

BAB II

TINJAUAN PUSTAKA

Pada bagian tinjauan pustaka dijelaskan mengenai penelitian terdahulu serta referensi yang dapat menunjang permasalahan yang akan diteliti. Referensi tersebut berkaitan dengan *supply chain management*, *vendor managed inventory* (VMI), sistem informasi, *cloud computing*, dan kerangka pikiran dari penelitian yang akan dilakukan. Bab ini bertujuan mendukung permasalahan yang akan diteliti serta mendukung hasil penelitian.

2.1 Penelitian Terdahulu

Beberapa penelitian telah dilakukan berkenaan dengan analisis sistem informasi untuk kegiatan *supply chain management* di berbagai bidang dan dijadikan sebagai referensi dalam penelitian ini. Berikut merupakan *review* dari beberapa penelitian yang berkaitan dengan penelitian mengenai sistem informasi manajemen saat ini.

1. Santoso, Choiri, Rahman (2013), integrasi *supplier*, produsen, dan pelanggan pada UKM Keramik Dinoyo dengan *cloud computing*, UKM Keramik Dinoyo memerlukan pengelolaan *supply chain* mengenai pemesanan maupun sistem peramalan, proses produksi dan proses distribusi kepada pelanggan. Sistem informasi *cloud computing* dirancang dengan membangun *database* yang memberikan akses kepada *customer* maupun *supplier* dengan lebih mudah dan cepat.
2. Renitasari (2014), penentuan kebijakan order produk *skincare* dan *plaster* dengan pendekatan *vendor managed inventory*. Permasalahan proses *supply chain* pada perusahaan ini yaitu tidak dapat memenuhi permintaan distributor disebabkan informasi yang tidak akurat. Oleh karena itu dilakukan kebijakan order dengan metode *vendor managed inventory* (VMI) dengan memberikan informasi permintaan pelanggan, persediaan serta informasi lain yang mempengaruhi *demand* di masa depan. Berdasarkan hasil skenario dibandingkan metode tradisional dengan metode VMI menghasilkan total biaya dan *service level* yang lebih baik.
3. Yardinal dan Rusdiansyah (2014), pengkajian konsep *vendor managed inventory* sebagai alternatif sistem *advanced payment* pada distribusi BBM. Kebijakan *prepayment* pada sistem distribusi BBM *retail* oleh SPBU membentuk *bullwhip effect* dan berdampak bagi seluruh rantai pasok. Pendekatan konsep VMI mengatasi

permasalahan koordinatif sistem rantai pasok antara Terminal BBM Surabaya Group dengan SPBU yang disuplainya.

4. Febriani (2016), perancangan sistem informasi paguyuban keramik dinoyo malang berbasis *cloud computing* dengan menggunakan *office 365* dan *website*. Pemasalahan kurangnya koordinasi dan kolaborasi dalam metode terintegrasi antar UMKM dan kurangnya informasi pemasaran yang didapat pelanggan. *Cloud computing* pada sistem informasi paguyuban akan mampu membantu menyelesaikan permasalahan koordinasi, kolaborasi, integrasi antar UMKM dan meningkatkan informasi pemasaran pelanggan dengan *website*.
5. Hafidz (2016), perancangan sistem basis data persediaan *spare parts* berdasarkan frekuensi pemakaian pada AHASS 1380 Haur Kuning Motor. Pengolahan data persediaan *part* menggunakan *Dealer Integrated Management System (DIMS)*. Namun mengalami keterlambatan *update data* karena proses input menunggu *file* khusus dari *main dealer* dan tidak ada sistem *alert* untuk meningkatkan stok kritis. Metode SDLC digunakan dalam pembuatan program dengan *Microsoft Access 2013* untuk mengatasi keterlambatan *update data* karena sistem input data sudah *real time* dan program memiliki sistem *alert*.

Tabel 2.1 menjelaskan penelitian terdahulu yang digunakan sebagai acuan dalam melakukan penelitian serta gambaran umum penelitian yang dilakukan.

Tabel 2.1
Perbandingan Penelitian Terdahulu dan Penelitian Ini

Peneliti	Objek Penelitian	Metode dan Pembahasan	Hasil Penelitian
Santoso, Rahman dan Choiri (2013)	UKM Keramik Dinoyo	Pembuatan sistem <i>database</i> menggunakan aplikasi <i>cloud computing</i> dengan layanan <i>zoho.com</i> dengan metode <i>prototyping</i>	Sistem <i>database</i> menggunakan aplikasi <i>cloud computing</i> <i>zoho.com</i> mampu mengintegrasikan <i>supplier</i> , produsen dan pelanggan dengan <i>sharing database</i>
Renitasari (2014)	PT Beirsdorf Indonesia	Penerapan pendekatan <i>vendor managed inventory</i> untuk menentukan kebijakan order yang paling optimal menggunakan metode simulasi	Penerapan kebijakan VMI menyebabkan peningkatan <i>total cost inventory</i> dan <i>service level</i> sehingga penerapan VMI lebih baik dibandingkan kebijakan eksisting
Yardinal dan Rusdiansyah (2014)	Terminal BBM Surabaya Group	Meminimalisir terjadinya <i>bullwhip effect</i> dengan konsep VMI dengan menggunakan simulasi	Konsep VMI dapat meminimalisir terjadinya <i>Bullwhip Effect</i> dibuktikan dengan penurunan biaya persediaan dari sistem eksisting
Febriani (2016)	Paguyuban Keramik	Perancangan sistem informasi untuk pemasaran dan <i>supply chain</i> menggunakan <i>cloud</i>	Penggabungan sistem pemasaran dan sistem terintegrasi antar UMKM

Peneliti	Objek Penelitian	Metode dan Pembahasan	Hasil Penelitian
	Dinoyo Malang	<i>computing</i> dan <i>tools website</i> dan <i>office 365</i>	menjawab permasalahan sistem pemasaran dan sistem <i>inventory</i>
Hafidz (2016)	AHASS 1380 Haur Kuning Motor	Pembuatan Sistem Informasi Inventori <i>Spare Part</i> (SIISP) dirancang dengan metode SDLC dengan menggunakan <i>Microsoft Office 2013</i>	SIISP menyediakan informasi pemesanan, penerimaan, dan pengeluaran <i>part</i> dalam bentuk <i>report</i>
Penelitian ini	PT Indofarma Global Medika Malang	Pengendalian <i>inventory retailer</i> menggunakan pendekatan <i>vendor managed inventory</i> dengan dilakukan perancangan sistem informasi berbasis <i>cloud computing</i>	Penerapan kebijakan VMI pada perancangan sistem informasi berbasis <i>cloud computing</i> dapat menyelesaikan permasalahan koordinasi dan kolaborasi antara distributor dengan <i>retailer</i>

2.2 Pengendalian Pemasok dan Sistem Distribusi Persediaan

Pemasok (*supplier*) memiliki peran penting untuk kerja sama dalam meningkatkan pelayanan pelanggan. Menurut Gaspersz (1998), terdapat 4 (empat) prinsip penting dalam memahami rantai proses manufaktur bernilai tambah (*value added manufacturing process*) dalam sistem industri manufaktur modern, yaitu:

1. Masing-masing memainkan peran sebagai pemasok, pembuat, dan pelanggan
2. Kebutuhan mengendalikan hubungan diantara rantai proses manufakturing
3. Peran pembuat dalam rantai proses manufakturing akan melaksanakan proses yang dibatasi oleh *input* dan *output*
4. Umpan balik (*feedback*) merupakan kunci perbaikan proses manufakturing secara terus menerus

2.3 Supply chain Management

Supply chain merupakan jaringan dari beberapa perusahaan yang bekerja sama untuk menciptakan dan menghantarkan suatu produk ke tangan pemakai akhir. Perusahaan tersebut diantaranya *supplier*, pabrik, *distributor*, *retailer* serta perusahaan pendukung lainnya (Pujawan dan Mahendrawathi, 2010).

Menurut Heizer dan Render (2011), perencanaan dan pengelolaan dari seluruh aktivitas yang melibatkan pengadaan, pengonversian, dan seluruh kegiatan logistik. Hal yang paling penting dari *supply chain* yaitu mengenai koordinasi dan kolaborasi dengan partner seperti *supplier*, perantara, pihak ketiga dan *customer*.

Persediaan muncul dengan berbagai bentuk dan fungsi di sepanjang *supply chain*. Mengelola aliran material/produk dengan tepat merupakan salah satu tujuan *supply chain*.

Aliran yang tepat merupakan aliran yang tidak mengalami keterlambatan namun juga tidak terlalu dini, jumlah yang sesuai kebutuhan, dan dapat terkirim ke tempat yang sedang membutuhkan. Persediaan dapat muncul karena berbagai ketidakpastian baik yang sudah direncanakan maupun akibat dari kekurangan informasi. Menurut Pujawan dan Mahendrawathi (2010), ketidakpastian dalam *supply chain* tidak hanya muncul karena permintaan namun juga dari arah pasokan dan operasi internal. Ketidakpastian pengiriman dan harga bahan baku menyebabkan penimbunan persediaan. Ketidakpastian pengiriman menyebabkan perlunya penyimpanan persediaan cadangan (*safety stock*). Ketidakpastian proses internal juga menyebabkan penyimpanan cadangan barang setengah jadi (WIP). Selain ketidakpastian, perbedaan lokasi membuat munculnya *lead time* pengiriman. Oleh karena itu, perlu dilakukan pengendalian persediaan di sepanjang *supply chain*.

2.4 Vendor managed Inventory (VMI)

Chandra dan Kumar (2000) mengemukakan beberapa cara yang telah dilakukan untuk mengatasi permasalahan persediaan salah satunya adalah dengan *vendor managed inventory*. Perusahaan pembeli mengubah model pemenuhan permintaan dengan model *vendor managed inventory* (VMI) dimana perusahaan pembeli tidak lagi memberikan informasi permintaan dari pelanggan mereka, persediaan tersisa, serta informasi lain seperti rencana promosi atau kegiatan lain yang bisa mempengaruhi penjualan di masa yang akan datang. Dengan mengetahui informasi tersebut, pemasok akan menentukan waktu dan jumlah pengiriman ke perusahaan pembeli (Pujawan dan Mahendrawathi, 2010).

Menurut Simchilevi (2000), terdapat dua langkah utama dalam mengimplementasikan VMI diantaranya:

1. Ketentuan perjanjian kontrak yang harus dinegosiasikan termasuk keputusan mengenai tanggung jawab pemesanan yang diperlukan.
2. Hal-hal yang harus dieksekusi yaitu dengan mengembangkan sistem informasi terintegrasi antara *supplier* dengan *retailer*, mengembangkan teknik peramalan yang efektif untuk digunakan, serta mengembangkan *decision support tools* untuk membantu mengoordinasikan kebijakan manajemen persediaan dan transportasi.

Adanya koordinasi dan pertukaran informasi dengan menerapkan model VMI adalah dengan memiliki infrastruktur komunikasi dan informasi yang bagus sehingga pembeli dapat memberikan data penjualan dan persediaan dari waktu ke waktu serta pemasok dapat mengambil keputusan pengiriman dengan tepat. Kemampuan analisis pola permintaan, *lead*

time pengiriman dan peramalan permintaan yang dilakukan oleh pemasok (Pujawan dan Mahendrawathi, 2010).

2.5 Persediaan Pengaman (*Safety stock*)

Menurut Rangkuti (2004), persediaan pengaman merupakan persediaan tambahan untuk melindungi kemungkinan terjadinya kekurangan bahan (*stock out*). Sedangkan menurut Hansen dan Mowen (2005), *safety stock* dapat dihitung dari perkalian tenggang waktu dengan selisih antara tingkat penggunaan bahan baku maksimal dengan tingkat rata-rata penggunaan bahan baku. *Safety stock* bertujuan untuk menentukan berapa stok yang dibutuhkan selama masa tenggang untuk memenuhi besarnya permintaan. Menurut Herjanto (2008), rumus untuk menentukan *safety stock* adalah:

$$SS = Z \times SD \quad (2-1)$$

Keterangan: $SS = \text{Safety stock}$

$Z = \text{Standar penyimpangan}$

$SD = \text{Standar deviasi}$

2.6 Reorder Point (ROP)

ROP atau batas /titik jumlah pemesanan kembali termasuk permintaan yang diinginkan atau dibutuhkan selama masa tenggang, misalnya tambahan/ekstra stok (Rangkuti, 2004). Sedangkan menurut Hansen dan Mowen (2005), *reorder point* adalah titik waktu dimana pesanan baru harus dilakukan. Model-model *reorder point* menurut Rangkuti (2004), yaitu:

1. *Constant demand rate, constant lead time*, besar permintaan maupun masa tenggang konstan sehingga tidak ada penambahan persediaan.

$$ROP = \text{kebutuhan} \times \text{lead time} \quad (2-2)$$

2. *Variable demand rate, constant lead time*, periode *lead time* atau masa tenggang tidak tergantung pada permintaan harian yang digambarkan melalui distribusi normal

$$ROP = \bar{d}LT + Z\sqrt{LT}(\sigma d) \quad (2-3)$$

Keterangan: $\bar{d} = \text{rata-rata tingkat kebutuhan}$

$LT = \text{masa tenggang}$

$\sigma d = \text{standar deviasi dan tingkat kebutuhan}$

3. *Contant demand rate, variable lead time, lead time* pada kondisi distribusi normal, diharapkan permintaan selama *lead time* pada kondisi normal, tetapi variannya tidak mencakup perhitungan atau penjualan barian model sebelumnya.

$$ROP = d \overline{LT} + z d \sigma LT \quad (2-4)$$

Keterangan: d = tingkat permintaan konstan

\overline{LT} = rata-rata masa tenggang

σLT = standar deviasi dan tingkat kebutuhan

4. *Variable demand rate, variable lead time*, besarnya permintaan dan masa tenggang dapat berubah-ubah sesuai dengan perubahan masa tenggang

$$ROP = \bar{d} \overline{LT} + z \sqrt{\overline{LT} \sigma d^2 + \bar{d}^2 \sigma \overline{LT}^2} \quad (2-5)$$

2.7 Teknologi Informasi dalam *Supply chain Management*

Menurut Pujawan dan Mahendrawathi (2010), informasi dalam pengambilan keputusan rantai pasok oleh manajer harus diperhitungkan untuk seluruh rantai pasok sehingga didapatkan keputusan yang optimal karena keputusan dalam satu perusahaan dapat meliputi seluruh aspek mengenai manufaktur, distribusi dan ritel serta permintaan akan membantu dalam pengambilan keputusan. Menurut Chopra dan Meindl (2007), suatu informasi dapat berguna dalam pengambilan keputusan rantai pasok apabila memenuhi karakteristik berikut.

1. Akurat maksudnya adalah informasi dapat menggambarkan kondisi sebenarnya dan tidak mengandung banyak kesaahan.
2. Tepat maksudnya adalah informasi yang digunakan merupakan informasi yang sesuai dengan kebutuhan. Sehingga perusahaan dapat mengumpulkan informasi, menyimpan dan memelihara informasi yang sesuai kebutuhan dalam pengambilan keputusan.
3. Dapat diakses saat dibutuhkan maksudnya adalah informasi yang dibutuhkan perusahaan harus tersedia saat informasi tersebut dibutuhkan sehingga informasi juga harus sesuai dengan perubahan waktu.

Menurut Rahman (2012), teknologi informasi memiliki fungsi yang berhubungan dengan aspek penyebaran terhadap entitas, data, informasi, *knowledge* dan *wisdom*. Aspek penyebaran tersebut diantaranya yaitu:

1. *Gathering* yaitu teknologi informasi memiliki fasilitas yang mampu mengumpulkan entitas-entitas dan mampu meletakkannya di dalam suatu media penyimpanan digital.
2. *Organizing* yaitu teknologi informasi memiliki mekanisme baku dalam mengorganisasikan penyimpanan entitas dalam media penyimpanan.
3. *Selecting* yaitu entitas yang sudah disediakan juga disertai dengan fasilitas teknologi informasi yang memudahkan pencarian dan pemilihan.

4. *Synthesizing* yaitu teknologi informasi harus memenuhi kebutuhan manajer dalam menggabungkan beberapa entitas menjadi satu paket kesatuan yang terintegrasi.
5. *Distributing* yaitu teknologi informasi memiliki infrastruktur yang dapat menyalurkan berbagai entitas dari tempat penyimpanan kepada pihak-pihak yang membutuhkannya.

2.8 Manajemen Pengadaan

Menurut Pujawan dan Mahendrawathi (2010), waktu merupakan salah satu faktor yang menentukan keberhasilan *supply chain* dalam persaingan industri. Kecepatan dan ketepatan waktu pengiriman memungkinkan untuk memenuhi permintaan pelanggan secara tepat serta dapat mengurangi biaya persediaan. Proses pembelian dapat dilakukan dengan pembelian tender maupun rutin. Proses pembelian rutin dilakukan untuk item kebutuhan yang berulang (*repetitive*). Dengan munculnya kecanggihan teknologi seperti internet, kegiatan pengadaan dapat dilakukan secara *online*. Selain dapat mengurangi *lead time* pengiriman, teknologi juga dapat mengendalikan tingkat persediaan dalam perusahaan.

Menurut Heizer dan Render (2011), permintaan pelanggan yang tidak stabil dan adanya peramalan yang berbeda menyebabkan rantai pasok menjadi satu hal yang penting dalam perkembangan industri. Informasi merupakan salah satu hal yang penting untuk menjembatani antar usaha guna menciptakan rantai pasok yang baik.

2.9 Sistem

Sistem sebagai seperangkat komponen yang saling terkait, bekerjasama untuk mencapai suatu tujuan dengan menerima input dan output dalam proses transformasi secara terorganisir (O'Brien dan Marakas, 2011).

Menurut Jogiyanto (1989), sistem merupakan sejumlah komponen atau subsistem yang memiliki sifat dalam menjalankan fungsi tertentu dan mempengaruhi sistem secara keseluruhan, yang saling berinteraksi dan bekerja sama membentuk satu kesatuan. Masing-masing subsistem terdiri dari subsistem-subsistem yang lebih kecil atau terdiri dari komponen-komponen. Suatu sistem memiliki karakteristik atau sifat tertentu, karakteristik tersebut harus dimiliki dalam satu sistem. Karakteristik dalam sistem tersebut diantaranya:

1. Komponen sistem, sistem terdiri dari sejumlah komponen yang saling berinteraksi dan saling bekerjasama membentuk suatu kesatuan.
2. Batas sistem (*Boundary*), batas sistem adalah daerah yang membatasi sistem dengan sistem lainnya atau dengan lingkungan sekitarnya.

3. Lingkungan luar sistem, lingkungan luar sistem mencakup semua hal yang diluar sistem yang mempengaruhi operasi sistem.
4. Penghubung sistem (*Interface*), *interface* adalah media penghubung antar subsistem. Melalui *interface*, sumber daya mengalir dari subsistem ke subsistem lainnya.
5. Masukan sistem (*Input*), *input* adalah semua hal yang dimasukkan ke dalam sistem. Masukan dapat berupa perawatan (*maintenance input*) maupun sinyal (*signal input*).
6. Keluaran sistem (*Output*), *output* adalah semua hal hasil olahan yang berguna. Keluaran dapat menjadi masukan (*input*) untuk subsistem lainnya.
7. Pengolah sistem (*Process*), pengolah sistem adalah bagian pengolah yang akan mengubah masukan menjadi keluaran.
8. Sasaran sistem (*Objective*), suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Sasaran sistem menentukan masukan yang dibutuhkan sistem yang akan dihasilkan sistem.

2.9.1 Sistem Informasi

Sistem informasi merupakan kombinasi teratur dari *people* (orang), *hardware* (perangkat keras), *software* (piranti lunak), *computer networks and data communications* (jaringan komunikasi) dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi (O'Brien & Marakas, 2011).

2.9.2 Analisis Sistem (PIECES)

Menurut Whitten dan Bentley (2007), dalam mengidentifikasi masalah, harus dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan yang dikenal dengan *PIECES analysis (performance, information, economy, control, efficiency dan services)*. Analisis ini menghasilkan permasalahan utama pada sistem yang diidentifikasi. Hal ini penting karena biasanya yang muncul dipermukaan bukan masalah utama, tetapi hanya gejala dari masalah utama saja.

PIECES framework diantaranya *performance*, apakah sistem menyediakan *throughput* dan *response time* yang memadai, *information*, apakah sistem menyediakan informasi yang tepat waktu, relevan, akurat dan bermanfaat, *economy*, apakah sistem menawarkan tingkat layanan dan kapasitas yang memadai untuk mengurangi *cost* atau meningkatkan *profit*, *control*, apakah sistem menawarkan kontrol untuk melindungi terhadap penipuan dan menjamin keakuratan dan keamanan data dan informasi, *efficiency*, apakah sistem memaksimalkan penggunaan sumber daya, dan *service*, apakah sistem menyediakan layanan

yang diinginkan dan dapat diandalkan bagi yang membutuhkan. Adapun penjelasan PIECES framework dapat dilihat pada Gambar 2.1

The Pieces Problem-Solving Framework and Checklist	
<p>The following checklist for problem, opportunity, and directive identification user Wetherbe's PIECES framework. Note that the categories of PIECES are not mutually exclusive; some possible problem show up in multiple lists. Also, the list of possible problem is not exhausted. The PIECES framework is equally suited to analyzing both manual and computerized systems and applications.</p>	
PERFORMANCE	
<ul style="list-style-type: none"> A. Throughput – the amount of work performed over some period of time B. Response times – the average delay between a transaction or request, and a response to that transaction or request. 	<ul style="list-style-type: none"> B. Profit <ul style="list-style-type: none"> 1. New markets can be explored 2. Current marketing can be improved 3. Orders can be increased
CONTROL (and Security)	
<p>INFORMATION (and Data)</p> <ul style="list-style-type: none"> A. Outputs <ul style="list-style-type: none"> 1. Lack of any information 2. Lack of necessary information 3. Lack of relevant information 4. Too much information – “information overload” 5. Information that is not in a useful format 6. Information that is not accurate 7. Information that is difficult to produce 8. Information is not timely to its subsequent use B. Inputs <ul style="list-style-type: none"> 1. Data is not captured 2. Data is not captured in time to be useful 3. Data is not accurately captured – contains errors 4. Data is difficult to capture 5. Data is captured redundantly – same data captured more than once 6. Too much data is captured 7. Illegal data is captured C. Stored Data <ul style="list-style-type: none"> 1. Data is stored redundantly in multiple files and/or <i>databases</i> 2. Same data items have different values in different files (poor data integration) 3. Stored data is not accurate 4. Data is not secure to accident or vandalism 5. Data is not well organized 6. Data is not flexible – not easy to meet new information needs from stored data 7. Data is not accessible 	<ul style="list-style-type: none"> A. Too little security or control <ul style="list-style-type: none"> 1. Input data is not adequately edited 2. Crimes (e.g., fraud, embezzlement) are (or can be) committed against data 3. Ethics are breached on data or information – refers to data or information getting to unauthorized people 4. Redundantly stored data is inconsistent in different files or <i>databases</i> 5. Data privacy regulation or guidelines are being (or can be) violated 6. Processing errors are occurring (either by people, machines, or software) 7. Decision-making errors are occurring B. Too much control or security <ul style="list-style-type: none"> 1. Bureaucratic red tape slows the system 2. Controls inconvenience customers or employees 3. Excessive controls cause processing delays
EFFICIENCY	
<p>ECONOMICS</p> <ul style="list-style-type: none"> A. Costs <ul style="list-style-type: none"> 1. Cost are unknown 2. Costs are untraceable to source 3. Costs are too high 	<ul style="list-style-type: none"> A. People, machines, or computer waste time <ul style="list-style-type: none"> 1. Data is redundantly input or copied 2. Data is redundantly processed 3. Information is redundantly generated B. People, machines, or computers waste materials and supplies C. Effort required for task is excessive D. Material required for task is excessive
SERVICE	
	<ul style="list-style-type: none"> A. The system produces inaccurate results B. The system produces inconsistent result C. The system produces unreliable result D. The system is not easy to learn E. The system is not easy to use F. The system is awkward to use G. The system is inflexible to new or exceptional situations H. The system is inflexible to change I. The system is incompatible with other systems

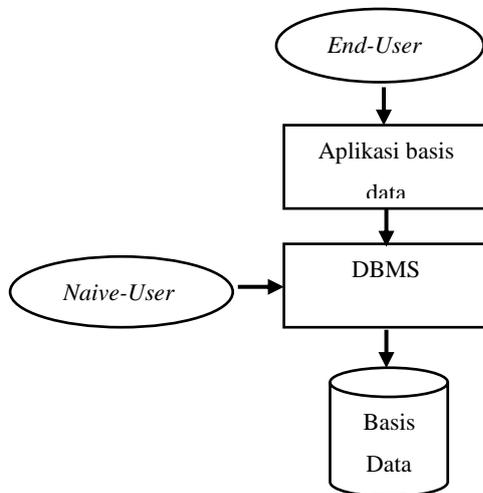
Gambar 2.1 PIECES framework

Sumber: Whitten dan Bentley (2007)

2.10 Database

Database merupakan kumpulan *file* yang memiliki kaitan antara satu *file* dengan *file* lain sehingga membentuk satu bangun data yang berfungsi untuk menginformasikan satu

perusahaan, instansi dalam batasan tertentu (Kristanto, 2004). *Database Management System* adalah kumpulan *file* yang saling berkaitan bersama dengan program untuk pengelolaannya. Jadi *database* merupakan kumpulan datanya, sedangkan DBMS merupakan program pengelolaan data. Bagan DBMS dapat dilihat pada Gambar 2.2.



Gambar 2.2 Bagan *database management system*

Sumber: Bina Nusantara (2010) <https://library.binus.ac.id>

Menurut Kristanto (2004), penyusunan satu *database* digunakan untuk mengatasi masalah-masalah penyusunan data, yaitu:

1. Redundansi dan inkonsistensi data
2. Kesulitan pengaksesan data
3. Isolasi data untuk standarisasi
4. *Multiple user*
5. Masalah keamanan
6. Masalah integrasi
7. Masalah data *independence*

Jadi DBMS adalah perangkat lunak yang berinteraksi dengan program aplikasi pengguna dan *database*. Menurut Kristanto (2004), DBMS menyediakan beberapa fasilitas sebagai berikut.

1. *Data Definition Language* (DDL) memungkinkan pengguna mendefinisikan tipe data (*data type*), struktur (*structure*) dan batasan–batasan (*constraints*) pada data yang disimpan ke dalam *database*.
2. *Data Manipulation Language* (DML) memungkinkan pengguna untuk memasukkan (*insert*), mengubah (*update*), menghapus (*delete*) dan menampilkan (*retrieve*) data dari *database*.

3. *Access control* menyediakan akses yang terkontrol ke *database*, seperti *security system*, *integrity system*, *concurrency control system*, *recovery control system*, dan *user-accessible catalog*.

2.11 Perancangan Sistem

Perancangan sistem menurut Al-Bahra (2005) adalah tahap yang memiliki tujuan untuk mendesain sistem baru dan dapat menyelesaikan masalah-masalah yang dihadapi perusahaan. Perancangan sistem memerlukan perlengkapan untuk memodelkan desain yang diperlukan, serta dibutuhkan persiapan dalam perancangan sistem. Berikut merupakan istilah yang digunakan dalam perancangan sistem.

1. *Data Flow Diagram* (DFD)

Proses *modelling* merupakan cara untuk menggambarkan bagaimana bisnis beroperasi. Mengilustrasikan aktivitas-aktivitas yang dilakukan dan bagaimana data berpindah diantara aktifitas-aktifitas itu. Ada banyak cara untuk merepresentasikan proses model, salah satunya dengan menggunakan *data flow diagram* (DFD). Terdapat 2 jenis DFD, *Logical DFD* yang menggambarkan proses tanpa menyarankan bagaimana mereka akan dilakukan. Kedua adalah Fisikal DFD yang menggambarkan proses model implementasi pemrosesan informasinya. Ada 4 elemen yang menyusun DFD yaitu:

a. Proses

Aktifitas atau fungsi yang dilakukan untuk alasan bisnis yang spesifik, biasa berupa manual maupun terkomputerisasi.

b. *Data flow*

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada suatu proses.

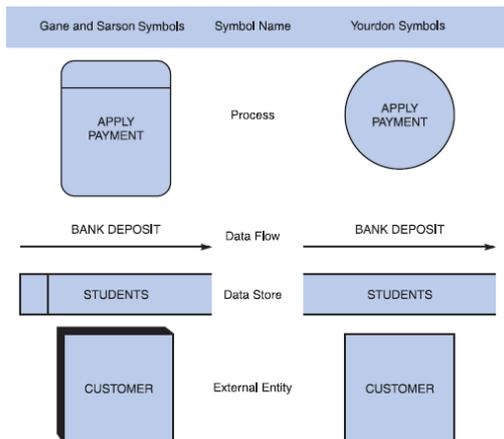
c. *Data Store*

Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam data store. Aliran data diupdate atau ditambahkan ke data store.

d. *External entity*

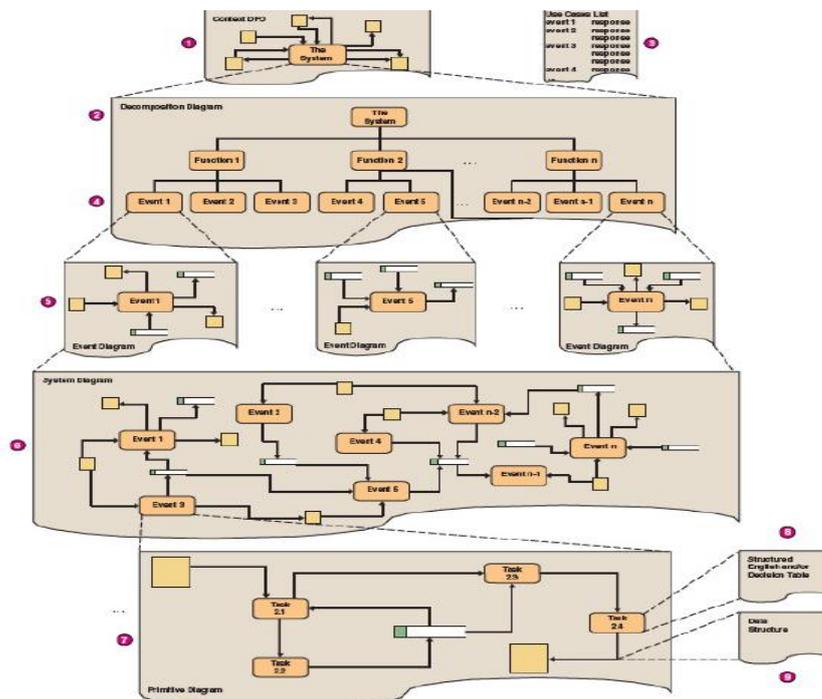
Orang, organisasi, atau sistem yang berada di luar sistem tetapi berinteraksi dengan sistem.

Masing-masing elemen akan diberi lambang tertentu untuk membedakan satu dengan yang lain. Ada beberapa metode untuk menggambarkan elemen-elemen tersebut. Untuk lebih jelasnya dapat dilihat pada Gambar 2.3.



Gambar 2.3 Elemen-elemen dari DFD dan lambangnya
Sumber: Cashman (2012)

Proses bisnis biasanya terlalu kompleks untuk ditunjukkan dalam 1 DFD. Dekomposisi adalah proses untuk menggambarkan sistem dalam hirarki dari diagram DFD. Diagram anak menggambarkan proses yang lebih detail dibandingkan dengan diagram induk. Harus ada proses *balancing* untuk menjamin informasi yang disajikan dalam satu level dari suatu DFD secara akurat direpresentasikan pada DFD level berikutnya. Adapun hirarki dari suatu DFD dapat dilihat pada Gambar 2.4.

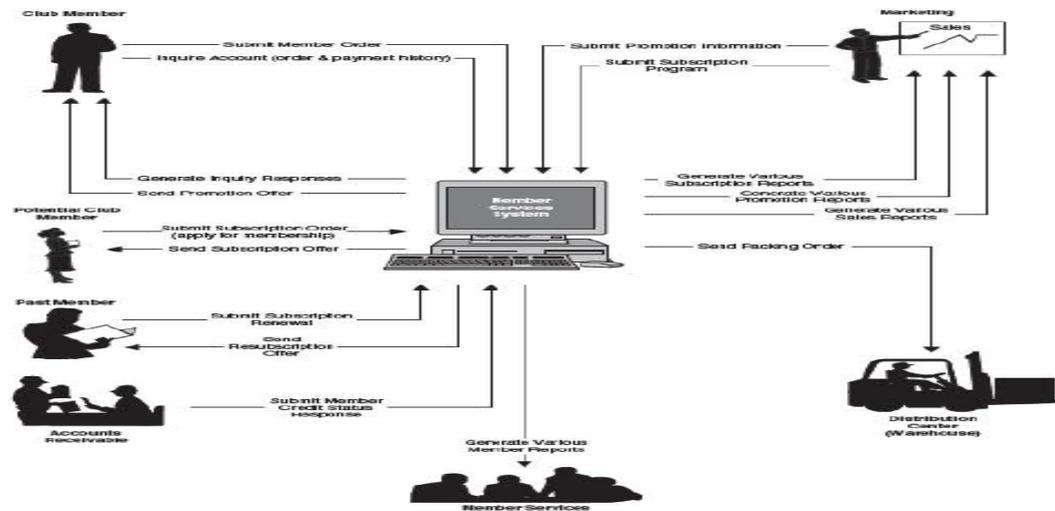


Gambar 2.4 Hirarki penyusunan DFD
Sumber: Whitten dan Bentley (2007)

Keterangan:

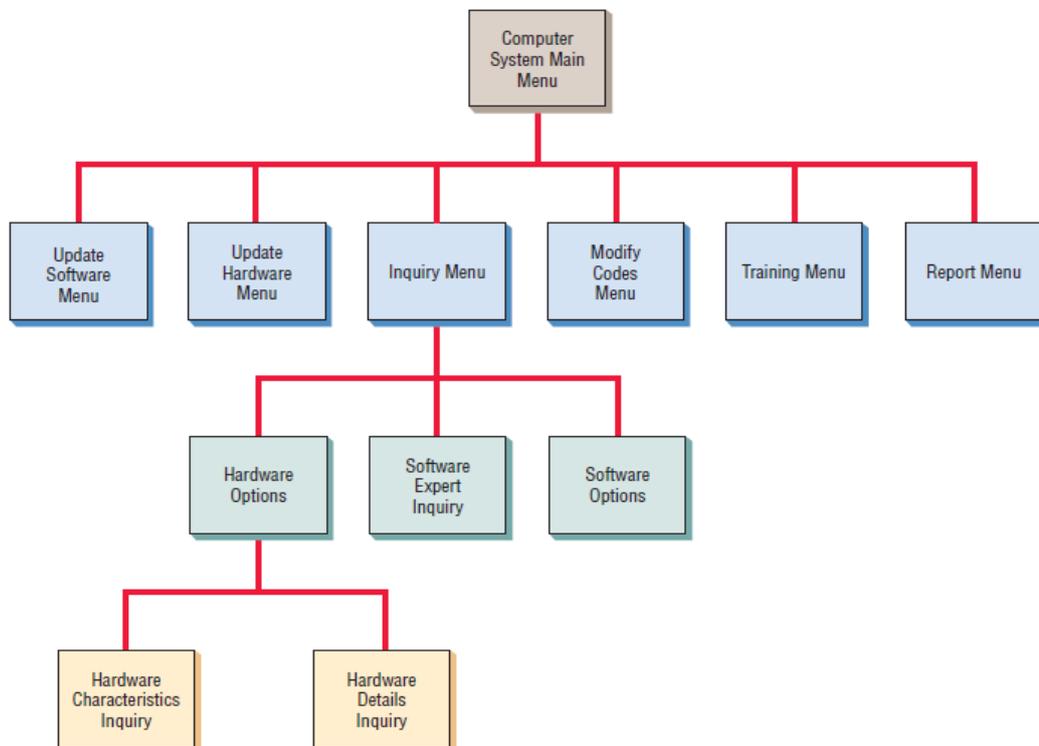
- Overview Diagram*: diagram yang mengilustrasikan gambaran umum sistem interaksi antara *stake holder* (*actor*) dengan sistem informasi, termasuk batasan

sistem yang memisahkan elemen sistem dan lingkungannya. Overview diagram dapat dilihat pada Gambar 2.5.



Gambar 2.5 Overview diagram
 Sumber: Whitten dan Bentley (2007)

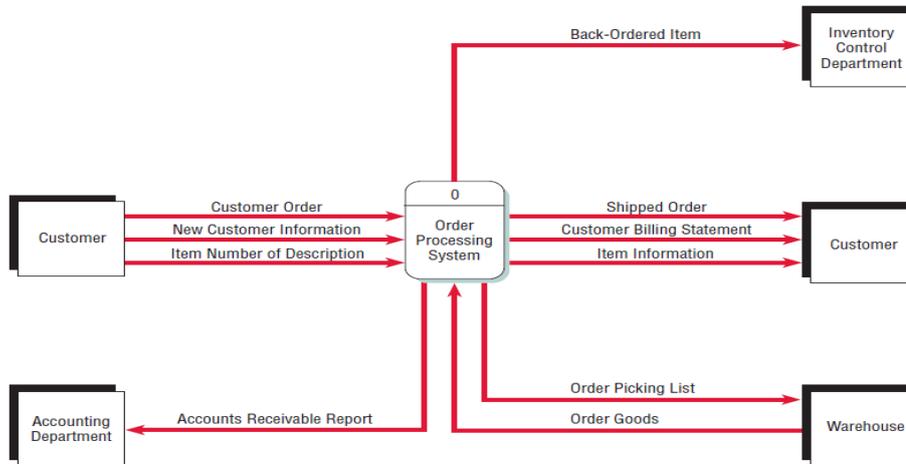
- b. *Hierarchy Chart Diagram:* diagram pohon yang menggambarkan hirarki dari sistem informasi yang diurai menjadi proses-proses bisnis yang lebih rinci di level lebih rendah.



Gambar 2.6 Hierarchy chart diagram
 Sumber: Whitten dan Bentley (2007)

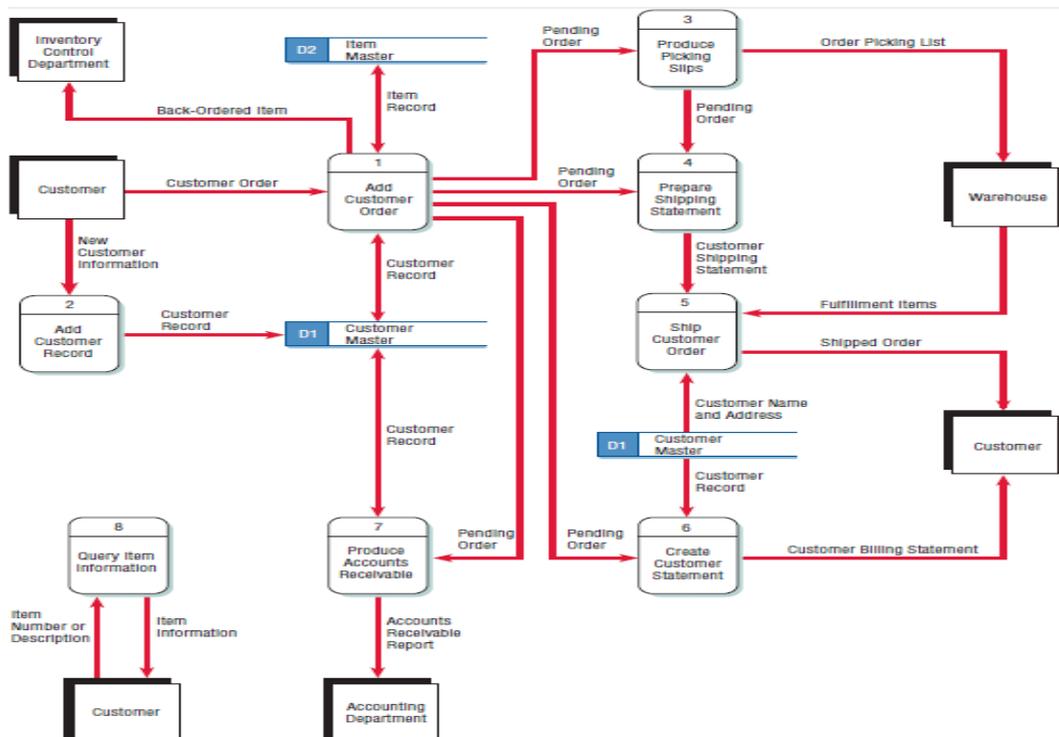
- c. *Context Diagram:* Menunjukkan konteks dimana proses bisnis berada. Selain itu, menunjukkan semua proses bisnis dalam 1 proses tunggal (proses 0). *Context*

Diagram juga menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke dalam sistem.



Gambar 2.7 Context diagram
 Sumber: Whitten dan Bentley (2007)

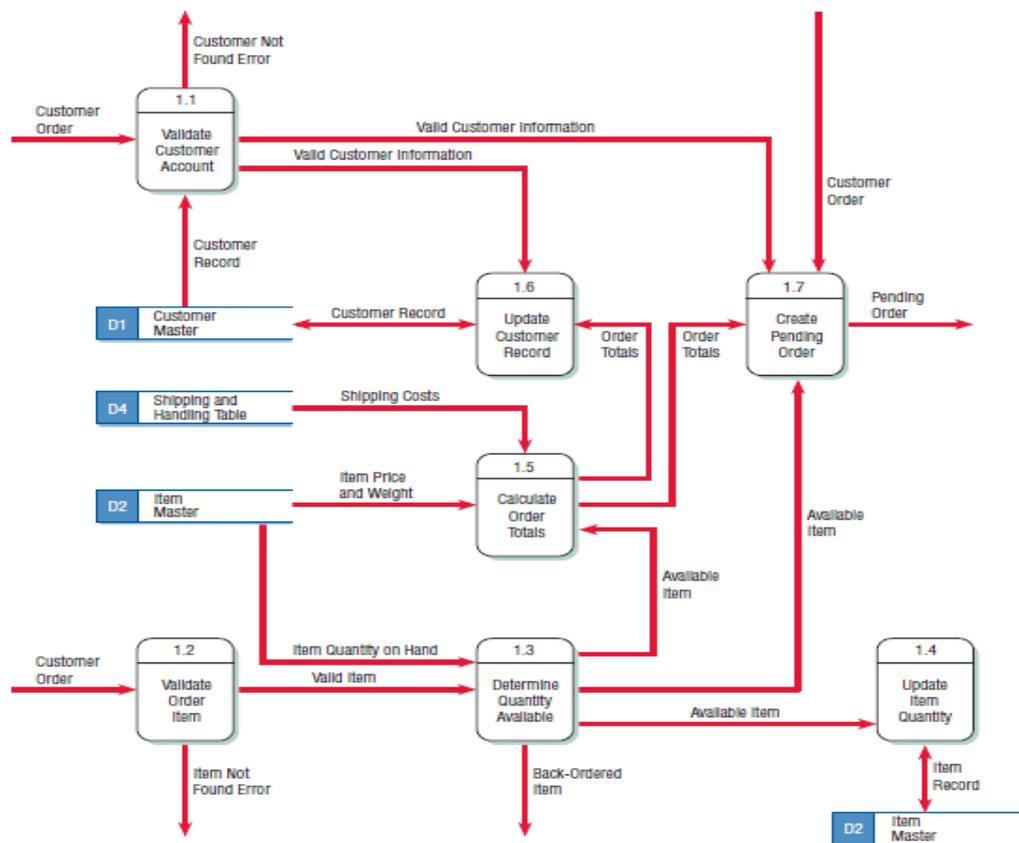
- d. *Level 0 Diagram: Level 0 Diagram* menunjukkan semua proses yang menyusun keseluruhan sistem. Level ini menunjukkan komponen internal dari proses 0 serta menunjukkan bagaimana proses-proses utama direlasikan. Selain itu, pada level ini juga ditunjukkan bagaimana proses-proses utama terhubung dengan entitas eksternal. Pada level ini juga dilakukan penambahan *data store*.



Gambar 2.8 DFD level 0
 Sumber: Whitten dan Bentley (2007)

- e. *Level 1 Diagrams: Diagram level 1* diciptakan dari setiap proses utama dari level 0. Level ini juga menunjukkan proses-proses internal yang menyusun setiap proses-

proses utama dalam level 0, sekaligus menunjukkan bagaimana informasi berpindah dari satu proses ke proses yang lainnya. Jika proses induk di pecah, menjadi proses anak, maka proses anak ini secara utuh menyusun proses induk.

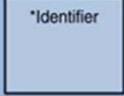
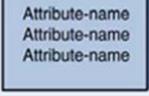
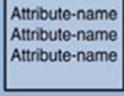
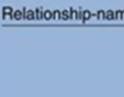


Gambar 2.9 DFD level 1

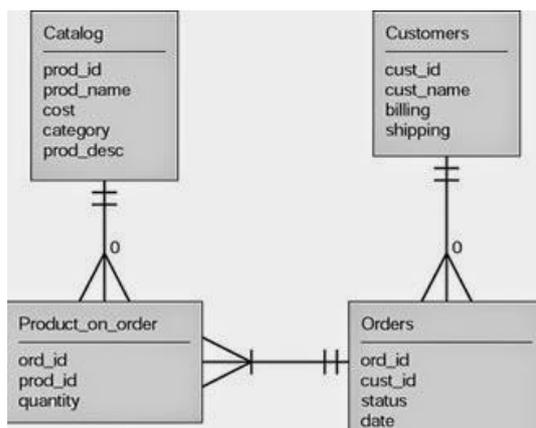
Sumber: Whitten dan Bentley (2007)

- f. *Level 2 Diagrams*: Menunjukkan semua proses yang menyusun proses pada level 1. Penyusunan DFD bisa saja tidak mencapai level 2 ini. Kemungkinan selanjutnya adalah mungkin saja DFD dilanjutkan ke level berikutnya (level 3, level 4).
2. *Entity Relation Diagram (ERD)*

ERD adalah gambar/diagram yang menunjukkan informasi dibuat disimpan dan digunakan dalam sistem bisnis. ERD juga menggambarkan entitas. Biasanya ERD menggambarkan jenis informasi yang sama serta menggambarkan garis yang menghubungkan antar entitas menunjukkan hubungan antar data. Seperti *data flow diagram*, ERD menggunakan simbol-simbol khusus untuk menggambarkan elemen-elemen ERD. Gambar 2.10 menunjukkan jenis-jenis simbol yang digunakan dalam ERD. ERD menggunakan simbol-simbol khusus tersebut dalam menggambarkan elemen-elemen ERD. Salah satu *symbol* yang umum digunakan adalah ERD simbol James Martin berikut merupakan contoh ERD menggunakan simbol James Martin yang ditunjukkan pada Gambar 2.11.

	IDEF1X	Chen	Information Engineering
An ENTITY: ✓ 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	ENTITY-NAME 	ENTITY-NAME 	ENTITY-NAME 
An ATTRIBUTE: ✓ 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	ENTITY-NAME 		ENTITY-NAME 
A RELATIONSHIP: ✓ 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	Relationship-name 		Relationship-name 

Gambar 2.10 Elemen-elemen dari ERD
 Sumber: Bressan dan Catania (2005)



Gambar 2.11 ERD James Martin
 Sumber: Kendall dan Kendall (2011)

Keterangan:

a. *Entity*

Entitas merupakan komponen ERD. Entitas bisa berupa orang, kejadian, atau benda dimana data akan dikumpulkan. Untuk menjadi sebuah *entity*, suatu objek harus menampilkan beberapa kali *event*. Contohnya jika sebuah firma hanya memiliki 1 gudang, maka gudang tersebut bukan entitas. Akan tetapi, jika perusahaan memiliki banyak gudang, maka gudang bisa menjadi entitas.

b. *Atribut*

Atribut adalah sebutan, sifat atau karakteristik yang dimiliki entitas yang dapat berupa parameter, variabel dan status.

- 1) Informasi yang diperoleh tentang sebuah entitas
- 2) Hanya yang digunakan oleh organisasi yang dimaukan dalam model
- 3) Nama atribut harus menggunakan kata benda
- 4) Nama entitas diletakkan di depan nama atribut untuk ketelitian.

c. *Identifier*

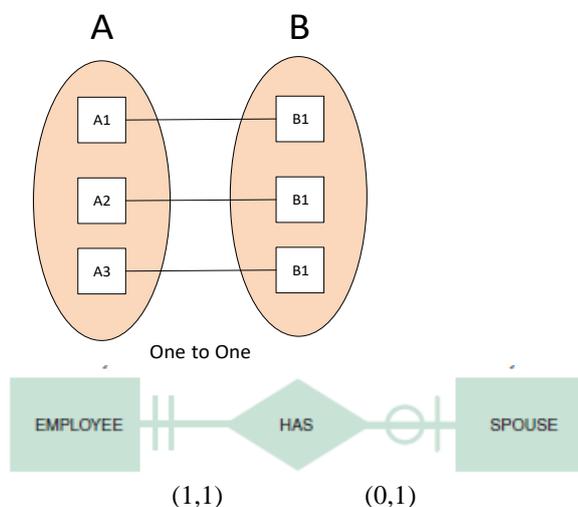
Satu atau lebih atribut dapat menjadi *identifier* entitas, yang secara unik mengidentifikasi setiap *instance* dari entitas. *Identifier* bisa saja artifisial, *identifier* dapat dilakukan dengan membuat *IDnumber*. *Identifier* tidak akan dikembangkan sampai fase desain. *Concatenated identifier* (*identifier* gabungan) terdiri dari beberapa atribut.

d. Relasi

Relasi adalah merupakan antar entitas. Entitas pertama disebut sebagai relasi induk dan entitas kedua disebut sebagai entitas anak. Relasi harus memiliki nama yang berupa kata kerja. Relasi berjalan 2 arah. Relasi dibagi menjadi dua yakni *connection* dan *join*. *Connection* adalah hubungan antar *class* setingkat. *Connection* terbagi menjadi berikut.

1) *One to one* (1:1)

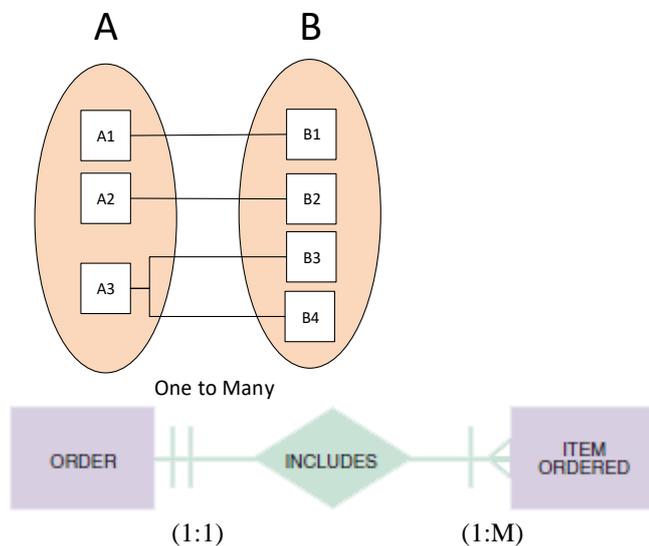
Satu *record* di *class* pertama berhubungan dengan satu *record* di *class* kedua, apabila ada hubungan *one to one* ($0,1 \rightarrow 0,1$). Setiap anggota entitas A berhubungan dengan minimal 0 anggota entitas B dan maksimal 1 anggota entitas B. Sebagai contoh, hubungan antara pekerja dan pasangannya. Seorang pekerja boleh memiliki seorang pasangan dan boleh tidak memiliki satu pasangan. Akan tetapi, 1 pasangan pasti memiliki satu pekerja. Simbol yang digunakan ditunjukkan pada Gambar 2.12.



Gambar 2.12 Simbol *one to one relationship*
 Sumber: Kendall dan Kendall (2011)

2) *One to many* (1:M)

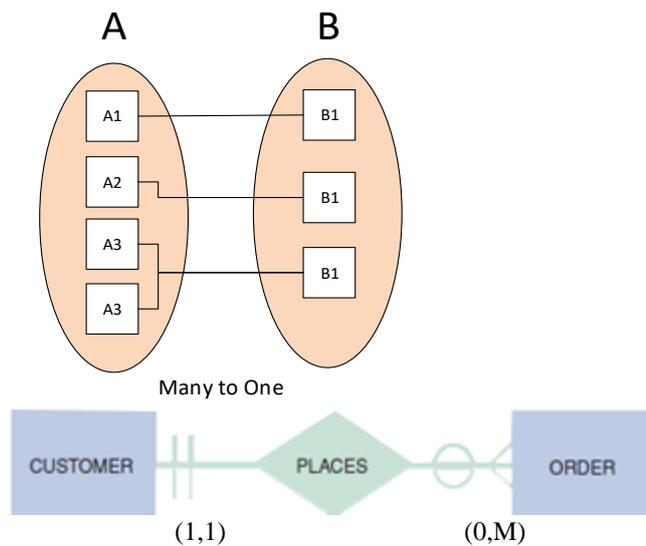
Satu *record* di *class* pertama berhubungan beberapa *record* di *class* kedua. Sebagai contoh dapat dilihat pada gambar 2.13. Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya. Setiap anggota entitas A berhubungan dengan minimal 1 anggota entitas B dan maksimal banyak anggota entitas B. Sebagai contoh, hubungan antara pesanan dengan item yang dipesan. 1 pesanan harus memiliki minimal 1 item yang akan dipesan dan bisa memiliki banyak item pesanan di dalamnya. Sedangkan 1 item pesanan, harus terdapat dalam minimal 1 pesanan, dan hanya bisa terdapat pada 1 pesanan saja agar tidak terjadi data yang *double* dengan item yang sama. Simbol yang digunakan pada Gambar 2.13 sebagai berikut.



Gambar 2.13 Simbol *one to many relationship*
 Sumber: Kendall dan Kendall (2011)

3) *Many to one* (M:1)

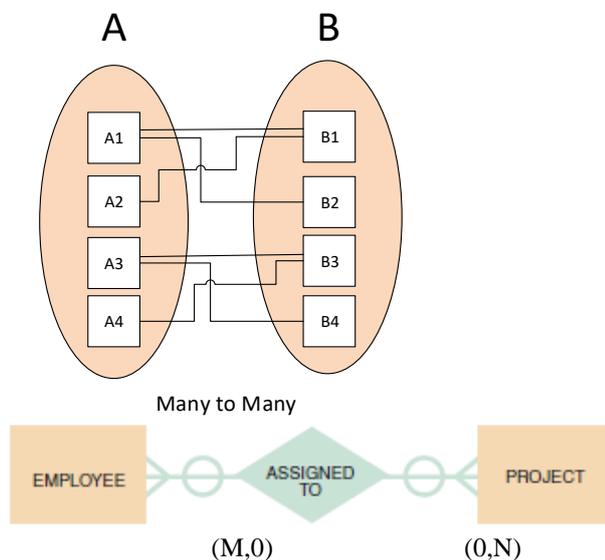
Beberapa *record* di *class* pertama berhubungan tepat satu *record* di *class* kedua. Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B tetapi tidak sebaliknya. Setiap anggota entitas A berhubungan dengan minimal 1 anggota entitas B dan maksimal 1 anggota entitas B. Sebagai contoh, hubungan antara pelanggan dan pesanan dalam sebuah toko. 1 pelanggan tidak harus melakukan pesanan di dalam toko tersebut, tetapi 1 orang pelanggan bisa memesan lebih dari 1 pesanan dalam sebuah toko. Sedangkan pesanan harus dimiliki minimal 1 pelanggan dan hanya boleh dimiliki oleh 1 pelanggan agar tidak terjadi *double data* pemesanan pada pelanggan yang sama. Simbol yang digunakan ditunjukkan pada Gambar 2.14.



Gambar 2.14 Simbol *many to one relationship*
 Sumber: Kendall dan Kendall (2011)

4) *Many to many* (M:N)

Beberapa *record* di *class* pertama berhubungan satu *record* di *class* kedua atau sebaliknya. Setiap entitas A dapat berhubungan dengan banyak entitas himpunan entitas B dan demikian pula sebaliknya. Setiap anggota entitas A berhubungan dengan minimal 0 anggota entitas B dan maksimal many anggota entitas B. Simbol yang digunakan dapat dilihat pada gambar 2.15.



Gambar 2.15 Simbol *many to many relationship*
 Sumber: Kendall dan Kendall (2011)

e. Kardinalitas

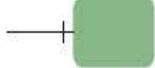
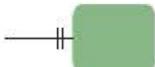
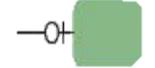
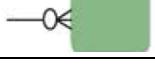
Kardinalitas mengacu pada jumlah *instance* dari satu entitas dapat berelasi dengan *instance* lainnya di entitas yang berbeda.

- 1) Satu *instance* dalam 1 entitas mengacu pada satu *instance* pada entitas lainnya (1:1)
- 2) Satu *instance* dalam suatu entitas mengacu ke satu atau lebih *instance* yang berelasi (1:N)
- 3) Satu atau lebih *instance* dalam suatu entitas mengacu pada satu atau lebih *instance* pada entitas yang berelasi (M:N)

Terdapat beberapa jenis kardinalitas yang dijelaskan pada Tabel 2.2 dibawah ini:

Tabel 2.2

Jenis-jenis Kardinalitas

Kardinalitas	Minimum	Maksimum	Notasi
<i>One and only one</i>	1	1	 Atau 
<i>Zero or one</i>	0	1	
<i>One or more</i>	1	<i>Many (>1)</i>	
<i>Zero, one, or more</i>	0	<i>Many (>1)</i>	

Sumber: Whitten dan Bentley (2007)

f. Modalitas

Mengacu pada apakah suatu *instance* dari entitas anak dapat ada tanpa suatu relasi dengan *instance* dari entitas induk atau tidak.

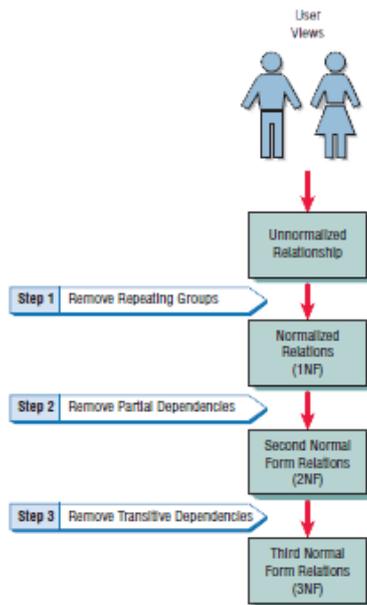
- 1) *Not Null* berarti bahwa suatu *instance* pada suatu entitas yang berelasi harus ada untuk suatu *instance* dari entitas lain untuk disebut valid
- 2) *Null* berarti bahwa tidak ada *instance* dalam entitas yang berelasi yang diperlukan untuk *instance* pada relasi lain untuk dikatakan valid

Terdapat beberapa pedoman yang perlu diperhatikan untuk membuat ERD diantaranya:

- a. Entitas harus memiliki banyak kejadian/realitas
- b. Hindari penggunaan atribut yang tidak digunakan
- c. Berilah label yang jelas dan tepat untuk semua komponen
- d. Pasangkan kardinalitas dan modalitas yang jelas dan benar
- e. Pecah atribut menjadi level serendah mungkin yang diperlukan
- f. Label harus merefleksikan istilah-istilah bisnis yang umum
- g. Asumsi harus disebutkan dengan jelas

3. Normalisasi

Normalisasi merupakan teknik yang digunakan untuk memvalidasi model data. Serangkaian aturan pada data model logika untuk meningkatkan pengaturannya. Teknik analisa data yang mengorganisasi data dalam *class* yang mengelompokkan entitas agar menjadi nonredundan, *stable*, *flexible* dan *adaptive*, sehingga menjaga akurasi, integritas dan validitas data serta ketersediaan data baik dalam *class* tunggal maupun hubungan antar *class* melalui *query* (Whitten, 2007). Gambar 2.16 merupakan langkah-langkah yang digunakan untuk melakukan normalisasi terhadap data model yang telah diperoleh:



Gambar 2.16 Langkah normalisasi
 Sumber: Whitten dan Bentley (2007)

a. First Normal Form (1NF)

Pada 1NF, cari kelompok-kelompok entitas yang berulang dan pisahkan ke dalam entitas yang berbeda.

ORDER IN 1NF	combination primary key				
RECORD#	ORDER- NUM	ORDER- DATE	PRODUCT- NUM	PRODUCT- DESC	NUM- ORDERED
1	40311	03/11/2011	304	All-purpose gadget	7
2	40311	03/11/2011	633	Assembly	1
3	40311	03/11/2011	684	Super gizmo	4
4	40312	03/11/2011	128	Steel widget	12
5	40312	03/11/2011	304	All-purpose gadget	3
6	40313	03/12/2011	304	All-purpose gadget	144

repeating groups have been eliminated

Gambar 2.17 Contoh 1NF
 Sumber: Cashman (2012)

b. Second Normal Form (2NF)

Jika ada entitas yang memiliki identifier gabungan, maka pada 2NF cari atribut yang hanya bergantung pada *identifier*. Jika ditemukan pindahkan ke entitas baru.

ORDER IN 2NF (primary key)

RECORD#	ORDER- NUM	ORDER- DATE
1	40311	03112011
2	40312	03112011
3	40313	03122011

PRODUCT IN 2NF (primary key)

RECORD#	PRODUCT- NUM	PRODUCT- DESC
1	128	Steel widget
2	304	All-purpose gadget
3	633	Assembly
4	684	Super gizmo

ORDER-LINE IN 2NF (primary key based on combination of two fields)

RECORD#	ORDER- NUM	PRODUCT- NUM	NUM- ORDERED
1	40311	304	7
2	40311	633	1
3	40311	684	4
4	40312	128	12
5	40312	304	3
6	40313	304	144

Gambar 2.18 Contoh 2NF
Sumber: Cashman (2012)

c. *Third Normal Form (3NF)*

Pada 3NF, cari atribut yang bergantung hanya pada atribut lain yang bukan merupakan *identifier*. Jika ditemukan pindahkan menjadi entitas baru, juga pindahkan atribut-atribut yang dirasa perlu dipindahkan.

CUSTOMER IN 3NF

RECORD#	CUSTOMER- NUM	CUSTOMER- NAME	ADDRESS	SALES-REP- NUM
1	108	Benedict, Louise	San Diego, CA	41
2	233	Corelli, Helen	Nashua, NH	22
3	254	Gomez, J.P.	Butte, MT	38
4	431	Lee, M.	Snow Camp, NC	74
5	779	Paulski, Diane	Lead, SD	38
6	800	Zuider, Z.	Greer, SC	74

SALES-REP IN 3NF (in 3NF, no nonkey field is dependent on another nonkey field)

RECORD#	SALES-REP- NUM	SALES-REP- NAME
1	22	McBride, Jon
2	38	Stein, Ellen
3	41	Kaplan, James
4	74	Roman, Harold

Gambar 2.19 Contoh 3NF
Sumber: Cashman (2012)

2.12 Sistem Desain

Sistem desain adalah sebuah fase yang berhubungan dengan desain fisik yang sesuai dengan spesifikasi yang telah digambarkan pada dokumen kebutuhan sistem. *User interface*, desain data, dan arsitektur sistem termasuk dalam fase ini. *Output* fase ini adalah spesifikasi desain sistem (Cashman, 2012).

Fase desain adalah fase yang menjelaskan mengenai pembangunan sistem. Pada tahapan ini dilakukan desain untuk mendefinisikan detail dari deskripsi teknis dari sistem. Hasil dari fase desain adalah spesifikasi sistem. Spesifikasi sistem adalah produk akhir dari fase desain, yang menjelaskan dengan tepat perancangan sistem seperti apa yang akan diimplementasikan oleh tim desain selama fase implementasi.

Adapun langkah-langkah fase desain dapat dijelaskan:

1. Menyajikan alternatif desain (membuat sendiri, membeli atau *outsourcing*)
2. Mengubah proses-proses logik dan model data ke dalam model fisik
3. Merancang arsitektur sistem
4. Membuat pemilihan *hardware* dan *software*
5. Merancang bagaimana data disimpan
6. Mendesain program untuk proses yang bersesuaian
7. Membuat spesifikasi sistem

2.12.1 Desain Arsitektur

Desain arsitektur adalah perencanaan bagaimana sistem didistribusikan di antara komputer-komputer yang ada dan *software* dan *hardware* apa yang akan digunakan untuk masing-masing komputer. Desain arsitektur terdapat dua hal yang perlu diperhatikan:

1. Spesifikasi dari *software* dan *hardware*
2. Deskripsi detail dari komponen *software/hardware* untuk mempermudah pihak yang akan membeli *software/hardware* tersebut.

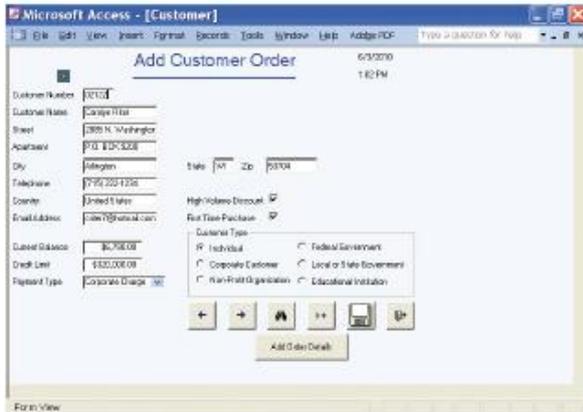
Elemen dari desain arsitektur jika dilihat dari sudut pandang *software* komponen arsitektural dapat dibagi menjadi:

1. *Data storage*: komponen yang digunakan untuk menyimpan data
2. *Data access logic*: prosedur yang digunakan untuk mengakses data yang disimpan dalam *data storage*
3. *Application logic*: bagian logika pemrosesan dalam lapis aplikasi
4. *Presentation logic*: komponen pemrosesan tampilan dan perintah dari *user*

2.12.2 User Interface

User Interface menggambarkan bagaimana *user* berinteraksi dengan sistem komputer, yang termasuk semua *hardware*, *software*, layar, *output*, fungsi, dan fitur yang mengakibatkan komunikasi dua arah diantar *user* dengan komputer.

Tujuan dari *User Interface* adalah untuk memungkinkan *user* menjalankan setiap tugas dalam *user requirement*. Jadi dalam membangun sebuah *user interface* harus berdasar pada *user requirement*. Contoh *user interface* dapat dilihat pada Gambar 2.20.



Gambar 2.20 Contoh *user interface*
Sumber: Kendall dan Kendall (2011)

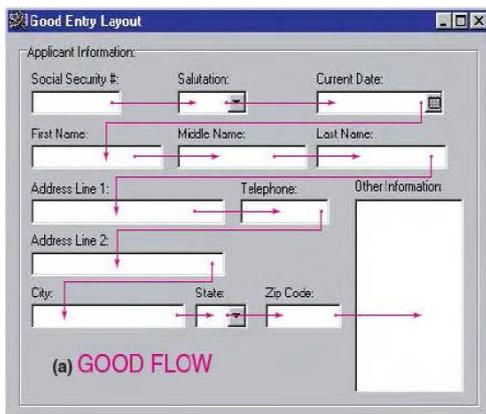
Prinsip-prinsip *user interface* yang baik adalah:

1. UI yang baik tidak mengharuskan pengguna untuk mengingat tampilan UI
2. UI menampilkan apa yang dimengerti oleh *user* atau visualisasi dari keadaan dari sistem sekarang

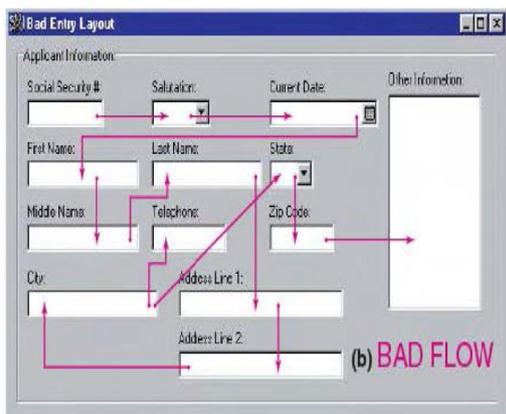
Hal yang harus dihindari dalam perancangan *user interface* adalah:

1. Menampilkan terlalu banyak informasi dan terlalu banyak pilihan.
2. Menampilkan terlalu sedikit informasi, terlalu sedikit pilihan dan tanpa konteks
3. Eksploitasi struktur menu standar yang sudah familiar dengan *software* yang sering digunakan *user*

Selain itu *user interface* memperhatikan aliran data untuk masukannya seperti pada Gambar 2.21 Aliran data yang baik berurutan agar memudahkan *user* dalam proses memasukkan data. Sedangkan aliran data yang buruk dapat dilihat pada Gambar 2.22.



Gambar 2.21 Aliran data baik
Sumber: Whitten dan Bentley (2007)



Gambar 2.22 Aliran data buruk
Sumber: Whitten dan Bentley (2007)

2.13 Cloud computing

Cloud computing (komputasi awan) adalah model yang memungkinkan untuk *ubiquitous* (dimanapun dan kapanpun), *on-demand*, nyaman, dan akses jaringan ke sumber daya komputasi (contoh: jaringan server, *storage*, aplikasi, dan layanan) yang dapat dengan cepat dirilis atau ditambahkan. *Cloud computing* juga merupakan teknologi yang sangat diperlukan di masa depan.

Menurut Hernita (2011) *Cloud computing* sebenarnya memiliki konsep yang cukup sederhana namun punya banyak variasi dan cabang. Untuk mengenal konsep *cloud computing*, maka harus diketahui beberapa kaidah dasar seperti mengenal cara kerja *cloud computing*. Selain itu, bagaimana tipe pengguna yang optimal untuk *cloud computing*. *Cloud computing* memiliki beberapa fasilitas seperti *upload* dokumen, *download* dokumen, *printing*, *sharing and privacy*, dan dapat diakses dimana saja.

2.13.1 Karakteristik *Cloud computing*

Menurut Hernita (2011), *Cloud computing* memiliki beberapa karakteristik, yaitu:

1. *On-Demand Self Services*: sebuah layanan *Cloud computing* harus dapat dimanfaatkan oleh pengguna melalui mekanisme dan secara langsung tersedia pada saat dibutuhkan.
2. *Broad Network Access*: sebuah layanan *Cloud computing* harus dapat diakses kapan saja, dari mana saja, dengan alat apapun, asalkan *user* terhubung ke jaringan layanan/ jaringan internet.
3. *Resources Pooling*: sebuah layanan dimana *Cloud computing* harus tersedia secara pusat dan dapat membagi sumber daya secara efisien.
4. *Rapid Elasticity* : sebuah layanan dimana *Cloud computing* harus dapat menaikkan atau menurunkan kapasitas sesuai kebutuhan

5. *Measured Service*: sebuah layanan dimana *Cloud computing* harus disediakan secara terukur, karena akan digunakan dalam proses pembayaran.

2.13.2 Manfaat *Cloud computing*

Menurut Hernita (2011), berikut ini adalah manfaat dari penggunaan *cloud computing*.

1. Skalabilitas, yaitu dengan *cloud computing* dapat menambah kapasitas penyimpanan data tanpa harus membeli peralatan tambahan.
2. Aksesibilitas, yaitu dapat mengakses data kapanpun dan dimanapun berada, asal terkoneksi dengan internet.
3. Keamanan, yaitu data dapat terjamin keamanannya oleh penyedia layanan, sehingga dapat disimpan secara aman.
4. Kreasi, yaitu para *user* bias melakukan/mengembangkan kreasi atau *project* tanpa harus mengirimkan *project* secara langsung.
5. Kecemasan, ketika terjadi bencana alam data akan tersimpan aman di *cloud*.

2.13.3 Model Layanan *Cloud computing*

Menurut Mell dan Grance (2012), terdapat 3 macam model layanan *cloud computing* diantaranya:

1. *Cloud Software as a Service (SaaS)*, merupakan kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia dapat beroperasi pada infrastruktur *cloud*. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka seperti *web browser* (misalnya, email berbasis web). Contohnya adalah *Google Apps*, *SalesForce.com* dan aplikasi jejaring sosial seperti *Facebook*.
2. *Cloud Platform as a Service (PaaS)*, merupakan kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur *cloud computing* menggunakan bahasa pemrograman dan peralatan yang didukung oleh *provider*. Konsumen tidak mengelola atau mengendalikan infrastruktur *cloud* yang mendasar termasuk jaringan, server, sistem operasi, atau penyimpanan, namun memiliki kontrol atas aplikasi yang disebarkan dan memungkinkan aplikasi melakukan *hosting* konfigurasi. Contohnya yang sudah mengimplementasikan ini adalah *Force.com* dan *Microsoft Azure Investment*.
3. *Cloud Infrastructure as a Service (IaaS)*, merupakan kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, berjejaring, dan sumber komputasi penting yang lain, dimana konsumen dapat menyebarkan dan menjalankan perangkat

lunak secara bebas, yang dapat mencakup sistem operasi aplikasi. Konsumen tidak mengelola atau mengendalikan infrastruktur *cloud* yang mendasar tetapi memiliki kontrol atas sistem operasi, penyimpanan, aplikasi yang disebarkan, dan mungkin kontrol terbatas komponen jaringan yang pilih (misalnya, *firewall host*). Contohnya seperti *Amazon Elastic Compute Cloud* dan *Simple Storage Service*.

2.13.4 Model Penyebaran *Cloud computing*

Menurut Mell dan Grance (2012), terdapat 4 macam model penyebaran *cloud computing* diantaranya:

1. *Private cloud*, yaitu infrastruktur *cloud* yang semata-mata dioperasikan bagi suatu organisasi. Ini mungkin dimiliki, dikelola dan dijalankan oleh suatu organisasi, pihak ketiga atau kombinasi dari beberapa pihak dan mungkin ada pada *on premis* atau *off premis*.
2. *Community cloud*, infrastruktur *cloud* digunakan secara bersama oleh beberapa organisasi dan mendukung komunitas tertentu yang telah berbagi concerns (misalnya; misi, persyaratan keamanan, kebijakan, dan pertimbangan kepatuhan). Ini mungkin dikelola oleh organisasi atau pihak ketiga dan mungkin ada pada *on premis* atau *off premis*.
3. *Public cloud*, infrastruktur *cloud* yang disediakan untuk umum atau kelompok industri besar dan dimiliki oleh sebuah organisasi yang menjual layanan *cloud*.
4. *Hybrid cloud*, infrastruktur *cloud* merupakan komposisi dari dua atau lebih *cloud* (swasta, komunitas, atau publik) yang masih entitas unik namun terikat bersama oleh standar atau kepemilikan teknologi yang menggunakan data dan portabilitas aplikasi (e.g., *cloud bursting for load-balancing between clouds*).

2.14 PHP

PHP merupakan singkatan dari PHP singkatan dari *Perl Hypertext Preprocessor*. Bahasa pemrograman ini pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada versi pertamanya, PHP terdiri dari sekumpulan *Script PERL* yang digunakan untuk mengolah data *form* dari *website*. Awalnya PHP bernama FI (*Form Interpreted*), kemudian oleh Rasmus dilepaskan kode sumbernya. Sehingga terbentuklah nama PHP/FI (*Personal Home Page/Form Interpreter*). Sejak saat itulah PHP bersifat *open source* dan dapat digunakan oleh banyak orang hingga kini (Sukarno, 2006).

Menurut Madcoms (2011), bahasa pemrograman PHP merupakan bahasa pemrograman yang beroperasi dalam *web server*. *Script-script* PHP yang telah dibuat harus tersimpan pada sebuah *server* yang selanjutnya dieksekusi atau diproses dalam *server* tersebut. Penggunaan bahasa pemrograman PHP memungkinkan *website* untuk menjadi lebih dinamis dan interaktif. Proses dari bahasa pemrograman PHP yakni data yang dikirim oleh pengunjung *website* atau *computer client* selanjutnya diolah dan disimpan dalam *database website server*. Data tersebut dapat ditampilkan kembali apabila diperlukan. Selain dengan instalasi *server* pada komputer, cara lain untuk menggunakan bahasa pemrograman PHP yakni dengan mendaftarkan *hosting* pada *website* yang melayani jasa *hosting*. Berikut merupakan beberapa keunggulan dari bahasa pemrograman PHP.

1. PHP bersifat *free* atau gratis
2. Beberapa *server* seperti Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhhttpd, dan xitami mampu menjalankan PHP
3. Tingkat akses PHP lebih cepat serta memiliki tingkat keamanan yang tinggi

Beberapa *database* yang sudah ada, baik bersifat *free* atau gratis ataupun komersial sangat mendukung akses PHP, diantaranya MySQL, PostgreSQL, mSQL, Informix, dan Microsoft SQL server.

2.15 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standar *Structured Query Language* (SQL). MySQL merupakan sebuah *database server* yang *free*, artinya *database* ini dapat digunakan untuk keperluan pribadi atau perusahaan tanpa harus membeli atau membayar lisensinya. *Database* MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database* relasional atau dalam bahasa basis data disebut *Relation Database Management System* (RDBMS) yang menggunakan bahasa perintah bernama SQL (Bunafit, 2006).

2.16 Website

Menurut Astropudin (2013) web adalah sebuah kumpulan halaman yang diawali dengan halaman muka yang berisi informasi, iklan, serta program aplikasi. Menurut Andrea (2012) web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan *surfer* dalam melakukan *browsing* informasi melalui internet. Dari pengertian diatas dapat disimpulkan bahwa web adalah suatu layanan atau kumpulan halaman yang

berisi informasi, iklan serta program aplikasi yang dapat digunakan oleh *surfer*. Terdapat beberapa manfaat *website* diantaranya:

1. Memperluas jaringan promosi sehingga produk kita dapat diketahui oleh masyarakat
2. Media melakukan kegiatan bisnis
3. Internet dapat diakses diseluruh lapisan masyarakat
4. Media sharing informasi

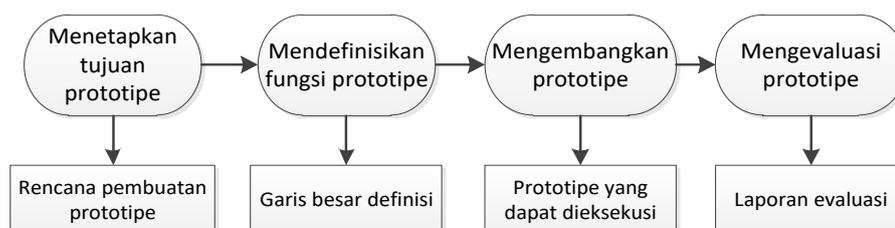
2.17 Prototyping

Tahap rancangan sistem dilakukan dengan analisis untuk mengetahui segala sesuatu yang dibutuhkan oleh sistem. *Cloud computing* yang dibuat sebagai saran perbaikan yang ditawarkan dalam perancangan sitem baru. Rancangan sistem dilakukan dengan menggunakan tahapan pembuatan *prototype*. *Prototyping* merupakan pendekatan terstruktur dalam rekayasa perangkat lunak yang bertujuan memfasilitasi produksi perangkat lunak kualitas tinggi dengan cara efektif dalam hal biaya. *Prototyping* secara langsung mendemonstrasikan sebuah perangkat lunak atau komponen-komponen perangkat lunak bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan (Sommerville, 2003).

Perangkat lunak merupakan sintesis unsur-unsur rancangan yang dipadukan dengan metode ilmiah agar menjadi suatu model yang memenuhi spesifikasi tertentu (SRC). Langkah studi kelayakan yang panjang lebar dilompati dulu (*Feasibility study: economics, technics, time, resources, etc.*). Langkah-langkah pengembangan *prototype* yaitu:

1. Menetapkan tujuan *prototype*

Tujuan dari pembuatan *prototype* harus dibuat secara jelas dari awal proses. Dalam menetapkan tujuan dari *prototype* yaitu memecahkan masalah yang dihadapi, digunakan analisis *Performance, Information, Economy, Control, Eficiency* dan *Services* (PIECES). Analisis ini digunakan untuk mengidentifikasi masalah-masalah yang ada pada sistem yang sedang berjalan.



Gambar 2.23 Model proses pengembangan *prototype*

Sumber: Sommerville (2003)

2. Mendefinisikan fungsi *prototype*

Pengembang *prototype* harus membuat daftar kebutuhan sistem (*system requirements checklist*) untuk mengidentifikasi dan menggambarkan seluruh kebutuhan sistem. SRC merupakan salah satu barometer ukuran kesuksesan suatu *prototype* sedangkan kebutuhan sistem adalah fitur yang disertakan ke dalam sistem informasi untuk memenuhi kebutuhan bisnis dan dapat diterima oleh pengguna. Kebutuhan sistem dapat digambarkan ke dalam lima kategori umum: *output*, *input*, kinerja, proses, dan kontrol.

3. Mengembangkan *prototype*

Output dari tahap ini adalah *prototype* yang dapat dieksekusi. Untuk menunjang hal tersebut, pengembang *prototype* membuat rancangan terlebih dahulu. Rancangan yang dimaksud adalah desain basis data (desain logis dan desain fisik), desain antarmuka pengguna, dan desain proses.

- a. Desain basis data bertujuan untuk merancang basis data yang akan digunakan pada aplikasi ini. Desain basis data terdiri dari desain basis data logis dan basis data fisik.
 - 1) Desain logis merupakan desain yang masih berupa konsep. Konsep yang telah dibuat akan menjadi dasar pembuatan desain fisik. Desain logis terdiri dari tiga bagian, yaitu desain *database* dengan menggunakan ERD dan DFD (*data modeling dan process modeling*)
 - 2) Relasi dan Normalisasi, membawa *entity* ke bentuk relasi, untuk dinormalisasi. Normalisasi dilakukan minimal sampai dengan bentuk normal ke-3
 - 3) Desain fisik merupakan aktualisasi dari desain logis. Desain fisik dibagi menjadi dua bagian, yaitu desain *database* fisik dan desain aplikasi
- b. Desain antarmuka pengguna digunakan untuk merancang antarmuka yang efektif untuk sistem perangkat lunak. Efektif artinya sistem siap digunakan, dan hasilnya sesuai dengan kebutuhan pengguna.
- c. Desain proses/algoritma dilakukan untuk merancang tahap proses yang harus dilakukan hingga *output* yang diharapkan dari aplikasi dapat ditampilkan. Algoritma merupakan nyawa aplikasi/*software* dan dapat dinyatakan dengan *flowchart* ataupun *pseudocode*.
- d. Tahap Pengembangan/Pembuatan *Prototype*

Pertama yang dilakukan adalah memilih alat pengembang (misal My SQL dengan PHP; atau Excel dengan VBA), lalu dilaksanakan pembuatan *prototype* sesuai petunjuk alat pengembang.

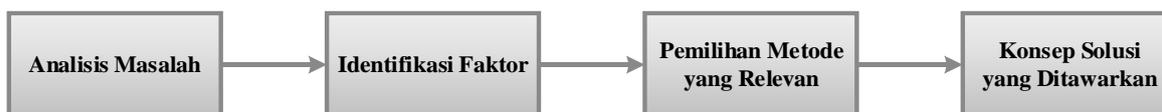
4. Mengevaluasi *prototype*

Pada tahap ini dilakukan pengujian terhadap aplikasi yang telah dibuat. Pengujian aplikasi ini ditinjau dari tiga segi yaitu: verifikasi, validasi, dan *prototype*.

- a. Verifikasi merupakan evaluasi dengan mengacu pada pertanyaan apakah program yang dibuat telah sesuai dengan hasil rancangannya
- b. Validasi merupakan evaluasi yang dilakukan dengan mengacu pada pertanyaan apakah program yang dibuat telah sesuai dengan fungsinya
- c. *Prototype*, menguji *prototype* yang telah dibuat untuk mengetahui apakah *prototype* dapat menjawab kelemahan yang didapat dari hasil analisis PIECES

2.18 Kerangka Pikiran

Kerangka pikiran digunakan sebagai narasi atau pernyataan tentang kerangka konsep pemecahan masalah yang telah diidentifikasi atau dirumuskan. Berikut merupakan tahap yang dilakukan dan sebagai kerangka pikiran dari penelitian ini dapat dijelaskan pada Gambar 2.24.



Gambar 2.24 Tahapan kerangka pikiran

1. Analisis Masalah

Tahap analisis masalah sebagai teknik pengambilan data untuk menganalisis masalah yang terjadi di perusahaan dapat dilakukan dengan dua cara yaitu melakukan wawancara dengan pihak terkait secara langsung dan data sekunder yang berasal dari data penelitian sebelumnya. Hasil identifikasi di lapangan didapatkan permasalahan yaitu sistem informasi manajemen persediaan *retailer* PT IGM yang kurang baik sehingga perlu dilakukan perbaikan. Selain itu tidak efisiennya alur proses pemesanan oleh *retailer* menyebabkan sering terjadinya kesalahan pemesanan dan membutuhkan waktu yang cukup lama.

2. Identifikasi Faktor Pendukung

Tahap identifikasi faktor pendukung sistem informasi dapat dilakukan setelah dilakukan identifikasi masalah pada PT IGM Malang.

Tabel 2.3

Faktor Pendukung Sistem Informasi

No	Faktor	Jenis Data	Cara Mendapatkan
1	Profil Perusahaan PT. IGM Malang	Primer	Pengambilan data dilakukan dengan melakukan wawancara dan dokumentasi secara langsung kepada manager PT IGM Malang

No	Faktor	Jenis Data	Cara Mendapatkan
2	Alur Proses Pemesanan <i>Retailer</i>	Primer	Pengambilan data dilakukan dengan melakukan wawancara dan dokumentasi secara langsung kepada supervisor sales dan salesman PT IGM Malang
3	Proses Pengambilan Barang Pesanan	Primer	Pengambilan data dilakukan dengan melakukan wawancara dan dokumentasi secara langsung kepada supervisor sales PT IGM Malang
4	Proses Pengiriman kepada <i>Retailer</i>	Primer	Pengambilan data dilakukan dengan melakukan wawancara dan dokumentasi secara langsung kepada fakturis dan petugas pengiriman PT IGM Malang
5	Proses Pembayaran Tagihan	Primer	Pengambilan data dilakukan dengan melakukan wawancara dan dokumentasi secara langsung kepada salesman dan inkaso PT IGM Malang
6	Jenis Produk	Sekunder	Data jenis produk didapatkan dari sumber dokumentasi berupa laporan tahunan/bulanan PT. IGM Malang
7	Atribut	Primer	Atribut didapatkan dengan melakukan wawancara langsung dengan supervisor dan pelaksana gudang serta supervisor sales, pengiriman dan salesman.
		Sekunder	Atribut didapatkan dari dokumentasi laporan tahunan/bulanan PT. IGM Malang
8	<i>Retailer</i>	Primer	Data <i>retailer</i> didapatkan dengan melakukan wawancara langsung dengan supervisor sales, pengiriman dan salesman serta sampel apotek disekitar kota malang
		Sekunder	Data <i>retailer</i> didapatkan dari dokumentasi laporan tahunan/bulanan PT. IGM Malang

3. Pemilihan Metode yang Relevan

Tahap pemilihan metode yang relevan terhadap permasalahan yang telah diidentifikasi sebelumnya dapat dilakukan setelah mengetahui faktor-faktor apa saja yang mendukung keberhasilan penelitian ini. Berikut merupakan metode yang relevan digunakan pada penelitian ini.

Tabel 2.4

Metode yang Relevan

No	Metode	Kegunaan
1	Teknologi Informasi dalam <i>Supply chain Management</i>	Teknologi informasi dalam <i>Supply chain Management</i> merupakan metode untuk menyelesaikan permasalahan kegiatan koordinasi dan kolaborasi yang terintegrasi antara <i>retailer</i> dengan PT. IGM Malang
2	<i>Vendor managed Inventory</i> (VMI)	VMI merupakan metode yang digunakan untuk menyelesaikan permasalahan pengendalian dari setiap <i>retailer</i> dari PT. IGM Malang untuk kemudian dapat diawasi oleh PT IGM Malang sehingga dapat lebih mudah dalam kegiatan pengiriman dan perencanaan <i>inventory</i> pada gudang PT IGM Malang
3	<i>Database</i>	<i>Database</i> digunakan sebagai tempat penyimpanan segala informasi yang ada pada sistem informasi yang dirancang.
4	<i>Cloud computing</i>	<i>Cloud computing</i> merupakan metode untuk menghubungkan seluruh aspek yang berkaitan dengan menggunakan teknologi diatas awan. Dengan <i>Cloud computing</i> , pihak-pihak yang berkaitan dapat dengan mudah mengakses segala kegiatan yang berkaitan dengan kegiatan pemesanan yang dilakukan <i>retailer</i> dengan PT IGM Malang
5	PHP	Alat yang digunakan dalam menyelesaikan permasalahan manajemen industri farmasi pada PT IGM Malang. PHP dapat menampilkan

No	Metode	Kegunaan
		informasi yang memudahkan proses bisnis antara PT IGM Malang sebagai <i>supplier</i> dengan retailrnya dimana PHP ini akan terintegrasi dengan <i>database</i> MySQL
6	MySQL	Perancangan sistem <i>input / output</i> dalam proses bisnis baik mengenai informasi pemesanan maupun persediaan outlet dan gudang disimpan dalam <i>database</i> server sehingga dapat diakses pada <i>website</i>
7	<i>Website</i>	<i>Website</i> merupakan suatu alat yang digunakan untuk menyelesaikan kemudahan akses dari <i>retailer</i> PT IGM Malang

4. Konsep Solusi yang Ditawarkan

Konsep solusi yang ditawarkan berkaitan dengan perancangan sistem informasi manajemen pemesanan yang terintegrasi antara *retailer* dengan PT IGM Malang. Sistem informasi ini akan menyelesaikan permasalahan yang telah diidentifikasi sebelumnya yaitu untuk memperbaiki permasalahan manajemen persediaan pada *retailer* dari PT IGM Malang serta perancangan sistem informasi pemesanan menggunakan bahasa pemrograman PHP *database* MySQL. Perancangan sistem informasi dibuat sebatas *prototype*.

Halaman ini sengaja dikosongkan