

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

A. Konsep dasar sistem informasi

Suatu sistem pada dasarnya merupakan sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu. Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Dan informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi = *input – proses – output*.

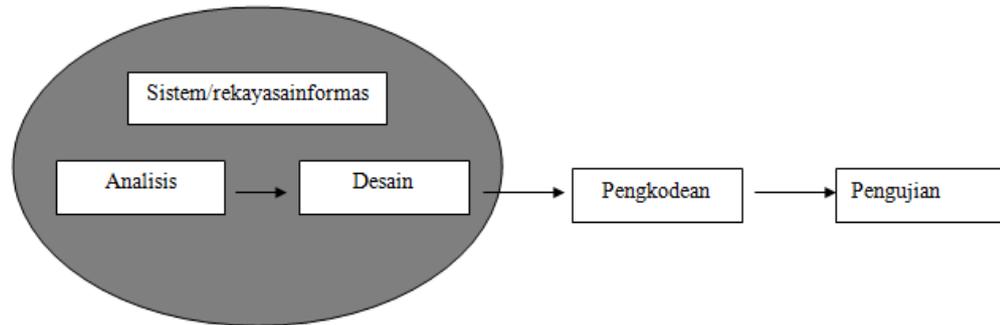
Menurut Kristanto (2008:13) “Sistem informasi ada sekumpulan dari prosedur organisasi yang pada saat dilaksanakan akan memberi informasi bagi pengambil keputusan dan untuk mengendalikan organisasi. “

Sistem informasi dapat berupa kombinasi teratur apapun dari orang – orang , *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi.

Sistem fisik (*Physical system*) perusahaan terdiri atas sumber – sumber daya berwujud seperti bahan baku, karyawan, mesin dan uang. Sedangkan sistem *virtual* (*virtual system*) terdiri atas sumber daya informasi yang digunakan untuk mewakili sistem fisik. Orang bergantung pada sistem informasi, untuk berkomunikasi antara satu sama lain dengan menggunakan berbagai jenis alat fisik (*hardware*), perintah dan prosedur pemrosesan informasi (*software*), saluran

komunikasi (jaringan) dan data yang disimpan (sumber daya data). Model pengembangan sistem yang digunakan dalam pembuatan strategi *web e-commerce* ini adalah model SDLC air terjun (*waterfall*).

Berikut adalah gambar model :



Sumber dan Sahalahuddin (2011:29)

Gambar II.1

Model *Waterfall*

Tahapan-tahapan perancangan sebuah *website e-commerce* secara umum adalah sebagai berikut :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memesifikasi kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apakah yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antar muka dan prosedur pengodean.

3. Pembuatan Kode Program

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

4. Pengujian

Dari Proses logika yang sudah dibuat, diperiksa apakah program tersebut sudah benar dan bebas dari kesalahan atau masih harus diperbaiki kembali. Semua kesalahan yang terjadi diperbaiki agar program komputer dapat dijalankan dan memberi hasil sesuai yang diharapkan.

5. Pendukung atau Pemeliharaan

Pemeliharaan digunakan untuk menjabarkan aktivitas dari analisis sistem pada saat perangkat lunak telah dipergunakan oleh pemakai. Berdasarkan pengertian diatas dapat dirinci lebih lanjut mengenai sistem secara umum yaitu :

1. Komponen atau lemen yang dapat dilihat, didengar dan dirasakan.
2. Proses atau kegiatan untuk mengkoordinasikan komponen yang telah terlihat dalam sebuah sistem.
3. Tujuan mengenai sasaran akhir yang ingin dicapai dari kegiatan koordinasi komponen tersebut.

B. E-commerce

“Menurut Ahmadi dan Hermawan (2013:7) Menyatakan bahwa *E-Commerce* atau *Electronic Commerce* adalah penjualan atau pembelian barang dan jasa antara perusahaan, rumah tangga, individu, pemerintahan dan mastarakan atau organisasi swasta lainnya, yang dilakukan melalui komputer pada media jaringan”.

Barang-barang dan jasa dipesan melalui jaringan tersebut, tetapi pembayaran dan pengiriman barang atau jasa dapat dilakukan diakhir atau *offline*. *Electronic commerce (e-commerce)* atau perdagangan lewat elektronik adalah perdagangan yang dilakukan dengan memanfaatkan jaringan telekomunikasi terutama *internet*. Internet memungkinkan orang atau organisasi yang berada pada jarak yang jauh dapat saling berkomunikasi dengan biaya yang murah. Hal ini kemudian dimanfaatkan untuk melakukan transaksi perdagangan.

C. UML (Unified Modeling Language)

Menurut Rosa dan Shalahuddin (2013:133) “ *UML (Unified Modelling Language)*” adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek. “

Menurut Chonoles dalam Widodo dan Herlawati (2011:6) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik UML diaplikasikan untuk maksud tentu biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.

3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Adapun diagram-diagram yang ada didalam UML, adalah sebagai berikut.

1. *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2013:155) “Yang dimaksud dengan use case adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. “

Menurut pilone dalam widodo (2011:21) “Use Case diagram menggambarkan fungsi tertentu dalam suatu sistem berupa komponen, kejadian atau kelas”. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. sebuah *Use Case* mempresentasikan sebuah interaksi antara aktor dengan sistem. Komponen pembentukan diagram *Use Case* adalah:

- a. Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- b. *Use Case*, aktivitas atau sarana yang disiapkan oleh bisnis atau sistem.
- c. Hubungan (*link*), aktor mana saja yang terlibat dalam *use case* ini.

Beberapa cara menawarkan *use case* yang baik menurut Widodo dan Herlawati (2011:22) yaitu :

- a. **Pilihlah nama yang baik**

Use case adalah sebuah *behavior* (pelaku), jadi seharusnya dalam frase kata kerja. Untuk membuat namanya lebih detail. Tambahkan kata benda yang mengidikasikan dampak aksinya terhadap suatu kelas objek. Oleh karena itu diagram *Use case* seharusnya berhubungan dengan diagram kelas.

b. Ilustrasikan pelaku dengan lengkap

Use case dimulai dari inisiasi oleh aktor primer dan berakhir pake aktor dan menghasilkan tujuan. Jangan membuat *use case* kecuali anda mengetahui tujuannya.

c. Identifikasikan perilaku dengan lengkap

Untuk mencapai tujuan dan menghasilkan nilai tertentu dari aktor, *use case* harus lengkap. Ketika memberi nama pada *use case*, pilihlah rase kata kerja yang impikasinya hingga selesai. Misalnya gunakan frasa *reserve a room*(Pemesanan kamar) dan jangan *reserving a room* (memesan kamar) karena memesan menggambarkan pelaku yang belum selesai.

d. Menyediakan *use case* lawan (*inverse*)

Kita biasanya membutuhkan *use case* yang membatalkan tujuan, misalnya pada *use case* pemesanan kamar, dibutuhkan pula *use case* pembatalan pemesanan kamar.

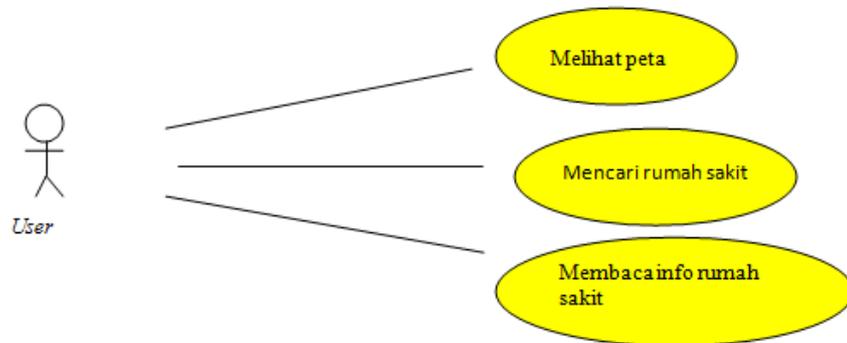
e. Batasi *use case* hingga satu perilaku saha

Kadang kita cenderung membuat *use case* yang menghasikan lebih dari sau tujuan aktivitas. Guna menghindari kerancuan, jagalah *use case* hanya focus pada satu hal. Misalnya penggunaan *use case check-in* dan *check out* dalam satu *use case* menghasilkan ketidakfokusan, karena memiliki dua perilaku yang berbeda

f. Nyatakan *use case* dari sudut pandang aktor

Tulislah *use case* dari sudut pandang aktor bukan dari sistem. Sebagai contoh, pilihlah nama *use case* pemesanan kamar, bukannya pencatatan pemesanan

kamar karena pemesanan kamar sudut pandangnya aktor tamu sedangkan pencatatan pemesanan hotel.



Sumber : Rosa dan Shalahudin (2013:155)

Gambar II.2

Contoh penggambaran *Use Case Diagram*

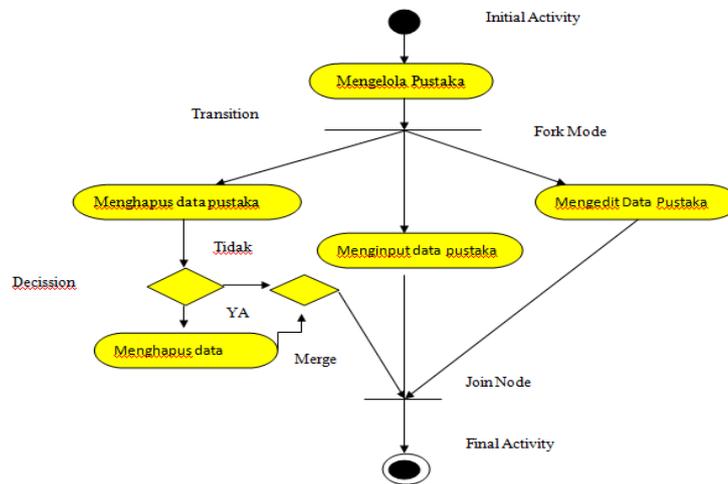
2. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2013:161) “Diagram aktivitas atau activity diagram menggambarkan *worklow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

Menurut Widodo dan Herlawati (2011:11)” Activity Diagram adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainya dalam sebuah sistem. “

Aktifitas merupakan kumpulan aksi-aksi. Aksi-aksi merupakan langkah sekali saja tidak boleh dipecah menjadi beberapa langkah. Contoh aksi menurut widodo dan Herlawati (2011:144) yaitu :

- a. Fungsi Matematika
- b. Pemanggilan Prilaku
- c. Pemrosesan Data



Sumber : Rosa dan Shalahudin (2013:161)

Gambar II.3

Contoh Penggambaran Activity Diagram

3. *Deployment Diagram*

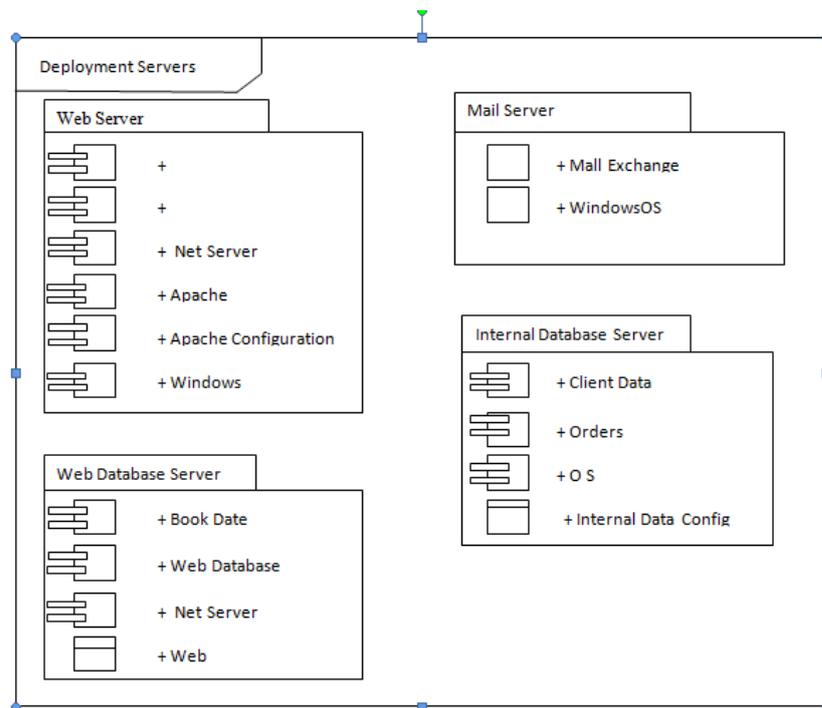
Menurut Rosi dan Shalahuddin (2013:1540) “*Diagram deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.”

Menurut Widodo dan Helawati (2011:12) Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Deployment diagram menyediakan gambaran bagaimana system secara fisik akan terlihat. Sistem terdiri dari *node-node* dimana *node* diwakili bentuk

sebuah kubus garis yang menghubungkan antara kedua kubus menunjukkan hubungan antara kedua *node* tersebut. Tipe *node* bisa berupa *device* yang terwujud *hardware* dan bisa juga *processor* (yang mengeksekusi *component*) atau *execution environment* (*software* yang menjadi *host* atau mengandung *software* lain).

Sumber : Rosa Shalahudin (2013 : 154)



Gambar II .4

Contoh Penggambaran *Deployment Diagram*

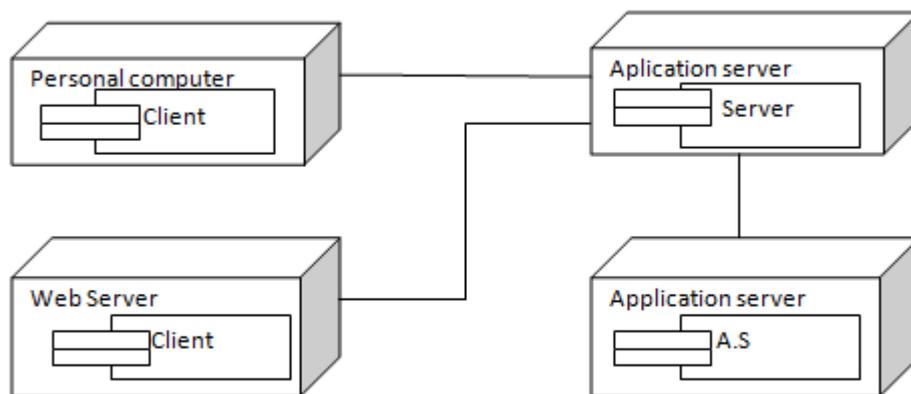
4. *Component Diagram*

Menurut Rosa dan Shalahudin (2013 : 148) “Diagram komponen adalah komponen yang dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem “.

Menurut Widodo dan Herlawati (2011:95) “*Component* merupakan subsistem berisi kelas-kelas yang bekerja sama merealisasikan antarmuka yang diinstruksikan”.

Manfaat diagram komponen adalah bila ada salah satu komponen yang rusak atau tidak sesuai dengan tujuan system, kita tinggal mengganti komponen tersebut dengan komponen yang lain. Bersifat statis, memperlihatkan organisasi serta kebergantungan system/perangkat lunak pada komponen yang telah ada sebelumnya dengan diagram kelas.

Sumber : Rosa dan Shalahudin (2013:148)



Gambar II.5

Contoh Penggambaran *Component Diagram*

D. Entity Relationship Diagram (ERD)

Menurut Priyadi (2014:20) menyatakan bahwa : Pemodelan basis data dengan menggunakan diagram relasi antara entitas, dapat dilakukan dengan menggunakan suatu pemodelan basis data yang bernama *Diagram Entity Relationship* yang disingkat Diagram E-R. ERD juga merupakan gambaran yang menghubungkan antara objek satu dengan objek yang lain dalam dunia nyata. Bisa

dikatakan bahwa bahan yang akan digunakan untuk membuat ERD adalah dari objek di dunia nyata. Secara umum ERD terdiri dari 4 komponen, yakni :

1. Entitas

Entitas merupakan notasi untuk mewakili suatu objek dengan karakteristik sama, yang dilengkapi oleh atribut, sehingga pada suatu lingkungan nyata objek akan berbeda dengan objek lainnya.



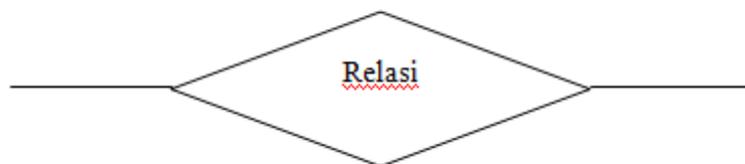
Sumber : Priyadi (2014:22)

Gambar II.6

Simbol Entitas

2. Relasi

Relasi merupakan notasi yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan.



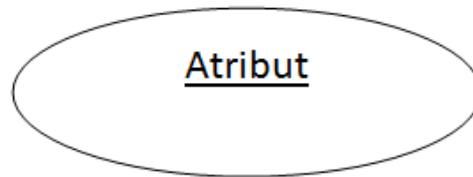
Sumber : Priyadi (2014:22)

Gambar II.7

Simbol Relasi

3. Atribut

Atribut merupakan notasi yang menjelaskan karakteristik suatu entitas dan juga relasinya. Atribut dapat sebagai key yang bersifat unik, yaitu *primary key* atau *foreign key*. Selain itu, atribut juga dapat sebagai atribut deskriptif saja, yaitu sebagai pelengkap deskripsi suatu entitas dan relasi.



Sumber : Priyadi (2014 : 22)

Gambar II.8

Simbol Atribut

4. Garis Penghubung

Garis penghubung merupakan notasi untuk merangkai keterkaitan antara notasi-notasi yang digunakan dalam Diagram E-R , yaitu entitas, Relasi , dan atribut.



Sumber Priyadi (2014:22)

Gambar II.9

Simbol Garis Penghubung

Setelah menjelaskan tentang definisi ERD serta komponen ERD , pada bagian ini penulis juga akan mendefinisikan tentang derajat *relationship* dan mendefinisikan tentang LRS.

1. Derajat *Relationship*

Menurut Ladjmudin (2006 : 142) “ Derajat relasi adalah jumlah entitas yang Berpartisipasi dalm suatu relasi “ . Derajat-derajat relasi yang umum dijumpai pada penggambaran diagram E-R adalah :

a. Unary Relationship

Relasi berderat 1 (*unary relationship*) adalah relasi dimana entitas yang terlibatnya hanya 1 (relasi yang berderajat 1 sering juga dinamakan relasi rekursif /*recursive relationship*)

b. Binary Relationship

Relasi berderajat 2, sering juga disebut sebagai relasi biner (*binary relationship*), adalah relasi yang melibatkan 2 entitas.

c. Ternary Relationship

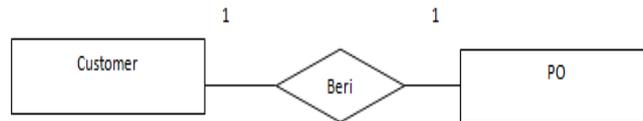
Ternary Relationship adalah relasi berderajat 3, yaitu relasi tunggal yang menghubungkan 3 entitas yang berbeda.

2. Tingkat Hubungan (*Cardinality Ratio*)

Tingkat hubungan yaitu menjelaskan hubungan jumlah keterhubungan satu entitas dengan entitas lainnya atau banyaknya entitas yang bersesuaian dengan yang lain melalui *relationship*. Ada 3 (tiga) kemungkinan hubungan yang ada yaitu :

a. Satu ke satu (*one to one* atau 1:1)

Tingkat hubungan dinyatakan satu ke satu jika suatu kejadian pada entitas pertama hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua. Demikian juga sebaliknya.



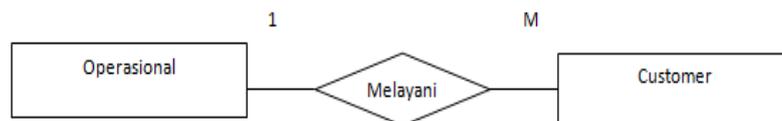
Sumber : Priyadi (2014:25)

Gambar II.10

Cardinality Ratio 1:1

b. Satu ke Banyak (*one to many* atau 1: M)

Tingkat hubungan satu ke banyak (satu : M) adalah sama dengan ke satu (M: satu), tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian entitas pertama dapat mempunyai hubungan dengan kejadian entitas kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya bisa mempunyai satu hubungan dengan satu kejadian entitas yang pertama.



Sumber : Priyadi (2014:25)

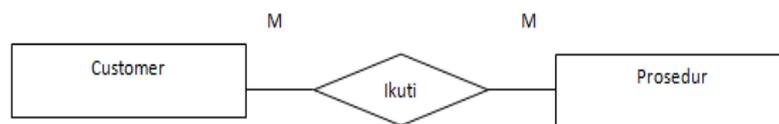
Gambar II.11

Cardinality Ratio 1:M

c. Banyak ke banyak (*many to many M : M*)

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi entitas yang kedua.

Sumber : Priyadi (2014:26)



Gambar II.12

Cardinality Ratio M:M

3. *Logical Record Structure (LRS)*

Menurut Simarmata (2008:12) “LRS adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas. Menentukan Kardinalitas, Jumlah Tabel, dan *Foreign Key (FK)*”. *Logical Record Structure* dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda dengan Diagram E-R nama tipe record berada diluar kotak *field* tipe *record* ditempatkan. *Logical Record Structure* terdiri dari *link-link* diantaranya tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* lainnya. Banyak link dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*. Penggambaran LRS mulai dengan menggunakan

model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat konversikan ke LRS. Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

E. PHP

Menurut Prasetio (2014:122) “PHP : *Hypertext Preprocessor* adalah bahasa *script* yang ditanam di sisi *server*”. Seperti sebagian besar bahasa *script* lainnya, PHP dapat ditanamkan langsung ke dalam HTML. Kode PHP dipisahkan dari HTML. Dengan menggunakan tanda start `<? Php` dan `End ?>`. Ketika sebuah dokumen dibaca, prosesor PHP hanya menterjemahkan area yang ditandai saja dan menampilkan hasilnya pada tempat yang sama.

PHP memiliki koneksitas yang baik dengan beberapa DBMS antara lain *Oracle, Sybase, SQL, MYSQL, Microsoft SQL Server, Solid, Postgre SQL, Adabas, Filepro, Velocis dBase, Unix dbm* dan tidak terkecuali semua database ber-*interface* ODBC. Hampir seluruh aplikasi berbasis *web* saat ini dibuat dengan PHP.

F. MySQL (My Structure Query Language)

Menurut Sibero (2013:97) :MySQL adalah suatu RDBMS (*Relational Database Management System*) Yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database management System*) atau DBMS yang *multithread, multi-user* dengan sekitar 6 juta instalasi diseluruh dunia.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Struktur Query Language*). SQL digunakan

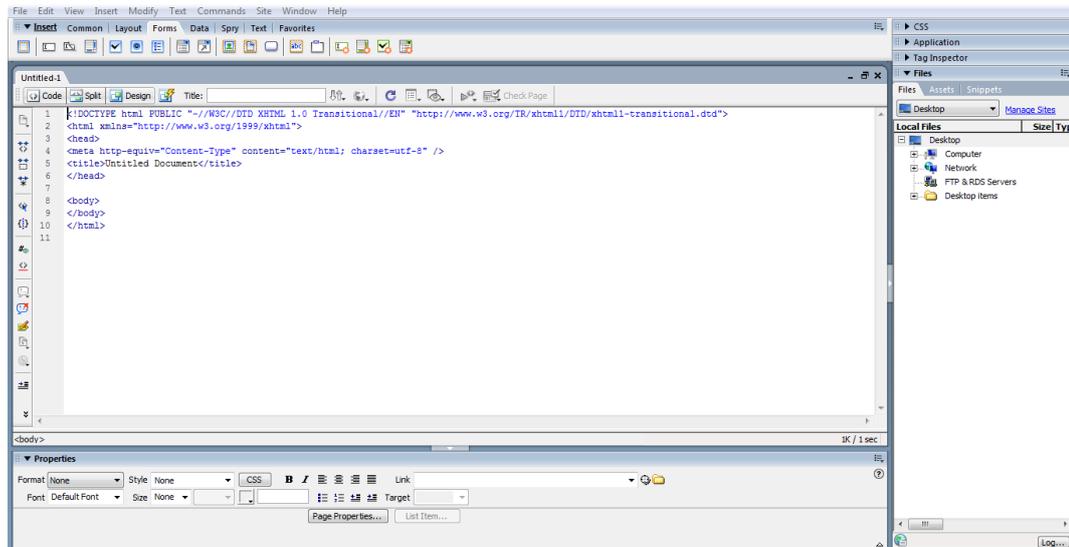
untuk berkomunikasi dengan sebuah database. SQL merupakan bahasa standar untuk sistem manajemen database relasional.

G. *Adobe Dreamweaver CS8*

Menurut Mack Dkk (2006:24) “*Dreamweaver has system that allows one to build a template for a page that has certain fixed, unchangeable regions and other areas designed to be edited*”. Aplikasi ini juga yang biasa dikenal dengan istilah WYSIWYG (*What You See IS What You Get*) yang intinya adalah bahwa anda tidak harus berurusan dengan tag-tag HTML, untuk membuat sebuah situs *web*.

Selain itu, Dreamweaver juga memberikan keluasaan kepada anda untuk menggunakannya sebagai media penulisan bahasa pemrograman *web*. Dengan kemampuan fasilitas yang optimal dalam jendela desain membuat program ini memberikan kemudahan mendesain web meskipun untuk pemula sekalipun. Sedangkan kemampuan dreamweaver untuk berinteraksi dengan beberapa bahasa pemrograman seperti PHP, ASP, *Javascript* dan yang lainnya juga memberikan fasilitas maksimal kepada para desainer web yang menyertakan bahasa pemrograman *web* didalamnya. *Adobe* kembali mengeluarkan varian terbaru dari *Dreamweaver* yaitu *Dreamweaver CS8*. Ada beberapa fitur baru yang dapat ditemukan pada versi terbaru dari *Dreamweaver CS8* diantaranya : Integrasi dengan *Adobe Business Catalyst*, integrasi *Adobe Browser Lab*, Integrasi dengan CMS, Pengecekan CS8 dan lain.lain

Sumber :Mack dkk (2006:24)



Gambar II.13

Contoh Lembar Kerja *Dreamweaver*

H. CSS (*Cascading Style Sheet*)

Menurut Prasetio (2014:252) “CSS adalah suatu teknologi yang digunakan untuk memperindah tampilan halaman *website* (situs)”. Metode CSS dapat dengan mudah mengubah secara keseluruhan sekaligus memformat ulang sebuah situs.

CSS juga memiliki beberapa properties yang dinamis, yang membuat suatu item bisa tampil atau tidak dan digunakan untuk membuat *drop-down* dan komponen interaktif lainnya. Selain itu, CSS bisa dibuat diluar halaman web dan dijalankan pada semua halaman web dalam waktu yang bersamaan.

2.2 Penelitian Terkait

Dalam penelitian terkait ini penulis mengambil dua jurnal sebagai berikut :

Menurut (Wibowo, 2014:1) ”seiring dengan perkembangan teknologi informasi yang semakin pesat dan semakin bertambahnya penggunaan sistem informasi, menuntut suatu perusahaan atau instansi untuk mengedepankan layanan informasi yang lebih cepat dan lebih efisien. Sehubungan dengan itu, manajemen usaha, juga telah mengalami perubahan dan peningkatan dalam sistem yang digunakan. Hal itu dilihat pada pengolahan data yang telah banyak menggunakan komputer. Toko Joice group proses penjualan, pembelian dan pembuatan laporan masih menggunakan metode konvensional sehingga kesulitan dalam pengontrolan persediaan barang, kesulitan dalam pembuatan laporan penjualan, pembelian dan membutuhkan waktu lama dalam pencarian data barang”.

Menurut (Hasanah, 2013:1) “Toko kreatif Suncom Pacitan merupakan salah satu toko yang menjual beraneka ragam merk komputer, hardware, software serta berbagai aksesoris yang berkaitan dengan komputer di kota pacitan. Di Toko Kreatif Suncom pacitan ini masih terjadi beberapa permasalahan. Hingga saat ini sistem penjualannya masih bersifat konvensional(Offline) dimana pembeli yang ingin melakukan transaksi pembelian masih harus datang ke toko. Dengan demikian pembeli membutuhkan biaya yang lebih besar dan waktu yang cukup lama untuk membeli ataupun sekedar ingin mengetahui tentang beberapa merk yang tersedia di Toko Kreatif Suncom Pacitan, selain itu pembeli sulit mendapatkan informasi tentang berbagai produk yang dijual di Toko Kreatif Suncom Pacitan”.