

SISTEM INFORMASI PENGELOLAAN DATA PENJUALAN PRODUK KALBE DI PT. ABC DENGAN MENGGUNAKAN VISUAL STUDIO.NET DAN METODE WATERFALL

Agus Herwanto

Fakultas Ilmu Komputer, Universitas Esa Unggul
Jalan Arjuna Utara No. 9, Kebon Jeruk, Jakarta, 11510
agus.herwanto@esaunggul.ac.id

Abstract

PT. ABC is a company engaged in the sale of health products. In managing the sales data process, it is still not effective, that is, still relying on papers for each process. For the sales process, there are often mistakes in recording codes and prices, especially if there is an increase in prices, this can cause losses for PT. A B C. It also often results in loss of sales notes due to a poor filing system. Need to develop from the previous system using a database-based system. The design of this sales data management system applies the waterfall method in which the preparation of the system starts from the analysis, design, coding, testing and maintenance. The reason the authors use this method is the quality of the resulting system will be good, this is due to the gradual implementation. Each process has its own specifications, so that a system can be developed in accordance with what is desired. The result of applying this methodology is an information system for managing health product sales data at PT. ABC which will simplify and speed up the sales process by implementing a desktop-based management system using Visual Studio. Net and the SQL Server database.

Keywords: *Data management, sales, visual studio.net, waterfall.*

Abstrak

PT. ABC adalah sebuah perusahaan yang bergerak di bidang penjualan produk kesehatan. Dalam melakukan proses pengelolaan data penjualan masih belum efektif yaitu masih mengandalkan kertas-kertas untuk setiap prosesnya. Untuk proses penjualan, sering kali terjadi kesalahan-kesalahan pencatatan kode dan harga terutama bila terjadi kenaikan harga, hal ini dapat menyebabkan kerugian bagi PT. ABC. Sering juga terjadi kehilangan nota-nota penjualan dikarenakan sistem pengarsipan yang kurang baik. Perlu dilakukan pengembangan dari sistem sebelumnya menggunakan sistem yang berbasis *database*. Perancangan sistem pengelolaan data penjualan ini menerapkan metode *waterfall* di mana penyusunan sistem dimulai dari analisis, desain, *coding*, *testing* dan pemeliharaan. Alasan penulis menggunakan metode ini adalah kualitas dari sistem yang dihasilkan akan baik, ini dikarenakan oleh pelaksanaannya secara bertahap. Setiap proses memiliki spesifikasinya sendiri, sehingga sebuah sistem dapat dikembangkan sesuai dengan apa yang dikehendaki. Hasil dari penerapan metodologi ini adalah sebuah sistem informasi pengelolaan data penjualan produk kesehatan pada PT. ABC yang akan mempermudah dan mempercepat proses penjualan dengan menerapkan sistem pengelolaan berbasis *desktop* menggunakan Visual Studio.Net dan *database* SQL Server.

Kata kunci : Pengelolaan data, penjualan, visual studio.net, waterfall.

Pendahuluan

Seiring dengan perkembangan dan kemajuan zaman serta berkembangnya peradaban manusia, aplikasi komputer secara terus-menerus mengalami peningkatan dalam pengembangannya. Hal ini juga memicu pesatnya perkembangan teknologi informasi untuk dapat memberikan kemudahan dan fasilitas-fasilitas yang dapat mendukung manusia dalam upaya untuk menyelesaikan tugasnya. Penggunaan komputer sebagai alat bantu penyelesaian pekerjaan di

bidang teknologi sistem informasi kian marak dan berkembang di segala bidang. Salah satunya dalam bidang penjualan yaitu dalam hal mengoperasikan proses pengolahan data penjualan. Salah satu keunggulan komputer yaitu dapat menelusuri data yang sedemikian banyak dan rumit dengan cepat, akurat dan dapat memperlancar pekerjaan. Komputer merupakan alat pengolahan data dengan akses yang tepat dan teliti dibandingkan dengan akses kerja otak manusia. Dengan adanya sistem komputerisasi, dapat membantu kelancaran dalam pengolahan data dan mempercepat pembuatan hasil laporan penjualan secara akurat, selain itu komputer memiliki keunggulan yang salah satunya adalah menyimpan data dalam jumlah besar.

PT. ABC adalah sebuah perusahaan yang bergerak di bidang penjualan produk kesehatan Kalbe seperti Fatigon, Promag, Mixagrip, Komix, Prenagen, Diabetasol, Diva dan lain-lain. Dalam sistem pengolahan data penjualan produk masih belum efektif yaitu masih mengandalkan kertas-kertas untuk setiap prosesnya. Untuk proses penjualan, sering kali terjadi kesalahan-kesalahan pencatatan kode dan harga terutama bila terjadi kenaikan harga. Hal ini dapat menyebabkan kerugian. Selain itu sering juga terjadi kehilangan nota-nota penjualan karena sistem pengarsipan yang kurang baik.

Penjualan

Penjualan adalah suatu sistem kegiatan pokok perusahaan untuk memperjual-belikan barang dan jasa yang perusahaan hasilkan. Dalam sistem penjualan terdapat dua macam, yaitu sistem penjualan tunai dan sistem penjualan kredit.

Sistem Penjualan Tunai

Sistem penjualan tunai merupakan sistem yang diberlakukan oleh perusahaan dalam menjual barang dengan cara mewajibkan pembeli untuk melakukan pembayaran harga terlebih dahulu sebelum barang diserahkan pada pembeli. Setelah pembeli melakukan pembayaran, baru barang diserahkan, kemudian transaksi penjualan dicatat. Prosedur yang membentuk sistem dalam sistem penjualan tunai adalah sebagai berikut :

- a. Prosedur *order* penjualan
Dalam prosedur ini, bagian penjualan menerima *order* dari pembeli dan membuat faktur penjualan tunai. Setelah pembeli membayar, bagian gudang mengirimkan barang kepada pembeli.
- b. Prosedur penerimaan kas
Dalam prosedur ini, bagian kasa menerima pembayaran dari pembeli dan memberikan tanda pembayaran (berupa pita register kas dan cap “Lunas” pada faktur penjualan tunai), kemudian pembeli mengambil barang.
- c. Prosedur pembungkusan dan penyerahan barang
Dalam prosedur ini pengiriman hanya menyerahkan barang kepada pembeli.
- d. Prosedur pencatatan penjualan tunai
Dalam prosedur ini, bagian akuntansi melakukan pencatatan transaksi penjualan tunai. .

Sistem Penjualan Kredit

Sistem penjualan kredit merupakan sistem penjualan di mana pembayarannya dilakukan setelah barang diterima pembeli. Jumlah dan jatuh tempo pembayaran disepakati oleh kedua pihak.

1. Prosedur Penjualan Kredit

Prosedur yang membentuk sistem dalam sistem penjualan kredit adalah sebagai berikut :

- a. Prosedur order penjualan
Dalam prosedur ini, fungsi penjualan menerima *order* dari pembeli.
- b. Prosedur persetujuan kredit
Dalam prosedur ini bagian penjualan meminta persetujuan kredit pada bagian kredit yaitu pada bagian keuangan.

- c. Prosedur pengiriman
Dalam prosedur ini, bagian pengiriman mengirimkan barang pada pembeli sesuai surat *order* pengiriman.
- d. Prosedur faktur atau penagihan
Dalam prosedur ini, bagian penagihan membuat faktur penjualan dan dikirim pada pembeli.
- e. Prosedur pencatatan akuntansi
Dalam prosedur ini, bagian akuntansi membuat kartu piutang berdasarkan faktur penjualan (Sujarweni, 2015, p. 89).

Sistem

Definisi sistem menurut buku sistem informasi dan implementasinya adalah sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware* dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain. *Software* mencakup semua perangkat lunak yang dibangun dengan bahasa pemrograman tertentu, pustaka untuk kemudian menjadi sistem operasi, aplikasi dan *driver*. Sistem operasi, aplikasi dan *driver* saling bekerja sama agar komputer dapat berjalan dengan baik. *Hardware* mencakup semua perangkat keras (*motherboard*, *processor*, VGA dan lainnya) yang disatukan menjadi sebuah komputer. Dalam konteks yang luas, bukan hanya sebuah komputer, namun sebuah jaringan komputer. *Brainware* mencakup kemampuan otak manusia yang mencakup ide, pemikiran, analisis di dalam menciptakan dan menggabungkan *hardware* dan *software*. Penggabungan *software* dan *hardware* dengan bantuan *brainware* inilah (melalui sejumlah prosedur) yang dapat menciptakan sebuah sistem yang bermanfaat bagi pengguna (Pratama, 2014, pp. 7–8).

Informasi

Informasi merupakan hasil pengolahan data dari satu atau berbagai sumber, yang kemudian diolah, sehingga memberikan nilai, arti dan manfaat. Proses pengolahan ini memerlukan teknologi. Berbicara mengenai teknologi memang tidak harus selalu berkaitan dengan komputer, namun komputer sendiri merupakan salah satu bentuk teknologi. Dengan kata lain, alat tulis dan mesin ketik pun dapat dimasukkan sebagai salah satu teknologi yang digunakan selain komputer dan jaringan komputer.

Pada proses pengolahan data, untuk dapat menghasilkan informasi, juga dilakukan proses verifikasi secara akurat, spesifik dan tepat waktu. Hal ini penting agar informasi dapat memberikan nilai dan pemahaman kepada pengguna. Pengguna dalam hal ini mencakup pembaca, pendengar dan penonton bergantung pada bagaimana cara pengguna tersebut menikmati sajian informasi dan melalui media apa informasi tersebut disajikan (Pratama, 2014, pp. 8–9).

Sistem Informasi

Menurut Pratama, sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak, perangkat keras, 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. Di dalamnya juga termasuk proses perencanaan, kontrol, koordinasi dan pengambilan keputusan. Sehingga, sebagai sebuah sistem yang mengolah data menjadi informasi yang akan disajikan dan digunakan oleh pengguna, maka sistem informasi merupakan sebuah sistem yang kompleks. Bukan hanya komputer saja yang bekerja (beserta *software* dan *hardware* di dalamnya), namun juga manusia (dengan *brainware* yang dimiliki). Manusia dalam hal ini menggunakan seluruh ide, pemikiran dan perhitungan untuk dituangkan ke dalam sistem informasi yang digunakan (Pratama, 2014, p. 10).

Komponen-Komponen pada Sistem Informasi

Menurut Pratama, komponen-komponen yang terdapat di dalam semua jenis sistem informasi mencakup tujuh poin. Berikut ketujuh komponen tersebut beserta dengan penjelasannya masing-masing, yaitu :

Masukan (Input)

Sebuah informasi berasal dari data yang telah diolah dan diverifikasi sehingga akurat, bermanfaat dan memiliki nilai. Komponen masukan ini berfungsi untuk menerima semua masukan dari pengguna. Masukan yang diterima dalam bentuk data. Data ini berasal dari satu maupun beberapa buah sumber.

Keluaran (Output)

Sebuah sistem informasi akan menghasilkan keluaran (*output*) berupa informasi. Komponen keluaran berfungsi untuk menyajikan hasil akhir ke pengguna sistem informasi. Informasi yang disajikan ini merupakan hasil dari pengolahan data yang telah dimasukkan sebelumnya. Pada komponen keluaran, informasi yang disajikan disesuaikan dengan data yang dimasukkan dan fungsionalitas dari sistem informasi bersangkutan.

1. Perangkat Lunak (*Software*)

Komponen perangkat lunak mencakup semua perangkat lunak yang digunakan di dalam sistem informasi. Adanya komponen perangkat lunak ini akan membantu sistem informasi di dalam menjalankan tugasnya dan untuk dapat dijalankan sebagaimana mestinya. Komponen perangkat lunak ini melakukan proses pengolahan data, penyajian informasi, penghitungan data dan lain-lain. Komponen perangkat lunak mencakup sistem operasi, aplikasi dan *driver*.

2. Perangkat Keras (*Hardware*)

Komponen perangkat keras mencakup semua perangkat keras komputer yang digunakan secara fisik di dalam sistem informasi, baik di komputer *server* maupun di komputer *client*. Komponen perangkat keras ini meliputi komputer *server* beserta komponen di dalamnya, komputer *desktop* beserta komponen di dalamnya, komputer jinjing beserta komponen di dalamnya, *mobile device* dan lain-lain. Termasuk juga di dalamnya *hub*, *switch*, *router* yang berperan di dalam jaringan komputer.

3. Basis Data (*Database*)

Mengingat bahwa sistem informasi menyajikan informasi yang berasal dari satu maupun beberapa data yang dimasukkan dan diolah, maka tentu diperlukan sebuah aplikasi untuk penyimpanan, mengolah dan menyajikan data dan informasi tersebut secara komputerisasi. Komponen basis data berfungsi untuk menyimpan semua data dan informasi ke dalam satu atau beberapa tabel. Setiap tabel memiliki *field* masing-masing. Setiap tabel memiliki fungsi penyimpanan masing-masing, serta antar tabel dapat juga terjadi relasi.

4. Kontrol dan Prosedur (*Control and Procedure*)

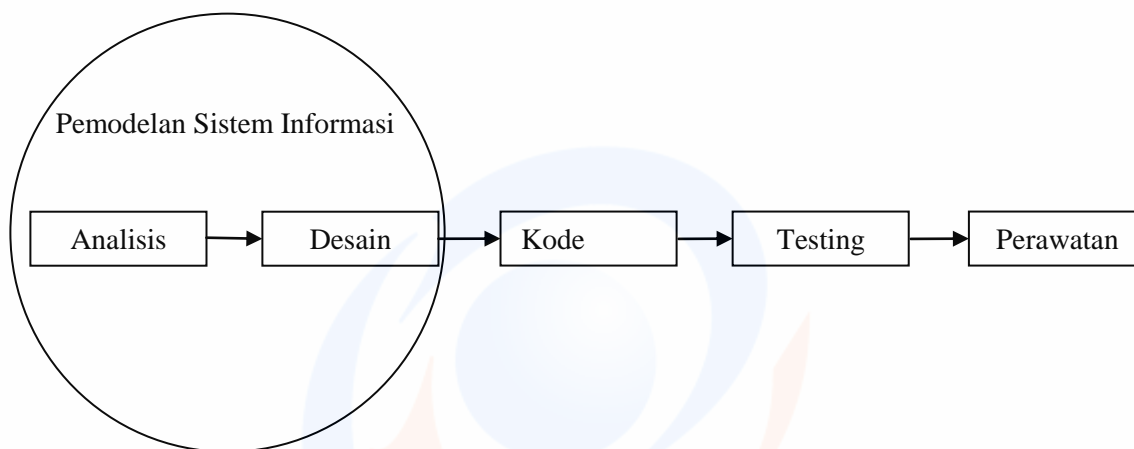
Kontrol dan prosedur adalah dua buah komponen yang menjadi satu. Komponen kontrol berfungsi untuk mencegah terjadinya beragam gangguan dan ancaman terhadap data dan informasi yang ada di dalam sistem informasi, termasuk juga sistem informasi itu sendiri beserta fisiknya. Perlu dilakukan pencegahan sejak dini terhadap kemungkinan ancaman dan gangguan yang dapat terjadi sewaktu-waktu. Kemungkinan tersebut antara lain dapat berupa kejahatan di dunia komputer, bencana alam, listrik yang tidak stabil, pencurian data, pencurian secara fisik dan lainnya. Kontrol juga mencakup pembuatan keputusan terkait dengan pencegahan kemungkinan gangguan atau ancaman tersebut.

5. Teknologi dan Jaringan Komputer (*Technology and Computer Network*)

Komponen terakhir di dalam sistem informasi ini, yaitu teknologi dan jaringan komputer, memegang peranan terpenting untuk sebuah sistem informasi. Komponen teknologi mengatur *software*, *hardware*, *database*, kontrol dan prosedur, *input* dan *output* sehingga sistem dapat berjalan dan terkendali dengan baik (Pratama, 2014, pp. 11–14).

Waterfall

Metode pengembangan sistem yang diterapkan yaitu *Waterfall* (analisis, desain, *coding*, *testing*, pemeliharaan). Metode *waterfall* bisa juga disebut siklus hidup perangkat lunak. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Secara operasional langkah-langkah metode *waterfall* dapat digambarkan dalam gambar berikut (Sukanto & Shalahuddin, 2015).



Gambar 1
Langkah-langkah metode *waterfall*

Analisis

Tahap analisis meliputi kegiatan sebagai berikut :

- a) Identifikasi Masalah
- b) Analisis PIECES
- c) Analisis Kebutuhan Sistem
- d) Analisis Biaya dan Manfaat
- e) Analisis Kelayakan Sistem

Desain

Proses desain (*design*) menterjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak (*software*) dengan kualitas yang diharapkan.

Kode

Tahap pembuatan aplikasi terdiri dari kegiatan-kegiatan penting di antaranya adalah sebagai berikut :

- a. Pembuatan *Database*
- b. Pembuatan Aplikasi

Testing

Proses pengujian berfokus pada logika *internal software* untuk memastikan bahwa semua pernyataan sudah diuji, dan juga fungsi eksternal. Metode yang digunakan untuk melakukan testing, yaitu *White Box Testing* dan *Black box*.

Perawatan

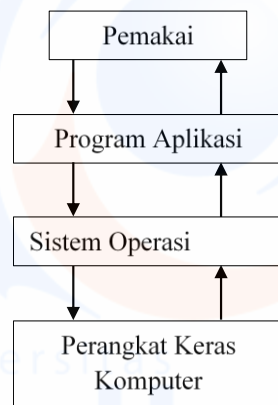
Perawatan *software* mengaplikasikan kembali setiap fase program sebelumnya namun tidak dilakukan dengan membuat *software* yang baru. Perawatan program dapat dilakukan dengan cara :

- a. *Backup database*
- b. *Backup Aplikasi* (Sukamto & Shalahuddin, 2015).

Sistem Operasi

Sistem operasi adalah sekumpulan rutin perangkat lunak yang berada di antara program aplikasi dan perangkat keras. Semua perangkat lunak berjalan di bawah kendali sistem operasi, mengakses perangkat keras lewat sistem operasi dan mengikuti aturan-aturan yang dijalankan oleh sistem operasi.

Karena sistem operasi bertindak sebagai antarmuka dengan perangkat keras maka pemrogram aplikasi tidak berkomunikasi secara langsung dengan perangkat keras sehingga menyederhanakan pemrograman. Pengaksesan ke perangkat keras merupakan hal yang biasa dilakukan aplikasi, menyatukan fungsi pengaksesan perangkat keras ke sistem operasi membuat kode itu tidak perlu diduplikasi di masing-masing aplikasi. Karena semua aplikasi mengakses perangkat keras lewat sistem operasi, posisi netral ini membuatnya ideal sebagai repositori rutin-rutin sistem dipakai bersama yang berguna untuk semua aplikasi (Hariyanto, 2014, pp. 25–26).



Gambar 2

Sistem operasi bertindak sebagai antarmuka antara program aplikasi dan perangkat keras

Tujuan Mempelajari Sistem Operasi

Tujuan yang dapat dicapai dalam mempelajari sistem operasi, yaitu :

1. Tujuan tertinggi adalah agar dapat merancang sendiri atau memodifikasi sistem operasi yang telah ada sesuai kebutuhan khusus kita.
2. Agar dapat menilai sistem operasi dan memilih alternatif sistem operasi sesuai tujuan sistem berbasis komputer yang hendak dibangun. Keputusan pemilihan sistem operasi merupakan keputusan utama dan mendasar dalam pembangunan sistem berbasis komputer.
3. Pemakai harus berinteraksi dengan sistem operasi untuk menyelesaikan tugasnya karena sistem operasi merupakan antarmuka utama dalam berhubungan dengan sistem komputer.
4. Banyak konsep dan teknik pada sistem operasi yang dapat diterapkan pada pengembangan aplikasi-aplikasi lain (Hariyanto, 2014, p. 26).

Basis Data

Basis data terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi

fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli dan pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Sebagai satu kesatuan istilah, basis (*database*) sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

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

Basis data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data atau arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronik seperti cakram magnetis (*magnetic disk* atau disingkat sebagai *disk* saja). Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola melalui perantara mesin pintar elektronik (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metode yang dapat digunakan dalam upaya penyimpanan.

Satu hal yang juga harus diperhatikan, bahwa basis data bukan hanya sekedar penyimpanan data secara elektronik (dengan bantuan komputer). Artinya, tidak semua bentuk penyimpanan data secara elektronik bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam *file* teks (dengan program pengolahan kata), *file spread sheet* dan lain-lain, tetapi tidak bisa disebut sebagai basis data. Hal ini, karena di dalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis kata. Kelak ketika *file-file* tersebut sudah cukup banyak, maka situasi ini maka akan menyulitkan pencarian data tertentu. Yang sangat ditonjolkan dalam basis data adalah pengaturan, pemilahan, pengelompokan dan pengorganisasian data yang akan kita simpan sesuai fungsi atau jenisnya. Pemilahan, pengelompokan, pengorganisasian ini dapat berbentuk sejumlah tabel terpisah atau dalam bentuk pendefinisian kolom-kolom (*field*) data dalam setiap tabel (Fathansyah, 2012, pp. 2–4).

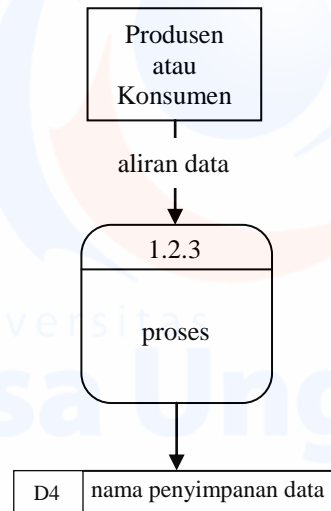
Tujuan Basis Data

Secara lebih lengkap pemanfaatan basis data dilakukan untuk memenuhi tujuan berikut ini :

- Kecepatan dan kemudahan (*speed*)
- Efisiensi ruang penyimpanan (*space*)
- Keakuratan (*accuracy*)
- Ketersediaan (*availability*)
- Kelengkapan (*completeness*)
- Keamanan (*security*)
- Pemakaian bersama (*sharability*) (Hidayatullah, 2015, p. 140).

DFD (Data Flow Diagram)

Data Flow Diagram (DFD) awalnya dikembangkan oleh Chris Gane dan Trish Sarson pada tahun 1979 yang termasuk dalam *Structured Systems Analysis and Design Methodology* (SSADM) yang ditulis oleh Chris Gane dan Trish Sarson. Sistem yang dikembangkan ini berbasis pada dekomposisi fungsional dari sebuah sistem. Berikut adalah contoh DFD yang dikembangkan oleh Chris Gane dan Trish Sarson :



Gambar 3
DFD yang dikembangkan Chris Gane & Trish Sarson

Edward Yourdon dan Tom DeMarco memperkenalkan metode yang lain pada tahun 1980-an di mana mengubah persegi dengan sudut lengkung (pada DFD Chris Gane dan Trish Sarson) dengan lingkaran untuk menotasikan. DFD Edward Yourdon dan Tom DeMarco populer digunakan sebagai model analisis sistem perangkat lunak untuk sistem perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur. Informasi yang ada di dalam perangkat lunak dimodifikasi dengan beberapa transformasi yang dibutuhkan. *Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi *Diagram Alir Data* (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level abstraksi. DFD dapat dibagi menjadi beberapa level yang lebih detail untuk merepresentasikan aliran informasi atau fungsi yang lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur karena pemrograman terstruktur membagi-bagi bagiannya dengan fungsi-fungsi dan prosedur-prosedur.

DFD tidak sesuai untuk memodelkan sistem perangkat lunak yang akan dibangun menggunakan pemrograman berorientasi objek. Paradigma pemrograman terstruktur dan pemrograman berorientasi objek merupakan hal yang berbeda. Jangan mencampuradukkan pemrograman terstruktur dan pemrograman berorientasi objek. Notasi-notasi pada DFD (Edward Yourdon dan Tom DeMarco).

Berikut ini adalah tahapan-tahapan perancangan dengan menggunakan DFD :

- 1) Membuat DFD level 0 atau sering disebut juga *Context Diagram*
DFD level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.
- 2) Membuat DFD level 1
DFD level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD level 1 merupakan hasil *breakdown* DFD level 0 yang sebelumnya sudah dibuat.

3) Membuat DFD level 2

Modul-modul pada DFD level 1 dapat di-breakdown menjadi DFD level 2. Modul mana saja yang harus di-breakdown lebih detail tergantung pada tingkat kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu untuk di-breakdown lagi. Untuk sebuah sistem, jumlah DFD level 2 sama dengan jumlah modul pada DFD level 1 yang di-breakdown.

4) Membuat DFD level 3 dan seterusnya

DFD level 3,4,5 dan seterusnya merupakan breakdown dari modul pada DFD level di atasnya. Breakdown pada level 3,4,5 dan seterusnya aturannya sama persis dengan DFD level 1 atau level 2. Pada satu diagram DFD sebaiknya jumlah modul tidak boleh lebih dari 20 buah modul, diagram akan terlihat rumit dan susah untuk dibaca sehingga menyebabkan sistem yang dikembangkan juga menjadi rumit. (Sukamto & Shalahuddin, 2015, pp. 69–73).

Kamus Data

Kamus data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi :

1. Nama → nama dari data
2. Digunakan pada → merupakan proses-proses yang terkait data
3. Deskripsi → merupakan deskripsi data
4. Informasi tambaha → seperti tipe data, nilai data, batas nilai data dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel 1
Simbol-simbol kamus data

No.	Simbol	Keterangan
	=	disusun atau terdiri dari
	+	dan
	[]	baik ... atau ...
	{ } ⁿ	n kali diulang atau bernilai banyak
	()	data opsional
	...	batas komentar

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perancangan basis data berarti hasil perancangan basis data dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya atau keduanya (Sukamto & Shalahuddin, 2015, pp. 73–74).

Hasil dan Pembahasan

ERD (*Entity Relationship Diagram*)


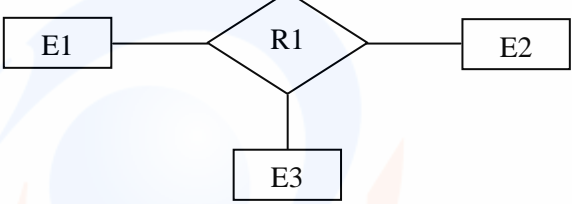
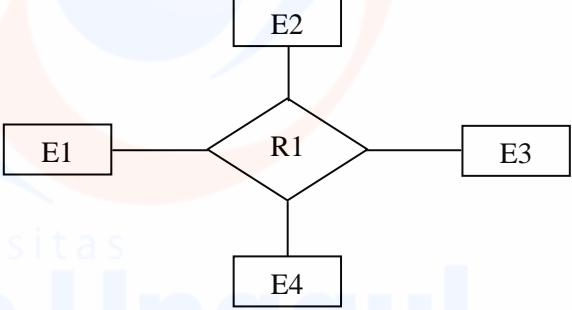
Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer dan Harry Ellis), notasi Crow's Foot dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*.

Berikut adalah contoh bentuk hubungan relasi dalam ERD :

Tabel 2
Bentuk hubungan relasi dalam ERD

No.	Nama	Gambar
1	<i>Binary</i>	
2	<i>Ternary</i>	
3	<i>N-ary</i>	

Normalisasi

Beberapa hal yang perlu diperhatikan dalam proses normalisasi adalah :

1. Suatu teknik formal untuk menganalisis relasi berdasarkan *primary key* dan *functional dependencies* antar-atribut.
2. Dieksekusi dalam beberapa langkah. Setiap langkah mengacu ke bentuk normal tertentu, sesuai dengan sifat yang dimilikinya.
3. Setelah normalisasi diproses, relasi menjadi secara bertahap lebih terbatas atau kuat mengenai bentuk formatnya dan juga mengurangi tindakan *update* yang anomali.

a. *Unnormalized form (UNF)*

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang *unnormalized*, yaitu dengan memindahkan data dari sumber informasi. Contoh : nota penjualan yang disimpan ke dalam format tabel dengan baris dan kolom.

b. *First normal form (1NF)*

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan satu dan hanya satu nilai.

c. Proses UNF ke 1NF

- 1) Tentukan satu atau kumpulan atribut sebagai kunci untuk tabel *unnormalized*.
- 2) Identifikasikan grup yang berulang dalam tabel *unnormalized* yang berulang untuk kunci atribut.
- 3) Hapus grup yang berulang dengan cara :
 1. Masukkan data yang semestinya ke dalam kolom yang kosong pada baris yang berisikan data yang berulang (*flattening the table*).
 2. Menggantikan data yang ada dengan menulis ulang dari kunci atribut yang sesungguhnya ke dalam relasi terpisah.

Contoh :

Tabel Kuliah (*no_siswa, nama, wali_studi, kelas1, kelas2, kelas3*). Atribut kelas1, kelas2, kelas3 merupakan atribut bernilai ganda (*multivalued*).

Contoh pertama atribut bernilai banyak:

Tabel Universal (*nim, nama_mhs, alamat_mhs, tgl_lahir, kode_kul, nama_kul, sks, semester, indeks_nilai, waktu, tempat, nama_dos, alamat_dos*).

Contoh kedua atribut bernilai banyak:

Atribut waktu merupakan atribut bernilai banyak, karena waktu kuliah bisa beberapa hari.

Tabel Mahasiswa (*nim, nama_mhs, alamat_mhs, tgl_lahir, hobi*). Atribut hobi merupakan atribut bernilai banyak, karena hobi mahasiswa banyak.

d. Second normal form (2NF)

- 1) Berdasarkan pada konsep *full functional dependency*, yaitu A dan B merupakan atribut dari sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functionally dependent* pada A tetapi tidak pada *proper subset* dari A.
- 2) 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary-key* bersifat *fully functionally dependent* pada *primary key*.
- 3) 1NF ke 2NF
 1. Identifikasikan *primary key* untuk relasi 1NF.
 2. Identifikasikan *functional dependencies* dalam relasi.
 3. Jika terdapat *partial dependencies* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama dengan salinan determinannya.

e. Third normal form (3NF)

- 1) Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi di mana A, B dan C merupakan atribut dari sebuah relasi, maka $A \rightarrow B$ dan $B \rightarrow C$, maka *transitively dependent* pada A melalui B (jika A tidak *functionally dependent* pada B atau C).
- 2) 3NF adalah sebuah relasi dalam 1NF dan 2NF dan di mana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* pada *primary key*.
- 3) 2NF ke 3NF
 1. Identifikasikan *primary key* dalam relasi 2NF.
 2. Identifikasikan *functional dependencies* dalam relasi.
 3. Jika terdapat *transitive dependencies* terhadap *primary key*, hapus dengan menempatkannya dalam relasi yang baru bersama dengan salinan determinannya

f. Boyce-Codd normal form (BCNF)

- 1) Berdasarkan pada *functional dependencies* yang dimasukkan ke dalam hitungan seluruh *candidate key* dalam suatu relasi, bagaimana pun BCNF juga memiliki batasan-batasan tambahan disamakan dengan definisi umum dari 3NF.

- 2) Suatu relasi dikatakan BCNF, jika dan hanya jika setiap determinan merupakan *candidate key*.
- 3) Perbedaan antara 3NF dan BCNF yaitu untuk *functional dependency* $A \rightarrow B$, 3NF memungkinkan *dependency* ini dalam suatu relasi jika B adalah atribut *primary key* dan A bukan merupakan *candidate key*.
- 4) Sedangkan BCNF menetapkan dengan jelas bahwa untuk *dependency* ini agar ditetapkan dalam relasi A, maka A harus merupakan *candidate key*.
- 5) Setiap relasi BCNF juga merupakan 3NF, tetapi relasi dalam 3NF belum tentu termasuk ke dalam BCNF.
- 6) Dalam BCNF kesalahan jarang sekali terjadi. Kesalahan dapat terjadi pada relasi yang :
 1. Terdiri atas 2 atau lebih *composite candidate key*.
 2. *Candidate key overlap*, sedikit satu atribut (Indrajani, 2014, pp. 217–220).

Contoh:

Table Mahasiswa

$nim \rightarrow nama_mhs \text{ alamat_mhs tgl_lahir}$ (sudah BCNF)

Table Kuliah

$kode_kul \rightarrow nama_kul \text{ sks semester waktu tempat nama_dos}$ (sudah BCNF)

Tabel Dosen

$nama_dos \rightarrow alamat_dosen$ (sudah BCNF)

Tabel Nilai

$kode_kul \text{ nim} \rightarrow indeks_nilai$ (sudah BCNF)

Jika karena kebutuhan tertentu untuk tabel Mahasiswa *alamat_mhs* menjadi ***alamat_jalan nama_kota kode_pos***, karena memang atribut tersebut adalah atribut komposit. Jika tabel Mahasiswa didekomposisi maka menjadi seperti tabel di bawah ini:

Tabel Mahasiswa

$nim \rightarrow nama_mhs \text{ alamat_jalan nama_kota tgl_lahir}$ (sudah BCNF)

Tabel Alamat

$alamat_jalan \text{ nama_kota} \rightarrow kode_pos$ (bukan BCNF)

Bukan BCNF karena pada tabel Alamat terdapat 2 buah KF, yaitu: $alamat_jalan \text{ nama_kota} \rightarrow kode_pos$ dan $kode_pos \rightarrow nama_kota$

Gabungan atribut (*alamat_jalan, nama_kota*) memang merupakan *superkey* pada tabel tersebut, tapi (*kode_pos*) jelas bukanlah *superkey* pada tabel tersebut. Nilai *kode_pos* ada yang sama (tidak unik) (Supardi, 2015, pp. 21–22).

Visual Basic .NET

Pemrograman Visual Basic.NET adalah bahasa pemrograman terpopuler. Ini merupakan pemrograman yang berjalan di atas *platform* NET Framework. Karena itu setiap kali pemrograman VB.NET ini merilis versi barunya, tentu saja akan diikuti atau berbarengan dengan perkembangan NET Framework terbaru.

Di tahun 2013, Microsoft merilis Visual Basic 2013 dengan Net Framework 4.5 yang sangat kompatibel untuk Windows 8 ataupun versi Windows-Windows sebelumnya. Walaupun di tahun

2015 rencananya Microsoft merilis versi terbaru, tapi hingga saat buku ini ditulis, versi terbaru masih versi 2013. Framework adalah software yang berisi *library* yang amat banyak serta menyediakan interoperabilitas bahasa pemrograman.

Program yang ditulis untuk NET Framework dieksekusi dalam lingkungan perangkat lunak yang disebut sebagai *Common Language Runtime (CLR)*. CLR adalah sebuah mesin virtual yang menyediakan layanan penting seperti keamanan, manajemen memori, *exception handling* dan *class library*. Ini menjadikan NET Framework merupakan sebuah *framework* yang lengkap dan handal. Berikut ini adalah contoh *source code* Visual Basic .NET : (Winarno & Zaki, 2015, pp. 17–18).

Sejarah Visual Basic .NET

Bahasa pemrograman Visual Basic telah mengalami banyak perubahan sejak satu dekade belakangan. Walaupun demikian, perkembangannya sudah mulai tercatat sejak tahun 1964. Visual Basic termasuk bahasa pemrograman yang populer. Bahkan sebuah laporan menunjukkan bahwa Visual Basic termasuk satu dari lima bahasa pemrograman terpopuler di dunia. Visual Basic sekarang ini merupakan bahasa pemrograman berorientasi objek yang multiguna. Awalnya, BASIC cuma sekedar bahasa tingkat tinggi (*high level language*) yang memungkinkan semua orang membuat program komputer.

Di pertengahan tahun 1960-an, komputer adalah barang yang sangat mahal dan hanya digunakan untuk tugas-tugas tujuan khusus. Pada saat itu, jenis pemrosesan yang dapat dilakukan adalah *batch processing*, yang artinya komputer hanya dapat melakukan pemrosesan satu pekerjaan pada satu waktu dan pekerjaan lainnya akan dilakukan setelah pekerjaan lainnya selesai. Akan tetapi, di akhir tahun 1960an, muncul komputer yang lebih cepat dan lebih terjangkau. Dengan kemampuan prosesor yang lebih kuat, komputer bisa *idle* atau menganggur, tanpa ada pekerjaan yang ia lakukan sama sekali. Saat inilah kebutuhan pemrosesan menjadi berlebihan.

Bahasa pemrograman di era *batch programming* didesain untuk tujuan-tujuan khusus, begitu juga dengan PC di mana mereka berjalan, misalnya untuk kalkulasi formula ilmiah, pemrosesan data bisnis atau program yang sekedar untuk penyuntingan teks. Bahasa BASIC ini awalnya didesain oleh John G Kemeny dan Thomas E Kurtz. Tujuannya adalah untuk menyediakan akses pemrograman bagi programmer yang bukan berasal dari kalangan teknologi. Awalnya, BASIC diimplementasikan untuk sekelompok siswa di Dartmouth College. BASIC di bawah arahan mereka berdua didesain agar para siswa dapat menulis program untuk Dartmouth *Time-Sharing System*.

Bahasa BASIC didesain untuk mengatasi masalah kerumitan yang selama ini melekat pada bahasa-bahasa pemrograman yang sudah ada. Karena itulah BASIC hadir dengan desain baru yang memang dikhususkan untuk para pemula, yakni para pengguna yang kurang begitu memahami masalah keteknisan dan juga tidak memiliki latar belakang matematika, serta kurang berminat untuk menekuni bidang matematika.

Penggunaan komputer untuk mendukung proses pengajaran dan riset ternyata menarik perhatian banyak kalangan. Pada beberapa tahun kemudian, seiring dengan munculnya beberapa dialek bahasa BASIC lainnya, dialek BASIC buatan Kemeny dan Kurtz dinamakan dengan Dartmouth BASIC. BASIC ini singkatannya adalah dari *Beginner's All Purpose Symbolic Instruction Code*. Berikut ini beberapa sifat bahasa BASIC :

1. Mudah digunakan bagi para pemula.
2. Dapat digunakan sebagai sebuah bahasa pemrograman untuk tujuan umum (*general purpose*).
3. Bisa ditambahi fitur-fitur tambahan dan tingkat lanjut untuk para ahli, tetapi tetap mempertahankan kesederhanaan bahasa untuk para pemula.
4. Sifatnya interaktif.
5. Pesan-pesan kesalahan harus jelas dan mudah dipahami.
6. Waktu respon cepat untuk program-program yang kecil.
7. Tidak membutuhkan pengetahuan dan pemahaman perangkat keras komputer.

8. Pengguna juga tidak harus tahu mengenai sistem operasi (Winarno & Zaki, 2015, pp. 23–25).

SQL Server

SQL Server merupakan *Relational Database Management System* (RDBMS) yang dikembangkan oleh Microsoft. Sebagai *software*, fungsi utama SQL Server adalah menampung dan menggunakan data yang terintegrasi dengan aplikasi baik pada komputer yang sama atau dari komputer lain dalam jaringan (IKAPI, 2015, p. 2).

SQL (*Structured Query Language*) 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 (*International Organization for Standardization*) dan disebut sebagai SQL-86. Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini :

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukkan sama. Berikut ini adalah contoh pengaksesan data pada DBMS dengan SQL yang secara umum terdiri dari empat hal sebagai berikut :

1. Memasukkan data (*insert*)

```
INSERT INTO Tabel_mahasiswa
(nim, nama, tanggal_lahir)
VALUES
('13501058', 'Rosa', '1986-01-01');
```

Query di atas digunakan untuk memasukkan data mahasiswa dengan NIM 13501058, nama Rosa dan tanggal lahir 1 Januari 1986 ke tabel “Tabel_mahasiswa”.

2. Mengubah data (*update*)

```
UPDATE Tabel_mahasiswa
SET
tanggal_lahir = '1990-03-04'
WHERE
Nim = '13501058';
```

Query di atas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM = 13501058 menjadi 4 Maret 1990 dalam tabel “Tabel_mahasiswa”.

3. Menghapus data (*delete*)

```
DELETE FROM Tabel_mahasiswa
WHERE
Nim = '13501058'
```

Query di atas digunakan untuk menghapus data mahasiswa dengan NIM = 13501058 dari tabel “Tabel_mahasiswa”.

4. Menampilkan data (*select*)

```
SELECT nim,
nama
FROM Tabel_mahasiswa
WHERE
Nim = '13501058';
```

Query di atas digunakan untuk menampilkan data mahasiswa yang tersimpan dalam “Tabel_mahasiswa” dengan NIM = 13501058 (Sukamto & Shalahuddin, 2015, pp. 46–48).

Perangkat Keras

Sistem komputer terdiri dari perangkat keras dan perangkat lunak. Tanpa perangkat lunak, perangkat keras hanya berfungsi sebagai benda metal saja yang tidak dapat mengerjakan sesuatu. Tanpa perangkat keras, perangkat lunak hanya merupakan kode-kode komputer saja yang tidak dapat menggerakkan perangkat kerasnya. Oleh karena itu, perangkat keras dan perangkat lunak harus bekerja bersama-sama membentuk suatu sistem, yaitu sistem komputer.

Perangkat keras sebagai subsistem dari sistem komputer juga mempunyai komponen, yaitu komponen peranti masukan (*input device*), komponen peranti pemroses (*processing device*), komponen peranti keluaran (*output device*) dan komponen peranti penyimpanan (*storage*) eksternal (Sutabri, 2014, p. 32).

Peranti Masukan

Peranti masukan adalah alat yang digunakan untuk menerima masukan yang dapat berupa masukan data ataupun masukan program. Beberapa peranti masukan mempunyai fungsi ganda, yaitu sebagai peranti masukan dan sekaligus sebagai peranti keluaran untuk menampilkan hasil. Peranti masukan atau keluaran ini disebut dengan terminal. Peranti masukan dapat digolongkan ke dalam beberapa golongan, yaitu *keyboard*, peranti penunjuk (*pointing device*), *scanner*, sensor dan pengenalan suara (*voice recognizer*) (Sutabri, 2014, p. 32).

Peranti Pemroses

Peranti pemroses adalah alat dengan instruksi-instruksi program yang dieksekusi untuk memproses data yang dimasukkan lewat peranti masukan yang hasilnya nanti akan ditampilkan di peranti keluaran. Peranti pemroses terdiri dari *central processor* atau CPU (*Central Processing Unit*) dan memori utama (Sutabri, 2014, p. 39).

Peranti Keluaran

Keluaran yang dihasilkan dari pengolahan data dapat digolongkan ke dalam 3 macam bentuk, yaitu tulisan (huruf, kata, angka, karakter khusus dan simbol-simbol lain), grafik atau gambar dan suara (bentuk musik atau omongan). Untuk mendapatkan bentuk-bentuk keluaran tersebut, dibutuhkan alat untuk menampilkannya, yaitu peranti keluaran atau yang biasa disebut dengan *output device* atau output unit. Peranti keluaran dapat berbentuk *hardcopy device* dan *softcopy device* (Sutabri, 2014, p. 46).

Peranti Penyimpanan Eksternal

Memori utama dalam peranti pemroses merupakan simpanan yang kapasitasnya tidak begitu besar dan umumnya bersifat *volatile*, yaitu informasi yang dikandungnya akan hilang bila aliran listrik terputus. Terkadang diperlukan suatu penyimpanan yang mempunyai kapasitas besar dan bersifat *nonvolatile* untuk menyimpan data dan program dalam kurun waktu tertentu.

Penyimpanan yang mempunyai ciri-ciri tersebut adalah *eksternal memory* (memori eksternal, karena terletak di luar alat prosesnya), atau disebut *mass storage* (penyimpanan masal, karena kapasitasnya umumnya lebih besar dari memori utama) atau disebut *secondary storage* (penyimpanan sekunder, karena penyimpanan pertama adalah memori utama) atau disebut *auxiliary storage* (penyimpanan tambahan, karena penyimpanan utama adalah memori utama) atau disebut *permanent storage* (penyimpanan tetap, karena *nonvolatile*) atau disebut juga *backing storage* (penyimpanan pendukung) atau disebut juga bank data computer (Sutabri, 2014, p. 52).

Jaringan Komputer

Terdapat banyak definisi oleh para ahli mengenai jaringan komputer. Menurut Forouzan di dalam bukunya yang berjudul *Computer Network A Top Down Approach*, disebutkan bahwa jaringan komputer adalah hubungan dari sejumlah perangkat yang dapat saling berkomunikasi satu

sama lain (*a network is a interconnection of a set of device capable of communication*). Perangkat yang dimaksud pada definisi ini mencakup semua jenis perangkat komputer (komputer dekstop, komputer jinjing, *smartphone* dan PC tablet) dan perangkat penghubung (*router, switch, modem* dan *hub*).

Jadi, dapat dibayangkan bahwa jika menyebutkan jaringan komputer (*computer network*), akan terdapat minimal dua buah komputer atau perangkat yang saling terhubung satu sama lain. Di dalam sebuah jaringan komputer yang lebih luas, akan terdapat beragam perangkat komputer dan perangkat terhubung lainnya yang saling terhubung. Terjadi proses komunikasi dan transfer paket data di dalamnya (Pratama, 2015, p. 12).

Syarat Sebuah Jaringan Komputer

Berdasarkan definisi mengenai jaringan komputer, maka untuk dapat disebut sebagai sebuah jaringan komputer, terdapat empat buah syarat yang harus dipenuhi. Keempat syarat tersebut yaitu:

1. Minimal terdapat dua buah perangkat atau komputer yang terhubung. Hubungan ini dapat menggunakan sarana kabel (*wired*) maupun nirkabel (*wireless*).
2. Terdapat pengguna di dalamnya yang berinteraksi dengan pengguna lainnya maupun terhadap layanan dan penyedia layanan.
3. Terdapat data yang dipertukarkan di dalamnya. Selain data juga terdapat konten (teks dan multimedia) maupun informasi (hasil pengolahan data).
4. Terdapat pemakaian secara bersama-sama (*sharing*) terhadap perangkat keras (*hardware*) dan perangkat lunak (*software*) (Pratama, 2015, p. 13).

LAN (Local Area Network)

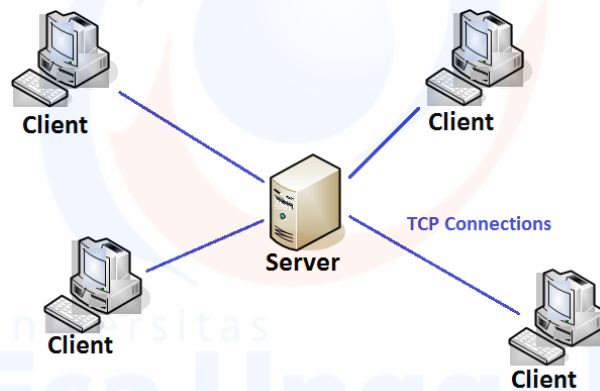
LAN (*Local Area Network*) merupakan jaringan komputer terkecil untuk pemakaian pribadi. LAN (*Local Area Network*) memiliki skala jangkauan mencakup 1 KM hingga 10 KM, dalam bentuk koneksi *wired* (kabel), *wireless* (nirkabel), maupun kombinasi keduanya. Umumnya LAN (*Local Area Network*) lebih banyak diimplementasikan di dalam sebuah ruangan maupun sebuah gedung. Sebagai contoh:

1. Jaringan lokal di perpustakaan untuk penyediaan *repository ebook* dan layanan sistem informasi perpustakaan.
2. Jaringan lokal pada laboratorium untuk sarana riset, penelitian, maupun bertukar data dan informasi.
3. Jaringan lokal pada kantor swasta, instansi pemerintahan, kampus atau perguruan tinggi dan sekolah untuk penggunaan bersama printer, sistem informasi, pertukaran data dan lain-lain.

Jaringan LAN umum juga disebut sebagai intranet. LAN beda dengan internet. Sesuai namanya, jaringan ini bersifat privat, yaitu hanya diperuntukkan bagi pengguna di dalam internal organisasi atau perusahaan atau instansi atau ruangan bersangkutan saja (Pratama, 2015, pp. 32–33).

Paradigma Client Server

Paradigma *client server* merupakan paradigma yang telah lama digunakan secara umum dan luas pada jaringan pada jaringan komputer. Pada paradigma *client server* terdapat minimal sebuah komputer yang bertindak sebagai *server* dan sebuah *client*. *Server* bertugas untuk melayani permintaan *client*. Sedangkan *client* bertugas untuk mengirim permintaan kepada *server*. Semua *client* berpusat kepada *server*. Pada paradigma *client server* ini, komunikasi di antara *client* pada *Application Layer* dilakukan di antara dua buah program yang sedang berjalan. Aplikasi atau program yang sedang berjalan di sisi *server* maupun *client* ini disebut dengan proses yang sedang berjalan (*process*). Ini berarti bahwa baik di sisi *server* maupun di sisi *client*, masing-masing terdapat proses yang sedang berjalan.



Gambar 4
Client server

Client didefinisikan sebagai komputer yang menjalankan proses untuk berkomunikasi dengan cara mengirimkan permintaan. *Server* didefinisikan sebagai komputer yang menjalankan proses untuk berkomunikasi dengan cara menunggu permintaan dari *client* dan membalas atau melayani permintaan *client*.

Terminal *server* adalah sebuah perangkat keras atau *server* yang menyediakan terminal (PC, *printer*, dan perangkat lain) dengan titik koneksi umum untuk jaringan area lokal atau lebar. Terminal terhubung ke *server* terminal dari *port* serial RS-232C atau RS-423 mereka. Sisi lain dari *server* terminal terhubung melalui kartu antarmuka jaringan (NIC) ke jaringan area lokal (LAN) (biasanya *Ethernet* atau *token ring* LAN) melalui modem ke jaringan area luas *dial-in/out*, atau ke X. 25 jaringan atau 3270 gerbang. Merek yang berbeda dari terminal *server* menawarkan berbagai jenis interkoneksi. Beberapa dapat dipesan dalam konfigurasi yang berbeda berdasarkan kebutuhan pelanggan. Penggunaan *server* terminal berarti bahwa setiap terminal tidak membutuhkan antarmuka kartu jaringan atau modem sendiri. Sumber daya koneksi di dalam terminal *server* biasanya dibagi secara dinamis oleh semua terminal terpasang.

Beberapa *server* terminal dapat dibagi hingga 128 terminal. Terminal dapat PC, terminal yang meniru 3270s, *printer*, atau perangkat lain dengan antarmuka RS-232/423. Terminal dapat menggunakan TCP / IP untuk koneksi Telnet ke *host*, LAT ke *host* Digital Equipment Corporation, atau TN3270 untuk koneksi Telnet ke *host* IBM dengan 3270 aplikasi. Dengan beberapa *server* terminal, pengguna terminal yang diberikan dapat memiliki beberapa koneksi *host* untuk berbagai jenis sistem operasi *host* (UNIX, IBM, DEC). Terminal *server* bertindak seperti sebuah *multiplayer* yang memungkinkan sejumlah komputer kecil, atau terminal-terminal yang lain, untuk mengakses ke sebuah titik LAN yang sama. Terminal *server* dapat digunakan untuk menyediakan akses ke komputer pusat untuk sejumlah terminal dengan menggunakan biaya yang rendah. Sebuah terminal *server* juga bisa disebut sebagai *server* serial atau konsol *server* akses *server* jaringan. Terminal *server* memungkinkan organisasi untuk menghubungkan perangkat dengan antarmuka serial RS-232, RS-422 atau RS-485 ke jaringan area lokal (LAN) (Pratama, 2015, pp. 146–147).

Kesimpulan

Setelah melakukan analisis dan desain sistem pengelolaan data penjualan pada PT. ABC, penulis telah merancang sistem informasi yang perlu dikembangkan. Kesimpulan yang dapat dikemukakan berdasarkan penjelasan di atas adalah sebagai berikut : Kondisi sistem informasi pengelolaan data penjualan produk kesehatan yang berjalan pada PT. ABC masih belum efektif (lambat dan belum akurat dalam menyajikan informasi) dan belum efisien (banyak menggunakan kertas) sehingga sering kali terjadi kesalahan-kesalahan pencatatan kode dan harga terutama bila terjadi kenaikan harga. Perancangan sistem informasi pengelolaan data penjualan berbasis *desktop* dapat membantu kinerja para karyawan dari segi kecepatan dan keakuratan informasi yang

dihasilkan serta menghemat penggunaan kertas-kertas dari segi biaya operasional. Sistem informasi pengelolaan data penjualan dapat memberikan kemudahan dalam melakukan pencarian data penjualan, karena telah disimpan ke dalam aplikasi.

Daftar Pustaka

- Fathansyah. (2012). *Basis Data*. Bandung: Informatika Bandung.
- Hariyanto, B. (2014). *Sistem Operasi*. Bandung: Informatika Bandung.
- Hidayatullah, P. (2015). *Visual Basic.Net*. Bandung: Informatika Bandung.
- IKAPI. (2015). *Mengenal Pemrograman Database*. Jakarta: Gramedia.
- Indrajani. (2014). *Pengantar Sistem Basis Data*. Jakarta: PT Elex Media Komputindo.
- Pratama, I. P. A. E. (2014). *Sistem Informasi dan Implementasinya*. Bandung: Informatika Bandung.
- Pratama, I. P. A. E. (2015). *Handbook Jaringan Komputer*. Bandung: Informatika Bandung.
- Sujarweni, V. W. (2015). *Sistem Akuntansi*. Yogyakarta: Pustaka Baru Press.
- Sukamto, R. A., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika Bandung.
- Supardi, Y. (2015). *Mudah dan Cepat Membuat Skripsi dengan VB 2012*. Jakarta: PT Elex Media Komputindo.
- Sutabri, T. (2014). *Pengantar Teknologi Informasi*. Yogyakarta: Andi Offset.
- Winarno, E., & Zaki, A. (2015). *VB. NET untuk Skripsi*. Jakarta: PT Elex Media Komputindo.