BAB 2
TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori dari penulisan laporan tugas akhir ini menguraikan tentang konsep-konsep atau dasar teori dari berbagai sumber buku maupun jurnal yang akan dijadikan sebagai acuan atau landasan dalam mengembangkan sistem pengelolaan paper seminar penelitian menggunakan konsep *content management system* berbasis *web*.

2.1.1 Konsep Dasar Sistem

Pengertian sistem adalah "suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu" [3].

Dalam hal analisis dan perencanaan suatu sistem, seorang analis sistem dan perancang sistem harus mengerti terlebih dahulu mengenai komponen-komponen dari sistem yang akan dibuat. Dalam hal ini harus dibuat terlebih dahulu mengenai perencanaan sistem.

2.1.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu [3] :

1. Komponen-komponen (*Components*)

   Beberapa komponen yang saling berinteraksi dan bekerja sama untuk membentuk satu kesatuan yang dapat menjadi suatu sistem. Komponen tersebut dapat berupa suatu subsistem atau bagian-bagian dari suatu sistem.

2. Batas sistem (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya (yang berada di lingkungan luar dari suatu sistem). Suatu batas dari sistem dapat menunjukkan ruang lingkup dari sistem itu sendiri.

3. Lingkungan Luar Sistem (*Environments*)

Pengertian lingkungan luar sistem adalah hal apapun yang berada di luar batas dari sistem yang dapat mempengaruhi kerja dari suatu sistem operasi. Lingkungan luar tersebut dapat menguntungkan yaitu berupa energi dari sistem yang harus tetap dipelihara. Namun, juga dapat merugikan sehingga harus ditahan dan dikendalikan jika tidak maka akan berdampak negatif terhadap kelangsungan hidup dari suatu sistem.

4. Penghubung (*Interface*)

Penghubung yang dimaksud merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini dapat memungkinkan sumber-sumber daya mengalir dari suatu subsistem ke subsistem lainnya. Keluaran dari subsistemlah yang akan menjadi masukan untuk subsistem yang lainnya melalui penghubung.

5. Masukan (*Input*)

Energi yang dimasukkan ke dalam suatu sistem dapat disebut sebagai masukan. Masukan dapat berupa masukan pemeliharaan serta masukan sinyal. Agar sistem dapat beroperasi maka dimasukkan suatu energi yang disebut masukan pemeliharaan (*maintenance input*). Sedangkan masukan sinyal merupakan suatu energi yang perlu diproses terlebih dahulu untuk mendapatkan suatu keluaran.

6. Keluaran (*Output*)

Hasil dari suatu energi yang telah diolah serta diklasifikasikan menjadi keluaran yang berguna dan menjadi sisa pembuangan disebut sebagai keluaran. Suatu keluaran dapat menjadi masukan untuk subsistem yang lain atau pun kepada supra sistem.

7. Pengolah (*Process*)

Suatu sistem produksi akan mengolah masukan menjadi keluaran. Misalnya seperti bahan baku yang akan diolah menjadi suatu barang yang siap dipakai oleh pengguna. Oleh sebab itu, bagian pengolah dapat menjadi bagian dari suatu sistem.

8. Sasaran atau Tujuan (*Objective or Goal*)

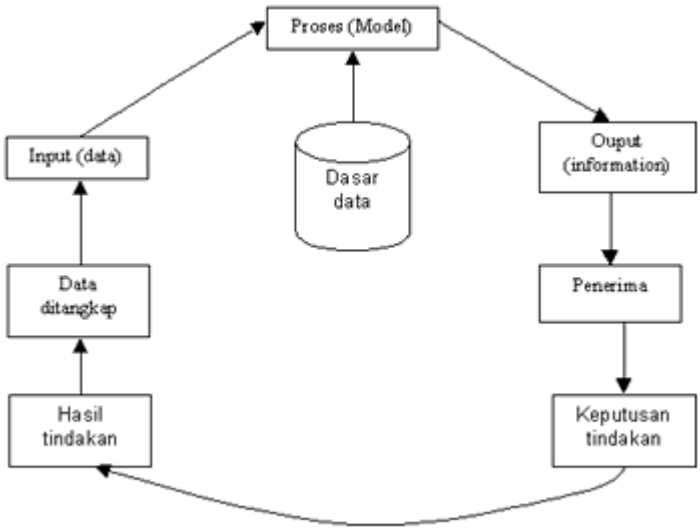
   Suatu sistem pastinya mempunyai sasaran karena kalau tidak maka kinerja dari sistem tersebut tidak akan ada gunanya. Sasaran sistem sangat menentukan tentang masukan yang dibutuhkan beserta keluaran yang akan dihasilkan oleh sistem.

2.1.2 Konsep Dasar Informasi

Informasi sangat penting di dalam suatu organisasi. Suatu informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang mendapatkan informasi tersebut.

2.1.2.1 Siklus Informasi

Data adalah suatu bentuk mentah yang membutuhkan suatu pengolahan lebih lanjut agar dapat diketahui isinya. Suatu informasi dihasilkan dari pengolahan data melalui suatu model. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu. Siklus informasi terdiri dari data yang diolah melalui suatu model menjadi informasi, kemudian penerima informasi tersebut membuat suatu keputusan dan melakukan tindakan yang menghasilkan suatu tindakan lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai masukan kemudian diproses kembali melalui suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus informasi tersebut dapat dilihat pada gambar 2.1.



Gambar 2.1 Siklus Informasi [3]

2.1.2.2 Kualitas Informasi

John Burch dan Gary Grudnitski [3] membentuk bangunan seperti yang dapat dilihat pada gambar 2.2. Bangunan tersebut ditunjang oleh tiga buah pilar sebagai gambaran dari kualitas suatu informasi. Masing-masing pilar tersebut merupakan penunjang dari kualitas suatu informasi yang terdiri dari [3] :

1. Akurat

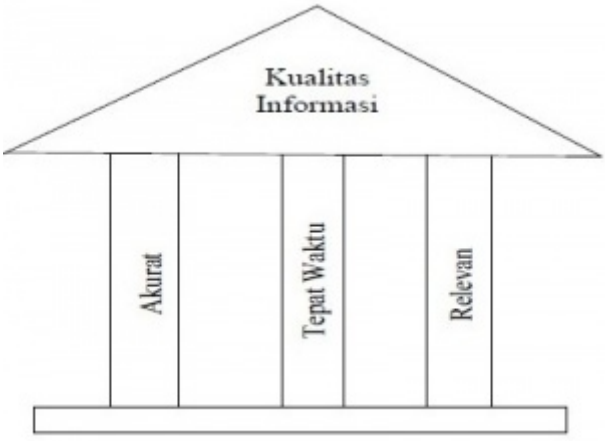
   Suatu informasi dikatakan akurat apabila bebas dari kesalahan serta tidak menyesatkan bagi yang mendapatkan informasi tersebut. Akurat juga dapat memiliki arti bahwa suatu informasi dapat dengan jelas mencerminkan apa yang dimaksud oleh informasi itu sendiri.

2. Tepat pada waktunya

   Informasi yang datang pada penerima tidak boleh terlambat. Hal tersebut yang disebut dengan tepat pada waktunya. Informasi yang dinilai sudah usang tidak akan mempunyai nilai lagi karena suatu informasi merupakan landasan di dalam pengambilan keputusan sehingga apabila terlambat maka dapat berakibat fatal pada suatu organisasi.

3. Relevan

   Relevan mempunyai arti bahwa informasi mempunyai manfaat untuk pemakaiannya. Relevansi informasi dapat berbeda untuk setiap orang. Hal tersebut dikarenakan setiap orang memiliki kebutuhan pemakaian yang berbeda-beda sehingga menyebabkan fungsi serta manfaat yang didapatkan juga berbeda.



Gambar 2.2 Bangunan Kualitas Informasi Beserta Pilarnya [3]

2.1.3 Kegiatan Management

Kegiatan *management* dihubungkan dengan tingkatannya di dalam suatu organisasi. Kegiatan *management* mempengaruhi pengolahan informasi karena informasi yang dibutuhkan berbeda untuk masing-masing tingkatan. Kegiatan *management* untuk masing-masing tingkatan dapat dikategorikan sebagai berikut[3] :

1. Perencanaan strategi, merupakan kegiatan *management* tingkat atas. Perencanaan ini dapat meliputi beberapa hal sebagai berikut :
   a. Proses evaluasi lingkungan luar organisasi
   b. Penetapan tujuan
   c. Penentuan strategi
2. Pengendalian *management*, merupakan kegiatan *management* tingkat menengah.
3. Pengendalian operasi, merupakan kegiatan *management* tingkat bawah.

2.1.3.1 Tipe Keputusan *Management*

Pengambilan keputusan adalah tindakan *management* di dalam pemilihan altenatif untuk mencapai suatu sasaran yang telah ditentukan di awal. Keputusan oleh *management* dapat diklasifikasikan ke dalam tiga tipe, yaitu [3] :

1. Keputusan Tidak Terstruktur

   Keputusan ini sifatnya tidak terjadi berulang-ulang dan tidak selalu terjadi dalam jangka waktu yang sama. *Management* tingkat ataslah yang melakukan keputusan tidak terstruktur ini. Informasi yang didapatkan tidak mudah tersedia dan biasanya berasal dari lingkungan luar sistem.

2. Keputusan Setengah Terstruktur

   Sifat dari keputusan ini adalah sebagian yang dapat diprogram sehingga masih membutuhkan beberapa pertimbangan dari pengambil keputusan. Sifatnya rumit dan membutuhkan perhitungan-perhitungan serta analisis yang terperinci atau mendetail.

3. Keputusan Terstruktur

   Keputusan terstruktur bersifat berulang-ulang dan rutin sehingga dapat diprogram untuk tahap berikutnya yang diharapkan memiliki kemajuan dibandingkan dengan kegiatan sebelumnya. Keputusan ini terjadi dan dilakukan terutama pada *management* tingkat bawah. Contohnya pada struktur kepanitiaan, ketua panitia beserta anggotanya.

2.1.4 Pengembangan Sistem

Pengembangan sistem dapat berarti menyusun suatu sistem baru yang akan dapat menggantikan sistem lama, baik secara keseluruhan atau pun memperbaiki sistem yang telah ada. Pengembangan sistem tersebut dapat terjadi karena beberapa hal, yaitu [3] :

1. Adanya masalah yang timbul di sistem yang lama. Masalah tersebut dapat berupa :
   a. Ketidakberesan

      Ketidakberesan yang terdapat di dalam sistem yang lama dapat menyebabkan sistem tersebut tidak dapat beroperasi sesuai dengan yang

diharapkan. Sehingga pengoperasian sistem tersebut justru dapat memperlambat kinerja suatu organisasi yang sedang menjalankannya.

b. Pertumbuhan organisasi

Pertumbuhan organisasi yang menyebabkan harus disusunnya suatu sistem yang baru sehingga dengan adanya perubahan ini, maka menyebabkan sistem yang lama sudah tidak atau kurang efektif dan tidak dapat lagi memenuhi semua kebutuhan informasi yang dibutuhkan oleh *management*.

2. Untuk Meraih Kesempatan-Kesempatan

Dalam keadaan bersaing, kecepatan informasi atau efisiensi waktu sangat menentukan berhasil atau tidaknya suatu strategi dan rencana-rencana yang telah disusun untuk meraih kesempatan yang ada.

3. Adanya Instruksi-Instruksi

Penyusunan sistem yang baru dapat juga terjadi dikarenakan adanya suatu instruksi dari pimpinan atau pun dari luar organisasi. Seperti misalnya ketua panitia yang merasa perlu meningkatkan kinerja sistem yang sedang berjalan dan memerintahkan pada anggotanya untuk memperbaiki pada kegiatan berikutnya.

2.1.4.1 Prinsip Pengembangan Sistem

Prinsip-prinsip pengembangan sistem yang tidak boleh dilupakan, yaitu[3] :

1. Sistem yang dikembangkan adalah untuk *management*.
2. Sistem yang dikembangkan adalah investasi modal yang besar.
3. Sistem yang dikembangkan memerlukan orang yang terdidik.
4. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengembangan sistem.
5. Proses pengembangan sistem tidak harus urut.
6. Tidak takut untuk membatalkan proyek.
7. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem.

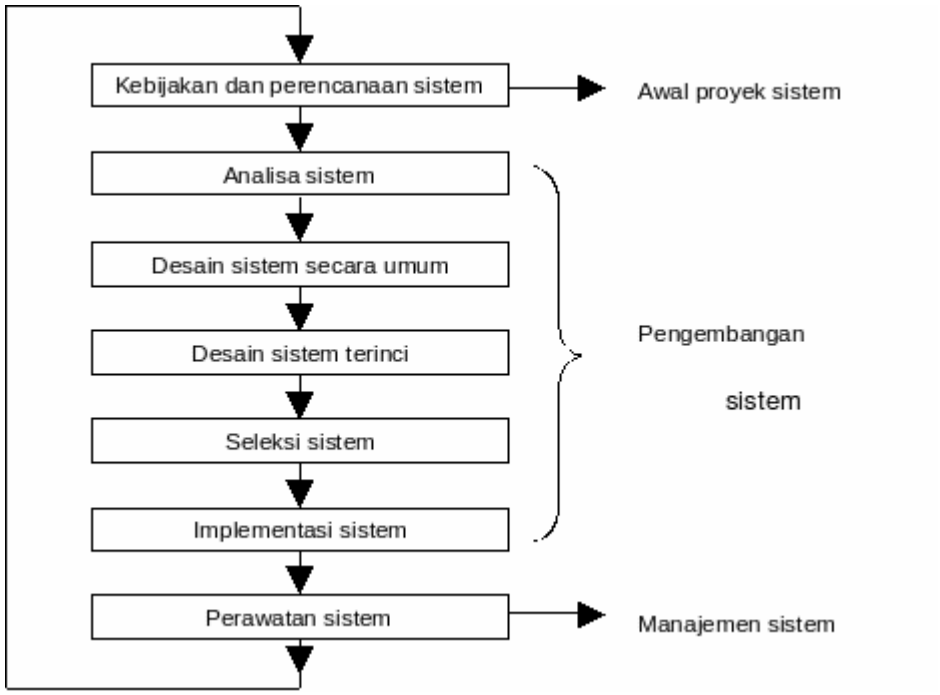
2.1.4.2 Siklus Hidup Pengembangan Sistem

Pengembangan sistem yang menggunakan komputer sebagai alat pelaksanaannya dapat menjadi tugas yang lebih kompleks sehingga membutuhkan banyak sumber daya dan dapat memakan waktu yang lama untuk menyelesaikannya. Proses pengembangan melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan penerapan, pengoperasian, hingga pemeliharaan.

Ide dari siklus hidup sistem merupakan hal yang sederhana dan masuk akal. Dalam siklus tersebut, masing-masing bagian pengembangan sistem dibagi menjadi beberapa tahapan kerja yang mempunyai karakteristik sendiri. Tahapan utama terdiri dari [3] :

1. Perencanaan sistem
2. Analisis sistem
3. Desain sistem
4. Seleksi sistem
5. Implementasi sistem
6. Perawatan sistem

Sistem hidup pengembangan sistem dengan langkah-langkah utama dapat dilihat pada gambar 2.3.



Gambar 2.3 Siklus Hidup Pengembangan Sistem [3]

2.1.5 Content Management System

Content Management System (CMS) adalah suatu aplikasi yang dapat digunakan untuk mengubah, mengurutkan, mengelola berbagai *content* yang jenisnya dapat berupa teks, gambar, suara, maupun video [4]. Hal tersebut dapat dilakukan sesuai dengan keinginan dari pihak yang berhubungan langsung dengan suatu *website*. Biasanya CMS dikenal dalam mengelola *content website* namun tidak menutup kemungkinan jika aplikasi desktop pun dapat menerapkan CMS ke dalam aplikasinya. Penggunaan CMS tidak memerlukan pengetahuan pemrograman *web* yang andal karena sudah *user friendly* [5].

2.1.5.1 Elemen Penting Dalam CMS

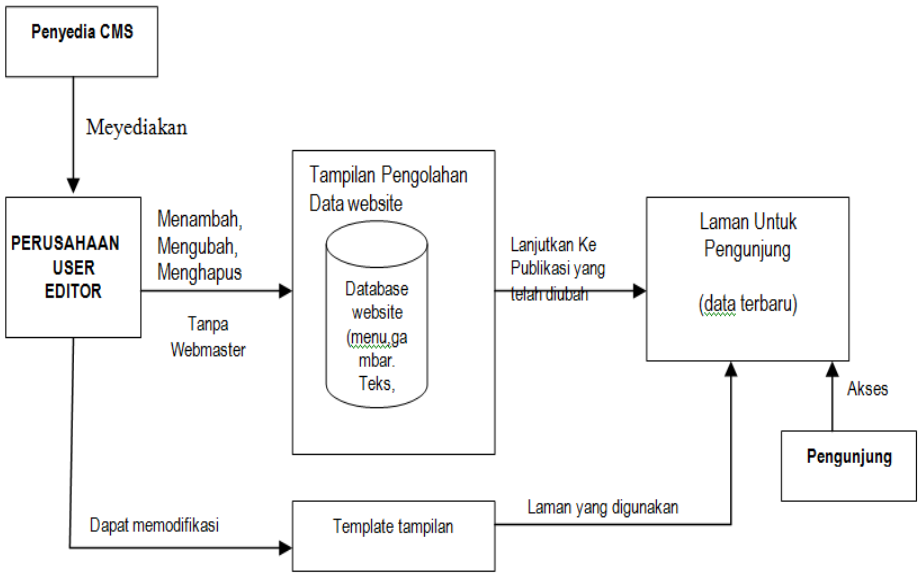
Pada umumnya suatu *content management system* mempunyai dua buah elemen yang penting, yaitu [4] :

1. Aplikasi *management content* merupakan suatu elemen dari CMS yang dapat berupa aplikasi yang dapat digunakan untuk mengatur *content-content* yang akan dimanfaatkan dalam membangun suatu *website* atau aplikasi desktop yang berbasis CMS.

2. Aplikasi pengiriman *content* merupakan suatu elemen dari CMS berupa aplikasi yang dapat digunakan untuk mengirimkan *content-content* yang akan digunakan dalam membangun suatu *website* atau aplikasi desktop yang berbasis CMS.

2.1.5.2 Cara Kerja CMS

Gambaran cara kerja CMS pada umumnya dapat dilihat pada gambar 2.4.



Gambar 2.4 Skema Siklus Pengelolaan *Website* [9]

2.1.6 Model Analisis

Model analisis yang digunakan untuk mengembangkan sistem pengelolaan *paper* seminar penelitian menggunakan *content management system* berbasis *web* adalah sebagai berikut :

2.1.6.1 Flowchart

Flowchart merupakan bagan yang dapat digunakan untuk menunjukkan aliran di dalam suatu program atau prosedur sistem secara logis atau sesuai kenyataan. Suatu *flowchart* dapat mengikuti pedoman-pedoman berikut ini [3] :

1. *Flowchart* sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowchart* harus ditunjukkan dengan jelas.

3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan di dalam *flowchart* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan di dalam *flowchart* harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakanlah simbol-simbol *flowchart* yang standar.

Ada lima macam *flowchart* yang ada, yaitu [3] :

1. *Systems flowchart* (bagan alir sistem)

   Merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan tentang urutan dari prosedur yang ada di dalam sistem serta menunjukkan apa saja yang dikerjakan di dalam suatu sistem.
2. *Document flowchart* (bagan alir dokumen)

   Merupakan bagan yang menunjukkan arus dari laporan dan formulir termasuk tembusannya. Bagan ini juga sering disebut sebagai *flowmap*.
3. *Schematic flowchart* (bagan alir skematik)

   Merupakan *flowchart* yang hampir sama dengan *systems flowchart* yang menggambarkan prosedur di dalam sistem. Yang membedakannya adalah pada *schematic flowchart* selain menggunakan simbol pada *systems flowchart* namun juga menggunakan gambar komputer dan peralatan lain yang digunakan dengan maksud agar memudahkan komunikasi pada orang yang kurang paham dengan simbol-simbol *system flowchart*.
4. *Program flowchart* (bagan alir program)

   Merupakan bagan yang menjelaskan secara rinci mengenai langkah-langkah dari suatu proses program. Program *flowchart* dibuat dari derivikasi [3] *flowchart* sistem.

5. *Process flowchart* (bagan alir proses)

   Merupakan *flowchart* yang banyak digunakan di teknik industri namun juga berguna bagi seorang analis sistem untuk menggambarkan proses yang terjadi di dalam suatu prosedur yang diterapkan pada sistem yang akan dibangun.

2.1.6.2 Entity Relationship Diagram

Model E-R adalah "suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antarentitas" [7].

Model ini biasanya dinyatakan dalam bentuk diagram sehingga sering dikenal dengan istilah *entity relationship diagram* (ERD). Berikut ini merupakan komponen yang umumnya terdapat dalam pembentukan suatu ERD [7].

1. Entitas merupakan sesuatu di dalam dunia nyata yang keberadaannya tidak bergantung pada yang lain.
2. Atribut merupakan properti atau karakteristik yang terdapat pada setiap entitas.
3. Tipe entitas dan instans entitas. Tipe atau himpunan entitas adalah sebuah kumpulan dari entitas yang saling berbagi atribut. Sedangkan instans entitas adalah satu kejadian dalam sebuah tipe entitas.
4. Atribut pengenal. Setiap instans entitas pasti memiliki suatu nilai untuk setipa atributnya yang nilainya bersifat unik sehingga dapat digunakan untuk membedakan antara satu entitas dengan entitas yang lainnya.
5. Jenis atribut terdiri dari tiga kategori, yaitu atribut sederhana versus atribut komposit, atribut bernilai-tunggal versus atribut bernilai banyak, dan atribut tersimpan versus atribut turunan.
6. Relasi yang berfungsi untuk menyatakan keterkaitan antara beberapa tipe entitas. Jenis kerelasian antar entitas dikelompokkan dalam tiga jenis [13] , yaitu :
   a. Kerelasian jenis satu ke satu (*one to one*). Misalkan satu peserta hanya dapat membuat satu *paper*. Sedangkan satu *paper* (satu judul

paper) hanya boleh dibuat oleh satu orang. Maka hubungan tersebut dinamakan kerelasian satu ke satu.

b. Kerelasian jenis banyak ke satu (*many to one*) atau satu ke banyak (*one to many*). Misalkan satu peserta dapat mengirimkan tiga *paper*. maka hubungan tersebut dinamakan kerelasian banyak ke satu. Kerelasian ini juga dapat berarti satu ke banyak karena pengertiannya sama saja.
c. Kerelasian jenis banyak ke banyak (*many to many*). Misalkan satu *paper* di *review* oleh tiga *reviewer* atau lebih. Sedangkan satu *review*er dapat me*review* banyak *paper*. Maka hubungan tersebut dinamakan kerelasian banyak ke banyak.

2.1.6.3 Enhanced Entity Relationship

Enhanced entity relationship (EER) merupakan pengembangan dari *entity relationship diagram*. Pada EER dikenal istilah supertipe (generalisasi) dan subtipe (spesialisasi). Penotasian dasar dalam hubungan generalisasi atau spesialisasi adalah sebagai berikut [7] :

1. Bentuk kotak : menyatakan generalisasi atau spesialisasi.
2. Bentuk lonjong : menggambarkan atribut.
3. Bentuk lingkaran : menghubungkan antara generalisasi atau spesialisasi ke dalam lingkaran.
4. Tanda lengkungan seperti U : menyatakan spesialisasi, sedangkan yang tidak ditandai U menyatakan generalisasi.

EER ini sama dengan model data *hierarchycal* [13] karena bentuknya yang bisa dikatakan seperti struktur pohon. Hanya saja simbol yang digunakan berbeda namun memiliki makna yang sama. Misalkan satu universitas memiliki fakultas a,b,c. Sedangkan satu fakultas dapat memiliki jurusan d,e,f. Hal ini sesuai dengan pengertian generalisasi dan spesialisasi.

Notasi pada EER bisa juga disamakan dengan notasi ISA (is a). menggunakan simbol segitiga dengan tulisan ISA di dalamnya. Jika segitiga

membentuk kerucut maka hubungan tersebut merupakan generalisasi, jika sebaliknya maka merupakan spesialisasi.

2.1.6.3.1 Memetakan EER ke Tabel

Cara untuk memetakan EER ke dalam sebuah hubungan dapat dilihat pada penjelasan berikut ini [7]. Untuk menjelaskannya diperlukan contoh tabel misalnya pengguna sebagai tabel generalisasi dan tabel peserta dengan tabel panitia sebagai tabel spesialisasi :

1. Buat suatu tabel bernama tabel pengguna. Jangan masukkan field terlebih dahulu.
2. Barulah kemudian masukkan field termasuk *primary key* ke dalam tabel pengguna. Misalkan field no_pengguna, username, password.
3. Tambahkan ke dalam tabel pengguna salah satu di antara ini :
   a. Pembeda subtipe (aturan disjoint)
      Misal : field kategori_pengguna
   b. Pengkualifikasi (aturan overlap)
      Misal : field ketua dan field anggota panitia
4. Masukan *primary key* dari tabel pengguna ke dalam tabel peserta dan juga panitia. Tabel peserta dan tabel panitia masing-masing memiliki *primary key* no_pengguna.
5. Masukan atribut field yang berbeda ke tabel peserta dan panitia. Misalkan untuk tabel peserta. Misalkan tabel peserta memiliki field nama_peserta dan tabel panitia memiliki field nama _panitia.

Agar lebih mudah maka akan dipakai aturan disjoint yang memakai field kategori_pengguna. Dari kategori pengguna ini dapat menentukan speasialisasi yang terkait antara peserta atau panitia. Apabila kategori_pengguna berisi ‘Peserta’ atau nilai yang dapat merepresentasikan bahwa kategori peserta tersebut adalah peserta maka pasangan dari suatu baris dalam pengguna terdapat pada tabel peserta. Begitupun halnya dengan tabel panitia.

2.1.6.4 Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan dapat digunakan untuk menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari suatu DFD atau dapat disebut juga sebagai DFD level nol yang menggambarkan secara keseluruhan masukan ke dalam sistem maupun keluaran dari sistem. Sistem akan dibatasi oleh *boundary* yang dapat digambarkan dengan garis putus. Dalam diagram konteks hanya boleh terdapat satu proses dan tidak diperbolehkan adanya *data store* dalam diagram konteks [8].

2.1.6.5 Data Flow Diagram

Data flow diagram (DFD) merupakan gambaran suatu sistem yang ada atau pun sistem baru yang akan dikembangkan tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau disimpan. Beberapa simbol yang digunakan dalam DFD antara lain [3] :

1. Kesatuan Luar (*external entity*) atau Batas Sistem (*boundary*)

   Fungsi batas sistem adalah untuk memisahkan suatu sistem dengan lingkungan luarnya dimana sistem akan menerima masukan dan menghasilkan keluaran kepada lingkungan luarnya.

2. Arus Data (*data flow*)

   Arus data mengalir di antara proses, simpanan data dan kesatuan luar yang menunjukkan arus dari data yang dapat berupa masukan atau hasil dari proses sistem.

3. Proses (*process*)

   Proses adalah suatu kegiatan yang dilakukan oleh orang, mesin maupun komputer dari hasil suau arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

4. Simpanan Data (*data store*)

   Simpanan data merupakan simpanan dari data yang dapat berupa sebagai berikut ini [3] :

   a. Suatu file atau database di sistem komputer
   b. Suatu arsip atau catatan manual

c. Suatu kotak tempat data di meja seseorang
d. Suatu tabel acuan manual
e. Suatu agenda atau buku

2.1.6.6 Kamus Data

Kamus data merupakan sebuah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi [3]. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang masukan, keluaran, dan merancang *database* program. Kamus data dibuat berdasarkan arus dari data yang ada [8].

2.1.7 Bentuk Laporan

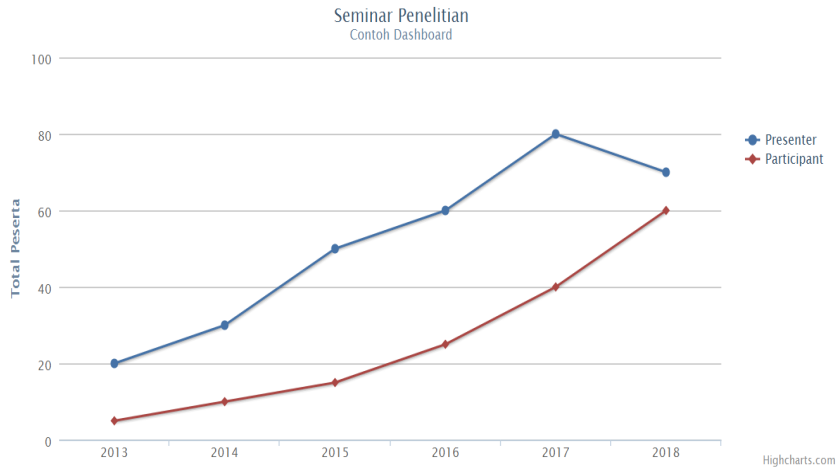
Bentuk laporan yang dihasilkan oleh sistem contohnya yang paling banyak digunakan adalah sebagai berikut :

2.1.7.1 Laporan Berbentuk Grafik

Laporan dalam bentuk grafik dapat diklasifikasikan sebagai berikut [3] :

1. *Line chart* (bagan garis)

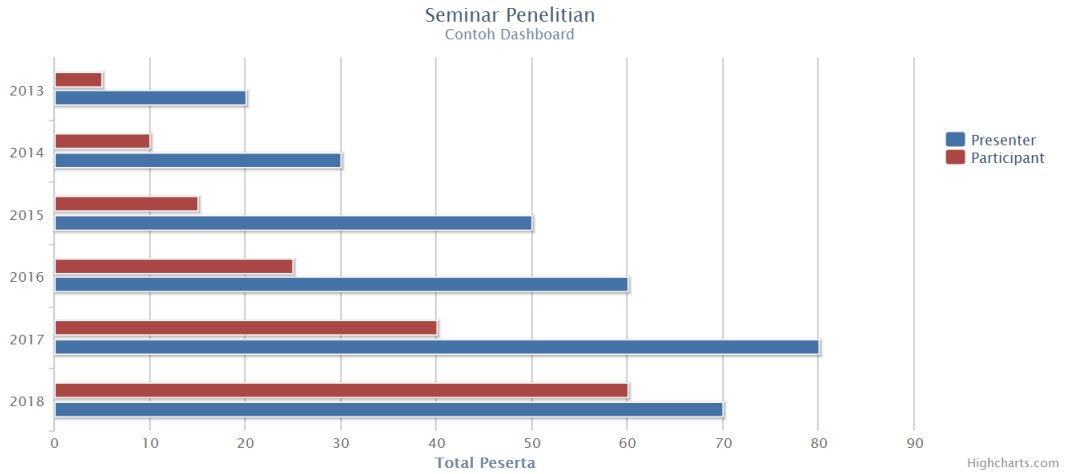
   Contoh laporan berbentuk bagan garis dapat dilihat pada gambar 2.5.



Gambar 2.5 Laporan Grafik Line Chart

2. *Bar chart* (bagan batang)

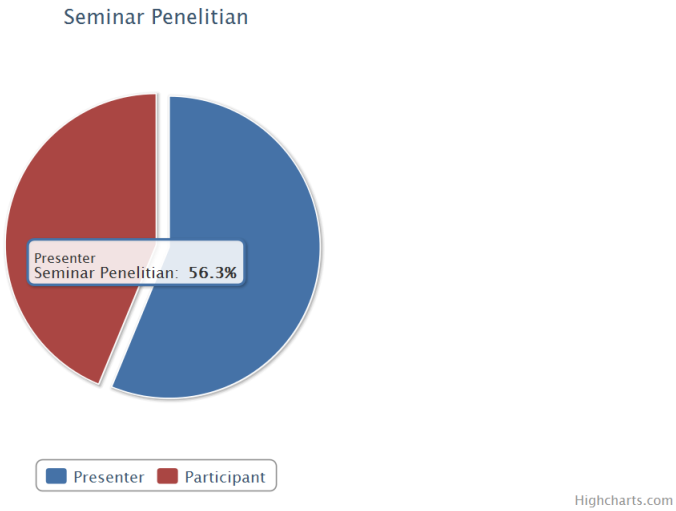
   Contoh laporan berbentuk bagan batang dapat dilihat pada gambar 2.6.



Gambar 2.6 Laporan Grafik Bar Chart

3. *Pie chart* (bagan pastel)

   Contoh laporan berbentuk bagan pastel dapat dilihat pada gambar 2.7.



Gambar 2.7 Laporan Grafik Pie Chart

2.1.7.1.1 Dashboard

Di antara berbagai macam laporan berbentuk grafik, *dashboard* merupakan salah satu contoh. Pada umumnya, kebanyakan *dashboard* sistem ditampilkan dalam bentuk bagan batang. *Dashboard* adalah tampilan

visual dari informasi terpenting yang diperlukan untuk mencapai satu atau lebih obyektif, dikonsolidasikan dan diatur dalam sebuah layar sehingga informasi dapat diawasi dalam sekilas. Tujuan *dashboard* adalah agar seseorang dapat secara efisien terhubung dengan informasi yang diperlukan untuk melakukan sesuatu atau menjalankan tujuan tertentu [6].

Ada beberapa macam tipe *dashboard*. Lebih spesifik dijelaskan bahwa menurut Rasmussen, Bansal dan Chen [10] yang mengemukakan tentang tiga buah tipe *dashboard*, yaitu :

1. *Dashboard* Strategis

   "*Dashboard* strategis digunakan untuk mendukung *management* level strategis memberikan informasi dalam membuat keputusan bisnis, memprediksi peluang, dan memberikan arahan pencapaian tujuan strategis" [10].

2. *Dashboard* Taktis

   "*Dashboard* tipe ini berfokus pada proses analisis untuk menentukan penyebab dari suatu kondisi atau kejadian tertentu" [10].

3. *Dashboard* Operasional

   "*Dashboard* operasional yang berfungsi sebagai pendukung pengawasan dari aktifitas proses bisnis yang spesifik. Fokus pada pengawasan aktifitas dan kejadian yang tidak berubah secara konstan" [10].

2.1.8 Teori Kuesioner

Kuesioner atau dikenal juga dengan sebutan angket adalah suatu "teknik pengumpulan data yang dilakukan dengan cara memberi seperangkat pertanyaan atau pernyataan tertulis kepada responden untuk dijawabnya" [11].

2.1.8.1 Kuesioner

Menurut pernyataan Uma Sekaran [11] prinsip-prinsip yang biasanya terdapat dalam suatu proses penulisan kuesioner sebagai salah satu cara dari teknik pengumpulan data, sebagai berikut [11] :

1. Prinsip Penulisan Kuesioner

   Prinsip penulisan kuesioner ini mengenai beberapa faktor berikut ini :

a. Isi dan tujuan pertanyaan

Jika suatu pertanyaan merupakan sebuah bentuk dari pengukuran yang ingin dilakukan maka pertanyaan tersebut haruslah akurat sehingga nantinya akan mudah dalam melakukan perhitungan variabel yang sedang diteliti.

b. Bahasa yang digunakan

Jangan menyusun kuesioner dengan menggunakan bahasa yang tidak dimengerti oleh responden. Hal-hal yang harus diperhatikan dalam bahasa kuesioner seperti jenjang pendidikan, sosial budaya, dan juga "*frame of reference*" [11].

c. Tipe dan bentuk pertanyaan

Tipe pertanyaan dibagi menjadi dua, yaitu [11] :

a) Pertanyaan terbuka

Kuesioner tipe ini tidak memberikan batasan tentang jawaban yang harus diberikan responden. Cara ini membutuhkan waktu lebih lama dan sulit untuk mengukur tingkat pencapaian sesuatu yang ingin diukur menggunakan kuesioner.

b) Pertanyaan tertutup

Kuesioner yang tertutup dapat memudahkan pihak peneliti dalam mengukur jawaban-jawaban responden karena sudah diberikan batasan untuk responden dalam menjawab.

d. Pertanyaan tidak mendua

Jangan menggunakan kata penghubung "dan" di tengah suatu pertanyaan kuesioner. Hal tersebut akan membuat bingung responden yang hendak menjawab.

e. Tidak menanyakan yang sudah lupa

Untuk pertanyaan yang diajukan hendaknya tidak membahas tentang sesuatu yang sudah terjadi puluhan tahun yang lalu sehingga pikiran responden menjadi berat karena harus mengingat kembali peristiwa tersebut.

f. Pertanyaan tidak menggiring

Pertanyaan yang dikategorikan baik adalah pertanyaan yang tidak menggiring ke suatu jawaban yang baik saja atau pun yang jelek saja.

g. Panjang pertanyaan

Jumlah pertanyaan yang dikatakan memadai adalah antara 20 sampai dengan 30 pertanyaan. Hal tersebut bertujuan untuk menghindari kebosanan responden dalam mengisi kuesioner.

h. Urutan pertanyaan

Ada tiga urutan pertanyaan yang boleh dituliskan di dalam kuesioner, yaitu [11] :

a) Dari umum ke khusus
b) Dari mudah ke sulit
c) Acak

i. Prinsip pengukuran

Instrument kuesioner harus dapat digunakan untuk mendapatkan data yang valid dan reliabel tentang variabel yang akan diukur.

j. Penampilan fisik kuesioner

Jika hendak mendapatkan keseriusan responden dalam menjawab maka buatlah kuesioner dalam media yang akan membuat responden menjadi berkesan baik.

2.1.8.2 Skala Likert

Skala likert ini biasanya “digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial” [11].

Dengan menggunakan skala Likert, maka variabel akan dijabarkan menjadi suatu indikator yang akan dijadikan sebagai suatu bahan untuk tolak ukur dalam menyusun item-item instrumen pertanyaan atau pernyataan.

Biasanya jawaban responden dalam skala Likert dinyatakan dari negatif, netral sampai positif dalam bentuk sebagai berikut [11] :

a. Sangat setuju (SS)
b. Setuju (S)

c. Ragu-ragu / Netral (RG / N)
d. Tidak setuju (TS)
e. Sangat tidak setuju (STS)

Setiap jawaban diberikan nilai yang bertujuan agar dapat memudahkan perhitungan. Nilai yang diberikan tidak tergantung pada nilai tertentu saja karena angka yang dipakai hanya merupakan simbol dan bukan untuk menunjukkan nilai yang sebenarnya [16]. Salah satu contohnya adalah sebagai berikut [11] :

a. Sangat setuju diberi nilai 5
b. Setuju diberi nilai 4
c. Ragu-ragu / Netral diberi nilai 3
d. Tidak setuju diberi nilai 2
e. Sangat tidak setuju diberi nilai 1

Instrument penelitian skala Liker terdiri dari bentuk *checklist* atau pun pilihan ganda seperti pada contoh berikut ini [11] :

a. Contoh Bentuk *Checklist*

Contoh bentuk checklist dari kuesioner menurut skala Likert dapat dilihat pada tabel 2.1.

Tabel 2.1 Contoh Skala Likert Bentuk Checklist [11]

No	Pertanyaan	Jawaban				
		SS	S	RG	TS	STS
1	Sistem pengelolaan paper yang baru akan diterapkan pada kepanitiaan?	√				
2	<pertanyaan selanjutnya>	…	…	…	…	…

Kemudian dengan teknik pengumpulan data angket, maka instrumen tersebut misalnya diberikan kepada 50 orang mahasiswa dan 50 orang panitia. Dari 100 orang mahasiswa dan panitia setelah dilakukan analisis misalnya [11] :

15	orang menjawab	SS
50	orang menjawab	S
15	orang menjawab	RG
15	orang menjawab	TS
5	orang menjawab	STS

Data interval tersebut dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan nilai setiap jawaban dari responden. Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut [11].

Jumlah skor untuk 25 orang yang menjawab SS	=	15 x 5	= 75
Jumlah skor untuk 40 orang yang menjawab ST	=	50 x 4	= 200
Jumlah skor untuk 5 orang yang menjawab RG	=	15 x 3	= 45
Jumlah skor untuk 20 orang yang menjawab TS	=	15 x 2	= 30
Jumlah skor untuk 10 orang yang menjawab STS	=	5 x 1	= 5
Jumlah total			= 355

Jumlah nilai ideal untuk seluruh item adalah 5 x 100 = 500. Jumlah skor yang diperoleh dari penelitian = 355. Jadi berdasarkan data tersebut, maka tingkat persetujuan terhadap sistem pengelolaan paper yang baru itu = (355 : 500) x 100% = 71% dari total yang diharapkan (100%) [11].

b. Contoh Bentuk Pilihan Ganda

Sistem pengelolaan paper yang baru akan diterapkan pada kepanitiaan?

a. Sangat setuju
b. Setuju
c. Ragu-ragu / Netral
d. Tidak setuju

e. Sangat tidak setuju

Dari hasil perhitungan nilai lebih lanjut maka dapat digambarkan sebagai berikut seperti yang terlihat pada gambar 2.8 :



Gambar 2.8 Gambaran Perhitungan Nilai Skala Likert [11]

Gambar tersebut menunjukkan bahwa 355 dari 500 nilai responden total yang ada terletak pada daerah setuju.

2.1.9 Bahasa Pemrograman

Bahasa pemrograman merupakan bahasa yang dapat dimengerti oleh komputer dalam membuat program. Berikut ini adalah penjelasan mengenai bahasa pemrograman yang digunakan dalam proses pengembangan sistem pengelolaan seminar penelitian menggunakan *content management system* berbasis *web*.

2.1.9.1 SQL

SQL (*Structured Query Language*) adalah "bahasa standar yang digunakan untuk melakukan akses dan manipulasi database" [12].

Berikut ini adalah contoh dari perintah-perintah dasar yang terdapat dalam SQL yang dapat dilihat pada tabel 2.2.

Tabel 2.2 Contoh Perintah Dasar SQL

No	Nama Perintah	Perintah SQL	Contoh	Keterangan
1	Create database	`Create database <nama_database>;`	`Create database cms;`	Digunakan untuk membuat sebuah database.
2	Use database	`Use <nama_database`	`Use cms;`	Digunakan untuk beralih ke database yang ingin

No	Nama Perintah	Perintah SQL	Contoh	Keterangan
		>;		dipakai.
3	Drop database	Drop database <nama_database >;	Drop database cms;	Digunakan untuk menghapus database.
4	Create tabel	Create tabel <nama tabel>(Nama_field karakter_tipe_ data(jumlah karakter) kata tambahan, Nama_field karakter_tipe_ data(jumlah karakter) kata tambahan);	Create tabel peserta (Kode_peserta char(5) primary key not null, Nama varchar(50) not null, Jenis_kelami n enum('P','W'));	Digunakan untuk membuat tabel beserta strukturnya.
5	Desc tabel	Desc <nama_tabel>;	Desc peserta;	Digunakan untuk melihat struktur tabel.
6	Drop tabel	Drop <nama_tabel>;	Drop peserta;	Digunakan untuk menghapus tabel.
7	Alter Tabel	Alter tabel <nama_tabel> change/add foreign key/add primary key <field_tabel_l ama> <field_tabel_b aru>;	Alter tabel peserta change jenis_kelami n jk enum('P','W') default 'P';	Digunakan untuk mengubah field tabel, menambah primary key atau foreign key.
8	Insert	Insert into <nama_tabel> (<nama_field yang akan	Insert into peserta (kode_pesert a, nama,	Digunakan untuk memasukan data ke dalam field tabel.

No	Nama Perintah	Perintah SQL	Contoh	Keterangan
		diinsert>) values (<data yang akan diinsert>);	jenis_kelamin) values (‘12345’,’Mper’,’W’);	
9	*Update*	*Update* <nama_tabel> se<nama_field> = <nilai_field_lama> where <nama_field> = <nilai_field_baru>;	*Update* peserta set nama = ‘Ferawati’ where nama = ‘Mper’;	Digunakan untuk mengubah isi field tabel.
10	Delete	Delete from <nama_tabel> where <nama_field> = <nilai_field yang akan dihapus>;	Delete from peserta where kode_peserta = ‘12345’;	Digunakan untuk menghapus isi field tertentu.
11	Select	Select <nama_field> from <nama_tabel>;	Select * from peserta;	Digunakan untuk menampilkan seluruh atau sebagian dari isi field tabel.
12	View	Create view <nama_view> as <query>;	Create view user as Select kode_peserta, nama from peserta;	Digunakan sebagai bentuk representasi data yang dapat dibuat dengan melibatkan data pada satu atau beberapa tabel.
13	Trigger	Create trigger <nama_trigger> <before | after> <insert | *update* |	Create trigger trig_*update*_nama After *update* on peserta	Digunakan untukmengubah data sebelum terjadinya suatu proses atau untuk memberikan nilai default.

No	Nama Perintah	Perintah SQL	Contoh	Keterangan
		delete> On <nama_tabel> For each row <statement sql>;	For each row Insert into history_nama Values (now(), OLD.nama, user());	
14	Join	Select <nama_field> from <nama_tabel_satujoin <nama_tabel_dua> on <nama_field yang sama>;	Select * from peserta join *paper* on peserta.nip = *paper*.nip;	Digunakan untuk menggabungkan beberapa tabel yang mempunyai kesamaan field tabel.
15	Union	Select <nama_field> from <nama_tabel_satu> Union Select <nama_field> from <nama_tabel_dua>;	Select * from peserta Union Select * from panitia;	Digunakan untuk menggabungkan hasil eksekusi beberapa SQL select sehingga menjadi satu kesatuan hasil.
16	Subquery	Select <nama_field> from <nama_tabel> where <nama_field> = (query select);	Select * from peserta where kode_peserta =(select kode_peserta from peserta where kode_peserta	Digunakan untuk menangani masalah yang kompleks sehingga sulit untuk dilakukan hanya dengan sebuah query.

No	Nama Perintah	Perintah SQL	Contoh	Keterangan
			='12345');	

2.1.9.2 PHP

PHP (*Personal Home Page*) adalah bahasa pemrograman server side yang membuat dokumen HTML secara *on the fly* [12] dieksekusi di server *web*. Jika melakukan pemeliharaan suatu situs *web* maka PHP menjadi pilihan yang akan mempermudah proses pemeliharaan tersebut. Pada dasarnya, PHP mampu mengerjakan hal-hal yang dapat dikerjakan oleh program CGI (*Common Gateway Interface*). Berikut ini adalah contoh kode php.

```
<?php
      Include('include/library.php');
?>
```

Berikut ini adalah disertakan daftar *database* yang telah didukung oleh bahasa pemrograman PHP [12] :

1. Adabas D
2. dBase
3. Empress
4. FilePro (read only)
5. FrontBase
6. Hyperwave
7. IBM DB2
8. Informix
9. Ingres
10. Interbase
11. MSQL
12. Direct MS SQL
13. MySQL
14. ODBC
15. Oracle (OCI7 dan OCI8)
16. Ourimos
17. PostgreSQL
18. Solid
19. SQLite
20. Sybase
21. Velocis
22. Unix DBM

2.1.9.3 Javascript

Javascript merupakan bahasa skrip yang terkenal di internet dan dapat bekerja dengan baik di hamper semua *browser*. Javascript bisa ditempatkan pada halaman html dengan ditambahkan tag <script> dan diakhiri dengan </script>. Contoh penulisan Javascript adalah sebagai berikut :

```
<script type=”text/javascript”>
      Alert(“Hello World”);
</script>
```

2.1.9.4 CSS

CSS (*Cascading Style Sheet*) merupakan suatu kumpulan perintah yang dibentuk dan disusun menurut urutan tertentu sehingga mampu mengatasi perbedaan style. CSS dapat mengatur warna teks, ukuran gambar, border, warna mouse over dan parameter lainnya. Dengan menggunakan CSS sehingga memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda. Contoh penulisan CSS adalah sebagai berikut :

```
body
{
      font-family:Verdana, Geneva, Arial, Helvetica,
sans-serif;
      font-size:12px;
}
```

2.1.10 Perangkat Lunak Pendukung

Perangkat lunak yang dijadikan alat untuk membantu pengembangan sistem pengelolaan *paper* seminar penelitian menggunakan *content management system* berbasis *web* adalah sebagai berikut :

2.1.10.1 MySQL

MySQL adalah perangkat lunak RDBMS (*Relational Data Base Management System*) yang dapat mengelola data dalam database dalam jumlah yang sangat besar dengan sangat cepat, dapat diakses oleh banyak pengguna dan dapat melakukan suatu proses secara bersamaan [15].

MySQL yang diciptakan oleh David Axmark, Allan Larson dan Michael Widenius [4] ini mempunyai lisensi GNU General Public License (GPL) tapi lisensi tersebut dapat juga dijual untuk beberapa kasus tertentu karena penggunaannya yang tidak cocok dengan penggunaan perangkat GPL.

Pada tanggal 16 Januari 2008, Sun Microsystems Inc mengumumkan aksi korporasi - akuisisi terhadap MySQL sehingga menjadikan Sun sebagai salah satu perusahaan dengan produk platform open source terbesar seperti Java, OpenSolaris dan akhirnya MySQL. Berselang setahun kemudian, tepatnya pada tanggal 20 April 2009 giliran Oracle melakukan akusisi terhadap Sun Microsystems. Sejak saat itu berkembang isu Oracle akan mematikan MySQL.

Berikut ini dijelaskan beberapa alasan utama dalam penggunaan MySQL[15] :

1. Fleksibel

   Artian MySQL yang fleksibel adalah baik aplikasi web atau desktop dapat dibuat menggunakan MySQL. Dalam MySQL juga mendukung stired procedure, fungsi, trigger, view, dan lain-lain

2. Performa tinggi

   MySQL dapat memproses transaksi dengan sangat cepat karena MySQL memiliki suatu mesin query dengan performa yang tinggi.

3. Lintas platform

   MySQL dapat digunakan pada sistem operasi Microsoft windows, linux maupun unix yang menyebabkan proses migrasi data dapat dilakukan lebih mudah jika ingin melakukan migrasi antarsistem operasi.

4. Gratis

   MySQL dapat digunakan secara gratis namun tentunya yang gratis tidak memiliki fitur selengkapnya MySQL yang bersifat komersial.

5. Proteksi data yang handal

   MySQL menyediakan fasilitas seperti manajemen pengguna, enkripsi data dan lainnya yang dapat digunakan sebagai mekanisme yang penuh tenaga dikarenakan suatu perlindungan terhadap keamanan

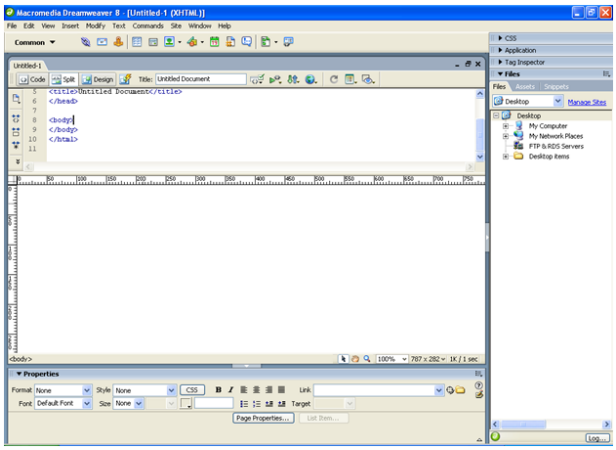
data merupakan hal yang sangat penting bagi seorang profesional database.

6. Komunitas luas

   MySQL memiliki banyak komunitas yang siap membantu jika terdapat suatu permasalahan dalam proses pengolahan data menggunakan MySQL.

2.1.10.2 Macromedia Dreamweaver 8

Macromedia dreamweaver adalah perangkat lunak sebagai *editor* bahasa pemrograman. Versi yang digunakan dalam pengembangan sistem pengelolaan *paper* ini adalah versi 8. Walau pun versi tersebut sudah lama tapi tetap mampu memenuhi kebutuhan *programmer* dalam membuat kode program. Berikut ini adalah tampilan utama layar ketika membuka program Macromedia Dreamweaver 8 seperti pada gambar 2.9.



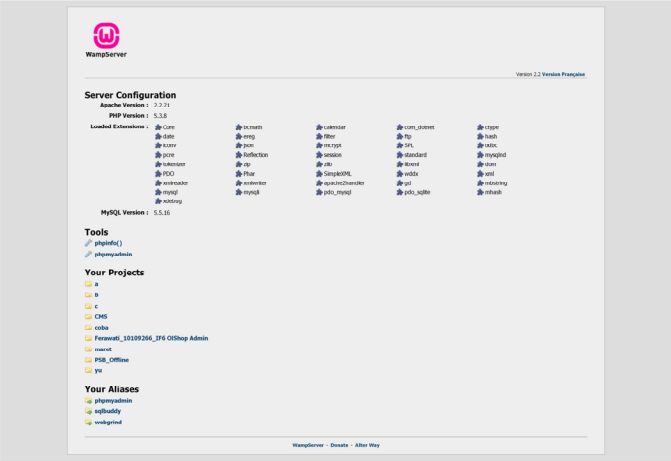
Gambar 2.9 Tampilan Utama Macromedia Dreamweaver 8

2.1.10.3 Wamp Server

Wamp server adalah *web* server *offline* yang dapat dijadikan sebagai server saat membuat file PHP di komputer. Pengertian web server itu sendiri adalah sebagai berikut [14] :

Sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

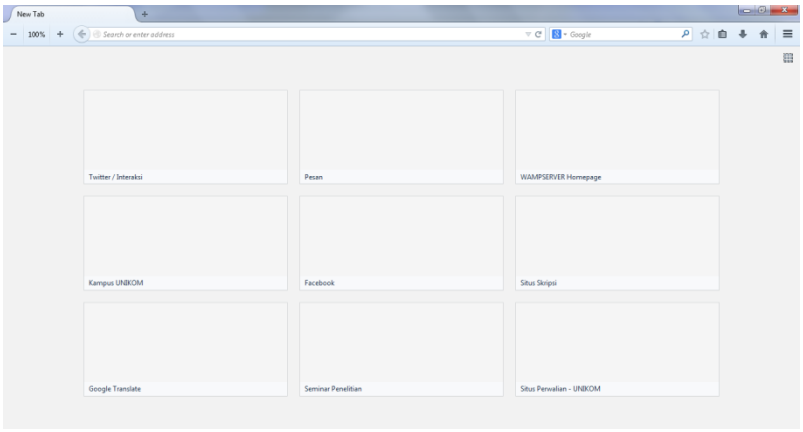
Salah satu kegunaan dari *web* server *offline* ini dapat membantu kita membuat file server tanpa harus terhubung ke koneksi internet. Untuk menampilkan file *web* yang telah dibuat jangan lupa untuk menyimpannya di folder wamp – www – lalu buatlah folder *web* sendiri. Untuk mengakses melalui *browser* ketikan di bagian url dengan localhost/nama folder maka akan tampil index dari halaman *web*. Untuk menampilkan phpmyadmin hanya dengan mengetikkan localhost/phpmyadmin pada bagian url di *browser*. Tampilan wamp server dapat dilihat pada gambar 2.10.



Gambar 2.10 Tampilan Utama Localhost Wamp Server

2.1.10.4 Mozilla Firefox

Mozilla firefox (rubah api) adalah salah satu *browser* yang cukup dikenal di kalangan pengguna internet. Akses *domain* apa saja, dan mencari melalui mesin pencari apa saja juga dapat diatur di *web browser* ini. Untuk tampilan *browser*nya sendiri dapat dilihat pada gambar 2.11.



Gambar 2.11 Tampilan Utama Mozilla Firefox