

BAB II

LANDASAN TEORI

2.1 Dotuku Coffee

Dotuku Coffee adalah Sebuah kedai kopi yang di miliki oleh perorangan yang berada di Jalan. Danau Toba No.19, Surabaya, Kecamatan. Kedaton, Kota Bandar Lampung, Lampung, 35141. Dotuku Coffee dilengkapi dengan berbagai fasilitas, seperti tersedianya wifi yang bisa diakses oleh pelanggan secara gratis dan tempat yang nyaman, Dotuku Coffee juga menyediakan minuman berbagai macam kopi dan makanan. Selain itu jadi Dotuku Coffee juga menjual biji kopi pilihan.

2.2 Rancang

Perancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari perancangan ialah untuk memberi gambaran yang jelas lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan harus berguna dan mudah dipahami sehingga mudah digunakan.

Perancangan adalah Sebuah Proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

Menurut Pressman (2009) perancangan atau rancang merupakan serangkaian prosedur untuk menterjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem di implementasikan.

2.3 Bangun

Menurut Pressman (2009) pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Jadi dapat disimpulkan bahwa Rancang Bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau

pengaturan dari beberapa elemen yang terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

2.4 Penjualan

Definisi penjualan menurut (Soemarso, 2018) dalam bukunya yang berjudul Akuntansi Suatu Pengantar, menyatakan: “Penjualan adalah penjualan barang dagang oleh perusahaan, penjualan dapat dilakukan secara kredit dan tunai”.

a. Penjualan Tunai

Definisi penjualan secara tunai menurut Mulyadi (2017) dalam bukunya yang berjudul Sistem Akuntansi menyatakan: “Penjualan yang terdiri dari penjualan barang atau jasa baik secara kredit maupun secara tunai.”

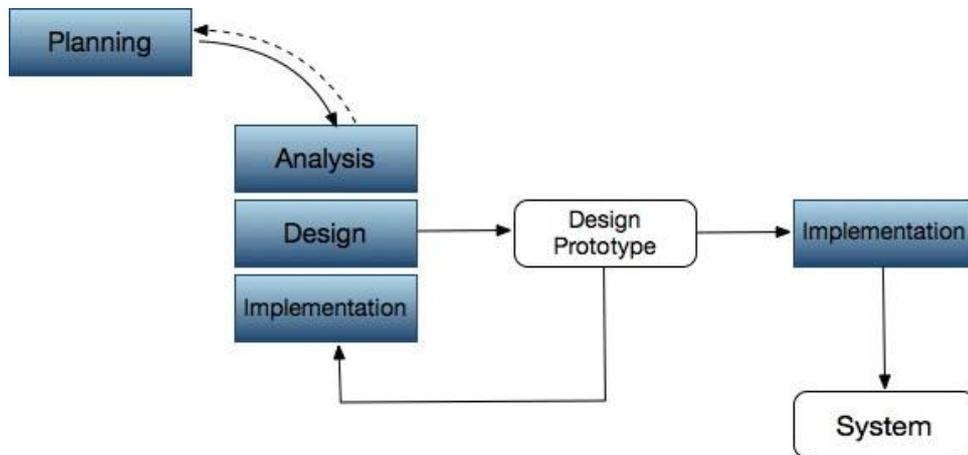
b. Penjualan Kredit

Definisi penjualan secara kredit adalah “Penjualan barang secara kredit dicatat sebagai debit pada akun piutang dagang dan kredit pada akun penjualan. Transaksi ini dicatat dalam buku penjualan.” (Soemarso, 2017). Penulis menyimpulkan bahwa penjualan tunai, yaitu proses penjualan barang dimana pembayaran atas transaksi tersebut dilakukan secara langsung sehingga menjadi pemasukan tunai bagi perusahaan. Penjualan kredit, yaitu proses penjualan barang dimana pembayaran atas transaksi tersebut ditangguhkan dan akan dibayar pada batas waktu tertentu dengan atau syarat sehingga menjadi piutang bagi perusahaan.

2.5 Metode Prototyping

Metode prototyping adalah pengembangan yang cepat dan pengujian terhadap model kerja dari aplikasi melalui proses interaksi secara berulang-ulang yang digunakan oleh para ahli sistem informasi dan ahli bisnis.

Prototyping membantu menemukan kebutuhan sistem pada tahap awal pengembangan. Selain itu prototyping juga berguna sebagai alat untuk mendesain dan memperbaiki *user interface* bagaimana sistem akan terlihat oleh orang-orang yang menggunakannya.



Sumber: Dennis, Wixom, & M.Roth (2014)

1. Planning

Tahap planning adalah tahapan dasar untuk membangun suatu sistem. Sebelum melakukan pembangunan sistem, diharuskan memiliki tujuan untuk apa sistem informasi dibuat.

2. Analisis

Pada tahapan ini dilakukan pengumpulan data yang bertujuan untuk mendapatkan data primer dan sekunder. Data primer adalah data-data yang dibutuhkan dimana data itu berkaitan langsung dengan objek pada penelitian. Sedangkan data sekunder adalah data pendukung yang dijadikan sebagai referensi suatu penelitian.

3. Desain

Tahapan desain merupakan proses pembuatan sebuah perancangan dari sistem penjualan biji kopi yang akan dibuat secara terstruktur.

4. Implementasi

Tahap terakhir dari metode prototype adalah tahap implementasi. Ketika sebuah rancangan belum memenuhi kebutuhan dan memecahkan masalah, maka rancangan itu akan direvisi dan kembali ke tahap sebelumnya. Ketika sistem sudah melalui proses evaluasi, barulah sistem akan siap untuk diimplementasikan pada implementasi terakhir.

2.6 Unified Modelling Language (UML)

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang sebesar pada developer pengembangan bahasa pemrograman berorientasi objek selanjutnya, (Rosa A.S, M.Shalahuddin, 2016).

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, CLOS. Secara actual, pengguna bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian disaat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sallay Shlaer dan Stephen Mellor (Shlaer-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon,1991), diikuti oleh Grady Booch (Booch,1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha- Premerlani-Eddy-Lorensen,1991), dan masih banyak lagi.

Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep Object Modelling Technique (OMT) dari Rumbaugh dan Booch (1991), konsep The Classes, Responsibilities, Collaborators (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lain nya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan Unified Modelling Language (UML).

Pada 1996, Object Management Group (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML

diakomodisi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar didalam metodologi berorientasi objek dan hal-hal yang terkait didalamnya.

Secara fisik, UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu Diagram Interchange Specification, UML Infrastructure, UML Superstructure, dan Object Constraint Language (OCL). Seluruh spesifikasi tersebut dapat diakses diwebsite <http://www.omg.org>, (Rosa A.S, M.Shalahuddin, 2016).

2.6.1 Diagram UML

Untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun, UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Ada beberapa diagram yang disediakan dalam UML antara lain :

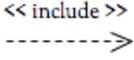
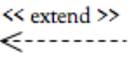
1. Use Case Diagram

Use case atau diagram Use case merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendefinisikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu, (Rosa A.S, M.Shalahuddin, 2016). syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

1. 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.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar kesan antar unit atau aktor.

Berikut adalah tabel simbol-simbol yang ada diagram *use case*:

Tabel 2.1 Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Include	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
3		Extend	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
4		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.
5		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
6		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

2. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem, (Rosa A.S, M.Shalahuddin, 2016). Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujiannya.

Berikut adalah tabel simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol *Activity Diagram*

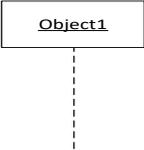
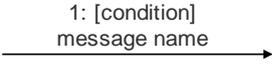
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Swimlane</i>	Menunjukkan siapa yang bertanggung jawab dalam melakukan aktivitas dalam suatu diagram.
2		<i>Action</i>	Langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meninggalkan activity, atau pada event yang spesifik.
3		<i>Initial State</i>	Menunjukkan dimana aliran kerja dimulai.
4		<i>Activity Final Node</i>	Menunjukkan dimana aliran kerja diakhiri.
5		<i>Decision Node</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi dan dua atau lebih transisi sesuai dengan suatu kondisi.
6		<i>Control Flow</i>	Menunjukkan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu.

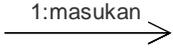
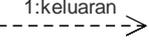
3. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendefinisikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstalasi menjadi objek itu. Membuat diagram sequence juga dibutuhkan untuk melihat skenario yang ada pada use case (Rosa A.S, M.Shalahuddin, 2016).

Berikutnya adalah table simbol-simbol yang ada pada Sequence Diagram:

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor</p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang dibuat itu sendiri</p>
<p>Garis hidup</p> 	<p>Men yatakan kehidupan suatu objek.</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe call</p> 	<p>Arah panah mengarah pada objek yang memiliki operasi atau metode karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
<p>Pesan tipe send</p>	<p>Menyatakan bahwa suatu objek mengirimkan data /masukan / informasi</p>

	ke objek lainnya, arah panah mengarah pada objek yang dikirim.
<p>Pesan tipe return</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode yang menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
<p>Pesan tipe create</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

4. Class Diagram

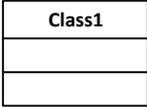
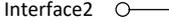
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi, (Rosa A.S, M.Shalahuddin, 2016).

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.
3. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Banyak berbagai kasus, perancangan kelas yang dibuat tidak sesuai dengan kelas-kelas yang dibuat pada perangkat lunak.

4. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas didalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada digram kelas sebaiknya memiliki jenis-jenis kelas berikut:
 1. Kelas main
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
 2. Kelas yang menangani tampilan sistem (*view*)
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
 3. Kelas yang diambil dari pendefinisian *use case* (*controller*)
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use care*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
 4. Kelas yang diambil dari pendefinisian data (*model*)
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat dibasis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut mutivalue pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggung jawabkan atau tetap ada didalam perancangan kelas.

Berikutnya adalah table simbol-simbol yang ada pada Class Diagram :

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Natarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan 	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.7. Bahasa Pemrograman

2.7.1 PHP

PHP atau Hypertext Preprocessor merupakan bahasa berbentuk script yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh Client. Php juga bisa disisipkan dalam bahasa HTML. (Adelheid dan Nist, 2012:2). PHP pertama kali diciptakan oleh seorang pria berkewarganegaraan Denmark yang bernama Rasmus Lerdorf pada tahun 1995. Banyak programmer yang tertarik untuk mengembangkan php karena bersifat Open Source. Pada awal peluncurannya, php hanya dibuat untuk diintegrasikan dengan Web Server Apache. Namu sekarang php juga dapat bekerja dengan Web Server seperti PWS (Personal Web Server), IIS (Internet Information Server), dan Xitami.

2.7.2 HTML

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 browser. Tag HTML tidak case sensitive. Jadi bisa menggunakan <HTML> atau <html>. Keduanya menghasilkan output yang sama. (Suyanto, 2007-2009:83).

2.7.3 CSS

CSS (Cascading Sytle Sheets) banyak dipergunakan untuk memperluas kemampuan HTML dalam memformat dokumen web atau untuk mempercantik tampilan web, bahkan untuk pemosisian dan layouting halaman web. Dengan mendefinisikan suatu style sekali saja maka style itu akan dapat digunakan berulang kali. (Suyanto, 2007-2009:83).

2.7.4 Xampp

Menurut Aditya (2011:16) “XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program: fungsinya adalah sebagai server yang berdiri sendiri (localhost),

yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public License dan bebas. Merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis”.

2.7.5 Javascript

Menurut Irawan (2008:3) “Javascript merupakan bahasa pemrograman berbasis script. Javascript memiliki kemampuan untuk menciptakan halaman web yang dinamis serta didukung oleh banyak web browser.”

Hal ini menjadikan javascript sebagai bahasa script yang paling populer dan banyak digunakan oleh banyak programmer web dalam pengembangan web. Pada aplikasi client-side, javascript berjalan didalam kode HTML. Menggunakan javascript kita bisa juga membuat aplikasi interaktif pada halaman web.

2.8 Data Base

Basis data (Database) adalah kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (key) dari tiap file yang ada. Basis data diartikan sebagai markas atau gedung, sedangkan data merupakan representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya (Eko Putra Membara, dkk, 2014 :

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Sebagai file data, suatu basis data tidak dapat menyajikan informasi secara langsung kepada pengguna. Pengguna harus menjalankan aplikasi untuk mengakses data dari basis data dan menyajikannya dalam bentuk yang bias dimengerti. Basis data (database system) adalah suatu

sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi (Muhammad Agus Suripto, dkk, 2017 : 2).

2.9 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah banyak oleh para pemogram aplikasi web. Contoh DBMS lainnya adalah : PostgreSQL (freeware), SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle dan Oracle Corp, Dbase, FoxPro, dsb (Hidayatullah dan Jauhari,2015:180).

Sibero (2013:97) MySQL atau dibaca “My Sekuel” dengan adalah suatu RDBMS(Relational Database Management System) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. Menurut Raharjo (2011:21), MySQL merupakan RDBMS (atau server database) yang mengelola database dengan cepat menampung dalam jumlah sangat besar dan dapat di akses oleh banyak user.

2.10 Point Of Sales

Point of sales (POS) merupakan skema dalam menyelesaikan proses transaksi penjualan. Pada POS penjual akan melakukan perhitungan jumlah yang harus dibayar oleh pelanggan, menunjukkan jumlah itu, dapat menyiapkan faktur untuk pelanggan dan menunjukkan opsi bagi pelanggan untuk melakukan pembayaran. Pada saat itu juga merupakan waktu di mana pelanggan melakukan pembayaran kepada pedagang dengan imbalan barang atau setelah penyediaan layanan. Setelah menerima pembayaran, pedagang dapat mengeluarkan tanda terima untuk transaksi, yang biasanya dicetak tetapi juga dapat dikeluarkan atau dikirim secara elektronik (Dyson, 2015). Pada era ini setiap bisnis penjualan sangat mungkin untuk menerapkan sistem POS karena dinilai sangat efisien serta memungkinkan untuk penjual lebih mudah untuk mengatur pemberian diskon dan hal lain untuk memanjakan pelanggannya.

2.11 Cloud Computing

Cloud computing merupakan sebuah paradigma dimana suatu informasi dapat disimpan secara permanen dalam sebuah server di internet dan tersimpan secara sesaat pada komputer klien termasuk juga didalamnya terdapat desktop, komputer tablet, notebook, handheld, sensor-sensor, dan sebagainya, sehingga dapat menghemat dari segi tempat, biaya dan perangkat keras (Hewitt 2008). Cloud computing sangat bergantung pada internet, karena semua proses komputasinya berada pada jaringan internet.

Cloud Computing berdasarkan tipe servicenya terbagi menjadi tiga yaitu :

1. *Infrastructure as a Service (IaaS)* IaaS merupakan bagian cloud computing yang terdiri dari bagian fisik (hardisk, memory , dll). Pada umumnya menggunakan teknologi virtual mechine, sumber daya yang ada digunakan secara bersama-sama. (Voorsluys,2011).
2. *Platform as a Service (PaaS)* PaaS merupakan bagian cloud computing yang terdiri aplikasi-aplikasi dasar contohnya sistem operasi, bahasa pemrograman dasar, database dan web server. (Voorsluys,2011).
3. *Software as a Service (SaaS)* SaaS merupakan bagian dari cloud computing yang terdiri aplikasi untuk digunakan oleh *end user*. Aplikasi biasanya disesuaikan dengan kebutuhan *end user*. (Voorsluys,2011).