**BAB II**

**LANDASAN TEORI**

* 1. **Tinjauan Pustaka**
		1. **Konsep Dasar Sistem**

Menurut Ludwig dalam Rochaety dkk. (2011:3) “Sistem adalah seperangkat unsur yang saling berhubungan dan saling mempengaruhi dalam satu lingkungan tertentu”.

Setiap sistem memiliki tujuan. Tujuan inilah yang menjadi daya dorong atau motivasi yang mengarahkan kearah mana sistem bergerak. Tujuan antara satu sistem dengan sistem yang lain berbeda termasuk berlaku juga dalam sistem informasi.

Sekalipun tujuan dari setiap sistem berbeda, secara umum ada tiga macam tujuan utama (Zakiyudin, 2011:1), yaitu:

1. Untuk mendukung fungsi kepengurusan manajemen
2. Untuk mendukung pengambilan keputusan manajemen
3. Untuk mendukung kegiatan operasi perusahaan

Secara khusus, tujuan informasi bergantung pada kegiatan yang ditangani, namun kecenderungan penggunaan sistem informasi lebih ditujukan pada usaha menuju keunggulan kompetitif yaitu mampu bersaing dan mengungguli pesaing.

1. **Konsep Dasar Informasi**

Menurut Gordon B. Davis dalam Rochaety dkk (2011:5) “Informasi adalah data yang telah diproses kedalam suatu bentuk yang mempunyai arti bagi penerima dan memiliki nilai nyata yang dibutuhkan untuk proses pengambilan keputusan saat ini maupun saat mendatang”. Ada tiga hal yang menentukan kualitas informasi yaitu:

1. Akurat

Artinya informasi harus bebas dari kesalahan-kesalahan juga harus mencerminkan maksudnya.

1. Tepat Waktu

Artinya informasi yang datang pada penerima tidak boleh terlambat karena informasi merupakan landasan didalam mengambil keputusan.

1. Relevan

Artinya informasi tersebut harus mempunyai manfaat bagi penggunannya.

Nilai suatu informasi juga ditentukan oleh dua hal yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai jika manfaatnya lebih efektif dibanding dengan biaya untuk mendapatkannya.

1. **Konsep Dasar Sistem Informasi**

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, dan bersifat manajerial dan kegiatan strategi dari suatu organisasi dan pihak luar tertentu dengan laporan yang diperlukan (Zakiyudin, 2011:9).

Sistem Informasi permintaan anggaran berbasis *web* pada Sekolah Pah Tsung Jakarta merupakan suatu program usulan yang berfungsi mengelola transaksi permohonan dan pertanggungjawaban anggaran yang diminta oleh tiap unit sekolah sehingga mempermudah dalam penyusunan laporan pengurangan anggaran yang sudah disusun tiap tahun ajaran baru, yang sebelumnya masih dilakukan secara manual dalam melakukan pencatatan proses transaksi hingga laporan sehingga waktu yang terpakai lebih efektif dan efesien.

1. **Konsep Dasar *Web***
2. **Pengertian *Web***

*Web* merupakan layanan *internet* paling terkenal sehingga banyak orang yang beranggapan bahwa *web* itulah yang disebut *internet*. Padahal *web* hanya salah satu bagian dari *internet*, seperti halnya *e-mail*. *Internet* lebih dulu ada sebelum *web*, tapi *internet* menjadi sangat terkenal setelah adanya *World Wide Web* (yang biasa disingkat WWW).

Menurut Wahana (2009:2) “*Web* merupakan media informasi berbasis jaringan komputer yang dapat diakses di mana saja dengan biaya relatif murah. *Web* merupakan bentuk implementasi dari bahasa pemrograman *web* (*web programming*)”.

Sedangkan menurut Yuhefizar (2013:2) pengertian *website* adalah “keseluruhan halaman-halaman *web* yang terdapat dari sebuah domain yang mengandung informasi”. Sebuah *website* biasanya dibangun atas banyak halaman web yang saling berhubungan. Hubungan antara satu halaman *web* dengan halaman *web* yang lainnya disebut dengan *Hyperlink*,sedangkan teks yang dijadikan media penghubung disebut dengan *Hypertext.*

1. ***Personal Home Pages* (PHP)**

Pengertian PHP menurut Anhar (2010:23) “PHP adalah (*PHP Hypertext Preprocessor*) adalah bahasa pemrograman *web* berupa *script* yang dapat diintegrasikan dengan HTML”.

Sedangkan menurut Wahana (2009:3) menerangkan bahwa “PHP (*PHP Hypertext Preprocessor)* merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memroses dan mengolah data secara dinamis”. PHP dapat dikatakan sebagai sebuah *server-side embedded script language,* artinya semua sintak dan perintah program yang anda tulis akan sepenuhnya dijalankan oleh *server*, tetapi dapat disertakan pada halaman HTML biasa.

Pada buku yang lain menurut Winarno dan Utomo (2010:137) “PHP merupakan bahasa pemrograman *web* yang apabila dilihat dari cara kerjanya merupakan bahasa pemrograman *web* yang diolah di sisi *server* atau sering disebut dengan *server side scripting*”.

Kode PHP diawali dangan sintaks “<?php**”** dan diakhiri dengan sintaks “?>”. PHP pertama kali diciptakan oleh Rasmus Lerford pada tahun 1994 ketika dia membuat sejumlah *script* Perl untuk mengetahui siapa saja yang melihat daftar riwayat hidupnya. *Script* ini kemudian dikenal sebagai “*Personal Home Page*”. Dari paket *script* inilah kemudian lahir PHP.

Pada tahun 1995, Rasmus membuat PHP/F1 versi 2 dan pada versi inilah kode dapat dimasukkan kedalam *tag* HTML dan dapat berkomunikasi dengan *Database.* Saat ini singkatan PHP menjadi *PHP Hypertext Preprocessor.* Sebuah singkatan yang menjadi ciri khas program GNU.

Tahun 1995 dianggap sebagai tahun kelahiran PHP, dan di tahun ini PHP banyak digunakan orang. Lalu pada tahun 1997, terjadi perubahan struktur organisasi pengembang PHP. Pengembangan tidak lagi dilakukan oleh Rasmus saja, tetapi juga dikembangkan oleh Zeev Suraski dan Andi Gutmans.

1. **HTML (*Hyper Text Markup Language)***

Menurut Winarno dan Utomo (2010:66) “HTML singkatan dari *Hypertext Markup Language* dan berguna untuk menampilkan halaman *web*”. Kode HTML ini digunakan sebagai bahan untuk melakukan *rendering* halaman *web.* Karena berbasis teks murni (*plain text*) sehingga ukurannya kecil dan tidak memboroskan *bandwidth* apabila ditransfer melalui jaringan internet. Dalam pemrograman HTML dikenal adanya istilah *Tag. Tag* adalah sintak dari HTML yang ditulis diantara dua tanda lebih kecil dan lebih besar “< >”.

1. **CSS *(Cascading Style Sheet)***

Menurut Winarno dan Utomo (2010:106) menerangkan bahwa “CSS merupakan bahasa pemrograman *web* yang digunakan untuk mengatur *style-style* yang ada di *tag-tag* HTML”.

Menurut Saputra dan Agustin (2013:6) CSS atau yang memiliki kepanjangan *Cascading Style Sheet* merupakan “suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur dan seragam”. CSS merupakan pemrograman wajib yang harus dikuasai oleh setiap pembuat *web* (*web programmer*), terlebih lagi itu adalah pendesain *web* (*web designer*).

CSS lintas platform, maksudnya dapat dijalankan pada berbagai macam sistem operasi dan *web browser*. Secara umum, yang dilakukan CSS adalah pengaturan layout, kerangka, teks, gambar, warna, tabel, spasi, dan lain sebagainya.

Penggunaan CSS pada suatu *web* memang tidak dijadikan suatu keharusan, namun penggunaan CSS pada suatu *web* akan mempunyai kelebihan tersendiri terutama dalam proses perbaikan (*maintenance*) *situs* akan lebih mudah*.*

1. **Javascript**

Menurut Sidik (2011:1) menjelaskan bahwa “JavaScript adalah bahasa pemrograman yang digunakan untuk membuat program yang digunakan agar dokumen HTML yang ditampilkan dalam *browser* menjadi lebih interaktif, tidak sekedar indah saja”. JavaScript memberikan beberapa fungsionalitas ke dalam halaman *web,* sehingga dapat menjadi sebuah program yang disajikan dengan menggunakan antarmuka *web*.

JavaScript merupakan bahasa *script,* bahasa yang tidak memerlukan kompiler untuk menjalankannya, cukup dengan *interpreter*. Tidak perlu ada proses kompilasi terlebih dahulu agar program dapat dijalankan. *Browser web Netscape Navigator* dan *Internet Explorer* adalah salah satu *interpreter,* karena kedua *browser* ini telah dilengkapi dengan *interpreter* JavaScript.

1. **Pengertian Basis Data (*Database*)**

Pengertian *Database* menurut Winarno dan Utomo (2010:142) “*Database* atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan *field*/kolom pada tiap tabel yang ada”.

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data (seperti tabel, indeks, dan lain-lain). Disamping berisi data, setiap basis data juga menyimpan definisi struktur (baik untuk basis data maupun objek-objeknya secara rinci).

Secara fisik, pengelolaan basis data tidak dilakukan oleh pemakai secara langsung. Tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak inilah yang disebut dengan DBMS (*Database Management System*) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali.

Perangkat lunak yang termasuk DBMS diantaranya dBase, FoxBase, Ms. Access, My SQL Server, Oracle Database, MySQL dan lain-lain.

1. **MySQL**

Menurut Raharjo (2011:5) ”MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*”. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware.*

MySQL memiliki beberapa kelebihan dan keuntungan dibanding database lain di antaranya adalah :

1. Banyak ahli berpendapat MySQL merupakan server tercepat.
2. MySQL merupakan *system* manajemen database yang *Open Source* (kode sumbernya terbuka), yaitu *software* ini bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
3. MySQL mempunyai performa yang tinggi tapi simpel*.*
4. Database MySQL mengerti bahasa SQL (*Structured Query Language*).
5. MySQL dapat diakses melalui protocol ODBC (*Open Database Connectivity*) buatan *Microsoft.* Ini menyebabkan MySQL dapat diakses oleh banyak *software.*
6. Semua klien dapat mengakses *server* dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses database.
7. Database MySQL dapat diakses dari semua tempat di *internet* dengan hak akses tertentu.
8. MySQL merupakan database yang mampu menyimpan data berkapasitas besar, sampai berukuran *Gigabyte*.
9. MySQL dapat berjalan di berbagai *operating system,* seperti *Linux, Windows, Solaris,* dan lain-lain.
	1. ***Tools Program***
10. **Struktur Navigasi**

Menurut Othman dkk (2006:51), Proses menghasilkan satu dokumen *web* seperti membangun modul-modul teks dan menghubungkan antara keduanya, pengembang *web* mungkin membangun tabel dan gambar serta mengaitkan keduanya ke teks. untuk memungkinkan akses informasi *web* yang lebih efektif pengembang *web* perlu menciptakan menu dan petunjuk sehingga pembaca mudah mengakses dokumen tersebut.

 Pengembangan situs *web* melibatkan aplikasi sistem *hypertext* dalam sesuatu sistem *hypertext*, satu informasi mungkin dapat dihubungkan dengan satu informasi lain. Ini memungkinkan ide dikaitkan dengan cara-cara yang menguntungkan. meskipun demikian, seseorang penulis yang tidak memiliki keterampilan akan tertarik untuk menciptakan hubungan yang teratur menyebabkan pembaca akan tersesat. Untuk menciptakan dokumen *web* yang berbasis *hypertext* yang berkualitas, satu struktur yang jelas harus ditempatkan ke informasi dan membuat struktur itu nyata kepada pengguna. Bentuk struktur yang lazim digunakan adalah hirarki.

1. Struktur Hirarki

 Menurut Othman dkk (2006:52), Menu utama adalah tingkat yang tertinggi dan dipecah menjadi sub halaman, berikut sub situs dipecahkan lagi dan tingkat ini merupakan spesialisasi yang paling mendalam, struktur hierarkhis harus memiliki konten pada setiap tingkat karena kategori yang kosong mewakili nilai kosong kepada pengguna. kebanyakan struktur ini memiliki kedalaman kurang dari pada empat tingkat kategori dan setiap cabang dalam struktur harus memiliki setidaknya dua kategori. selain itu setiap kategori harus memiliki konten yang cukup untuk mempertahankan eksistensinya.



Sumber : Othman dkk (2006:52)

**Gambar II.1. Struktur Hirarki**

1. Struktur Linier

 Menurut Othman dkk (2006:53), “Struktur linear adalah menyerupai cara dokumen cetak diorganisasikan”.

 Melalui struktur linear, setiap muka akan turut-menuruti antara satu sama lain seperti yang dirujukan dalam Gambar. *link* pada halaman menghubungkan halaman tersebut dengan cara yang sama sebagaimana halaman disusun. struktur linear adalah mudah untuk dibuat. struktur ini tidak memerlukan desain antar muka yang kompleks dan pengujian kegunaan yang banyak. implementasi struktur horizontal untuk struktur informasi yang dapat dilihat dengan jelas biasanya memiliki 30 kategori atau kurang. struktur ini memudahkan penjelajahan dan pemahaman konteks karena pengguna tidak diberi banyak pilihan untuk menjelajah ke dokumen yang lain. Namun, struktur yang tergerakan membatasi penjelajahan serta presentasi ide penulis *web* itu sendiri.



Sumber : Othman dkk (2006:53)

**Gambar II.2. Struktur Linier**

1. Kombinasi Struktur Linier dan Hirarki

 Menurut Othman dkk (2006:54), Satu metode organisasi isi dokumen *web* yang sering digunakan adalah kombinasi struktur linear dan hirarkis. organisasi seperti ini sering terlihat pada dokumen yang terstruktur tetapi bersifat linear, seperti segmen pertanyaan lazim pada kebanyakan situs. kombinasi ini efektif jika pembaca diberikan penanda posisi. pembaca mudah hilang arah ketika bergerak dari satu hirarki ke hirarki yang lain.

 Diagram menunjukan struktur linear dan hierarki yang banyak digunakan dalam situs hari ini. struktur ini memungkinkan pengguna mengakses situs tanpa harus mengikuti urutan tertentu. pengguna bebas untuk menjelajahi antara hirarki situs dan ini memudahkan pergerakan pengguna yang biasa melawar situs yang menggunakan struktur ini.



 Sumber : Othman dkk (2006:54)

**Gambar II.3. Struktur Linier dan Hirarki**

1. Struktur Rangkaian

 Menurut Othman dkk (2006:55), Struktur jaringan atau yang juga dikenal sebagai *web* adalah satu set dokumen yang memiliki sedikit atau tidak ada struktur keseluruhan sebenarnya. apa yang mengaitkan sesuatu situs dengan yang lain adalah kaitan seperti yang ditunjukan dalam gambar. *hypertext* harus dibangun dengan hirarki-hirarki yang dapat dilihat dengan jelas dan hubungan-hubungan yang teratur. ada banyak hubungan antara kategori atau node dan setiap hubungan yang ada dapat memiliki makna yang berbeda. setiap kategori dalam struktur jaringan ini dapat dihubungkan ke setiap kategori lain. oleh karena itu, satu kategori mungkin memiliki lebih dari satu kategori yang lebih tinggi tahapnya.

 Tesaurus, peta konsep dan jaringan semantik merupakan contoh struktur jaringan yang dapat diimplementasikan secara jelas atau tidak signifikan.

****

Sumber : Othman dkk (2006:55)

**Gambar II.4. Struktur Rangkaian**

1. Struktur Bersudut

 Menurut Othman dkk (2006:56), Struktur miring mencerminkan struktur horizontal ketika diimplementasikan tetapi memiliki struktur dan tujuan yang berbeda. Struktur miring juga tidak memiliki hubungan alami antara kategorinya seperti struktur horisontal. gambar 2.16 menunjukan struktur miring diwakili sebagai struktur bintang dengan setiap node dipermulaan struktur dihubungkan ke fokus di tengah. Struktur ini terlihat mudah tetapi menjadi kompleks dengan cepat

 Menurut Othman dkk (2006:56), isi desain yang paling penting untuk struktur bersudut ialah penyesuaian struktur dengan tujuannya. pengguna harus berkemampuan untuk memanipulasi node dengan jelas. Oleh itu, karakteristik setiap node harus didefinisikan dengan lengkap dan jelas tanpa tumpang tindih dengan node yang lain. Kebanyakan struktur miring adalah struktur yang tidak signifikan. Sekitarnya dijadikan signifikan, struktur ini akan diwakilkan dalam bentuk catatan atau format tabel .

 Menurut Othman dkk (2006:57), Aplikasi utama struktur miring yang tidak signifikan pada hari ialah catatan metadata perpustakaan, sistem e-perniaga dan portal merupakan pengguna utama metadata.



Sumber : Othman dkk (2006:56)

**Gambar II.5. Struktur Bersudut**

1. ***Unified Modeling Language* (UML)**

Menurut Nugroho (2010:6), UML (Unified Modeling Language) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek”. Pemodelan (modeling) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut Widodo (2011:10), “Beberapa literature menyebutkan bahwa UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung, misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi”. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain:

1. Diagram kelas *(Class Diagram)*

Bersifat statis, Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

1. Diagram paket *(Package Diagram)*

Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.

1. Diagram use-case *(Usecase Diagram)*

Bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.

1. Diagram interaksi dan sequence *(Sequence Diagram)*

Bersifat dinamis. Diagram urutan adalah iterasiksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

1. Diagram komunikasi *(Communication Diagram)*

Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan

1. Diagram statechart *(Statechart Diagram)*

Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas.

1. Diagram aktivitas *(Activity Diagram)*

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

1. Diagram komponen *(Component Diagram)*

Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

1. Diagram deployment *(deployment diagram)*

Bersifat statis. Diagram inimemperlihatkan konfigurasi saat aplikasi dijalankan (run-time). Memuat simpul-simpul beserta komponen-komponen yang di dalamnya. Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai kebutuhan. Pada UML dimungkinkan kita menggunakan diagram-diagram lainnya misalnya data flow diagram, entity relationship diagram, dan sebagainya.

1. ***Entity Relationship Diagram* (ERD)**

Menurut Rosa dan Shalahudin (2013:50) “Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD)”. ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*. Berikut adalah contoh bentuk hubungan relasi dalam ERD

1. *Binary*

Tingkat hubungan satu kesatu ,dinyatakan dengan satu kejadian pada entitas pertama.



Sumber: Rosa dan Shalahudin (2013:52)

**Gambar II.6. Binary**

1. *Ternary*

Tingkat hubungan satu kebanyak adalah sama dengan banyak kesatu tergantung dari arah mana hubungan tersebut dilihat.



Sumber: Sumber: Rosa dan Shalahudin (2013:52)

**Gambar II.7. Ternary**

1. *N-ary*

Tingkat hubungan banyak kebanyak terjadi tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Relasi antar keduanya diwakili dengan tanda panah ganda disetiap sisinya untuk menunjukan hubungan banyak tersebut.



Sumber: Rosa dan Shalahudin (2013:52)

**Gambar II.8. N-ary**

1. ***Entity Relationship Diagram* (ERD)**

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol.

Menurut Brady dan Loonam (2010), Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analys dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERDdigunakan digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

Menurut Simarmata (2007:107) menyatakan bahwa “derajat relasi adalah sejumlah entitas yang berhubungan dengan hubungan. Hubungan n-ary adalah format yang umum untuk derajat n. Kasus khusus adalah seperti biner dan ternary, dimana derajatnya adalah 2 dan 3 secara berturut-turut.

Menurut Fathansyah (2007:77) menyatakan bahwa “Kardinalitas Relasi menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain”. Konektivitas dari suatu hubungan menguraikan pemetaan dari kejadian entitas yang dihubungkan. Nilai-nilai konektivitas adalah “satu” atau “banyak”. Kardinalitas suatu hubungan menyatakan sejumlah kejadian terkait untuk masing-masing dua entitas. Jenis dasar konektivitas untuk hubungan adalah satu-kesatu, satu-ke-banyak, dan banyak-ke-banyak.

1. ***Logical Record Structure* (LRS)**

Menurut Janner (2007:115) “ *Logical Structure Record* (LRS) adalah representasi dari struktur *record - record* pada tabel – tabel yang terbentuk dari hasil antar himpunan entitas”.

LRS dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dengan nama yang unik. LRS juga terdiri dari hubungan diantara tipe *record*. dua metode lain di mulai dengan ER dan dikonversi ke LRS.

1. ***BlackBox Testing***

Rosa dan Shalahudin (2013:275) “*Black Box* *Testing* bisa disebut juga pengujian kotak hitam, yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan.

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk proses kasus login maka kasus uji yang dibuat adalah :

1. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar
2. Jika user memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah atau sebaliknya atau keduanya.
3. ***Model Waterfall***

Menurut Rosa dan Shalahuddin (2013:28) “Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*Support)*”.

Langkah-langkah yang penting dalam model ini sebagai berikut:

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dubutuhkan oleh *user.*

1. Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosuder pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahapselanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

1. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai desain yang telah dibuat pada tahap desain.

1. Pengujian

Pengujian focus pada perangkat lunak secara dari segi lojik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisirkesalahan (*error)* dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

1. Pendukung (*support)* atau pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahn yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

 Sumber: Rosa dan Salahudin (2013:29)

**Gambar II.9. Ilustrasi model *waterfall***

1. **Adobe Dreamweaver CS6**

Menurut Madcoms (2013:2) “Adobe Dreamweaver CS6 adalah versi terbaru dari Adobe Dreamweaver yang merupakan bagian dari Adobe Creative Suite 6”. Dreamweaver sendiri merupakan aplikasi yang digunakan sebagai HTML editor profesional untuk mendesain *web* secara visual. Aplikasi ini juga biasa dikenal dengan istilah WYSIWYG (*What You Say Is WhatYou Get*). Yang intinya adalah anda tidak harus berurusan dengan *tag-tag* HTML untuk membuat sebuah *site* dan dapat melihat hasil desainnya secara langsung.

Dengan kemampuan fasilitas yang optimal dalam jendela design akan memberikan kemudahan untuk mendesain *web* meskipun untuk para *web* *desainer* pemula sekalipun. Kemampuan Adobe Dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, JavaScript, dan yan lainnya juga memberikan fasilitas maksimal kepada desainer *web* dengan menyertakan bahasa pemrograman di dalamnya.

Adapun tampilan awal Adobe Dreamweaver CS6 adalah sebagai berikut:



Sumber: Madcoms (2013:2)

**Gambar II.10.Tampilan awal Adobe Dreamweaver CS6**

1. **XAMPP**

Menurut Wahana (2009:30) “XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”. XAMPP bersifat *freeware*, sehingga banyak programmer yang bermigrasi ke PHP dari bahasa pemrograman bebayar. Pada XAMPP logika pemrograman lebih mudah karena hampir sama dengan bahasa pemrograman pada umumnya. Selain itu, XAMPP cukup mudah diinstal dan digunakan yakni hanya dengan mendownload dan mengekstaknya sebelum dapat digunakan.

* 1. **Penelitian Terkait**

Dalam penyusunan skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian - penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Adapun penelitian yang berhubungan dengan skripsi ini antara lain yaitu :

Menurut pendapat Mulyadi (2002) dalam jurnal Sundjaja dkk, anggaran merupakan suatu rencana kerja yang dinyatakan secara kuantitatif, yang diukur dalam satuan moneter standar dan satuan ukuran lain, yang mencakup jangka waktu satu tahun. Karakteristik anggaran yang baik adalah anggaran disusun berdasarkan program, anggaran yang disusun berdasarkan karakteristik pusat pertanggungjawaban yang dibentuk dalam organisasi institusi, dan anggaran yang berfungsi sebagai alat perencanaan dan pengendalian.

Penelitian yang dilakukan oleh Raeni (2014:143) Produktivitas merupakan komponen penting dalam kualitas pendidikan SMK. Pengelolaan keuangan merupakan aktualisasi program kerja sekolah untuk mencapai produktivitas. Pengelolaan ini berlandaskan prinsip pengelolaan dana pendidikan.