

BAB II

LANDASAN TEORI

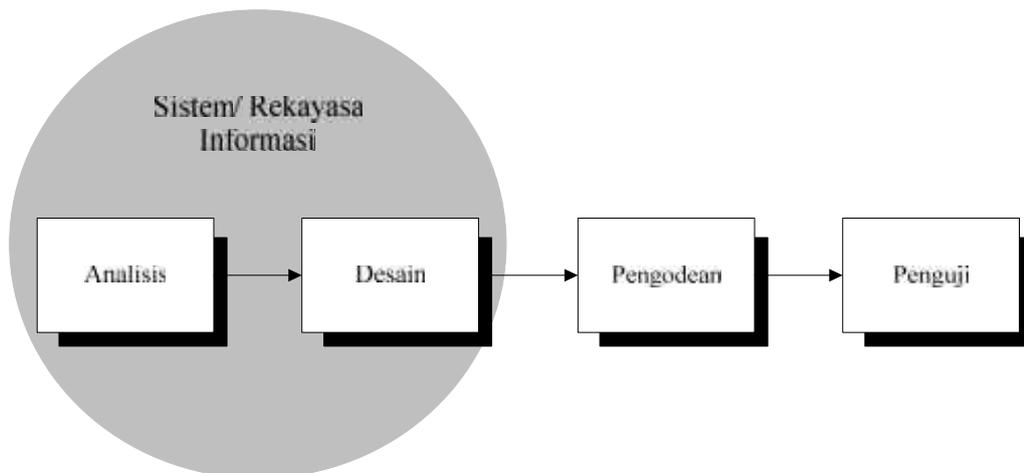
2.1. Tinjauan Pustaka

Teori dasar dalam mendukung penulisan skripsi sesuai dengan tinjauan pustaka adalah sebagai berikut :

A. Konsep Dasar Sistem Informasi

Menurut Wilkinson dalam Kadir (2014:9) “Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (*informasi*), guna mencapai sasaran-sasaran perusahaan”.

Pada awal pengembangan perangkat lunak, para pembuat program (*programer*) langsung melakukan pengodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak. SDLC dimulai dari tahun 1960-an, untuk mengembangkan sistem usaha skala besar secara fungsional untuk para komlomerat pada jaman itu. Menurut Shalahuddin dan Rosa (2013:28) ”Model sekuensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*) Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, penguji dan tahapan pendukung (*support*)”. Tahapan–tahapan yang ada pada *waterfall* secara global adalah sebagai berikut:



Sumber : Shalahuddin dan Rosa (2013:29)

Gambar II.1

Ilustrasi Model *Waterfall*

Metode Pengembangan Perangkat Lunak

1. Analisis kebutuhan perangkat lunak

Pada tahap ini akan dilakukan proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, resperensi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada

tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pengkodean

Desain harus ditranslasikan kedalam perangkat lunak. Hasil tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

B. Konsep Dasar Program

1. Konsep Dasar Program

Menurut Menurut Shalahuddin dan Rosa (2013:67) "Pemrograman terstruktur adalah konsep atau paradigma atau sudut pandang pemrograman yang membagi-bagi program berdasarkan fungsi-fungsi atau prosedur-prosedur yang dibutuhkan program komputer.

a. Ciri-ciri program yang baik/terstruktur

1. Dapat dijalankan dengan baik dan benar (sesuai dengan keinginan program).
2. Dapat dijalankan secara efisien dan efektif dengan tidak banyak menggunakan sintaks yang tidak perlu (menghasilkan *output* yang tepat dan benar dalam waktu yang singkat).

3. Mudah dibaca dan dipahami oleh orang lain maksud dan tujuan objek yang akan ditampilkan (memiliki logika perhitungan yang tepat dalam memecahkan masalah, dan ditulis dengan bahasa yang standar yaitu bahasa Indonesia/Inggris terstruktur sehingga tidak menimbulkan arti ganda).
4. Mudah diperbaiki jika terjadi kesalahan (semua operasi yang dilakukan harus terdefinisi dengan jelas dan berakhir setelah jumlah langkah dilakukan).
5. Biaya pengujian yang dibutuhkan rendah.
6. Memiliki dokumentasi yang baik.
7. Biaya perawatan dan dokumentasi yang dibutuhkan rendah.
8. Program hanya terdiri dari tiga struktur kontrol, yaitu struktur urutan, struktur seleksi, dan struktur repetisi/iterasi.

2. Komponen Program Terstruktur

Pemrograman terstruktur memiliki tiga komponen utama yaitu: pemrograman top-down, pemrograman modular, dan teorema struktur (Al-Bahra, 2006:264).

a. Pemrograman *Top-Down*

Teknik *Top-Down* didefinisikan sebagai suatu masalah yang kompleks dibagi-bagi kedalam beberapa kelompok masalah yang lebih kecil. Dari kelompok masalah yang kecil tersebut dianalisis. Apabila dimungkinkan maka masalah tersebut akan dipilah pilah lagi menjadi sub-bagian yang lebih kecil, dan setelah itu mulai disusun langkah-langkah untuk menyelesaikan secara detail.

b. Pemrograman *Modular*

Modul program didefinisikan sebagai sekumpulan instruksi yang memiliki operasi-operasi dan data yang didefinisikan, memiliki struktur internal yang tidak tergantung pada sub-program yang lain, dan merupakan satu kesatuan yang utuh yang akan dieksekusi secara berulang-ulang.

c. Teorema Struktur/Struktur Kontrol

Perancangan program terstruktur tidak dipengaruhi oleh kelebihan dan keunggulan salah satu bahasa pemrograman yang sedang populer pada saat itu, namun lebih dipengaruhi oleh tiga teorema struktur/struktur kontrol yang akan digunakan. Karena pada dasarnya algoritma dari program yang telah terstruktur akan dapat ditransformasikan kedalam bahasa pemrograman dan jenis sistem operasi apa saja. Terdapat tiga teorema terstruktur yaitu:

1. Struktur urut
2. Struktur seleksi
3. Struktur repetisi/pengulangan

3. Bahasa Pemrograman

1. PHP (*Personal Home Page*)

Menurut Hidayatullah dan Jauhari (2017:223) “*PHP* yang merupakan singkatan dari *Personal Home Page* adalah suatu bahasa *scripting* khususnya digunakan untuk web development. Karena sifatnya server side scripting, maka untuk menjalankan PHP harus menggunakan web server.

PHP juga dapat diintegrasikan dengan HTML, JavaScript, JQuery, Ajax. Namun pada umumnya, PHP lebih banyak digunakan bersamaan dengan file

bertipe HTML. Dengan menggunakan PHP kita bisa membuat website powerfull yang dinamis dengan disertai manajemen databasenya. Selain itu juga penggunaan PHP yaang sebagian besar dapat jalan di banyak platform.

Rasmus Ledorf merupakan orang yang menemukan bahasa *PHP*. *Ledorf*-lah yang pertama kali menulis mesin pengurai(*parsing engine*) dalam bahasa PHP sebagai CGI Perl pada tahun 1995, yang disebutnya “*Personal Home Page*” atau PHP. Tujuan awalnya untuk mencatat pengunjung-pengunjung yang membuka halaman resumanya di *web*. Kemudian ia menulis ulang kode-kode keseluruhan kedalam bahasa C, yang membuat program tersebut semakin canggih dan mampu berhubungan dengan basis data dan program-program lainnya. Tahun-tahun berikutnya Zeev Suraski dan Andi Gutmann menulis kembali PHP sehingga muncul versinya yang ketiga

Selain ASP dan Perl, PHP yang secara sintak sangat mirip dengan bahasa pemrograman C/C++, merupakan satu dari sejumlah besar dari bahasa *skrip server-side* yang paling banyak dipakai saat ini. Bahasa ini tidak melulu digunakan untuk membuat aplikasi-aplikasi internet, ia juga digunakan untuk aplikasi-aplikasi stand-alone. Walaupun demikian PHP sangat sering dipakai pada server-server *web* berbasis UNIX. Bahkan, PHP merupakan modul Apache yang paling sering digunakan.

2. *HTML (Hypertext Markup Language)*

Menurut Hidayatullah dan Jauhari (2017:15) “*Hypertext Markup Language (HTML)* adalah bahasa *standard* yang digunakan untuk menampilkan halaman web. Yang bisa dignakan dengan *HTML* yaitu:

- a. Mengatur tampilan dari halaman *web* dan isinya.
- b. Membuat tabel dalam halaman *web*.
- c. Mempublikasikan halaman *web* secara *online*.
- d. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi *via web*.
- e. Menambahkan objek-objek seperti citra, audio, video. Animasi, *java applet* dalam halaman *web*.
- f. Menampilkan area gambar (*canvas*) di *browser*.

Dokumen-dokumen *HTML* merupakan berkas teks yang mengandung dua bagian yaitu isi yang merupakan sesuatu yang ingin ditampilkan dan diperlihatkan dalam dokumen *web*, dan tag yang merupakan informasi pemformatan, yang tersembunyi dari pandangan pengguna, yang memberitahu browser bagaimana caranya menampilkan isi dokumen dihadapan pengguna.

HTML pada dasarnya merupakan himpunan bagian dari bahasa yang lebih luas jangkauannya yaitu *Standard Generalized Markup Language* (SGML), yang merupakan system pengkodean dan pemformatan untuk dokumen yang ditampilkan dilayar komputer maupun kertas. *Standard Generalized Markup Language* (SGML), yang merupakan himpunan bahasa pemformatan dokumen darimana *HTML* diturunkan. SGML pada mulanya merupakan bahasa pemformatan teks untuk digunakan oleh para penerbit dan industri multimedia untuk menyelesaikan masalah-masalah yang sering sekali muncul dari ketidakkompetebelan metode antara penyuntingan teks, pemformatan, serta aplikasi-aplikasi basis data. Dokumen-dokumen yang dikodekan dengan SGML memiliki informasi yang dinamakan markup, yang menandai dan

memerintahkannya bagaimana teks diformat untuk mendapat tampilan seperti yang dikehendaki. Sebagai contoh, awal suatu bab dapat ditandai dengan `<chapter>` dan akhir dari suatu bab ditandai dengan `</chapter>`. Penandaan ini memberitahu komputer untuk melakukan proses pemformatan luar (*output*).

3. CSS (*Cascading Style Sheet*)

Menurut Wahana Komputer (2013:2) “*Cascading Style Sheet* (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam”. CSS bukan merupakan bahasa pemrograman. Sama halnya dengan *style* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa style, misalnya *heading*, *subbab*, *bodytext*, *footer*, *image*, dan *style* lainnya. Untuk dapat digunakan bersama-sama dengan dalam beberapa berkas (*file*), pada umumnya CSS dipakai untuk memformat tampilan *website* yang dibuat dengan bahasa *HTML* dan *XHTML*.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna table, ukuran border, warna border, warna *hyperlink*, mana *mouse over*, spasi antar paragraph, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan anda untuk menampilkan halaman yang sama dengan format yang berbeda.

Adanya CSS memudahkan anda untuk mengatur dan memelihara sebuah *website* dan tampilannya karena CSS memisahkan antar bagian presentasi dan

isi dari *web* yang dibuat. Selain itu, kegunaan CSS dalam pembuatan *web* sebagai berikut:

- a. Kode *HTML* menjadi lebih sederhana dan mudah diatur
- b. Ukuran file menjadi lebih kecil sehingga *load file* lebih cepat
- c. Mudah untuk mengubah tampilan, hanya dengan mengubah file CSS saja
- d. Dapat berkolaborasi dengan *javascript* dan merupakan pasangan setia *HTML*
- e. Digunakan dalam hampir semua browser.

4. *SQL (Structure Query Language)*

Menurut Shalahuddin dan Rosa (2013:46) “*SQL* adalah bahasa yang digunakan untuk mengelola data pada *RDBMS*”. *SQL* awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. *SQL* mulai berkembang pada tahun 1970an. *SQL* mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh *ISO* dan disebut sebagai *SQL-86*.

Meskipun *SQL* diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar *RDBMS* yang berada saat ini, tetapi tidak semua standar yang tercantum dalam *SQL* diimplementasikan oleh seleruh *DBMS* tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh *DBMS* yang berbeda padahal *Query* yang dimasukan sama.

C. *E-Commerce*

Menurut Wong (2010:33) “Secara garis besar perdagangan elektronik (*E-Commerce*) didefinisikan sebagai cara untuk menjual dan membeli barang-barang (jasa) lewat jaringan elektronik”. Sejak awal, perdagangan elektronik mencakup transaksi pembelian serta transfer dana via jaringan Komputer.

a. Keuntungan Perdagangan Elektronik

Perdagangan secara elektronik menawarkan kepada perusahaan keuntungan jangka pendek dan jangka panjang. Perdagangan elektronik tidak hanya membuka pasar baru bagi produk dan/ atau jasa yang ditawarkan, mencapai konsumen baru, tetapi ia juga mempermudah cara perusahaan melakukan bisnis. Disamping itu, perdagangan elektronik juga sangat bermanfaat bagi pelanggan dan masyarakat umum.

a. Keuntungan Perdagangan elektronik bagi perusahaan adalah:

1. Memperpendek jarak

Perusahaan dapat lebih mendekatkan diri dengan konsumen. Dengan hanya mengklik *link-link* yang ada pada situs konsumen dapat menuju perusahaan dimana pun saat mereka berada

2. Perluasan pasar

Jangkauan pemasaran semakin luas dan tidak terbatas oleh area geografis dimana perusahaan berada

3. Perluasan jaringan mitra bisnis

Pada perdagangan tradisional, sangat sulit bagi suatu perusahaan untuk mengetahui posisi geografis mitra kerjanya yang berada di negara-negara lain atau benua lain.

4. Efisien

Perusahaan-perusahaan yang berdagang elektronik tidak membutuhkan kantor dan toko yang besar, menghemat kertas-kertas untuk transaksi, periklanan, serta pencatatan. Selain itu, perdagangan elektronik juga sangat efisien dari sudut waktu yang di gunakan

b. Keuntungan perdagangan elektronik bagi konsumen diantaranya:

1. Efektif

Konsumen dapat memperoleh informasi tentang produk/jasa yang dibutuhkannya dan bertransaksi dengan cepat dan murah.

2. Aman secara fisik

Konsumen tidak perlu mendatangi toko tempat perusahaan menjajakan barangnya dan ini memungkinkan konsumen dapat bertransaksi dengan aman sebab didaerah tertentu mungkin sangat berbahaya jika berkendara dan membawa uang yang besar.

3. *Fleksibel*

Konsumen dapat melakukan transaksi dari berbagai lokasi, baik dari rumah, kantor warnet, atau tempat-tempat lainnya.

c. Keuntungan Perdagangan Elektronik bagi masyarakat umum adalah :

1. Mengurangi Polusi dan Pencemaran Lingkungan

Dengan adanya perdagangan elektronik dimana saja, konsumen tidak perlu melakukan perjalanan-perjalanan ke toko-toko.

2. Membuka Peluang Kerja Baru

Era perdagangan elektronik akan membuka peluang-peluang kerja baru bagi mereka yang tidak buta teknologi.

3. Menguntungkan Dunia Akademis

4. Meningkatkan Kualitas Sumber Daya Manusia

b. Kerugian Perdagangan Elektronik

Di samping segala hal yang menguntungkan dari perdagangan elektronik memiliki sisi negatif. Namun, dari sudut pandang manapun, perdagangan elektronik memiliki segi positif lebih banyak dari segi negatifnya, kita perlu memahami beberapa segi negatif perdagangan elektronik diantaranya:

a. Meningkatkan Individualisme

Pada perdagangan elektronik seseorang dapat bertransaksi dan mendapatkan barang/jasa tanpa harus bertemu dengan siapa pun. Ini membuat beberapa orang menjadi berpusat pada diri sendiri.

b. Terkadang Menimbulkan Kekecewaan

Apa yang dilihat dilayar monitor kadang berbeda dengan apa yang dilihat dengan kasat mata.

D. *Unified Modelling Language (UML).*

Menurut Shalahuddin dan Rossa (2013:133) UML (*Unified Modeling language*) adalah “salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan requirement, membuat analis dan desain, serta

menggambarkan arsitektur dalam pemrograman berorientasi objek”. Yang termasuk dalam diagram UML adalah :

1. 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 (Shalahuddin dan Rossa, 2013:161). Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dilakukan oleh sistem.

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 aktifitas dianggap memiliki sebuah rancangan antar muka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan yang ditampilkan pada perangkat lunak.

2. Use case Diagram.

Use case atau diagram use case merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat (Shalahuddin dan Rossa,

2013:155). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

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* yaitu:

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah orang tapi aktor belum tentu orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.

3. *Component Diagram*

Diagram komponen atau *Component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem (Shalahuddin dan Rossa, 2013:148). Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. *Source kode* program perangkat lunak.
- b. Komponen *executable* yang dilepas ke *user*.
- c. Basis data secara fisik.
- d. Sistem yang harus beradaptasi dengan sistem lain.
- e. *Framework* sistem, *Fremwork* pada perangkat lunak yang merupakan perangkat kerja yang dibuat untuk memudahkan mengembangkan dan pemeliharaan aplikasi.

4. *Deployment Diagram*

Diagram deployment atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (Shalahuddin dan Rossa, 2013:148). Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut:

- a. Sistem tambahan (*embedded System*) yang menggambarkan rancangan *device*, *node*, dan *hardware*.
- b. Sistem *client/server*.
- c. Sistem terdistribusi murni.
- d. Rekayasa ulang aplikasi.

4. *Entity Relationship Diagram (ERD)*.

Menurut Shalahuddin dan Rosa (2013:53) “*ERD* adalah bentuk paling awal dalam melakukan perancangan basis data relasional”. Pemodelan basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*. *ERD* dikembangkan berdasarkan teori himpunan dalam matematika. *ERD* digunakan untuk basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan *ERD*. *ERD* memiliki banyak notasi seperti notasi *Chen*, *Barker*, notasi *Crow’s Foot*. Dan beberapa notasi lain namun yang banyak di gunakan adalah notasi *Chen*. Berikut adalah simbol-simbol yang digunakan pada *ERD* dengan notasi *Chen*.

Menurut Shalahuddin dan Rosa (2013:50) Tujuan utama dari *Entity relation diagram (ERD)* adalah mewakili objek data dan hubungan mereka. Komponen utama identifikasi untuk *Entity relation diagram (ERD)* berupa :

1. Entitas

Adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak, dengan informasi gabungan dapat diartikan sesuatu yang memiliki sejumlah sifat dan atribut yang berbeda. Atribut menentukan properti satu objek data dan mengambil salah satu dari tiga karakteristik yang berbeda.

2. *Relationship*

Hubungan ditunjukkan dengan garis yang diberi label yang menghubungkan objek. Sambungan antara objek dan hubungan dibangun dengan menggunakan kardinalitas dan modalitas.

3. *Link* adalah adalah tanda garis yang digunakan untuk menghubungkan komponen-komponen *ERD*.

Pada model *entity relationship diagram* hubungan antar *file* direlasikan dengan kunci relasi (*relation key*), yang merupakan kunci utama dari masing-masing *file*. Untuk membantu gambaran relasi secara lengkap terdapat juga tiga macam relasi dalam hubungan atribut dalam satu *file*, yaitu :

1. *One to one relationship*

Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antar keduanya digambarkan dengan panah tunggal.

2. *One to Many relationship*

Hubungan antara *file* pertama dan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Hubungan tersebut dapat digambarkan dengan panah banyak untuk menunjukkan hubungan banyak tersebut.

3. *Many to many relationship*

Hubungan antara *file* pertama dan *file* kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan panah ganda untuk menunjukkan hubungan banyak tersebut.

2.2. Penelitian Terkait

- a. Menurut Wibowo (2013:62) penjualan makanan khas Pacitan pada Toko Sari Rasa Pacitan belum menggunakan atau menerapkan teknologi informasi untuk memudahkan berlangsungnya proses penjualan. Sehingga menimbulkan masalah dalam penjualan produk tersebut, masalah yang timbul pihak Toko Sari Rasa kesulitan dalam memasarkan produknya di luar kota karena media pemasaran atau promosi hanya ada di daerah Pacitan saja padahal peluang penjualan di luar kota sangat menguntungkan dan konsumen di luar kota juga mengalami kesulitan dalam memperoleh informasi produk makanan khas Pacitan yang akan dibeli selain harus datang langsung ke Pacitan. Kesulitan yang terjadi dalam bisnis adalah dimana perusahaan kesulitan dalam memasarkan produk dan menemukan pembeli yang tepat. Begitu juga dengan pihak customer kesulitan dalam mencari produk-produk yang dibutuhkan.
- b. Menurut Astuti (2011:34) pada Apotek Jati Farma sistem yang masih digunakan adalah sistem konvensional yaitu melakukan pencatatan, baik itu transaksi penjualan ataupun pembelian barang ke dalam sebuah buku. Karena sistem konvensional tersebut membuat kinerja apotek menjadi kurang efektif dan efisien. Dan untuk proses kalkulasi penjualan obat yang hanya menggunakan cara konvensional yaitu dengan alat penghitung kalkulator.