

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

A. Metode Pembelajaran Berbasis Web

Menurut Kadir (2014:305), *website* adalah kumpulan dari halaman-halaman situs yang terangkum dalam sebuah domain atau subdomain yang tempatnya berada didalam *World Wide Web* (WWW) dalam internet. *Web* merupakan kumpulan-kumpulan dokumen yang banyak tersebar di beberapa komputer server yang berada diseluruh penjuru dunia dan terhubung menjadi satu jaringan melalui jaringan yang disebut internet.

Faktor utama yang membuat *website* begitu cepat berkembang adalah karena penyebaran informasi melalui *website* sangat cepat dan mencakup area yang luas, tidak dibatasi oleh jarak dan waktu, disamping itu saat ini juga lagi tren pembuatan *website* pribadi atau blog.

1. Kelebihan Pembelajaran Berbasis Web

- a. Memungkinkan setiap orang dimanapun, kapanpun untuk mempelajari apapun.
- b. Belajar dapat sesuai dengan karakteristik dan langkah dirinya sendiri karena pembelajaran berbasis web membuat pembelajaran menjadi bersifat individual.
- c. Kemampuan untuk membuka tautan (*link*), sehingga pembelajar dapat mengakses informasi dari berbagai sumber, baik didalam maupun diluar lingkungan belajar.

2. Kekurangan Pembelajaran Berbasis Web

- a. Keberhasilan pembelajaran berbasis web tergantung pada kemandirian dan motivasi pembelajar
- b. Akses untuk mengikuti pembelajaran dengan menggunakan web seringkali menjadi masalah bagi pembelajar.

B. Sistem

1. Pengertian Sistem

Pada dasarnya sistem adalah suatu kerangka dari prosedur-prosedur yang saling berhubungan, yang disusun sesuai dengan skema yang menyeluruh untuk melaksanakan suatu kegiatan atau fungsi utama dari perusahaan yang dihasilkan oleh suatu proses tertentu yang bertujuan untuk menyediakan informasi untuk membantu mengambil keputusan manajemen operasi perusahaan dari hari ke hari serta menyediakan informasi yang layak untuk pihak di luar perusahaan.

Pengertian Sistem yang dikemukakan oleh para ahli adalah sebagai berikut:

Menurut Susanto (2013:22) “Sistem adalah kumpulan/group dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu”.

Menurut Romney dan Steinbart (2015:3): “Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar”.

Menurut Mulyadi (2016:5), Sistem adalah “suatu jaringan prosedur yang dibuat menurut pola yang terpadu untuk melaksanakan kegiatan pokok perusahaan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2. Karakteristik Sistem

Sebuah karakteristik utama dari sebuah sistem yang berisikan data-data, harus dapat memberikan arus informasi.

Menurut Jeperson (2014:3) supaya sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu:

- a. **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.
- b. **Batasan Sistem (*boundary*)** : 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.
- c. **Lingkungan luar sistem (*environment*)** : adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung sistem (*interface*) : merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan 8 sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

e. Masukan sistem (*input*) : adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintance input*), dan masukan sinyal (*signal input*).

f. Keluaran sistem (*output*) : hasil dari energi yang di olah dan diklasifikasikan menjadi keluararan yang berguna dan sisa pembuangan. Sedangkan informasi adalah keluaran yang dibutuhkan.

g. 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. Sasaran sistem : pasti mempunyai tujuan (*goal*) atau sasaran (*objective*).

3. Tujuan Sistem

Adapun tujuan sistem menurut Susanto (2013:23) “Target atau sasaran akhir yang ingin dicapai oleh sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai sasaran tanpa mengetahui ciri- ciri atau kriteria dari sasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian”.

C. Website

1. Pengertian Website

Menurut Supriyanta dalam Raharjo (2015:2) “*World Wide Web (WWW)*, sering disingkat dengan web, adalah suatu layanan di dalam jaringan yang berupa ruang informasi”.

Menurut Sibero (2013:11) “*website* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet”.

Berdasarkan pendapat para ahli diatas dapat disimpulkan *website* adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang 9 mengandung informasi atau kumpulan halaman yang digunakan untuk menampilkan informasi teks, gambar, animasi, suara dan gabungan dari semua baik yang bersifat statis maupun dinamis.

Website dibagi menjadi 2 yaitu *web server* dan *web browser*.

a. *Web Server*

Menurut Sadeli (2013:2), *Web server* adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

Web server merupakan program aplikasi yang memiliki fungsi sebagai tempat penyimpanan dokumen yang terdapat didalam web. Dokumen web yang ditulis menggunakan metode *client-side scripting* atau *server-side scripting* disimpan di direktori utama *web server (document root)*. Contoh aplikasi web

server yang umum digunakan saat ini ialah *Apache*, *Web server apache tomcat*, *Web server microsoft internet information*.

b. *Web Browser*

Browser merupakan *software* atau alat yang digunakan untuk menjelajahi internet. Pengertian *browser* sejalan dengan adanya istilah “*browse*”.

Menurut Arief (2016:19), *web browser* merupakan program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML. *Web browser* dibuat menggunakan standarisasi yang dibuat oleh *World Wide Web Consortium* (W3C) yang merupakan badan tunggal yang mengurus semua hal yang berkaitan dengan web. Berikut Contoh *Web browser* yang cukup populer beserta *web engine* yang di gunakan :

- 1) *Web Engine WebKit: Safari, Google Chrome*
- 2) *Web Engine Trident: Microsoft Internet Exploler*
- 3) *Web Engine Gecko: Mozila Firefox*
- 4) *Web Engine Presto: Opera*

2. Internet

Menurut Ahmadi dan Hermawan (2013:68), Internet adalah komunikasi jaringan komunikasi global yang menghubungkan seluruh komputer di dunia meskipun berbeda sistem operasi dan mesin.

Jaringan internet bekerja berdasarkan suatu protocol atau aturan yang telah ditetapkan. TCP/IP (*Transmission Control Protocol Internet Protocol*) adalah protocol yang umum digunakan agar dapat menghubungkan suatu komputer dengan komputer lainnya di mana setiap komputer diberikan nomor unik yang disebut nomor IP.

3. Xampp

Menurut Supriyanta dalam Yudhanto dan Agus Purbaya (2014:11) “*XAMPP* merupakan program paket PHP dan MySQL berbasis *opensource* yang saat ini merupakan andalan para programmer PHP dalam melakukan *programming* dan melakukan testing hasil programnya”. Sedangkan menurut Sadeli (2013:4) “*Xampp* adalah program yang berisi paket *Apache*, *MySQL*, dan *phpMyAdmin*.”

a. Apache

Menurut Sadeli (2013:2) “*Apache (Server HTTP Apache* atau *ServerWeb/WWW Apache)* adalah *webserver* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web”.

b. MySQL

Menurut Fridayanthie dalam Arief (2016:151) “*MySQL (My Structure Query Language)* adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya”. *MySQL* bersifat *open source* dan menggunakan *SQL (Structured Query Language)*. *MySQL* biasa dijalankan diberbagai platform misalnya windows Linux, dan lain sebagainya.

c. PhpMyAdmin

PhpMyAdmin Menurut Rahman (2013:21) “*PHPMYADMIN* adalah sebuah *software* berbasis pemrograman PHP yang dipergunakan sebagai administrator MySQL melalui browser (web) yang digunakan untuk *management database*”.

4. PHP (*Perl Hypertext Preprocessor*)

Menurut Hidayatullah dan Kawistara (2014:231), “PHP singkatan dari *Perl Hypertext Preprocessor* yaitu bahasa pemrograman web *server-side* yang bersifat *open source*. PHP merupakan script yang berintegrasi dengan HTML dan berada pada server (*server side HTML embedded scripting*)”. PHP adalah *script* yang digunakan untuk membuat halaman web dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan.

PHP dirancang untuk dapat bekerja sama dengan *database server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses *database* menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi dimana aplikasi tersebut dibangun oleh PHP pada umumnya akan memberikan hasil kepada *web browser*, tetapi proses keseluruhannya dijalankan di server.

Adapun kelebihan-kelebihan dari PHP yaitu:

- a. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
- b. PHP dapat berjalan pada web server yang dirilis oleh *Microsoft*, seperti IIS atau PWS juga pada apache yang bersifat *open source*.
- c. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.

- d. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
- e. PHP dapat berjalan pada 3 (tiga) *operating system*, yaitu: Linux, unix, dan windows, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

5. Adobe Dreamweaver

Menurut Sibero, dikutip dalam Utomo & Bakara (2013) *Adobe Dreamwaver* adalah suatu produk Web Developer yang dikembangkan oleh *Adobe System Inc.* Sebelumnya produk *Dreamweaver* dikembangkan oleh *Macromedia Inc.* Yang kemudian sampai saat ini pengembangannya diteruskan oleh *Adobe System Inc.* Setelah diambil oleh *Adobe Systems Inc.*, *dreamweaver* dikembangkan dan dirilis dengan kode nama *Creative Suit (CS)*. Ruang kerja atau *Workspace* adalah bagian keseluruhan tampilan *Adobe dreamweaver* yang terdiri dari *Welcome Screen*, *Menu*, *Insert bar*, *Document Window*, *CSS Panel*, *Application Panel*, *Tag Inspector* *Property Inspector*, *Result Panel*, dan *Files Panel*, masing-masing dari komponen tersebut memiliki fungsi dan aturan. *Adobe dreamweaver* merupakan *web editor* yaitu program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen web baik *client side scripting* maupun *server side scripting*. Saai ini banyak tersedia web editor mulai dari yang paling sederhana hingga yang lebih smart.

D. Basis Data

1. Pengertian Basis Data

Menurut Lubis (2016:2) Basis Data merupakan gabungan file data yang dibentuk dengan hubungan / relasi yang logis dan dapat diungkapkan dengan

catatan serta bersifat independen. Basis data merupakan perangkat lunak yang digunakan untuk mengelola data dan memiliki penjelasan yang tersusun didalamnya.

Menurut Ladjamudin (2013:129), “*Database* adalah sekumpulan *data store* (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk*, *optical disk*, *magnetic drum*, atau media penyimpanan sekunder lainnya”.

Menurut Sukamto dan M. Shalahudin (2014:43), “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Dari pengertian diatas penulis menyimpulkan *Database* adalah sekumpulan file yang saling berhubungan yang menyimpan data dan tersimpan dalam sebuah media penyimpanan.

2. Database Management System

Menurut Sukamto dan M. Shalahudin (2014:45) DBMS (*Database Management System*) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data.

Semua DBMS memiliki suatu pengolah bahasa deskripsi data (*data description language processor*) yang digunakan untuk menciptakan *database* serta suatu pengelola *database* yang menyediakan isi *database* bagi pemakai. Pemakai menggunakan manipulasi data dan *query language*. Orang yang bertanggung jawab atas *database* dan DBMS adalah pengelola *database* (*database administrator*) atau disingkat DBA. DBMS menyediakan keuntungan yang nyata bagi perusahaan yang menggunakan komputer mereka sebagai suatu sistem

informasi. Mengetahui dengan jelas arti, batasan database dan sistem database adalah penting.

E. Model Pengembangan Perangkat Lunak

Sebuah pengembangan sistem akan melakukan aktivitas untuk menghasilkan sistem informasi dan menyelesaikan masalah.

Menurut Sukamto dan M. Shalahudin (2014:67), bahwa Pemrograman terstruktur adalah konsep atau paradigm atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program computer. Modul-modul (pembagian program) biasanya dibuat dengan mengelompokkan fungsi-fungsi dan prosedur-prosedur yang diperlukan sebuah proses tertentu.

Aktivitas-aktivitas dalam metodologi *waterfall* adalah:

1. Analisis Kebutuhan Sistem

Penulis menganalisa data yang telah diperoleh untuk mengembangkan sistem yang ada dengan tujuan memperoleh hasil yang lebih baik. Pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.

2. Desain

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat pengkodean.

3. Code Generation

Pada tahap ini penulis akan mulai mencoba membuatkan *code generation* sesuai desain yang telah ada. Pembuatan *code generation* menggunakan bahasa pemrograman HTML, PHP dan *Javascript*.

4. Testing

Pada tahap ini penulis akan mulai mencoba menerapkan web informasi akademik dan pengolahan data akademik tersebut dengan melakukan tes terlebih dahulu sebelum hasil perancangan dari penelitian tersebut benar-benar diterapkan.

2.2 Teori Pendukung

A. Entity Relationship Diagram dan Logical Record Structure

1. Entity Relationship Diagram

Menurut Sukamto dan M. Shalahuddin (2014) "Entity Relation Diagram adalah "bentuk paling awal dalam melakukan perancangan basis data relasional, Entity Relation Diagram memiliki beberapa aliran notasi seperti notasi Chen, Barker, notasi Crow's Foot dan beberapa notasi lainnya".

Entity Relationship Diagram adalah suatu model jaringan (*network*) yang menggunakan susunan data yang disimpan dari sistem secara abstrak. Tujuan utama penggambaran ERD adalah untuk menunjukkan objek-objek data (*entity*) dan *relationship-relationship* yang ada pada objek-objek tersebut.

Dalam ERD untuk memodelkan struktur data dan hubungan antar data digunakan beberapa notasi dan simbol. Terdapat tiga simbol yaitu :

1. Entitas

Entitas adalah objek dalam dunia nyata yang dapat dibedakan dengan objek lain. Ada dua jenis macam entitas, yaitu:

- a. *Strong Entity Type* adalah entitas yang keberadaanya tidak tergantung pada entitas lain. Terkadang disebut *parent, owner dominant*.
- b. *Weak Entity Type* adalah entitas yang keberadaanya bergantung pada entitas lain. Disebut juga *child dependent, subordinate*.

2. Atribut

Atribut adalah ciri-ciri kualitatif yang dimiliki oleh suatu objek yang mencerminkan suatu objek tersebut. Setiap entitas pasti mempunyai elemen yang disebut atribut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain.

Ada dua jenis atribut, yaitu:

- a. *Identifier (key)* digunakan untuk menentukan suatu *entity* secara unik (*primary key*)
- b. *Descriptor (nonkey attribute)* digunakan untuk menspesifikasikan karakteristik dari suatu *entity* yang tidak unik.

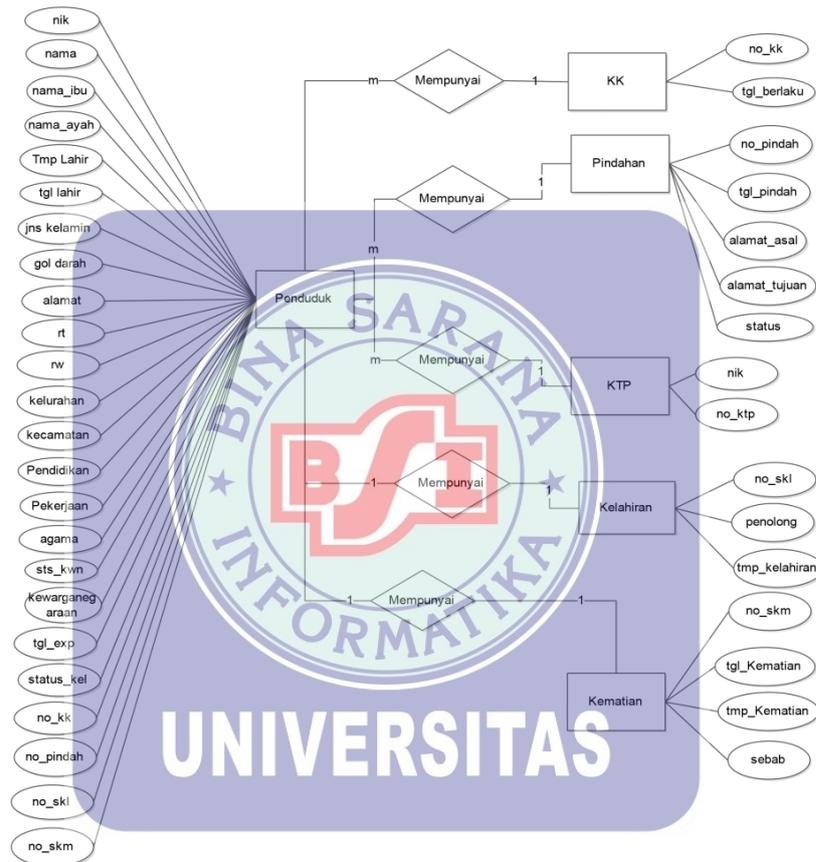
3. Relasi atau hubungan

Relasi adalah hubungan antara beberapa entitas. Hubungan tersebut dapat merepresentasikan kejadian yang menghubungkan entitas atau hanya persamaan logika yang terdapat diantara entitas. Setiap relasi antar entitas memiliki sebuah derajat relasi atau kardinalitas. Kardinalitas ialah jumlah minimum dan maksimum kemunculan suatu entitas yang dihubungkan dengan kemunculan tunggal entitas lain.

Setiap relasi antar entitas memiliki sebuah derajat relasi atau kardinalitas. Kardinalitas ialah jumlah minimum dan maksimum kemunculan suatu entitas

yang dihubungkan dengan kemunculan tunggal entitas lain. Jenis-jenis kardinalitas antara lain:

- Satu ke Satu (*One to One*)
- Satu ke Banyak (*One to Many*)
- Banyak ke Banyak (*Many to Many*)



Sumber: Supriyanto (2015:17)

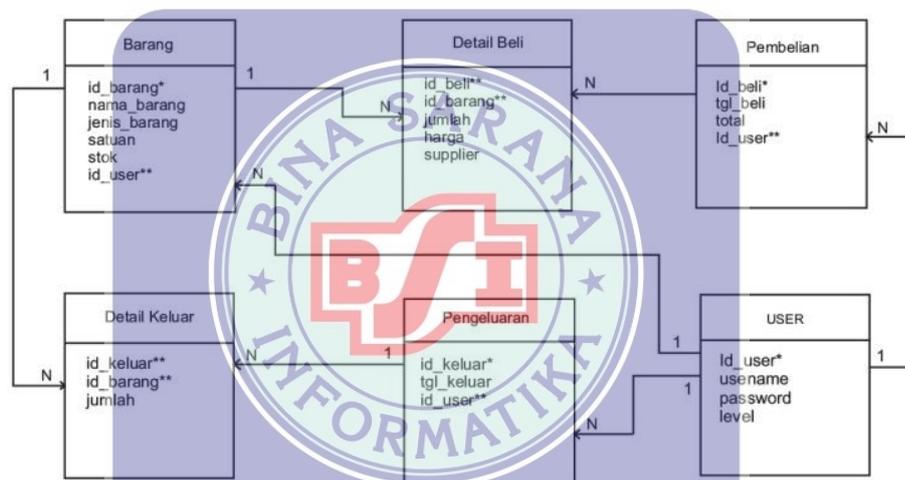
Gambar II.1 *Entity Relationship Diagram* Administasi Data Kependudukan

2. Logical Record Structure

Menurut Fridayanthie dalam Simarmata dan Paryudi (2016:115), “*Logical Record Structured (LRS)* adalah representasi dari struktur *record - record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah tabel, dan *Foreign Key (FK)*.

Berikut adalah cara membentuk skema database atau LRS (Logical Record Structured) berdasarkan Entity Relationship Diagram:

- Jika relasinya satu-ke-satu, maka foreign key diletakan pada salah satu dari dua entitas yang ada tau menyatukan kedua entitas tersebut.
- Jika relasinya satu-ke-banyak, maka foreign key diletakan pada entitas Many.
- Jika relasinya banyak-ke-banyak, maka dibua “file konektor” yang berisi dua foreign key yang berasal dari kedua entitas.



Saputra (2017:65)

Gambar II.2 Logical Record Structure Pendataan Bahan Bangunan

B. Unified Modeling Language

1. Pengertian Unified Modeling Language

Menurut Sukamto dan M. Shalahuddin (2014:133) “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.

UML menyediakan sekumpulan tata cara terbaik dalam proses perancangan suatu sistem yang luas dan kompleks. UML memiliki notasi yang terdefinisi dengan baik sehingga dapat memungkinkan bagi perancang sistem peranti lunak untuk menggambarkan skenario atau merumuskan arsitektur sistem sehingga dapat dikomunikasikan dengan baik kepada orang lain dalam mengambil keputusan.

2. Use Case Diagram

Menurut Fridayanthie dalam Sukanto dan M. Shalahuddin (2014:156), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Use case diagram memperlihatkan hubungan antara aktor (*Actors*) dengan sistem (*Case*). *Use case diagram* memperlihatkan hubungan antara aktor (*Actors*) dengan sistem (*Case*). Aktor menggambarkan user yang akan berinteraksi dengan sistem yang dibuat.

Use case adalah beberapa aksi atau tugas yang dilakukan user yang dapat dilakukan untuk menyelesaikan suatu masalah.

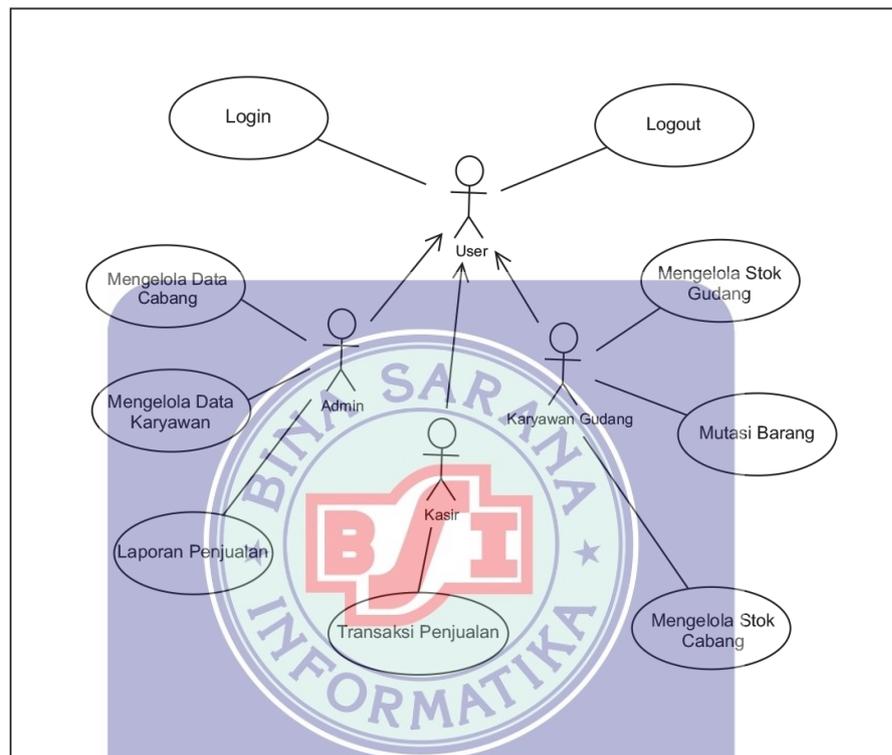
Diagram *use case* dapat menggunakan include dan extend yang digunakan untuk menjadikan diagram *use case* lebih baik dan mencegah penyalahgunaan antara aktor dan case.

1. << include >>

Hubungan *Include* digunakan untuk menggambarkan kebutuhan suatu *use case* terhadap *use case* lainnya. Penggunaan *include* bertujuan untuk mengurangi pengulangan *use case*.

2. << extend >>

Extend digunakan untuk menggambarkan suatu *use case* merupakan perluasan dari *use case* lainnya. *Extend* digunakan jika sebuah *use case* (*base use case*) memiliki perilaku *use case* lain (*extending use case*).



Ade Hendini (2016:6)

Gambar II.3 Use Case Diagram Monitoring Penjualan dan Stok Barang

3. Class Diagram

Menurut Sukanto dan M. Shalahudin (2014:33), *Class Diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class di dalam sebuah *class diagram* dapat digabungkan dengan *class* lainnya. Metode penggabungan *class* dapat dilakukan dengan metode *Association*, *Generalization*, *Aggregation*, dan *Compotition*.

1. Association

Association merupakan suatu metode penghubungan *class diagram* dengan menunjukkan hubungan antar dua *class* serta sifat dari hubungan tersebut. *Association* dapat diimplementasikan dengan menggunakan beberapa syntax yaitu:

- a. 1 Hanya Satu
- b. * Tidak terhingga (nol hingga tak terhingga)
- c. 0..* Nol atau banyak
- d. 1..* Satu atau banyak
- e. 0..1 Nol atau satu
- f. 3..7 Jarak yang lebih spesifik sesuai kebutuhan

2. Generalization

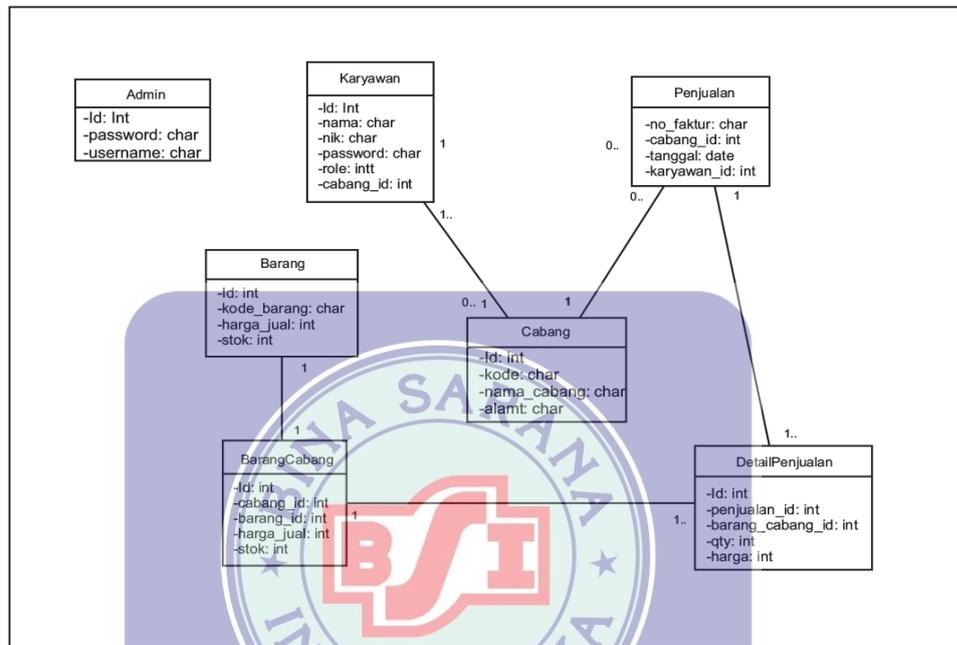
Generalization menunjukkan sebuah generalisasi dari hubungan *class* dimana pada metode ini digambarkan dengan panah tertutup. Panah mengarah ke *superclass*, sebaliknya *class* yang menurunkan *superclass* ialah *subclass*. *Subclass* mewarisi struktur dan perilaku *superclass* di mana setiap *subclass* memiliki satu atau banyak sifat warisan dari *superclass*.

3. Aggregation

Hubungan *aggregation* digambarkan dengan menggunakan lambang *diamond* yang tidak terisi. *Diamond* akan menunjukkan bahwa yang menjadi titik akhir ialah agregat sedangkan disisi lain ialah bagian dari agregat itu sendiri.

4. Composition

Hubungan *composition* digunakan apabila diinginkan hasil yang bersifat lebih detail. *Composition* digambarkan dengan diamond yang terisi, di mana titik akhir tujuan menggambarkan agregat.

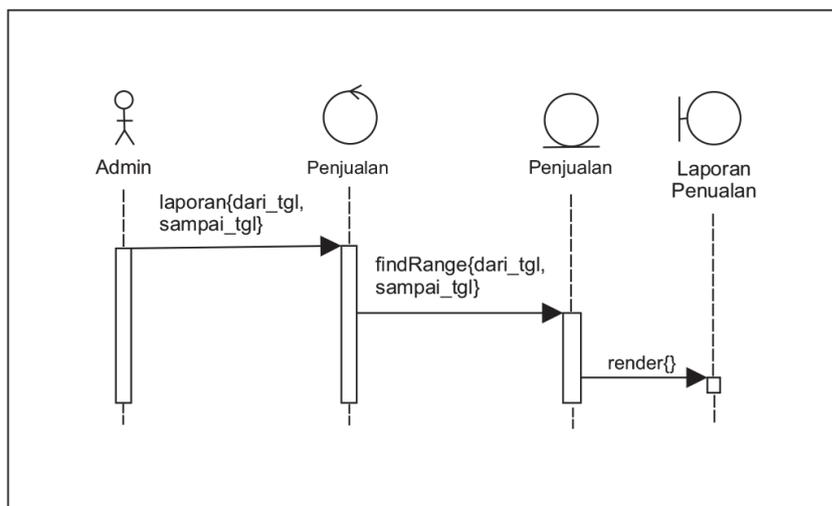


Ade Hendini (2016:9)

Gambar II.4 *Class Diagram* Monitoring Penjualan dan Stok Barang

4. Sequence Diagram

Menurut Sukamto dan M. Shalahudin (2014:165), *Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.



Ade Hendini (2016:8)

Gambar II.5 *Sequence Diagram* Laporan Penjualan

5. Activity Diagram

Menurut Fridayanthie dalam Sukanto dan M. Shalahuddin (2014:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. *Activity Diagram* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Diagram ini fokus terhadap kegiatan yang dilakukan dan siapa yang bertanggung jawab dengan kinerja kegiatan tersebut. Terdapat beberapa proses dan simbol-simbol yang umum di gunakan di dalam *Activity diagram*, yaitu :

1. Action State

Merupakan unsur pelaku dalam *Activity diagram* dimana kegiatan dapat memiliki banyak tindakan yang menunjukkan jalannya suatu sistem.

2. Initial dan Final Nodes

Initial nodes merupakan aktifitas awal dari suatu sistem sedangkan *Final nodes* merupakan hasil akhir dari suatu sistem.

3. *Decision dan Merge Node*

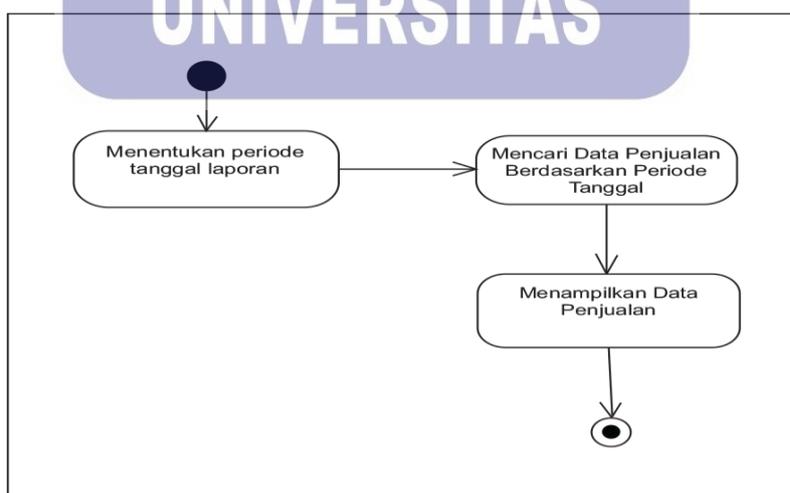
Decision dan Merge node mengendalikan aliran pada *Activity diagram* dimana setiap node di gambarkan berbentuk berlian dengan panah masuk dan keluar. *Decision node* memiliki satu arah panah masuk dan beberapa panah keluar.

4. Partisi

Dalam suatu *Activity diagram* dapat di kelompokkan dengan menggunakan partisi. Tujuannya ialah untuk menunjukkan dimana tanggung jawab yang harus dilakukan dalam suatu kegiatan.

5. *Forks, Joins, dan Concurrency*

Fork dan Join nodes di analogikan dengan *Decision dan Merge node* dimana *fork* memiliki satu aliran masuk dan beberapa arus keluar sedangkan *Join* memiliki beberapa aliran masuk dan memiliki satu aliran keluar, tetapi dalam *Join* seluruh aliran yang masuk harus diselesaikan terlebih dahulu sebelum aliran keluar dimulai.



Ade Hendini (2016:7)

Gambar II.6 *Activity Diagram* Laporan Penjualan