

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Umum**

Teori umum yang melandasi dan mendukung dalam penelitian ini, antara lain yaitu:

##### **2.1.1. Teori Sistem**

Menurut (Riandy, 2011) Sistem adalah kumpulan dari komponen atau elemen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu. Elemen tersebut bisa berupa organisasi, orang atau benda yang melakukan suatu pekerjaan. Pengertian informasi adalah hasil dari pengolahan data dalam bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

Menurut (Purnama, 2010) Pengertian sistem secara umum adalah kumpulan elemen-elemen yang saling berkaitan dan menangani pemrosesan masukan atau input sehingga menghasilkan keluaran yang diinginkan, Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi.

Menurut (Pratomo, 2013) Sistem adalah suatu jaringan dimana pengertian itu digunakan. Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau

himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain, Sistem adalah sekumpulan unsur/elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan.

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

a. *Komponen Sistem (System component)*

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari subsistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar disebut dengan supra sistem, misalnya suatu perusahaan dapat disebut sebagai suatu sistem sedang industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya. Kalau sistem akuntansi dipandang sebagai suatu sistem, maka perusahaan adalah supra sistem dan industri adalah supra dari supra sistem.

b. *Batas sistem (System boundary)*

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipasang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*System environment*)

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan - dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung Sistem (*System Interface*)

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran (*output*) dari satu subsistem akan menjadi satu masukan (*input*) bagi subsistem yang lain dan akan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

e. Masukan Sistem (*System input*)

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan peralatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang diproses agar didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya sedangkan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran Sistem (*System output*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah transaksi menjadi laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

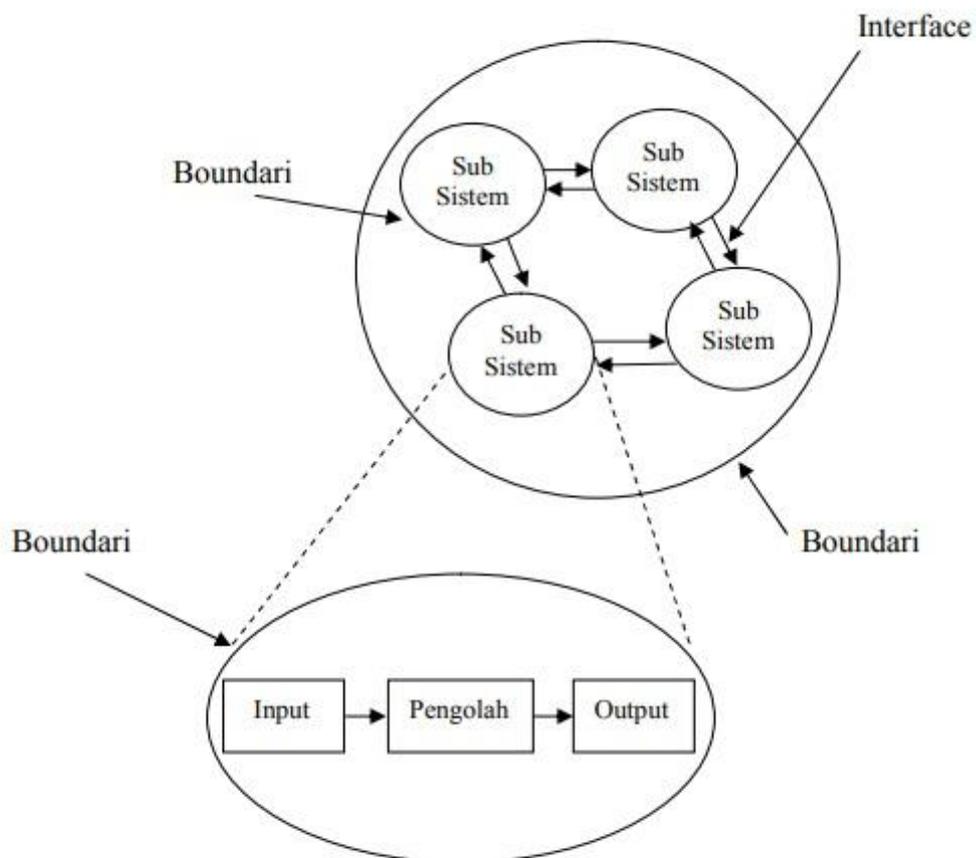
g. Pengolah Sistem (*System process*)

Suatu *system* dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu system produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

h. Sasaran Sistem (*System objective*) atau Tujuan (*goal*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objektif*). Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Hubungan antar tiap karakteristik sistem digambarkan pada gambar berikut :



**Gambar 2. 1** Hubungan antar tiap karakteristik sistem

### 2.1.2 Teori Informasi

Menurut (Maulana, 2015) Informasi dapat diperoleh dari sistem informasi atau disebut dengan *processing system* atau *processing systems* atau *generating*

*sistem* yang didefinisikan sebagai berikut : “Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Menurut (Purnamayudhia, 2015) Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi.

Menurut (Pratomo, 2013) Informasi sebagai data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini, informasi adalah sekumpulan data yang telah diklasifikasi atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Informasi dapat dikatakan baik jika memiliki kriteria dan karakteristik sebagai berikut:

a. *Information must be pertinent*

Informasi harus berhubungan. Pernyataan informasi harus berhubungan dengan urusan dan masalah yang penting bagi penerima informasi (orang yang membutuhkan informasi tersebut).

b. *Information must be accurate*

Informasi harus bebas dari kesalahan-kesalahan dan tidak memiliki bias atau menyesatkan. Informasi yang dihasilkan harus mencerminkan maksudnya. Keakuratan informasi seringkali bergantung pada keadaan.

c. *Information must be timely*

Informasi harus ada ketika dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan di dalam pengambilan keputusan.

d. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang yang satu dengan yang lainnya pasti berbeda.

e. Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkan informasi tersebut. Suatu informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Sebagian besar informasi tidak dapat ditaksir secara pasti nilai keuntungannya (dalam satuan uang), tetapi kita dapat menaksir nilai efektifitas dari informasi tersebut. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *costbenefit*.

### **2.1.3 Teori Sistem Informasi**

Menurut (Widyawati, 2013) Sesungguhnya yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer disebut sistem informasi berbasis komputer. Dalam prakteknya, istilah sistem informasi lebih sering dipakai tanpa embel-embel berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting. Dalam penulisan tugas akhir ini, yang dimaksud dengan sistem informasi adalah sistem informasi yang berbasis komputer. Salah satunya “Sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau suatu tujuan”.

Menurut (Anthony, 2017) Sistem informasi adalah kumpulan elemen yang saling berhubungan dan berfungsi untuk memproses, mengumpulkan, mendistribusikan serta menyimpan informasi yang berguna sebagai pendukung dalam pembuatan keputusan juga pengawasan dalam organisasi. Maka dengan adanya sistem informasi, pemilik toko akan lebih mudah dalam memberikan informasi sehingga penjualan yang dilakukan tidak terganggu.

Suatu sistem bisa dikatakan sebagai sebuah sistem informasi apabila memnuhi karakteristik utama dari sebuah sistem informasi. Karakteristik utama ini menunjukkan bahwa sebuah sistem memang benar-benar sebuah sistem yang dapat memberikan arus informasi dari host menuju usernya. Berikut ini adalah beberapa karakteristik yang dimiliki oleh sistem informasi:

a. Memiliki Komponen

Karakteristik pertama dari sebuah sistem informasi adalah memiliki komponen. Komponen ini merupakan bagian dari sebuah sistem interaksi, dimana keseluruhan komponen tersebut saling berinteraksi satu sama lain. Setiap komponen atau yang bisa juga disebut sebagai subsistem di dalam sebuah sistem informasi memiliki sifat untuk menjalankan fungsi-fungsi tertentu di dalam sebuah sistem informasi. Jadi, apabila subsistem atau komponen dari sistem informasi ini tidak dapat bekerja optimal, maka keseluruhan sistem informasi yang diimplementasikan tidak akan dapat berjalan secara optimal.

b. Memiliki Batasan atau *Boundary*

Karakteristik dari sebuah sistem informasi berikutnya adalah sebuah sistem informasi haruslah memiliki sebuah batasan sistem atau yang dikenal dengan istilah *boundary*. Batasan ini merupakan pembatas dari sebuah sistem informasi dengan sistem informasi lainnya, yang membuat sistem informasi tersebut menjadi satu buah kesatuan sistem informasi yang utuh, dan menunjukkan ruang lingkup yang dimiliki oleh sistem informasi tersebut. Jadi, dengan adanya *boundary* ini, sebuah sistem informasi tidak akan bekerja saling tumpang tindih satu sama lainnya, dan dapat berfungsi sesuai dengan tugas dan juga perannya masing-masing.

c. Memiliki Lingkungan Luar dari Sistem atau *Environment*

Karakteristik dari sistem informasi berikutnya adalah memiliki lingkungan luar dari sebuah sistem, atau yang disebut dengan *environment*. *Environment*

merupakan keseluruhan sistem dan juga lingkungan yang berad di luar batasan atau *boundary* dari sebuah sistem informasi. Sebuah sistem akan disebut sebagai sistem informasi, apabila sistem tersebut memiliki batasan atau *boundary*, dan juga memiliki lingkungan luar yang berbatasan langsung dengan sistem informasi tersebut.

d. Memiliki *Interface*

Interface atau antar muka merupakan karakteristik berikutnya yang harus dimiliki oleh sebuah sistem informasi. Ya, suatu sistem akan dianggap sebagai sebuah sistem informasi yang dapat dioperasikan dengan baik dan juga optimal apabila sistem informasi tersebut memiliki *interface* atau antar muka. *Interface* atau antarmuka ini merupakan media yang digunakan untuk dapat menghubungkan sebuah komponen atau subsistem yang terdapat pada sebuah sistem informasi.

Hal ini mengacu pada karakteristik pertama pada sebuah sistem informasi, dimana sistem informasi memiliki beberapa komponen dan juga subsistem yang menjadi dasar terbentuknya suatu keseluruhan sistem. Keseluruhan komponen dan juga subsistem tersebut di hubungkan dengan apa yang disebut dengan *interface*.

Berarti, sudah jelas terlihat, apabila suatu sistem informasi tidak memiliki *interface*, maka sistem tersebut tidak akan dapat berjalan dengan optimal.

e. Memiliki Input atau Masukan Sistem

Karakteristik berikutnya dari sebuah sistem informasi adalah sistem input atau masukan. *Input system* atau sistem masukan ini merupakan jenis

*energy* yang digunakan untuk dimasukkan ke dalam suatu sistem. Masukan atau input ini terdiri dari dua jenis, yaitu:

1. *Maintenance input*

*Maintenance input* merupakan input yang berhubungan dengan perawatan suatu sistem, dimana merupakan sebuah energi yang dimasukkan ke dalam sistem informasi, agar sistem informasi tersebut bisa berjalan dengan baik dan optimal

2. Signal input

Signal input merupakan *energy* yang merupakan sinyal, yang artinya, *energy* ini sangat berpengaruh terhadap proses transfer dan juga transmisi data atau informasi yang dimiliki sebuah host untuk diteruskan melalui sistem informasi menuju keluaran atau output.

- f. Memiliki *Output* atau Keluaran dari Sebuah Sistem

*Output* atau keluaran merupakan karakteristik dari sistem informasi yang berikutnya. *Output* merupakan keluaran *energy* atau hasil yang diteruskan oleh input. Hasil atau output ini bisa berupa tampilnya data dan juga informasi yang muncul pada *display user*, yang berisi informasi. Dengan adanya *output* ini, maka setiap user yang menggunakan sistem informasi dapat mengakses dan juga memanfaatkan layanan informasi yang ditunjukkan kepada dirinya, sehingga membuat sistem informasi dapat bekerja dengan optimal dan bermanfaat.

g. Memiliki Pengolah dan Pemrosesan Sistem

Karakteristik berikutnya yang harus dimiliki oleh sistem informasi adalah sebuah pengolah data atau pemrosesan sistem. Pengolah data atau pemrosesan sistem ini merupakan komponen atau bagian di dalam sebuah sistem informasi yang memiliki tugas utama untuk memproses input dari sebuah sistem informasi menjadi keluaran atau *output* dari sebuah sistem informasi.

Singkatnya, *processing system* ini membantu proses pengolahan data secara keseluruhan yang ada di dalam sebuah sistem informasi, lalu mentransmisikan hasil dari pengolahan data tersebut menuju *output* yang dikeluarkan oleh sistem dan dapat diakses oleh *user*.

h. Memiliki Sasaran dari Sistem

Karakteristik terakhir merupakan karakteristik yang mungkin paling penting dari sebuah sistem informasi. Karakteristik tersebut adalah sasaran dari sistem. Ya, sasaran dari sistem merupakan analisis berupa siapa saja yang akan menggunakan sistem informasi ini. Tanpa adanya sasaran dari pembuatan sistem, maka sudah pasti sebuah sistem informasi tidak akan bisa bermanfaat dan juga berguna.

Misalnya adalah, sebuah sistem informasi diimplementasikan untuk para auditor dan juga akuntan. Maka jenis dari sistem informasi yang akan diimplementasikan dan juga dikembangkan adalah jenis dari sistem informasi akuntansi, yang berisi data – data keuangan suatu perusahaan dan juga organisasi.

#### 2.1.4 Teori Basis Data (*Database*)

Basis data terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang /berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Sebagai satu kesatuan istilah, basis data (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (*arsip*) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama-sama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file/tabel/arsip* yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Menurut (Enterprise, 2017) *Database* merupakan sekumpulan informasi atau data yang digunakan dalam aplikasi. Apabila anda mengombinasikan aplikasi dengan *database*, maka data-data yang di input oleh user dapat disimpan ke dalam database dan data tersebut dapat digunakan sewaktu-waktu.

Menurut (Paillin, 2012) Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data, tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data-data yang saling berhubungan (*relation*), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Secara prinsip, dalam suatu database tercakup dua komponen penting, yaitu data dan informasi.

Menurut (Widyawati, 2013) *Database* merupakan kumpulan data dan informasi dalam jumlah yang tidak sedikit. Oleh karena itu, database harus disusun sebagai kriteria terpola dengan jelas sejak dari awalnya. *Database* akan diubah dalam bentuk tabel dalam *Microsoft Access* dengan memperhatikan: *File Table*, *Field*, *Primary Key*, dan sebagaimana yang merupakan bagian dari *database*.

Menurut (Athoillah, 2014) *Database* atau basis data adalah himpunan kelompok data (*arsip*) yang saling berhubungan dan diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan kembali dengan cepat, jadi dengan kata lain prinsip pembuatan basis data adalah untuk mengatur data (*arsip*).

Menurut (Akbar, 2013) Data adalah bahan yang akan diolah, dapat berupa angka, huruf, simbol, atau kata-kata yang menunjukkan suatu situasi. Sedangkan basis data itu sendiri adalah kumpulan *file* yang saling terkait dan membentuk hubungan yang mempunyai kaitan satu dengan yang lainnya sehingga membentuk suatu bangunan data yang digunakan untuk menginformasikan suatu perusahaan, instalasi, dan organisasi dalam batasan tertentu.

Database memiliki tiga karakteristik utama, yaitu:

- a. Data yang sama dapat diakses secara serempak oleh beberapa pengguna untuk berbagai kegunaan yang berbeda.
- b. Data tidak bergantung kepada struktur penyimpanan atau cara membaca data dari program aplikasi, atau data bersifat transparan terhadap program aplikasi.
- c. Data memiliki integritas (yaitu akurasi dan validasi) yang terkendali.



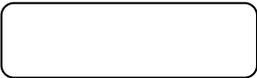
**Gambar 2. 2** Logo Database

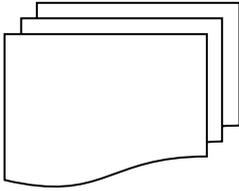
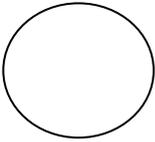
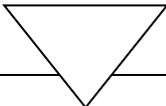
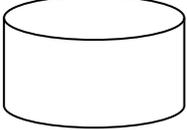
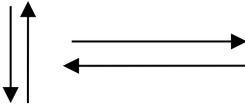
### 2.1.5 Teori Aliran Sistem Informasi (ASI)

Pengertian Aliran Sistem Informasi adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dari awal suatu program sampai akhir program. Bagan alir terdiri dari simbol-simbol yang akan dikerjakan.

Berikut adalah simbol-simbol dari aliran sistem informasi :

**Tabel 2. 1** Simbol – Simbol Pada Aliran Sistem Informasi

Gambar	Keterangan	Fungsi
	Simbol proses	Menunjukkan kegiatan proses dari operasi program komputer
	Simbol alternative	Menunjukkan alternative

	Simbol multi dokumen	Menunjukkan dokumen input dan output untuk proses manual, mekanik atau komputer
	Simbol kegiatan manual	Menunjukkan pekerjaan manual
	Simbol penghubung	Menunjukkan penghubung dalam satu halaman
	Simbol simpanan offline	Menunjukkan simpan data secara manual
	Simbol harddisk	Menunjukkan penyimpanan data secara elektronik
	Simbol garis alir	Menunjukkan arus dari proses

### 2.1.6 Teori SDLC

Menurut (Susanto & Andriana, 2016) SDLC (*Systems Development Life Cycle*, Siklus Hidup Pengembangan Sistem) atau *Systems Life Cycle* (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada

sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*). Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Terdapat 3 jenis metode siklus hidup sistem yang paling banyak digunakan, yakni: Siklus hidup sistem tradisional (*traditional system life cycle*), Siklus hidup menggunakan prototyping (*life cycle using prototyping*), dan Siklus hidup sistem orientasi objek (*object-oriented system life cycle*). SDLC juga merupakan metodologi umum dalam pengembangan sistem yang menandai kemajuan usaha analisis dan desain. Fase-fase atau tahapan-tahapan yang terdapat didalam SDLC meliputi :

a. Tahapan Perencanaan Sistem (*Systems Planning*)

Tahapan perencanaan adalah tahap awal pengembangan system yang mendefinisikan perkiraan kebutuhan-kebutuhan sumber daya seperti perangkat fisik, manusia, metode (teknik dan operasi), dan anggaran yang sifatnya masih umum (belum detail/rinci)

b. Tahapan Analisis Sistem (*Systems Analysis*)

Tahapan analisis system adalah tahap penelitian yang telah ada dengan tujuan untuk merancang system yang baru atau diperbarui

c. Tahapan Perancangan Sistem (*Systems Design*)

Tahap desain system adalah tahap setelah analisis system yang menentukan proses dan data yang diperlukan oleh system baru. Desain system dibedakan menjadi dua macam, yaitu desain system umum dan desain system terinci.

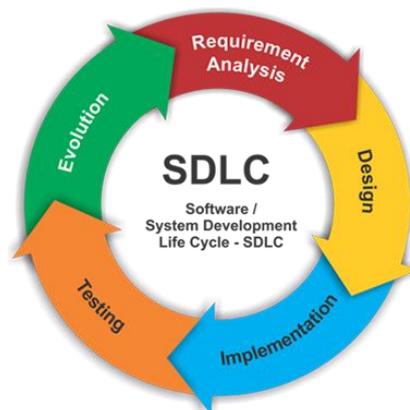
d. Tahapan Penerapan atau Implementasi Sistem (*Systems Implementation*)

Tahap implementasi atau penerapan adalah tahap dimana desain system dibentuk menjadi suatu kode (program) yang siap untuk dioperasikan.

e. Tahapan Pemeliharaan atau Perawatan Sistem (*Systems Maintenance*)

Tahapan pemeliharaan atau perawatan system merupakan tahap yang dilakukan setelah tahap implemementasi yang meliputi penggunaan system, audit system, penjagaan system, perbaikan system, dan peningkatan system.

Konsep SDLC mendasari model pengembangan perangkat lunak lainnya. Model pengembangan perangkat lunak tersebut antara lain *waterfall*, *prototype*, *iterative*, *spiral*, *rapid application development (RAD)* dan lainnya.



**Gambar 2. 3** Logo SDLC (*Systems Development Life Cycle*)

Menurut (Muslihudin.M) Pengembangan SDLC adalah proses yang digunakan oleh analis system untuk mengembangkan sistem informasi, termasuk persyaratan, validasi, pelatihan, dan pengguna (stakeholder) kepemilikan.

### 2.1.7 Teori Diagram UML

*UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program. *UML* juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainnya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya *UML*.

*UML* diciptakan oleh *Object Management Group* yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal:

- a. *Object*
- b. *Class*
- c. *Abstraction*
- d. *Encapsulation*
- e. *Inheritance*
- f. *Polymorphism*

Definisi *UML* :

- A. *Unified Modeling Language (UML)* merupakan metode pengembangan perangkat lunak (sistem informasi) dengan menggunakan metode grafis serta merupakan bahasa untuk visualisasi, spesifikasi, konstruksi serta dokumentasi.
- B. *Unified Modeling Language (UML)* adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan arti suatu sistem perangkat lunak.
- C. *Unified Modeling Language (UML)* dapat didefinisikan sebagai sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak.
- D. *Unified Modeling Language (UML)* merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti :
  - a. Spesifikasi
  - b. Visualisasi
  - c. Desain arsitektur
  - d. Konstruksi
  - e. Simulasi dan testing
  - f. Dokumentasi

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa "*Unified Modeling Language (UML)* adalah sebuah bahasa

yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis *Objek (OOP) (Object Oriented programming)*”.

Dalam *UML* sendiri terdapat beberapa diagram yang wajib dikuasai yaitu:

#### A. *Structural Diagram*

- a. *Class Diagram*, diagram ini terdiri dari *class*, *interface*, *association*, dan *collaboration*. Diagram ini menggambarkan objek - objek yang ada di sistem.
- b. *Object Diagram*, diagram ini menggambarkan hasil instansi dari *class diagram*. Diagram ini digunakan untuk membuat *prototype*
- c. *Component Diagram*, diagram ini menggambarkan kumpulan komponen dan hubungan antar komponen. Komponen terdiri dari *class*, *interface*, atau *collaboration*
- d. *Deployment Diagram*, diagram ini menggambarkan kumpulan node dan hubungan antar *node*. *Node* adalah entitas fisik dimana komponen di-*deploy*. Entitas fisik ini dapat berupa server atau perangkat keras lainnya.

#### B. *Behavioral Diagram*

- a. *Use case Diagram*, diagram ini menggambarkan kumpulan *use case*, aktor, dan hubungan mereka. *Use case* adalah hubungan antara fungsionalitas sistem dengan aktor *internal/eksternal* dari sistem.
- b. *Sequence Diagram*, diagram ini menggambarkan interaksi yang menjelaskan bagaimana pesan mengalir dari objek ke objek lainnya.

- c. *Collaboration Diagram*, diagram ini merupakan bentuk lain dari *sequence diagram*. Diagram ini menggambarkan struktur organisasi dari sistem dengan pesan yang diterima dan dikirim.
- d. *Statechart Diagram*, diagram ini menggambarkan bagaimana sistem dapat bereaksi terhadap suatu kejadian dari dalam atau luar. Kejadian (*event*) ini bertanggung jawab terhadap perubahan keadaan sistem.
- e. *Activity Diagram*, menggambarkan aliran kontrol sistem. Diagram ini digunakan untuk melihat bagaimana sistem bekerja ketika dieksekusi.

*UML* umum digunakan di dunia kerja dan dunia pendidikan. Dunia kerja menggunakan *UML* untuk menggambarkan sistem dan aplikasi berbasis objek yang sudah ada sebelumnya atau merancang sistem dan aplikasi di awal pengembangan sebelum memasuki tahap *coding*. Beberapa posisi pekerjaan yang sering bersinggungan dengan *UML* antara lain *system analyst*, *software architect*, dan *technical writer*. *System analyst* selain perlu memiliki kemampuan *coding*, sudah tentu wajib menguasai *UML*, jika akan mengembangkan aplikasi atau sistem berbasis objek agar dapat menjadi acuan formal bagi para *developer* yang bekerja sama dengannya. Tidak hanya itu *system analyst* pun terkadang dituntut untuk menguasai pembuatan diagram lain seperti *data flow diagram*, *flowchart*, *entity relationship diagram*, dan lainnya. Sedangkan *UML* menjadi kemampuan wajib bagi seorang *technical writer*. Pekerjaan di posisi ini cukup banyak dibutuhkan oleh beberapa perusahaan yang kesulitan dalam membuat dokumentasi sistem atau aplikasi yang mereka kembangkan. Selain membuat *user guide*, *technical writer* diperlukan untuk membangun keseluruhan dokumentasi dalam *UML* dari aplikasi

atau sistem, sehingga keberadaannya dapat menambah nilai aplikasi menjadi lebih tinggi.

Berikut ini adalah beberapa alat yang membantu kita untuk menggunakan *UML* disebut dengan *CASE* (*computer aided software engineering*). Ada beberapa *CASE* yang umum digunakan oleh para profesional yang menggunakan *UML*, diantaranya:

- a. *Sybase Power Designer*
- b. *Rational Rose*
- c. *Microsoft Visio*
- d. *Enterprise Architect*

Ada juga beberapa *CASE* yang *open source* dan dapat digunakan secara bebas tanpa terbentur dengan masalah pembayaran lisensi seperti berikut:

- a. *StarUML*
- b. *DIA Diagram Editor*
- c. *Umbrello*
- d. *ArgoUML*
- e. *Software Ideas Modeller*
- f. *Eclipse UML2*

Beberapa alat yang berbayar mempunyai kemampuan untuk melakukan *forward engineering* dan *reverse engineering*. *Forward engineering* adalah sebuah fitur dari suatu *CASE* dimana *UML* yang telah dirancang dapat menghasilkan kode program yang akan menjadi pijakan dan panduan untuk mengawali pengembangan aplikasi. Sehingga koherensi antara aplikasi dan *UML* dapat terjaga. *Reverse*

*engineering* adalah fitur dimana suatu *CASE* dapat membaca kode program dan menghasilkan *UML* dari kode program yang telah ada atau sedang dibuat.

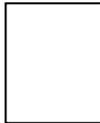
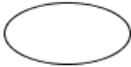
*UML* menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam *UML* antara lain:

a. Diagram *use case* (*use case diagram*)

*Diagram use case* menyajikan interaksi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

**Tabel 2. 2** Simbol diagram UML

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Generalization	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.

NO	GAMBAR	NAMA	KETERANGAN
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari

NO	GAMBAR	NAMA	KETERANGAN
			jumlah dan elemen- elemennya (sinergi).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. Diagram aktivitas (*activity diagram*)

Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagrama aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam *use case*.

c. Diagram sekuensial (*sequence diagram*)

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*. Kita dapat membaca diagram ini dengan memperhatikan obyek-obyek dan pesan-pesan yang ada di diagram. Obyek yang terlibat dalam aliran ditunjukkan dengan bujur sangkar yang ada di atas diagram.

d. Diagram kolaborasi (*collaboration diagram*)

Diagram kolaborasi menunjukkan informasi yang sama persis dengan diagram sekuensial, tetapi dalam bentuk dan tujuan yang berbeda. Sebagaimana diagram sekuensial, diagram kolaborasi digunakan untuk menampilkan aliran skenario tertentu di dalam *use case*. Jika diagram sekuensial disusun berdasarkan urutan waktu, diagram kolaborasi lebih berkonsentrasi pada hubungan antar obyek-obyek.

e. Diagram kelas (*class diagram*)

Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket di dalam sistem dan relasi antar mereka. Ia memberikan gambaran sistem secara statis. Biasanya, dibuat beberapa diagram kelas untuk satu sistem.

f. Diagram *statechart*

Diagram status atau *state diagram* atau *statechart* diagram menunjukkan kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status. Diagram status diadopsi dari penggambaran kondisi mesin status (*state machine*) yang menggambarkan status apa saja yang dialami oleh mesin, misalnya mesin pembelian kopi dengan uang koin.

Diagram Status menggambarkan seluruh *state/status* yang memungkinkan obyek-obyek dalam *class* dapat dimiliki dan kejadian-kejadian yang menyebabkan status berubah. Perubahan dalam suatu *state* disebut juga transisi (*transition*). Suatu transisi juga dapat memiliki sebuah aksi yang dihubungkan pada status, lebih spesifik apa yang harus dilakukan dalam hubungannya dengan transisi status. Pada diagram ini, perilaku sistem ditunjukkan. Sebuah status adalah kondisi selama

hidup objek atau interaksi selama memenuhi suatu kondisi, melaksanakan suatu aksi, atau menunggu suatu kejadian.

Simbol-simbol yang ada pada diagram status adalah sebagai berikut:

Simbol	Deskripsi
Status awal / kondisi awal 	Status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
Status 	Status yang dialami objek selama hidupnya
Status akhir / kondisi akhir 	Kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
Transisi	Garis transisi antar status pada daur hidup objek, transisi biasanya diberi nama pesan yang ada pada diagram

(Sumber:<http://www.sistem-informasi.xyz/2016/10/pengertian-diagram-status-diagram-state.html>)

Simbol	Deskripsi
Nama transisi 	Sekuen sehingga pesan pada diagram sekuen menjadi transisi bukan sebagai status, status merupakan kondisi yang dialami objek, bukan merupakan pesan ( <i>message</i> ), transisi juga bias memutar pada sebuah status

	Transisi internal melingkar/ke status sendiri
---	---

(Sumber:<http://www.sistem-informasi.xyz/2016/10/pengertian-diagram-status-diagram-state.html>)

g. Diagram komponen (*component diagram*)

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

1. *Source code* program perangkat lunak.
2. komponen *executable* yang di lepas ke *user*.
3. Basis data secara fisik.
4. Sistem yang harus beradaptasi dengan sistem lain.
5. *Framework* sistem, *framework* pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi, contohnya seperti struts dari *apache* yang menggunakan prinsip desain *Model-View-Controller (MVC)* dimana *source code* program dikelompokkan berdasarkan fungsinya. dimana *controller* berisi *source code* yang menangani *request* dan validasi, model berisi *source code* yang menangani manipulasi data dan *bussiness logic*, dan *view* berisi *source code* yang menangani tampilan.

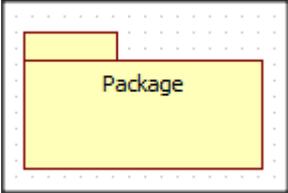
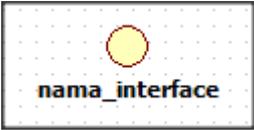
Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

- a. Komponen *user interface* yang menangani tampilan.

- b. Komponen *bussiness process* yang menangani fungsi-fungsi proses bisnis.
- c. Komponen data yang menangani manipulasi data.
- d. Komponen *security* yang menangani keamanan sistem.

Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan. Berikut adalah simbol-simbol standar yang biasa digunakan untuk membuat *component diagram* dengan menggunakan program *StarUML* :

**Tabel 2. 3** Simbol-simbol standar *component diagram*

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih komponen.
Komponen 	Komponen system.
Ketergantungan (Dependency) 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
Antarmuka/interface 	Sama dengan interface pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.

<p>Link</p> 	<p>Relasi antar komponen.</p>
---	-------------------------------

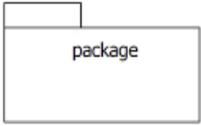
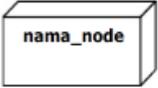
## 6. Diagram deployment

Diagram *Deployment* adalah pandangan yang terkait dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan *layout* jaringan dan lokasi komponen-komponen dalam jaringan. *Deployment View* berisikan prosesor-prosesor, peralatan-peralatan, proses-proses dan hubungan antar prosesor dan antar peralatan. Semua informasi ini digambarkan dalam suatu *deployment Diagram*. Hanya ada satu deployment Diagram dalam setiap *system*, sehingga hanya satu *deployment* dalam setiap model. Suatu *deployment* diagram menampilkan semua titik (*node*) dalam suatu jaringan, hubungan antar mereka, dan proses-proses yang dijalankan pada masing-masing *node*.

Ada dua jenis *node*.

1. *Device Node*, secara fisik menghitung sumber daya dengan pengolahan *memory* dan *services* untuk mengeksekusi software, seperti komputer atau *mobile phone*.
2. *Execution Environment Node (EEN)* adalah software penghitung sumber daya yang berjalan dalam *outer node* dan menyediakan layanan untuk *host* dan mengeksekusi *executable software*.

**Tabel 2. 4** Simbol-simbol pada *Deployment Diagram*

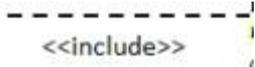
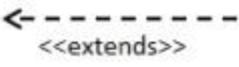
Simbol	Deskripsi
Package 	package merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
Node 	biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan / <i>dependency</i> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
Link 	relasi antar <i>node</i>

### 2.1.8 Teori Use Case

Menurut (Hendini, 2016) *Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case Diagram* yaitu:

**Tabel 2. 5** Simbol-Simbol *Use Case Diagram*

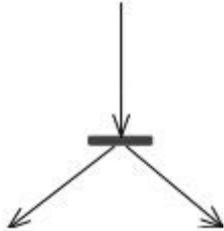
No	Gambar	Keterangan
1		<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja <i>Actor</i></p>
2		<p><i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktir, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
3		<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa</p>

		yang meminta interaksi secara langsung dan bukannya mengindikasikan data
4		Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan <i>system</i>
5		<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
6		<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

### 2.1.9 Teori Activity Diagram

Menurut (Hendini, 2016) Diagram aktivitas atau *activity diagram* menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagrama aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis (*business work flow*). Dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*) dalam *use case*. Simbol-simbol yang digunakan dalam *activity Diagram* yaitu:

**Tabel 2. 6** Simbol-Simbol Activity Diagram

No	Gambar	Keterangan
1		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
2		<i>End Point</i> , akhir aktivitas
3		<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis
4		<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
5		<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi

### 2.1.10 Teori Sequence Diagram

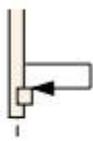
Menurut (Hendini, 2016) Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita

membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*. Kita dapat membaca diagram ini dengan memperhatikan obyek-obyek dan pesan-pesan yang ada di diagram. Obyek yang terlibat dalam aliran ditunjukkan dengan bujur sangkar yang ada di atas diagram.

Simbol-simbol yang digunakan dalam *Sequence Diagram* yaitu:

**Tabel 2. 7** Sequence Diagram

No	Gambar	Keterangan
1		<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
2		<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interfaces atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>
3		<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
4		<i>Message</i> , simbol mengirim pesan antar class

5		<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
6		<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
7		<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

### 2.1.11 Teori Class Diagram

Menurut (Hendini, 2016) Teori *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 Diagram* secara khas meliputi : Kelas (*Class*), Relasi *Associations*, *Generalization* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

## 2.2. TEORI KHUSUS

Teori Khusus yang melandasi dan mendukung dalam penelitian ini, antara lain yaitu:

### 2.2.1 Teori Kasir

Menurut (Muthohari, 2013) Kasir merupakan bentuk pemberian layanan yang diberikan oleh produsen baik terhadap pengguna barang diproduksi maupun jasa yang ditawarkan. Hal yang paling penting dalam suatu usaha adalah kualitas pelayanan yang diberikan konsumen akan merasa puas jika pelayanan yang diberikan sangat baik.

Kasir menurut kamus besar bahasa Indonesia adalah seorang pemegang kas (uang) atau orang yang bertugas menerima dan membayarkan uang. Selain istilah kasir, ada juga istilah *kassa*. Keduanya mempunyai perbedaan. Kasir adalah orangnya, sedangkan *kassa* adalah tempatnya. Profesi kasir di toko retail atau minimarket adalah seseorang yang pekerjaannya menerima uang pembayaran saat pembelian produk barang atau jasa dan melakukan pengembalian uang sisa pembayaran, sekaligus menyerahkan produk barang atau jasa kepada pelanggan (*customer*) di loket-loket kasir di suatu toko, super market, mini market, hotel, mall, restoran, rumah sakit, ataupun *departement store*. Selain itu, tugas kasir juga melakukan penghitungan jumlah total penjualan per hari, per minggu, ataupun per bulan serta mengenali barang yang paling laris terjual.

1. Tugas dan Tanggung Jawab Kasir, sebagai berikut:
  - a. Menjalankan proses penjualan dan pembayaran

- b. Melakukan pencatatan atas semua transaksi
  - c. Membantu pelanggan dalam memberikan informasi mengenai suatu produk
  - d. Melakukan proses transaksi pelayanan jual beli serta melakukan pembungkusan
  - e. Melakukan pengecekan atas jumlah barang pada saat penerimaan barang
  - f. Melakukan pencatatan kas fisik serta melakukan pelaporan kepada atasan.
  - g. Melakukan pengecekan atas stok bulanan
2. Kewajiban Khusus Kasir, sebagai berikut:
- a. Melakukan penghitungan secara teliti agar tidak terjadi selisih antara produk dan uang yang ada dengan laporan yang dibuat.
  - b. Melakukan pembukuan dengan baik dan benar
  - c. Mampu berkomunikasi dengan baik dan benar
  - d. Terampil dalam melayani konsumen
  - e. Jujur dan andal, tepat, jujur
  - f. Keterampilan dalam berkomunikasi

### **2.2.2 Teori Penjualan**

Menurut (Pakaja, 2012) Penjualan merupakan salah satu indikator paling penting dalam sebuah perusahaan, bila tingkat penjualan yang dihasilkan oleh perusahaan tersebut besar, maka laba yang dihasilkan perusahaan itu pun akan besar pula sehingga perusahaan dapat bertahan dalam persaingan bisnis dan bisa mengembangkan usahanya. Prediksi penjualan/ sales forecasting adalah salah satu

cara yang efektif untuk dapat meningkatkan laba perusahaan. Penjualan merupakan pembelian sesuatu (barang atau jasa) dari suatu pihak kepada pihak lainnya dengan mendapatkan ganti uang dari pihak tersebut.

#### A. Fungsi Penjualan

Fungsi ini bertanggung jawab untuk melayani order dari pembeli yang datang, setelah itu fungsi ini membuatkan faktur penjualan tunai, dan menyerahkan faktur tersebut kepada pembeli untuk kepentingan pembayaran harga barang ke fungsi kas.

#### B. Tujuan Penjualan

Dalam suatu perusahaan kegiatan penjualan adalah kegiatan yang penting, karena dengan adanya kegiatan penjualan tersebut maka akan terbentuk laba yang dapat menjamin kelangsungan hidup perusahaan. Tujuan utama penjualan yaitu mendatangkan keuntungan atau laba dari produk ataupun barang yang dihasilkan produsennya dengan pengelolaan yang baik dan mengharapkan keuntungan yang sebesar-besarnya, namun hal ini perlu peningkatan kinerja dari pihak distributor dalam menjamin mutu barang atau jasa yang akan di jual tersebut. Mencapai tujuan yaitu dalam perusahaan setiap penjualan harus memiliki tujuan penjualan yang dicapai. maka dapat disimpulkan bahwa tujuan umum perusahaan dalam kegiatan penjualan adalah untuk mencapai volume penjualan, mendapat laba yang maksimal dengan modal sekecil-kecilnya, dan menunjang pertumbuhan suatu perusahaan.

#### C. Jenis Penjualan

Secara umum terdapat dua jenis penjualan yaitu penjualan tunai dan penjualan kredit. Tetapi sebenarnya penjualan memiliki empat jenis penjualan yaitu penjualan

tunai, penjualan kredit, penjualan cicilan, dan penjualan konsinyasi. Biasanya masyarakat umum mendefinisikan arti dari penjualan kredit dan penjualan cicilan adalah sama, tetapi sebenarnya hal tersebut berbeda. Berikut ini adalah jenis-jenis dari penjualan, yaitu:

1. Penjualan tunai adalah penjualan barang dengan pembayaran cash atau langsung dibayar begitu barang diserahkan.
2. Penjualan kredit adalah penjualan barang dengan pembayaran tempo atau menunda pembayaran.
3. Penjualan angsuran atau cicilan adalah penjualan barang dagangan yang pembayarannya dilakukan secara bertahap dalam jumlah dan waktu yang telah ditentukan.
4. Penjualan Konsinyasi (*consignment*), Apabila metode penjualan konsinyasi (*consignment*) yang digunakan, barang dikapalkan tetapi kepemilikan tetap dipertahankan oleh penjual. Hal ini berarti bahwa produk diserahkan berdasarkan pembayaran tertangguh (*deferred-payment*) dan pada waktu produk dijual, maka penjual dibayar kembali oleh tertitip (*consignee*).

#### D. Sistem Penjualan

Sistem penjualan adalah suatu kesatuan proses yang saling mendukung dalam usahanya untuk memenuhi kebutuhan pembeli dan bersama-sama mendapatkan kepuasan dan keuntungan. Sistem penjualan adalah sistem yang melibatkan sumber daya dalam suatu organisasi, prosedur, data serta sarana pendukung untuk mengoperasikan sistem penjualan, sehingga menghasilkan informasi yang bermanfaat bagi pihak manajemen dalam pengambilan keputusan. Dari definisi di

atas dapat diartikan bahwa Sistem Penjualan merupakan suatu serangkaian metode penjualan produk barang maupun jasa yang saling berhubungan untuk menghasilkan pembeli baru. Sistem penjualan adalah suatu kesatuan proses yang saling mendukung dalam usahanya untuk memenuhi kebutuhan pembeli dan bersama-sama mendapatkan kepuasan dan keuntungan. Penjualan berkaitan dengan menghubungi pelanggan, menjual produk dan jasa, mengambil pesanan dan melanjutkan penjualan. Aktivitas ini dapat dianggap sebagai proses bisnis.

#### E. Sistem Penjualan Tunai

Aktivitas perusahaan pada umumnya berujung pada kegiatan penjualan. Penjualan merupakan suatu fungsi yang dianggap sebagai ujung tombak dalam suatu perusahaan. Karena fungsi itulah perusahaan memperoleh pendapatan. Sistem penjualan tunai adalah sistem yang diberlakukan oleh perusahaan dalam menjual barang dengan cara mewajibkan pembeli untuk melakukan pembayaran harga terlebih dahulu sebelum barang diserahkan pada pembeli. Setelah pembeli melakukan pembayaran, baru barang diserahkan, kemudian transaksi penjualan dicatat. Berdasarkan penjelasan di atas, maka dapat disimpulkan bahwa Penjualan tunai merupakan penjualan dengan mengambil barang dari supplier dan langsung dikirim ke customer secara pembayaran langsung dengan menggunakan uang tunai. Sistem penjualan tunai adalah sistem penjualan yang melakukan pembayaran terlebih dahulu, baru penyerahan barang kepada pembeli yang dapat menambah kas perusahaan dengan menggunakan suatu media agar dapat menyediakan informasi yang dibutuhkan manajemen.

### 2.2.3 Teori Visual Basic

Menurut (Widyawati, 2013) *Visual Basic* dibuat oleh *Microsoft*, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, *Visual Basic* juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. *Visual Basic* adalah pengembangan dari bahasa komputer *BASIC (Beginner's Allpurpose Symbolic Instruction Code)*. Bahasa *BASIC* diciptakan oleh *Professor John Kemeny* dan *Thomas Kurtz* dari Kampus *Darmouth* pada pertengahan tahun 1960-an. Bahasa program tersebut tersusun mirip dengan bahasa Inggris yang biasa digunakan oleh para programmer untuk menulis program-program komputer sederhana yang berfungsi sebagai pembelajaran bagi konsep dasar pemrograman komputer.

Sejak saat itu, banyak versi *BASIC* yang dikembangkan untuk digunakan pada berbagai platform komputer. Beberapa versinya seperti *Microsoft QBASIC*, *QUICKBASIC*, *GWBASIC*, *IBM BASICA*, *Apple BASIC* dan lain-lain. Hingga akhirnya *Visual Basic* juga telah berkembang menjadi beberapa versi, sampai yang terbaru, yaitu *Visual Basic 2008*. Bagaimanapun juga *Visual Basic 6.0* tetap menjadi versi yang paling populer karena mudah dalam membuat programnya dan ia tidak menghabiskan banyak memory komputer. *Microsoft Visual Basic* merupakan sebuah pemrograman komputer yang berjalan pada sistem operasi *Windows*. Bahasa Pemrograman adalah sekumpulan perintah/intruksi yang dimengerti oleh komputer untuk mengerjakan tugas-tugas tertentu. *Microsoft Visual Basic* selain disebut sebagai sarana Bahasa Pemrograman (*language*

*program*), juga sering disebut juga sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis *windows*.

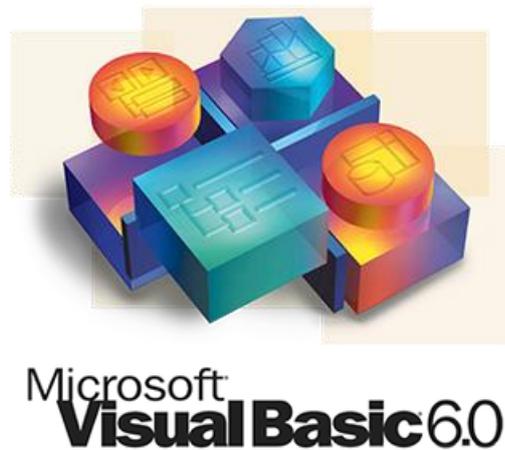
Menurut (Ginanti, 2016) *Visual Basic* adalah salah satu “bahasa pemrograman komputer”. Bahasa pemrograman adalah perintah atau intruksi-intruksi yang dapat dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Merupakan alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem *operasi windows*. *Microsoft Visual Basic* memiliki beberapa versi yaitu *Microsoft Visual Basic 3.0*, *Microsoft Visual Basic 5.0*, *Microsoft Visual Basic 6.0*.

Menurut (Enterprise, 2017) *Visual Basic* Sebagai berikut :

1. *Visual Basic* merupakan *high-level programming*. *High-level programming* adalah pemrograman yang telah menggunakan *syntax* (perintah dan struktur pemrograman) “bahasa manusia” sehingga lebih mudah dimengerti, terutama oleh orang awam.
2. *Visual Basic* sangat mudah digunakan untuk membuat aplikasi berbasis *GUI (Graphical User Interface)*, yaitu aplikasi yang dilengkapi jendela, menu-menu, ikon, dan tatap muka lainnya.
3. *Visual Basic* dikemas dan dikembangkan dalam sebuah *IDE (Integrated Development Environment)* yang bernama *Visual Studio*. Dengan demikian, mengembangkan aplikasi menjadi jauh lebih praktis karena tidak membutuhkan banyak sumber daya di luar *Visual Studio*.
4. *Visual Basic* sebenarnya *software* yang memiliki sejarah panjang. Jika anda di era tahun 90-an pernah menggunakan *Visual Basic* dan bertahun-tahun

kemudian tidak pernah mempelajari pemrograman ini, maka mungkin anda kan sedikit pangling. Mengapa? Sebab *Microsoft* tidak lagi membuat *Visual Basic* sebagai sebuah aplikasi pemrograman *stand-alone*. Sebagai gantinya, *Microsoft* merilis *MS Visual Studio* yang salah satu fiturnya adalah pengembangan aplikasi menggunakan *Visual Basic*. Jadi, dapat diandaikan seperti *MS Office* yang di dalamnya tersedia *MS Word*, *MS Excel*, dan seterusnya. Namun bedanya, anda tidak bisa menginstal *Visual Basic* sebagai aplikasi yang terpisah dari *Visual Studio*. *Visual Studio* sendiri disebut dengan istilah *Integrated Development Environment* atau yang disingkat dengan *IDE*. Selain *Visual Basic*, anda juga bisa bekerja dengan bahasa pemrograman lainnya, seperti C# dan C++, bersama-sama dengan *Script Visual Basic* untuk membuat aplikasi yang lebih kompleks.

*Microsoft Visual Studio 2012* merupakan sebuah *IDE (Integrated Development Environment)* yang dikembangkan oleh *microsoft*. *IDE* ini mencakup semua bahasa pemrograman berbasis *.NET framework* yang dikembangkan oleh *microsoft*. Keunggulan *Microsoft Visual Studio 2012* ini antara lain adalah *support* untuk *windows 8*, editor baru dengan *WPF (Windows Presentation Foundation)*, dan banyak peningkatan fitur lainnya.



**Gambar 2. 4** Logo Microsoft Visual Basic 6.0

#### 2.2.4 Teori MySQL

Menurut (Khairil & Koesoemanegara, 2011) MySQL adalah sebuah server database SQL *multiuser* dan *multi-threaded*. SQL sendiri adalah salah satu bahasa database yang populer di dunia. Implementasi program server database ini adalah program *daemon* “mysqld” dan beberapa program lain serta beberapa pustaka. MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah database berisi 10000 tabel dan 500 diantaranya memiliki 7 juta baris. Database ini dibuat untuk keperluan sistem database yang cepat, andal, dan mudah digunakan. Walaupun memiliki kemajuan yang cukup baik, MySQL untuk sistem operasi *Unix* bersifat *freeware*, dan terdapat juga versi *shareware* untuk sistem operasi Windows. Menurut pembuatnya, MySQL disebut seperti “*my-essque-ell*” dan bukan *my-sequell*. MySQL adalah Sebuah program database server yang mampu menerima dan mengirimkan datanya sangat cepat, multi user serta menggunakan perintah dasar SQL (*Structured Query Language*). MySQL merupakan dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. MySQL yang biasa kita gunakan

adalah MySQL *Free Software* yang berada dibawah Lisensi GNU/GPL (*General Public License*). MySQL Merupakan sebuah database server yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama *Michael Widenius*. Selain database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai Server, yang berarti program kita berposisi sebagai *Client*. Jadi MySQL adalah sebuah database yang dapat digunakan sebagai *Client* maupun server. Database MySQL merupakan suatu perangkat lunak database yang berbentuk database relasional atau disebut *Relational Database Management System* (RDBMS) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*). Database MySQL memiliki beberapa kelebihan dibanding database lain, diantaranya MySQL merupakan *Database Management System* (DBMS), MySQL sebagai *Relation Database Management System* (RDBMS) atau disebut dengan database Relational. MySQL Merupakan sebuah database server yang *free*, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya, MySQL merupakan sebuah database client MySQL mampu menerima *query* yang bertupuk dalam satu permintaan. Beberapa tipe data dalam MySQL yang sering dipakai:

**Tabel 2. 8** Tipe data dalam MySQL

Tipe data	Keterangan
INT(M) [UNSIGNED]	Angka -2147483648 s/d 2147483647
FLOAT(M,D)	Angka pecahan
DATE	Tanggal Format : YYYY-MM-DD
DATETIME	Tanggal dan Waktu Format : YYYY-MM-DD HH:MM:SS
CHAR(M)	String dengan panjang tetap sesuai dengan yang ditentukan. Panjangnya 1-255 karakter
VARCHAR(M)	String dengan panjang yang berubah-ubah sesuai dengan yang disimpan saat itu. Panjangnya 1 – 255 karakter
BLOB	Teks dengan panjang maksimum 65535 karakter
LONGBLOB	Teks dengan panjang maksimum 4294967295 karakter

Menurut (Enterprise, 2017) *SQL* merupakan bahasa yang digunakan untuk menjalankan *query database* baik untuk membuat, menghapus, atau memodifikasi suatu *database*. *MySQL* merupakan software yang tergolong kedalam *DBMS* yang bersifat *open source* menyatakan bahwa software ini dilengkapi dengan *source* (kode yang dipakai untuk membuat *MySQL*), selain itu tentu saja bentuk *executable* nya atau kode yang dijalankan secara langsung dalam sistem operasi dan bisa diperoleh dengan cara mendownload di internet secara gratis.

*SQL (Structured Query Language)* adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional, *MySQL* adalah salah satu perangkat lunak sistem basis data berbahasa *SQL* yang merupakan implementasi dari sistem manajemen basis data relasional (*RDBMS*), *MySQL* dapat digunakan secara langsung dengan mengetikkan perintahnya atau syntaxnya dan bisa juga

digunakan secara *embeded SQL*, artinya perintahnya dapat disisipkan kedalam bahasa pemrograman tertentu, misalkan saja pada *PHP*.



**Gambar 2. 5** Logo MySQL

(Sumber: <https://www.mysql.com/>)

### 2.2.5 Teori Xampp

Menurut (Takasana, 2017) *Xampp* merupakan perangkat lunak bebas dan mendukung berbagai macam sistem operasi. *Xampp* sendiri adalah kompilasi beberapa program yang digabungkan. Fungsi dari *xampp* ini digunakan untuk *server local host* atau *server* yang berdiri sendiri dan terdiri dari beberapa program seperti *perl*, *apache http server*, penerjemah bahasa yang sudah ditulis sesuai dengan Bahasa pemrograman *pop*, dan *mysql database*. Sedangkan nama *Xampp* sendiri adalah singkatan dari X yaitu empat sistem operasi apapun, *apache*, *mysql*, *perl* dan *php*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

Dalam *XAMPP* banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang pemrograman *web* *XAMPP* berfungsi sebagai *server offline*

yang berdiri sendiri (seringkali disebut sebagai *localhost*). *XAMPP* terdiri dari beberapa program yaitu *Apache HTTP Server*, *MySQL Database*, *PHP*, dan *Pearl*. *XAMPP* juga dilengkapi fitur manajemen database *PHPMYAdmin* seperti pada server hosting sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database secara mudah.

Berikut ini Penjelasan dari bagian-bagian *XAMPP*:

- a. *X*, Kenapa disebut dengan *system* operasi? karena *XAMPP* bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah *Windows*, *Linux*, *Mac OS* dan *Solaris*.
- b. *A* (*Apache*) merupakan aplikasi *web server*. *Apache* ini bersifat *opensource* yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama *Apache* adalah menghasilkan halaman *web* yang benar kepada *user* berdasarkan kode *PHP* yang dituliskan oleh pembuat halaman *web*. jika diperlukan juga berdasarkan kode *PHP* yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam *MySQL*) untuk mendukung halaman *web* yang dihasilkan.
- c. *M* (*MySQL*), merupakan aplikasi database *server*. Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah database. *MySQL* dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan *MySQL* untuk

menambahkan, mengubah, dan menghapus data yang berada dalam *database*.

- d. *P (PHP)*, bahasa pemrograman *web*. Bahasa pemrograman *PHP* merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*. *PHP* memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama *PHP* adalah *MySQL*. namun *PHP* juga mendukung sistem manajemen database *Oracle*, *Microsoft Access*, *Interbase*, *d-base*, *PostgreSQL*, dan sebagainya.
- e. *P (Perl)*, bahasa pemrograman, pertama kali dikembangkan oleh *Larry Wall* di mesin *Unix*. *Perl* pertama kali dirilis pada tanggal 18 Desember 1987 ditandai dengan keluarnya *Perl 1*. Dua diantara karakteristik utama *perl* adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. *Perl* sangat populer di gunakan dalam program-program *CGI (Common Gateway Interface)* dan protokol internet lainnya.



**Gambar 2. 6** Logo XAMPP

(Sumber : <https://www.apachefriends.org/index.html>)

### 2.3 Penelitian Terdahulu

Penelitian ini bukan merupakan penelitian yang pertama yang telah dilakukan metode yang sama. Penulis mendapatkan penelitian yang serupa yang telah dilakukan sebelumnya dengan studi kasus yang berbeda, yaitu :

1. (Andriyanti, 2016), JSIKA, Vol.5, No.12. Tahun 2016, ISSN 2338-137X  
**“Rancang Bangun Aplikasi Penjualan Air Mineral pada Toko Tirta Arlita Gresik”** , Tirta Arlita Gresik adalah toko yang mengkhususkan diri dalam penjualan mineral air. Sistem jual yang digunakan saat ini adalah sistem konvensional, pembeli harus datang langsung ke toko untuk melihat produk yang ditawarkan, sehingga memiliki beberapa masalah diantaranya: sering *error*, pengulangan data input yang masih berupa pembukuan, dan keterlambatan dalam pembuatan laporan. Untuk mengatasi masalah, dalam penyimpanan dan pencatatan transaksi penjualan perlu data yang terkomputerisasi. Untuk memenuhi kebutuhan tersebut, proyek akhir ini dibuat dari aplikasi penjualan air mineral di Tirta Arlita Gresik.
2. (Helmud, 2015), Jurnal Teknologi Informatika dan Komputer Atma Luhur, Vol 2. ISSN: 2406-7962 **“Rancang Bangun Aplikasi Penjualan Tunai Pada Diary Studio”** Kegiatan penjualan di *Diary Studio* dilakukan dengan cara datang langsung ke *Diary Studio* dan membuat pesanan kepada bagian penjualan. Kemudian pesanan akan di catat oleh bagian penjualan ke dalam buku pesanan penjualan. *Diary Studio* belum menerapkan penggunaan komputer dalam pengelolaan administrasi toko karena semua itu masih dilakukan dengan cara ditulis secara manual dalam bentuk buku yang

sederhana saja. Disini saya akan meriset tentang sistem penjualan tunai mulai dari proses pendataan barang, proses penjualan barang dan proses pembuatan laporan. Proses ini masih dilakukan secara manual. dan setelah mengevaluasi, sistem penjualan yang ada pada *Diary Studio* belum terlaksana dengan efektif dan efisien dan masih ada kelemahan dalam sistem yang ada pada saat ini di *Diary Studio*. Untuk mengatasi hal tersebut, maka diperlukan suatu sistem komputerisasi penjualan tunai yang lebih baik untuk mendukung kemajuan dan perkembangan studio tersebut. Sehingga memberikan kemudahan dalam pelayanan dan penyajian informasi serta pengelolaan data yang lebih baik, cepat, tepat dan akurat dari sistem yang sedang berjalan saat ini. Hasil yang diperoleh adalah aplikasi penjualan tunai yang bisa digunakan pemanfaatannya secara baik dan benar untuk pengawasan pelaksanaan sistem penjualan dengan mudah dan efisien.

3. (Purba, 2015b), Jurnal Teknik Informatika Politeknik Sekayu (TIPS) Volume II, No. 1, ISSN-P 2407-2192, "**Rancang Bangun Pengolahan Data Penjualan Sparepart Alat Berat (*Hose Hidrolik*) Pada PT.Sumatra Unggul Menggunakan Visual Basic 6.0**", Objek penelitian ini adalah PT. Sumatra Unggul Palembang. PT sumatera unggul Palembang merupakan perusahaan yang bergerak dibidang penjualan sparepart, kendala yang dialami ialah proses pencarian transaksi penjualan masih secara manual dan belum terkomputerisasi, proses pengolahan penjualan, data masih dicatat dalam buku tulis dan proses pembuatan kwitansi masih dilakukan dengan tulisan tangan menggunakan kwitansi biasa. Proses update data stok

sparepart juga dicatat dalam kartu stok dan *Microsoft excel* dengan input ulang, sehingga tidak adanya operasi otomatisasi, sehingga masih sering terjadi kesalahan dalam perhitungan stok. Hingga pada pembuatan laporan penjualan, data dibuat dengan merekap ulang tagihan sewa yang ada sehingga laporan cenderung tidak akurat dan tidak bisa dibuat secara cepat. Maka dari itu, sangat diperlukan sebuah Aplikasi yang dapat membantu Pengolahan data penjualan pada PT. Sumatera Unggul Palembang dengan cara membuat sebuah Aplikasi yaitu Aplikasi Penjualan Pada PT. Sumatera Unggul Palembang Dengan Menggunakan *Microsoft Visual Basic.Net* dan *Database Microsoft Access*. Diharapkan dengan adanya aplikasi ini dapat membantu administrasi dalam melakukan pekerjaannya khususnya transaksi penjualan menjadi lebih cepat dan tepat.

4. (Ginanti, 2016), Jurnal Ilmiah Mahasiswa Ekonomi Akuntansi (JIMEKA) Vol. 1, No. 2, (2016) , **“Perancangan Basis Data Pendapatan Cargo dan Pencatatan Data Pelanggan Menggunakan Visual Basic 6.0 pada PT.Intan Lubuk Pertama”** Tujuan dari penelitian ini adalah untuk mengetahui bagaimana sistem informasi akuntansi pendapatan dan pelanggan catatan data yang sedang berjalan di PT.Intan Lubuk Pertama. Perusahaan masih menggunakan sistem manual untuk melakukan setiap transaksi. Hal ini mengakibatkan perusahaan lambat dalam menyajikan informasi yang *relevan*, sehingga tidak dapat bekerja secara efektif dan optimal dalam melayani laporan yang dibutuhkan. Berdasarkan permasalahan tersebut, penulis tertarik untuk melakukan penelitian dengan

judul “Merancang Pendapatan Data Basis Data dan Rekaman Data Pelanggan Menggunakan Visual Basic 6.0 pada PT Intan Lubuk Pertama”. Data yang digunakan dalam penelitian ini adalah data kualitatif, menggunakan metode penelitian deskriptif, lanjut Teknik pengumpulan data menggunakan studi lapangan yang terdiri dari wawancara, observasi, dan literatur yang diperoleh langsung dari perusahaan. Alat desain ini menggunakan *Diagram Context*, *Data Flow Diagram (DAD)*, *Entity Relationship Diagram (ERD)*, dan normalisasi. Kemudian akan dilanjutkan ke desain database, input dan output. Berdasarkan hasil penelitian penulis menemukan bahwa 1) Prosedur sistem penerimaan perusahaan tidak beroperasi secara efektif. 2) Kelemahan dari dokumen yang digunakan, pencatatan transaksi dan laporan pendapatan yang dihasilkan masih dilakukan secara manual. 3) Proses perancangan informasi produksi yang diproses dan digunakan, mempersingkat waktu dalam presentasi serta memungkinkan perusahaan untuk menghitung pendapatan yang diperoleh untuk setiap laporan perhitungan pendapatan perusahaan yang dihasilkan secara otomatis.

5. (Purnama, 2010), *Journal Speed – Sentra Penelitian Engineering dan Edukasi – Volume 2 No 4 - 2010 - ijns.org*, ISSN : 1979-9330 (Print) - 2088-0154 (Online), **“Pembangunan Sistem Informasi Pendataan Rakyat Miskin Untuk Program Beras Miskin (Raskin) Pada Desa Mantren Kecamatan Kebonagung Kabupaten Pacitan”** Kemiskinan merupakan salah satu masalah utama yang dihadapi dalam kehidupan bermasyarakat.

Dengan adanya hal ini pemerintah Indonesia berinisiatif untuk membantu masyarakat yang miskin, hal ini bertujuan untuk memberantas kemiskinan yang ada di negara Indonesia. Untuk memperlancar program ini desa berkewajiban untuk mendata rakyat miskin yang pantas untuk mendapat bantuan langsung dari pusat. Dan data yang didapatkan disetorkan ke pusat. Adapun cara pendataan rakyat miskin tersebut biasanya harus sesuai dengan aturan yang ditentukan oleh pemerintah pusat. Namun, Desa Mantren dalam sistem pendataan raskin masih menggunakan metode konvensional dengan mencatat pada suatu kertas, dan sering terjadi kehilangan atau kerusakan arsip data raskin. Hal ini mengakibatkan suatu ketidak efisienan waktu dan ketidak akuratan dalam pengarsipan data. Dari proses pendataan ini merupakan salah satu permasalahan yang timbul di setiap desa. Oleh karena itu, di dalam laporan ini dibuat suatu sistem informasi pendataan rakyat miskin untuk program raskin dan Desa Mantren sebagai obyek Skripsi. Sistem informasi tersebut adalah sistem informasi pendataan rakyat miskin. Hal ini bertujuan untuk membangun Sistem Informasi Pendataan Rakyat Miskin Untuk Program Beras Miskin (Raskin) dan mempermudah kaur desa dalam mendata raskin yang ada pada Desa Mantren Kecamatan Kebonagung, Kabupaten Pacitan.

6. (Anthony, 2017), Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK) p-ISSN: 2355-7699 Vol. 4, No. 2, Juni 2017, hlm. 136-147 e-ISSN: 2528-6579 136, **“ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN BERDASARKAN STOK GUDANG BERBASIS**

**CLIENT SERVER (STUDI KASUS TOKO GROSIR “RESTU ANDA”)**, Restu Anda merupakan toko grosir dengan sistem penjualan yang belum terkomputerisasi atau masih manual. Dimana sistem penjualan yang berjalan adalah dengan menjual barang kepada downline berdasarkan pada berapa banyak barang yang masih tersedia di gudang. Karena proses pencatatan barang masih manual sehingga ketika pemilik toko memeriksa barang-barang di gudang, kadang-kadang jumlah item yang tercatat tidak sesuai dengan yang dilaporkan Akibatnya saat menerima pesanan pembelian barang dari downline, pemilik toko yang melihat pencatatan stok barang dan barang yang diinginkan downline tercatat habis atau kosong maka terpaksa pemilik toko menolak dan menawarkan barang lain yang pastinya tidak semua downline menerima tawarang tersebut. Maka dari itu untuk menyelesaikan masalah tersebut adalah dengan merancang sistem informasi penjualan berbasis *client-server*. Aplikasi ini dibangun dengan menggunakan teknologi *RMI (Remote Method Invocation)* yang memudahkan pengguna dalam mengelola data dari dua atau lebih komputer yang berbeda. Teknologi ini memudahkan pemilik toko untuk memantau jumlah stok barang yang ada digudang tanpa membutuhkan waktu yang lama, sehingga penjualan barang di toko Restu Anda dapat dijalankan dengan benar. Dan dengan aplikasi ini pemilik toko dapat memprediksikan berapa banyak kebutuhan barang yang harus disediakan untuk dijual kepada downline dengan melihat data penjualan yang sudah tersinkronisasi dengan data member.

7. (Nurlaela, 2013), IJNS – Indonesian Journal on Networking and Security, Volume 2 No 4, ISSN: 2302-5700 – <http://ijns.org>, “**APLIKASI SMS GATEWAY SEBAGAI SARANA PENUNJANG INFORMASI PERPUSTAKAAN PADA SEKOLAH MENENGAH PERTAMA NEGERI 1 ARJOSARI**”, Perpustakaan merupakan tempat untuk mengumpulkan, menyimpan, mengelola dan mengatur koleksi bahan pustaka secara sistematis. Perpustakaan berkaitan erat dengan proses belajar mengajar di sekolah yaitu memberikan sumbangan dalam upaya meningkatkan aktivitas siswa serta meningkatkan kualitas pendidikan dan pangajaran. Penerapan *SMS Gateway* untuk menunjang informasi perpustakaan sangatlah efektif. Dengan *SMS* maka lebih mudah dan cepat dalam menyampaikan informasi. Adapun untuk tujuan penelitian adalah memudahkan petugas perpustakaan dalam mengelola sistem informasi perpustakaan dan menyampaikan informasi kepada anggota perpustakaan dengan format *SMS*. Hasil yang diperoleh dari penelitian ini adalah aplikasi *SMS Gateway* sebagai sarana penunjang informasi perpustakaan dengan memanfaatkan komputer sebagai alat bantu untuk mengakses data.
8. (Xie, 2011), *Procedia Engineering* 00 (2011) 000–000 *Procedia Engineering* 15 (2011) 4004 – 4007, “**Procedia Engineering Design of University Employment Information System**”, Ada masalah yang sangat tidak efisien untuk menangani banyak pekerjaan pekerjaan dengan tugas berat. Untuk mengumpulkan, memproses, berbagi, dan menyimpan sejumlah besar informasi lulusan perguruan tinggi, pasar perekrutan, perusahaan dan

lembaga-lembaga publik, sistem database informasi pekerjaan dirancang untuk masalah ini. Menurut prinsip analisis permintaan dan desain sistem database, modul perangkat lunak dilakukan dalam bahasa C # dari *MS Studio visual*. Hasil pengujian menunjukkan bahwa data ketenagakerjaan dapat diatur dan disimpan secara teratur, mengurangi kekuatan dan kesulitan dari universitas dan lulusan dengan lebih sedikit beban manajemen informasi ketenagakerjaan.

9. (Lei, 2012), *Physics Procedia* 24 (2012) 2151 – 2158, "**Research on Design Information Management System for Leather Goods**", Ide pengaturan sistem manajemen informasi desain barang-barang kulit diajukan untuk memecahkan masalah yang ada dalam manajemen informasi barang-barang kulit saat ini. Prinsip kerja dari sistem manajemen informasi desain untuk barang-barang kulit dianalisis secara rinci. Pertama, pendekatan akuisisi informasi desain barang-barang kulit diperkenalkan. Kedua, metode pengolahan informasi desain diperkenalkan. Ketiga, manajemen informasi desain dalam database dipelajari. Akhirnya, penerapan sistem itu dibahas dengan mengambil produk sepatu sebagai contoh.