

BAB III

LANDASAN TEORI

3.1 Pengertian Surat - Menyurat

Dalam kehidupan seperti sekarang ini selain komunikasi yang dilakukan secara lisan, komunikasi tulis mutlak pula diperlukan oleh masyarakat modern untuk menunjang aktivitas kehidupan sehari – hari, dan dalam komunikasi tulis salah satu media yang paling banyak dipakai adalah surat.

Secara umum surat dapat dikatakan sebagai alat untuk menyampaikan maksud tertentu dari penulisannya secara tertulis (Lamuddin Finosa, 1991). Dan dalam arti sehari – hari surat umumnya hanya dikenal sebagai alat untuk menyampaikan berita secara tertulis. Sedangkan pengertian surat yang lebih luas memiliki fungsi sebagai penyampai informasi tertulis rekaman berita berupa catatan tentang suatu aktivitas baik itu aktivitas pribadi ataupun suatu organisasi. Selain itu surat juga berfungsi sebagai tanda bukti tertulis, alat pengingat, pedoman untuk bertindak, keterangan pengamatan, duta atau wakil organisasi, dokumentasi historis dari suatu kegiatan.

Berdasarkan uraian di atas maka surat dapat didefinisikan sebagai suatu informasi tertulis yang dapat dipergunakan sebagai alat komunikasi tulis yang dibuat dengan persyaratan tertentu yang berlaku khusus pada bidang surat menyurat.

Sedangkan penyampaian maksud melalui surat dari satu pihak ke pihak lain dapat atas nama perseorangan atau pribadi dan dapat pula atas nama jabatan seseorang dalam suatu organisasi. Kegiatan saling berkirim surat baik oleh

perseorangan maupun organisasi disebut surat menyurat atau korespondensi. Dan pelakunya disebut koresponden dimana yang biasa melakukan adalah para pegawai, namun pimpinan dan staf lain juga perlu menguasainya karena mereka juga akan terlibat dalam proses kegiatan surat menyurat tersebut.

3.2 Bagian – Bagian Surat

Pada umumnya surat terdiri atas empat komponen utama yaitu kop surat (kepala surat), leher surat, badan surat, dan kaki surat. Setiap komponen memiliki fungsi sendiri dalam memperkenalkan dan mengomunikasikan pesan si pengirim kepada si penerima surat. Untuk lebih jelasnya bagian – bagian surat dapat diuraikan sebagai berikut :

1. Kepala surat

Pada umumnya, organisasi, badan hukum, dan perusahaan memiliki kertas kop surat. Berikut ini adalah fungsi dari kepala atau kop surat.

a. Sebagai identitas

Pada kop surat tercantum logo atau lambing perusahaan atau instansi. Logo atau lambing ini dipilih dan dirancang dengan cermat agar memberi kesan yang baik, mudah diingat, khas, berwibawa, dan mewakili perusahaan yang dilambangkannya.

b. Sebagai pemberi informasi

Kop surat yang lengkap mencantumkan nama perusahaan atau instansi, logo, bidang usaha atau jenis aktivitas yang dijalani, alamat lengkap, nomor telepon, kotak pos, facsimile dan kantor cabang sehingga dapat

memberikan informasi yang jelas kepada pembaca mengenai perusahaan atau instansi pengirimnya.

2. Tanggal surat

Tanggal surat sebaiknya ditulis dengan lengkap, yaitu dengan mencantumkan tanggal, bulan, dan tahun. Fungsi tanggal surat adalah sebagai berikut :

- a. Mempermudah penetapan waktu membalasnya.
- b. Mempermudah pengingatan kembali dan pengagendanya bagi si penerima
- c. Sebagai referensi dan petunjuk bagi petugas administrasi dan kearsipan.

3. Nomor surat

Nomor surat berfungsi sebagai berikut :

- a. Referensi atau petunjuk bagi petugas kearsipan.
- b. Petunjuk unit atau departemen asal surat.
- c. Mengetahui jumlah surat keluar pada suatu periode tertentu.
- d. Memudahkan pengaturan dan pencarian jika surat diperlukan kembali.

4. Perihal surat

Perihal atau hal merupakan petunjuk mengenai intisari atau pokok isi surat. Fungsinya sebagai penyimpul pokok isi atau intisari surat, sehingga memudahkan penerima mengetahui pokok permasalahan dan memberikan jawaban.

5. Lampiran

Lampiran ini berfungsi sebagai petunjuk mengenai dokumen yang menyertai surat. Lampiran ditempatkan dibawah dan sejajar dengna nama

jabatan, sedangkan dalam surat – surat pemerintah penulisannya setelah penulisan nomor surat.

6. Alamat yang dituju.

Alamat yang dituju berfungsi sebagai pemberi informasi pada si pengantar agar tidak mengalami kesulitan dalam pengiriman surat.

7. Kalimat pembuka

Kalimat pembuka berfungsi sebagai pengantar isi surat yang sesungguhnya agar si penerima mengetahui alasan dikirimnya surat tersebut.

8. Isi surat

Isi surat merupakan uraian mengenai maksud pembuatan surat dan hal – hal yang ingin disampaikan. Oleh sebab itu, surat harus diutarakan dengan jelas dan tidak bertele – tele dan harus sama dengan yang dinyatakan dalam perihal atau hal.

9. Kalimat penutup

Kalimat penutup berfungsi sebagai ucapan terima kasih, penegasan, penghargaan atau pengarahan.

10. Salam penutup

Salam penutup berfungsi sebagai tanda bahwa pembicaraan selesai dan surat siap ditandatangani.

3.3 Analisa dan Perancangan Sistem

Analisa sistem merupakan tahap yang paling penting dari suatu pengembang sistem karena merupakan tahap awal untuk melakukan evaluasi permasalahan yang terjadi serta kendala – kendala yang dihadapi dari sebuah sistem yang telah berjalan.

Analisa sistem itu sendiri dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian – bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan – permasalahan, kesempatan – kesempatan, hambatan – hambatan yang terjadi dan kebutuhan – kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan – perbaikannya (Jogianto, 1989:129).

Analisa yang efektif akan memudahkan pekerjaan penyusunan rencana yang baik ditahap berikutnya. Sebaliknya, kesalahan yang terjadi pada tahap analisis ini akan menyebabkan kesulitan yang lebih besar, bahkan menyebabkan gagalnya penyusunan sebuah sistem (Jogianto, 2005).

Untuk itu, diperlukan ketelitian dalam mengerjakan, sehingga tidak dapat kesalahan dalam tahap selanjutnya, yaitu tahap perancangan sistem. Langkah – langkah yang diperlukan dalam menganalisa sistem adalah :

1. Tahap perencanaan sistem
2. Tahap analisis sistem
3. Tahap perancangan sistem
4. Tahap penerapan sistem
5. Membuat laporan dari hasil analisa

Pada tahap perancangan, dilakukan identifikasi masalah serta diperlukan adanya analisa yang digunakan untuk menentukan factor – factor yang menjadi permasalahan dalam sistem yang telah ada atau digunakan.

Data –data yang baik yang berasal dari sumber – sumber internal seperti misalnya laporan – laporan, dokumen observasi, maupun sumber – sumber diluar lingkungan sistem seperti pemakai sistem, dikumpulkan sebagai bahan pertimbangan analisa. Jika semua permasalahan sudah diidentifikasi, dilanjutkan dengan mempelajari dan memahami alur kerja dari sistem yang digunakan.

Kemudian diteruskan dengan menganalisa dan membandingkan sistem yang terbentuk dengan sistem yang sebelumnya digunakan. Dengan adanya perubahan tersebut, maka langkah selanjutnya adalah membuat laporan – laporan hasil analisa sebelumnya dan sistem yang akan diterapkan. Perancangan sistem adalah proses menyusun atau mengembangkan sistem informasi yang baru. Dalam tahap ini, harus dipastikan bahwa semua persyaratan untuk menghasilkan informasi dapat terpenuhi.

Hasil sistem yang dirancang harus sesuai dengan kebutuhan pemakai, karena rancangan tersebut meliputi perancangan mulai dari sistem yang umum hingga diperoleh sistem yang lebih spesifik. Dari hasil rancangan tersebut, dibentuk pula rancangan *database* disertai dengan struktur *file* antara sistem yang satu dengan sistem yang lainnya. Selain itu, dibentuk pula rancangan *input* dan *output system*, misalnya menentukan berbagai bentuk *input* data dan isi laporan.

Apabila di dalam perancangan sistem terdapat kesalahan maka kita perlu melihat kembali dari analisa sistem yang telah dibuat. Sehingga dapat diambil

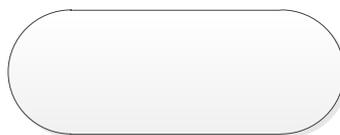
kesimpulan bahwa analisa sistem mempunyai hubungan erat dengan perancangan sebuah sistem.

3.4 System Flow

System Flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem dimana bagan ini menjelaskan urutan prosedur – prosedur yang ada dalam sistem dan biasanya dalam membuat *system flow* sebaiknya ditentukan pula fungsi – fungsi yang melaksanakan atau bertanggung jawab terhadap sub – sistem yang ada (Jogianto, 1998:10).

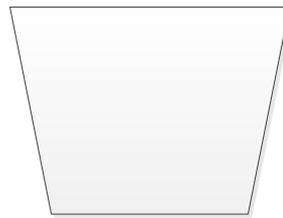
Terdapat berbagai bentuk symbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah *terminator*, *manual operation*, *document*, *process*, *database*, *manual input*, *off-line storage*, *on-page reference*, dan *off-page reference*.

Terminator merupakan bentuk symbol yang digunakan sebagai tanda dimulainya jalan proses ataupun tanda akhir dari sebuah pengerjaan suatu sistem. Simbol dari *terminator* dapat dilihat pada gambar 3.1.



Gambar 3.1 *Terminator*

Manual operation digunakan untuk menggambarkan sebuah proses kerja yang digunakan tanpa menggunakan computer sebagai medianya (menggunakan proses manual). Simbol dari *manual operation* dapat dilihat pada gambar 3.2.



Gambar 3.2 *Manual Operation*

Document merupakan symbol dari dokument yang berupa kertas laporan, surat – surat, memo, maupun arsip – arsip secara fisik. Simbol dari *document* dapat dilihat pada gambar 3.3.



Gambar 3.3 *Document*

Process adalah sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi. Simbol dari *process* dapat dilihat pada gambar 3.4.



Gambar 3.4 *Process*

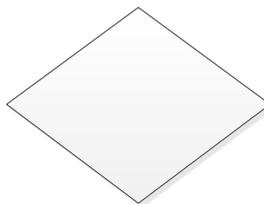
Database digunakan sebagai media penyimpanan data yang bersifat terkomputerisasi. Simbol dari *database* dapat dilihat pada gambar 3.5.



Gambar 3.5 *Database*

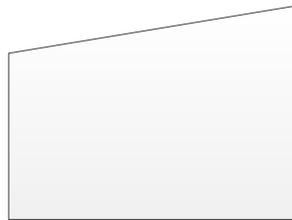
Decision merupakan operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar atau salah.

Simbol dari *decision* dapat dilihat pada gambar 3.6.



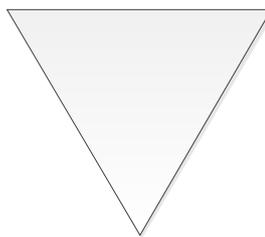
Gambar 3.6 *Decision*

Manual input digunakan untuk melakukan proses input kedalam *database* melalui *keyboard*. Simbol dari *manual input* dapat dilihat pada gambar 3.7.



Gambar 3.7 *Manual Input*

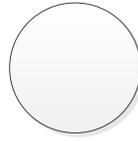
Off-line storage merupakan bentuk media penyimpanan yang berbeda dengan *database*, dimana media penyimpanan ini menyimpan document secara manual atau lebih dikenal dengan nama arsip. Simbol dari *off-line storage* dapat dilihat pada gambar 3.8.



Gambar 3.8 *Off-Line Storage*

On-page reference digunakan sebagai symbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh

dalam permasalahan letaknya. Simbol dari *on-page reference* dapat dilihat pada gambar 3.9.



Gambar 3.9 *On-Page Reference*

Off-page reference memiliki sifat yang sedikit berbeda dengan *on-page reference*, karena symbol ini hanya digunakan apabila arus data ada dilanjutkan ke halaman yang berbeda. Simbol dari *off-page reference* dapat dilihat pada gambar 3.10.



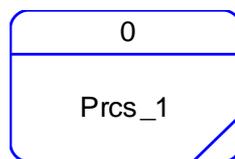
Gambar 3.10 *Off-Page Reference*

3.5 Data Flow Diagram (DFD)

Menurut (Kritanto, 2004:12), *Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data tersebut disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

Data Flow Diagram merupakan suatu metode pengembangan sistem yang terstruktur (*structure analysis dan design*). Penggunaan notasi dalam data flow diagram sangat membantu untuk memahami suatu sistem pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Didalam *data flow diagram* terdapat empat simbol yang digunakan yaitu *process*, *external entity*, *data store*, dan *data flow*. Simbol *process* digunakan untuk melakukan suatu perubahan berdasarkan data yang diinputkan dan menghasilkan data dari perubahan tersebut. Simbol *process* dapat dilihat pada gambar 3.11.



Gambar 3.11 Simbol *Process*

Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas process. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa

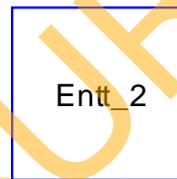
proses tersebut adalah sebuah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *context diagram* dapat dilakukan dengan terlebih dahulu menentukan nama sistemnya, menentukan batasan dari sistem, dan menentukan *terminator* yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

Nomor 1, 2, 3, dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses level-0 (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses level-0 membahas sistem secara lebih mendetail, baik dipandang dari segi kegiatan dari sebuah bagian, alur data yang ada, maupun *database* yang digunakan didalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing – masing proses dari pada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar dari suatu level harus sama dengan alur data yang masuk dan keluar pada level berikutnya), memunculkan data store sebagai sumber maupun tujuan data (optional), menggambarkan diagram level-0, menghindari perpotongan arus data, dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses).

Nomor 1.1, 1.2, 2.2, dan seterusnya merupakan sebuah proses turunan atau *decompose* dari proses level -0 yang disebut sebagai proses level-1 (satu). Proses level-1 menggambarkan detil kerja dari sebuah bagian dalam sebuah sistem. Penggambarannya dilakukan dengan cara menentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level-0, menentukan apa yang diterima

atau diberikan masing – masing sub-proses daripada sistem dan tetap memperhatikan konsep keseimbangan, memunculkan data store sebagai sumber maupun tujuan alur data (optional), menggambarkan DFD level-1, dan berusaha untuk menghindari perpotongan arus data. Hasil turunan akhir disebut sebagai *the lowest level*, dimana hasil akhir ini tergantung dari kompleksitas sistem yang ada.

- a. *External entity* disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku – pelaku sistem yang terkait, data berupa orang – orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada *process* dan mendapatkan keluaran dari *process*. *External entity* dapat dilihat pada gambar 3.12.



Gambar 3.12 Simbol *External Entity*

- b. *Data store* digunakan sebagai media penyimpanan suatu data yang dapat berupa *file* atau *database*, arsip atau catatan manual, lemari *file*, dan table – table dalam *database*. Penamaan *data store* harus sesuai dengan bentuk data yang tersimpan pada *data store* tersebut, misalnya table pelamar, table pendidikan, table lulus seleksi, dan lain – lain. *Data store* dapat dilihat pada gambar 3.13.

Gambar 3.13 Simbol *Data Store*

- c. *data flow* merupakan penghubung antar *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran data dari satu titik ke titik lainnya dengan tanda anak panah mengarah ke tujuan data. Penamaan *data flow* harus menggunakan kata benda, karena didalam data flow mengandung sekumpulan data. *Data flow* dapat dilihat pada gambar 3.14.

Gambar 3.14 Simbol *Data Flow*

3.6 Entity Relationship Diagram (ERD)

Proses *reverse engineering* terhadap suatu basis data menjadi suatu kebutuhan bagi perancang basis data untuk mengetahui struktur dari sebuah basis data. Struktur tersebut biasanya dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD).

ERD dibagi menjadi dua macam yaitu : *Conceptual Data Model* (CDM), dan *Physical Data Model* (PDM). Symbol – simbol yang sering digunakan adalah sebagai berikut :

1. *Entity*

Entity merupakan sesuatu yang mudah didefinisikan. Sebuah *entity* bisa berupa obyek, tempat, orang, konsep, atau aktivitas. *Entity* dinyatakan dalam simbol persegi panjang. Simbol *entity* pada gambar 3.15.



Gambar 3.15 *Entity*

2. *Atribut*

Atribut merupakan penjelasan – penjelasan dari *entity* yang membedakan *entity* satu dengan yang lain. Sebuah *atribut* juga merupakan sifat – sifat dari sebuah *entity*. *Atribut* dinyatakan dalam simbol ellips. Simbol *atribut* pada gambar 3.16.



Gambar 3.16 *Atribut*

3. *Relationship*

Relationship adalah penghubung antara suatu *entity* dengan *entity* yang lain dan merupakan bagian yang sangat penting didalam mendesain *database*.

Ada tiga tipe *relationship* yang dikenal yaitu :

a. *One-to-One Relationship*

Jenis hubungan antar table yang menggunakan secara bersama sebuah kolom *primary key*. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk alasan keamanan atau kecepatan akses data. Seperti departemen hanya mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

b. *One-to-Many Relationship*

Jenis hubungan antar table dimana satu *record* pada satu table terhubung dengan beberapa *record* pada table lain. Jenis hubungan ini yang paling sering digunakan. Misalnya satu pekerjaan hanya dikerjakan oleh satu departemen saja, namun satu departemen dapat mengerjakan beberapamacam pekerjaan sekaligus.

c. *Many-to-Many Relationship*

Jenis hubungan antar table dimana beberapa record pada satu table terhubung dengan beberapa record pada table lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.

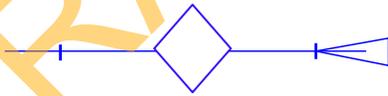
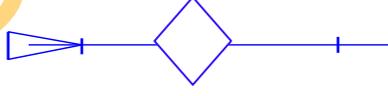
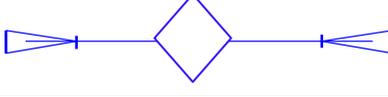
d. *Many-to-One Relationship*

Jenis hubungan antar table dimana beberapa record pada satu table terhubung dengan satu record pada table lain. Misalnya satu departemen

mampu mengerjakan banyak pekerjaan, namun satu pekerjaan hanya dikerjakan oleh satu departemen saja.

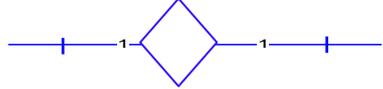
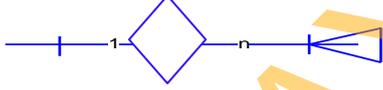
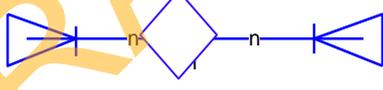
Menurut Sutanta (2004), relasi antar entitas dapat digambarkan melalui salah satu gambar 3.17 dan gambar 3.18 dari pilihan berikut ini :

1. Pilihan 1

Jenis relasi	Simbol yang digunakan
1-ke-1 :	
1-ke-n :	
n-ke-1 :	
n-ke-n :	

Gambar 3.17 Simbol Relasi Antar Entitas (Pilihan 1).

2. Pilihan 2

Jenis relasi	Simbol yang digunakan
1-ke-1 :	
1-ke-n :	
n-ke-1 :	
n-ke-n :	

Gambar 3.18 Simbol Relasi Antar Entitas (Pilihan 2).

ERD dapat digambarkan menggunakan salah satu dari pilihan diatas, namun penggunaannya harus konsisten. Jika menggunakan symbol pilihan 1, maka untuk seluruh bagian ERD harus menggunakan symbol kelompok pilihan 1.

3. Kunci relasi

Kunci relasi atau *key* adalah suatu property yang menentukan apakah suatu kolom pada table sangat penting atau tidak. Berdasarkan macamnya, kunci relasi terdiri dari :

a. Kunci kandidat

Yaitu satu atau gabungan minimal atribut yang bersifat unik yang dapat digunakan untuk mengidentifikasi setiap *record* dalam relasi.

b. Kunci primer

Yaitu bagian atau salah satu dari kunci kandidat yang digunakan sebagai kunci utama untuk membedakan setiap *record* dalam relasi. Kunci primer biasa disebut sebagai *primary key*.

c. Kunci alternative

Yaitu bagian dari kunci kandidat yang tidak digunakan sebagai kunci utama.

d. Kunci penghubung

Kunci penghubung atau *foreign key* yaitu satu atau gabungan sembarang atribut yang menjadi kunci utama dalam relasi lain yang mempunyai hubungan secara logik. Kunci penghubung dan kunci utama harus memiliki tipe dan ukuran data yang sama.

3.7 Kebutuhan Aplikasi dan Bahasa Pemrograman.

Adapun beberapa aplikasi yang dibutuhkan dalam pembuatan aplikasi surat menyurat pada Pengadilan Tinggi Tata Usaha Negara Surabaya beserta bahasa pemrograman yang digunakan dalam membangun aplikasi surat menyurat.

3.7.1 Web Text Editor

Pertama dalam pembuatan web, memerlukan sebuah aplikasi *HTML*, *code autoring* yang dapat mempermudah kita dalam membuat maupun mendesain web. Dalam pembuatan web kali ini software yang digunakan adalah Eclipse.

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*).

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plug-in*.

3.7.2 Web Server

Dalam pembuatan web secara offline dibutuhkan sebuah *web server* local (*localhost*). Hal ini berguna untuk mensimulasikan sebuah server sebenarnya (*online*). Pada *web server* akan diletakkan file beserta *databasenya*.

Web server yang digunakan kali ini adalah Apache Tomcat 6.0 yang merupakan komponen utama untuk server local.

3.7.3 Web Browser

Web Browser merupakan software untuk membuka halaman web atau melihat web yang dibuat. Web yang telah dibuat, sebelum di *publish* secara online atau dalam tahap pembuatan web, maka di cek terlebih dahulu dengan *view* melalui *web browser* akan terlihat hasil atau tampilan web yang dibuat.

Web browser yang dipakai kali ini adalah Mozilla Firefox versi terbaru untuk saat ini adalah versi 17.0.

3.7.4 Bahasa Pemrograman JSP

JSP adalah suatu teknologi web berbasis bahasa pemrograman Java dan berjalan di Platform Java, serta merupakan bagian teknologi J2EE (*Java 2 Enterprise Edition*). JSP sangat sesuai dan tangguh untuk menangani presentasi di web. Sedangkan J2EE merupakan platform Java untuk pengembangan sistem aplikasi enterprise dengan dukungan API (*Application Programming Interface*) yang lengkap dan portabilitas serta memberikan sarana untuk membuat suatu aplikasi yang memisahkan antara *business logic* (sistem), presentasi dan data.

JSP merupakan bagian dari J2EE dan khususnya merupakan komponen web dari aplikasi J2EE secara keseluruhan. JSP juga memerlukan JVM (*Java Virtual Machine*) supaya dapat berjalan, yang berarti juga mengisyaratkan keharusan menginstal *Java Virtual Machine* di server, dimana JSP akan dijalankan. Selain JVM, JSP juga memerlukan server yang disebut dengan *Web Container*.

Teknologi JSP menyediakan cara yang lebih mudah dan cepat untuk membuat halaman-halaman web yang menampilkan isi secara dinamik. Teknologi JSP didesain untuk membuat lebih mudah dan cepat dalam membuat aplikasi berbasis web yang bekerja dengan berbagai macam *web server, application server, browser* dan *development tool*.

Java Server Pages (JSP) adalah bahasa *scripting* untuk web *programming* yang bersifat *server side* seperti halnya PHP dan ASP. JSP dapat berupa gabungan antara baris HTML dan fungsi-fungsi dari JSP itu sendiri. Berbeda dengan *Servlet* yang harus dikompilasi oleh *USER* menjadi class sebelum dijalankan, JSP tidak perlu dikompilasi oleh *USER* tapi *SERVER* yang akan

melakukan tugas tersebut. Makanya pada saat user membuat pertama kali atau melakukan modifikasi halaman dan mengeksekusinya pada web browser akan memakan sedikit waktu sebelum ditampilkan.

3.7.5 Database MySQL

MySQL adalah sistem untuk mengelola database atau manajemen data. Untuk menyimpan data dan informasi ke komputer kita menggunakan data, contoh kita menyimpan data karyawan pada suatu perusahaan dan memasukkan suatu file. File data yang dikelompokkan inilah yang disebut *database*, dan MySQL bertugas mengatur dan mengelola data – data pada *database*. Dalam mengelola *database* MySQL menggunakan struktur atau kerangka yang berbentuk table. Dalam table – table itulah data di atur dan dikelompokkan.

Keunggulan MySQL adalah termasuk *open source*, dikenal sebagai sistem database yang efisien dan reliable, proses *query* cepat dan mudah. MySQL juga mudah dihubungkan dengan berbagai bahasa pemrograman dan aplikasi, sehingga tingkat komabilitasnya tinggi disamping itu kode – kode perintah MySQL sederhana sehingga dapat mudah dipahami.