

BAB II

LANDASAN TEORI

2.1. Tinjauan Pusaka

Pada bab ini penulis akan membahas teori yang berhubungan penelitian yang penulis lakukan.

A. Konsep Dasar Sistem Informasi.

Menurut Fatta (2009:3) terdapat banyak pengertian system yang diungkapkan oleh beberapa pakar manajemen yang dijelaskan dari berbagai sudut pandang yang berbeda, namun mempunyai tujuan yang sama. Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan.

Berikut adalah definisi sistem secara umum :

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama. Contoh :
 - a. Sistem tata surya
 - b. Sistem pencernaan
 - c. Sistem informasi
 - d. Sistem komputer
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain.

Menurut Supriyanto (2010:238). “Sistem informasi merupakan suatu sistem yang tujuannya menghasilkan informasi”. Informasi adalah data yang diolah menjadi bentuk yang berarti bagi penerimanya dan bermanfaat dalam mengambil keputusan saat ini atau mendatang. Dasar dari informasi adalah data, kesalahan dalam mengambil atau menamasukan data, dan kesalahan dalam mengolah data akan menyebabkan kesalahan dalam memberikan informasi. Jadi data yang didapatkan dan diinputkan harus valid (benar) hingga bentuk pengolahannya, agar bias menghasilkan informasi yang dapat dipercaya. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata. Kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu kesatuan. Nyata (*fact and entity*) adalah berupa objek nyata seperti tempat, benda dan orang atau benar-benar ada dan terjadi.

Menurut Sutanta (2010:20). “Suatu informasi dapat dikatakan memiliki manfaat dalam proses pengambilan keputusan apabila informasi tersebut mempunyai kualitas dan nilai”. Kriteria informasi yang berkualitas adalah :

1. Akurat

Yang berarti informasi harus tidak bias atau menyesatkan dan bebas dari kesalahan dengan tujuan mengurangi kesalahan dalam mengambil keputusan.

2. Tepat Waktu

Yang berarti informasi yang sampai kepada penerimaan tidak boleh terlambat. Mahalnya nilai informasi saat ini adalah karena harus cepatnya informasi tersebut didapatkan, mengolah, dan mengirimkannya. Keterlambatan informasi tidak akan memberikan nilai bagi karena keterlambatan akan mengakibatkan perubahan dalam pengambilan keputusan.

3. Relevan

Yang berarti informasi harus mempunyai manfaat bagi pihak yang menerimanya, harus sesuai dengan kebutuhan dan siapa yang membutuhkan.

Informasi diperoleh dari sistem informasi (information system) atau disebut juga processing sistem atau information processing sistem. Jadi sistem informasi adalah suatu sistem di dalam suatu organisasi, yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan diperlukan.

Sistem informasi mempunyai enam buah komponen, yaitu :

1. Komponen Input

Input merupakan data yang masuk ke dalam sistem informasi. Komponen ini perlu ada karena merupakan bahan dasar dalam pengolahan informasi. Sistem informasi tidak akan dapat menghasilkan informasi jika tidak mempunyai komponen input. Input yang masuk ke dalam sistem informasi dapat langsung diolah menjadi informasi atau jika belum dibutuhkan segera dapat disimpan terlebih dahulu di storage dalam bentuk basis data (*database*).

2. Komponen Model

Informasi yang dihasilkan oleh sistem informasi berasal dari data yang diambil dari basis data yang diolah lewat suatu model-model tertentu. Model-model yang digunakan di sistem informasi dapat berupa model logika yang menunjukkan suatu proses perbandingan logika atau model matematik yang menunjukkan proses perhitungan matematik

3. Komponen Output

Produk dari sistem informasi adalah output berupa informasi yang berguna bagi pemakainya. Output merupakan komponen yang harus ada di komponen sistem informasi. Sistem informasi yang tidak pernah menghasilkan output tetapi selalu menerima input dikatakan bahwa input yang diterima masuk ke dalam lubang yang dalam (*deep hole*). Output dari sistem informasi dibuat dengan menggunakan data yang ada di basis data dan proses menggunakan model tertentu.

4. Komponen Teknologi

Teknologi merupakan komponen yang penting di sistem informasi. Tanpa adanya teknologi yang mendukung, maka sistem informasi tidak akan dapat menghasilkan informasi yang tepat waktunya. Komponen teknologi mempercepat sistem informasi dalam pengolahan datanya.

5. Komponen Basis Data

Basis data (*database*) adalah kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Komponen Kontrol

Komponen kontrol juga merupakan komponen yang penting dan harus ada di sistem informasi. Komponen control ini digunakan untuk menjamin bahwa informasi merupakan informasi yang akurat.

B. Sistem Informasi Penggajian

Menurut Saksono (2012:24), “Salah satu faktor utama yang pertama kali dinilai oleh seorang karyawan untuk bekerja disuatu perusahaan adalah mengenai

balas jasa yang akan diberikan oleh perusahaan tersebut.” Dari keterangan diatas dapat dijelaskan bahwa gaji merupakan pembayaran balas jasa yang dibayarkan secara berkala kepada manajer, tenaga administrasi dan tenaga professional dan pemberian jasa sejenis. Yang perlu diperhatikan adalah bahwa pembayaran balas jasa hendaklah disesuaikan dengan prestasi kerja yang baik, pengalaman loyalitas dan tanggung jawab yang diberikan oleh karyawan terhadap perusahaan. Hal ini dapat tercapai melalui perancangan pembayaran balas jasa yang efektif.

Gaji merupakan peranan yang sangat penting dalam kinerja suatu perusahaan dan pegawai yaitu:

1. Pegawai bekerja baik
2. Memenuhi kebutuhan hidup
3. Kepuasan pegawai
4. Ketenangan dalam bekerja

Sistem penggajian merupakan langkah awal penerapan manajemen kepegawaian. Peraturan sistem penggajian harus disetujui baik oleh pihak pimpinan maupun pegawai, sehingga tanggung jawab pelaksanaan sistem penggajian tidak hanya ditanggung oleh pemimpin saja.

Penggajian diambil dari kata gaji, dimana pengertian gaji ini telah dibahas sebelumnya. Istilah gaji biasanya digunakan untuk pegawai yang menerima hasil kerjanya dalam bentuk uang yang dibayarkan setiap awal bulan sekali.

Dari definisi diatas dapat disimpulkan bahwa sistem penggajian adalah seperangkat unsur yang saling berkaitan dan membentuk totalitas yang menekankan pada pemberian imbalan atau jasa atas hasil kerja seseorang.

C. Sekilas tentang PHP

Menurut Peranginangin (2006:2) mengatakan bahwa “PHP singkatan dari (*Hypertext Preprocessor*) yang digunakan sebagai bahasa *scriptserver-side* dalam pengembangan *web* yang disisipkan pada dokumen HTML”.

D. Sekilas tentang MySQL

Menurut Sutarman (2011:26), “MySQL dikembangkan oleh sebuah perusahaan Swedia bernama MySQL AB, yang pada saat itu bernama TcX Data Konsult AB, sejak sekitar 1994-1995, meski cikal bakal kodenya bisa disebut sudah ada sejak 1979”. Tujuan mula-mula TcX membuat MySQL pada waktu itu juga memang untuk mengembangkan aplikasi Web untuk klien. TcX adalah perusahaan pengembang software dan konsultan database. Pada saat itu Michael Widenius, pengembang satu-satunya di TcX, memiliki aplikasi UNIREG dan rutin ISAM yang dibuat sendiri dan sedang mencari antarmuka SQL untuk ditempelkan di atasnya.

MySQL versi 1.0 dirilis Mei 1996 secara terbatas kepada empat orang. Baru di bulan Oktober versi 3.11.0 dilepas ke publik. Namun kode ini mula-mula tidak diberikan dibawah lisensi khusus yang intinya kurang lebih begini : “Sourcecode MySQL dapat dilihat dan gratis, serta server MySQL dapat dipakai tanpa biaya tapi hanya untuk kebutuhan non komersial. Untuk kebutuhan komersial (misal : mengemas dan menjual MySQL, atau menyertakan MySQL dalam program komersial lain) anda harus membayar lisensi. Sementara distribusi Windows MySQL sendiri dirilis secara shareware. Barulah pada Juni 2000 MySQL AB mengumumkan bahwa sejak versi 3.23.19, MySQL adalah software bebas berlisensi GPL. Artinya “Source code MySQL dapat dilihat dan gratis, serta

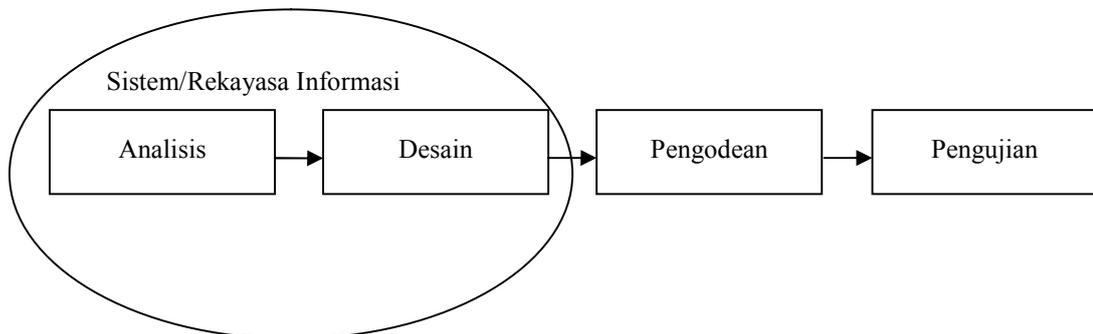
server MySQL dapat dipakai tanpa biaya untuk untuk kebutuhan siapapun. Tapi jika anda memodifikasi source code, anda juga harus melepaskannya di bawah lisensi yang sama yaitu GPL”. Kini perusahaan MySQL AB, yang beranggotakan sekitar 10 programmer dan 10 karyawan lain itu, memperoleh pemasukan terutama dari jasa konsultasi seputar MySQL.

Seri 4.x. di seri yang baru berjalan hingga 4.0 tahap alfa ini, pengembang MySQL berjanji akan mengadakan MySQL satu derajat lebih tinggi lagi. Fitur-fitur yang sejak dulu diminta akan dikabulkan, seperti *subselect* (di 4.1), *union* (4.0), *foreign key constraint* (4.0 atau 4.1, meski InnoDB sudah menyediakan inidi 3.23.x), *stored procedure* (4.1), *view* (4.20), *cursor* (4.1 atau 4.2), *trigger* (4.1). MySQL AB tetap berdedikasi mengembangkan dan memperbaiki MySQL, serta mempertahankan MySQL sebagai database *open source* terpopuler.

E. Konsep Dasar Model Pengembangan Sistem

Metode pengembangan perangkat lunak yang penulis gunakan adalah metodologi siklus hidup pengembangan sistem (*System Development Life Cycle*) atau SDLC. SDLC memiliki beberapa model dalam penerapan prosesnya, model yang akan penulis ambil dalam skripsi ini adalah model *waterfall*.

Model *waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahapan pendukung (*support*).



Sumber : Rosa dan Shalahuddin (2011:27)

Gambar II.1
Ilustrasi model *waterfall*

Berikut adalah tahapan-tahapan dalam model *waterfall* :

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa 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. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

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 tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik 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 program pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

F. Konsep Basis Data

Basis Data terdiri dari 2 kata, yaitu Basis dan Data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (karyawan, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Menurut Hariyanto (2010:15) “Basis Data sendiri dapat didefinisikan sebagai kumpulan file-file yang mempunyai kaitan antara satu file dengan file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu”. Bila terdapat file yang tidak dapat dipadukan atau dihubungkan dengan file yang lainnya berarti file tersebut bukanlah kelompok dari satu basis data, ia akan dapat membentuk satu basis data sendiri.

Sehingga prinsip utama dari basis data adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau harddisk) yang lebih dikenal sebagai komputer. Agar data-data tersebut lebih bermanfaat, maka data tersebut harus diorganisasikan dalam sebuah file basis data, yang bertujuan memperoleh informasi tertentu yang dapat dimanfaatkan. Cara penyajian informasi bermacam-macam, ada dengan memvisualisasikan dengan grafik, diagram, gambar, dan sebagainya. Bagi sebuah organisasi atau perusahaan, sistem informasi merupakan asset paling besar dan sangat berguna. Dengan informasi ini, organisasi akan mempengaruhi perkembangan perusahaannya.

Sehingga ada anggapan bahwa kualitas proses pengolahan data yang menjadi bahan terbentuknya informasi masih dipengaruhi oleh beberapa unsur yaitu :

1. *Hardware* yaitu : perangkat keras yang dibutuhkan untuk membantu pengolahan basis data. Dalam hal ini saja perangkat komputer beserta perangkat-perangkat pendukungnya.

2. *Brainware* yaitu : perangkat manusianya. Bagaimana dengan manusia sebagai pengelolanya.
3. *Software* yaitu : perangkat lunak yang digunakan untuk mengoperasikan komputer. Unsur software ini meliputi sistem operasi yang meruankan dasar pengoprasiaan komputer, program aplikasi yang telah dibuat programmer, serta yang tak kalah pentingnya adalah unsur DBMS (*Database Management System*).

Secara umum sebuah sistem basis data merupakan sistem yang terdiri atas kumpulan file (*table*) yang saling berhubungan (dalam sebuah sistem basis data di sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai atau program lain untuk mengakses dan memanipulasi file-file (*table-table*) tersebut.

Dalam sistem basis data ini mempunyai beberapa elemen penting yaitu basis data sebagai inti dari sistem basis data, perangkat lunak untuk mengelola basis data, perangkat keras sebagai pendukung operasi pengolahan data, dan manusia yang mempunyai peranan penting dalam sistem tersebut.

Ada beberapa ketentuan yang harus diperhatikan pada pembuatan file basis data agar dapat memenuhi kriteria sebagai suatu basis data yaitu :

1. Kerangkapan data (*data redundancy*), yaitu : munculnya data-data yang sama secara berulang-ulang pada file basis data yang semestinya tidak perlu. Dalam penjelasannya data redundansi akan mengakibatkan proses updating yang lebih lama dan memungkinkan terjadinya ketidak konsistenan data (*data inconsistency*), yang semakin besar.

2. Ketidak konsistenan data (*data inconsistency*), yaitu : munculnya data yang tidak konsisten pada medan yang sama untuk beberapa file dengan kunci yang sama. Hal ini terjadi akibat terjadinya kesalahan dalam pemasukan data (*data entry*) atau *update anomaly*, yaitu: suatu proses untuk mengupdate data tetapi mengakibatkan munculnya data yang tidak konsisten atau kehilangan informasi tentang objek yang ditinjau.
3. Data terisolasi, yaitu: disebabkan oleh pemakai beberapa file basis data. Program aplikasi yang digunakan tidak dapat mengakses file tertentu dalam sistem basis data tersebut, kecuali jika program aplikasi dibuat atau ditambah sehingga seolah-olah ada file yang terpisah atau terisolasi terhadap file yang lain.
4. Masalah keamanan (*security problem*), yaitu : berhubungan dengan masalah keamanan data dalam sistem basis data. Pada prinsipnya file basis data hanya digunakan oleh pemakai tertentu yang mempunyai wewenang untuk mengaksesnya. Pembatasan ini dilakukan secara intern dalam program aplikasi yang digunakan. *Security problem* juga dimaksudkan untuk memberikan perlindungan kepada data-data dalam sistem basis data dari kerusakan akibat bahaya kebakaran, banjir, badai, dan sebagainya.
5. Masalah integritas (*integrity problem*), yaitu : berhubungan dengan untuk kerja sistem agar dapat melakukan kendali atau kontrol pada semua bagian sistem sehingga sistem selalu beroperasi dalam pengendalian yang penuh.

Abstraksi data adalah penggambaran yang diungkapkan dalam bahasa dan gambar yang mudah dimengerti dengan penyembunyian kerumitan dari data. Ada

tiga tingkatan atau kelompok (user) dalam tingkatan abstraksi data saat memandang suatu basis data, ketiga tingkatannya adalah:

1. Level Fisik (*Physical Level*)

Merupakan level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada level ini, pemakai melihat data sebagai gabungan dari struktur dan datanya sendiri.

2. Level Logik (*Conseptual Level*)

Merupakan level berikutnya dalam abstraksi data yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data dan hubungannya dengan data yang lain.

3. Level Penampakan/Pandangan Pemakai (*View Level*)

Merupakan level tertinggi dari abstraksi data yang hanya menunjukkan sebagian dari basis data atau menyembunyikan data yang lain yang tidak diperlukan oleh kelompok pemakai tersebut. Di level ini diciptakan kemudahan interaksi antara dengan sistem.

G. UML (*Unified Modelling Language*)

Adapun pengertian UML menurut Rosa dan Shalahuddin (2011:118), Sebuah standarisasi bahasa pemodelan untuk pembangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *unifield Modeling Language* (UML).

Adapun beberapa jenis diagram pada UML yang dapat membantu perancangan sistem, menurut Rosa dan Shalahuddin (2011:122) adalah sebagai berikut:

1. *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada dua hal utama pada *use case* yaitu :

- 1) Aktor : Merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem informasi yang akan dibuat sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- 2) *Use Case* : Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2. *Activity Diagram*

Diagram aktivitas atau *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

3. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlihat dalam sebuah *use case* beserta

metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

4. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan di antara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

5. *Deployment Diagram*

Diagram *Deployment* atau *Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* dapat juga digunakan untuk memodelkan sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node*, dan *hardware*, sistem *client/server*, sistem terdistribusi murni dan rekayasa ulang aplikasi.

H. ERD (*Entity Relationship Diagram*)

Menurut Al-Bahra (2008:142) "ERD adalah suatu model jaringan yang menggunakan susunan data yang di simpan dalam sistem secara abstrak". *Entity Relation Diagram* (ERD) digunakan untuk menggambarkan hubungan antara *data store* yang ada di dalam diagram hubungan data antara lain:

a. *Entity*

Entity di gambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat di kelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsure waktu didalamnya).

b. *Relationship*

Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara *entitas*. Pada umumnya *relationship* diberi nama dengan kata kerja, sehingga memudahkan untuk melakukan pembacaan relasinya. Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk persegi panjang.

c. *Atribut*

Atribut adalah sifat atau karakteristik dari tiap *entitas* maupun tiap *relationship*. Menjelaskan apa yang sebenarnya yang dimaksud *entitas* maupun *relationship*, sehingga atribut adalah *element* dari setiap *entitas* dan *relationship*.

d. *Kardinalitas (cardinality)*

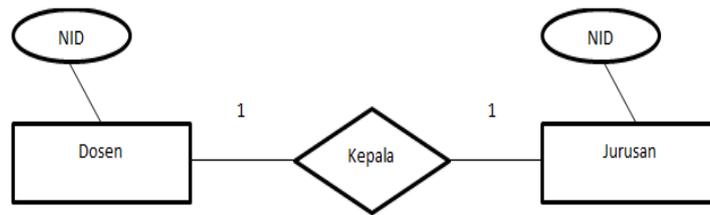
Kardinalitas Relasi menunjukkan jumlah *maksimum tupelo* yang dapat berelasi dengan *entitas* pada entitas yang lain.

Kardinalitas relasi merujuk kepada hubungan *maksimum* yang terjadi dari *entitas* yang satu ke *entitas* yang lain dan begitu juga sebaliknya.

Terdapat 3 macam kardinalitas relasi, yaitu:

1. *One to One*

Tingkat hubunga satu ke satu, dinyatakan dengan suatu kejadian pada *entitas* pertama, hanya mempunyai satu hubunga dengan satu kejadian pada *entitas* yang kedua sebaliknya.



Menurut Al-Bahra (2010:160)

Gambar II.2

Diagram Kardinalitas One To One

2. *One to Many atau Many to One*

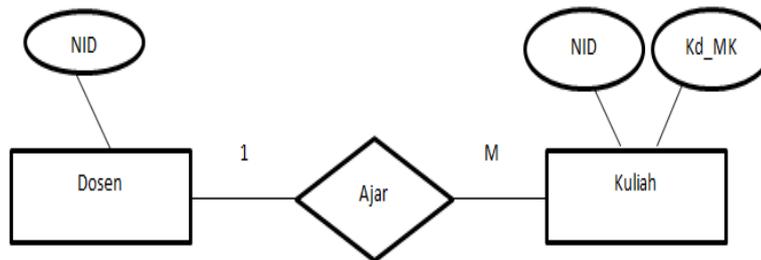
Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

a. *One to Many* (satu ke banyak)

Satu *tupel* pada entitas A dapat berhubungan dengan banyak *tupelo* pada entitas B, tetapi tidak sebaliknya, dimana setiap *tupel* pada entitas B, berhubungan dengan paling banyak satu *tupel* pada entitas A.

b. *Many to One* (banyak ke satu)

Setiap *tupelo* pada entitas A dapat berhubungan dengan paling banyak satu *tupelo* pada entitas B, tetapi tidak sebaliknya, dimana setiap *tupelo* pada entitas B berhubungan dengan paling banyak satu *tupel* pada entitas A.



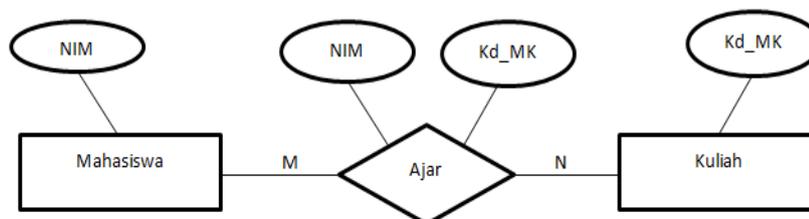
Menurut Al-Bahra (2010:165)

Gambar II.3

Diagram Kardinalitas One To Many

3. *Many to Many*

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



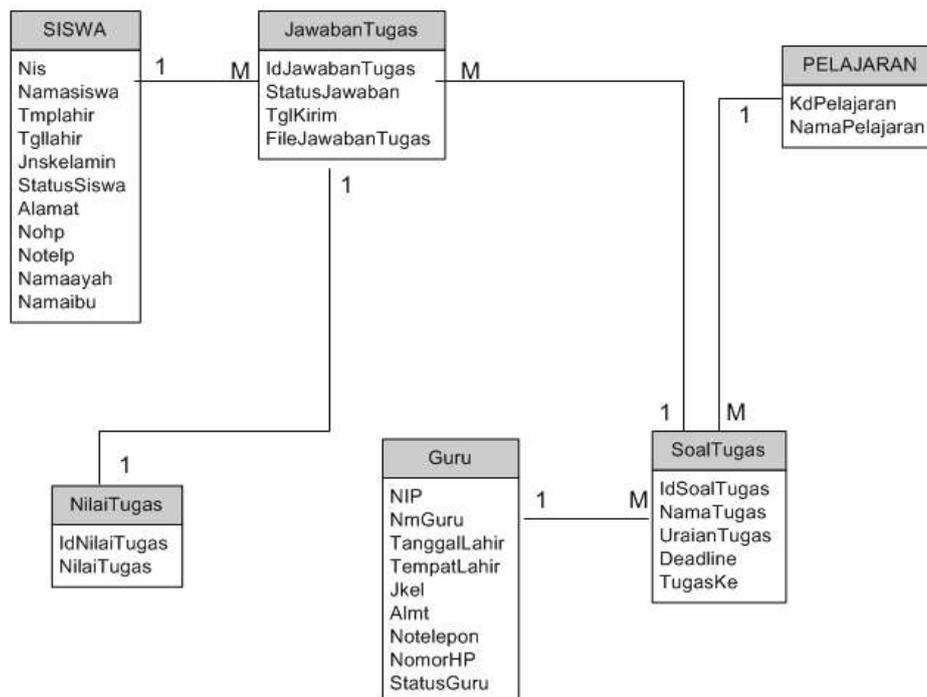
Menurut Al-Bahra (2010:169)

Gambar II.4

Diagram Kardinalitas Many To Many

I. Logical Record Structured (LRS)

Menurut Purnomo (2008:15) pengertian LRS adalah *representasi* dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil antar himpunan *entitas* yang digunakan untuk menentukan *kardinalitas*, jumlah table dan *Foreign Key (FK)*. Berikut adalah contoh gambar LRS :



Sumber: Purnomo (2008:15)

Gambar II.5.

Logical Record Structure (LRS)

2.2. Penelitian Terkait

Menurut Jayanti dkk (2014:36) "Sistem konvensional seperti ini membutuhkan waktu dan tenaga yang cukup lebih sehingga berjalan kurang efektif dan kesulitan dalam mencari data-data gaji yang diperlukan karena data tersebut disimpan hanya sebagai arsip. Adapun sistem penggajian yang sedang berjalan sekarang ini terkadang masih menemui banyak masalah diantaranya kesulitan dalam proses pembuatan slip gaji karyawan karena masih belum terkomputerisasi. Pihak administrasi pun juga kesulitan dalam proses pembuatan laporan gaji karyawan perbulan untuk diberikan pada Direktur CV. Blumbang Sejati. (Sumber : Wawancara dengan bagian personalia CV. Blumbang Sejati)."

Menurut Syaifudin dkk (2013:i) "Sejak awal berdirinya hingga saat ini sistem yang digunakan dalam pengolahan data penggajian masih menggunakan pembukuan secara konvensional, sehingga dalam menghitung jumlah gaji pegawai masih kurang efektif dan efisien serta data yang dihasilkan masih diragukan kevalidannya. Penghitungan gaji yang didasarkan pada berbagai rincian yang berbeda-beda menghambat proses kinerja bendahara instansi tersebut. Kesalahan dalam menghitung gaji bisa berakibat fatal karena akan berpengaruh dalam pembuatan laporan keuangan. Seluruh identifikasi masalah didapat dari proses wawancara pada bagian administrasi dan pembagian kuisioner pada seluruh karyawan ditoko winscom Kabupaten Pacitan."