

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Berikut adalah uraian tentang teori yang dibahas secara umum yang dijadikan sebagai teori yang memperkuat penelitian ini, diantaranya adalah:

2.1.1. Pengertian Sistem

Sistem berasal dari bahasa Latin *systema* dan bahasa Yunani *systema* adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Menurut beberapa ahli terdapat pengertian sistem antara lain:

1. Menurut Jerry FithGerald dalam buku Husda (2012: 111), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
2. Menurut Ludwig Von Bertalanfy dalam buku Husda (2012: 112), sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.
3. Menurut L.Ackof dalam buku Husda (2012: 112), sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.
4. Menurut James Havery dalam buku Husda (2012: 112), sistem adalah prosedur logis dan rasional untuk merancang suatu rangkaian komponen

yang berhubungan satu dengan yang lainnya dengan bermaksud untuk berfungsi sebagai suatu kesatuan dalam usaha mencapai suatu tujuan yang telah ditentukan.

2.1.1.1. Karakteristik Sistem

Model umum sebuah sistem terdiri dari *input*, proses, dan *output*. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan luar sistem (*Environment*)

Bentuk apapun yang ada di luar lingkup atau batasan yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan juga dapat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*Input*)

Energi yang dimasukkan kedalam sistem disebut dengan masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh, di dalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolahan sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan bila mengenai sasaran atau tujuan yang telah direncanakan. (Husda, 2012: 112)

2.1.1.2. Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam penerapan skstem atau subsistem informasi berbasis komputer. Siklus hidup sistem terdiri dari serangkaian tugas yang mengikuti langkah-langkah pendekatan sistem, karena tugas-tugas tersebut mengikuti pola yang teratur. Beberapa fase atau tahapan daur hidup sistem:

1. Mengenali adanya kebutuhan

Sebelum segala sesuatu terjadi, pastilah terlebih dahulu timbul suatu kebutuhan yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan organisasi. Volume kebutuhan itu meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan ini harus dapat didefinisikan dengan jelas. Tanpa adanya kejelasan mengenai kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitasnya.

2. Pembangunan sistem

Suatu proses atau seperangkat prosedur yang harus diikuti guna menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk memenuhi kebutuhan tersebut.

3. Pemasangan sistem

Setelah tahap pembangunan sistem selesai, kemudian sistem akan dioperasikan. Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem, dimana peralihan dari tahap pembangunan menuju operasional adalah pemasangan sistem, yang merupakan langkah akhir dari suatu pembangunan sistem.

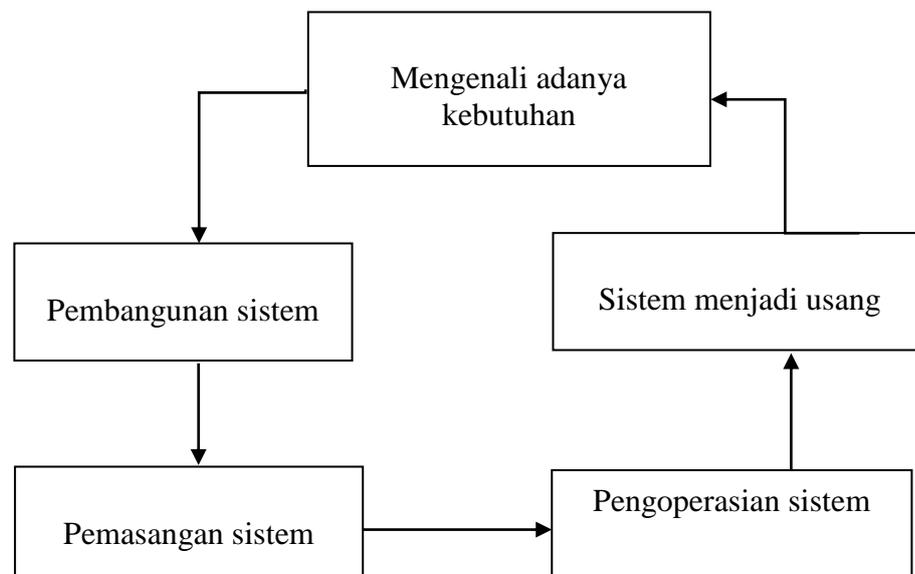
4. Pengoperasian sistem

Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi yang ditunjang oleh sistem informasi selalu mengalami perubahan karena pertumbuhan kegiatan, perubahan peraturan, ataupun

kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaiki atau diperbaharui.

5. Sistem menjadi usang

Kadang-kadang perubahan yang terjadi begitu drastic sehingga tidak dapat diatasi hanya dengan melakukan perbaikan pada sistem yang sedang berjalan. Tiba saat di mana secara ekonomis dan teknis, sistem yang ada sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantinya, (Sutabri, 2012: 20).



Gambar 2.1. Daur Hidup Sistem menurut Sutabri (2012:)

2.1.2. Konsep Dasar Informasi

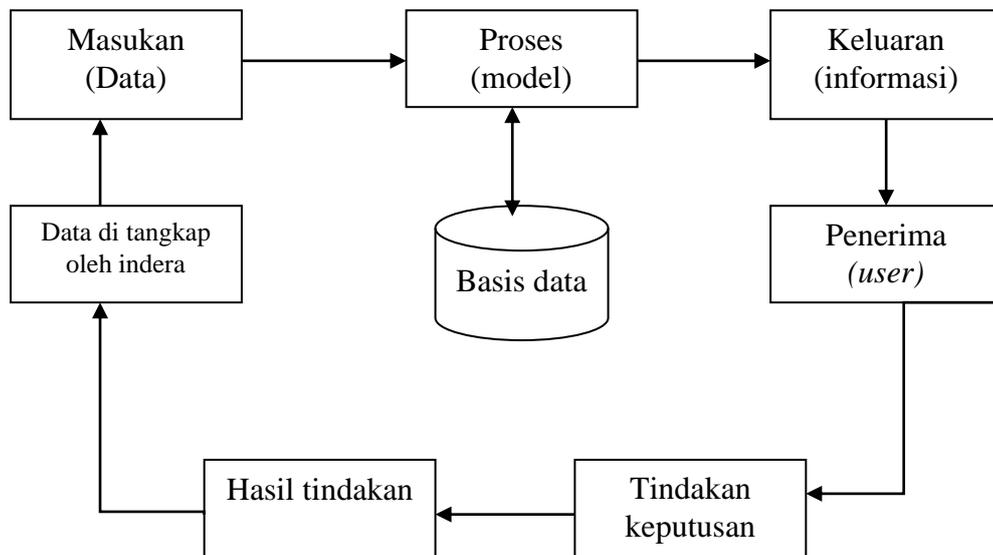
Dalam suatu organisasi atau perusahaan, informasi merupakan suatu yang memiliki arti yang sangat penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Secara umum informasi dapat didefinisikan

sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya.

Menurut Davis dalam buku Husda (2012:117) informasi adalah data yang telah diolah menjadi suatu bentuk yang penting bagi si penerimadan mempunyai nilai yang nyata yang dapat dirasakan dalam keputusan-keputusan yang sekarang atau keputusan-keputusan yang akan datang.

Menurut Raymond Mcleod dalam buku Husda (2012:117) informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Secara umum informasi dapat didefinisikan sebagai hasil dari pengumpulan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yg berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi ini dapat dilihat pada gambar 2.2 sebagai berikut:



Gambar 2.2. Siklus Informasi menurut Husda (2012: 118)

2.1.3. Sistem Informasi

Dalam kehidupan sehari-hari di era teknologi dan komputerisasi saat ini, kita sering mendengar istilah mengenai sistem informasi. Bahkan disadari atau tidak kita sering menggunakan atau memanfaatkan layanan yang diberikan sistem informasi baik di dunia akademis, kesehatan, industri maupun yang lainnya.

Sistem informasi merupakan suatu kumpulan dari komponen-komponen dalam suatu perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi. Dalam hal ini, TI hanya merupakan salah satu komponen dalam perusahaan. Komponen-komponen lainnya adalah prosedur, struktur organisasi, sumber daya manusia, produk, pelanggan, rekanan dan sebagainya. Keandalan suatu sistem informasi dalam organisasi terletak pada keterkaitan antar komponen yang ada, sehingga dapat dihasilkan dan dialirkan

suat informasi yang berguna (akurat, terpercaya, detail, cepat, relevan, dan sebagainya) untuk lembaga yang bersangkutan (Ali dan Wangdra, 2010: 13).

Menurut Pratama (2014: 10), sistem informasi merupakan gabungan dari empat bagian utama, keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah data menjadi informasi yang bermanfaat.

Menurut Raymond Mcleod dalam buku Husda (2012: 119), sistem informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.

2.1.3.1. Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (*building block*), keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak akan dapat melakukan fungsinya, yaitu pengolahan data dan tidak akan dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu dan akurat.

Menurut Husda (2012: 121) komponen-komponen dari sistem informasi ini dapat diuraikan sebagai berikut ini:

1. Blok Masukan (*Input Block*)

Input merupakan data yang masuk kedalam sistem informasi.

2. Blok Model (*Model Block*)

Kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan kotak alat (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

Merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

2.1.4. Pengembangan dan Perancangan Sistem Informasi

Pengembangan sistem informasi merupakan tindakan mengubah, mengganti atau menyusun sistem lama menjadi sistem baru baik secara sebagian maupun keseluruhan untuk memperbaiki sistem yang selama ini berjalan (yang telah ada). Desain atau perancangan merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implicit atau eksplisit dari segi performansi maupun sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat.

2.1.4.1. SDLC (*Software Development Life Cycle*)

SDLC atau *Software Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem-sistem perangkat lunak sebelumnya. Menurut Rosa (2011: 24) tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Merdefinikasikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resource*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirement analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem focus pada bagaimana dapat memenuhi fungsi-fungsi kebutuhan.

6. Pengembangan (*development*)

Mengkonversi *design* ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan. Membuat basis data dan mempersiapkan prosedur kasus pengujian; mempersiapkan berkas atau *file* pengujian, pengodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan pengujian.

7. Intergrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang spesifikasi pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operation and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi, termasuk implementasi akhir dan masuk pada proses peninjauan.

Ada beberapa model SDLC yang dapat digunakan dalam penerapan tahapan prosesnya, semuanya memiliki kelebihan dan kelemahan pada setiap model SDLC.

Menurut Rosa (2011: 26-37) model-model SDLC adalah sebagai berikut:

1. Model *waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, *design*, pengkodean, pengujian, dan tahap pendukung (*support*).

2. Model *prototype*

Model *prototype* dimulai dari pengumpulan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program *Prototype* agar pelanggan lebih terbayang dengan apa sebenarnya diinginkan. Program *Prototype* biasanya merupakan program *Prototype* yang belum jadi.

3. Model *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun untuk mengembangkan setiap komponen perangkat lunak.

4. Model *iterative*

Model iteratif mengkombinasikan proses-proses pada model air terjun dan iteratif pada *prototype*. Model inkremental akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (*inkremen/increment*).

5. Model *spiral*

Model spiral memasangkan iteratif dan model *prototype* dengan *control* dan aspek sistematis yang diambil dari model air terjun.

Pada penyusunan penelitian ini, diantara model SDLC diatas peneliti menggunakan pendekatan model *waterfall* sebagai perancangan sistem informasi penjualan pada minimarket JJS Mart. Karena model *waterfall* sangat cocok digunakan untuk kebutuhan pelanggan, mudah dipahami dan kemungkinan terjadinya perubahan selama pengembangan perangkat lunak sangat kecil. Hal positif dari model *waterfall* adalah struktur tahap pengembangan sistem jelas, dokumentasi dihasilkan di setiap tahap pengembangan, dan sebuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan.

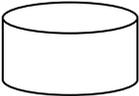
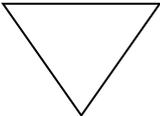
2.1.4.2. Aliran Sistem Informasi

Aliran sistem informasi sangat berguna untuk mengetahui permasalahan yang ada pada suatu sistem. Dari sini dapat diketahui apakah system informasi tersebut masih layak dipakai atau tidak, masih manual atau komputerisasi. Jika sistem informasinya tidak layak lagi maka perlu adanya perubahan dalam pengolahan datanya sehingga menghasilkan informasi yang cepat dan akurat serta keputusan yang lebih baik. Berikut ini simbol-simbol sistem aliran sistem informasi:

Tabel 2.1. Simbol aliran sistem informasi

<i>Input</i>		Dokumen
		Gabungan dokumen
		Manual keying
		Kartu plong
Proses		Proses berbasis komputer
		Proses manual
<i>Output</i>		Dokumen atau Laporan
		Catatan akuntansi (jurnal, buku besar)

Tabel 2.1 Lanjutan

Arsip		Disk
		Pita magnetik
Penyimpanan		Arsip A = menurut abjad C = menurut nomor T = menurut tanggal

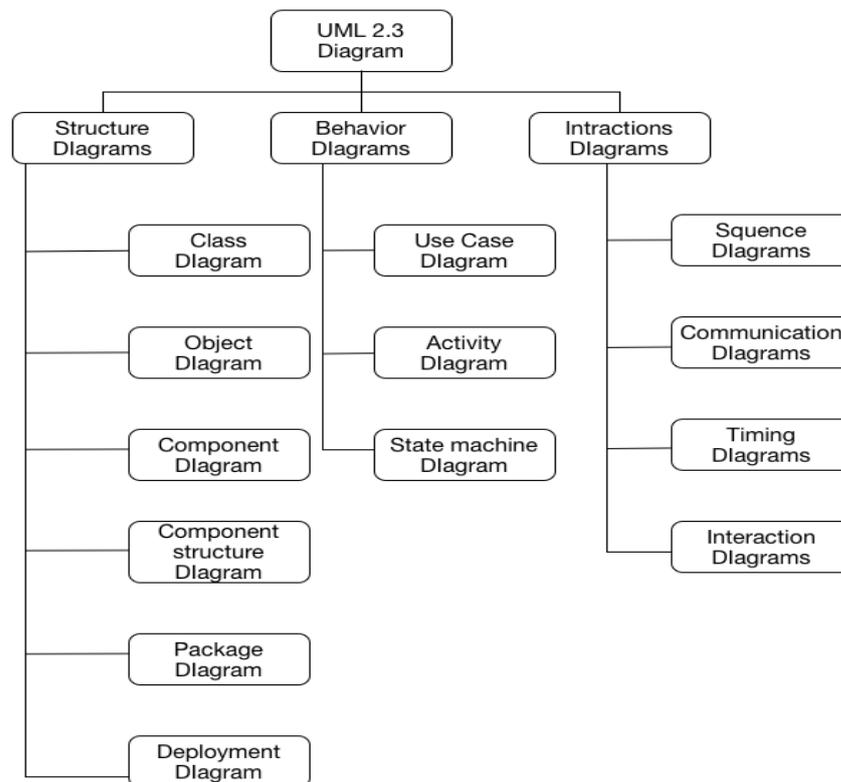
Sumber: Rosa (2011: 43)

2.1.4.2. UML (*Unified Modeling Language*)

Menurut Rosa (2011: 120) UML adalah sekumpulan spesifikasi yang dikeluarkan oleh *Object Management Group* (OMG) yang terdiri dari UML *Superstructure*, dan *Object Constraint Language* (OCL).

2.1.4.2.1. Diagram UML

Diagram UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.3 dibawah ini:



Gambar 2.3. Diagram UML menurut Rosa (2011: 121)

Berikut ini penjelasan singkat dari pembagian kategori tersebut:

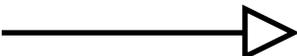
1. *Structure diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkain perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antara subsistem pada suatu sistem.

Dari 13 diagram dalam melakukan perancangan peneliti menggunakan *Class diagram*, *Use Case diagram*, dan *Sequence diagram*, berikut penjelasan dari masing-masing diagram:

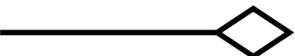
1. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Dalam mendefinisikan metode yang ada di dalam kelas perlu memperhatikan *cohesion* dan *coupling*. *Cohesion* adalah ukuran seberapa dekat keterkaitan instruksi di dalam sebuah metode terkait satu sama lain sedangkan *coupling* adalah ukuran seberapa dekat keterkaitan instruksi antara metode yang satu dengan metode yang lain dalam suatu sebuah kelas. Dalam diagram kelas terdapat beberapa simbol dalam penggunaannya. (Rosa, 2011).

Tabel 2.2. Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka / <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi / <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)

Tabel 2.2 Lanjutan

Kebergantungan / <i>dependency</i> 	Relasi antara kelas dengan makna kebergantungan antar kelas
Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whol -part</i>)

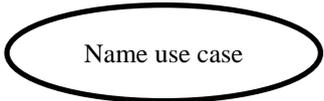
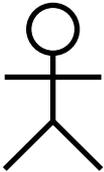
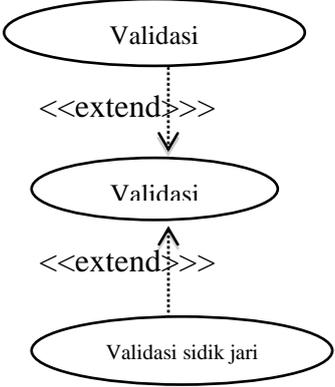
Sumber: Rosa (2011: 123)

2. Use Case Diagram

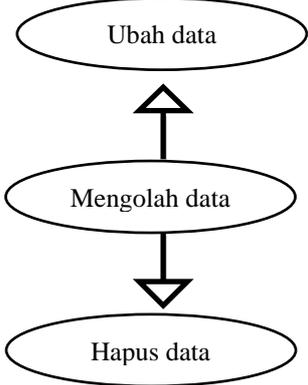
Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

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

Tabel 2.3. Simbol *Use Case Diagram*

Simbol	Deskripsi
<p data-bbox="435 517 544 544"><i>Use case</i></p> 	<p data-bbox="799 517 1361 689">Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal di awal frase nama <i>use case</i>.</p>
<p data-bbox="435 752 592 779">Aktor / <i>actor</i></p> 	<p data-bbox="799 752 1361 1003">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, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
<p data-bbox="312 1039 595 1066">Asosiasi / <i>association</i></p> 	<p data-bbox="799 1039 1361 1144">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="312 1296 520 1323">Ektensi / <i>extend</i></p> <p data-bbox="312 1368 472 1395"><<extend>></p> 	<p data-bbox="799 1296 1361 1435">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.</p> 

Tabel 2.3 Lanjutan

<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum)</p> 
<p>Menggunakan / <i>include / uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambah ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i>. Ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> yang ditambahkan dijalankan. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.</p>

Sumber: Rosa (2011: 135)

3. Sequence Diagram

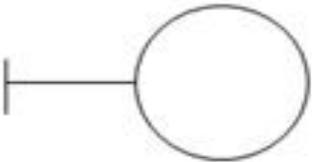
Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

Tujuan penggunaan sequence diagram adalah sebagai berikut:

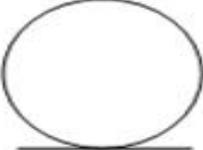
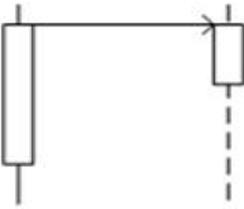
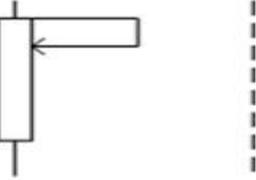
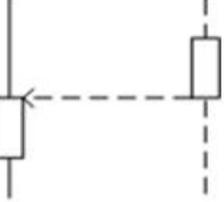
1. Mengkomunikasikan requirement kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model design
2. Merupakan diagram yang paling cocok untuk mengembangkan model deskripsi use-case menjadi spesifikasi design.
3. Analisa dan desain, memfokuskan pada identifikasi method didalam sebuah system.

Berikut adalah simbol-simbol yang ada pada *Diagram Activity*:

Tabel 2.4 Diagram Sequence

Simbol	Deskripsi
Aktor 	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan system
<i>Boundary</i> 	Menggambarkan interaksi antara satu atau lebih actor dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.

Tabel 2.4 Lanjutan

<p><i>Control</i></p> 	<p>Menggambarkan penghubung antara boundary dengan table</p>
<p><i>Entity</i></p> 	<p>Menggambarkan hubungan kegiatan yang akan dilakukan</p>
<p><i>Object Message</i></p> 	<p>Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi</p>
<p><i>Message to Self</i></p> 	<p>Mengambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi</p>
<p><i>Return Message</i></p> 	<p>Menggambarkan pesan/hubungan antar obyek, yang menunjukan urutan kejadian yang terjadi.</p>
<p><i>Life Line</i></p> 	<p>Menggambarkan tempat mulai dan berakhirnya sebuah pesan</p>

2.2. Tinjauan Teori Khusus

Berikut adalah uraian tentang teori yang dibahas secara khusus yang dijadikan sebagai teori yang memperkuat penelitian ini, diantaranya adalah:

2.2.1. Penjualan

Dalam dunia perdagangan saat ini, toko barang kebutuhan sehari-hari dengan ruangan yang tidak terlalu luas (minimarket) bukan lagi merupakan istilah asing bagi masyarakat umum, terutama yang tinggal dikota-kota besar. Minimarket merupakan perantara pemasar antara produsen dan konsumen akhir dimana aktivitasnya adalah melaksanakan penjualan eceran

Menurut Kotler (2010: 84) pengertian minimarket adalah toko yang mengisi kebutuhan masyarakat akan warung yang berformat modern yang dekat dengan permukiman penduduk sehingga dapat mengungguli toko atau warung.

Sebagai minimarket yang menyediakan barang kebutuhan sehari-hari suasana dan keseluruhan minimarket sangat memerlukan suatu penanganan yang profesional dan khusus agar dapat menciptakan daya tarik pada minimarket. Tata letak minimarket dapat mempengaruhi sirkulasi kembali untuk berbelanja. Kadang-kadang suasana yang nyaman bersih dan segar lebih diutamakan dari pada hanya sekedar harga rendah yang belum tentu dapat menjamin kelangsungan hidup dari minimarket tersebut. Salah satu usaha yang dilakukan oleh pengusaha minimarket ini untuk menarik konsumen agar melakukan pembelian yaitu melalui promosi.

Pengertian penjualan menurut Kotler (2010: 457): "Penjualan merupakan sebuah proses dimana kebutuhan pembeli dan kebutuhan penjualan dipenuhi, melalui antar pertukaran informasi dan kepentingan", jadi konsep penjualan adalah cara untuk mempengaruhi konsumen untuk membeli produk yang ditawarkan. Konsep penjualan adalah gagasan bahwa konsumen tidak akan membeli cukup banyak produk perusahaan kecuali jika perusahaan tersebut melakukan usaha penjualan dan promosi dalam skala besar.

Pentingnya promosi penjualan karena promosi penjualan adalah kegiatan pemasaran selain *personal selling*, periklanan dan publisitas yang mendorong efektifitas pembelian konsumen dan pedagang dengan menggunakan alat peragaan, pameran, demonstrasi dan sebagainya yang ditujukan untuk meningkatkan penjualan barang tertentu.

Secara umum pelaksanaan penjualan terdiri dari beberapa jenis metode penjualan yaitu:

1. Penjualan tunai

Dalam transaksi penjualan tunai, pembeli langsung melunasi barang yang dibelinya pada saat barang tersebut diserahkan. Penjualan tunai menambah kas dan mengurangi penjualan. Dalam transaksi tunai pembeli langsung membayar lunas atau cash atas kepemilikan barang atau jasa yang dibelinya.

2. Penjualan Kredit

Yaitu penjualan barang dimana pembayarannya dilakukan setelah barang diterima dengan cara mengangsur. Pada penjualan kredit dimulai dari pesanan penjualan dari pelanggan oleh pelanggan persetujuan kredit dan pengiriman barang. Organisasi pada penjualan kredit meliputi persetujuan kredit, yang

bertugas untuk pemberian kredit kepada pelanggan dengan melihat kartu pembantu piutang pelanggan yang hendak membeli secara kredit dengan demikian dapat diketahui kreabilitas pelanggan yang bersangkutan.

2.2.2. *Visual Basic.NET 2008*

Menurut Wahana Komputer (2011: 1), *Visual Basic.NET* merupakan salah satu bahasa pemrograman yang dapat digunakan untuk membuat program aplikasi. Bahasa pemrograman ini menyediakan beberapa *tool* untuk otomatisasi proses pengembangan, yaitu *visual tool* yang digunakan untuk melakukan beberapa operasi pemrograman dan desain umum dan juga fasilitas-fasilitas lain yang dapat menunjang dalam pemrograman.

Visual Basic.NET merupakan bagian dari Visual Studio.NET. *Visual Studio.NET* merupakan suatu lingkungan (*Environment*) terintegrasi untuk membangun dan melakukan uji coba (*Testing and Debugging*) berbagai macam aplikasi. Diantaranya adalah aplikasi *Windows, web, control, class* serta aplikasi *console*. Dengan Visual Studio.NET, kita akan dapat lebih mudah membuat aplikasi karena dalam *Visual Studio.NET* ada dukungan fasilitas baru yang ditambahkan, antara lain *Integrated Development Environment (IDE), Microsoft Intellisense, debugging* yang lebih baik dan kemampuan dalam *XML Web Services*. Dengan *Visual Basic.NET*, user dapat mendesain, mengeksekusi dan *men-debug* program aplikasi yang telah dibuat.

Keuntungan dengan adanya *framework* ini pembuatan program dari *Visual Basic* terkesan lebih mudah dan singkat karena dalam *framework* telah terbungkus berbagai komponen dan *class* yang siap pakai sehingga kita tidak

perlu menulis kode yang terlalu panjang untuk melakukan berbagai fungsi tertentu.

Kekurangannya tentu saja program yang dibuat akan memakan lebih banyak tempat (memori) baik memori *hardisk* maupun *Ram* karena *framework* juga harus didistribusikan dengan terpasang pada komputer target.

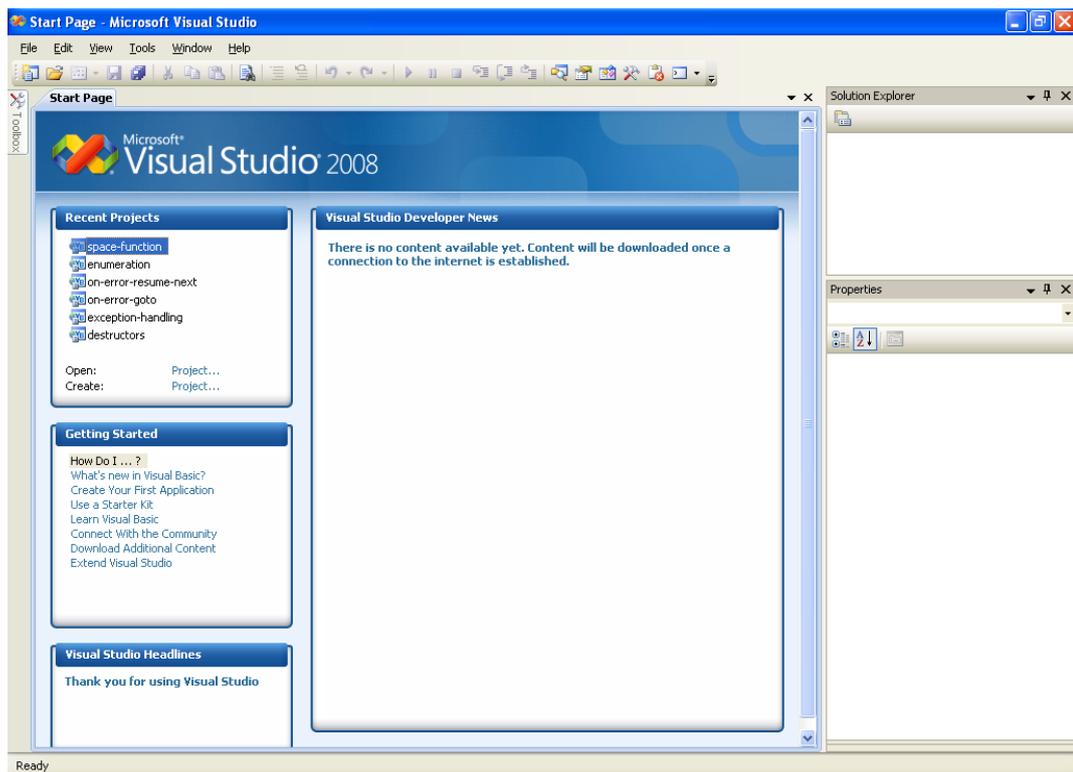
VB.Net 2008 adalah salah satu bahasa pemrograman yang berbasis OOP atau dengan kata lain suatu gaya pemrograman yang berorientasi pada objek. Ketika kita berbicara mengenai bahasa pemrograman berbasis objek, dimana hal yang paling penting dan mendasar dari istilah tersebut adalah kata objek. Dalam konteks pemrograman *VB.Net 2008* arti objek secara teoritis adalah sebuah struktur dalam bahasa pemrograman yang membungkus bahasa dan fungsinya sebagai satu kesatuan yang hanya dapat diakses secara *public* melalui antarmuka struktur pemrograman (*property, metode, even*).

Budiharto (2010:1) menyebutkan, *VB.Net 2008* ialah bahasa pemrograman terbaru yang memudahkan programmer VB 6/VB.Net beralih ke *VB.NET 2008*. Setelah *Visual Basic versi 6.0*, *Microsoft* melakukan perubahan besar pada bahasa pemrograman *Visual Basic versi* selanjutnya. Dimana ditambahkan suatu pustaka-pustaka yang terangkai menjadi suatu kesatuan yang disebut dengan *.Net (dotnet)* framework. Selain itu ditambahkan (diperkuat) pula permodelan pemrograman berorientasi objek yang disebut *Object Oriented Programming* atau sering disingkat dengan OOP.

Budiharto (2010:3-4) juga menyebutkan alasan penting lainnya untuk melakukan migrasi *VB.NET 2008*, yaitu:

- 1) *VB.Net 2008* mengatasi semua masalah yang sulit di sekitar pengembangan aplikasi berbasis *Windows* dan mengurangi penggunaan aplikasi lainnya serta versi komponen, bahkan mewarisi sifat *C++* dan berbau *Java*.
- 2) *VB.Net 2008* memiliki fasilitas penanganan *bug* yang hebat dan *real time background compiler* yang mengakibatkan *developer visual C#* dapat mengetahui kesalahan kode yang terjadi secara *up-to-date*.
- 3) *Windows Form designer* memungkinkan *developer* memperoleh aplikasi desktop dalam waktu yang singkat.
- 4) Bagi *developer*, *VB.Net 2008* menyediakan model pemrograman data akses *ActiveX Data Object (ADO)* yang sudah dikenal dan diminati, ditambah XML baru yang berbasis *Microsoft ADO.Net* dengan *ADO.Net*, *developer* akan memperoleh akses ke komponen yang lebih *powerfull*, seperti *control DataSet*.
- 5) secara mendasar *.NET Framework* dibuat untuk dipasangkan dengan *Windows 2003* dengan keunggulan memonitor kelalaian dari aplikasi yang sedang berjalan, dan mengisolasi setiap aplikasi yang sedang berjalan dan mengisolasi setiap aplikasi.
- 6) *Developer* dengan berbagai latar belakang dapat dengan segera menguasai *.NET* karena kemudahan dan kemiripan kode yang ditawarkannya.
- 7) *Deployment/Penyebaran* yang mudah, baik untuk aplikasi windows maupun aplikasi *web* karena sudah tersedia *wizard* atau tool secara khusus dengan fasilitas tambahan yang menarik. *Tool* canggih ini tidak tersedia pada aplikasi sebelumnya bahkan pada bahasa pemrograman lain.

- 8) Integrasi dengan sistem yang sudah ada sangat mudah, *Net Framework com* memungkinkan Anda berinteraksi dan dengan sistem yang sudah ada menggunakan *XML web Service*. Terakhir, *Visual Studio Upgrade tool* yang tersedia pada *Visual Studio.NET* dan *Java Language Convention Assistant* membantu Anda mengkonversi *Visual Basic 6* dan *Visual J++* agar berjalan pada *.Net Framework*.
- 9) Mendukung lebih dari 20 bahasa pemrograman, *NET Framework* mendukung integrasi lebih dari 20 bahasa pemrograman yang tidak terbayang sebelumnya. Memungkinkan pengembang memilih bahasa pemrograman yang tepat sesuai latar belakang pemrogramannya.



Gambar 2.4. *Visual Basic.Net 2008*

2.2.3. Database Access

Microsoft Access adalah sekumpulan objek yang terdiri dari tabel, query, form, report, pages, macro dan module. Objek-objek ini ditampung dalam satu wadah atau *database*. Database di *Microsoft access* selain terdiri dari objek atau *objects* terdapat juga *groups*.

Microsoft Access adalah suatu aplikasi yang dapat membantu kita membuat sebuah aplikasi *database* dalam waktu yang relatif singkat. Biasanya digunakan untuk pembuatan aplikasi-aplikasi yang kecil. Misalnya Program untuk kasir di koperasi, penjualan untuk toko dan dipakai di perusahaan kecil lainnya.

Microsoft Access Memang harus diakui, aplikasi *database Microsoft Access* adalah satu-satunya paket dari *Microsoft Office* yang dipandang sebelah mata oleh para *user*, tidak terkecuali penulis sendiri. *Microsoft Access* adalah aplikasi yang jarang digunakan, karena dianggap kalah bersaing jika dibandingkan dengan *MySQL*, *Oracle* ataupun aplikasi *database open source* seperti *postgreSQL*. Selama 13 tahun lamanya sejak *Microsoft Access* resmi dipasarkan perubahan-perubahan fitur dan komponen pendukung terlihat tidak cukup memuaskan bagi para *user*.

Menurut Frieyadie (2015: 56), *Microsoft Access 2013* merupakan paket dari paket *Microsoft Office 2013* untuk pengolahan data, terutama *database*. Bekerja dengan *Microsoft Access* dapat mencakupi 3 (tiga) hal, selain pengolahan *database*, yaitu

- a. Pemrograman operasi dengan *macro*.
- b. Pengolahan data dengan *Structured Query Language (SQL)*.

- c. Pemrograman dengan modul yang didukung oleh VBA (*Visual Basic For Application*), yakni fasilitas pemrograman dengan bahasa *Visual Basic*.

Secara umum struktur *database* pada *Ms. Access* adalah:

1. *Table*.

Merupakan tempat penyimpanan data yang akan diperoleh dalam bentuk format baris dan kolom.

2. *Queries*.

Digunakan untuk menyaring data dengan urutan yang kita inginkan.

3. *Form*.

Dipergunakan untuk menampilkan data, memasukkan data (*input*) dan mengedit data.

4. *Report*.

Dipergunakan untuk *output* (menampilkan/mencetak) informasi dalam bentuk laporan.

5. *Pages*.

Dipergunakan untuk membuat halaman *web* berupa data *access page*.

6. *Macros*.

Untuk mengotomatisasi perintah-perintah yang sering kita gunakan dalam mengolah data.

7. *Modules*.

Merupakan program kecil atau prosedur yang kegunaannya adalah untuk perancangan modul aplikasi pengolahan data bertingkat lanjut.

Microsoft Access 2013 merupakan penyempurnaan atau pengembangan dari versi sebelumnya. Beberapa hal kelebihan atau penyempurnaan di dalam *Access 2003* antara lain :

- a. Fasilitas sub *datasheets* baru memungkinkan kita melihat *record* yang berkaitan dengan sebuah *datasheet*.
- b. Dapat berfungsi sebagai *font end* untuk data *Microsoft SQL server*.
- c. Dilengkapi dengan *Tool web* yang lebih canggih.
- d. Memungkinkan kita untuk mempublikasikan data ke *Web*, namun kita tidak dapat melihat hasil halaman *Web* di *Access*.
- e. Adanya fasilitas bantuan *page wizard*.

Microsoft Access 2013 merupakan salah satu *software database* yang berjalan di bawah sistem *Windows*, karena disamping *Microsoft Access*, masih banyak *software-software database* lainnya yang dapat kita temukan. Dengan *Microsoft Access 2013*, kita dapat merancang, membuat dan mengelola *database* dengan cara mudah dan cepat.