

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Era globalisasi sekarang ini merupakan era modernisasi yang akan membawa ke arah kemajuan segala bidang terutama bidang teknologi. Globalisasi juga menimbulkan dampak adanya kebutuhan yang besar terhadap kemampuan dan penguasaan teknologi informasi. Munculnya kebutuhan akan teknologi informasi merupakan hal yang wajar, karena teknologi informasi merupakan salah satu sarana untuk mencapai kemajuan bangsa.

Sistem Informasi semakin dibutuhkan oleh organisasi maupun perusahaan, banyaknya pekerjaan manusia yang dapat dibantu oleh komputer sehingga dapat menjadi lebih produktif dan memberikan hasil kerja yang lebih baik dan cepat, khususnya dalam meningkatkan kualitas dan efektifitas aliran informasi dalam perusahaan dan menciptakan aliansi atau kerjasama dengan baik.

Setiap instansi memiliki cara berbeda-beda pula, ada yang sudah menggunakan komputer tetapi juga ada yang masih menggunakan cara yang manual. Lajunya dunia teknologi informasi, mengakibatkan sistem-sistem informasi yang lama menjadi tertinggal. Di sini terlihat jelas bahwa permasalahan yang sering timbul terjadi adalah dalam penyajian data dan

informasi yang kurang tepat dan akurat serta membutuhkan waktu yang sangat lama.

Sehubungan dengan permasalahan yang terjadi, maka PT.SIBELA berusaha menyajikan informasi penjualan harga *sparepart* kepada pelanggan secara cepat dan tepat, sehingga pengelolaan *sparepart* menjadi lebih mudah dalam pendataan antara barang masuk dan keluar. Adapun PT.SIBELA masih menggunakan cara manual dalam sistem penjualan dan pendataan barang, yaitu dengan cara pendataan barang masuk dan keluar masih dicatat dibuku besar stok barang dan di dalam transaksi juga masih menggunakan pencatatan dalam buku transaksi dan menggunakan nota, sehingga mungkin saja terjadi kesalahan perhitungan yang kurang akurat dalam mengarsip hasil laporan. dalam pencarian data stok barang juga memerlukan waktu yang lama karena masih menggunakan pencarian manual dengan mencari data buku stok. Oleh sebab itu penulis berkeinginan untuk merancang,

“Sistem Informasi Inventori *Sparepart* menggunakan MS.ACCESS dengan studi kasus di PT.SIBELA ”.

## **1.2. Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan tersebut, maka dapat dirumuskan permasalahan sebagai berikut :

- a. Bagaimana membangun sistem informasi inventori *sparepart* pada PT.SIBELA agar dapat lebih efektif dan cepat dalam pencarian *sparepart* dan pelayanannya ?
- b. Bagaimana membangun system yang dapat membantu pengguna mempertahankan stok barang ?

## **1.2. Tujuan Penulisan**

Tujuan dilakukannya penelitian ini diantaranya adalah sebagai berikut.

1. Merancang sistem informasi penjualan dan pendataan *sparepart* pada PT.SIBELA agar suatu pelayanan penjualan *sparepart* tidak membutuhkan waktu yang lama.
2. Mengelolah pendataan stok barang dan jenis *sparepart* dalam bentuk sistem informasi.

## **1.3. Ruang Lingkup Studi Kasus**

Ruang lingkup penulisan ini ada pada PT.SIBELA, mengingat PT. SIBELA masih menggunakan cara manual dalam sistem penjualan dan pendataan barang, yaitu dengan cara pendataan barang masuk dan keluar masih dicatat dibuku besar stok barang dan di dalam transaksi juga masih menggunakan pencatatan dalam buku transaksi dan menggunakan nota, sehingga ‘Aplikasi Sistem Informasi Inventori *Sparepart*’ sangat di perlukan.

#### **1.4. Kegunaan Studi Kasus**

Kegunaan Studi Kasus adalah :

1. Menerapkan ilmu pengetahuan yang di dapat selama menimba ilmu pengetahuan dari perkuliahan pada dunia kerja.
2. Menambah pengetahuan, wawasan dan pengalaman.
3. Dan bagi kampus bisa menjadi bahan referensi atau acuan bagi penulis atau peneliti berikutnya.

## **BAB II**

### **LANDASAN TEORI**

Dalam bab ini akan dijelaskan tentang landasan teori yang di pergunakan dalam penulisan studi kasus.

#### **2.1. Aplikasi**

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user.

Aplikasi juga merupakan sebuah kata serapan yang berasal dari bahasa Inggris, yaitu *application*. Mengalami proses serapan kedalam bahasa Indonesia menjadi “aplikasi”, arti dari kata ini sendiri adalah sebuah aksi yang dilakukan oleh suatu hal yang berasal dari hasil sebuah analisa, strategi, perhitungan matang dari orang yang membuatnya. Aplikasi ini di jaman sekarang memang lebih cenderung digunakan sebagai sebutan untuk sebuah

teknologi, yaitu lebih mudahnya kita sebut istilah komputer. Karena teknologi jaman sekarang sering sekali orang menyebutkan kata “aplikasi” disetiap program-program yang telah dibuat para perusahaan besar untuk kebutuhan semua lapisan masyarakat.

Jika kita mendasari kata “aplikasi” dengan pengertian dunia teknologi komputerisasi saat ini, pengertian dari aplikasi adalah sebuah perangkat lunak pada bagian teknologi komputer untuk dipergunakan dan dimanfaatkan sebagai mediasi pelaksana sebuah tugas dari hasil perintah penggunanya. Contohnya adalah aplikasi pemutar media untuk lebih spesifiknya misal mendengarkan musik, program ini dibuat sebagai mediator kita (sebagai pengguna) untuk membantu menerapkan keinginan mendengarkan musik di sebuah komputer, kemudian kita aktifkan aplikasi tersebut kemudian memilih yang diinginkan oleh kita dan pada akhirnya aplikasi tersebut akan membantu Anda mengaktifkan segala data yang diperlukan dan tersedia oleh komputer tersebut agar bisa menyalakan program suara, mengaktifkan program waktu, serta menjalankan lagu yang telah kita pilih.

## **2.2. Microsoft Access**

*Microsoft Access* adalah suatu program aplikasi basis data komputer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar.

**Microsoft Access** (atau **Microsoft Office Access**) juga adalah sebuah program aplikasi basis data komputer relasional yang ditujukan untuk kalangan rumahan dan perusahaan kecil hingga menengah. Aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint. Aplikasi ini menggunakan mesin basis data Microsoft Jet Database Engine, dan juga menggunakan tampilan grafis yang intuitif sehingga memudahkan pengguna.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemrograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

Microsoft Access digunakan kebanyakan oleh bisnis-bisnis kecil dan menengah, di dalam sebuah organisasi yang kecil bahkan mungkin juga digunakan oleh perusahaan yang cukup besar, dan juga para programmer untuk

membuat sebuah sistem buatan sendiri untuk menangani pembuatan dan manipulasi data. Access juga dapat digunakan sebagai sebuah basis data untuk aplikasi Web dasar yang disimpan di dalam *server* yang menjalankan Microsoft Internet Information Services (IIS) dan menggunakan Microsoft Active Server Pages (ASP). Meskipun demikian, penggunaan Access kurang disarankan, mengingat telah ada Microsoft SQL Server yang memiliki kemampuan yang lebih tinggi.

Beberapa pengembang aplikasi profesional menggunakan Microsoft Access untuk mengembangkan aplikasi secara cepat (digunakan sebagai *Rapid Application Development/RAD tool*), khususnya untuk pembuatan purwarupa untuk sebuah program yang lebih besar dan aplikasi yang berdiri sendiri untuk para *salesman*.

Microsoft Access kurang begitu bagus jika diakses melalui jaringan sehingga aplikasi-aplikasi yang digunakan oleh banyak pengguna cenderung menggunakan solusi sistem manajemen basis data yang bersifat klien/server. Meskipun demikian, tampilan muka Access (*form, report, query*, dan kode Visual Basic) yang dimilikinya dapat digunakan untuk menangani basis data yang sebenarnya diproses oleh sistem manajemen basis data lainnya, seperti halnya Microsoft Jet Database Engine (yang secara *default* digunakan oleh



Microsoft Access), Microsoft SQL Server, Oracle Database, dan beberapa produk lainnya yang mendukung ODBC.

Salah satu keunggulan Microsoft Access dilihat dari perspektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statemen-statement SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

Microsoft SQL Server Desktop Engine (MSDE) 2000, yang merupakan sebuah versi mini MAL dari Microsoft SQL Server 2000, dimasukkan ke dalam Office XP Developer Edition dan dapat digunakan oleh Microsoft Access sebagai alternatif dari Microsoft Jet Database Engine.

Tidak seperti sebuah sistem manajemen basis data relasional yang komplit, Microsoft JET Database Engine tidak memiliki fitur *trigger* dan *stored procedure*. Dimulai dari Microsoft Access 2000 yang menggunakan Microsoft Jet Database Engine versi 4.0, ada sebuah sintaksis yang mengizinkan pembuatan kueri dengan beberapa parameter, dengan sebuah cara seperti halnya sebuah *stored procedure*, meskipun prosedur tersebut dibatasi hanya untuk

sebuah pernyataan tiap prosedurnya. Access juga mengizinkan *form* untuk mengandung kode yang dapat dieksekusi ketika terjadi sebuah perubahan terhadap tabel basis data, seperti halnya *trigger*, selama modifikasi dilakukan hanya dengan menggunakan form tersebut, dan merupakan sesuatu hal yang umum untuk menggunakan kueri yang akan diteruskan (*pass-through* dan teknik lainnya di dalam Access untuk menjalankan *stored procedure* di dalam RDBMS yang mendukungnya.

Dalam berkas Access Database Project (ADP) yang didukung oleh Microsoft Access 2000 dan yang selanjutnya, fitur-fitur yang berkaitan dengan basis data berbeda dari versi format/struktur data yang digunakan Access (\*.MDB), karena jenis berkas ini dapat membuat koneksi ke sebuah basis data MSDE atau Microsoft SQL Server, ketimbang menggunakan Microsoft JET Database Engine. Sehingga, dengan menggunakan ADP, adalah mungkin untuk membuat hampir semua objek di dalam server yang menjalankan mesin basis data tersebut (tabel basis data dengan *constraints* dan *trigger*, *view*, *stored procedure*, dan *UDF*). Meskipun demikian, yang disimpan di dalam berkas ADP hanyalah *form*, *report*, *macro*, dan modul, sementara untuk tabel dan objek lainnya disimpan di dalam server basis data yang membelakangi program tersebut.

## **2.3. Pemodelan Apalikasi**

### **2.3.1. Diagram Konteks**

Diagram konteks merupakan tingkat tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu entitas-entitas eksternal serta aliran data-aliran data menuju dan dari sistem diketahui menganalisis dari wawancara dengan user dan sebagai hasil analisis dokumen.

Diagram konteks dimulai dengan penggambaran terminator, aliran data, aliran kontrol penyimpanan, dan proses tunggal yang menunjukkan keseluruhan sistem. Bagian termudah adalah menetapkan proses yang hanya terdiri dari satu lingkaran dan diberi nama yang mewakili sistem.

Nama hal ini dapat menjelaskan proses atau pekerjaan atau dalam kasus ekstrim berupa nama perusahaan yang dalam ini mewakili proses yang dilakukan keseluruhan organisasi. Terminator ditunjukkan dalam bentuk persegi panjang dan berkomunikasi langsung dengan sistem melalui aliran data atau penyimpanan eksternal.

Antar terminator tidak diperbolehkan komunikasi langsung. Pada kenyataannya hubungan antar terminator dilakukan, tetapi secara definitif karena terminator adalah bagian dari lingkungan, maka tidak relevan jika dibahas dalam diagram konteks.

Jika terdapat banyak terminator yang mempunyai banyak masukan dan keluaran diperbolehkan untuk digambarkan lebih dari satu kali sehingga mencegah penggambaran yang terlalu rumit, dengan ditandai secara khusus untuk menjelaskan bahwa terminator yang dimaksud adalah identik. Tanda tersebut dapat berupa asterik (\*) atau pagar (#).

Jika terminator mewakili individu sebaiknya diwakili oleh peran yang dimainkan personil tersebut. Alasan pertama adalah personil yang berfungsi untuk melakukan itu dapat berganti sedang diagram konteks harus tetap akurat walaupun personil berganti. Alasan kedua adalah seorang personil dapat memainkan lebih dari satu peran.

Karena fokus utama adalah mengembangkan model, maka penting untuk membedakan sumber (resource) dan pelaku (handler). Pelaku adalah mekanisme, perangkat atau media fisik yang mentransportasikan data dari sistem, karena pelaku seringkali familier dengan pemakai dalam implementasi sistem berjalan, maka sering

menonjol sebagai sesuatu yang harus digambarkan lebih dari sumber data itu sendiri. Sedangkan sistem baru dengan konsep pengembangan teknologinya membuat pelaku menjadi sesuatu yang tidak perlu digambarkan.

### **2.3.2. Data Flow Diagram**

Data Flow Diagram (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data system, yang penggunaannya sangat membantu untuk memahami system secara logika, terstruktur dan jelas.

Data flow diagram (DFD) merupakan alat bantu dalam menggambarkan atau menjelaskan DFD ini sering disebut juga dengan nama Bubble chart, bubble diagram, model proses, diagram alur kerja, atau model fungsi.

Tujuan Data Flow Diagram (DFD) adalah Memberikan indikasi mengenai bagaimana data ditransformasi pada saat data bergerak melalui system, menggambarkan fungsi-fungsi dan sub fungsi yang mentransformasi aliran data.

Fungsi atau manfaat Data flow diagram (DFD) adalah alat pembuatan model yang memungkinkan professional system untuk menggambarkan system sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi system merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh system. Dengan kata lain, DFD adalah alat pembuat model yang memberikan penekanan hanya pada fungsi system.

DFD merupakan alat perancangan system yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan system yang mudah dikomunikasikan oleh professional system kepada pemakai maupun pembuat program.

### **2.3.3. Entity Relationship Diagram (ERD)**

ERD adalah kependekan dari Entity Relationship Diagram, yang apabila diterjemahkan secara bebas adalah bagan atau sketsa suatu hubungan entitas. Sedangkan entitas dalam ERD diagram adalah suatu objek yang benar-benar sungguh ada, yang dapat

didefinisikan dalam lingkungan pemakai serta memiliki sesuatu yang penting bagi pemakai dalam konteks pengembangan sistem yang direncanakan.

Dalam definisi lain ERD diagram adalah alat peraga atas desain database yang menjadi dasar sistem informasi yang tengah dikembangkan. Jadi, sampai di sini dapat kita simpulkan bahwa untuk melakukan pengembangan sistem informasi diperlukan desain-desain database sebagai pola awal pengembangan.

Keberadaan ERD adalah sebagai alat peraga atas segala desain sehingga lebih mudah untuk dipahami oleh user pengembang. ERD merupakan model data yang pada gilirannya dapat digunakan untuk spesifikasi database.

Komponen Penyusun ERD Diagram adalah Dalam memodelkan suatu desain menjadi ERD diperlukan beragam teknik, salah satunya pembuat ERD harus membuat gambar simbol dan beberapa notasi yang menunjukkan makna desain dalam rupa ERD. Ada tiga komponen utama dalam pembuatan ERD, yaitu:

**2.3.3.1.** Entitas atau Entity (sesuatu yang sungguh ada): merupakan suatu objek unik yang berbeda dengan yang lain, dan dapat diwujudkan ke dalam basis data. Objek ini memiliki karakter yang menarik untuk dijadikan suatu model. Dalam

proses visualisasi ERD, simbol entitas biasanya diwujudkan dalam bentuk persegi panjang dalam posisi horizontal.

**2.3.3.2.** Atribut: digunakan untuk memberikan informasi yang lebih rinci tentang segala jenis entitas, sebab setiap entitas memiliki elemen yang berfungsi untuk memberikan deskripsi karakteristik. Atribut selalu digambarkan dengan simbol elips. Atribut terbagi ke dalam lima jenis, yaitu: atribut key, atribut simple, atribut multivalued, atribut composite, dan atribut derivatif.

**2.3.3.3.** Relasi atau Relationship (hubungan): hubungan di sini diartikan sebagai pola yang dipergunakan dalam ERD dalam menghubungkan antar entity, biasanya digambarkan dalam bentuk garis lurus di mana kedua ujungnya menyentuh simbol-simbol entity sebagai tanda penghubung. Demikian penjelasan yang dapat saya berikan, ERD diagram adalah sekumpulan simbol yang terintegrasi dalam sebuah diagram entitas atas sebuah sistem.