BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Persediaan merupakan salah satu unsur yang paling aktif dalam operasi perusahaan yang secara continue diperoleh, diubah, yang kemudian dijual kembali. Sebagian besar dari sumber-sumber perusahaan juga sering dikaitkan didalam persediaan yang akan digunakan dalam perusahaan *manufactur*.

Dengan tersedianya persediaan maka diharapkan perusahaan dapat melakukan proses produksi sesuai kebutuhan atau permintaan konsumen. Selain itu dengan adanya persediaan yang cukup tersedia di gudang juga diharapkan dapat memperlancar kegiatan produksi pelayanan kepada konsumen. Perusahaan dapat menghindari terjdinya kekurangan barang. Keterlambatan jadwal dalam pemenuhan produk yang dipesan konsumen dan merugikan perusahaan dalam hal dapat memberikan dampak yang kurang baik bagi kualitas perusahaan`

Menurut Heizer dan rander (2015:553), "Persediaan adalah menentukan keseimbangan tidak akan pernah mencapau strategi berbiaya rendah tanpa persediaan managemen yang baik".

A. Model Pembelajaran Berbasis *Web*

Pembelajaran berbasis *web* atau yang biasa dikenal dengan *web based learning* merupakan salah satu jenis penerapan dan pembelajaran elektronik (*e-learning*).

Sedangkan menurut Hartley dalam Harsanto (2014:9) mengemukakan bahwa, *"E-Learning* adalah pembelajaran menggunakan internet, intranet, atau jaringan elektronik lain baik dalam hal pengembangan, penyampaian maupun evaluasi konten*"*.

Dari pengertian diatas dapat disimpulkan bahwa model pembelajaran berbasis *web* merupakan suatu kegiatan belajar mengajar yang memanfaatkan media situs atau *website* yang bisa diakses melalui jaringan internet dan lain-lain.

B. Pengertian Sistem

Menurut Djahir dan Pratita (2015:7) menyatakan bahwa "sistem dikelompokan menjadi dua bagian yang menekankan pada prosedurenya dan ada yang menekankan pada elemennya.kedua kelompok ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatanya".

Sedangkan, menurut Tyoso (2016:1),"sistem merupakan suatu kumpulan dari komponen-komponen yang membentuk satu kesatuan".

Dari definisi di atas, dapat disimpulkan bahwa sistem adalah sebagai suatu kumpulan atau himpunan antar group dan subssistem/ atau bagian /komponen yang terorgaisasi baik fisik maupun nonfisik

1. Karakteristik Sistem

Sutabri (2016:10) mengemukakan bahwa: "Model umum sebuah sistem adalah *input*, proses, dan *output*, hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat memiliki beberapa keluaran."

Adapun karakteristik-karakteristik sistem tersebut menurut Sutabri (2016:10), adalah:

a. Komponen Sistem *(Component)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerjasama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk sub sistem dimana setiap sub sistem memiliki sisfat dari sistem yang menjalankan suatu fungsi tertentu dan memengaruhi prosen sistem secara keseluruhan sehingga terjadilah suatu proses sistem tersebut untuk mencapai tujuan bersama.

b. Batasan Sistem *(Boundary)*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

c. Lingkungan Luar Sistem *(Environtment)*

   Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut. Dengan demikian, lingkungan luar tersebut harus tetap dijaga dan dipelihara, sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak, maka akan mengganggu kelangsungan hidup sistem tersebut.

d. Penghubung Sistem *(Interface)*

Media yang menghubungkan sistem dengan sub sistem lain disebut penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu sub sistem ke sub sistem lain. Bentuk keluaran dari satu sub sistem akan menjadi masukkan untuk sub sistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

e. Masukan Sistem *(Input*)

Energi yang dimasukkan kedalam sistem tersebut masukkan sistem, yang dapat berupa pemeliharaan *(maintenance input)* agar sistem tersebut beroprasi. Contoh, didalam suatu unit sistem komputer. "Program" adalah *maintenance* input yang digunakan untuk mengoprasikan komputernya dan "data" adalah sinyal input untuk diolah menjadi informasi. Selain itu sistem masukan dapat juga berupa masukan signal (signal input) yang bertujuan agar energi yang dimasukkan menghasilkan keluaran (output) contohnya informasi.

f. Keluaran Sistem (*Ouput*)

Keluaran adalah hasil energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna dan berupa sisa pembuangan sehingga keluaran ini dapat menjadi masukan bagi sub sistem yang lainnya. Contoh, sistem informasi, keluaran yang dihasilkan tentunya adalah informasi yang mana informasi ini dapat digunakan sebagai masukkan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi sub sistem lainnya.

g. Pengolah Sistem

Suatu sistem harus memiliki suatu perangkat yang bertugas mengolah bagian pengolah ini yang akan berubah maskan menjadi keluaran. Sebagai contoh, sistem produksi akan mengolah masukan yang berupa bahan baku dan barang-barang lainnya menjadi barang jadi, sistem akuntansi akan mengolah data transaksi menjadi laporan-laporan termasuk laporan keuangan yang dibutuhkan oleh pihak manajemen.

h. Sasaran Sistem (*Objektive*)

Tujuan dan sasaran merupakan sesuatu yang harus dimiliki oleh sistem. Suatu sistem harus memiliki tujuan dan sasaran yang pasti karena akan menentukan masukan yang dibutuhkan sistem yang tentunya juga berpengaruh pada keluaran sistem. Suatu dapat dikatakan berhasil bila dapat mengenai sasaran dan tujuan sesuai yang telah direncanakan.

2. Klasifikasi Sistem

Beberapa aspek dari sistem ini membuat pengguna sistem dapat mengklasifikasikan sistem yang relevan sesuai dengan arah pandang pengguna sistem.klasifikasi sistem menurut Tyoso (2016:5), terdiri dari:

a. Sistem alamiah

Sistem alamiah (*natural system)* muncul secara alamiah tanpa campur tangan manusia.

b. Sistem tiruan

Sistem tiruan (*artificial system)* diciptakan untuk mendukung tujuan tertentu.

c. Sistem deterministic

Sisitem deterministic (*deterministic system),* bekerjanya sistem ini dapat diramalkan sebelumnya. Masukan untuk sistem ini secara pasti menentukan jenis keluaranya.

d. Sistem probabilistic

Sistem probabilistic (*probabilistic system)* dapat dilacak hanya dengan menggunakan nilai distribusi probabilitas, selalu ada nilai ketidakpastian yang sesungguhnya pada sembarang waktu.

e. Sistem tertutup

Sistem tertutup (*closed system),* pada sistem ini tidak terjadi pertukaran atau penggunaan sumber daya dengan atau dari lingkunganya, mengingat sistem ini tidak menggunakan *input* dari lingkunganya, maka *output* dari sistem ini tidak berkaitan dengan lingkunganya pula.

f. Sisitem terbuka

Sisitem terbuka (*opened system)* mneggunakan sumber daya dari lingkunganya sehingga keluaranya berkaita dengan lingkunganya.

3. Pengertian Informasi

Hasil dari suatu sistem informasi yang dibutuhkan oleh pengguna sistem adalah informasi. Informasi ini tercipta dari kumpulan data-data yang telah diproses sebelumnya.

Menurut Cushing dalam Fauzi (2017:10) menyatakan bahwa ”informasi adalah kumpulan data yang relevan dan mempunyai arti yang menggambarkan suatu kejadian-kejadian atau kegiatan-kegiatan”. Sedangkan menurut Hutahean

(2015:9) ”informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Dapat disimpulkan bahwa informasi adalah kumpulan data yang telah diolah menjadi suatu bentuk yang lebih berguna bagi penerimanya dan mempunyai nilai nyata atau yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

4. Daur Hidup Sistem (*System Development Life Cycle*)

Dalam penulisan Tugas Akhir ini penulis mengembangkan sistem informasi dengan menggunakan metodologi pengembangan sistem informasi. Menurut Kadir (2014:244) mengemukakan bahwa “*SDLC* merupakan metodologi klasik yang digunakan untuk mengembangkan, memelihara dan menggunakan sistem informasi”.

C. Pemrograman

Menurut Kadir (2014:192) memberi pengertian bahwa “Program adalah sekumpulan instruksi yang digunakan untuk mengatur perangkat keras komputer agar melaksanakan tindakan tertentu”.

1. Pemrograman Berorientasi Objek

Menurut Kadir (2014:204) mengemukakan bahwa pemrograman berorientasi objek adalah mengombinasikan data dan prosedur-prosedur untuk mengakses data menjadi sebuah kesatuan unit.

2. Karakteristik Pemrograman Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama:

1. Pengkapsulan (*Encapsulation*)
    a. *Encapsulation* merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
    b. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
    c. Data terlindungi dari prosedur atau objek lain, kecuali prodesur yang berada dalam objek itu sendiri.
2. Pewarisan (*Inheritance*)
    a. *Inheritance* adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek induk diturunkan kepada anak objek, demiian seterusnya.
    b. *Inheritance* mempunyai arti bahwa atribut dan operasi yang dimiiliki bersama di antara kelas yang mempunyai hubungan secara hirarki.
    c. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimilikinya.
    d. Kelas objek dapat didefinisikan atribut dan *service* dari kelas objek lainnya.
    e. *Inheritance* menggambarkan generalisasi sebuah kelas.
3. *Polimorfisme*
    a. *Polimorfisme* yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.

b. *Polimorfisme* mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.
c. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon *message* yang sama.
d. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan objek.

3. Bahasa Pemrograman *JavaNetbeans*

Menurut isnandi dkk (2015:2)" Java merupakan bahasa pemograman berorientasi objek dan bebas platform, dikembangkan oleh SUN Micro System dengan jumlah keunggulan yang memungkinkan java dijadikan sebagai bahasa pengembangan enterprise". Java merupakan bahsa yang *powerfull* yang bisa digunakan dalam hamper semua bentuk pengembangan software. Anda dapat menggunakan java untuk membuat game, aplikasi desktop, aplikasi web, aplikasi enterprise, aplikasi jaringan, dan lain-lain. Yang menarik adalah bahwa java bias digunakan untuk membuat laporan yang dapat berjalan diatas HP,PDA, dan peralatan lain yang dilengkapi dengan Java Virtual Machine (JVM).

Menurut isnandi dkk (2015:2)"NeatBeans merupakan salah satu IDE yang dikembangkan dengan bahasa pemograman java". NetBeans mempunyai lingkup pemograman java terintergrasi dalam suatu perangkat lunak yang didalamnya menyediakan pembangunan pemrograman GUI, text editor, complier, dan interpreter. Netbeans adalah sebuah perangkat lunak open source sehingga dapat digunakan secara gratis untuk keperluan komersil maupun nonkomersial yang didukung oleh Sun Microsystem.

D. Basis Data *(Database)*

Menurut Lubis (2016:3) “Basis Data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/ perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut.

Pada kehidupan sehari-hari didunia komputer, basis data akan menggunakan media penyimpanan *(storage)*, yaitu berkaitan dengan setiap alat yang dapat menerima data yang dapat disimpan, dan dapat dipanggil kembali data itu pada waktu berikutnya atau setiap alat yang dapat digunakan untuk menyimpan data. Adapun media penyimpanan yang dapat digunakan terdiri dari *harddisk, diskette* atau *floppy disk, tape* maupun dengan *compact disk* (CD) atau DVD dan kini juga dapat digunakan *flashdisk*.

Dengan bantuan basis data ini diharapkan bahwa sistem informasi yang dibuat dapat terintegrasi antara bagian yang satu dengan yang lainnya, sehingga pada akhirnya tidak ada pembatas area dalam perusahaan. Dalam pembuatan dan penggunaan basis data, terdapat 4 (empat) komponen dasar sistem basis data, yaitu:

1. Data

   Data yang digunakan dalam sebuah basis data, haruslah mempunyai ciri sebagai berikut:

   a. Data disimpan secara integrasi, yaitu *database* merupakan kumpulan dari berbagai macam *file* dari aplikasi-aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian-bagian yang rangkap.

b. Data dapat dipakai secara bersama-sama, yaitu masing-masing bagian dari *database* dapat diakses oleh pemakai dalam waktu yang bersamaan, untuk aplikasi yang berbeda.

2. *Hardware*

Terdiri dari semua peralatan perangkat keras komputer yang digunakan untuk pengelolaan sistem *database*, seperti:

a. Peralatan untuk penyimpanan, *disk*, drum dan lain-lain.

b. Peralatan *input* dan *output.*

c. Peralatan komunikasi data.

3. *Software*

Berfungsi sebagai perantara antara pemakai dengan data fisik pada *database*, dapat berupa :

a. *Database Management System* (DBMS).

b. Program-program aplikasi dan prosedur-prosedur yang lain seperti Oracle, SQL Server, MySQL.

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya. MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini benama Tex Data Konsult AB sekitar tahun 1994-1995.

MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *web* yang *ideal.* MySQL lebih sering digunakan untuk membangun aplikasi berbasis

web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP. Adapun kelebihan-kelebihan dari MySQL yaitu:

1) *Source* MySQL dapat diperoleh dengan mudah dan gratis.
2) Sintaksnya lebih mudah dipahami dan tidak rumit.
3) Pengaksesan basis data dapat dilakukan dengan mudah.
4) MySQL merupakan program yang *multithreaded*, sehingga dipasang pada server yang memiliki multi CPU.
5) Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, dsb.
6) Bekerja pada berbagai *platform*. (tersedia berbagai versi untuk berbagai sistem operasi).
7) Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
8) Mendukung ODBC untuk sistem operasi *Windows*.
9) Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.

4. *User*

Terdiri menjadi 3 klasifikasi:

a. *Database Administrator*, yaitu orang yang bertugas mengelola sistem *database* secara keseluruhan.
b. *Programmer*, yaitu orang yang membuat program aplikasi yang mengakses *database* dengan menggunakan bahasa pemograman.

c. *End User,* orang yang mengakses *database* melalui terminal dengan menggunakan *query language* atau program aplikasi yang dibuat oleh *programmer.*

Penggunaan data pada basis data mempunyai peran yang kuat pada masing-masing bagian, yang dikelompokkan dalam 3 jenis data pada sistem basis data, yaitu:

a. Data operasional dari suatu organisasi, berupa data yang disimpan di dalam basis data.
b. Data masukan *(input data),* data dari luar sistem yang dimasukkan melalui peralatan *input* yang dapat mengubah data operasional.
c. Data keluaran *(output data),* berupa laporan melalui peralatan *output* sebagai hasil dari dalam sistem yang mengakses daa operasional.

1. Aplikasi *Database* MYSQL

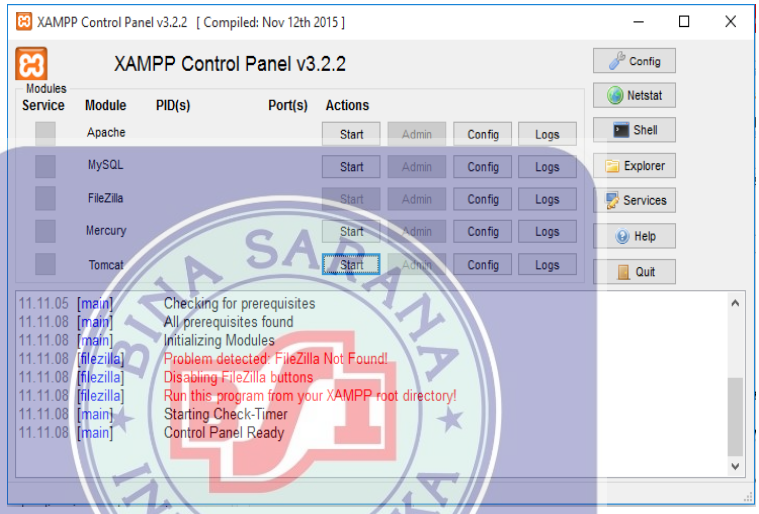
Aplikasi basis data sering digunakan oleh para pembuat aplikasi sebagai media pengolahan basis data. Aplikasi basis data yang sering digunakan dalam pengolahan basis data yaitu *My Structure Query Language (MySQL).*

Salah satu aplikasi basis data yang sering digunakan untuk mengolah dan menata *file-file* yaitu *MySQL*. Menurut Sadeli (2013:10) memberikan pengertian bahwa "*MySQL* adalah *database* yang menghubungkan *script php* menggunakan perintah *query* dan *escaps character* yang sama dengan *php*". Menurut Madcome (2013), "*Mysql* merupakan *databases* yang sangat popular. Beberapa keuntungan yang dimiliki *mysql* yaitu: bersifat *open source,* menggunakan bahasa *Structure*

Query Language (SQL), super *perfomance* dan *reliable,* mudah dipelajari, mampu bekerja dilintas *platform* dan *multi user*.”

2. XAMPP

XAMPP adalah aplikasi *web server* bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi Linux maupun di sistem operasi Windows.

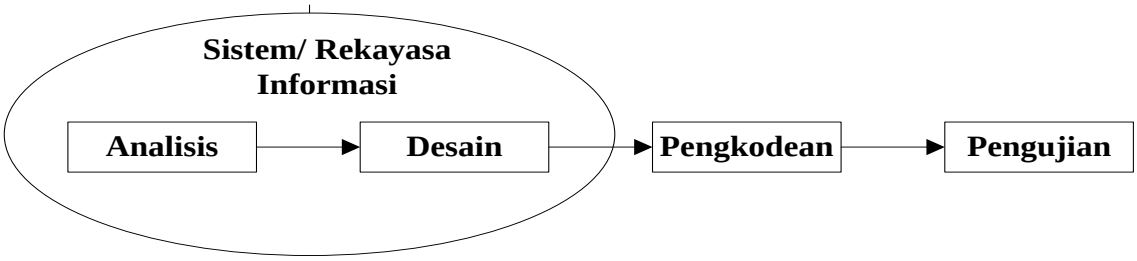


Sumber: Pratama (2014:440)

Gambar: II.1. Tampilan XAMPP

E. Model Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak ini menggunakan model *waterfall.* Menurut Rosa A.S dan M. Shalahuddin (2013:28) menjelaskan bahwa “Model *waterfall* sering juga disebut model sekuensial linier atau alur hidup klasik”.Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung.



Sumber: Rosa dan M. Shalahuddin (2013:29)

Gambar II.2. Gambar Tahapan model *waterfall*

1. Analisa 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*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

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 pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan kerepresentasi 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 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 segi *logic* 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 tahap analisis spesifikasi untuk perubahan perangkat lunak baru.

2.2. Teori Pendukung

Peralatan pendukung merupakan alat yang tepat digunakan untuk menggambarkan model logika dari suatu program, model logika dari program lebih menjelaskan dari pemakaian bagaimana nantinya fungsi-fungsi dari program secara logika akan bekerja. Adapun perlatan yang dimaksud adalah UML, Diagram UML, *Use Case Diagram*, *Activity Diagram*, *Entity Relationship Diagram* (ERD), *Logical Record Structure* (LRS), Teknik pengkodean, *Class Diagram*, dan *Sequence Diagram.*

A. *Entity Relationship Diagram (ERD)*

Model *Entity Relationship Diagram* merupakan pemodelan yang menggambarkan komponen-komponen himpunan entitas dan himpunan relasi

yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan fakta atau kejadian yang terjadi pada dunia nyata.

1. Definisi *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. Dengan menggunakan *Entity Relationship Diagram* dapat memodelkan struktur data dan hubungan antar data yang begitu kompleks, sehingga mudah untuk dibaca dan dimengerti. Pengguna *Entity Relationship Diagram* dalam pemodelan struktur data dapat mengabaikan proses yang harus dilakukan, selain itu pemodelan struktur data dengan menggunakan *Entity Relationship Diagram* dapat menjawab pertanyaan seperti, data apa saja yang kita butuhkan dan bagaimana data yang satu berhubungan dengan data yang lainnya. Adapun simbol-simbol yang digunakan dalam *Entity Relationship Diagram*:

Adapun derajat kardinalitas yang terdapat dalam *Entity Relationship Diagram* adalah:

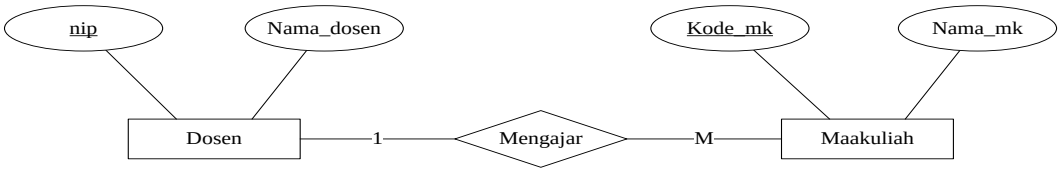
a. *One To one*

Adalah derajat kardinalitas yang menunjukkan adanya relasi himpunan entitas yang satu dengan satu entitas lainnya.

Sumber: Sukamto dan Shalahuddin (2016:166)

b. *One to many*

Adalah derajat kardinalitas yang menunjukkan adanya relasi antar himpunan entitas yang satu dengan entitas banyak entitas lainnya.



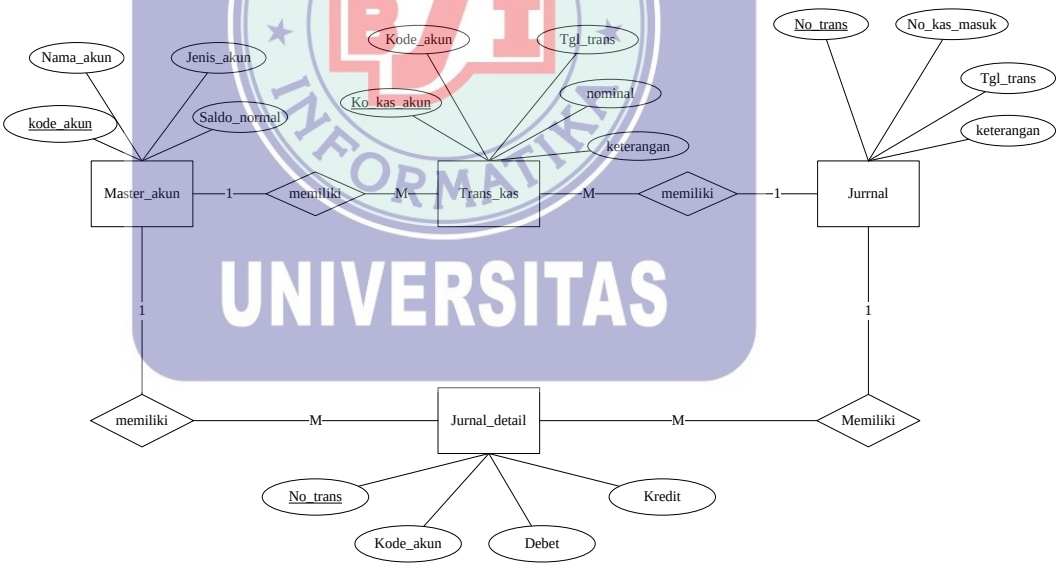
Sumber: Rosa dan Shalahuddin (2016:166)

c. *Many to many*

Adalah derajat kardinalitas yang menunjukkan adanya relasi antar himpunan banyak entitas dengan banyak entitas lainnya. Jika terjadi relasi *many to many* maka akan menghasilkan sebuah entitas baru.

Sumber: Rosa dan Shalahuddin (2016:166)

Berikut contoh dari *Entity Relationship Diagram*:



Sumber: Syara, Chintya. *Jurnal Paradigma BSI, Vol. XX, No. 1, Maret 2018.*

Gambar II.3. *Entity Relationship Diagram*

2. Komponen *Entity Relationship Diagram*

Penjelasan dari simbol-simbol yang digunakan untuk menggambarkan *entity relationship diagram (*ERD) menurut Chen dalam buku Sukamto dan Shalahudin (2015:50) :

a. Entitas /*entity*

Entitas merupakan data inti yang akan disimpan, bakal table pada data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer. Penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama table.

b. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas

c. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan, biasanya berupa id. Kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

d. Atribut multinilai / *multivalue*

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki lebih dari satu.

e. Relasi

Relasi yang menghubungkan antar entitas, diawali dengan kata kerja.

f. Asosiasi / *assocatiation*

Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki *multriplicity* kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kerdanalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B.

3. LRS (*Logical Relational Structure)*

LRS merupakan hasil transformasi diagram E-R (ERD) menggunakan aturan aturan tertentu. Aturan-aturan tersebut yaitu : (1) setiap *entity* akan diubah kedalam bentuk sebuah koyak dengan nama *entity* berada diluar koyak dan atribut berada didalam kotak, (2) sebuah relasi kadang disatukan dalam sebuah kotak bersama entity, kadang dipisah dalam sebuah kotak tersendiri(Ladjamudin, 2013:159).

Dapat disimpulkan bahwa *logical record structure (*LRS) yaitu pemodelan basis data yang merupakan hasil dari transformasi diagram E-R menggunakan aturan-aturan tertentu.

Aturan pokok yang telah diuraikan (Ladjamudin, 2013:159) mempengaruhi langkah pentransformasian yaitu kardinalitas. Adapun kardinalitas tersebut (Ladjamudin,2013:160) yaitu:

1. 1:1 (*one to one)*

Relasi yang terjadi antara suatu *entity* dengan entity lainya yang memiliki hubungan 1:1.

2. 1:M (*one to many)*

   Relasi yang terdiri antara suatu *entity* dengan *entity* lainya yang memiliki hubungan 1:M.

3. M:N (*many to many)*

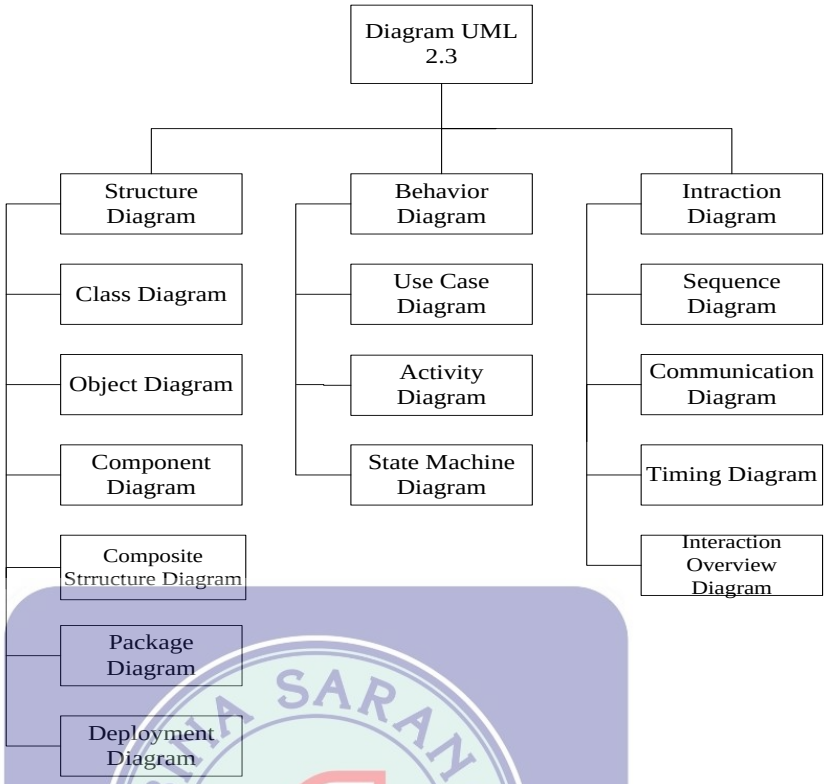
   Relasi yang terjadi antara suatu *entity* dengan *entity* lainya yang memiliki hubungan M:N. Pada relasi ini biasa digunakan table bantuan untuk memecahkan relasi tersebut menjadi 1:1 atau 1:M.

B. *Unified Modeling Language (UML)*

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifiksaikan,menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodelogi tertentu, meskipun pada kenyataanya *UML* paling banyak digunakan pada metodelogi berorientasi objek (Rosa A.S dan M.Shalahudin, 2014:133).

Pada UML 2.3 terdiri dari tiga belas (13) macam diagram yang dikelompokkan kedalam tiga kategori. Adapun pembagian kategori dan macam-macam diagram tersebut akan digambarkan pada diagram berikut.



Sumber: Rosa dan Shalahuddin (2016:140)

Gambar II.4 Diagram UML

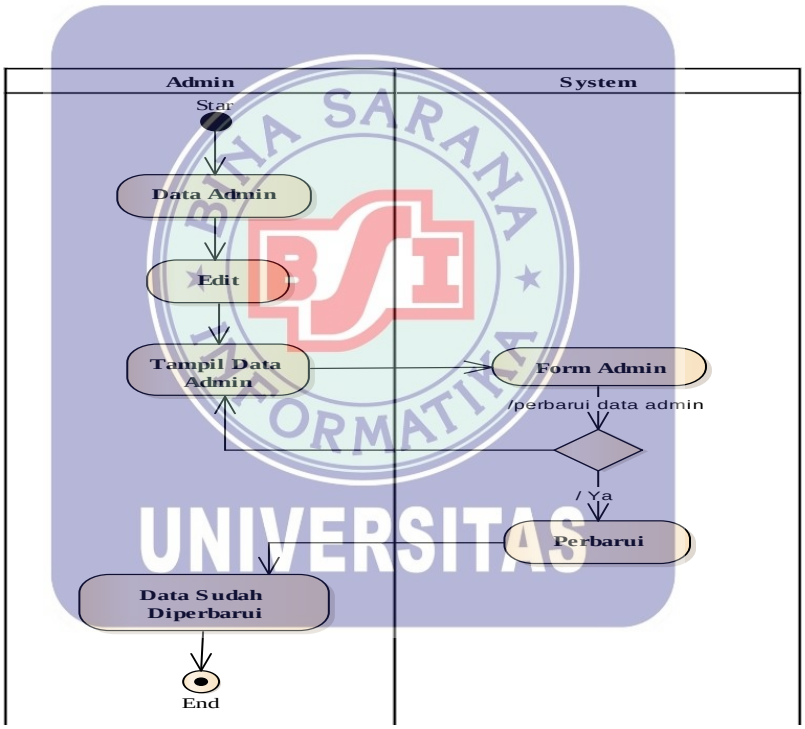
Secara garis besar penjelasan gambar diatas adalah sebagi berikut:

a. *Structure Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
b. *Behavior Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
c. *Interaction Diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sisteem lain maupun interaksi antara sub sistem pada suatu sistem.

1. *Activity Diagram*

Menurut Rosa dan Shalahuddin (2016:157) "*Activity diagram* atau diagram aktivitas adalah *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak". Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol dalam diagram aktivitas:

Berikut contoh dari *Acivity Diagram:*



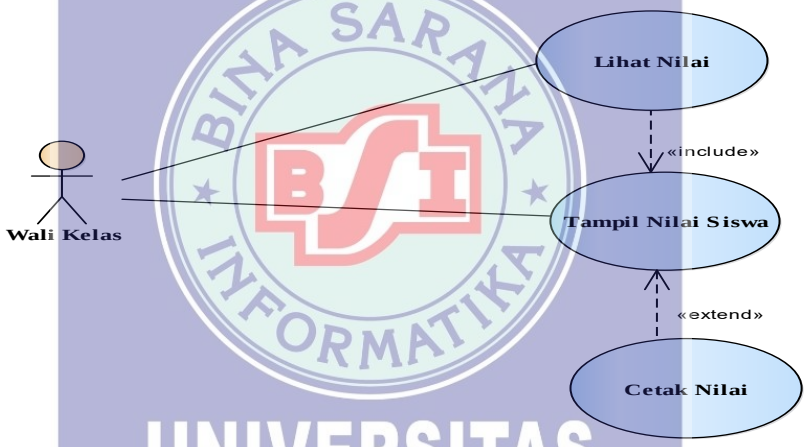
Sumber: Juniardi Dermawan dan Sri Hartini dari *Jurnal Paradigma BSI , Vol. 19, No. 2,September 2017*

Gambar II.5. Contoh *Activity Diagram*

2. *Use Case Diagram*

Menurut Rosa dan Shalahuddin (2016:155) *"Use Case Diagram* merupakan pemodelan untuk kelakuann (*behavior*) sistem informasi yang aan dibuat. *Use Case* mendeskripsi 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 didalam sebuah informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah simbol-simbol yang terdapat didalam *use case diagram*: Berikut contoh dari *Use Case Diagram:*



Sumber: Juniardi Dermawan dan Sri Hartini dari *Jurnal Paradigma BSI , Vol. 19, No. 2,September 2017*

Gambar II.6. Contoh *Use Case Diagram*

3. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem, dimana kelas ini memiliki apa yang disebut dengan atribut atau variabel-variabel yang

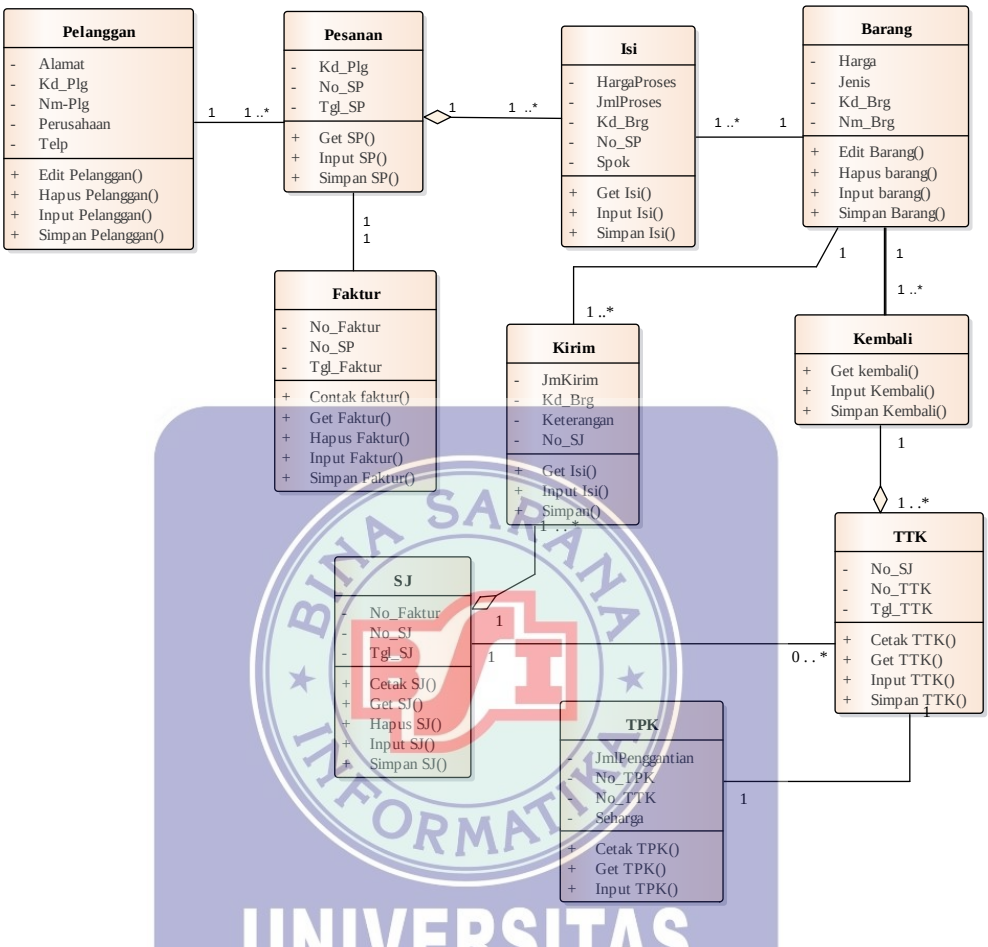
dimiliki oleh suatu kelas dan operasi atau metode yang berarti fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuat atau *programmer* membuat kelas-kelas yang sesuai dengan rancangan didalam diagram kelas sehingga terjadi kesesuaian antara dokumentasi perancangan dan perangkat lunak, dengan demikian kelas-kelas yang ada pada struktur sistem dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem tersebut. Susunan struktur kelas dalam diagram kelas terdiri dari beberapa jenis kelas sebagai berikut:

1. Kelas *Main* adalah kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas *View* adalah kelas yang menangani tampilan sistem ke pemakai.
3. Kelas *Coniroller* adalah kelas yang menangani fungsi-fungsi yang diambil dari pendefinisian *Use Case* dan menangani proses bisnis pada perangkat lunak.
4. Kelas Model adalah kelas yang digunakan untuk membungkus data menjadi sebuah kesatuan yang diambil maipun disimpan dalam basis data.

Berikut adalah simbol-simbol yang terdapat dalam kelas diagram:

Berikut contoh dari *Class Diagram:*



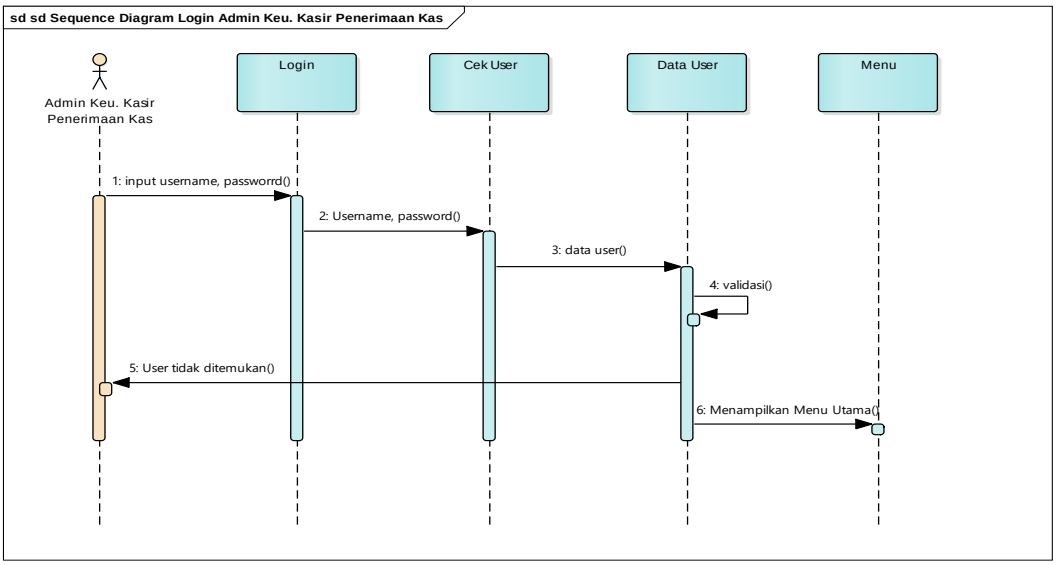
Sumber: Sunarti *Jurnal Paradigma BSI, Vol XVI no.2 September 2014*

Gambar II.7. *Class Diagram*

4. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu, dalam menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* terlebih dahulu.

Berikut contoh dari *Sequence Diagram:*



Sumber: Syara, Chintya. *Jurnal Paradigma BSI, Vol. XX, No. 1, Maret 2018.*

Gambar II.8. Contoh *Sequence Diagram*