BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Dalam penyusunan ini dicantumkan beberapa teori umum yang dapat dijadikan sebagai landasan teori, berikut penjelasan dari beberapa teori, diantaranya adalah sebagai berikut :

2.1.1. Definisi Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu. (Hutahaean, 2015).

Sistem dapat diartikan sebagai serangkaian komponen-komponen yang saling berinteraksi dan bekerja sama untuk mencapai tujuan tertentu, dan didalam sistem terkandung tiga elemen penting, yaitu rangkaian komponen, interaksi dan kerja sama dan yang terakhir adalah tujuan. (Soeherman dan Pinontoan, 2008 *dalam* Saputra dan Bukhori, 2014).

Sistem adalah sebuah tatanan yang terdiri atas sejumlah komponen fungsional yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses / pekerjaan tertentu. (Kusrini, 2006 *dalam* Jayanti dan Iriani , 2014).

2.1.1.1. Karakteristik Sistem

Supaya sistem itu dikatakan sistem yang baik harus memiliki karakteristik seperti berikut : (Hutahaean, 2015).

1. Komponen

   Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*boundary*)

   Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*environment*)

   Environment adalah diluar batas dari sistem yang mempengaruhi sistem operasi. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem (*interface*)

   Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukkan (*input*) untuk subsistem lain melalui penghubung.

5. Masukkan Sistem (*input*)

   Masukkan adalah energy yang dimasukkan kedalam sistem yang dapat berupa perawatan (*maintenance input*) dan masukkan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*output*)

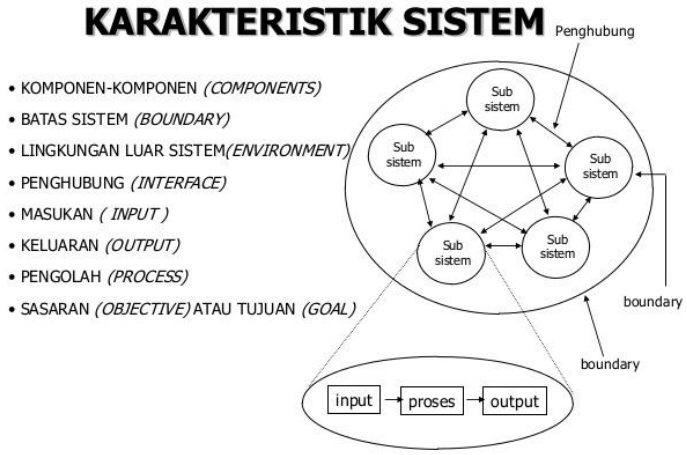
   Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

   Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan – laporan keuangan.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan *input* yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.



Gambar 2.1 Karakteristik dari Suatu Sistem

2.1.2. Definisi Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya. Sumber informasi adalah data. Data kenyataan yang menggambarkan suatu kejadian-kejadian (*event*) adalah kejadian yang terjadi pada saat tertentu. (Hutahaean, 2015).

Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Data belum memiliki nilai sedangkan informasi sudah memiliki nilai. Informasi dikatakan bernilai bila manfaatnya lebih besar dibandingkan biaya untuk mendapatkannya. (Abdullah, 2015) Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan. (Kadir, 2005 *dalam* Saputra dan Bukhori, 2014).

2.1.3. Definisi Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat managerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. (Jogiyanto, 1999 *dalam* Aminudin, dkk, 2015).

Sistem informasi dapat didefinisikan sebagai kumpulan elemen yang saling berhubungan, berkumpul bersama-sama dan membentuk satu kesatuan, saling berinteraksi dan bekerjasama antara satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, *input, processing, output* berupa informasi. (Sutanta, 2004 *dalam* Jayanti dan Iriani, 2014).

Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai. (Abdullah, 2015).

2.1.4. *Software Development Life Cycle* (SDLC)

SDLC atau *Software Development Life Cycle* adalah proses mengubah atau mengembangkan suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik). (Rosa dan Shalahuddin, 2013).

2.1.5. *Database*

Basis data atau *Database* merupakan koleksi dari data-data yang teroganisir dengan cara sedemikian rupa sehingga data tersebut mudah disimpan dan dimanipulasi. (Luqman, 2012 *dalam* Aminuddin, dkk, 2015).

Sistem basis data adalah suatu sistem menyusun dan mengelola record-record menggunakan komputer untuk menyimpan atau merekan serta memelihara data operasional lengkap sebuah organisasi atau perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses pengambilan keputusan. (Marlinda, 2004 *dalam* Jayanti dan Iriani, 2014).

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan, karakter, atau simbol). (Abdullah, 2015).

2.1.6. Pengertian Gaji

Gaji didefinisikan secara umum sebagai pembayaran atas penyerahan jasa yang dibayarkan kepada karyawan yang memiliki jenjang jabatan manajer dan umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (bagian produksi) dan dibayarkan berdasarkan hari kerja, jam kerja, atau jumlah satuan produk yang telah dihasilkan oleh karyawan. Dalam pelaksanaan sistem penggajian di tiap-tiap perusahaan tidak selalu sama, hal tersebut tergantung dari kondisi perusahaan. (Mulyadi, 2001 *dalam* Agustina dan Sukadi, 2014).

Gaji merupakan sejumlah pembayaran kepada pegawai yang diberi tugas administratif dan manajemen yang biasanya ditetapkan secara bulanan, sedangkan upah merupakan imbalan yang diberikan kepada buruh yang melakukan pekerjaan kasar dan lebih banyak mengandalkan kekuatan fisik, jumlah pembayaran upah biasanya ditetapkan secara harian atau berdasarkan unit pekerjaan yang diselesaikan. (Sugiyarso dan Winarni, 2005 *dalam* Jayanti dan Iriani, 2014).

Istilah gaji, upah, dan imbalan merupakan istilah yang sering kita temukan dalam hubungan bisnis antara karyawan dan perusahaan. (Nilasari, 2016).

1. Gaji

   Gaji atau yang dalam bahasa inggris dikenal dengan istilah *salary* merupakan sebuah bentuk pembayaran secara periodik dari sebuah perusahaan kepada seorang karyawan yang mungkin didasarkan pada kontrak kerja.

2. Upah

   Upah atau dalam bahasa inggris disebut dengan *wage* merupakan kompensasi berupa uang yang dibayarkan oleh perusahaan kepada pekerjanya sebagai balasan dari pekerjaan yang telah dilakukan.

3. Imbalan

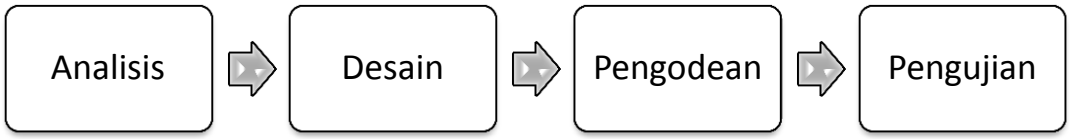
   Dalam bahasa inggris, kata "imbalan" bisa merujuk pada kata compensation. Imbalan merupakan istilah yang cakupannya lebih luas dari gaji dan upah. Menurut Achmad S. Ruky (2006), imbalan ini mencakup semua pengeluaran yang dikeluarkan oleh perusahaan untuk pekerja dan diterima atau dinikmati oleh pekerja, baik diterima langsung, rutin maupun tidak langsung.

2.2. Tinjauan Teori Khusus

Dalam penyusunan ini dicantumkan beberapa teori khusus yang menjelaskan tentang proses pembuatan sistem informasi penggajian karyawan, berikut penjelasan dari beberapa teori, diantaranya adalah sebagai berikut :

2.2.1. *Waterfall Model*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential liniear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*). Berikut adalah gambar model air terjun : (Rosa dan Shalahuddin, 2013)



Gambar 2.2 Ilustrasi Model *Waterfall*

1. Analisa kebutuhan perangkat lunak

   Proses penginputan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk dilakukan dokumentasi.

2. Desain

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

3. Pembuatan kode program

   Desain harus diimplementasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

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

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

   Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan 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.

2.2.2. Diagram Konteks

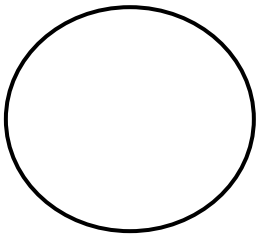
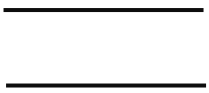
Diagram konteks memaparkan diagram konteks berisi gambaran umum (secara garis besar) sistem akan dibuat. Secara kalimat, dapat dikatakan bahwa diagram konteks ini berisi "siapa saja yang memberi data (dan data apa saja) ke sistem, serta kepada siapa saja informasi (dan informasi apa saja) yang harus dihasilkan". (Abdullah, 2015).

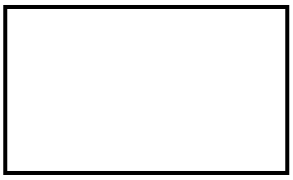
2.2.3. *Data Flow Diagram* (DFD)

Data Flow Diagram (DFD) merupakan alat pemodelan dari proses analisis kebutuhan perangkat lunak. Dalam DFD dibahas fungsi-fungsi apa saja yang diperlukan oleh suatu sistem dan aliran data yang terdapat diantara proses didalamnya. DFD berguna sebagai alat untuk memverifikasi apakah sistem yang akan dibangun sudah memenuhi kriteria yang diinginkan oleh *user* atau belum. (Abdullah, 2015).

Data flow diagram (DFD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco) adalah sebagai berikut : (Rosa dan Shalahuddin, 2013).

Tabel 2.1 Notasi pada DFD

Notasi	Keterangan
	Proses atau fungsi atau prosedur pada pemodelan perangkat lunak yang akan di implementasikan dengan pemrograman terstruktur maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program. Catatan : Nama yang diberikan pada sebuah proses biasanya berupa kata kerja
	*File* atau basis data atau penyimpanan (*storage*) pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel basis data yang dibutuhkan tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (*Entity Relationship Diagram* (ERD), *Conceptual Data Model* (CDM), *Phisycal Data Model* (PDM)). Catatan : Nama yang diberikan pada sebuah penyimpanannya biasanya kata benda.

	Entitas luar (*external entity*) atau masukan (*input*) atau keluaran (*output*) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan. Catatan : Nama yang digunakan pada masukan (*input*) atau keluaran (*output*) biasanya berupa kata benda.
	Aliran data; merupakan data yang dikirim antar proses dari penyimpanan ke proses, atau dari proses ke masukan (*input*) atau keluaran (*output*). Catatan : Nama yang digunakan pada aliran data biasanya dapat diawali dengan kata data, misalnya “data siswa” atau tanpa kata data, misalnya “siswa”.

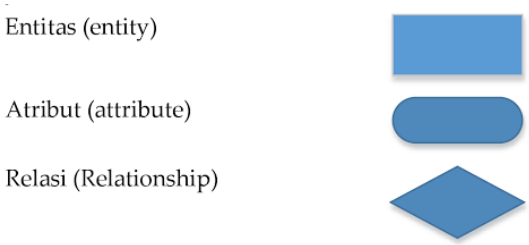
2.2.4. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah metode konseptual yang mendeskripsikan hubungan antara penyimpanan dalam DFD untuk memodelkan struktur data dan hubungan antar data. (Abdullah, 2015).

ERD adalah suatu permodelan berbasis pada persepsi dunia nyata yang mana terdiri dari kumpulan objek dasar yang disebut dengan entitas (*entity*) dan hubungan diantara objek-objek tersebut dengan menggunakan perangkat konseptual dalam bentuk diagram. (Lubis, 2016).

ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas. (Yanto, 2016).

ERD terdiri dari 3 Komponen Utama, yaitu :



Gambar 2.3 Komponen Entity Relationship Diagram

1. Entitas (*Entity*)

Entitas adalah suatu objek di dunia nyata yang dapat dibedakan dengan objek lainnya. Objek tersebut dapat berupa orang, benda ataupun hal lainnya. (Yanto, 2016).

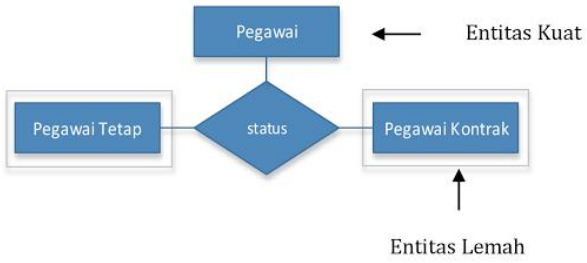
Dilihat dari jenisnya entitas terbagi atas dua yaitu :

a. Entitas Kuat (*Strong Entity*)

Entitas kuat adalah entitas yang dapat berdiri sendiri tidak bergantung pada entitas lainnya, entitas kuat memiliki atribut key dan entitas kuat digambarkan sebagai kotak persegi panjang bergaris tunggal.

b. Entitas Lemah (*Weak Entity*)

Entitas lemah adalah entitas yang tidak dapat berdiri sendiri. Entitas lemah merupakan hasil dari pembentukan entitas kuat, entitas lemah tidak memiliki atribut key dan entitas lemah digambarkan sebagai kotak persegi panjang bergaris ganda.



Gambar 2.4 Jenis Entitas

2. Atribut (*Attribute*)

Atribut merupakan semua informasi yang berkaitan dengan entitas. Atribut sering dikenal dengan property dari suatu entitas atau objek. Atribut digambarkan dalam bentuk lingkaran elips. Macam-macam atribut : (Yanto, 2016).

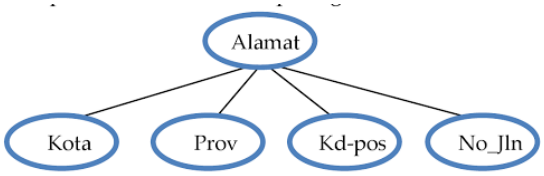
a. Atribut Sederhana (*Simple Attribute*)

Atribut sederhana adalah atribut yang nilainya tidak dapat dibagi lagi menjadi banyak yang lebih kecil.

Gambar 2.5 Atribut Sederhana

b. Atribut Komposit (*Composite Attribute*)

Atribut komposit adalah atribut gabungan yang nilainya dapat dipecah menjadi bagian lebih, atau sering disebut atribut yang terdiri dari beberapa atribut kecil di dalamnya.



Gambar 2.6 Atribut Komposit

c. Atribut Bernilai Tunggal (*Single Value Attribute*)

Atribut bernilai tunggal adalah jenis atribut yang nilainya hanya suatu dari entitas.

Gambar 2.7 Atribut Bernilai Tunggal

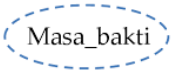
d. Atribut Bernilai Banyak (*Multivalues Attribute*)

Atribut bernilai banyak adalah jenis atribut yang nilainya lebih dari satu dalam suatu entitas tertentu.

Gambar 2.8 Atribut Bernilai Banyak

e. Atribut Turunan (*Derived Attribute*)

Atribut turunan adalah jenis atribut yang nilainya diperoleh dari atribut yang lain.



Gambar 2.9 Atribut Turunan

f. Atribut Identitas (*Key Attribute*)

Atribut identitas adalah atribut yang dijadikan sebagai kunci pada suatu tabel. Sifat atribut identitas ini unik, tidak ada yang menyamai, atribut identitas ini terdiri dari beberapa jenis, yaitu :

1. *Super Key*

   Super key adalah satu atribut atau kumpulan atribut yang secara unik mengidentifikasi sebuah baris di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dalam *set* entitas.

2. *Candidate Key*

   Candidate key adalah atribut yang menjadi determinan yang dapat dijadikan identitas baris pada sebuah relasi. Biasanya *super key* minimum.

3. *Primary Key*

   Primary key adalah *candidat key* yang dipilih untuk mengidentifikasi baris data secara unik dalam relasi.

4. Alternative Key

   Alternative key adalah *candidate key* yang tidak terpilih sebagai *primary key* atau atribut untuk menggantikan kunci utama.

5. *Foreign Key*

   Foreign key adalah atribut dengan *domain* yang sama yang menjadi kunci utama sebuah relasi, tetapi pada relasi lain atribut tersebut sebagai atribut biasa.

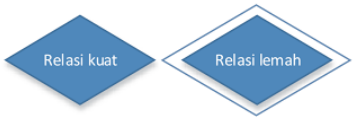
6. *Composite Key*

   Composite key adalah kunci yang terdiri dari dua atribut atau lebih. Atribut-atribut tersebut jika berdiri sendiri tidak menjadi identitas baris, tetapi bila dirangkaikan menjadi satu kesatuan akan dapat mengidentifikasi secara unik.

Gambar 2.10 Atribut Identitas

3. Tipe Relasi

Gambar belah ketupat merupakan perlambangan relasi antar entitas atau sering disebut kerelasian. Ada 2 macam penggambaran relasi yaitu kuat dan lemah. Relasi kuat adalah untuk menghubungkan antara entitas kuat sedangkan lemah untuk menghubungkan antar entitas kuat dengan entitas lemah.



Gambar 2.11 Tipe Relasi

Ada tiga macam relasi menurut derajatnya, yaitu [1] *Unary* adalah relasi yang menghubungkan entitas yang menghubungkan entitas yang sejenis, [2] *Binary* adalah relasi yang menghubungkan entitas yang tidak sejenis, [3] *Ternary* adalah relasi yang menghubungkan lebih dari dua entitas yang tidak sejenis.

Gambar 2.12 Derajat Relasi *Unary*

Derajat hubungan *unary* adalah entitas dosen hanya bekerja sama dengan entitas dosen yang entitas-nya sejenis, begitu juga dengan entitas karyawan (pimpinan) mengkoordinasi entitas karyawan (pekerja) yang entitas-nya sejenis.

Gambar 2.13 Derajat Relasi *Binary*

Derajat relasi *binari* adalah entitas kepala program studi berelasi dengan entitas program studi.

Gambar 2.14 Derajat Relasi *Ternary*

Derajat relasi *ternary* adalah entitas mahasiswa berelasi dengan dua entitas yang berbeda yaitu matakuliah dan nilai.

4. Derajat Kardinalitas

Derajat kardinalitas merupakan penjabaran dari hubungan antar entitas. Derajat kardinalitas dibagi atas 3 bagian, yaitu :

1. Derajat Kardinalitas ***One to One***

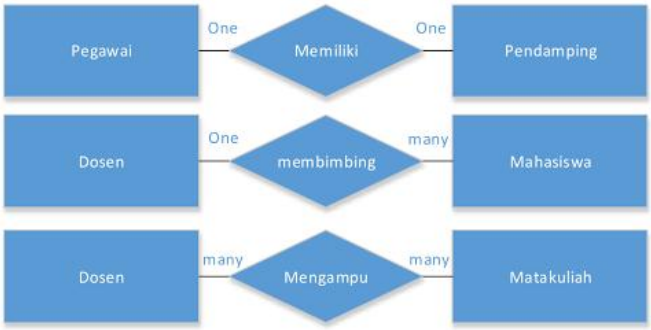
   Derajat kardinalitas *one to one* terjadi jika satu entitas x hanya berelasi dengan satu entitas y, ataupun sebaliknya.

2. Derajat Kardinalitas ***One to Many***

   Derajat kardinalitas *one to many* terjadi jika satu entitas x berelasi dengan banyak entitas y, ataupun sebaliknya.

3. Derajat Kardinalitas ***Many to Many***

   Derajat kardinalitas *many to many* terjadi jika banyak entitas x berelasi dengan banyak entitas y, ataupun sebaliknya.



Gambar 2.15 Contoh Derajat Kardinalitas

2.2.5. HTML (Versi 5)

HTML merupakan bahasa pemrograman yang digunakan untuk menampilkan dokumen pada *browser* dalam sebuah *web*. HTML bertujuan untuk mendefinisikan struktur dokumen web dan tata letak tampilan. HTML menggunakan beragam *tag* dan *atribut*. Sebuah dokumen HTML ditandai dengan *tag* awal <HTML> dan diakhiri </HTML. (Sari, 2006 *dalam* Jayanti dan Iriani, 2014).

2.2.6. PHP (Versi 5)

PHP adalah bahasa *scripting* yang menyatu dengn HTML dan dijalankan pada *serverside*. Artinya semu *sintaks* yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirim hanya hasilnya saja. PHP singkatan dari *Hypertext Preprocessor* yang digunakan sebagai bahasa *script serverside* dalam pengembangan web yang disisipkan pada dokumen HTML.

PHP merupakan *software Open-Source* yang disebarkan dan dilisensi secara gratis serta dapat di *download* secara bebas dari situs resminya. (Luqman, 2012 *dalam* Aminuddin, dkk, 2015).

Hypertext Preprocessor atau yang sering disingkat dengan PHP menurut merupakan bahasa pemrograman *web* dari sisi server yang disisipkan (*embedded script*) dalam dokumen HTML. Kode PHP diapit dengan menggunakan tag awal <?php dan diakhiri ?>. (Sidik, 2004 *dalam* Sari, 2006 *dalam* Jayanti dan Iriani, 2014).

PHP adalah bahasa *scripting* yang menampilkan *output* HTML ataupun *output* lain sesuai dengan keinginan pemrogram (misalnya : PDF dan lain-lain) yang dijalankan pada *server side*. Artinya semua yang sintak diberikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirim pada browserhanya menghasilkan *output* saja. (Abdullah, 2015).

2.2.7. MySQL (Versi 5)

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multiuser*. MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang *Free Software* bebas digunakan untuk keperluan pribadi atau usaha tanpa harus membayar atau membeli lisensi *GNU/GPL(General Public Lisenci)*. (Luqman, 2012 *dalam* Aminudin, dkk, 2015).

MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread* dan *multi-user*. MySQL sangat popular pengembang *web* (*web developers*) karena memiliki kecepatan dan ukuran yang kecil membuat MySQL lebih ideal untuk *website* ditambah lagi dengan fakta bahwa MySQL adalah *open source*yang berarti gratis. (Abdullah, 2015).

2.2.8. Pengertian *Javascript*

Javascript adalah bahasa pemrograman yang memungkinkan anda memodifikasi HTML anda dengan animasi, interaktivitas dan efek visual yang dinamis. *Javascript* dapat membuat halaman *web* lebih berguna dengan menyediakan umpan balik secara langsung. (McFarland, 2014).

2.2.9. Pengertian *Jquery* (Versi 3)

Jquery adalah *library* dari javascript yang memungkinkan anda memulai pemrograman anda dengan menangani banyak rincian berantakan *javascript* untuk anda. (McFarland, 2014).

2.2.10. Pengertian CSS (Versi 3)

CSS (*cascading style sheets*) - adalah tata letak dan format bahasa untuk menciptakan dan bahasa format markup seperti HTML. Idealnya, dokumen HTML hanya berisi informasi semantik, dan dengan CSS ini kemudian diformat dalam desain dan tipografi. (Krause, 2016).

2.2.11. *Bootstrap* (Versi 4)

Bootstrap merupakan produk *open source* yang dibuat oleh Mark Otto dan Jacob Thornton yang ketika awal dirilis, keduanya merupakan karyawan di twitter, dan ada kebutuhan untuk menstandarisasi perlengkapan (*toolsets*) dari antarmuka para insinyur yang ada di perusahaan. (Spurlock, 2013).