

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Melihat dari pesatnya laju perkembangan teknologi informasi, sistem pencatatan data logistik bahan baku produksi akan lebih efektif menggunakan sistem komputerisasi berbasis *website*. Hal ini memungkinkan pengguna bisa melakukan pengolahan data dari mana saja. Sehingga dapat memudahkan pengguna dalam pengoperasiannya.

Fahmi Hakam dan Noor Alis Setiadi membuat sebuah jurnal yang berjudul “Pengembangan Sistem Pencatatan dan Pelaporan Data di Klinik Muhammadiyah Medical Center”. Jurnal tersebut bertujuan untuk mengembangkan sistem pencatatan dalam kinerja rumah sakit menggunakan metode *prototyping* dengan sistem pencatatan berbasis desktop.

Dalam sebuah skripsi yang berjudul “Perancangan Sistem Informasi Arsip Surat Menyurat Universitas U’Budiyah Indonesia Menggunakan PHP dan MySQL”. Skripsi tersebut bertujuan untuk memperbaiki sistem sebelumnya yang masih manual dan laporannya masih disimpan pada sebuah buku besar (Junidar, 2012).

Aplikasi berikutnya juga dibuat dalam skripsi dengan judul “Manajemen Arsip Borang Berbasis Desktop Aplikasi”. Pembuatan aplikasi tersebut menggunakan aplikasi yang berbasiskan Java dan MySQL (Sinta Adiyasa, 2016).

Dari penelitian sebelumnya, aplikasi yang dibuat hanya berupa pencatatan arsip keluar dan masuk serta masih berbasiskan desktop. Belum bisa dipergunakan kedalam sistem maupun penggunaan yang dapat diakses dari mana saja (berbasis *website*). Berdasarkan analisa dari penelitian sebelumnya maka penulis membuat sistem pencatatan berbasis *website* dengan beberapa fitur sebagai berikut:

- Pencatatan dapat dilakukan secara *online*
- Memudahkan dalam masalah *trace* atau pencarian data

- Sistem dapat menyimpan data arsip kemudian dapat mengekspor ke dalam bentuk *pdf file*.

2.2. Landasan Teori

2.2.1. Logistik

Logistik merupakan seni dan ilmu, barang, energi, informasi, dan sumber daya lainnya, seperti produk, dan jasa dari sumber produksi ke pasar dengan tujuan mengoptimalkan penggunaan modal. Manufaktur dan marketing akan sulit dilakukan tanpa dukungan logistik. Logistik juga mencakup integritas informasi, transportasi, inventori, pergudangan, *reverse logistics* dan pemaketan. Secara etimologi, logistik berasal dari bahasa Yunani kuno yang terdiri dari dua suku kata, yaitu “*Logic*” yang berarti rasional, masuk akal dan dapat dipertanggungjawabkan. Suku kata yang kedua adalah “*Thios*” yang berarti berpikir. Jika arti kedua suku kata dirangkai, memiliki makna berpikir rasional, masuk akal dan dapat dipertanggungjawabkan (Sutarman, 2005). Seiring berkembangnya jaman, arti logistik mengalami pergeseran. Logistik merupakan keseluruhan dari bahan, barang, alat dan sarana yang diperlakukan dan dipergunakan oleh suatu organisasi dalam rangka pencapaian tujuan dan berbagai sarasanya (Siagian, 2003).

Pendapat di atas diperkuat dengan pendapat (Dwiantara dan Rumsari, 2004) yang menyebutkan “Logistik adalah segala sesuatu atau benda yang berwujud dan dapat diperlakukan secara fisik (*tangible*), baik yang digunakan untuk menyelenggarakan kegiatan pokok maupun kegiatan penunjang (administrasi)”. Sementara (Yolanda Siagian, 2005) melihat logistik dari segi dunia bisnis yakni “Logistik merupakan bagian dari proses rantai suplai yang berfungsi merencanakan, melaksanakan, mengontrol secara efektif, efisien proses pengadaan, pengelolaan, penyimpanan barang, pelayanan dan informasi mulai dari titik awal (*point of origin*) hingga titik konsumsi (*point of consumption*) dengan tujuan memenuhi kebutuhan konsumen. Dengan demikian logistik adalah segala sesuatu baik itu berupa bahan, barang, alat, atau sarana yang digunakan untuk membantu kegiatan organisasi dalam rangka pencapaian tujuan.

2.2.2. Industri

Industri sering diidentikkan dengan semua kegiatan ekonomi manusia yang mengolah barang mentah atau bahan baku menjadi barang setengah jadi atau barang jadi. Dari definisi tersebut, istilah industri sering disebut sebagai kegiatan manufaktur (*manufacturing*). Padahal, pengertian industri sangatlah luas, yaitu menyangkut semua kegiatan manusia dalam bidang ekonomi yang sifatnya produktif dan komersial.

Karena merupakan kegiatan ekonomi yang luas maka jumlah dan macam industri berbeda-beda untuk tiap negara atau daerah. Pada umumnya, makin maju tingkat perkembangan perindustrian di suatu negara atau daerah, makin banyak jumlah macam industri, dan kompleks pula sifat kegiatan dan usaha tersebut. Cara penggolongan atau klasifikasian industri pun berbeda-beda. Tetapi pada dasarnya, pengklasifikasian industri didasarkan pada kriteria yaitu berdasarkan bahan baku, tenaga kerja, pangsa pasar, modal, atau jenis teknologi yang digunakan. Selain faktor-faktor tersebut, perkembangan dan pertumbuhan ekonomi suatu negara juga turut menentukan keanekaragaman industri negara tersebut, semakin besar dan kompleks kebutuhan masyarakat yang harus dipenuhi, maka semakin beranekaragam jenis industrinya.

Adapun klasifikasi industri berdasarkan kriteria masing-masing. Adalah sebagai berikut:

A. Klasifikasi industri berdasarkan bahan baku

Tiap-tiap industri membutuhkan bahan baku yang berbeda, tergantung pada apa yang akan dihasilkan dari proses industri tersebut. Berdasarkan bahan baku yang digunakan, industri dapat dibedakan menjadi:

- a. Industri ekstraktif, yaitu industri yang bahan bakunya diperoleh langsung dari alam. Misalnya: industri hasil pertanian, industri hasil perikanan, dan industri hasil kehutanan.
- b. Industri nonekstraktif, yaitu industri yang mengolah lebih lanjut hasil industri lain. Misalnya: industri kayu lapis, industri permintalan, dan industri kain.

- c. Industri fasilitatif atau disebut juga industri tertier. Kegiatan industrinya adalah dengan menjual jasa layanan untuk keperluan orang lain. Misalnya: perbankan, perdagangan, angkutan, dan pariwisata.

B. Klasifikasi industri berdasarkan tenaga kerja

Berdasarkan jumlah tenaga kerja yang digunakan, industri dapat dibedakan menjadi:

- a. Industri rumah tangga, yaitu industri yang menggunakan tenaga kerja kurang dari empat orang. Ciri industri ini memiliki modal yang sangat terbatas, tenaga kerja berasal dari anggota keluarga, dan pemilik atau pengelola industri biasanya kepala rumah tangga itu sendiri atau anggota keluarganya. Misalnya: industri anyaman, industri kerajinan, industri tempe atau tahu, dan industri makanan ringan.
- b. Industri kecil, yaitu industri yang tenaga kerjanya berjumlah sekitar 5 sampai 19 orang. Ciri industri kecil adalah memiliki modal relative kecil, pekerjanya berasal dari lingkungan sekitar atau masih ada hubungan saudara. Misalnya: industri genteng, industri batubata, dan industri pengolahan rotan.
- c. Industri sedang yaitu industri yang menggunakan tenaga kerja sekitar 20 sampai 99 orang. Ciri industri sedang adalah memiliki modal yang cukup besar, tenaga kerja memiliki keterampilan tertentu, dan pimpinan perusahaan memiliki kemampuan manajerial tertentu. Misalnya: industri konveksi, industri bordir, dan industri keramik.
- d. Industri besar, yaitu industri dengan jumlah tenaga kerja lebih dari 100 orang. Ciri industri besar adalah memiliki modal besar yang dihimpun secara kolektif dalam bentuk pemilikan saham, tenaga kerja harus memiliki keterampilan khusus, dan pimpinan perusahaan dipilih melalui uji kemampuan dan kelayakan (*fit and profer test*). Misalnya: industri tekstil, industri mobil, industri besi baja, dan industri pesawat terbang.

C. Klasifikasi berdasarkan proses produksi

Berdasarkan proses produksi, industri dapat dibedakan menjadi:

- a. Industri hulu, yaitu industri yang hanya mengolah bahan mentah menjadi barang setengah jadi. Industri ini sifatnya hanya menyediakan bahan baku untuk kegiatan industri yang lain. Misalnya: industri kayu lapis, industri alumunium, industri pemintalan, dan industri baja.
 - b. Industri hilir, yaitu industri yang mengolah barang setengah jadi menjadi barang jadi sehingga barang yang dihasilkan dapat langsung dipakai atau dinikmati oleh konsumen. Misalnya: industri pesawat terbang, industri konveksi, industri otomotif, dan industri meubeler.
- D. Klasifikasi industri berdasarkan produksi yang dihasilkan
- Berdasarkan produksi yang dihasilkan, industri dapat dibedakan menjadi:
- a. Industri premier, yaitu industri yang menghasilkan barang atau benda yang tidak perlu pengolahan lebih lanjut. Barang atau benda yang dihasilkan tersebut dapat dinikmati atau digunakan secara langsung. Misalnya: industri anyaman, industri konveksi, industri makanan dan minuman.
 - b. Industri sekunder, yaitu industri yang menghasilkan barang atau benda yang membutuhkan pengolahan lebih lanjut sebelum dinikmati atau digunakan. Misalnya: industri pemintalan benang, industri ban, industri baja, dan industri tekstil.
 - c. Industri tertier, yaitu industri yang hasilnya tidak berupa barang atau benda yang dapat dinikmati atau digunakan baik secara langsung maupun tidak langsung, melainkan berupa jasa layanan yang dapat mempermudah atau membantu kebutuhan masyarakat. Misalnya: industri angkutan, industri perbankan, industri perdagangan, dan industri pariwisata.
- E. Klasifikasi industri berdasarkan barang yang dihasilkan
- Berdasarkan barang yang dihasilkan, industri dibedakan menjadi:
- a. Industri ringan, yaitu industri yang menghasilkan barang siap pakai untuk dikonsumsi. Misalnya: industri obat-obatan, industri makanan, dan industri minuman.
 - b. Industri berat, yaitu industri yang menghasilkan mesin-mesin atau alat produksi lainnya. Misalnya: industri alat-alat berat, industri mesin, dan industri percetakan.

2.2.3. Arsip

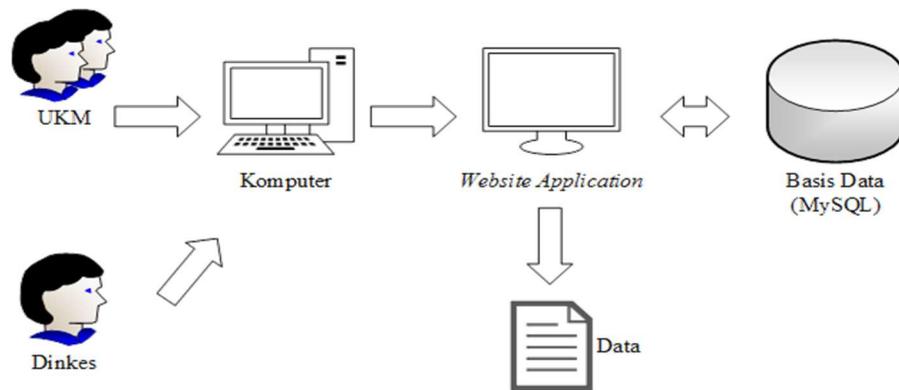
Arsip dapat diartikan sebagai suatu badan (*agency*) yang melakukan segala kegiatan pencatatan, penanganan, penyimpanan dan pemeliharaan surat-surat atau warkat-warkat yang mempunyai arti penting baik ke dalam maupun keluar, baik yang menyangkut soal-soal pemerintahan maupun non-pemerintahan, dengan menerapkan kebijaksanaan dan sistem tertentu yang dapat dipertanggungjawabkan (Basir Barthos, 2009).

Kearsipan adalah setiap catatan tertulis atau bergambar yang memuat keterangan mengenai suatu hal atau peristiwa yang dibuat orang untuk membantu ingatannya (The Liang Gie, 2000). Arsip merupakan keseluruhan dokumen-dokumen tertulis, lukisan-lukisan dan barang-barang cetakan yang secara resmi diterima atau dihasilkan oleh suatu badan pemerintahan atau salah seorang dari pejabat-pejabatnya sepanjang dokumen-dokumen itu di maksudkan untuk berada di bawah pemeliharaan dari badan itu atau pejabat itu.

Dapat dikatakan bahwa aktivitas pokok di bidang pencatatan ialah menyimpan data suatu barang atau produksi, tetapi tujuannya yang utama ialah menemukan kembali data yang diperlukan secara cepat.

2.2.4. Arsitektur perangkat lunak

Dalam mengembangkan aplikasi diperlukan perancangan arsitektur perangkat lunak yang bertujuan untuk menggambarkan bagaimana sistem dikembangkan dan dijalankan. Arsitektur perangkat lunak pada aplikasi dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Perangkat Lunak

2.2.5. *Unified Markup Language (UML)*

Unified Markup Language (UML) merupakan bahasa pemodelan secara grafis yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan seluruh rancangan aplikasi perangkat lunak. Penggunaan model UML bertujuan untuk mengidentifikasi bagian-bagian yang termasuk dalam lingkup sistem di dalam aplikasi. Model UML yang dipakai dalam pengembangan aplikasi penjualan dan pembelian antara lain adalah *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

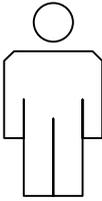
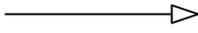
a. *Use Case Diagram*

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi yang dibutuhkan. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Simbol-Simbol Dalam *Use Case Diagram*

| No | Gambar | Nama Gambar | Keterangan |
|----|--------|-----------------|--|
| 1. | | <i>Use Case</i> | Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan <i>actor</i> . |

Tabel 2.2 Simbol-Simbol Dalam Use Case Diagram (Lanjutan)

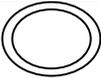
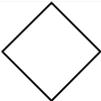
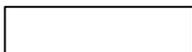
| No | Gambar | Nama Gambar | Keterangan |
|----|---|-----------------------|--|
| 2. |  | <i>Actor</i> | Merupakan <i>abstraction</i> dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> . |
| 3. |  | <i>Association</i> | Digambarkan dengan garis tanpa panah yang mengindikasikan siapa yang berinteraksi secara langsung dengan sistem. |
| 4. |  | <i>Generalization</i> | Mengindikasikan siapa yang berinteraksi secara pasif dengan sistem. |
| 5. | <code>-<<include>></code> | <i>Include</i> | Mengidentifikasi hubungan antar dua <i>use case</i> dimana satu <i>usecase</i> memanggil <i>usecase</i> yang lain. |
| 6. | <code>-<<extend>></code> | <i>Extend</i> | Merupakan perluasan dari <i>use case</i> jika kondisi atau syarat terpenuhi. |

b. Activity Diagram

Activity diagram merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau

proses bisnis. Simbol-simbol yang digunakan dalam *activity* diagram dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol-Simbol Dalam *Activity* Diagram

| No | Gambar | Nama Gambar | Keterangan |
|----|---|-----------------------|--|
| 1. |  | <i>End Point</i> | Merupakan akhir dalam aktifitas. |
| 2. |  | <i>Activities</i> | Menggambarkan suatu proses atau kegiatan bisnis |
| 3. |  | <i>Decision Point</i> | Menggambarkan pilihan untuk pengambilan keputusan dalam aktifitas. |
| 4. |  | <i>Fork Node</i> | Menggambarkan awalan dari percabangan aktifitas. |

2.2.8. *Back End*

Back end merupakan tempat dimana proses suatu aplikasi atau sistem berjalan, pada *back end* ini data di proses di tambahkan, diubah, atau dihapus. *Back end* mengurus segala sesuatu yang biasanya tidak dilihat atau berinteraksi langsung kepada *user*, seperti *database* dan *server*.

Orang yang biasanya bekerja sebagai *back end* adalah *programmer* atau *developer* yang fokus pekerjaannya pada keamanan, desain sistem, dan manajemen data pada sistem.

Bahasa pemrograman yang biasanya dikuasai *back end developer* adalah bahasa pemrograman yang dapat digunakan untuk mengelola *database*, mengolah *file* dan I/O seperti PHP, ASP, Java, NodeJs, atau C# (Digitaltutors.com, 2016).

2.2.9. NetBeans

NetBeans merupakan perangkat lunak (*software*) yang digunakan untuk mengembangkan aplikasi, di dalam perangkat lunak terdapat IDE (*Integrated Development Environment*). Sebuah IDE merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan GUI (*Graphic User Interface*).

2.2.10. JavaServer Faces (JSF)

JavaServer Faces merupakan suatu lingkungan pemrograman komponen antarmuka pemakai (UI atau *user interface*) yang menyediakan komponen antarmuka pemakai baku yang lebih variatif, dan dijalankan pada sisi server (Yanto, 2009). JSF merupakan Platform dari JavaPlatform, Enterprise Edition. JSF 2 menggunakan Facelets sebagai sistem *template defaultnya*. Teknologi tampilan lainnya juga dapat digunakan, seperti XUL.

Cara kerja JSF yang menyediakan komponen antarmuka pemakai dapat dihubungkan langsung ke server melalui servlet. Dengan komunikasi langsung dari komponen antarmuka pemakai ke servlet, class pada servlet dan *javabean* yang mirip tidak perlu ditulis ulang untuk setiap komponen antarmuka pemakai dan setiap halaman *web* dengan melihat pola tersebut pada *web*. *Web* dapat dibuat lebih sederhana tapi kaya akan variasi (Yanto, 2009). Pengecekan terhadap kesalahan juga lebih mudah dilakukan karena pengecekan hanya perlu dilakukan pada UI component atau class pada servlet yang terdapat kesalahan.

2.2.11. Hibernate

Hibernate ORM (atau yang biasa disebut *Hibernate*) merupakan sebuah library yang digunakan untuk *Object Relation Mapping* (ORM) antara objek-objek yang ada pada sebuah aplikasi (Fadlika Dita N, 2013). Biasanya disebut sebagai framework untuk pemetaan orientasi objek pada sebuah domain model sehingga merelasikan dengan *database*. *Hibernate* menangani masalah ketidakcocokan

objek relasi dengan mengganti akses *database* langsung, dan persisten dengan fungsi penanganan objek tingkat tinggi.

Fitur utama *Hibernate* adalah pemetaan dari kelas Java ke tabel *database*, dan pemetaan dari tipe data Java ke tipe data SQL. *Hibernate* juga menyediakan *query* data dan fasilitas pengambilan. Kemudian akan memanggil SQL dan mengurangi pengembangan dalam hal penanganan manual dan konversi objek kumpulan hasil.

2.2.12. Model View Controller (MVC)

Model View Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (*Trygve Reenskaug*) untuk meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan di tampilan (*view*) untuk direpresentasikan pada sebuah *user interface* (Deaceon, 2009). MVC mengikuti pendekatan yang paling umum dari *Layering*. *Layering* hanyalah sebuah logika yang membagi kode ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode (Satich, 2004).

Definisi teknis dari arsitektur MVC dibagi menjadi tiga lapisan (Burbeck, 1992)

A. Model

Model digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Hanya *model* yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Sebuah model meringkas lebih dari sekedar data dan fungsi yang beroperasi di dalamnya. Pendekatan *model* yang digunakan untuk komputer *model* atau abstraksi dari beberapa proses dunia nyata. Hal ini tidak hanya menangkap keadaan proses atau sistem bagaimana sistem bekerja. Sebagai contoh, *programmer* dapat menentukan *model* yang menjembatani komputasi *back-end* dengan *front-end* GUI (*Graphical user interface*).

B. View

View bertanggung jawab untuk pemetaan grafis sebuah perangkat. *View* biasanya memiliki hubungan dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. *View* melekat pada *model* dan me-render isinya ke permukaan layar. Selain itu, ketika *model* berubah, *view* secara otomatis menggambar ulang bagian layar yang terkena perubahan tersebut. Terdapat kemungkinan beberapa *view* pada *model* yang sama dan masing-masing *view* tersebut dapat merender isi *model* untuk permukaan tampilan yang berbeda.

C. *Controller*

Controller menerima *input* dari pengguna dan menginstruksikan *model* dan *view* melakukan aksi berdasarkan masukan tersebut. Sehingga, *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi. Sebagai contoh, ketika pengguna mengklik tombol atau memilih item menu, *controller* bertanggung jawab untuk menentukan bagaimana aplikasi seharusnya merespon.

Arsitektur MVC memiliki manfaat yaitu pemisahan antara *model* dan *view* memungkinkan beberapa *view* menggunakan *model* sebuah aplikasi lebih mudah untuk diterapkan, diuji, dan dipelihara, karena semua akses ke *model* berjalan melalui komponen ini (Balani, 2002).

2.2.13. Basis Data

Basis data adalah dua atau lebih simpanan data dengan elemen-elemen data penghubung, yang dapat diakses lebih dari satu cara. Basis data dinyatakan dengan teknik-teknik formal dan manajemen basis data.

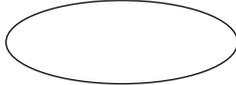
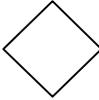
Basis data juga merupakan kumpulan sistem informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelolakan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS).

Dari definisi diatas, maka dapat dikatakan bahwa basis data merupakan kumpulan dari data sistem informasi yang saling berhubungan satu dengan yang lainnya.

2.2.14. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan atau berasal dari dunia nyata yang disebut dengan entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi (Doro Edi, 2009). Simbol-simbol yang membentuk ERD dapat dilihat pada tabel 2.4 dibawah ini.

Tabel 2.4 Simbol-Simbol ERD

| No | Gambar | Nama Gambar | Keterangan |
|----|---|---------------------|---|
| 1. |  | <i>Entity</i> | Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain. |
| 2. |  | <i>Attribute</i> | Atribut yang dimiliki oleh entitas, dimana dapat mendeskripsikan karakteristik dari entitas tersebut. |
| 3. |  | <i>Relationship</i> | Menunjukkan hubungan antara sejumlah entitas yang berbeda. |

Pada ERD terdapat kardinalitas relasi dimana merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas yaitu:

A. *One to One Relationship (1-1)*

Hubungan yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada entitas kedua.



Gambar 2.2 *One To One Relationship*

B. *One to Many Relationship (1-M)*

Hubungan yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berelasi dengan banyak entitas pada himpunan entitas yang lain.



Gambar 2.3 *One To Many Relationship*

C. *Many to Many Relationship*

Hubungan yang menunjukkan hubungan antara entitas pertama dengan entitas kedua adalah banyak ke banyak.



Gambar 2.4 *Many To Many Relationship*

2.2.15. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional yang di distribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tidak boleh dijadikan produk turunan yang bersifat komersial.

Berikut beberapa keistimewaan dari MySQL menurut beberapa sumber sebagai bahan pertimbangan:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka (*open source*). MySQL didistribusikan sebagai *open source* sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed or unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti password yang terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP atau IP, *Unix Socket* (UNIX), atau named pipes (NT).

10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.