BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Koperasi

Menurut Ibid dalam Widiyanti (2010:18) mengemukakan bahwa "Koperasi merupakan bentuk usaha bersama, maka pilihan usaha koperasi itu ditentukan oleh kepentingan usaha atau mata pencaharian anggotanya. Koperasi bukan koperasi jika usahanya ditentukan berdasarkan besarnya untung yang akan diperoleh tanpa ada kaitannya dengan usaha anggotanya atau meningkatkan daya beli anggotanya".

Menurut Muslimin Nasution dalam Widiyanti (2010:19), koperasi haruslah ditinjau dari tiga karakteristik, yaitu :

1. Koperasi sebagai suatu proses, karena pembangunan koperasi adalah rentetan perubahan kearah pertumbuhan dan perkembangan.
2. Koperasi sebagai suatu metode, sebab peembangunan koperasi menempuh cara-cara yang terencana diatas disiplin keteraturan dan kesinambungan, sesuai dengan asas swakerta, swadaya, dan swasembada.
3. Koperasi sebagai suatu program, karena pembangunan koperasi merupakan paduan dari berbagai kegiatan dalam bidang kehidupan yang menyentuh kepentingan masyarakat kecil, baik di daerah perkotaan maupun pedesaan.

Tujuan koperasi sebagaimana dikemukakan dalam pasal 3 UU No. 25 tahun 1992 adalah sebagai berikut : "Koperasi bertujuan memajukan

kesejahteraan anggota pada khususnya dan masyarakat pada umumnya serta ikut membangun tatanan perekonomian nasional dalam rangka mewujudkan masyarakat yang maju, adil, dan makmur berdasarkan Pancasila dan UUD 1945". Landasan koperasi Indonesia merupakan pedoman dalam menentukan arah, tujuan, peran serta kedudukan koperasi terhadap pelaku – pelaku ekonomi lainnya di dalam sistem perekonomian Indonesia.

Koperasi digolongkan berdasar daerah kerja diantaranya adalah sebagai berikut:

1) Koperasi primer adalah koperasi yang beranggotakan orang yang biasanya didirikan pada lingkup wilayah terkecil tertentu.
2) Koperasi pusat adalah koperasi yang beranggotakan koperasi – koperasi primer biasanya didirikan sebagai pemusatan dari berbagai koperasi primer dalam lingkup wilayah tertentu. Koperasi pusat mempunyai tujuan untuk memperkuat kedaulatan ekonomi koperasi – koperasi yang bergabung di dalamnya.
3) Koperasi gabungan hampir sama dengan koperasi pusat, koperasi gabungan tidak beranggotakan orang – orang, melainkan beranggotakan koperasi – koperasi pusat yang berasal dari wilayah tertentu. Tujuan pembentukannya adalah untuk memperkuat kedudukan koperasi – koperasi yang bergabung di dalamnya, di dalam wilayah kerja yang lebih luas.
4) Koperasi induk ialah koperasi yang beranggotakan berbagai koperasi pusat atau koperasi – koperasi gabungan yang berkedudukan di ibukota negara. Fungsinya ialah sebagai penyambung lidah koperasi – koperasi yang

menjadi anggotanya dalam berhubungan dengan lembaga nasional yang terkait dengan pembinaan koperasi – koperasi sejenis di negara lain ataupun organisasi – organisasi pengusaha pada tingkat nasional dan internasional.

2.1.2. Konsep Dasar Sistem Informasi

A. Konsep Informasi

Informasi mempunyai manfaat dan mempunyai peranan yang sangat dominan dalam suatu organisasi atau perusahaan. Tanpa tersedianya informasi para manajer tidak dapat mengambil keputusan dengan cepat dan mencapai tujuan dengan efektif dan efisien.

Menurut Gaol (2008:7) memberikan batasan bahwa "Informasi adalah segala sesuatu keterangan yang bermanfaat untuk para pengambil keputusan atau manajer dalam rangka mencapai tujuan organisasi yang sudah ditetapkan sebelumnya".

Menurut Kenneth C. Laudon dalam Gaol (2008:8) "Informasi adalah data yang sudah kedalam sebuah formulir bentuk yang bermanfaat dan dapat digunakan untuk manusia".

B. Konsep Sistem

Pengertian sistem menurut Gaol (2008:9) "Sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lainnya dan yang tidak dapat dipisahkan serta menuju satu kesatuan dalam rangka mencapai tujuan yang telah ditetapkan".

James A. O'Brien dalam Gaol (2008:10) mengatakan bahwa sistem adalah:

a. Sekelompok unsur yang berkaitan atau berhubungan untuk membentuk satu kesatuan yang utuh.
b. Sekelompok unsur yang Saling bekerja sama untuk menuju pada tujuan bersama dengan menerima masukkan dan menghasilkan keluaran dalam sebuah proses perubahan yang dikoordinasi.
c. Suatu penyusunan metode atau cara, tata cara, atau teknik yang disatukan melalui hubungan yang diatur untuk membentuk satu kesatuan yang utuh.
d. Sekumpulan orang, mesin, atau metode yang diperlukan untuk mencapai susunan fungsi yang khusus.

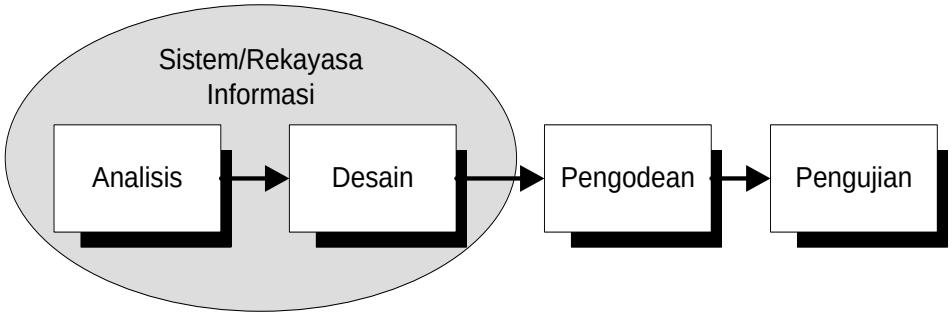
C. Konsep Sistem Informasi

Sebuah gagasan sistem informasi tidak dikenal sebelum munculnya komputer. Akan tetapi komputer telah banyak memberikan dampak dengan terwujudnya gagasan tersebut menjadi realitas. Dampaknya, sebuah sistem informasi berdasarkan komputer akan mengalami perbedaan dengan sistem-sistem yang diolah secara manual.

Menurut James A. O'Brien dalam Gaol (2008:17) "Sistem informasi adalah sebuah perpaduan atau gabungan orang-orang , perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya sumber daya yang mengumpulkan, mengubah, dan menyebarkan informasi pada sebuah organisasi".

2.1.3. Model *Waterfall*

Software development life cycle atau sering disebut juga *system development life cycle* (SDLC) adalah proses mengembangkan atau mengubah sistem perangkat lunak dengan menggunakan model dan metodologi yang digunakan untuk mengembangkan sistem perangkat lunak sebelumnya berdasarkan cara-cara yang sudah teruji dengan baik (Sukamto dan Shalahuddin, 2013:26). Salah satu model SDLC adalah air terjun *(waterfall)* yang sering juga disebut model sekuensial linier *(linier sequential)* atau alur hidup klasik *(classic life cycle)*. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung *(support)*. Berikut adalah gambar model air terjun:



Sumber : Sukamto dan Shalahuddin (2013:29)
Gambar II.1. Ilustrasi Pemodelan *Waterfall*

Berikut adalah penjelasan tahapan-tahapan yang ada dalam model *waterfall* menurut Sukamto dan Shalahuddin (2013:29) :

1. Analisis Kebutuhan Perangkat Lunak

   Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan kebutuhan perangkat lunak agar dapat dipahami seperti apa yang di butuhkan oleh *user.*

2. Desain

   Desain perangkat lunak merupakan proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean.

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 meminimalkan kesalahan 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.

Menurut Ladjamudin (2006:16) menyatakan bahwa “Model ini telah diperoleh dari proses *engineering* lainnya. Model ini menawarkan cara pembuatan perangkat lunak secara lebih nyata”.

Langkah – langkah yang penting dalam kondisi ini menurut Ladjamudin (2006:16) adalah :

1. Penentuan dan spesifikasi

   Jasa, kendala dan tujuan dihasilkan dari konsultasi dengan penggunaan sistem. Kemudian semuanya itu dibuat dalam bentuk yang dapat dimengerti oleh user dan staf pengembang.

2. Desain sistem dan perangkat lunak

   Proses desain sistem membagi kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluruhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin di transformasi kedalam satu atau lebih program yang dapat dijalankan.

3. Implementasi dan ujicoba unit

   Selama tahap ini desain perangkat lunak didasari sebagai sebuah program lengkap atau unit program. Uji unit termasuk pengujian bahwa setiap unit sesuai spesifikasi.

4. Integrasi dan uji coba sistem

   Unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah terpenuhi. Setelah ujicoba, sistem disampaikan ke *customer*.

5. Operasi dan pemeliharaan

   Normalnya, ini adalah fase yang terpanjang. Sistem dipasang dan digunakan. Pemeliharaan termasuk pembetulan kesalahan yang tidak

ditemukan pada langkah sebelumnya. Perbaikkan implementasi unit sistem dan peningkatan jasa sistem kebutuhan baru ditemukan.

2.1.4. Konsep Dasar Pemograman

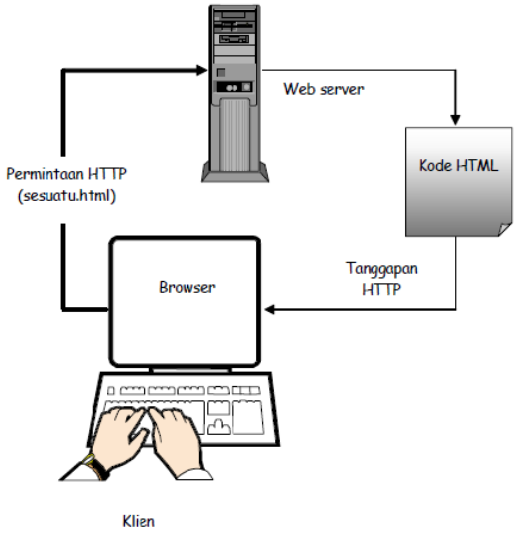
Bahasa pemrograman merupakan sarana komunikasi yang menjembatani antara manusia dan komputer. Bahasa pemrograman dikelompokkan menjadi dua kelompok besar yaitu bahasa pemrograman tingkat rendah atau *Low Level Language* dan bahasa pemrograman tingkat tinggi atau *High Level Language*. Proses pemrograman komputer atau biasa disebut *programming* bukan hanya menulis suatu urutan instruksi, tetapi bertujuan untuk memecahkan suatu masalah serta mempermudah pekerjaan atau yang lainnya, yang diinginkan oleh pemakai atau *user*. Adapun beberapa teori yang digunakan dalam pembuatan skripsi ini adalah :

A. *WEB*

Menurut Winarno dkk (2014:01) “Web adalah media informasi global yang bisa dipakai oleh *user*-nya untuk saling bertukar informasi, dan sekarang bahkan berfungsi melebar dari mulai sosialisasi hingga transaksi.”

1. Pengertian *Web Browser*

   Menurut Suyanto (2009:71) ”*Browser* adalah sebuah *software* yang digunakan untuk menampilkan halaman web”. *Browser* berkomunikasi dengan *web server* melalui protokol HTTP, yang membaca dan menerjemahkan bahasa HTML dan data lainnya dan kemudian menampilkannya secara visual sehingga informasi yang ada dapat dibaca.



Sumber : Suyanto (2009:71)

Gambar II.2. Mekanisme kerja Permintaan Dokumen Kepada Web Statis

2. Pengertian HTML (*Hyper Text Markup Language*)

Menurut Suyanto (2009:82) ”HTML merupakan bahasa yang digunakan untuk menulis halaman *web.* Biasanya mempunyai ekstensi .htm, .html, atau .shtml. HTML tersusun atas tag-tag digunakan untuk menentukan tampilan dari dokumen HTML yang diterjemahkan oleh *web browser*”.

HTML dapat dibaca oleh berbagai macam *platform.* HTML merupakan bahasa pemograman yang fleksibel dapat disisipi atau digabungkan dengan bahasa pemograman lain seperti PHP, ASP, JSP, *JavaScript*, dan lainnya.

a) HTML

Setiap dokumen HTML biasanya diawali dan ditutup dengan *tag* HTML, yang memberitahu browser bahwa yang berada didalam kedua *tag* tersebut adalah dokumen HTML.

b) *HEAD*

Bagian *header* dari dokumen HTML, berada di antara *tag HEAD*. Didalam bagian ini biasanya dimuat *tag TITLE* yang menampilkan judul halaman pada bagian *title* milik *browser*.

c) *BODY*

Dokumen *body* digunakan untuk menampilkan *text, image, link*, dan semua yang akan ditampilkan pada halaman *web*.

B. *Hypertext Preprocessor (PHP)*

Pengertian php menurut Madcoms (2011:216) ”PHP adalah salah satu bahasa pemograman yang berjalan dalam sebuah webserver dan berfungsi sebagai pengolah data pada sebuah server”.

Prasetio (2012:122) “*PHP (Hypertext Preprocessor)* adalah bahasa *script* yang di tanam di sisi *server”*. Sedangkan menurut Anhar (2008:3) “*PHP* adalah bahasa pemrograman *web server-side* yang bersifat *open source*. *PHP* merupakan *script* yang terintegrasi dengan *HTML* dan berada pada *server (server side HTML embedded scripting)”.*

PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminat oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script PHP* dieksekusi pada *server* di mana *script* tersebut di jalankan.

C. *Casecading Style Sheet (CSS)*

Menurut Suyanto (2009:84) "CSS banyak digunakan untuk memperluas kemampuan HTML dalam memformat dokumen *web* atau untuk mempercantik tampilan *web,* bahkan untuk pemosisian dan *layouting* halaman *web".*

D. *Java Script*

Menurut Suyanto (2009:85) "*Java Script* adalah bahasa skrip yang ditempatkan pada kode HTML dan diproses pada sisi klien. Dengan adanya bahasa ini maka kemampuan dokumen HTML menjadi lebih luas."

E. *MySQL Database*

Menurut Anhar (2008:21) "*MySQL (My Structure Query Language)* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, *MySQL*, PostagreSQL, dan lain-lain. multi-user yang bersifat gratis di bawah lisensi GNU *General Puplic Licenci* (GPL)"

F. *Adobe Dremweaver*

AdobeDreamweaver dalam Madcoms (2008:1) adalah "sebuah HTML editor profesional yang berfungsi mendesain secara visual dan mengelola situs *web* maupun halaman *web".* Untuk berurusan dengan kode-kode HTML secara manual atau lebih menyukai bekerja dengan lingkungan secara visual dalam melakukan *editing, Dreamweaver* mambuatnya menjadi lebih mudah dengan menyediakan *tool-tool* yang sangat berguna dalam peningkatan kemampuan dalam mendesain *web*.

G. *Adobe Photoshop CS3*

Adobe Photoshop CS3 dalam Madcoms (2008:49) adalah "salah satu program grafis yang dapat digunakan untuk mengolah gambar dan membuat desain dalam pembuatan situs web". *Adobe Photoshop CS3* juga merupakan suatu program pengolahan image atau gambar Bitmap. Image atau gamabar Bitmap sering disebut dengan Raster.

Photoshop merupakan program penyunting gambar yang paling hebat dan paling popular hingga saat ini. Kemampuan serta fasilitasnya yang lengkap membuatnya diminati oleh para seniman, *professional*, maupun pemula yang membutuhkan sebuah program gambar yang lengkap namun mudah dalam penggunaannya.

Photoshop saat ini digunakan untuk membuat gambar-gambar untuk keperluan seperti iklan, brosur, poster, serta berbagai macam output cetak lainnya. Bagi para penggemar photografi ,*Photoshop* juga dapat digunakan untuk *me-retouch* foto yang sudah rusak, hingga memodifikasi foto sehingga menjadi lebih baik.

H. Konsep Dasar Pemrograman Berorientasi Objek

Menurut Sukamto dan Shalahuddin (2014 : 100) "metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya, serta suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis".

I. Konsep Dasar Berorientasi Objek

Menurut Sukamto dan Shalahuddin (2014 : 103) “Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya).”

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek menurut Rosa dan Shalahuddin (2014 : 104) yaitu :

1. Kelas (*class*)

   Menurut Sukamto dan Shalahuddin (2014 : 104) “Kelas adalah kumpulan objek-objek dengan karakteristik yang sama, statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut.”

2. Objek (*object*)

   Menurut Sukamto dan Shalahuddin (2014 : 106) “Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak.”

   Menurut Sukamto dan Shalahuddin (2014 : 106) “ Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya.”

3. Metode (*method*)

   Menurut Sukamto dan Shalahuddin (2014 : 107) “ Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.”

4. Atribut (*attribute*)

   Menurut Sukamto dan Shalahuddin (2014 : 108) “ Atribut dari sebuah kelas adalah variable global yang dimiliki sebuah kelas.”

5. Enkapsulasi (*encapsulation*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai onjek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya (Sukamto dan Shalahuddin, 2014 : 108).

6. Pewarisan (*inheritance*)

   Mekanisme yang mungkin satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya (Sukamto dan Shalahuddin, 2014 : 109).

7. Antarmuka (*interface*)

   Antarmuka atau *interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya (Sukamto dan Shalahuddin, 2014 : 109).

8. Polimorfisme (*polymorphism*)

   Menurut Sukamto dan Shalahuddin (2014 : 110) “Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.”

9. *Package*

   Sukamto dan Shalahuddin (2014 : 110) “*Package* adalah container atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam *package* yang berbeda.”

J. Peralatan Pendukung Sistem

1. UML (*Unifield Modelling language*)

UML (*Unified Modelling Language*) merupakan salah satu bentuk pemodelan atau sebuah bahasa standar yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Sukamto dan Shalahuddin, 2013:133).

Sebagaimana yang telah dijelaskan diatas tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa alat bantu diagram visual yang menunjukan berbagai aspek dalam sistem. Ada beberapa alat bantu diagram yang disediakan dalam UML antara lain :

a) *Use Case Diagram*

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) dari sebuah sistem informasi yang dibuat dengan cara mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi. Dengan kata lain *use case diagram* dapat digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak untuk dapat menggunakan fungsi-fungsi tersebut (Sukamto dan Shalahuddin, 2013:155). Setiap *use case* dilengkapi dengan skenario yang merupakan alur jalannya proses *use case* dari sisi aktor dan sistem.

Ada dua hal utama dalam pada *use case diagram* yaitu pendefinisian apa yang disebut aktor dan *use case*.Berikut penjelasan simbol-simbol yang

digunakan pada *use case diagram* menurut Sukamto dan Shalahuddin (2013:156).

1) Aktor (*actor*)

Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. Aktor menggambarkan sebuah tugas atau peran yang memberi input atau menerima informasi dari sistem dan bukan merupakan posisi sebuah jabatan seseorang. Pada *use case diagram*, aktor biasanya menggunakan kata benda dan tidak boleh ada komunikasi langsung antar aktor.

2) Use case

Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, dan biasanya dinyatakan menggunakan kata kerja. *Use case* dibuat berdasarkan keperluan aktor, merupakan apa yang dilakukan aktor pada sistem, bukan bagaimana sistem mengerjakannya. *Use case* diberi nama yang menyatakan hal apa yang dicapai dari hasil interaksi dengan aktor serta boleh terdiri dari beberapa kata, akan tetapi tidak boleh ada nama *use case* yang sama dalam satu diagram.

3) Asosisasi (*association*)

Merupakan komunikasi antara aktor dan *use case* serta bukan menggambarkan aliran data atau informasi. A*ssociation* digunakan untuk menggambarkan bagaimana aktor terlibat dalam *use case*.

4) Ekstensi (*extend*)

Merupakan relasi tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu. *Use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. Ekstensi melambangkan pekerjaan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.

5) Generalisasi (*generalization*)

Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Generalisasi digambarkan dengan sebuah garis berpanah tertutup pada salah satu ujungnya yang menunjukkan lebih umum.

6) Menggunakan (*include* atau *uses*)

Merupakan relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya, misalnya pemanggilan sebuah program. *Include* melambangkan pekerjaan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.

Sistem memiliki 4 aktor yang berhubungan dengan fungsi-fungsi sistem, yaitu Administrator,Pimpinan, Pegawai dan Timer. Administrator memiliki autentifikasi untuk menambahkan,mengurangi dan mengupdate data pegawai. Sedangkan pimpinan dan pegawai hanya dapatmelakukan login dan melihat informasi absensi, pimpinan merupakan turunan dari pegawai.Untuk pegawai hanya dapat melihat informasi absensi untuk dirinya sendiri, sedangkan pimpinan dapat melihat informasi dari bawahannya 2.

b) *Activity Diagram*

Diagram aktivitas atau *activity* diagram menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah bahwa *activity diagram* menggambarkan aktivitas atau kegiatan yang dapat dilakukan oleh sistem bukan apa yang dilakukan aktor (Sukamto dan Shalahuddin, 2013:161). *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram.

Berikut penjelasan simbol-simbol yang digunakan pada *activity diagram* menurut Rosa dan Shalahuddin (2013:162).

1) Status awal (*initial state*)

Status awal menggambarkan status awal aktivitas sistem, dan sebuah diagram aktivitas memiliki sebuah status awal.

2) Aktivitas (*activity*)

Menggambarkan aktivitas yang dilakukan sistem dan biasanya diawali dengan kata kerja.

3) Percabangan (*decision*)

Merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari dua.

4) Penggabungan (*join*)

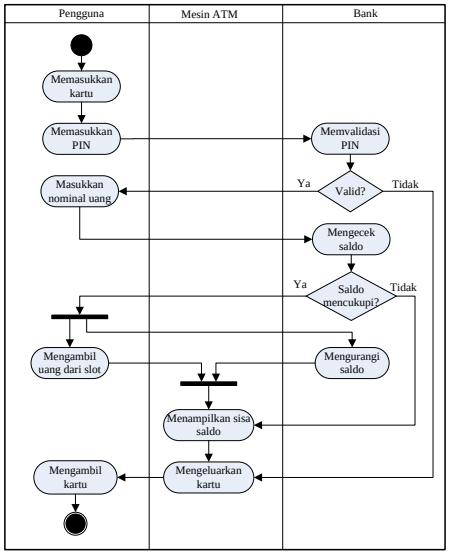
Merupakan asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

5) Status akhir (*final state*)

Menggambarkan status akhir dari sebuah sistem dan sebuah diagran aktivitas mempunyai sebuah status akhir.

6) *Swimline*

Swimline berfungsi memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

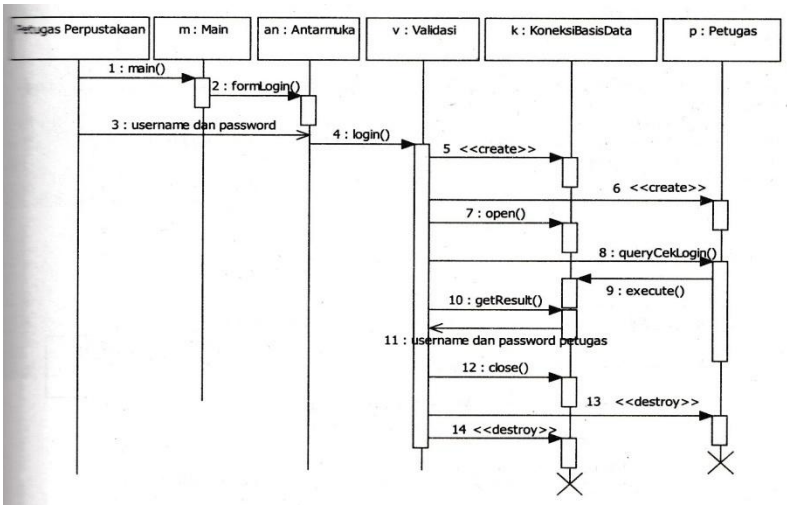


Sumber : (Sukamto dan Shalahuddin , 2013:161)

Gambar II.3. *Activity Diagram*

c) *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 menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* (Sukamto dan Shalahuddin, 2014 : 165).

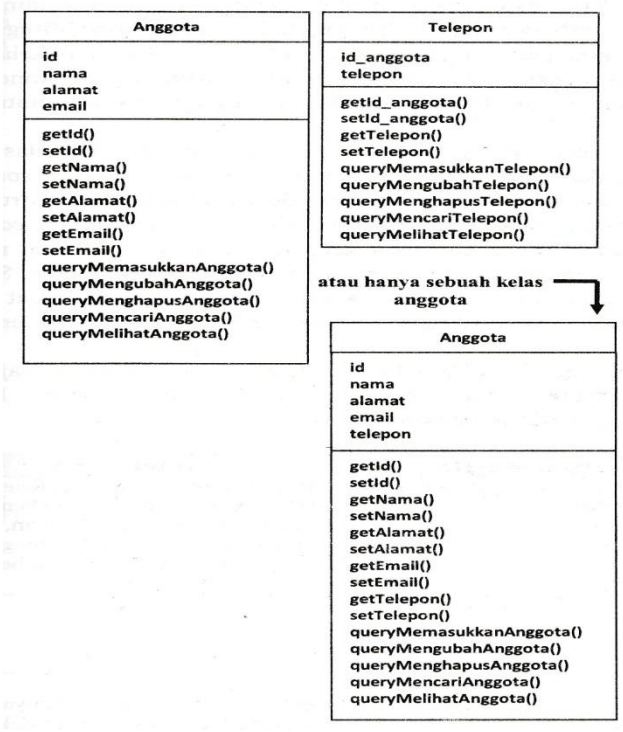


Sumber : Sukamto dan Shalahuddin (2014 : 209)

Gambar II.4. *Sequence Diagram*

d) Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron (Sukamto dan Shalahuddin, 2014 : 141).



Sumber : Sukamto dan Shalahuddin (2014 : 143)

Gambar II.5. *Class Diagram*

e) Component Diagram

Diagram komponen merupakan salah satu macam UML yang dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram ini fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Diagram ini juga dapat digunakan untuk memodelkan *source code* program perangkat lunak (Sukamto dan Shalahuddin, 2013:148).

Berikut penjelasan simbol-simbol yang ada pada *component diagram* menurut Sukamto dan Shalahuddin (2013:149).

1) Package

Package merupakan sebuah bungkusan dari satu atau lebih komponen.

2) Komponen

Merupakan kumpulan komponen yang membentuk suatu sistem.

3) Kebergantungan (*dependency*)

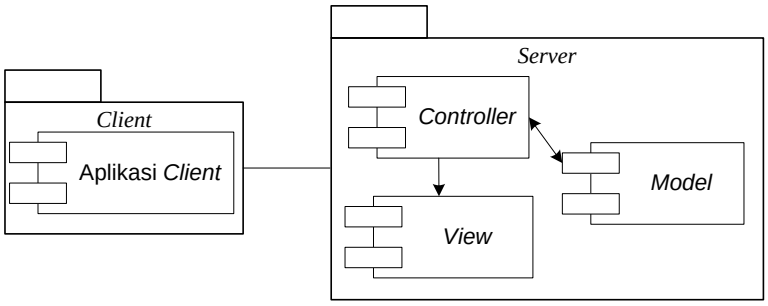
Menggambarkan kebergantungan antar komponen dan biasanya arah panah mengarah pada komponen yang dipakai.

4) Antarmuka (*interface*)

Berfungsi sebagai antarmuka agar tidak mengakses komponen secara langsung.

5) Link

Menggambarkan relasi antar komponen.



Sumber : (Sukamto dan Shalahuddin , 2013:148)

Gambar II.6. *Component Diagram*

f) Deployment Diagram

Deployment diagram berfungi untuk menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram ini juga dapat digunakan untuk memodelkan sistem tambahan (*embeded system*) yang menggambarkan rancangan *device, node* dan *hardware* (Sukamto dan Shalahuddin, 2013:154).

Berikut penjelasan simbol-simbol yang ada pada *deployment diagram* menurut Sukamto dan Shalahuddin (2013:154).:

1) Package

Package merupakan sebuah bungkusan dari satu atau lebih *node*.

2) Node

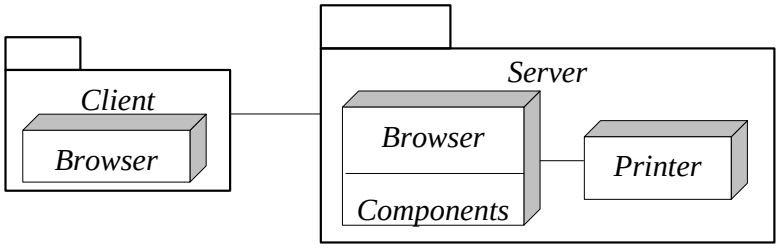
Node biasanya mengacu pada perangkat keras (*hardware*), peragkat lunak yang tidak dibuat sendiri (*software*).

3) Kebergantungan (*dependency*)

Menggambarkan kebergantungan antar *node* dan biasanya arah panah mengarah pada *node* yang dipakai.

4) Link

Menggambarkan relasi antar *node*.



Sumber : (Sukamto dan Shalahuddin , 2013:154)

Gambar II.7. *Deployment Diagram*

g) ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak serta memperlihatkan hubungan antar *data store* (Ladjamudin, 2005:142). Elemen-elemen diagram ERD menurut Ladjamudin (2005:143) adalah:

1) Entitas *(Entity)*

Pada diagram ERD, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem nyata

maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

2) Relationship

Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung *(relationship)* diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau kalimat pasif).

3) Derajat *Relationship(Relationship Degree)*

Relationship Degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*. Derajat *relationship* yang sering dipakai dalam ERD adalah:

(a) Unary Relationship

Unary Relationship adalah model *relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *Recursive Relationship* atau *Reflective Relationship*. Pada gambar berikut, *relationship* Menikah menunjukkan relasi satu ke satu antara *instance* dari entitas Pegawai.

(b) Binary Relationship

*Binary Relationshi*p adalah model *relationship* antara *instance-instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

(c) *Ternary Relationship*

Ternary relationship merupakan *relationship* antara *instance-instance* dari tiga tipe entitas secara sepihak. *Ternary relationship* tidak sama dengan gabungan dari tiga *binary relationship*.

(d) Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas atau *relationship*.

(e) Kardinalitas *(Cardinality)*

Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Terdapat tiga macam kardinalitas relasi, yaitu:

*(1) One to One (*1 *to* 1*)*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya. Yang berarti setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B dan begitu juga sebaliknya. Sebagai contoh, adanya relasi Kepala antara entitas Dosen dengan entitas Jurusan. Pada relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun tidak semua dosen menjadi kepala jurusan) dan setiap

jurusan dikepalai oleh paling banyak satu orang dosen. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.

Pada setiap kejadian kardinalitas relasi satu ke satu dapat ditangani dengan menyamakan *primary key* masing-masing entitas. Penyamaan *primary key* ini dilakukan untuk memudahkan daya ingat kita. Apabila *primary key* pada kedua entitas berbeda, maka gantilah nama *primary key* pada salah satu entitas tersebut dengan nama *primary key* yang sama pada entitas pasangannya.

*(2) One to Many atau Many to One (*1 *to M / M to* 1*)*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama. Sebagai contoh adanya relasi Ajar antara entitas Dosen dengan entitas Kuliah. Pada relasi ini, setiap dosen dapat mengajar lebih dari satu mata kuliah, sedangkan setiap mata kuliah diajar hanya oleh paling banyak satu orang dosen. Maka penggambarannya dapat dilihat pada gambar berikut ini.

Pada setiap kejadian relasi satu ke banyak dapat ditangani dengan cara menaruh *primary key* yang berada pada entitas satu ke entitas yang banyak, maka entitas yang banyak akan memiliki *foreign key* dimana *key* tersebut berasal dari entitas yang satu.

*(3) Many to many (*M *to* M*)*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya. Baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua. Yang berati setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, demikian juga sebaliknya dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A. Sebagai contoh adanya relasi Belajar antara entitas Mahasiswa dengan entitas Kuliah. Pada relasi ini, setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah, juga sebaliknya setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa. Maka penggambarannya adalah sebagai berikut.

Pada setiap kejadian kardinalitas relasi *many to many*, dapat ditangani dengan cara membuat file konektor (file baru) sehingga relasi langsung *many to many* berubah menjadi relasi tidak langsung satu lawan banyak melalui file konektor. Isi file konektor adalah minimal berisi dua *primary key* yaitu satu dari entitas dosen dan satu lagi dari entitas kuliah, maka pada setiap kasus kardinalitas relasi *many to many* akan menghasilkan tiga relasi baru. Seperti pada gambar diatas, entitas mahasiswa dengan *primary key* NIM dan entitas kuliah dengan *primary key* Kd_MK serta file konektor belajar dengan *primary key* NIM dan Kd_MK. Atribut NIM pada file konektor belajar berfungsi sebagai *foreign*

key dari entitas mahasiswa dan Kd_MK pada file konektor belajar berfungsi sebagai *foreign key* dari entitas kuliah. Dengan demikian maka kardinalitas relasi *many to many* diatas telah berubah seolah-olah menjadi *one to many*.

2.2. Penelitian Terkait

Menurut Arifudzaki dkk (2010:01) dalam penelitiannya :
Setiap perusahaan memiliki kebutuhan informasi yang berbeda beda untuk meningkatkan produktifitas suatu perusahaan tersebut. Dahulu perusahaan menggunakan program Microsoft Excel untuk menyimpan data-data barang yang masuk dan keluar, permintaan konsumen, ketersediaan barang yang ada didalam gudang. Hal ini menjadi kendala adalah ketika semua informasi tersebut dibutuhkan, maka harus membuka semua tabel yang ada *(Sheet by sheet)*. Berdasarkan permasalahan tersebut perlu dibuat sistem informasi yang akurat dan cepat.
Perusahaan pengeksport barang membutuhkan minimal satu Sistem Informasi Persediaan Barang. Sistem Informasi tersebut berisi informasi yang dibutuhkan pencari informasi yang diatur secara rapi, sehingga memudahkan dalam pencariannya. Sistem informasi yang banyak digunakan bahasa *scripting* seperti *Java script*, *PHP script*, *ASP script*. Bahasa *scripting* tersebut selanjutnya akan digabungkan dengan bahasa HTML.

Menurut Cahyana dkk (2012:D-252) dalam penelitiannya :
PT. Putera Agung Setia merupakan perusahaan yang bergerak pada bidang industri karoseri manufaktur. Salah satu tugas pokoknya, yauitu manajemen persediaan barang. Dalam hal ini PT. Putera Agung Setia masih menggunakan sistem manual, semua transaksi persediaan barang dicatat dan kemudian dilaporkan, dengan adanya sistem semacam ini banyak permasalahan yang terjadi terutama pada sistem pelaporan yang manual seperti ini. Untuk itu, menghindari pencatatan persedian yang manual dibutuhkan suatu wadah layanan sistem informasi persediaan barang yang dapat mempermudah manajemen perusahaan memperoleh informasi tentang laporan-laporan yang tersimpan dalam suatu databasepengolahan persediaan barang mulai dari pemesanan barang hingga sampai ke proses produksi. Didalam sistem informasi yang akan dibangun akan mudah diakses kapanpun saat dibutuhkan dengan adanya jaringan internet. Sistem ini dinamakan Sistem Informasi Persediaan Barang berbasis Web. Sistem ini diperuntukkan pada *staff* manajemen PT. Putera Agung Setia terutama manajer produksi. Sehingga kontrol persediaan barang produksi dapat terkendali dengan baik.

Menurut Dharma dkk (2015:01) dalam penelitiannya :
Pesatnya pertumbuhan Ilmu Pengetahuan Teknologi dan Komunikasi, khususnya dibidang sistem komputer disetiap aspek kehidupan dan penggunaan teknologi komputer dan teknologi komunikasi yang menghasilkan sebuah penggabungan sistem informasi yang saat ini menjadi lebih mudah untuk di akses tanpa mengenal adanya batasan waktu dan jarak dengan menggunakan jaringan internet. Model penjualan atau bisnis juga ikut terpengaruh dari perkembangan IPTEK tersebut, model yang sering digunakan diasanya dengan *affiliate marketing* terlebih dengan pesatnya pertumbuhan pengguna internet, akan memanjakan para *netizen* di dunia maya.
Netizen lahir disetiap belahan dunia khususnya negara-negara berkembang dan maju sebab mereka terbatas oleh waktu. Penggunaan internet dewasa ini menjadi salah satu bisnis yang menjadi *trend* baru dimasyarakat sekarang ini adalah bidang belanja *online (online shopping)*, sehingga memunculkan bentuk model toko-toko *virtual*, salah satu bentuk usaha dari para pebisnis *online* adalah memberikan fasilitas dan pelayanan yang memuaskan terhadap pelanggan, untuk mewujudkan hal tersebut dibutuhkan kualitas pelayanan yang baik.