

BAB II

TINJAUAN PUSTAKA

Tinjauan pustaka memiliki arti yaitu peninjauan kembali pustaka-pustaka yang terkait (*review of related literature*). Sesuai dengan arti tersebut, suatu tinjauan pustaka berfungsi sebagai peninjauan kembali (*review*) pustaka berkaitan dengan masalah yang diteliti. Tujuan dari tinjauan pustaka adalah untuk memberi pondasi dalam penelitian ini.

2.1 Penelitian Terdahulu

Penelitian terdahulu yang berhubungan dengan pembuatan aplikasi yang berkaitan dengan bisnis dan SMS *Gateway* adalah sebagai berikut:

1. Prakoso dan Christianti (2008) melakukan penelitian tentang analisis dan perancangan sistem informasi akademik. Studi kasus pada penelitian ini adalah Sekolah Menengah Atas Terpadu (SMAT) Krida Nusantara. Keadaan sekolah asrama ini membuat orang tua sukar untuk mengontrol nilai anaknya. Orang tua hanya dapat melihat nilai pada saat pembagian rapor saja. Berdasarkan hal tersebut, peneliti membuat sistem informasi yang dapat digunakan untuk memantau nilai siswa yang dapat diakses di mana saja dan kapan saja. Dengan aplikasi ini, orang tua siswa dapat memperoleh informasi perkembangan akademik anaknya dengan cepat.
2. Hilmansyah (2009) melakukan penelitian tentang perancangan prototipe *e-business* untuk implementasi *Vendor Managed Inventory* (VMI). Penelitian ini mengangkat sebuah permasalahan yaitu pada sistem pasang baru pelanggan di PT. PLN (Persero) APJ Kediri. Permasalahan tersebut berupa keterbatasan dalam melakukan koordinasi untuk pengadaan material dalam rangka memenuhi permintaan pelanggan. Peneliti memberikan solusi yaitu rancangan prototipe berupa *website* guna mempermudah akses data aliran material sehingga dapat *ter-update* lebih cepat dan akurat. Dengan aplikasi ini, akses data aliran material dan koordinasi dalam pengadaan material menjadi lebih mudah.
3. Saragih (2012) melakukan penelitian tentang Perancangan SMS *Gateway* untuk meningkatkan kualitas pelayanan di PT. PLN (Persero) Medan. Pada

penelitian ini, peneliti menyatakan bahwa SMS Gateway dengan tool Gammu dapat meningkatkan efektivitas pengolahan data SMS. Peneliti memilih teknologi komunikasi dalam bentuk SMS ini karena lebih praktis, mudah, dan efisien untuk menyampaikan informasi.

Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan

Karakteristik Penelitian	Peneliti			
	Prakoso dan Christianiti (2008)	Hilmansyah (2009)	Saragih (2012)	Penelitian ini
Topik	Analisis dan perancangan sistem informasi akademik	Perancangan prototipe <i>e-business</i> untuk implementasi VMI	Perancangan SMS Gateway untuk meningkatkan kualitas pelayanan	Rekayasa aplikasi <i>m-business</i> untuk membantu pemeriksaan jumlah barang
Objek Penelitian	SMAT Krida Nusantara	PLN (Persero) APJ Kediri	PT. PLN (Persero) Medan	Swalayan KPRI UB
Aplikasi yang dibuat	1. Web 2. WAP	Web	SMS Gateway	1. Web 2. Java Mobile 3. SMS Gateway

2.2 Supplier

Supplier adalah perusahaan atau pihak yang mengadakan atau menyediakan barang untuk dijual kepada *buyer* (Yushita, 2011). Peran *supplier* sangat penting bagi perkembangan industri manufaktur ataupun jasa. *Supplier* memberikan kontribusi yang sangat besar bagi keberhasilan penyaluran barang dari perusahaan kepada *buyer* (pelanggan). *Supplier* merupakan orang-orang yang berada pada posisi paling depan dari perusahaan. Pada sisi lain, *supplier* merupakan orang yang berada di luar perusahaan, memastikan barang yang dikirim berada pada kondisi yang seharusnya (Hardianti, 2011).

Buyer bertanggung jawab atas pemberian informasi yang jelas dan memadai sehingga *supplier* dapat mengetahui dengan tepat apa yang harus dikerjakannya. Di sisi lain, *supplier* bertanggung jawab atas jaminan mutu yang akan memberikan kepuasan kepada *buyer* dan juga bertanggung jawab atas penyerahan barang yang dipesan berdasarkan permintaan pelanggan. Baik *buyer* maupun *supplier* harus saling bertukar informasi yang diperlukan dan melakukan pengendalian aktivitas bisnis, seperti pembelian dan persediaan, agar tata hubungan *buyer-supplier* terjaga dengan baik dan memuaskan (Yuliandri, 2009 dalam Hardianti, 2011).

2.3 *Mobile Business (M-Business)*

M-Business adalah aplikasi infrastruktur yang dibutuhkan untuk mengelola segala sesuatu yang berhubungan dengan bisnis dan penjualan informasi, jasa, dan komoditi dengan menggunakan perangkat *mobile* (Kalakota dan Robinson, 2002 dalam Kim, 2002). Secara dasar, aplikasi peranti *mobile* terbagi atas (Mardiono, 2006):

1. *Personal Information Management (PIM)*: menyediakan fungsi kalender, buku alamat, jadwal, memo, dan tugas yang harus dilakukan. Kadang-kadang PIM juga dilengkapi dengan kemampuan menerima dan mengirim *email*.
2. Dokumen: menyediakan fungsi *word processing*, *spread sheet*, akses *web*, *ebook*, multimedia, dan presentasi.
3. Aplikasi *m-business*: menyediakan fungsi penangkapan, pemrosesan, dan pengiriman data. Komunikasi yang umum antara *server* dan aplikasi adalah melalui sistem *messaging* seperti SMS. Namun, aplikasi bisnis saat ini sudah bisa terhubung langsung dengan *server* melalui koneksi *web service*.
4. Permainan dan hiburan: menyediakan fungsi permainan, musik, video, dan sebagainya.

2.3.1 Aplikasi *M-Business*

Aplikasi *m-business* dapat dikategorikan secara horizontal dan vertikal. Kategori horizontal berdasarkan jenis aplikasinya, sedangkan aplikasi vertikal berdasarkan jenis bisnisnya. Beberapa kategori horizontal yang paling menguntungkan jika *m-business* dikembangkan antara lain (Mardiono, 2006):

1. *Customer Relationship Management (CRM)*
Pada umumnya CRM memiliki lima modul dasar yaitu: *Sales Force Automaton*, *Field Service*, *Customer Support*, *Call Center*, dan *Marketing Analysis*. Secara operasional, biasanya para pengguna garis depan (*front-end user*) CRM ini bekerja di lapangan (terutama *salesman*). Suatu target yang sangat cocok di mana penggunaan aplikasi *m-business* tentunya akan sangat mempercepat eksekusi proses dalam CRM.

2. *Enterprise Resource Planning (ERP)*

ERP dapat dikatakan sebagai inti dari sistem informasi sebuah organisasi. Modul-modul ERP antara lain: akuntansi, pemasaran, SDM (Sumber Daya Manusia), produksi, pembelian, dan sebagainya. Mengingat ERP adalah suatu *backbone system*, tentunya akan sangat menguntungkan sekali jika koneksinya ditingkatkan ke level *mobile*. Pekerja bermobilitas tinggi dapat menyelesaikan laporan dan proses lainnya di mana dan kapan saja melalui peranti *mobile*.

3. *Transportation Management System (TMS)*

Fungsi utama dari TMS adalah untuk mengatur utilitas *loading* (pemuatan), *routing* (perjalanan), dan estimasi pengiriman. Penggunaan aplikasi *mBusiness* akan sangat meningkatkan kecepatan proses TMS seperti pengiriman status posisi pengiriman barang lewat peranti *mobile*.

Beberapa kategori vertikal yang paling membutuhkan aplikasi *m-business* antara lain:

1. **Jasa Keuangan:** Proses-proses keuangan membutuhkan informasi secara cepat dan akses di mana dan kapan saja seperti informasi saldo tabungan, transfer uang, pembayaran rekening, dan sebagainya.
2. **Kesehatan:** Bisnis kesehatan menyangkut hubungan antara rumah sakit, dokter, *salesman* obat, apotek, dan pasien. Akses informasi stok obat di apotek rumah sakit akan memudahkan *salesman* dalam melakukan pengiriman obat.
3. **Distribusi:** *Supplier* dan *buyer* membutuhkan informasi stok barang, di mana aplikasi *m-business* akan memberikan nilai lebih berupa kemudahan dan kecepatan akses informasi tersebut. Nilai lebih tersebut akan semakin besar jika aplikasi *m-business* diterapkan di sektor distribusi ritel.

2.4 Pengembangan Perangkat Lunak Secara Cepat

Lingkungan bisnis saat ini berada pada lingkungan yang berubah dengan cepat. Perusahaan harus tanggap terhadap kesempatan baru, perubahan kondisi ekonomi, dan munculnya pesaing. Perangkat lunak menjadi bagian penting dari hampir seluruh kegiatan bisnis. Sudah menjadi suatu kebutuhan bahwa proses

pengembangan perangkat lunak dilakukan dengan cepat untuk dapat meraih keuntungan dari kesempatan baru dan tanggap terhadap tekanan pesaing.

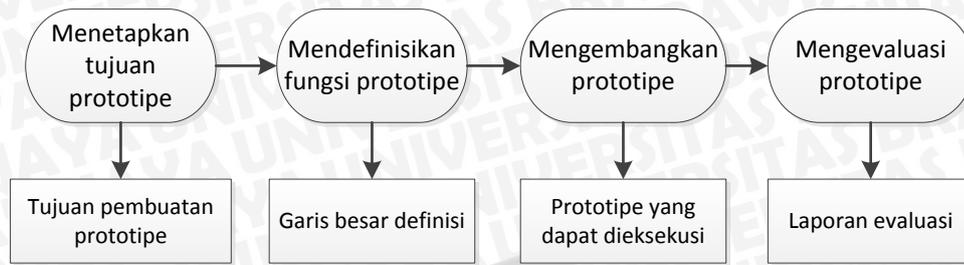
Proses pengembangan perangkat lunak yang berdasar pada pengumpulan kebutuhan secara lengkap lalu perancangan, pembangunan, dan pengujian sistem tidak mengarahkan kepada pengembangan perangkat lunak secara cepat. Perubahan kebutuhan atau munculnya masalah baru di lingkungan bisnis yang berubah dengan cepat, membuat perancangan sistem atau implementasi harus dikerjakan ulang atau diuji ulang. Akibatnya, metode pengembangan konvensional yang mendasarkan pada pengumpulan kebutuhan secara lengkap lalu menuju aktivitas selanjutnya dapat memperpanjang waktu pengembangan dan perangkat lunak sampai kepada pengguna jauh setelah penetapan kebutuhan awal.

Proses pengembangan perangkat lunak secara cepat (*Rapid Software Development*) dirancang untuk menghasilkan perangkat lunak yang bermanfaat secara cepat. Karakteristik dasar dari pengembangan perangkat lunak secara cepat yaitu (Sommerville, 2006):

1. Spesifikasi sistem tidak dibuat secara detail. Dokumen kebutuhan mendefinisikan hanya karakteristik yang paling penting dari sistem.
2. Pengguna akhir dan pihak yang berkepentingan pada sistem terlibat dalam penentuan dan pengevaluasian tiap tahapan pengembangan perangkat lunak.

2.4.1 Software Prototyping

Software prototyping merupakan salah satu pendekatan dalam pengembangan perangkat lunak dengan cepat yang akan menghasilkan sebuah prototipe. Prototipe adalah versi awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep. Pengembangan secara cepat dari prototipe adalah hal penting sehingga pihak yang berkepentingan dari sistem dapat melakukan percobaan dengan prototipe sesegera mungkin pada proses pengembangan perangkat lunak.



Gambar 2.1 Diagram Alir Pengembangan Prototipe
Sumber: Sommerville (2006)

Diagram alir untuk pengembangan prototipe ditunjukkan pada gambar 2.1. Tujuan dari pembuatan prototipe harus dibuat secara jelas dari awal proses pengembangan, seperti: pengembangan sistem untuk memvalidasi kebutuhan fungsional atau pengembangan sistem untuk menunjukkan kelayakan aplikasi kepada manajemen.

Tahap selanjutnya dalam proses pengembangan prototipe adalah pendefinisian fungsi prototipe. Pada tahap ini, pengembang memutuskan fungsi-fungsi apa saja yang terdapat di dalam prototipe dengan melakukan pengumpulan kebutuhan perangkat lunak dan membuat daftar kebutuhan sistem (*System Requirements Checklist*, SRC). Untuk mempercepat penyampaian prototipe ke pengguna, pengembang dapat menghilangkan beberapa fungsionalitas dari prototipe seperti waktu tanggap sistem dan penggunaan memori.

Tahap selanjutnya adalah proses pengembangan prototipe yang akan menghasilkan prototipe yang dapat dipakai. Tahap akhir dari proses pengembangan prototipe adalah evaluasi prototipe. Tujuan prototipe digunakan sebagai tolok ukur evaluasi yang dilakukan terhadap prototipe yang sudah dibuat.

2.5 Analisis PIECES

Analisis PIECES (*Performance, Information, Economy, Control, Efficiency, dan Service*) adalah teknik untuk mengidentifikasi permasalahan yang terjadi pada sistem yang sedang berjalan. Dengan dilakukannya analisis PIECES, pengembang akan menemukan beberapa masalah utama dari suatu sistem. Berdasarkan permasalahan utama tersebut, pengembang dapat memberikan solusi yang tepat sesuai dengan masalah yang dihadapi. Penjelasan dari komponen analisis PIECES adalah sebagai berikut (Al Fatta, 2007):

1. Analisis Kinerja (*Performance*)

Analisis ini difokuskan kepada kemampuan sistem untuk menyelesaikan tugas pelayanan dengan cepat sehingga sasaran dan tujuan segera tercapai. Kinerja dapat diukur lewat jumlah hasil (*throughput*) dan waktu tanggap dari sistem. *Throughput* adalah jumlah pekerjaan yang dapat diselesaikan selama jangka waktu tertentu, sedangkan waktu tanggap adalah keterlambatan rata-rata antara suatu transaksi dengan tanggapan yang diberikan kepada transaksi tersebut.

2. Analisis Informasi (*Information*)

Analisis ini difokuskan kepada evaluasi terhadap kemampuan sistem informasi dalam menghasilkan informasi yang cukup dan bermanfaat untuk menangani masalah yang muncul atau dalam pengambilan keputusan. Situasi yang membutuhkan peningkatan informasi meliputi:

- a. Kurangnya informasi mengenai keputusan atau situasi saat ini.
- b. Kurangnya informasi yang relevan mengenai keputusan atau situasi saat ini.
- c. Kurangnya informasi yang tepat waktu.
- d. Terlalu banyak informasi.
- e. Informasi tidak akurat.

Permasalahan yang dihadapi dalam analisis informasi adalah:

- a. Data yang berlebihan, yaitu data yang sama ditangkap dan/atau disimpan di banyak tempat.
- b. Kekakuan data, yaitu data ditangkap dan disimpan, tetapi diorganisasikan sedemikian rupa sehingga laporan dan pengujian tidak dapat atau sulit dilakukan.

3. Analisis Ekonomi (*Economy*)

Analisis ini fokus terhadap penilaian kepada sistem atas biaya dan keuntungan yang akan didapat dari sistem yang akan dikembangkan. Hal yang perlu diperhatikan pada analisis ini adalah:

- a. Biaya, contoh:
 - 1) Biaya tidak diketahui.
 - 2) Sumber biaya tidak dapat dilacak.

- 3) Biaya terlalu tinggi.
- b. Keuntungan, contoh:
 - 1) Pasar-pasar baru dapat dieksplorasi.
 - 2) Peningkatan nilai informasi dan keputusan yang dihasilkan.
 - 3) Pesanan-pesanan dapat ditingkatkan.
4. Analisis Kendali (*Control*)

Analisis ini difokuskan kepada keamanan sistem yang digunakan yang dapat meningkatkan kinerja sistem, mencegah atau mendeteksi kesalahan sistem, menjamin keamanan data dan informasi. Hal yang perlu diperhatikan pada analisis ini adalah:

 - a. Keamanan yang lemah, seperti:
 - 1) *Input* data tidak diedit dengan cukup.
 - 2) Kejahatan terhadap data, seperti penggelapan atau pencurian data.
 - 3) Pelanggaran etika pada data atau informasi, seperti data diakses oleh orang yang tidak berwenang.
 - 4) Terjadi *error* saat pembuatan keputusan.
 - 5) Terjadi *error* saat pemrosesan.
 - b. Keamanan yang berlebihan, seperti:
 - 1) Prosedur birokratis memperlambat sistem.
 - 2) Pengendalian yang berlebihan mengganggu pengguna.
 - 3) Pengendalian pemrosesan menyebabkan penundaan pemrosesan.
5. Analisis Efisiensi (*Efficiency*)

Analisis ini menyangkut bagaimana menghasilkan *output* sebanyak-banyaknya dengan *input* yang sekecil mungkin. Indikasi sistem dapat dikatakan tidak efisien adalah:

 - a. Banyak waktu yang terbuang pada aktivitas SDM, mesin, atau komputer.
 - b. Data dimasukkan atau disalin secara berlebihan.
 - c. Data diproses secara berlebihan.
 - d. Informasi yang dihasilkan berlebihan.
 - e. Usaha dan material yang dibutuhkan untuk suatu tugas berlebihan.
6. Analisis Layanan (*Service*)

Analisis ini menyangkut koordinasi aktivitas dalam pelayanan yang ingin dicapai sehingga tujuan dan sasaran pelayanan dapat dicapai. Beberapa kriteria di mana kualitas layanan sistem dapat dikatakan kurang baik yaitu:

- a. Sistem menghasilkan *output* yang tidak akurat, tidak konsisten, dan tidak dapat dipercaya.
- b. Sistem tidak mudah dipelajari dan tidak mudah digunakan.
- c. Sistem tidak fleksibel.

2.6 Database

Database adalah sebuah kumpulan data yang disimpan secara terintegrasi yang dikelola dan kendalikan secara terpusat (Satzinger, Jackson, dan Burd, 2010). *Database* terdiri dari entitas, atribut, dan *relationship* dari informasi organisasi/perusahaan. Entitas merupakan pondasi dasar untuk sebuah model data. Entitas dapat berupa orang, tempat, kejadian, pegawai, pesanan, atau produk. Atribut merupakan beberapa jenis informasi yang diambil dari sebuah entitas. Atribut dapat berupa nama akhir, alamat rumah, atau alamat *e-mail*. *Relationship* merupakan hubungan antara entitas dan ditunjukkan dengan garis yang menyambungkan entitas-entitas bersama (Dennis, Wixom, dan Roth, 2012).

Meskipun sebenarnya tujuan dari *database* tersebut sama, yaitu lebih mempermudah dalam pengolahan data, namun cara pembuatannya ada berbagai macam. Macam-macam dari *database* tersebut dapat dilihat dari bentuk konfigurasi sistemnya atau dari bentuk/isi dari *database* tersebut. Ada beberapa jenis dari *database*, mulai dari yang menggunakan teks biasa, menggunakan *excel*, *lotus*, *foxpro*, *dbase*, *paradoc*, *access*, *oracle*, *SQL*, dan lain-lain. Masing-masing memiliki perbedaan dari sisi format data, fasilitas yang disediakan dan teknik pengolahan basis data (*database engine*).

2.6.1 DBMS (*Database Management System*)

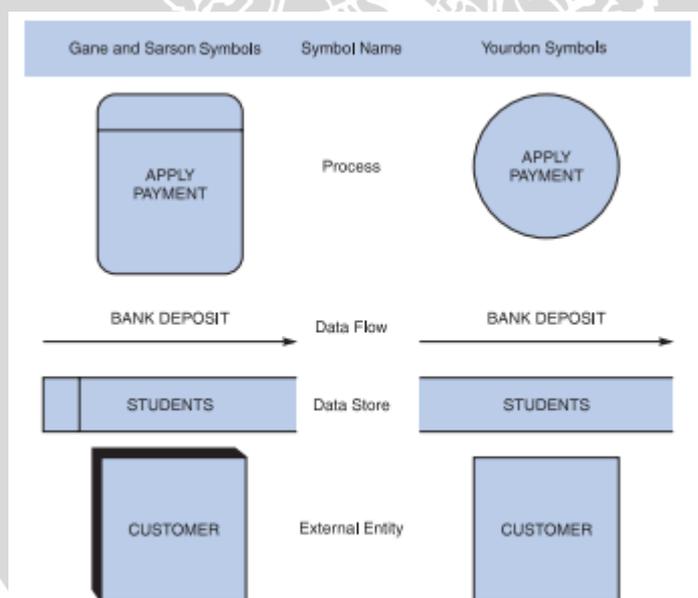
DBMS merupakan sistem perangkat lunak yang mengelola dan mengendalikan akses ke *database* (Satzinger, Jackson, dan Burd, 2010). DBMS mempunyai tujuan untuk memperoleh penciptaan struktur data dan membebaskan *programmer* dari masalah penyimpanan *file* yang kacau. DBMS dapat digunakan

untuk mendesain struktur *file* secara jauh lebih mudah dan menyiapkan suatu basis data yang dapat digunakan oleh sejumlah pemrograman aplikasi yang berlainan.

RDBMS atau *Relationship Database Management System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar tabel. Selain RDBMS, terdapat jenis DBMS lainnya seperti *Hierarchy DBMS* dan *Object Oriented DBMS*.

2.6.2 DFD (*Data Flow Diagram*)

DFD merupakan model analisis terstruktur yang menunjukkan *inputs*, proses-proses, penyimpanan, dan *outputs* dari sebuah sistem (Satzinger, Jackson, dan Burd, 2010). Serangkaian DFD memberikan sebuah model logis yang menunjukkan apa yang sistem lakukan (Shelly dan Rosenblatt, 2012). Simbol DFD yang sering digunakan dapat dilihat pada gambar 2.2.



Gambar 2.2 Simbol DFD

Sumber: Shelly dan Rosenblatt (2012)

DFD dapat digambarkan dalam diagram konteks, diagram 0, dan Diagram tingkat yang lebih rendah. Diagram tingkat yang lebih rendah menggambarkan tingkatan dan proses di setiap lingkaran. Berikut ini adalah penjelasan mengenai diagram konteks, diagram 0 dan diagram tingkat yang lebih rendah:

1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Dalam diagram konteks hanya ada satu proses dan tidak boleh ada *data store* di dalamnya.

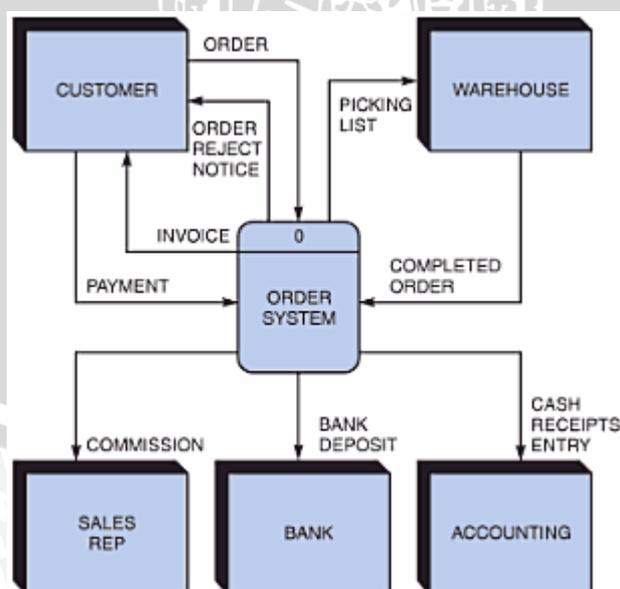
2. Diagram 0

Diagram 0 adalah diagram yang menggambarkan proses dari DFD. Diagram 0 memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, menunjukkan tentang fungsi-fungsi utama atau proses yang ada, aliran data, dan entitas luar. Pada level ini sudah dimungkinkan untuk digambarkan *data store* yang digunakan.

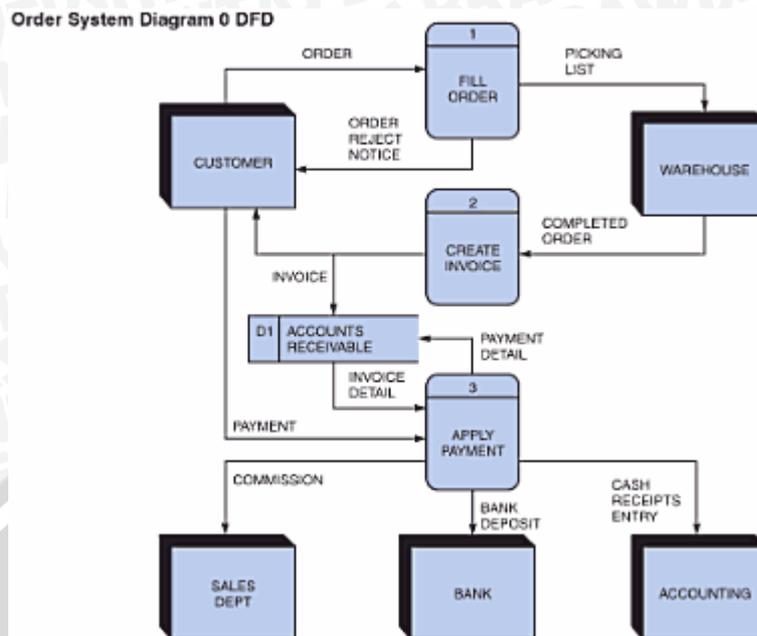
3. Diagram tingkat yang lebih rendah

Diagram tingkat yang lebih rendah adalah diagram yang menguraikan proses apa yang ada di dalam diagram 0. Diagram ini merupakan hasil pengembangan dari diagram tingkat di atasnya ke dalam komponen yang lebih detail.

Berikut ini adalah contoh gambar DFD:



Gambar 2.3 Diagram Konteks untuk Sebuah Sistem Pemesanan
Sumber: Shelly dan Rosenblatt (2012)



Gambar 2.4 Diagram 0 untuk Sebuah Sistem Pemesanan
Sumber: Shelly dan Rosenblatt (2012)

2.6.3 ERD (*Entity Relationship Diagram*)

ERD adalah model yang menunjukkan hubungan dan interaksi logis di antara entitas sistem (Shelly dan Rosenblatt, 2012). ERD memberikan pandangan menyeluruh dari sistem dan rancangan untuk membuat struktur data fisik. Beberapa notasi simbolik yang dapat digunakan untuk menggambarkan ERD, antara lain dapat dilihat pada tabel 2.2.

Dalam ERD terdapat kardinalitas relasi. Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Terdapat tiga macam kardinalitas relasi yaitu:

1. *One to One*

Tingkat hubungan satu ke satu dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.

2. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak dapat dinyatakan dengan satu kejadian pada entitas yang pertama mempunyai banyak hubungan dengan kejadian pada entitas kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama. Begitu juga sebaliknya untuk tingkat hubungan banyak ke satu.

3. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika setiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya.

Tabel 2.2 Simbol ERD

	IDEF1X	Chen	Crow's Foot
<p>An ENTITY</p> <ul style="list-style-type: none"> ✓ is a person, place, or thing. ✓ has a singular name spelled in all capital letters. ✓ has an identifier. ✓ should contain more than one instance of data. 	<p>ENTITY-NAME</p>	<p>ENTITY-NAME</p>	<p>ENTITY-NAME</p>
<p>An ATTRIBUTE</p> <ul style="list-style-type: none"> ✓ is a property of an entity. ✓ should be used by at least one business process. ✓ is broken down to its most useful level of detail. 	<p>ENTITY-NAME</p>		<p>ENTITY-NAME</p>
<p>A RELATIONSHIP</p> <ul style="list-style-type: none"> ✓ shows the association between two entities. ✓ has a parent entity and a child entity. ✓ is described with a verb phrase. ✓ has cardinality (1 : 1, 1 : N, or M : N). ✓ has modality (null, not null). ✓ is dependent or independent. 	<p><u>Relationship-name</u></p>		<p><u>Relationship-name</u></p>

Sumber: Dennis, Wixom, dan Roth (2012)

2.7 **Internet** (*Interconnection Networking*)

Internet merupakan sebuah sistem global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia (Iskandar, 2009 dalam Palupi, 2013). Dalam mengatur integrasi dan komunikasi jaringan komputer ini digunakan protokol yaitu TCP/IP. TCP (*Transmission Control Protocol*) bertugas memastikan bahwa semua hubungan bekerja dengan benar, sedangkan IP



(*Internet Protocol*) yang mentransmisikan data dari satu komputer ke komputer lain. TCP/IP secara umum berfungsi memilih rute terbaik transmisi data, memilih rute alternatif jika suatu rute tidak dapat digunakan, mengatur dan mengirimkan paket-paket pengiriman data.

Untuk dapat ikut serta menggunakan fasilitas internet, pengguna biasanya harus berlangganan ke salah satu ISP (*Internet Service Provider*). ISP dapat disebut sebagai penyelenggara jasa internet. Dengan memanfaatkan internet, pemakaian komputer di seluruh dunia dimungkinkan untuk saling berkomunikasi dan pemakaian bersama informasi dengan cara saling kirim surat elektronik (*e-mail*), menghubungkan ke komputer lain, mengirim dan menerima *file*, membahas topik tertentu pada *newsgroup* dan aktivitas lainnya.

Adapun fasilitas-fasilitas yang dapat dimanfaatkan dengan menggunakan internet, di antaranya adalah sebagai berikut:

1. *Web*, adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, animasi, dan data multimedia lainnya yang di antara data tersebut saling berhubungan satu sama lain. Untuk memudahkan membaca data dan informasi tersebut, dapat digunakan *web browser* seperti *Internet Explorer*, *Mozilla Firefox*, *Google Chrome*, ataupun *Netscape*.
2. *E-Mail (Electronic Mail)*, dengan fasilitas ini kita dapat mengirim dan menerima surat elektronik pada/dari pemakai komputer lain yang terhubung di internet, dan dapat menyertakan *file* sebagai lampiran (*attachment*).
3. *Newsgroup*, fasilitas ini digunakan untuk mendistribusikan artikel, berita, tanggapan, surat, penawaran ataupun *file* ke pemakai internet lain yang tergabung dengan kelompok diskusi, seminar, ataupun konferensi dengan cara elektronik tanpa terikat waktu, ruang, dan tempat.
4. *FTP (File Transfer Protocol)*, fasilitas ini digunakan untuk menghubungkan ke *server* komputer tertentu dan bila perlu menyalin (*download*) *file* yang dibutuhkan dari *server* tersebut dan menyimpannya di komputer.

Berikut ini adalah istilah-istilah yang sering digunakan dalam internet, di antaranya yaitu:

1. *WWW (World Wide Web)*, merupakan kumpulan *web server* dari seluruh dunia yang berfungsi menyediakan data dan informasi untuk digunakan bersama. Berbagai informasi dapat ditemukan pada *WWW*, seperti informasi politik, ekonomi, sosial, budaya, sastra, sejarah, teknologi, atau pendidikan. *WWW* dapat diumpamakan sebagai perpustakaan besar yang menyediakan berbagai informasi yang dibutuhkan.
2. *Website (situs web)*, merupakan tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. *Website* dapat diumpamakan sebuah buku yang berisi topik tertentu.
3. *Web pages (halaman web)*, merupakan sebuah halaman khusus dari situs *web* tertentu. *Web pages* dapat diumpamakan sebuah halaman khusus buku dari situs *web* tertentu.
4. *Homepage*, merupakan sampul halaman yang berisi daftar isi atau menu dari sebuah situs *web*.
5. *Browser*, merupakan program aplikasi yang digunakan untuk memudahkan melakukan navigasi berbagai data dan informasi pada *WWW*.

Secara umum ada banyak manfaat yang dapat diperoleh apabila seseorang mempunyai akses internet, antara lain sebagai berikut:

1. Informasi untuk kehidupan pribadi: kesehatan, rekreasi, hobi, pengembangan pribadi, rohani, sosial.
2. Informasi untuk kehidupan profesional/pekerja: sains, teknologi, perdagangan, saham, komoditas, berita bisnis, asosiasi profesi, asosiasi bisnis, berbagai forum komunikasi.

2.8 SMS Gateway

SMS Gateway merupakan pintu gerbang bagi penyebaran informasi dengan menggunakan *SMS (Short Message Service)*. Pengguna dapat menyebarkan pesan ke banyak nomor secara otomatis dan cepat yang langsung terhubung dengan *database nomor handphone*, tanpa harus mengetik ratusan nomor dan pesan pada *handphone*, karena semua nomor akan diambil secara otomatis dari *database* tersebut (Tarigan, 2012 dalam Zoni, 2013).

Selain itu, dengan SMS *Gateway* pengguna dapat mengelola pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita, yang mana pada umumnya pesan yang ingin dikirim berbeda-beda untuk masing-masing penerimanya. Salah satu fitur dalam SMS *Gateway* adalah SMS *alert* yaitu PC *Server* SMS memonitor kejadian-kejadian yang telah ditentukan oleh administrator, dan mengirim pemberitahuan kejadian lewat SMS ke HP Pengguna yang terdaftar.

2.9 Software yang Dipakai

Pada penelitian ini ada beberapa *software* yang dipakai untuk membangun aplikasi *m-business*. *Software* tersebut dibagi menjadi tiga yaitu *software* untuk pembangunan aplikasi *web*, pembangunan aplikasi *Java Mobile*, dan pembangunan aplikasi SMS *Gateway*.

2.9.1 Software Pembangunan Aplikasi Web

Aplikasi *web* yang dibangun akan dijadikan sebagai aplikasi *web server* yang menyediakan data jumlah barang dan data penjualan bagi *supplier*. *Software* yang digunakan meliputi PHP dan PHPMyAdmin, MySQL, dan XAMPP.

2.9.1.1 PHP (*PHP Hypertext Preprocessor*) dan PHPMyAdmin

PHP adalah bahasa *server side scripting* yang menyatu dengan HTML (*Hypertext Markup Language*) untuk membuat halaman *web* yang dinamis (Sunarfrihantono, 2002 dalam Palupi, 2013). Secara khusus, PHP memang dirancang untuk membentuk *web* dinamis, artinya PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini.

PHPMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan PHPMyAdmin, kita dapat membuat *database*, membuat tabel, memasukkan, menghapus, dan meng-*update* data dengan GUI (*Graphic User Interface*), tanpa perlu mengetik perintah SQL secara manual. Karena berbasis *web*, maka PHPMyAdmin dapat dijalankan di banyak sistem operasi selama dapat menjalankan *webserver* dan MySQL.

2.9.1.2 MySQL

MySQL merupakan *multi user database* yang menggunakan bahasa SQL (*Structured Query Language*) (Sunarfrihantono, 2002 dalam Palupi, 2013). MySQL mampu untuk melakukan banyak eksekusi perintah *query* dalam satu permintaan (*multithread*), baik itu menerima dan mengirimkan data. MySQL tersedia sebagai *software* gratis di bawah lisensi GNU *General Public License* (GPL). MySQL mempunyai kelebihan sebagai berikut:

1. Bebas diunduh (*free*).
2. Stabil dan tangguh.
3. Fleksibel dengan berbagai pemrograman.
4. Keamanan yang baik.
5. Dukungan dari banyak komunitas.
6. Kemudahan manajemen *database*.
7. Perkembangan *software* yang cukup cepat.

2.9.1.3 XAMPP

XAMPP adalah sebuah paket kumpulan *software* yang terdiri dari Apache, MySQL, PHPMyAdmin, PHP, Perl, Freetype2, dan lain-lain. XAMPP berfungsi untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan *web* memerlukan PHP, apache, MySQL, dan PHPMyAdmin serta *software-software* yang terkait dengan pengembangan *web*. Dengan menggunakan XAMPP, pengguna tidak perlu menginstal aplikasi-aplikasi tersebut satu persatu (Palupi, 2013).

2.9.2 Software Pembangunan Aplikasi Java Mobile

Java *Mobile* merupakan teknologi *mobile* yang menggunakan bahasa Java yang dapat dijalankan di berbagai perangkat yang mendukung Java (Mardiono, 2006). Aplikasi Java *Mobile* yang dibangun akan dijadikan sebagai aplikasi di sisi *client* yang berfungsi untuk mengambil data dari *web server*. *Software* yang digunakan adalah J2ME.

2.9.2.1 J2ME (Java 2 Mobile Edition)

J2ME adalah bingkai kerja pengembangan aplikasi *mobile* yang dikeluarkan oleh perusahaan Sun Java. Dengan konsep *write once run everywhere* (tulis program sekali, jalan di manapun) membuat banyak pengembang aplikasi *mobile* mengadopsi Java (Mardiono, 2006).

Selain itu, J2ME mengadopsi konsep *smart client*. Konsep ini berbicara tentang kemampuan *client* untuk meminta dan menangkap pesan dari *server*. Dalam proses meminta dan menangkap itu, *client* melakukan proses validasi dan verifikasi data. Keunggulan Java secara umum adalah:

1. *Multiplatform*: Aplikasi J2ME dapat berjalan di banyak *platform* yang di dalamnya terdapat JVM (Java *Virtual Machine*). Beberapa *platform* yang sudah menyediakan *installer mobile* JVM antara lain: Windows CE, Symbian, *embedded* Linux, dan sebagainya.
2. *Robust*: Kode-kode Java adalah kode-kode yang *robust* (kokoh), karena *virtual Machines* mengatur keamanan proses eksekusi aplikasi. JVM menyediakan *garbage collector* yang bertugas mencegah kebocoran memori.
3. Terintegrasi dengan baik: J2ME dapat terhubung dengan *web services* dengan mudah, karena J2ME menyediakan *library-library web services*.

2.9.3 Software Pembangunan Aplikasi SMS Gateway

Aplikasi *SMS Gateway* yang dibangun akan dijadikan sebagai aplikasi pengirim SMS ke *supplier* jika jumlah dari suatu barang sudah mencapai titik minimal. *Software* yang digunakan adalah Gammu.

2.9.3.1 Gammu (GNU All Mobile Management Utilities)

Gammu adalah sebuah aplikasi yang dikhususkan untuk membangun sebuah *SMS Gateway* yang menghubungkan antara operator *seluler* ke *SMS server* dan sebaliknya. Aplikasi ini bersifat *open source* dibawah lisensi GPL (*General Public License*). Aplikasi ini tersedia dalam versi Linux dan Windows (Fikri, 2007 dalam Saragih 2012).

Aplikasi ini menyediakan fitur yang lengkap untuk manajemen Gammu. Sebenarnya, setelah sukses membangun Gammu, maka kita bisa membangun aplikasi SMS Gateway dengan bahasa program atau platform apapun, baik itu *web based* dengan PHP (*Personal Home Page*) maupun ASP (*Active Server Pages*), dan juga *desktop* dengan menggunakan Delphi, VB atau lainnya. Saat ini, Gammu sudah *men-support* HP/Modem GSM terbaru sehingga tidak perlu khawatir dengan masalah kompatibilitasnya.

Kelebihan Gammu daripada *tools* SMS Gateway lainnya adalah :

1. Gammu dapat di jalankan di Windows maupun Linux.
2. Banyak *device* yang kompatibel oleh Gammu.
3. Gammu menggunakan *database* MySQL.
4. Kabel data USB maupun serial kompatibel di Gammu

