporer*. Bila terdapat *errot* dalam *script* secara kebetulan aku mengosongkan *system foldet temp*. Jadi mungkin ini adalah solusi yang bagus untuk membuat *folder temp* yang terpisah untuk PHP. Buat *folder* baru di "C:\Apache" namai foldernya dengan "temp" didalam "temp" buat 2 folder degan nama "upload" dan "session" Folder baru yakni "C:\Apache\temp\upload" dan "C:\Apache\temp\session"

Dalam *file* konfigurasi PHP (php.ini) cari baris berikut:

```
;upload_tmp_dir =
```

Buang komenya dan tambahkan *path* untuk *folder upload*:

```
upload_tmp_dir= "C:\Apache\temp\upload"
```

Dan kemudian baris berikut:

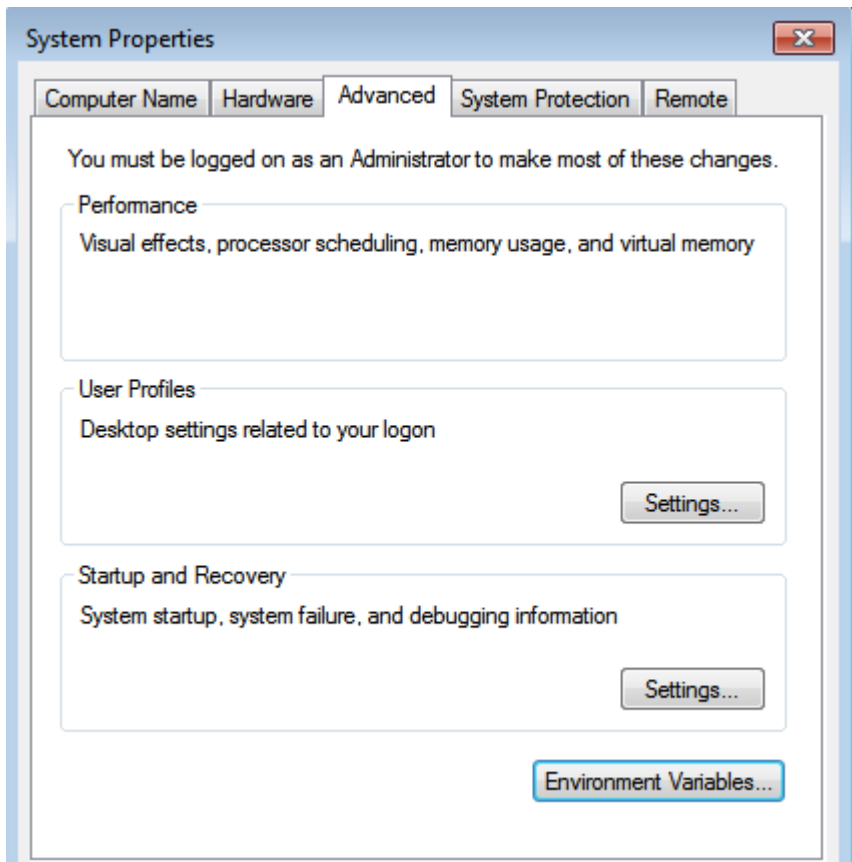
```
;session.save_path = "/tmp"
```

Buang komennya dan tambahkna *path* untuk *folder session*:

```
session.save_path= "C:\Apache\temp\session"
```

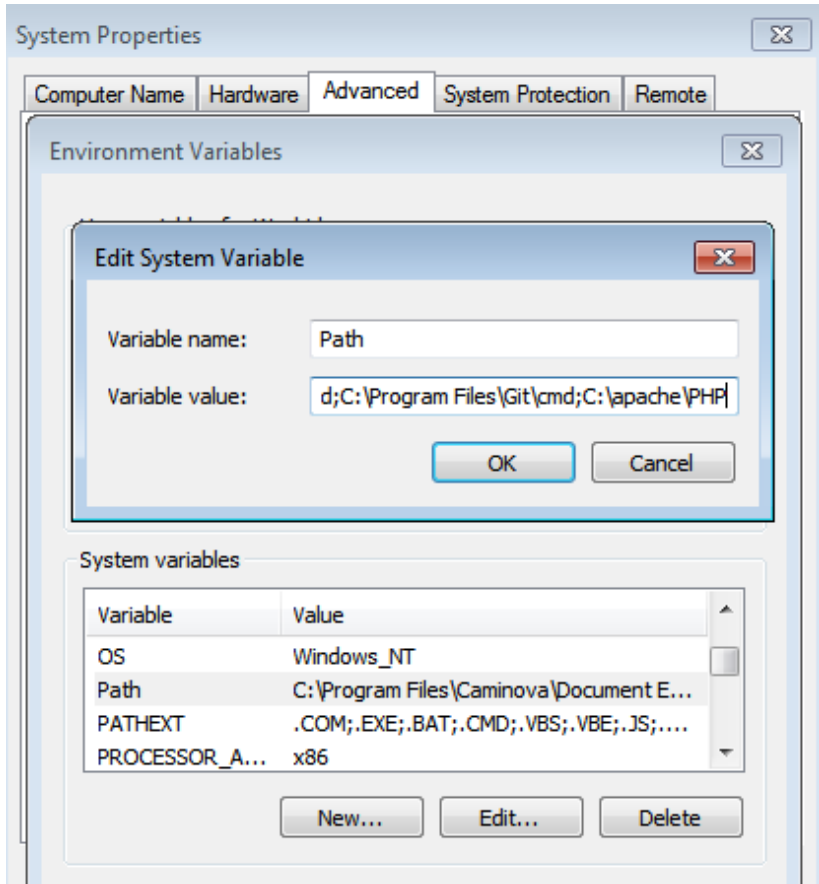
4. Menambahkan PHP ke *path System*. Buka *System Properties*, untuk membukanya bisa degan cara klik kanan pada *My Computer* dan klik *Properties*, pada *Home Menu Control Panel*

klik *Advanced System Settings*. Pada tab *Advanced* Klik *Environment Variables*.



Gambar 4.41 System Properties

5. Dalam "*System Variable*" double klik "*Path*" *Edit variabel* dan tambahkan *path* "*C:\Apache\php*" kedalam nilai *variabel*, click OK untuk menyimpan konfigurasi.



Gambar 4.42 Edit System Variabel

6. Komputer harus *restart* untuk melihat *effect*-nya. Menghubungkan *Apache Server* dengan *PHP*. Untuk menghubungkan *Apache Server* dengan *PHP* kita harus mengedit *Apache Configuration* yang terletak di `C:\Apache\conf\httpd.conf` Cari *Directory Index* seperti berikut:

```
<IfModule dir_module> DirectoryIndex index.html
</IfModule>
```


Tambahkan index.php seperti berikut ini

```
<IfModule dir_module> DirectoryIndex index.html  
index.php </IfModule>
```

```
237 # DirectoryIndex: sets the file that Apache will serve if  
238 # is requested.  
239 #  
240 <IfModule dir_module>  
241     DirectoryIndex index.html index.php  
242 </IfModule>  
243  
244 #
```

Gambar 4.43 Menambahkan Source Code

Tambahkn baris berikut diantara <IfModule mime_module> dan </ IfModule>

```
ScriptAlias    /php/      "C:/Apache/php/"    AddType  
application/x-httpd-php    .php    .php5    Action  
application/x-httpd-php    "/php/php-cgi.exe"  SetEnv  
PHPRC "C:/Apache/php"
```

```
382 AddType application/x-compress .Z  
383 AddType application/x-gzip .gz .tgz  
384  
385 ScriptAlias /php/ "C:/Apache/php/"  
386 AddType application/x-httpd-php .php .php5  
387 Action application/x-httpd-php "/php/php-cgi.exe"  
388 SetEnv PHPRC "C:/Apache/php"  
389  
390 #
```

Gambar 4.44 Menambahkan Source Code

Tamabhkan baris berikut pada bagian akhir httpd.conf

```
PHPIniDir "C:/Apache/php/" LoadModule php5_module  
"C:/Apache/php/php5apache2_2.dll"
```

```

493 <IfModule ssl_module>
494 SSLRandomSeed startup builtin
495 SSLRandomSeed connect builtin
496 </IfModule>
497
498 PHPIniDir "C:/Apache/php/"
499 LoadModule php5_module "C:/Apache/php/php5apache2_2.dll"

```

Gambar 4.45 Menambahkan Source Code

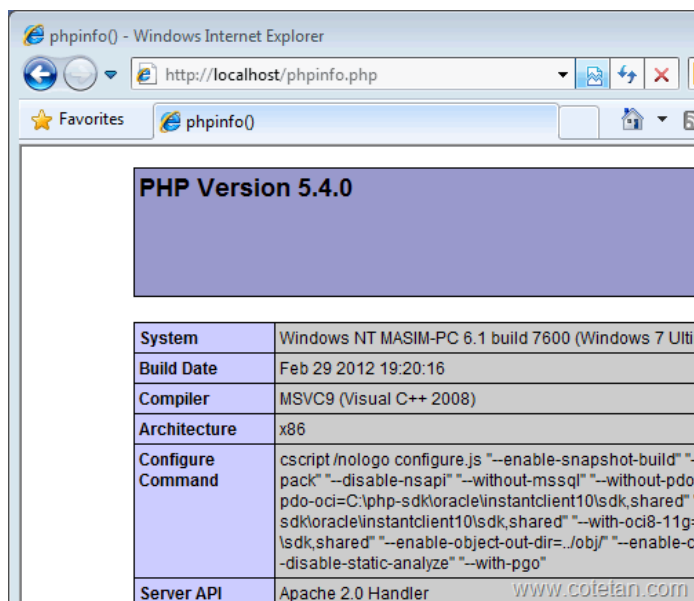
Jika menggunakan Apache v2.4, ubahlah baris terakhir *module* yang unggah "php5apache2_4.dll" .

LoadModule	php5_module
"C:/Apache/php/php5apache2_4.dll"	

7. *Restart Apache Server, test PHP*. Buka *Notepad*, *copy paste* kode dibawah ini:

<?php phpinfo(); ?>

8. Simpan *file* kedalam *root* dokumen Apache "D:\My Websites" as "phpinfo.php" Buka *browser*, pada *address bar* ketikan "http://localhost/phpinfo.php" dan tekan Enter.



Gambar 4.46 Test PHP

BAB V

TUTORIAL PEMBUATAN APLIKASI

Sebelum melakukan tutorial penulis akan menjelaskan sedikit pemaparan latar belakang mengapa sistem manajemen aplikasi arsip surat masuk dan surat keluar ini dibuat dan hasil yang diperoleh dari sistem yang sudah berhasil dibuat dan diuji. Model pengembangan yang digunakan pada penelitian ini ialah model *Waterfall*. Model *waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial. Alasan penggunaan model *waterfall* sebagai metode pengembangan sistem informasi ialah kebutuhan terdefinisi secara jelas dan tahap-tahap pada model *waterfall* terstruktur secara jelas. Dengan penerapan sistem kearsipan *Alphabetical Filing System* dan *Chronology Filing System* mempermudah pencarian surat masuk dan surat keluar, karena kedua sistem itu dapat mem-*filter* data secara huruf dan waktu.

Bagian *Assessment Center* pada PT. Pos Indonesia (Persero) merupakan SDM perusahaan. *Assessment Center* merupakan suatu metoda untuk mengidentifikasi dan menjaring pegawai yang dinilai memiliki potensi dari sisi manajerial (*managerial skill*) untuk menduduki suatu jabatan tertentu di kemudian hari (*future responsibility*). Kegiatan pengiriman dan penerimaan surat juga menjadi kegiatan bagian ini. Namun saat ini dalam pelaksanaannya pengelolaan surat banyak mengalami kendala. Seperti pada saat pengagendaan surat masuk yang diterima dan surat keluar yang akan dikirimkan, penomoran surat sulit dilakukan karena surat dicatat pada buku agenda konvensional. Pengarsipan dokumen fisik surat belum optimal. Terjadi kehilangan pada dokumen surat masuk. Serta media penyimpanan yang membutuhkan tempat yang luas karena

menampung dokumen tertulis yang bertambah setiap harinya, dan terjadi kesalahan pada proses penomoran surat. Berdasarkan permasalahan tersebut, Bagian *Assessment Center* membutuhkan sebuah sistem aplikasi untuk mengelola arsip surat masuk dan surat keluar.

5.1. Mempersiapkan *Tools*

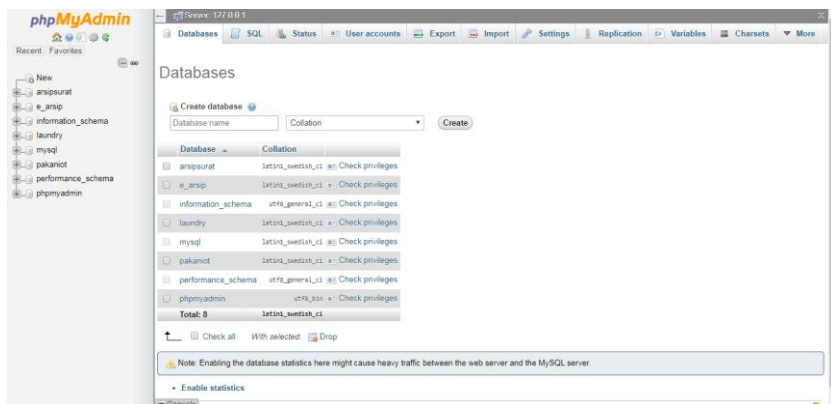
Untuk membuat sistem manajemen aplikasi arsip surat masuk dan surat keluar ini, penulis memerlukan beberapa *tools* yang dapat mendukung dalam pembuatannya dan *tools* ini sudah dibahas pada bab sebelumnya, berikut *tools* yang harus dipersiapkan dalam pembuatan sistem:

1. *Sublime Text 3*
2. *Framework Code Igniter*
3. Bahasa pemrograman *Hypertext Presprocessor* (PHP)
4. XAMPP

5.2. Membuat *database*

Dalam memulai pembuatan *database*, ada beberapa step diantaranya:

1. Buka *PHPMysqladmin*, kemudian buatlah *database* baru dengan nama “e_arsip”.

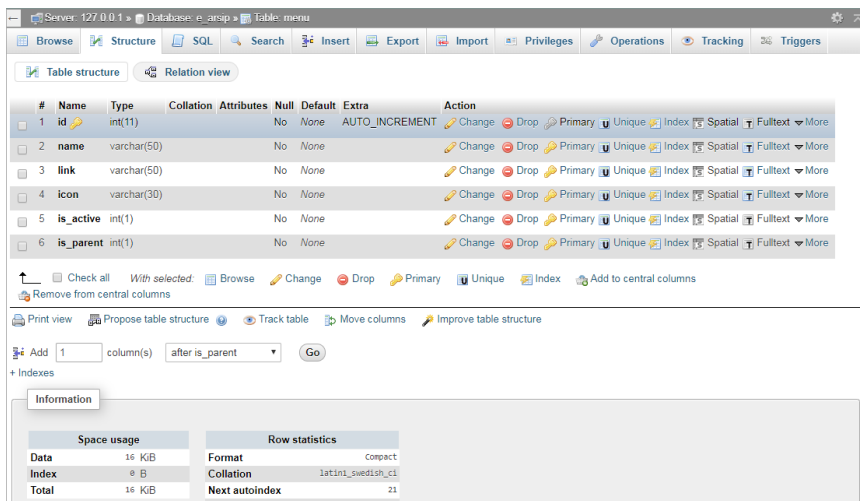


Gambar 5.47 Database *e_arsip*

2. Setelah itu, buat tabel “menu” dengan 6 kolom. Tabel ini nanti akan menyimpan data nama menu atau data menu list aplikasi.

Kolom yang dibutuhkan:

- id, bertipe data *string* dengan panjang 11 (Primary Key)
- name, bertipe data varchar dengan panjang 50
- link, bertipe data varchar dengan panjang 50
- icon, bertipe data varchar dengan panjang 30
- is_active, bertipe data integer
- is_parent, bertipe data integer



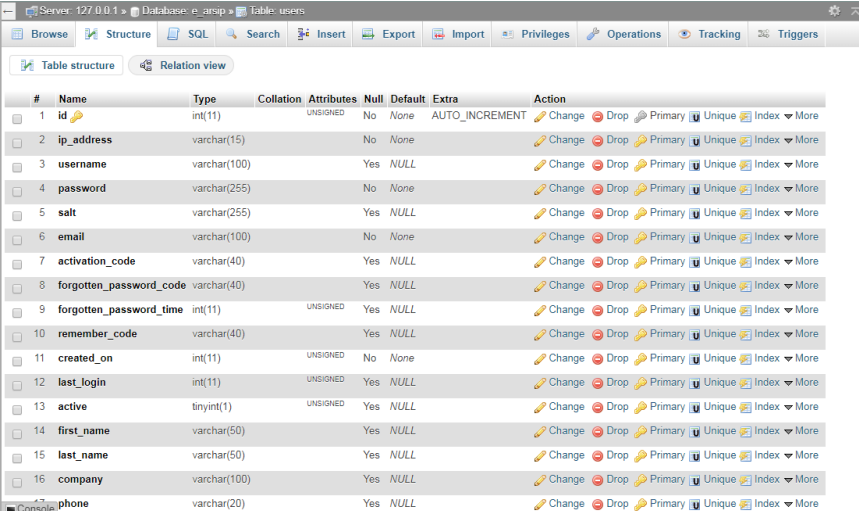
Gambar 5.48 Tabel Menu

3. Setelah itu, buat tabel “users” dengan 17 kolom. Tabel ini nanti akan menyimpan data nama *users*.

Kolom yang dibutuhkan:

- id, bertipe data integer dengan panjang 11 (Primary Key)
- ip_address, bertipe data varchar dengan panjang 15
- username, bertipe data varchar dengan panjang 100
- password, bertipe data varchar dengan panjang 255

- salt, bertipe data varchar dengan panjang 255
- email, bertipe data varchar dengan panjang 100
- activation_code, bertipe data varchar dengan panjang 40
- forgotten_password_code, bertipe data varchar dengan panjang 40
- forgotten_password_time, bertipe data integer dengan panjang 40
- remember_code, bertipe data varchar dengan panjang 40
- created_on, bertipe data integer dengan panjang 11
- last_login, bertipe data integer dengan panjang 11
- active, bertipe data tinyint dengan panjang 1
- first_name, bertipe data varchar dengan panjang 50
- last_name, bertipe data varchar dengan panjang 50
- company, bertipe data varchar dengan panjang 100
- phone, bertipe data varchar dengan panjang 20



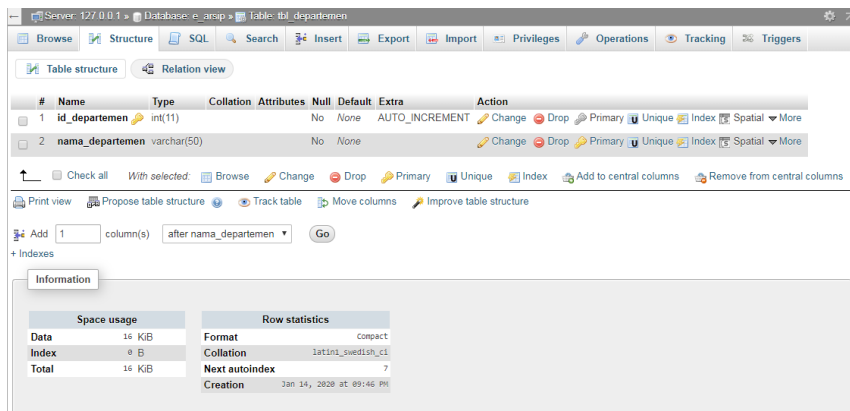
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)		UNSIGNED	No	None	AUTO_INCREMENT	Change Drop Primary Unique Index More
2	ip_address	varchar(15)			No	None		Change Drop Primary Unique Index More
3	username	varchar(100)			Yes	NULL		Change Drop Primary Unique Index More
4	password	varchar(255)			No	None		Change Drop Primary Unique Index More
5	salt	varchar(255)			Yes	NULL		Change Drop Primary Unique Index More
6	email	varchar(100)			No	None		Change Drop Primary Unique Index More
7	activation_code	varchar(40)			Yes	NULL		Change Drop Primary Unique Index More
8	forgotten_password_code	varchar(40)			Yes	NULL		Change Drop Primary Unique Index More
9	forgotten_password_time	int(11)		UNSIGNED	Yes	NULL		Change Drop Primary Unique Index More
10	remember_code	varchar(40)			Yes	NULL		Change Drop Primary Unique Index More
11	created_on	int(11)		UNSIGNED	No	None		Change Drop Primary Unique Index More
12	last_login	int(11)		UNSIGNED	Yes	NULL		Change Drop Primary Unique Index More
13	active	tinyint(1)		UNSIGNED	Yes	NULL		Change Drop Primary Unique Index More
14	first_name	varchar(50)			Yes	NULL		Change Drop Primary Unique Index More
15	last_name	varchar(50)			Yes	NULL		Change Drop Primary Unique Index More
16	company	varchar(100)			Yes	NULL		Change Drop Primary Unique Index More
17	phone	varchar(20)			Yes	NULL		Change Drop Primary Unique Index More

Gambar 5.49 Tabel Users

4. Setelah itu, buat tabel “tbl_departemen” dengan 2 kolom. Tabel ini nanti akan menyimpan data nama departemen atau data nama perusahaan.

Kolom yang dibutuhkan:

- id_departemen, bertipe data integer dengan panjang 11
- nama_departemen, bertipe data varchar dengan 50

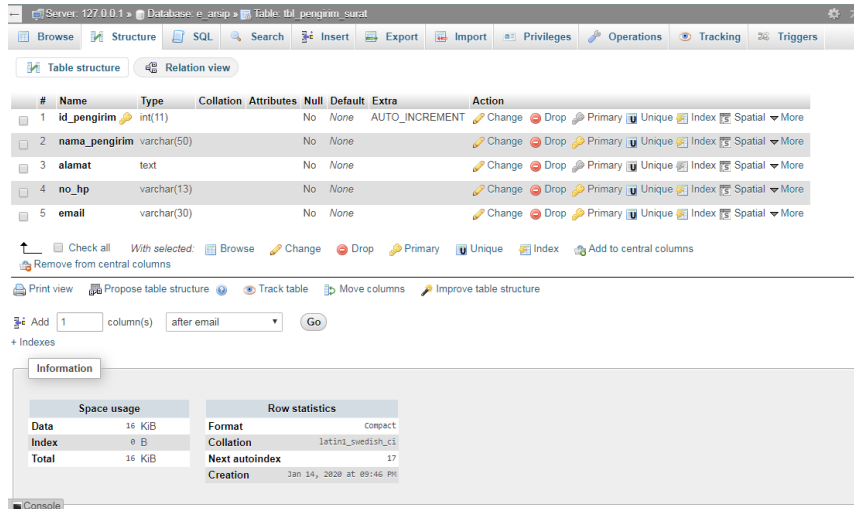


Gambar 5.50 Tabel tbl_departemen

5. Setelah itu, buat tabel “tbl_pengirim_surat” dengan 5 kolom. Tabel ini nanti akan menyimpan data nama pengirim surat atau nama yang menandatangani surat tersebut.

Kolom yang digunakan:

- id_pengirim, bertipe data integer dengan panjang 11
- nama_pengirim, bertipe data varchar dengan panjang 50
- alamat, bertipe data text
- no_hp, bertipe data varchar dengan panjang 13
- email, bertipe data varchar dengan panjang 30



Gambar 5.51 Tabel *tbl_pengirim_surat*

6. Setelah itu, buat tabel “tbl_pengirim_surat” dengan 5 kolom. Tabel ini nanti akan menyimpan data nama pengirim surat atau nama yang menandatangani surat tersebut.

Kolom yang digunakan:

- id_arsip, bertipe data integer dengan panjang 11
- no_surat, bertipe data varchar dengan panjang 30
- tanggal_surat, bertipe data date
- tanggal_diterima, bertipe data date
- tanggal_dikirim, bertipe data date
- perihal, bertipe data varchar dengan panjang 100
- id_departemen, bertipe data integer 11
- id_pengirim, bertipe data integer 11
- file, bertipe data text
- jenis_surat, bertipe data varchar dengan panjang 20

The screenshot shows a database management interface with the 'Table structure' tab selected for 'tbl_arsip'. The table has 10 columns with the following details:

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id_arsip	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	no_surat	varchar(30)			No	None		Change Drop Primary Unique Index Spatial More
3	tanggal_surat	date			No	None		Change Drop Primary Unique Index Spatial More
4	tanggal_diterima	date			No	None		Change Drop Primary Unique Index Spatial More
5	tanggal_dikirim	date			No	None		Change Drop Primary Unique Index Spatial More
6	perihal	varchar(100)			No	None		Change Drop Primary Unique Index Spatial More
7	id_departemen	int(11)			No	None		Change Drop Primary Unique Index Spatial More
8	id_pengirim	int(11)			No	None		Change Drop Primary Unique Index Spatial More
9	file	text			No	None		Change Drop Primary Unique Index Spatial More
10	jenis_surat	varchar(20)			No	None		Change Drop Primary Unique Index Spatial More

Below the table structure, there are options to 'Check all', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', and 'Add to central columns'. At the bottom, there is a section for 'Indexes' with a '+ Indexes' button and a table structure editor showing 'column(s)' and 'after jenis_surat'.

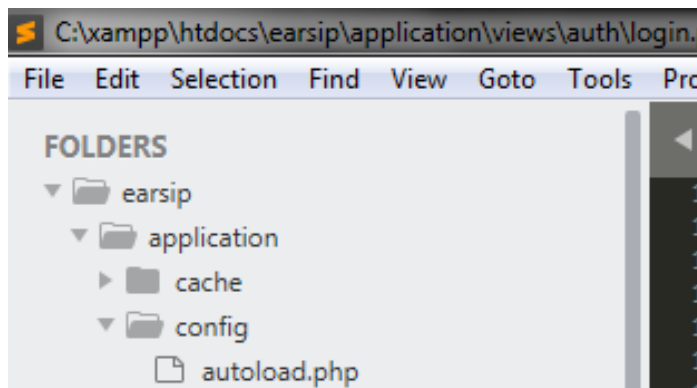
Gambar 5.52 Tabel *tbl_arsip*

5.3. Konfigurasi *CodeIgniter*

Dalam memulai *codeigniter*, ada beberapa konfigurasi dasar yang perlu dilakukan, diantaranya yaitu *autoload.php*, *config.php*, dan *database.php*:

1. *autoload.php*

File ini digunakan untuk mengatur fungsi-fungsi yang akan dimuat otomatis di awal ketika program dijalankan. Untuk melakukan konfigurasi pada *file* *autoload.php*, silahkan buka *folder* *application/config/autoload.php*



Gambar 5.61 *autoload.php*

Ada beberapa hal yang bisa di-load secara otomatis diantaranya: *packages, libraries, drivers, helper files, custom config files, language files*, dan *models*. Untuk konfigurasi dasar adalah *libraries* dan *helper files*. Hal ini bertujuan agar beberapa *library* dan *helper* tertentu berjalan secara otomatis. Untuk melakukan konfigurasi pada *libraries*, buka file *autoload.php* dengan *text editor* seperti *notepad++*, *sublime text*, atau lainnya.

Kemudian temukan kode berikut:

```
$autoload['libraries'] = array();
```

Atur menjadi seperti berikut:

```
$autoload['libraries'] =  
array('database', 'session', 'template', 'ion_auth');
```

Pada kode diatas, artinya kita me-load *library* “*database*” secara otomatis. Dengan demikian dapat menggunakan fungsi-fungsi *database* pada *codeigniter*. Selanjutnya, untuk melakukan konfigurasi pada *helper files*, buka file *autoload.php* dengan *text editor*.

Kemudian temukan kode berikut:

```
$autoload['helper'] = array();
```

Atur menjadi seperti berikut:

```
$autoload['helper'] = array('url', 'form', 'nuris');
```

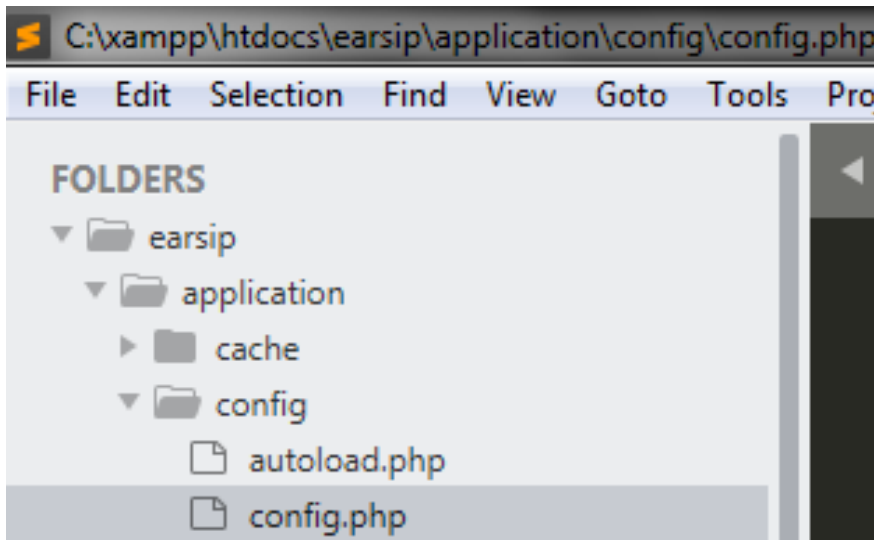
Pada kode diatas, artinya kita me-load *helper* “*url*” secara otomatis. Dengan demikian dapat menggunakan fungsi-fungsi *url* pada *codeigniter*. Seperti fungsi: *base_url()*, *site_url()*, *URI Segment*, dan sebagainya.

2. config.php

Pada *file* ini terdapat beberapa konfigurasi yang secara *standar* sudah terkonfigurasi. Namun terdapat beberapa konfigurasi yang perlu diperhatikan yaitu:

```
$config['base_url']  
$config['index_page']  
$config['encryption_key']
```

Untuk konfigurasi dasar, cukup mengetahui konfigurasi `base_url`. `base_url` merupakan *url* dasar dari *project*. Untuk mengkonfigurasi `base_url`, buka file `config.php` dengan text editor.



Gambar 5.62 *config.php*

Kemudian temukan kode berikut:

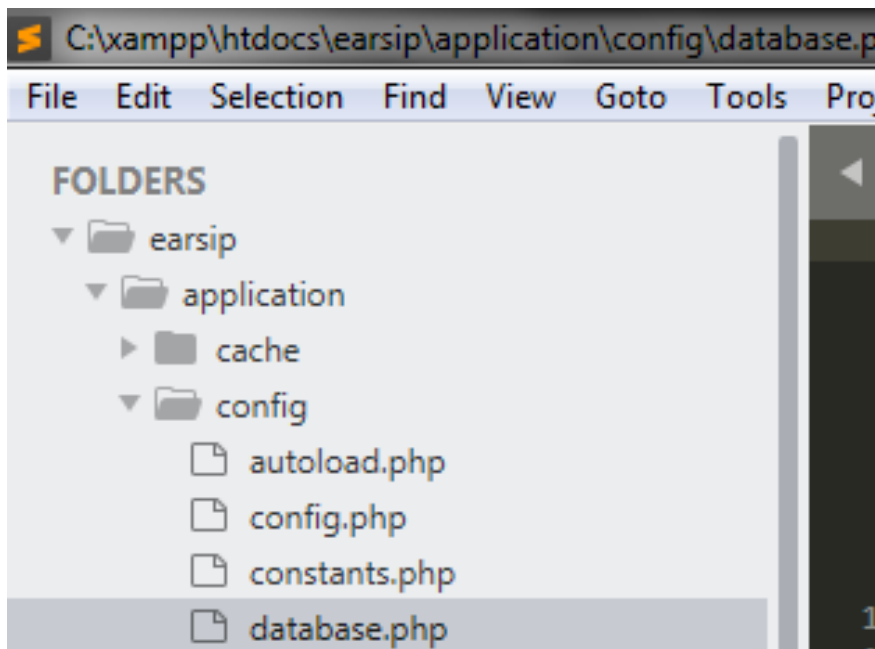
```
$config['base_url'] = '';
```

Atur menjadi seperti berikut:

```
$config['base_url'] = 'http://localhost/earsip/';
```

3. database.php

Dilihat dari nama *filenya* maka sudah dapat menangkap apa fungsi dari *file* ini. *File database.php* digunakan untuk melakukan konfigurasi yang berkaitan dengan konfigurasi *database* dari *website* yang akan dibuat. Adapun konfigurasi yang perlu diperhatikan yaitu: *hostname*, *username*, *password*, dan *database*. Untuk melakukan konfigurasi pada *database.php*. Buka *file database.php* dengan *text editor*.



Gambar 5.63 database.php

Kemudian temukan kode berikut:

```
$active_group = 'default';  
$query_builder = TRUE;  
  
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => '',
```

```

'password' => '',
'database' => '',
'dbdriver' => 'mysqli',
'dbprefix' => '',
'pconnect' => FALSE,
'db_debug' => (ENVIRONMENT !== 'production'),
'cache_on' => FALSE,
'cachedir' => '',
'char_set' => 'utf8',
'dbcollat' => 'utf8_general_ci',
'swap_pre' => '',
'encrypt' => FALSE,
'compress' => FALSE,
'stricton' => FALSE,
'failover' => array(),
'save_queries' => TRUE
);

```

Atur menjadi seperti berikut:

```

$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'e_arsip',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),

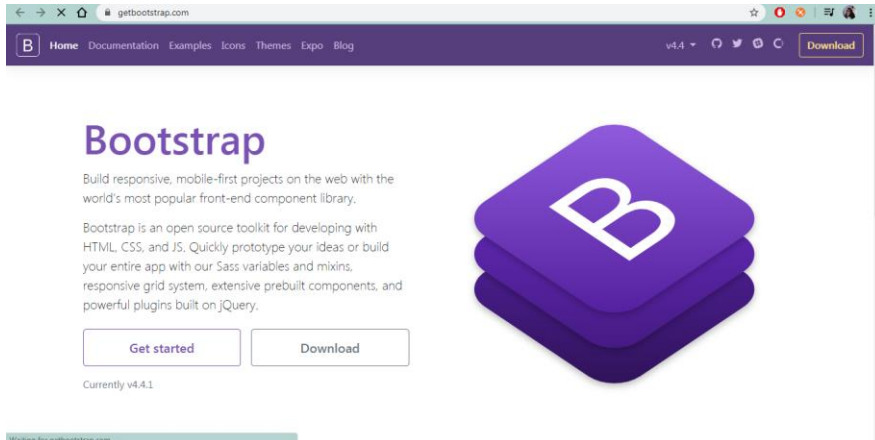
```

```
'cache_on' => FALSE,  
'cachedir' => '',  
'char_set' => 'utf8',  
'dbcollat' => 'utf8_general_ci',  
'swap_pre' => '',  
'encrypt' => FALSE,  
'compress' => FALSE,  
'stricton' => FALSE,  
'failover' => array(),  
'save_queries' => TRUE  
);
```

5.4. Konfigurasi *Template CSS Bootstrap*

Ada hal yang sangat penting untuk ketahui, yaitu mengkombinasikan *codeigniter* dengan *bootstrap*. Apa itu *bootstrap*? *Bootstrap* adalah *toolkit open source* untuk dikembangkan dengan *HTML*, *CSS*, dan *JS*. Dengan kata lain, *bootstrap* merupakan *framework* untuk mempercantik *user interface (UI)*. *Bootstrap* bersifat *responsive*. Dengan kata lain, merender dengan baik di berbagai macam perangkat (*platform*) seperti *tablet* maupun *mobile phone*.

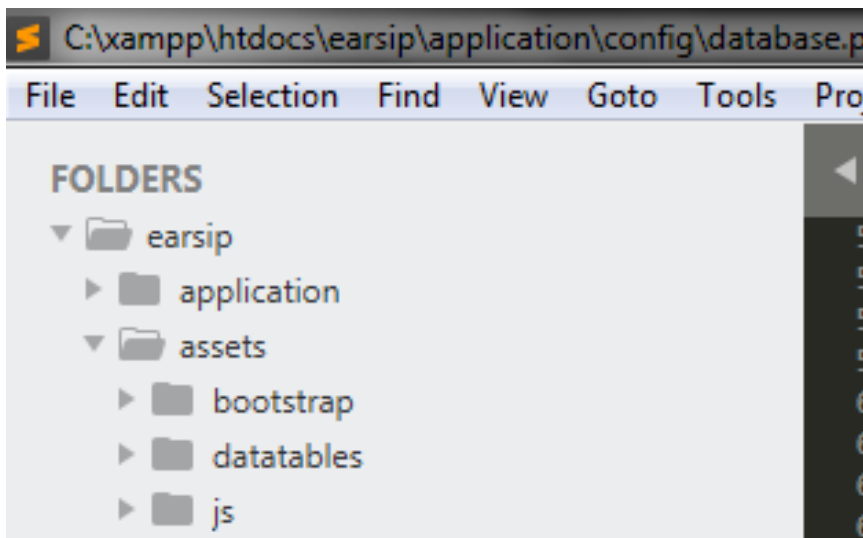
1. Pertama-tama, silahkan *download bootstrap* di situs resminya getbootstrap.com.



Gambar 5.64 Tampilan getbootstrap.com

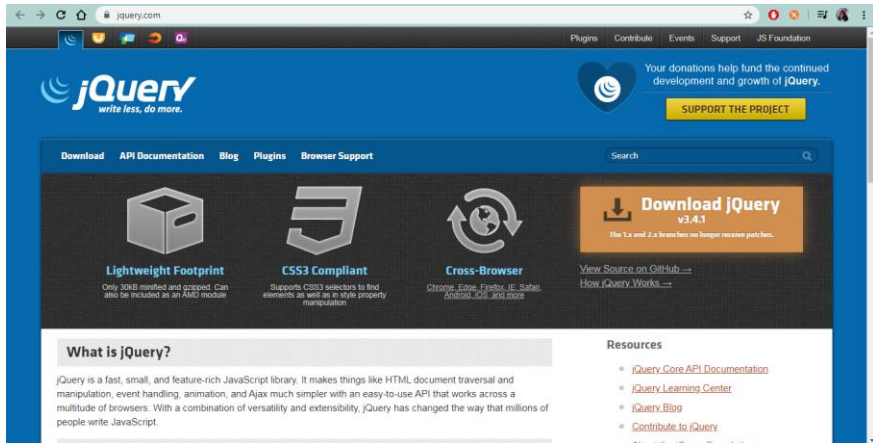
2. Kedua, buat sebuah *folder* baru pada *project* (*webroot*).

Disini saya beri nama *folder* “*assets*”. Kemudian *extract file* *bootstrap* yang telah di *download* tadi kedalam *folder assets*. Seperti berikut:



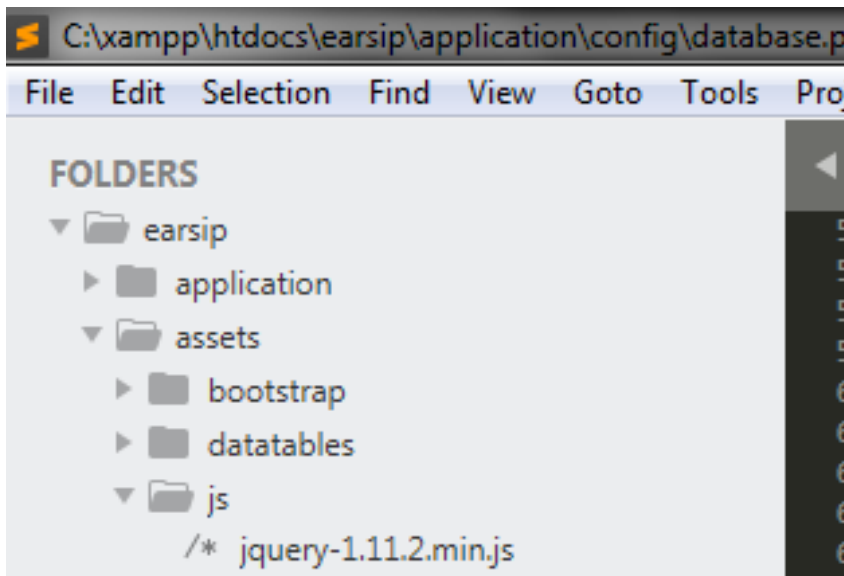
Gambar 5.65 Folder *assets* pada Text Editor

Selain *bootstrap*, kita juga membutuhkan *jquery* agar *javascript* pada *bootstrap* berjalan dengan optimal. Untuk mendownload *jquery*, silahkan *download* di situs resminya jquery.com.



Gambar 5.66 Tampilan jquery.com

Kemudian, letakkan *file jquery* pada *folder assets/js/* seperti gambar berikut:



Gambar 5.67 Folder assets/js/

3. Buka file *template.php* yang terletak pada *application/views*. *File template.php* merupakan tampilan awal setelah menginstall *CodeIgniter*. Kemudian hapus tag yang ada kecuali skrip PHP yang berada paling atas file:


```
<?php defined('BASEPATH') OR exit('No direct script  
access allowed'); ?>
```

Buka salah satu tampilan dari template yang unduh, kemudian memilih login.html yang akan menjadi tampilan awal saat dijalankan. Salin tag pada file tersebut dan pindahkan ke template.php. Sesuaikan beberapa hal seperti titlenya. Kemudian cari tag yang berisi tag untuk meload css dan JavaScript

```
<linkhref="css/bootstrap.min.css" rel="stylesheet">  
dan ubah menjadi <link href="<?php echo  
base_url('assets/css/bootstrap.min.css')?>"  
rel="stylesheet">
```

base_url merupakan salah satu fungsi dari *helper URL* dan *url* tersebut ditambahkan assets karena *folder* dan *file* pendukung diletakkan pada *folder assets*.

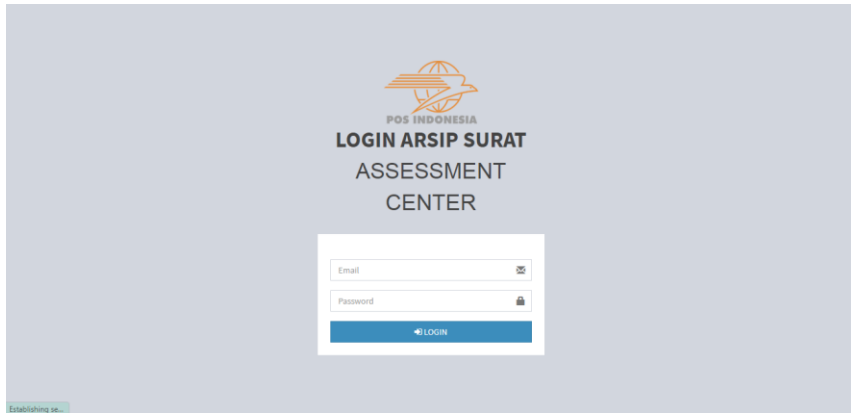
```
<script src="<?php echo base_url();  
?>assets/plugins/jquery/jquery-  
2.1.4.min.js"></script>  
  
<script src="<?php echo base_url();  
?>assets/bootstrap/js/bootstrap.min.js"></script>  
  
<script src="<?php echo base_url();  
?>assets/plugins/iCheck/ichack.min.js"></script>  
    <script>  
        $(function () {  
            $('input').iCheck({  
                checkboxClass:  
'icheckbox_square-blue',  
                radioClass: 'iradio_square-  
blue',  
                increaseArea: '20%' // optional
```

```

    });

});
</script>

```



Gambar 5.68 Tampilan Halaman Login

```

<link rel="stylesheet" href="<?php echo base_url()
?>template/bootstrap/css/bootstrap.min.css">

<link rel="stylesheet" href="<?php echo base_url()
?>template/font-awesome-4.4.0/css/font-
awesome.min.css">

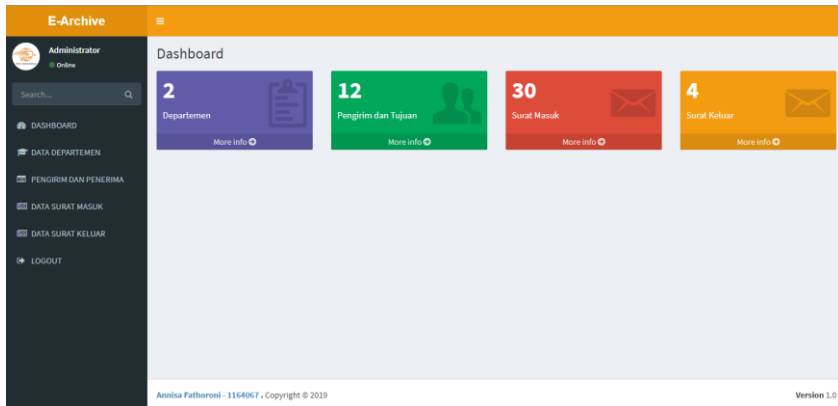
<link rel="stylesheet" href="<?php echo base_url()
?>template/ionicons-2.0.1/css/ionicons.min.css">

<link rel="stylesheet" href="<?php echo base_url()
?>template/plugins/datatables/dataTables.bootstrap.
css">

<link rel="stylesheet" href="<?php echo base_url()
?>template/dist/css/AdminLTE.min.css">

```

```
<link rel="stylesheet" href="<?php echo base_url()  
>template/dist/css/skins/_all-skins.min.css">
```



Gambar 5.68 Tampilan Halaman Dashboard

Selain *file CSS*, *base_url* juga dapat digunakan untuk memanggil *file JavaScript* dan *file gambar*, video dan suara yang merupakan pendukung dalam membuat tampilan.

5.5. Penjelasan Source Code Sistem Aplikasi

Beberapa penjelasan *source code* sistem aplikasi, diantaranya:

1. Controllers

Controllers bisa dikatakan sebagai otak dari sistem. Karena *controllers* yang menjadi penghubung antara bagian *model* dan *view*. *Controllers* berfungsi untuk menerima *request* dan data dari *user* kemudian diproses dengan menghubungkan bagian *model* dan *view* sehingga bisa di terima oleh *user*. Berikut *controllers* yang terdapat pada sistem aplikasi, diantaranya:

a. auth.php

auth.php merupakan *controllers* dari halaman *login*. *Function login* untuk pengaturan dalam proses login pada sistem dan *function*

logout untuk pengaturan dalam proses logout pada sistem, serta dengan menggunakan *form validation* untuk peng-*inputan* data.

Berikut *source code* untuk halaman *login* dan *logout*:

```
function login()
{
    $this->data['title'] = "Login";

    $this->form_validation-
>set_rules('identity', 'Identity', 'required');
    $this->form_validation-
>set_rules('password', 'Password', 'required');

    if ($this->form_validation->run() ==
true)
    {
        $remember = (bool) $this->input-
>post('remember');

        if ($this->ion_auth->login($this-
>input->post('identity'), $this->input-
>post('password'), $remember))
        {

            $this->session-
>set_flashdata('message', $this->ion_auth-
>messages());

            redirect('dashboard',
'refresh');

        }
        else
        {
```

```

                                $this->session-
>set_flashdata('message', $this->ion_auth-
>errors());

                                redirect('auth/login',
'refresh');

                                }
                                }
                                else
                                {

                                $this->data['message'] =
(validation_errors()) ? validation_errors() : $this-
>session->flashdata('message');

                                $this->data['identity'] =
array('name' => 'identity',
                                'id'      => 'identity',
                                'class' => 'form-
control',

'placeholder'=>'Email',
                                'type'   => 'text',
                                'value' => $this-
>form_validation->set_value('identity'),
                                );
                                $this->data['password'] =
array('name' => 'password',
                                'id'     => 'password',
                                'type'   => 'password',
                                'class' => 'form-
control',

'placeholder'=>'Password',

```

```

);

$this->_render_page('auth/login',
$this->data);
}
}

```

```

function logout()
{
    $this->data['title'] = "Logout";

    $logout = $this->ion_auth->logout();

    $this->session->set_flashdata('message',
$this->ion_auth->messages());
    redirect('auth/login', 'refresh');
}

```

b. Dashboard.php

Dashboard.php merupakan *controllers* dari halaman *dashboard*. Yang menggunakan function index untuk mengatur tampilan yang terdapat pada *dashboard*.

Berikut source code untuk halaman *dashboard*:

```

public function index()
{

    $jenis_surat = 'surat_masuk';
    $data = array(
        'pengirim' => site_url('pengirim'),
        'departemen' =>
site_url('departemen'),

```

```

        'surat_masuk' =>
site_url('surat_masuk'),
        'surat_keluar' =>
site_url('surat_keluar')
    );
    $data['total_departemen'] = $this->Dashboard_model->total_departemen();
    $data['total_pengirim'] = $this->Dashboard_model->total_pengirim();
    $data['total_surat_masuk'] = $this->Dashboard_model->total_surat_masuk();
    $data['total_surat_keluar'] = $this->Dashboard_model->total_surat_keluar();

    $this->template->load('template', 'dashboard_view', $data);
}

}

```

c. Departemen.php

Departemen.php merupakan *controllers* dari halaman departemen. *Function create* digunakan untuk membuat *form input* data, *function update* untuk melakukan perubahan data pada *form* data yang telah di-*inputkan* sebelumnya, *function create_update* merupakan *form input* perubahan pada sistem, *function rules* dan *function delete* melakukan hapus data pada sistem, serta *form validation* untuk peng-*inputan* data.

Berikut *source code* untuk halaman departemen:

```

public function create()
{

```

```

        $data = array(
            'button' => 'Create',
            'action' =>
site_url('departemen/create_action'),
            'id_departemen' =>
set_value('id_departemen'),
            'nama_departemen' =>
set_value('nama_departemen'),
        );

        $this->template-
>load('template','tbl_departemen_form', $data);
    }

    public function create_action()
    {
        $this->_rules();

        if ($this->form_validation->run() ==
FALSE) {
            $this->create();
        } else {
            $data = array(
                'nama_departemen' => $this->input-
>post('nama_departemen',TRUE),
            );

            $this->Tbl_departemen_model-
>insert($data);

            $this->session-
>set_flashdata('message', 'Create Record
Success');

            redirect(site_url('departemen'));
        }
    }

```



```

    }

    public function update($id)
    {
        $row = $this->Tbl_departemen_model->get_by_id($id);

        if ($row) {
            $data = array(
                'button' => 'Update',
                'action' =>
site_url('departemen/update_action'),
                'id_departemen' =>
set_value('id_departemen', $row->id_departemen),
                'nama_departemen' =>
set_value('nama_departemen', $row->nama_departemen),
            );

            $this->template->load('template','tbl_departemen_form', $data);
        } else {
            $this->session->set_flashdata('message', 'Record Not Found');
            redirect(site_url('departemen'));
        }
    }

    public function update_action()
    {
        $this->_rules();

        if ($this->form_validation->run() ==
FALSE) {

```

```

        $this->update($this->input-
>post('id_departemen', TRUE));
    } else {
        $data = array(
            'nama_departemen' => $this->input-
>post('nama_departemen', TRUE),
        );

        $this->Tbl_departemen_model-
>update($this->input->post('id_departemen',
TRUE), $data);

        $this->session-
>set_flashdata('message', 'Update Record
Success');

        redirect(site_url('departemen'));
    }
}

public function delete($id)
{
    $row = $this->Tbl_departemen_model-
>get_by_id($id);

    if ($row) {
        $this->Tbl_departemen_model-
>delete($id);

        $this->session-
>set_flashdata('message', 'Delete Record
Success');

        redirect(site_url('departemen'));
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
    }
}

```

```

        redirect(site_url('departemen'));
    }
}

public function _rules()
{
    $this->form_validation-
>set_rules('nama_departemen', 'nama departemen',
'trim|required');

    $this->form_validation-
>set_rules('id_departemen', 'id departemen',
'trim');

    $this->form_validation-
>set_error_delimiters('<span class="text-
danger">', '</span>');
}

```

d. Menu.php

Menu.php merupakan *controllers* dari halaman menu. *Function create* digunakan untuk membuat *form input* data, *function update* untuk melakukan perubahan data pada *form* data yang telah di-*inputkan* sebelumnya, *function create_update* merupakan *form input* perubahan pada sistem, *function rules* dan *function delete* melakukan hapus data pada sistem, serta *form validation* untuk peng-*inputan* data.

Berikut *source code* untuk halaman menu:

```

<?php

if (!defined('BASEPATH'))

```

```

        exit('No direct script access allowed');

class Menu extends CI_Controller
{

    function __construct()
    {
        parent::__construct();
        is_login();
        $this->load->model('Menu_model');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $menu = $this->Menu_model->get_all();

        $data = array(
            'menu_data' => $menu
        );

        $this->template-
>load('template','menu_list', $data);
    }

    public function read($id)
    {
        $row = $this->Menu_model->get_by_id($id);
        if ($row) {
            $data = array(
                'id' => $row->id,
                'name' => $row->name,

```

```

        'link' => $row->link,
        'icon' => $row->icon,
        'is_active' => $row->is_active,
        'is_parent' => $row->is_parent,
    );

    $this->template-
>load('template','menu_read', $data);
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('menu'));
    }
}

public function create()
{
    $data = array(
        'button' => 'Create',
        'action' =>
site_url('menu/create_action'),
        'id' => set_value('id'),
        'name' => set_value('name'),
        'link' => set_value('link'),
        'icon' => set_value('icon'),
        'is_active' => set_value('is_active'),
        'is_parent' => set_value('is_parent'),
    );

    $this->template-
>load('template','menu_form', $data);
}

public function create_action()
{

```

```

        $this->_rules();

        if ($this->form_validation->run() ==
FALSE) {

            $this->create();

        } else {

            $data = array(

                'name' => $this->input-
>post('name',TRUE),

                'link' => $this->input-
>post('link',TRUE),

                'icon' => $this->input-
>post('icon',TRUE),

                'is_active' => $this->input-
>post('is_active',TRUE),

                'is_parent' => $this->input-
>post('is_parent',TRUE),

            );

            $this->Menu_model->insert($data);

            $this->session-
>set_flashdata('message', 'Create Record
Success');

            redirect(site_url('menu'));

        }

    }

    public function update($id)
    {

        $row = $this->Menu_model->get_by_id($id);

        if ($row) {

            $data = array(

```

```

        'button' => 'Update',
        'action' =>
site_url('menu/update_action'),
        'id' => set_value('id', $row->id),
        'name' => set_value('name', $row->name),
        'link' => set_value('link', $row->link),
        'icon' => set_value('icon', $row->icon),
        'is_active' => set_value('is_active',
$row->is_active),
        'is_parent' => set_value('is_parent',
$row->is_parent),
    );

    $this->template-
>load('template','menu_form', $data);
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('menu'));
    }
}

public function update_action()
{
    $this->_rules();

    if ($this->form_validation->run() ==
FALSE) {
        $this->update($this->input-
>post('id', TRUE));
    } else {
        $data = array(
            'name' => $this->input-
>post('name', TRUE),

```

```

        'link' => $this->input->post('link',TRUE),
        'icon' => $this->input->post('icon',TRUE),
        'is_active' => $this->input->post('is_active',TRUE),
        'is_parent' => $this->input->post('is_parent',TRUE),
    );

    $this->Menu_model->update($this->input->post('id', TRUE), $data);
    $this->session->set_flashdata('message', 'Update Record Success');
    redirect(site_url('menu'));
}

public function delete($id)
{
    $row = $this->Menu_model->get_by_id($id);

    if ($row) {
        $this->Menu_model->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('menu'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('menu'));
    }
}

```



```

    }

    }

    public function _rules()
    {
        $this->form_validation->set_rules('name',
        'name', 'trim|required');
        $this->form_validation->set_rules('link',
        'link', 'trim|required');
        $this->form_validation->set_rules('icon',
        'icon', 'trim|required');
        $this->form_validation->set_rules('is_active',
        'is active', 'trim|required');
        $this->form_validation->set_rules('is_parent',
        'is parent', 'trim|required');

        $this->form_validation->set_rules('id', 'id',
        'trim');
        $this->form_validation->
        >set_error_delimiters('<span class="text-
        danger">', '</span>');
    }

```

e. Pengirim.php

Pengirim.php merupakan *controllers* dari halaman data pengirim. Dengan menggunakan *function create* untuk membuat *form input* data, *function update* untuk melakukan perubahan data pada *form* data yang telah di-*inputkan* sebelumnya, *function create_update* merupakan *form input* perubahan pada sistem, *function rules* dan *function delete* melakukan hapus data pada sistem, serta dengan menggunakan *form validation* untuk peng-*inputan* data.

Berikut *source code* untuk halaman pengirim:

```
public function create()
{
    $data = array(
        'button' => 'Create',
        'action' =>
site_url('pengirim/create_action'),
        'id_pengirim' => set_value('id_pengirim'),
        'nama_pengirim' =>
set_value('nama_pengirim'),

    );

    $this->template-
>load('template','tbl_pengirim_surat_form',
$data);
}

public function create_action()
{
    $this->_rules();

    if ($this->form_validation->run() ==
FALSE) {
        $this->create();
    } else {
        $data = array(
            'nama_pengirim' => $this->input-
>post('nama_pengirim',TRUE),
        );

        $this->Tbl_pengirim_surat_model-
>insert($data);
    }
}
```

```

        $this->session-
>set_flashdata('message', 'Create Record
Success');

        redirect(site_url('pengirim'));

    }

}

public function update($id)
{
    $row = $this->Tbl_pengirim_surat_model-
>get_by_id($id);

    if ($row) {
        $data = array(
            'button' => 'Update',
            'action' =>
site_url('pengirim/update_action'),
            'id_pengirim' =>
set_value('id_pengirim', $row->id_pengirim),
            'nama_pengirim' =>
set_value('nama_pengirim', $row->nama_pengirim),

        );

        $this->template-
>load('template','tbl_pengirim_surat_form',
$data);

    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');

        redirect(site_url('pengirim'));

    }

}

```

```

        public function update_action()
        {
            $this->_rules();

            if ($this->form_validation->run() ==
FALSE) {
                $this->update($this->input-
>post('id_pengirim', TRUE));
            } else {
                $data = array(
                    'nama_pengirim' => $this->input-
>post('nama_pengirim', TRUE),

                );

                $this->Tbl_pengirim_surat_model-
>update($this->input->post('id_pengirim', TRUE),
$data);

                $this->session-
>set_flashdata('message', 'Update Record
Success');

                redirect(site_url('pengirim'));
            }
        }

        public function delete($id)
        {
            $row = $this->Tbl_pengirim_surat_model-
>get_by_id($id);

            if ($row) {
                $this->Tbl_pengirim_surat_model-
>delete($id);
            }
        }
    }
}

```

```

        $this->session-
>set_flashdata('message', 'Delete Record
Success');

        redirect(site_url('pengirim'));
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('pengirim'));
    }
}

public function _rules()
{
    $this->form_validation-
>set_rules('nama_pengirim', 'nama pengirim',
'trim|required');

    $this->form_validation-
>set_rules('id_pengirim', 'id_pengirim', 'trim');
    $this->form_validation-
>set_error_delimiters('<span class="text-
danger">', '</span>');
}
}

```

f. Surat_keluar.php

Surat_keluar.php merupakan *controllers* dari halaman surat keluar. *Function-function* yang digunakan adalah *function cetak* untuk melakukan aksi cetak atau *print* pada sistem, *function upload* untuk melakukan aksi *upload file* pada sistem, *function create* untuk membuat *form input* data, *function update* untuk melakukan

perubahan data pada *form* data yang telah di-inputkan sebelumnya, *function create_update* merupakan *form input* perubahan pada sistem, *function rules* dan *function delete* untuk penghapusan data pada sistem, serta *form validation* untuk peng-inputan data.

Berikut *source code* untuk halaman menu surat keluar:

```
public function cetak($file)
{

    $data = [
        'file'=>$file
    ];

    $this->load->library('pdf2');
    $this->pdf2->filename = "file.pdf";
    $this->pdf2->load_view('cetak', $data);

}

public function read($id)
{
    $row = $this->Tbl_arsip_keluar_model-
>get_by_id($id);
    if ($row) {
        $data = array(
            'id_arsip' => $row->id_arsip,
            'no_surat' => $row->no_surat,
            'tanggal_surat' => $row->tanggal_surat,
            'tanggal_diterima' => $row-
>tanggal_diterima,
            'perihal' => $row->perihal,
            'id_departemen' => $row->id_departemen,
            'id_pengirim' => $row->id_pengirim,
```

```

        'file' => $row->file,
        'jenis_surat' => $row->jenis_surat,
    );

    $this->template-
>load('template','tbl_arsip_keluar_read', $data);
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat_keluar'));
    }
}

public function create()
{
    $data = array(
        'button' => 'Create',
        'action' =>
site_url('surat_keluar/create_action'),
        'id_arsip' => set_value('id_arsip'),
        'no_surat' => set_value('no_surat'),
        'tanggal_surat' =>
set_value('tanggal_surat'),
        'tanggal_diterima' =>
set_value('tanggal_diterima'),
        'tanggal_dikirim' =>
set_value('tanggal_dikirim'),
        'perihal' => set_value('perihal'),
        'id_departemen' =>
set_value('id_departemen'),
        'id_pengirim' =>
set_value('id_pengirim'),
        'file' => set_value('file'),

```

```

        'jenis_surat' =>
set_value('jenis_surat'),
    );

    $this->template->
load('template','tbl_arsip_keluar_form', $data);
    }

    function upload(){
        $config['upload_path']          =
'./uploads/';
        $config['allowed_types']        =
'gif|jpg|png|jpeg|pdf|zip|rar';

        $this->load->library('upload', $config);
        $this->upload->do_upload('file');
        $result = $this->upload->data();
        return $result;
    }

    function download($filename){
        $this->load->helper('download');

        force_download(base_url().'./uploads/'.$filename,
        NULL);
    }

    public function create_action()
    {

        $upload = $this->upload();
        $this->_rules();
    }

```



```

        if ($this->form_validation->run() ==
FALSE) {

            $this->create();

        } else {

            $data = array(

                'no_surat' => $this->input-
>post('no_surat',TRUE),

                'tanggal_surat' => $this->input-
>post('tanggal_surat',TRUE),

                'tanggal_diterima' => $this->input-
>post('tanggal_diterima',TRUE),

                'perihal' => $this->input-
>post('perihal',TRUE),

                'id_departemen' => $this->input-
>post('id_departemen',TRUE),

                'id_pengirim' => $this->input-
>post('id_pengirim',TRUE),

                'file' => $upload['file_name'],

                'jenis_surat' => $this->input-
>post('jenis_surat',TRUE),

            );

            $this->Tbl_arsip_keluar_model-
>insert($data);

            $this->session-
>set_flashdata('message', 'Create Record
Success');

            redirect(site_url('surat_keluar'));

        }

    }

    public function update($id)
    {

```

```

        $row = $this->Tbl_arsip_keluar_model-
>get_by_id($id);

        if ($row) {
            $data = array(
                'button' => 'Update',
                'action' =>
site_url('surat_keluar/update_action'),
                'id_arsip' =>
set_value('id_arsip', $row->id_arsip),
                'no_surat' =>
set_value('no_surat', $row->no_surat),
                'tanggal_surat' =>
set_value('tanggal_surat', $row->tanggal_surat),
                'tanggal_diterima' =>
set_value('tanggal_diterima', $row-
>tanggal_diterima),
                'tanggal' => set_value('tanggal',
$row->tanggal_dikirim),
                'perihal' => set_value('perihal',
$row->perihal),
                'id_departemen' =>
set_value('id_departemen', $row->id_departemen),
                'id_pengirim' =>
set_value('id_pengirim', $row->id_pengirim),
                'file' => set_value('file'),
                'jenis_surat' =>
set_value('jenis_surat', $row->jenis_surat),
            );

            $this->template-
>load('template','tbl_arsip_keluar_form', $data);
        } else {

```

```

        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat_keluar'));
    }
}

public function update_action()
{
    $upload = $this->upload();
    $this->_rules();

    if ($this->form_validation->run() ==
FALSE) {
        $this->update($this->input-
>post('id_arsip', TRUE));
    } else {
        $data = array(
            'no_surat'           => $this-
>input->post('no_surat',TRUE),
            'tanggal_surat'      => $this-
>input->post('tanggal_surat',TRUE),
            'tanggal_diterima'   => $this-
>input->post('tanggal_diterima',TRUE),
            'tanggal_dikirim'    => $this-
>input->post('tanggal',TRUE),
            'perihal'            => $this-
>input->post('perihal',TRUE),
            'id_departemen'      => $this-
>input->post('id_departemen',TRUE),
            'id_pengirim'        => $this-
>input->post('id_pengirim',TRUE),
            'file'               =>
$upload['file_name'],

```

```

        'jenis_surat' => $this->input->post('jenis_surat', TRUE),
    );

    $this->Tbl_arsip_keluar_model->update($this->input->post('id_arsip', TRUE), $data);

    $this->session->set_flashdata('message', 'Update Record Success');
    redirect(site_url('surat_keluar'));
}

public function delete($id)
{
    $row = $this->Tbl_arsip_keluar_model->get_by_id($id);

    if ($row) {
        $this->Tbl_arsip_keluar_model->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('surat_keluar'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat_keluar'));
    }
}

```

```

        public function _rules()
        {
            $this->form_validation->set_rules('no_surat',
            'no surat', 'trim|required');
            $this->form_validation-
            >set_rules('tanggal_surat', 'tanggal surat',
            'trim|required');
            $this->form_validation->set_rules('perihal',
            'perihal', 'trim|required');
            $this->form_validation-
            >set_rules('id_departemen', 'id departemen',
            'trim|required');
            $this->form_validation-
            >set_rules('id_pengirim', 'id pengirim',
            'trim|required');
            $this->form_validation-
            >set_rules('jenis_surat', 'jenis surat',
            'trim|required');

            $this->form_validation->set_rules('id_arsip',
            'id_arsip', 'trim');
            $this->form_validation-
            >set_error_delimiters('<span class="text-
            danger">', '</span>');
        }

```

g. Surat_masuk.php

Surat_masuk.php merupakan *controllers* dari halaman surat masuk. *Function-function* yang digunakan adalah *function cetak* untuk melakukan aksi cetak atau *print* pada sistem, *function upload* untuk melakukan aksi *upload file* pada sistem, *function create* untuk membuat *form input* data, *function update* untuk melakukan

perubahan data pada *form* data yang telah di-inputkan sebelumnya, *function create_update* merupakan *form input* perubahan pada sistem, *function rules* dan *function delete* untuk penghapusan data pada sistem, serta *form validation* untuk peng-inputan data.

Berikut *source code* untuk halaman menu surat masuk:

```
public function cetak($file)
{

    $data = [
        'file'=>$file
    ];

    $this->load->library('pdf2');
    $this->pdf2->filename = "file.pdf";
    $this->pdf2->load_view('cetak', $data);

}

public function read($id)
{
    $row = $this->Tbl_arsip_masuk_model-
>get_by_id($id);
    if ($row) {
        $data = array(
            'id_arsip' => $row->id_arsip,
            'no_surat' => $row->no_surat,
            'tanggal_surat' => $row->tanggal_surat,
            'tanggal_diterima' => $row-
>tanggal_diterima,
            'perihal' => $row->perihal,
            'id_departemen' => $row->id_departemen,
            'id_pengirim' => $row->id_pengirim,
```

```

        'file' => $row->file,
        'jenis_surat' => $row->jenis_surat,
    );

    $this->template-
>load('template','tbl_arsip_masuk_read', $data);
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat'));
    }
}

public function create()
{
    $data = array(
        'button' => 'Create',
        'action' =>
site_url('surat_masuk/create_action'),
        'id_arsip' => set_value('id_arsip'),
        'no_surat' => set_value('no_surat'),
        'tanggal_surat' =>
set_value('tanggal_surat'),
        'tanggal_diterima' =>
set_value('tanggal_diterima'),
        'perihal' => set_value('perihal'),
        'id_departemen' =>
set_value('id_departemen'),
        'id_pengirim' =>
set_value('id_pengirim'),
        'file' => set_value('file'),
        'jenis_surat' =>
set_value('jenis_surat'),
    );

```

```

        $this->template-
>load('template','tbl_arsip_masuk_form', $data);
    }

    function upload(){
        $config['upload_path']           =
'./uploads/';
        $config['allowed_types']         =
'gif|jpg|png|jpeg|pdf';
        $this->load->library('upload', $config);
        $this->upload->do_upload('file');
        $result = $this->upload->data();
        return $result;
    }

    function download($filename){
        $this->load->helper('download');

        force_download(base_url().'./uploads/'.$filename,
        NULL);
    }

    public function create_action()
    {

        $upload = $this->upload();
        $this->_rules();

        if ($this->form_validation->run() ==
FALSE) {
            $this->create();
        } else {
            $data = array(

```



```

        'no_surat' => $this->input-
>post('no_surat',TRUE),
        'tanggal_surat' => $this->input-
>post('tanggal_surat',TRUE),
        'tanggal_diterima' => $this->input-
>post('tanggal_diterima',TRUE),
        'perihal' => $this->input-
>post('perihal',TRUE),
        'id_departemen' => $this->input-
>post('id_departemen',TRUE),
        'id_pengirim' => $this->input-
>post('id_pengirim',TRUE),
        'file' => $upload['file_name'],
        'jenis_surat' => $this->input-
>post('jenis_surat',TRUE),
    );

    $this->Tbl_arsip_masuk_model-
>insert($data);

    $this->session-
>set_flashdata('message', 'Create Record
Success');

    redirect(site_url('surat_masuk'));
}

}

public function update($id)
{
    $row = $this->Tbl_arsip_masuk_model-
>get_by_id($id);

    if ($row) {
        $data = array(

```

```

        'button' => 'Update',
        'action' =>
site_url('surat_masuk/update_action'),
        'id_arsip' =>
set_value('id_arsip', $row->id_arsip),
        'no_surat' =>
set_value('no_surat', $row->no_surat),
        'tanggal_surat' =>
set_value('tanggal_surat', $row->tanggal_surat),
        'tanggal_diterima' =>
set_value('tanggal_diterima', $row-
>tanggal_diterima),
        'perihal' => set_value('perihal',
$row->perihal),
        'id_departemen' =>
set_value('id_departemen', $row->id_departemen),
        'id_pengirim' =>
set_value('id_pengirim', $row->id_pengirim),
        'file' => set_value('file'),
        'jenis_surat' =>
set_value('jenis_surat', $row->jenis_surat),
    );

    $this->template-
>load('template','tbl_arsip_masuk_form', $data);
    } else {
        $this->session-
>set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat_masuk'));
    }
}

public function update_action()
{

```

```

        $upload = $this->upload();
        $this->_rules();

        if ($this->form_validation->run() ==
FALSE) {

            $this->update($this->input-
>post('id_arsip', TRUE));
        } else {
            $data = array(
                'no_surat'           => $this-
>input->post('no_surat',TRUE),
                'tanggal_surat'      => $this-
>input->post('tanggal_surat',TRUE),
                'tanggal_diterima'   => $this-
>input->post('tanggal_diterima',TRUE),
                'perihal'           => $this-
>input->post('perihal',TRUE),
                'id_departemen'      => $this-
>input->post('id_departemen',TRUE),
                'id_pengirim'        => $this-
>input->post('id_pengirim',TRUE),
                'file'              =>
$upload['file_name'],
                'jenis_surat'        => $this-
>input->post('jenis_surat',TRUE),
            );

            $this->Tbl_arsip_masuk_model-
>update($this->input->post('id_arsip', TRUE),
            $data);

            $this->session-
>set_flashdata('message', 'Update Record
Success');

```

```

        redirect(site_url('surat_masuk'));
    }
}

public function delete($id)
{
    $row = $this->Tbl_arsip_masuk_model->get_by_id($id);

    if ($row) {
        $this->Tbl_arsip_masuk_model->delete($id);
        $this->session->set_flashdata('message', 'Delete Record Success');
        redirect(site_url('surat_masuk'));
    } else {
        $this->session->set_flashdata('message', 'Record Not Found');
        redirect(site_url('surat_masuk'));
    }
}

public function _rules()
{
    $this->form_validation->set_rules('no_surat', 'no surat', 'trim|required');
    $this->form_validation->set_rules('tanggal_surat', 'tanggal surat', 'trim|required');
    $this->form_validation->set_rules('perihal', 'perihal', 'trim|required');
}

```

```

        $this->form_validation-
>set_rules('id_departemen', 'id departemen',
'trim|required');

        $this->form_validation-
>set_rules('id_pengirim', 'id pengirim',
'trim|required');

        $this->form_validation-
>set_rules('jenis_surat', 'jenis surat',
'trim|required');

        $this->form_validation->set_rules('id_arsip',
'id_arsip', 'trim');

        $this->form_validation-
>set_error_delimiters('<span class="text-
danger">', '</span>');

    }

```

2. Model

Model adalah bagian yang berhubungan dengan manipulasi data didalam *database* misalnya *insert*, *create*, *update* dan *delete* dan lainnya. Model ini dihubungkan oleh kontrol aplikasi ke *interface user*. Berikut *model* yang terdapat pada sistem aplikasi, diantaranya:

a. Dashboard_model.php

Dashboard_model.php merupakan *model* dari *dashboard* atau halaman utama yang pertama kali *user* masuk pada sistem aplikasi. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*.

Berikut *source code* untuk halaman Dashboard_model.php:

```

public function total_departemen()
{

    $this->db->from('tbl_departemen');

    return $this->db->count_all_results();
}

public function total_pengirim()
{

    $this->db->from('tbl_pengirim_surat');

    return $this->db->count_all_results();
}

public function total_surat_masuk()
{

    $this->db->from('tbl_arsip');
    $this->db->where('jenis_surat','surat_masuk');

    return $this->db->count_all_results();
}

public function total_surat_keluar()
{

    $this->db->from('tbl_arsip');

```

```

        $this->db->
        >where('jenis_surat','surat_keluar');

        return $this->db->
        >count_all_results();
    }

}

```

b. Ion_auth_model.php

Ion_auth_model.php merupakan *model* dari halaman login atau halaman yang pertama kali muncul pada sistem saat menjalankan aplikasi. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*.

Berikut *source code* untuk halaman Ion_auth_model.php:

```

public function login($identity, $password,
    $remember=FALSE)
    {
        $this->trigger_events('pre_login');

        if (empty($identity) || empty($password))
        {
            $this->
            >set_error('login_unsuccessful');
            return FALSE;
        }

        $this->trigger_events('extra_where');
    }

```

```

        $query      =      $this->db->select($this-
>identity_column . ', email, id, password, active,
last_login')

                                ->where($this-
>identity_column, $identity)
                                ->limit(1)
                                ->order_by('id',
'desc')

                                ->get($this-
>tables['users']);

        if($this->is_time_locked_out($identity))
        {

                                $this-
>hash_password($password);

                                $this-
>trigger_events('post_login_unsuccessful');
                                $this->set_error('login_timeout');

                                return FALSE;
        }

        if ($query->num_rows() === 1)
        {

                $user = $query->row();

                $password      =      $this-
>hash_password_db($user->id, $password);

                if ($password === TRUE)
                {

                        if ($user->active == 0)

```



```

        {
            $this->
>trigger_events('post_login_unsuccessful');
            $this->
>set_error('login_unsuccessful_not_active');

            return FALSE;
        }

        $this->set_session($user);

        $this->
>update_last_login($user->id);

        $this->
>clear_login_attempts($identity);

        if ($remember && $this->
>config->item('remember_users', 'ion_auth'))
        {
            $this->
>remember_user($user->id);
        }

        $this->
>trigger_events(array('post_login',
'post_login_successful'));
        $this->
>set_message('login_successful');

        return TRUE;
    }
}

```

```

        $this->hash_password($password);

        $this->
>increase_login_attempts($identity);

        $this->
>trigger_events('post_login_unsuccessful');
        $this->set_error('login_unsuccessful');

        return FALSE;
    }

```

c. Menu_model.php

Menu_model.php merupakan *model* dari halaman menu yang terdapat pada sistem aplikasi. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*.

Berikut *source code* untuk halaman Menu_model.php:

```

function get_all()
{
    $this->db->order_by($this->id,      $this->
>order);
    return      $this->db->get($this->table)-
>result();
}

function get_by_id($id)
{
    $this->db->where($this->id, $id);
}

```

```

        return      $this->db->get($this->table)-
>row();
    }

    // get total rows
    function total_rows($q = NULL) {
        $this->db->like('id', $q);
        $this->db->or_like('name', $q);
        $this->db->or_like('link', $q);
        $this->db->or_like('icon', $q);
        $this->db->or_like('is_active', $q);
        $this->db->or_like('is_parent', $q);
        $this->db->from($this->table);
        return $this->db->count_all_results();
    }

    function get_limit_data($limit, $start = 0,
    $q = NULL) {
        $this->db->order_by($this->id,      $this-
>order);
        $this->db->like('id', $q);
        $this->db->or_like('name', $q);
        $this->db->or_like('link', $q);
        $this->db->or_like('icon', $q);
        $this->db->or_like('is_active', $q);
        $this->db->or_like('is_parent', $q);
        $this->db->limit($limit, $start);
        return      $this->db->get($this->table)-
>result();
    }

    function insert($data)
    {

```

```

        $this->db->insert($this->table, $data);
    }

    function update($id, $data)
    {
        $this->db->where($this->id, $id);
        $this->db->update($this->table, $data);
    }

    function delete($id)
    {
        $this->db->where($this->id, $id);
        $this->db->delete($this->table);
    }

}

```

d. Tbl_arsip_keluar_model.php

Tbl_arsip_keluar_model.php merupakan *model* dari halaman data arsip surat keluar yang menampilkan *list* dan *create* pada sistem. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*. Yang digabungkan dari beberapa data seperti data departemen atau perusahaan, data pengirim serta data penerima surat, dan data arsip surat keluar yang terdata atau *list* dalam sistem.

Berikut *source code* untuk halaman Tbl_arsip_keluar_model.php:

```

function get_all()
{

```

```

        $this->db->order_by($this->id, $this-
>order);
        return $this->db->get($this->table)-
>result();
    }

    function get_all_query($jenis_surat){
        $sql = "SELECT
a.*,d.nama_departemen,p.nama_pengirim
FROM tbl_arsip AS a,tbl_departemen
as d,tbl_pengirim_surat as p
WHERE
a.id_departemen=d.id_departemen and
a.id_pengirim=p.id_pengirim
and
a.jenis_surat='$jenis_surat'";
        return $this->db->query($sql)->result();
    }

    function get_by_id($id)
    {
        $this->db->where($this->id, $id);
        return $this->db->get($this->table)-
>row();
    }

    function total_rows($q = NULL) {
        $this->db->like('id_arsip', $q);
        $this->db->or_like('no_surat', $q);
        $this->db->or_like('tanggal_surat', $q);
        $this->db->or_like('tanggal_diterima', $q);
        $this->db->or_like('perihal', $q);
        $this->db->or_like('id_departemen', $q);
    }

```

```

$this->db->or_like('id_pengirim', $q);
$this->db->or_like('file', $q);
$this->db->or_like('jenis_surat', $q);
$this->db->from($this->table);
    return $this->db->count_all_results();
}

function get_limit_data($limit, $start = 0, $q
= NULL) {
    $this->db->order_by($this->id,      $this-
>order);
    $this->db->like('id_arsip', $q);
    $this->db->or_like('no_surat', $q);
    $this->db->or_like('tanggal_surat', $q);
    $this->db->or_like('tanggal_diterima', $q);
    $this->db->or_like('perihal', $q);
    $this->db->or_like('id_departemen', $q);
    $this->db->or_like('id_pengirim', $q);
    $this->db->or_like('file', $q);
    $this->db->or_like('jenis_surat', $q);
    $this->db->limit($limit, $start);
    return      $this->db->get($this->table)-
>result();
}

function insert($data)
{
    $this->db->insert($this->table, $data);
}

function update($id, $data)
{
    $this->db->where($this->id, $id);

```

```

        $this->db->update($this->table, $data);
    }

    function delete($id)
    {
        $this->db->where($this->id, $id);
        $this->db->delete($this->table);
    }
}

```

e. Tbl_arsip_masuk_model.php

Tbl_arsip_masuk_model.php merupakan *model* dari halaman data arsip surat masuk yang menampilkan *list* dan *create* pada sistem. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*. Yang digabungkan dari beberapa data seperti data departemen atau perusahaan, data pengirim surat, dan data arsip surat masuk yang terdata atau *list* dalam sistem.

Berikut *source code* untuk halaman Tbl_arsip_masuk_model.php:

```

function get_all()
{
    $this->db->order_by($this->id, $this->
>order);
    return $this->db->get($this->table)-
>result();
}

function get_all_query($jenis_surat){

```

```

        $sql="SELECT
a.*,d.nama_departemen,p.nama_pengirim
        FROM tbl_arsip AS a,tbl_departemen
as d,tbl_pengirim_surat as p
        WHERE
a.id_departemen=d.id_departemen and
a.id_pengirim=p.id_pengirim
        and
a.jenis_surat='$jenis_surat';
        return $this->db->query($sql)->result();
    }

    function get_by_id($id)
    {
        $this->db->where($this->id, $id);
        return $this->db->get($this->table)-
>row();
    }

    function total_rows($q = NULL) {
        $this->db->like('id_arsip', $q);
        $this->db->or_like('no_surat', $q);
        $this->db->or_like('tanggal_surat', $q);
        $this->db->or_like('tanggal_diterima', $q);
        $this->db->or_like('perihal', $q);
        $this->db->or_like('id_departemen', $q);
        $this->db->or_like('id_pengirim', $q);
        $this->db->or_like('file', $q);
        $this->db->or_like('jenis_surat', $q);
        $this->db->from($this->table);
        return $this->db->count_all_results();
    }

```



```

function get_limit_data($limit, $start = 0, $q =
NULL) {
    $this->db->order_by($this->id,      $this->
    >order);
    $this->db->like('id_arsip', $q);
    $this->db->or_like('no_surat', $q);
    $this->db->or_like('tanggal_surat', $q);
    $this->db->or_like('tanggal_diterima', $q);
    $this->db->or_like('perihal', $q);
    $this->db->or_like('id_departemen', $q);
    $this->db->or_like('id_pengirim', $q);
    $this->db->or_like('file', $q);
    $this->db->or_like('jenis_surat', $q);
    $this->db->limit($limit, $start);
    return      $this->db->get($this->table)-
    >result();
}

function insert($data)
{
    $this->db->insert($this->table, $data);
}

function update($id, $data)
{
    $this->db->where($this->id, $id);
    $this->db->update($this->table, $data);
}

function delete($id)
{
    $this->db->where($this->id, $id);
    $this->db->delete($this->table);
}

```

```
}  
  
}
```

f. Tbl_departemen_model.php

Tbl_departemen_model.php merupakan *model* dari halaman data departemen atau perusahaan yang menampilkan *list* dan *create* pada sistem. Pada *code* ini terdapat beberapa *code* atau perintah yang berhubungan dengan sistem *database*, sehingga untuk mengatur serta menampilkan isi dari tampilan ini sesuai dengan apa yang dimasukkan ke dalam *database*.

Berikut *source code* untuk halaman Tbl_departemen_model.php:

```
function get_all()  
{  
    $this->db->order_by($this->id, $this->  
>order);  
    return $this->db->get($this->table)-  
>result();  
}  
  
function get_by_id($id)  
{  
    $this->db->where($this->id, $id);  
    return $this->db->get($this->table)-  
>row();  
}  
  
function total_rows($q = NULL) {  
    $this->db->like('id_departemen', $q);  
    $this->db->or_like('nama_departemen', $q);  
    $this->db->from($this->table);
```

```

        return $this->db->count_all_results();
    }

    function get_limit_data($limit, $start = 0, $q
= NULL) {
        $this->db->order_by($this->id,      $this-
>order);

        $this->db->like('id_departemen', $q);
        $this->db->or_like('nama_departemen', $q);
        $this->db->limit($limit, $start);
        return      $this->db->get($this->table)-
>result();
    }

    function insert($data)
    {
        $this->db->insert($this->table, $data);
    }

    function update($id, $data)
    {
        $this->db->where($this->id, $id);
        $this->db->update($this->table, $data);
    }

    function delete($id)
    {
        $this->db->where($this->id, $id);
        $this->db->delete($this->table);
    }
}

```

g. Tbl_pengirim_surat_model.php

```
function get_all()
{
    $this->db->order_by($this->id, $this->order);
    return $this->db->get($this->table)->result();
}

function get_by_id($id)
{
    $this->db->where($this->id, $id);
    return $this->db->get($this->table)->row();
}

function total_rows($q = NULL) {
    $this->db->like('id_pengirim', $q);
    $this->db->or_like('nama_pengirim', $q);
    $this->db->or_like('alamat', $q);
    $this->db->or_like('no_hp', $q);
    $this->db->or_like('email', $q);
    $this->db->from($this->table);
    return $this->db->count_all_results();
}

function get_limit_data($limit, $start = 0, $q = NULL) {
    $this->db->order_by($this->id, $this->order);
    $this->db->like('id_pengirim', $q);
    $this->db->or_like('nama_pengirim', $q);
```

```

        $this->db->or_like('alamat', $q);
        $this->db->or_like('no_hp', $q);
        $this->db->or_like('email', $q);
        $this->db->limit($limit, $start);

        return $this->db->get($this->table)-
>result();
    }

    function insert($data)
    {
        $this->db->insert($this->table, $data);
    }

    function update($id, $data)
    {
        $this->db->where($this->id, $id);
        $this->db->update($this->table, $data);
    }

    function delete($id)
    {
        $this->db->where($this->id, $id);
        $this->db->delete($this->table);
    }

}

```

3. View

View merupakan bagian yang menangani terkait tampilan *user interface* sebuah aplikasi. Didalam aplikasi *web* biasanya pasti akan berhubungan dengan *html* dan *css*. Berikut *view* yang terdapat pada sistem aplikasi, diantaranya:

a. Dashboard_view.php

Dashboard_view.php merupakan *view* dari halaman *dashboard* atau halaman yang *user* masuk untuk pertama kali setelah *login*. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur warna pada *button*, penulisan judul atau penamaan pada *header*.

Berikut *source code* untuk halaman Dashboard_view.php:

```
<section class="content-header">
    <h1>
        Dashboard
    </h1>
</section>

<section class="content">
    <div class="row">
        <div class="col-lg-3 col-xs-6">
            <div class="small-box bg-purple">
                <div class="inner">
                    <h3><?php echo $total_departemen;
?></h3>

                    <p>Departemen </p>
                </div>
                <div class="icon">
                    <i class="ion ion-clipboard"></i>
                </div>
                <a href="<?php echo $departemen; ?>"
class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
            </div>
        </div>
    </div>
```

```

<div class="col-lg-3 col-xs-6">
    <div class="small-box bg-green">
        <div class="inner">
            <h3><?php    echo    $total_pengirim;
?></h3>

            <p>Pengirim dan Tujuan</p>
        </div>
        <div class="icon">
            <i        class="ion        ion-person-
stalker"></i>
        </div>
        <a href="<?php echo $pengirim; ?>"
class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
        </div>
    </div>
    <div class="col-lg-3 col-xs-6">
        <div class="small-box bg-red">
            <div class="inner">
                <h3><?php    echo    $total_surat_masuk;
?></h3>

                <p>Surat Masuk</p>
            </div>
            <div class="icon">
                <i class="ion ion-email"></i>
            </div>
            <a href="<?php echo $surat_masuk; ?>"
class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
            </div>
        </div>

```

```

<div class="col-lg-3 col-xs-6">
    <div class="small-box bg-yellow">
        <div class="inner">
            <h3><?php echo $total_surat_keluar;
?></h3>

            <p>Surat Keluar</p>
        </div>
        <div class="icon">
            <i class="ion ion-email"></i>
        </div>
        <a href="<?php echo $surat_keluar; ?>"
class="small-box-footer">More info <i class="fa fa-
arrow-circle-right"></i></a>
        </div>
    </div>
</div>
</section>

```

b. Tbl_arsip_keluar_form.php

Tbl_arsip_keluar_form.php merupakan *view* dari halaman *input* data arsip surat keluar pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur penggunaan *button*, *textbox*, dll., penulisan judul atau penamaan pada *header*.

Berikut *source code* untuk halaman Tbl_arsip_keluar_form.php:

```

<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>

```



```

<h3 class='box-title'>Form
Surat Keluar</h3>

<div class='box box-primary'>
    <form action="<?php echo
$action; ?>" method="post"
enctype="multipart/form-data">
        <table class='table
table-bordered'>
            <tr><td
width="200">No. Surat <?php echo
form_error('no_surat') ?></td>
                <td><input
type="text" class="form-control" name="no_surat"
id="no_surat" placeholder="No Surat" value="<?php
echo $no_surat; ?>" />
                    </td>
            <tr><td>Tanggal
Surat <?php echo form_error('tanggal_surat')
?></td>
                <td><input
type="date" class="form-control"
name="tanggal_surat" id="tanggal_surat"
placeholder="Tanggal Surat" value="<?php echo
$tanggal_surat; ?>" />
                    </td>
            <tr><td>Tanggal
Dikirim <?php echo form_error('tanggal_diterima')
?></td>
                <td><input
type="date" class="form-control"
name="tanggal_diterima" id="tanggal_diterima"

```

```

placeholder="Tanggal Diterima" value="php echo
$tanggal_diterima; ?" />

</td>

<?php

if($tanggal_diterima>0)

    { ?>

        <tr><td>Tanggal

Diterima <?php echo form_error('tanggal_dikirim')
?></td>

            <td><input

type="date"    class="form-control"    name="tanggal"
id="tanggal"    placeholder="Tanggal    Dikirim"
value="php echo $tanggal; ?" />

                </td>

                    <?php

                        }else {

                            } ?>

                <tr><td>Perihal

<?php echo form_error('perihal') ?></td>

                    <td><input

type="text"    class="form-control"    name="perihal"
id="perihal"    placeholder="Perihal"    value="php
echo $perihal; ?" />

                        </td>

                            <tr><td>Departemen

<?php echo form_error('id_departemen') ?></td>

                                <td>

                                    <?php echo

cmb_dinamis('id_departemen',    'tbl_departemen',
'nama_departemen',    'id_departemen',
$id_departemen) ?>

```

```

        </td>
        <tr><td>Penerima
<?php echo form_error('id_pengirim') ?></td>
        <td>
        <?php echo
cmb_dinamis('id_pengirim', 'tbl_pengirim_surat',
'nama_pengirim', 'id_pengirim', $id_pengirim) ?>
        </td>
        <tr><td>File <?php
echo form_error('file') ?></td>
        <td><input
type="file" name="file" id="file">
        </td></tr>
        <tr
hidden=""><td>Jenis        Surat        <?php        echo
form_error('jenis_surat') ?></td>
        <td>
        <input
type="hidden"        name="jenis_surat"
value="surat_keluar" />
        </td>
        <input
type="hidden" name="id_arsip" value="<?php echo
$id_arsip; ?>" />
        <tr><td
colspan='2'><button type="submit" class="btn btn-
primary"><?php echo $button ?></button>
        <a
href="<?php echo site_url('surat_keluar') ?>"
class="btn btn-default">Cancel</a></td></tr>

        </table></form>

```

```

        </div>
    </div>
</div>
</section>

```

c. Tbl_arsip_keluar_list.php

Tbl_arsip_keluar_list.php merupakan *view* dari halaman tampil data arsip surat keluar pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur tabel, mengatur letak *button action*, pemberian nama pada judul atau *header*.

Berikut *source code* untuk halaman Tbl_arsip_keluar_list.php:

```

<style>
    @media screen {
        #div-cetak {
            display : none;
        }
    }
    @media print {
        *:not (body):not (#div-cetak) {
            visibility: hidden;
        }
        #div-cetak *{
            font-size: 1.5rem;
            visibility: visible !important;
        }
    }
</style>
<section class="content-header">
    <h1>
        Data Tabel

```

```

        </h1>
    </section>
    <section class='content'>
        <div class='row'>
            <div class='col-xs-12'>
                <div class='box'>
                    <div class='box-header'>
                        <h3 class='box-title'>Data
Surat</h3>
                        <div style="margin-top:
1rem;"><?php echo
anchor('surat_keluar/create/', 'Create', array('clas
s'=>'btn btn-danger btn-sm'));?>
                        <?php echo
anchor(site_url('surat_keluar/excel'), ' <i
class="fa fa-file-excel-o"></i> Excel',
'class="btn btn-primary btn-sm"'); ?>
                    </div>
                    <div class='box-body'>
                        <table class="table table-bordered table-
striped" id="mytable">
                            <thead>
                                <tr>
                                    <th width="10px">No</th>
                                    <th>No. Surat</th>
                                    <th>Tanggal Surat</th>
                                    <th>Tanggal Dikirim</th>
                                    <th>Tanggal Diterima</th>
                                    <th>Perihal</th>
                                    <th>Departemen</th>
                                    <th>Penerima</th>
                                    <th width="140">Action</th>
                                </tr>

```

```

        </thead>
        <tbody>
            <?php
                $start = 0;
                foreach ($surat_data as $surat)
                {
                    ?>
                    <tr>
                        <td><?php echo ++$start ?></td>
                        <td><?php echo $surat->no_surat
?></td>
                        <td><?php echo $surat->tanggal_surat
?></td>
                        <td><?php echo $surat->tanggal_diterima ?></td>
                        <td><?php echo $surat->tanggal_dikirim
?></td>
                        <td><?php echo $surat->perihal
?></td>
                        <td><?php echo $surat->nama_departemen ?></td>
                        <td><?php echo $surat->nama_pengirim
?></td>
                        <td style="text-align:center"
width="140px">
                            <a target="_blank" class="btn btn-
danger btn-sm"
href="<?=site_url('surat_keluar/cetak/'.$surat->file)?>"><i class="fa fa-print"></i></a>
                            <?php
                                echo
                                anchor(base_url('uploads/'.$surat->file),'<i
class="fa fa-cloud-

```

```

download"></i>',array('title'=>'detail','class'=>'
btn btn-danger btn-sm','target'=>'new')));
        echo '  ';
        echo
anchor(site_url('surat_keluar/update/'.$surat-
>id_arsip),'<i      class="fa      fa-pencil-square-
o"></i>',array('title'=>'edit','class'=>'btn btn-
danger btn-sm')));
        echo '  ';
        echo
anchor(site_url('surat_keluar/delete/'.$surat-
>id_arsip),'<i      class="fa      fa-trash-
o"></i>','title="delete"      class="btn btn-danger
btn-sm" onclick="javascript: return confirm(\'Are
You Sure ?\')"');
        ?>
    </td>
</tr>
<?php
    }
    ?>
</tbody>
</table>
<script      src="<?php      echo
base_url('assets/js/jquery-1.11.2.min.js')
?>"></script>
    <script      src="<?php      echo
base_url('assets/datatables/jquery.dataTables.js')
?>"></script>
    <script      src="<?php      echo
base_url('assets/datatables/dataTables.bootstrap.j
s') ?>"></script>
    <script type="text/javascript">

```

```

$(document).ready(function () {
    $("#mytable").dataTable();
});
</script>
</div>
</div>
</div>
</div>
</section>

<div id="div-cetak" style="width: 100%;background-
color: white;position: absolute;top: 0;left: 0;">
    
</div>

<script>
    function cetakGambar(url) {
        console.log(url)
        window.print();
    }
</script>

```

d. Tbl_arsip_masuk_form.php

Tbl_arsip_masuk_form.php merupakan *view* dari halaman *input* data arsip surat masuk pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur penggunaan *button*, *textbox*, dll., penulisan judul atau penamaan pada *header*.

Berikut *source code* untuk halaman Tbl_arsip_masuk_form.php:


```

<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>

                    <h3      class='box-title'>Form
Surat Masuk</h3>

                    <div class='box box-primary'>
                        <form action="<?php echo
$action;          ?>"          method="post"
enctype="multipart/form-data">
                            <table      class='table
table-bordered'>
                                <tr><td
width="200">No.          Surat          <?php          echo
form_error('no_surat') ?></td>
                                    <td><input
type="text"  class="form-control"  name="no_surat"
id="no_surat"  placeholder="No Surat"  value="<?php
echo $no_surat; ?>" />
                                    </td>
                                <tr><td>Tanggal
Surat  <?php  echo  form_error('tanggal_surat')
?></td>
                                    <td><input
type="date"          class="form-control"
name="tanggal_surat"          id="tanggal_surat"
placeholder="Tanggal  Surat"  value="<?php  echo
$tanggal_surat; ?>" />
                                    </td>

```

```

                                <tr><td>Tanggal
Diterima <?php echo form_error('tanggal_diterima')
?></td>

                                <td><input
type="date"                                class="form-control"
name="tanggal_diterima"                    id="tanggal_diterima"
placeholder="Tanggal Diterima" value="<?php echo
$tanggal_diterima; ?>" />

                                </td>
                                <tr><td>Perihal
<?php echo form_error('perihal') ?></td>

                                <td><input
type="text"    class="form-control"    name="perihal"
id="perihal"   placeholder="Perihal"   value="<?php
echo $perihal; ?>" />

                                </td>
                                <tr><td>Departemen
<?php echo form_error('id_departemen') ?></td>

                                <td>

                                <?php echo
cmb_dinamis('id_departemen',      'tbl_departemen',
'nama_departemen',                'id_departemen',
$id_departemen) ?>

                                </td>
                                <tr><td>Pengirim
<?php echo form_error('id_pengirim') ?></td>

                                <td>

                                <?php echo
cmb_dinamis('id_pengirim',      'tbl_pengirim_surat',
'nama_pengirim', 'id_pengirim', $id_pengirim) ?>

                                </td>

```

```

                                <tr><td>File <?php
echo form_error('file') ?></td>

                                <td><input
type="file" name="file" id="file">

                                </td></tr>

</td>

                                <input
type="hidden" name="id_arsip" value="<?php echo
$id_arsip; ?>" />

                                <tr><td
colspan='2'><button type="submit" class="btn btn-
primary"><?php echo $button ?></button>

                                <a
href="<?php echo site_url('surat_masuk') ?>"
class="btn btn-default">Cancel</a></td></tr>

                                </table></form>

                                </div>

                                </div>

                                </div>

                                </div>

</section>

```

e. Tbl_arsip_masuk_list.php

Tbl_arsip_masuk_list.php merupakan *view* dari halaman tampil data arsip surat masuk pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur tabel, mengatur letak *button action*, pemberian nama pada judul atau *header*.

Berikut *source code* untuk halaman Tbl_arsip_masuk_list.php:

```

<style>
    @media screen {
        #div-cetak {
            display : none;
        }
    }
    @media print {
        *:not(body):not(#div-cetak) {
            visibility: hidden;
        }
        font-size: 1.5rem;
        visibility: visible !important;
    }
}
</style>
<section class="content-header">
    <h1>
        Data Tabel
    </h1>
</section>
<!-- Main content -->
<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>
                    <h3 class='box-title'>Data
Surat</h3>
                <div
                    style="margin-top:
1rem;"><?php
                                echo
anchor('surat_masuk/create/', 'Create', array('class
'=>'btn btn-danger btn-sm')));?>

```

```

        <?php                                                    echo
        anchor(site_url('surat_masuk/excel'), ' <i
        class="fa fa-file-excel-o"></i> Excel',
        'class="btn btn-primary btn-sm"'); ?>

        </div>

        <button class="btn btn-default btn-xs pull-
        right" onclick="print()"><i class="glyphicon
        glyphicon-print"></i> Print</button>

        </div>

        <div class='box-body'>

        <table class="table table-bordered table-
        striped" id="mytable">

            <thead>

                <tr>

                    <th width="10px">No</th>

                    <th>No. Surat</th>

                    <th>Tanggal Surat</th>

                    <th>Tanggal Diterima</th>

                    <th>Perihal</th>

                    <th>Departemen</th>

                    <th>Pengirim</th>

                    <th width="120">Action</th>

                </tr>

            </thead>

            <tbody>

                <?php
                $start = 0;
                foreach ($surat_data as $surat)
                {

                    ?>

                    <tr>

                        <td><?php echo ++$start ?></td>

```

```

        <td><?php      echo      $surat->no_surat
?></td>

        <td><?php echo $surat->tanggal_surat
?></td>

        <td><?php      echo      $surat-
>tanggal_diterima ?></td>
        <td><?php      echo      $surat->perihal
?></td>

        <td><?php      echo      $surat-
>nama_departemen ?></td>
        <td><?php echo $surat->nama_pengirim
?></td>

        <td      style="text-align:center"
width="140px">
                <a target="_blank" class="btn btn-
danger                                btn-sm"
href="<?=site_url('surat_masuk/cetak/' . $surat-
>file) ?>"><i class="fa fa-print"></i></a>

                <?php
                        echo
anchor(base_url('uploads/' . $surat->file), '<i
class="fa                                fa-cloud-
download"></i>', array('title'=>'detail', 'class'=>'
btn btn-danger btn-sm', 'target'=>'new'));
                        echo '    ';
                        echo
anchor(site_url('surat_masuk/update/' . $surat-
>id_arsip), '<i      class="fa      fa-pencil-square-
o"></i>', array('title'=>'edit', 'class'=>'btn btn-
danger btn-sm'));
                        echo '    ';
                        echo
anchor(site_url('surat_masuk/delete/' . $surat-

```

```

>id_arsip), '<i          class="fa          fa-trash-
o"></i>', 'title="delete"    class="btn    btn-danger
btn-sm" onclick="javasciprt: return confirm(\'Are
You Sure ?\')"'');

        ?>
    </td>
</tr>

    <?php
    }
    ?>
</tbody>
</table>

<script          src="<?php          echo
base_url('assets/js/jquery-1.11.2.min.js')
?>"></script>

    <script          src="<?php          echo
base_url('assets/datatables/jquery.dataTables.js')
?>"></script>

    <script          src="<?php          echo
base_url('assets/datatables/dataTables.bootstrap.j
s') ?>"></script>

    <script type="text/javascript">
        $(document).ready(function () {
            $("#mytable").dataTable();
        });
    </script>

        </div>

    </div>

</div>

</div>
</section>

```

```

<div id="div-cetak" style="width: 100%;background-
color: white;position: absolute;top: 0;left: 0;">
    
</div>

<script>
    function cetakGambar(url) {
        console.log(url)
        // $('#div-cetak img').attr('src',url);
        window.print();
    }
</script>

```

f. Tbl_departemen_form.php

Tbl_departemen_form.php merupakan *view* dari halaman *input* data departemen atau perusahaan pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur penggunaan *button*, *textbox*, dll., penulisan judul atau penamaan pada *header*.

Berikut *source code* untuk halaman Tbl_departemen_form.php:

```

<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>

                    <h3 class='box-title'>Form Data
Departemen</h3>

```



```


#### g. Tbl_departemen_list.php



Tbl_departemen_list.php merupakan view dari halaman tampil data departemen atau data perusahaan pada sistem aplikasi. Pada code ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur tabel, mengatur letak button action, pemberian nama pada judul atau header.



195


```

Berikut *source code* untuk halaman Tbl_departemen_list.php:

```
<section class="content-header">
    <h1>
        Data Tabel
    </h1>
</section>
<!-- Main content -->
<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>
                    <h3 class='box-title'>Data
Departemen </h3>
                <div style="margin-top:
1rem;"><?php echo
anchor('departemen/create/', 'Create', array('class'
=>'btn btn-danger btn-sm'));?>
                <?php echo
anchor(site_url('departemen/excel'), ' <i
class="fa fa-file-excel-o"></i> Excel',
'class="btn btn-primary btn-sm"'); ?>
            </div>
            <div class='box-body'>
                <table class="table table-bordered table-
striped" id="mytable">
                    <thead>
                        <tr>
                            <th width="20px">No</th>
                            <th>Nama Departemen</th>
                            <th>Action</th>
                        </tr>
```

```

        </thead>
        <tbody>
            <?php
                $start = 0;
                foreach ($departemen_data as
$departemen)
                {
                    ?>
                    <tr>
                        <td><?php echo ++$start ?></td>
                        <td><?php echo $departemen-
>nama_departemen ?></td>
                        <td style="text-align:center"
width="90px">
                            <?php
                                echo
                                anchor(site_url('departemen/update/'.$departemen-
>id_departemen), '<i class="fa fa-pencil-square-
o"></i>', array('title'=>'edit', 'class'=>'btn btn-
danger btn-sm'));
                                echo ' ';
                                echo
                                anchor(site_url('departemen/delete/'.$departemen-
>id_departemen), '<i class="fa fa-trash-
o"></i>', 'title="delete" class="btn btn-danger
btn-sm" onclick="javasciprt: return confirm(\'Are
You Sure ?\')"');
                                ?>
                            </td>
                        </tr>
                    <?php
                        }
                    ?>

```

```

        </tbody>
    </table>
    <script src="<?php echo
base_url('assets/js/jquery-1.11.2.min.js')
?>"></script>
    <script src="<?php echo
base_url('assets/datatables/jquery.dataTables.js')
?>"></script>
    <script src="<?php echo
base_url('assets/datatables/dataTables.bootstrap.j
s') ?>"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $("#mytable").dataTable();
        });
    </script>
</div>
</div>
</div>
</div>
</section>

```

h. Tbl_pengirim_surat_form.php

Tbl_pengirim_surat_form.php merupakan *view* dari halaman *input* data pengirim surat dan penerima surat pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur penggunaan *button*, *textbox*, dll., penulisan judul atau penamaan pada *header*.

Berikut *source code* untuk halaman Tbl_pengirim_surat_form.php:

```

<section class='content'>
    <div class='row'>

```

```

<div class='col-xs-12'>
    <div class='box'>
        <div class='box-header'>

            <h3 class='box-title'>Form
Data Pengirim atau Tujuan</h3>
            <div class='box box-primary'>
                <form action="<?php echo
$action; ?>" method="post"><table class='table
table-bordered'>

                    <tr><td
WIDTH="140">Nama Pengirim atau Tujuan <?php echo
form_error('nama_pengirim') ?></td>

                        <td><input
type="text" class="form-control"
name="nama_pengirim" id="nama_pengirim"
placeholder="Nama Pengirim" value="<?php echo
$nama_pengirim; ?>" />

                            </td>
                                <input
type="hidden" name="id_pengirim" value="<?php echo
$id_pengirim; ?>" />

                                    <tr><td
colspan='2'><button type="submit" class="btn btn-
primary"><?php echo $button ?></button>

                                        <a
href="<?php echo site_url('pengirim') ?>"
class="btn btn-default">Cancel</a></td></tr>

                                            </table></form>

                                                </div>
                                                    </div>
                                                        </div>

```

```
</div>
</section>
```

i. Tbl_pengirim_surat_list.php

Tbl_pengirim_surat_list.php merupakan *view* dari halaman tampil data pengirim dan penerima surat pada sistem aplikasi. Pada *code* ini terdapat beberapa perintah yang digunakan untuk mengatur tampilan pada sistem aplikasi, seperti mengatur tabel, mengatur letak *button action*, pemberian nama pada judul atau *header*.

Berikut *source code* untuk halaman Tbl_pengirim_surat_list.php:

```
<section class="content-header">
    <h1>
        Data Tabel
    </h1>
</section>
<section class='content'>
    <div class='row'>
        <div class='col-xs-12'>
            <div class='box'>
                <div class='box-header'>
                    <h3 class='box-title'>Data
Pengirim atau Tujuan </h3>
                    <div style="margin-top:
1rem;"><?php echo
anchor('pengirim/create/', 'Create', array('class'=>
'btn btn-danger btn-sm'));>?>
                        <?php echo
anchor(site_url('pengirim/excel'), ' <i class="fa
fa-file-excel-o"></i> Excel', 'class="btn btn-
primary btn-sm"'); ?>
                    </div>
```

```


201


```

```

o"></i>', 'title="delete"    class="btn    btn-danger
btn-sm" onclick="javasciprt: return confirm(\'Are
You Sure ?\')"');
        ?>
    </td>
</tr>
<?php
    }
    ?>
</tbody>
</table>

    <script            src="<?php            echo
base_url('assets/js/jquery-1.11.2.min.js')
?>"></script>
    <script            src="<?php            echo
base_url('assets/datatables/jquery.dataTables.js')
?>"></script>
    <script            src="<?php            echo
base_url('assets/datatables/dataTables.bootstrap.j
s') ?>"></script>
    <script type="text/javascript">
        $(document).ready(function () {
            $("#mytable").dataTable();
        });
    </script>
        </div><!-- /.box-body -->
    </div>
</div>
</div>
</section>

```