

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Kerangka Teori**

##### **2.1.1 Teori Umum**

###### **2.1.1.1 Pengertian Manajemen**

Menurut Robbins dan Coulter (2007, p8) pengertian manajemen adalah proses pengkoordinasian kegiatan-kegiatan pekerjaan sehingga pekerjaan tersebut terselesaikan secara efisien (melakukan segala sesuatu secara benar) dan efektif (melakukan segala sesuatu yang benar) dengan dan melalui orang lain. Selain itu, menurut Robbins dan Coulter (2007, pp9-10) fungsi manajemen, yakni:

- a. Merencanakan (*planning*) → mendefinisikan sasaran, menetapkan strategi, dan menyusun bagian-bagian rencana untuk mengkoordinasikan sejumlah kegiatan.
- b. Mengorganisasi (*organizing*) → menentukan apa yang perlu dilakukan, bagaimana cara melakukan, dan siapa yang harus melakukannya.
- c. Memimpin (*leading*) → mengarahkan dan memotivasi seluruh pihak yang terlibat dan menyelesaikan konflik.
- d. Mengendalikan (*controlling*) → memantau kegiatan guna meyakinkan bahwa kegiatan tersebut diselesaikan seperti yang direncanakan.

Jadi dapat disimpulkan bahwa manajemen adalah sebuah ilmu yang mempelajari bagaimana mengkoordinasikan dan mengelola sumber daya yang ada dalam perusahaan untuk melakukan kegiatan-kegiatan secara efisien dan efektif untuk mencapai tujuan perusahaan.

## 2.1.1.2 Sistem Informasi

### 2.1.1.2.1 Pengertian Sistem

Menurut O'Brien yang diterjemahkan oleh Fitriyani dan Kwary (2006, p29), sistem adalah sekelompok elemen yang saling berhubungan, bekerja bersama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam proses transformasi yang teratur. Selain itu, menurut O'Brien (2006, p32), sistem semacam ini (kadang disebut sebagai sistem dinamis) memiliki tiga komponen atau fungsi dasar yang berinteraksi, yakni:

- *Input* → penangkapan dan perakitan berbagai elemen yang memasuki sistem untuk diproses.
- Pemrosesan → proses transformasi yang mengubah *input* menjadi *output*.
- *Output* → perpindahan elemen yang telah diproduksi oleh proses transformasi ke tujuan akhirnya.

Konsep sistem akan makin berguna dengan memasukkan dua komponen tambahan, yakni :

- Umpan balik → data mengenai kinerja sistem.
- Pengendalian → pengawasan dan pengevaluasian umpan balik untuk menetapkan apakah sistem bergerak menuju pencapaian tujuan atau tidak. Fungsi pengendalian kemudian akan membuat penyesuaian yang dibutuhkan atas komponen *input* pemrosesan sistem, untuk memastikan bahwa sistem tersebut menghasilkan *output* yang sesuai.

Sistem yang memiliki komponen ini kadang disebut sebagai sistem *cybernetic*, yaitu sistem yang mengawasi dan mengatur dirinya sendiri.

Jadi dapat disimpulkan bahwa sistem adalah sekumpulan elemen-elemen yang saling berhubungan dan berkerja secara bersama-sama untuk mencapai tujuan bersama pula yang terdiri dari tiga komponen utama yakni: *input*, pemrosesan dan *output* serta ditambahkan dua komponen tambahan yakni: umpan balik dan pengendalian.

#### **2.1.1.2.2 Pengertian Informasi**

Menurut O'Brien yang diterjemahkan oleh Fitriyani dan Kwary (2006, p38), informasi adalah data yang telah diubah menjadi konteks yang berarti dan berguna bagi para pemakai akhir tertentu. Pemrosesan informasi meliputi: bentuk yang agregat, telah dimanipulasi, atau diatur, isinya dianalisis dan dievaluasi, dan ditempatkan dalam konteks yang tepat untuk pemakainya. Selain itu, menurut O'Brien (2006, p439), ringkasan atribut dan kualitas informasi, yakni:

##### **a. Dimensi Waktu**

- Ketepatan waktu → informasi harus tersedia ketika dibutuhkan.
- Kekinian → informasi harus selalu baru ketika disediakan.
- Frekuensi → informasi harus tersedia sesering yang dibutuhkan.
- Periode Waktu → Informasi harus tersedia untuk periode waktu lampau, sekarang dan masa depan.

##### **b. Dimensi Isi**

- Keakuratan → informasi harus bebas dari kesalahan.
- Relevansi → informasi harus berhubungan dengan kebutuhan informasi dari penerima tertentu untuk situasi tertentu.
- Kelengkapan → semua informasi yang dibutuhkan harus tersedia.

- Keringkasan → hanya informasi yang dibutuhkan yang disediakan.
- Cakupan → informasi dapat memiliki cakupan yang sempit dan luas, atau untuk fokus internal dan eksternal.
- Kinerja → informasi dapat menunjukkan kinerja dengan mengukur aktivitas yang diselesaikan, kemajuan yang dicapai, atau sumber daya yang diakumulasi.

c. Dimensi Bentuk

- Kejelasan → informasi harus tersedia dalam bentuk yang mudah dipahami.
- Rinci → informasi dapat disediakan dalam bentuk rinci dan ringkasan.
- Urutan → informasi dapat disusun dalam urutan yang telah ditentukan.
- Presentasi → informasi dapat disajikan dalam bentuk narasi, numerik, grafik, atau bentuk lainnya.
- Media → informasi dapat disediakan dalam bentuk dokumen tercetak, tampilan video, atau media lainnya.

Jadi dapat disimpulkan bahwa informasi adalah hasil dari pengolahan data menjadi konteks yang berarti dan berguna bagi pengguna informasi di mana informasi yang disediakan harus memenuhi tiga dimensi kualitas informasi yakni: dimensi waktu, dimensi isi dan dimensi bentuk.

### 2.1.1.2.3 Pengertian Sistem Informasi

Menurut O'Brien yang diterjemahkan oleh Fitriyani dan Kwary (2006, p703), sistem informasi adalah rangkaian orang, prosedur, dan sumber daya yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi. Atau sebuah sistem yang menerima sumber daya data sebagai *input* dan memprosesnya ke

dalam produk informasi sebagai *output*-nya. Selain itu, menurut O'Brien (2006, p34) komponen Sistem Informasi ada lima, yakni:

- a. Sumber daya data → dasar data dan pengetahuan.
- b. Sumber daya manusia → pemakai akhir dan ahli SI.
- c. Sumber daya *software* → program dan prosedur.
- d. Sumber daya *hardware* → mesin dan media.
- e. Sumber daya jaringan → media komunikasi dan dukungan jaringan.

Menurut Whitten, Bentley dan Ditman (2004, p10), sistem informasi adalah pengaturan orang, data, proses dan informasi (TI) atau teknologi informasi yang berinteraksi untuk mengumpulkan, memproses, menyimpan dan menyediakan sebagai *output* informasi yang diperlukan untuk mendukung sebuah organisasi.

Jadi dapat disimpulkan bahwa sistem informasi adalah kumpulan dari sumber daya manusia, *software*, *hardware*, jaringan dan memproses sumber daya data menjadi informasi yang berguna bagi pengguna akhir.

#### **2.1.1.2.4 Pengertian Sistem Informasi Manajemen**

Menurut McLeod dan Schell (2004, pp259-260), sistem informasi manajemen didefinisikan sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Para pemakai biasanya membentuk suatu entitas organisasi formal perusahaan atau subunit di bawahnya.

Menurut O'Brien yang diterjemahkan oleh Fitriyani dan Kwary (2006, p443), sistem informasi manajemen adalah jenis awal dari sistem informasi yang dikembangkan untuk mendukung pengambilan keputusan manajerial. SIM menghasilkan produk informasi yang mendukung banyak kebutuhan pengambilan keputusan harian dari para

manajer dan praktisi bisnis. Laporan, tampilan, dan respon yang dihasilkan oleh sistem informasi manajemen menyediakan informasi yang telah ditetapkan oleh para pengambil keputusan untuk mencukupi kebutuhan informasi.

Jadi dapat disimpulkan bahwa sistem informasi manajemen adalah sebuah sistem informasi berbasis komputer yang merupakan jenis awal dari sistem informasi yang digunakan untuk menyediakan informasi bagi pengguna akhir.

## **2.1.2 Teori Khusus**

### **2.1.2.1 Karet Alam**

Mengacu pada Tim Penulis PS (Penebar Swadaya) (2011, p3), tanaman Karet memiliki peranan yang besar dalam kehidupan perekonomian Indonesia. Banyak penduduk yang hidup dengan mengandalkan komoditas penghasil getah ini. Karet tak hanya diusahakan oleh perkebunan-perkebunan besar milik negara yang memiliki areal ratusan ribu hektar, tetapi juga diusahakan oleh swasta dan rakyat.

Total luas perkebunan Karet di Indonesia hingga saat ini berkisar 3 juta hektar lebih terluas di dunia. Malaysia dan Thailand yang merupakan pesaing utama Indonesia memiliki luas lahan yang jauh di bawah jumlah tersebut. Sayangnya, lahan Karet yang luas di Indonesia tidak diimbangi dengan pengelolaan yang memadai. Hanya beberapa perkebunan besar milik negara dan beberapa perkebunan swasta saja yang pengelolaannya sudah lumayan. Sementara kebanyakan perkebunan Karet milik rakyat dikelola seadanya, bahkan ada yang tidak dirawat dan hanya mengandalkan pertumbuhan alami. Akibatnya, produktivitas Karet alam Indonesia per tahunnya berada di bawah Malaysia dan Thailand yang memiliki luas lahan jauh lebih sedikit.

### 2.1.2.2 Klaster dan Sentra Industri

Menurut Taufik (2011, p1) klaster (*cluster*) mempunyai pengertian harfiah sebagai kumpulan, kelompok, himpunan atau gabungan obyek tertentu yang memiliki keserupaan atau atas dasar karakteristik tertentu. Selain itu, menurut Michael Porter mendefinisikan klaster sebagai sekumpulan perusahaan dan lembaga-lembaga terkait karena kebersamaan dan komplementaritas. Dalam konteks ekonomi, klaster industri merupakan terminologi yang mempunyai pengertian khusus tertentu. Klaster industri sebenarnya merupakan kelompok industri spesifik yang dihubungkan oleh jaringan mata rantai proses (peningkatan) nilai tambah, baik melalui hubungan bisnis maupun non bisnis. Klaster industri yang berkembang dan berdaya saing umumnya tidak “terisolasi” pada satu produk, komoditas atau sektor saja, tetapi bersifat lintas sektor dan terkait satu dengan lainnya serta berada pada suatu “wilayah geografis” tertentu.

Klaster industri di Indonesia juga dapat disebut sebagai sebuah sentra industri. Berdasarkan peraturan Menteri Perindustrian Republik Indonesia Nomor: 78/M-IND/PER/9/2007 pada pasal 1 *point* ke-4, menerangkan bahwa sentra adalah suatu wilayah atau kawasan tertentu tempat sekelompok perusahaan IKM (Industri Kecil Menengah) yang menghasilkan produk sejenis, menggunakan bahan baku sejenis, atau melakukan proses pengerjaannya sama.

Jadi dapat disimpulkan bahwa sentra industri adalah kelompok industri spesifik atau perusahaan-perusahaan yang berada pada suatu wilayah atau kawasan tertentu yang dihubungkan oleh jaringan mata rantai (terkait) peningkatan nilai tambah, baik melalui hubungan bisnis maupun non-bisnis seperti menghasilkan produk sejenis, menggunakan bahan baku sejenis, atau melakukan proses pengerjaannya sama.

### **2.1.2.3 Aktivitas Pembelian**

Menurut Mulyadi (2001, p299), sistem pembelian digunakan dalam perusahaan untuk pengadaan barang yang diperlukan oleh perusahaan. Transaksi pembelian dapat digolongkan menjadi dua, yakni: pembelian lokal dan impor. Pembelian lokal adalah pembelian dari pemasok dalam negeri, sedangkan impor adalah pembelian dari pemasok luar negeri.

Jadi dapat disimpulkan bahwa kegiatan pembelian dilakukan oleh perusahaan untuk memperoleh *input* ke dalam perusahaan berupa barang.

### **2.1.2.4 Aktivitas Persediaan Barang**

Menurut Mulyadi (2001, p553), perusahaan dagang memiliki persediaan hanya terdiri dari satu golongan, yaitu persediaan barang dagangan, yang merupakan barang yang dibeli untuk tujuan dijual kembali.

Jadi dapat disimpulkan bahwa kegiatan persediaan barang dilakukan oleh perusahaan dagang adalah untuk melakukan penyimpanan sementara yang kemudian akan dilakukan penjualan.

### **2.1.2.5 Aktivitas Penjualan**

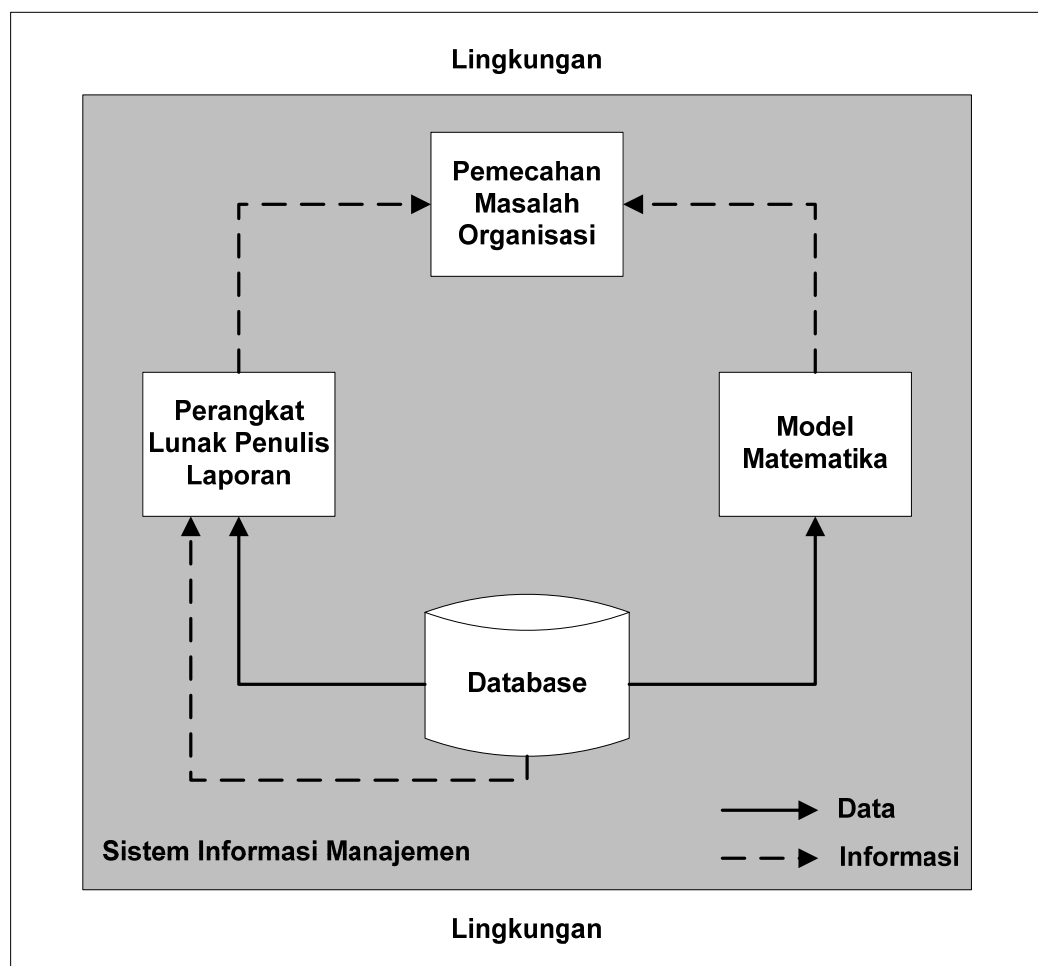
Menurut Mulyadi (2001, p202), kegiatan penjualan terdiri dari transaksi penjualan barang atau jasa, baik secara kredit maupun secara tunai. Dalam transaksi penjualan tunai, barang atau jasa baru diserahkan oleh perusahaan kepada pembeli jika perusahaan telah menerima kas dari pembeli.



Jadi dapat disimpulkan bahwa kegiatan penjualan terjadi ketika adanya penyerahan barang atau jasa kepada pembeli setelah pembeli melakukan pembayaran atas penerimaan barang atau jasa dari pihak penyedia barang atau jasa.

### 2.1.2.6 Sistem Informasi Manajemen

Menurut McLeod dan Schell (2004, p260), berikut ini adalah Gambar 2.1 suatu model Sistem Informasi Manajemen (SIM):



**Gambar 2.1 Model Sistem Informasi Manajemen**

Sumber : McLeod dan Schell, 2004, p260

Gambar 2.1 memperlihatkan dua jenis perangkat lunak yang menghasilkan informasi-penulis laporan dan pembuatan model matematika. Menurut McLeod dan Schell (2004,

p262), perangkat lunak penulis informasi terdiri dari program-program yang menghasilkan dua jenis laporan, yaitu:

- a. laporan periodik → laporan yang disiapkan sesuai jadwal tertentu. Contohnya adalah analisis penjualan bulanan menurut pelanggan
- b. dan laporan khusus → laporan yang disiapkan bila sesuatu yang luar biasa terjadi. Contohnya adalah laporan kecelakaan atau jawaban atas *database query*.

Kemudian untuk model matematika menurut McLeod dan Schell (2004, p265), ada tiga dimensi, yaitu:

- a. dimensi pengaruh waktu (Model Statis atau Dinamis) → model statis tidak menyertakan waktu sebagai variabel. Model ini berkaitan dengan suatu situasi pada suatu titik waktu tertentu. Sedangkan model dinamis adalah model yang menyertakan waktu sebagai variabel. Model ini menggambarkan perilaku entitas dari waktu ke waktu, seperti suatu film,
- b. dimensi tingkat keyakinan (Model Probabilistik atau Deterministik) → probabilitas adalah peluang terjadinya sesuatu dari 0,00 (sesuatu yang sama sekali tidak mungkin) hingga 1,00 (sesuatu yang pasti). Sedangkan deterministik adalah model sebaliknya.
- c. dan dimensi kemampuan mencapai optimisasi (Model Optimisasi atau Suboptimisasi) → model optimisasi adalah model yang memilih solusi terbaik dari berbagai alternatif. Sedangkan model suboptimisasi disebut *satisficing model*, memungkinkan *user* memasukkan serangkaian keputusan dan model akan memproyeksikan hasilnya.

Selain itu menurut McLeod dan Schell (2004, p274) Sistem Informasi Manajemen dan subsistem-subsistem organisasinya berkontribusi pada pemecahan masalah dalam dua cara dasar, yakni:

- a. Sumber daya informasi seorganisasi → Sistem Informasi Manajemen adalah suatu usaha seorganisasi untuk menyediakan informasi pemecahan masalah.
- b. Identifikasi dan pemecahan masalah → ide utama dibalik Sistem Informasi Manajemen adalah menjaga agar pasokan informasi terus mengalir ke manajer.

Menurut Masood dan Pires (2005, pp35-37), alasan diterapkannya MIS (*Management Information System*) adalah agar pengguna dapat bertemu untuk menghasilkan ide-ide, berbagi informasi dan untuk meningkatkan komunikasi melalui semua tingkatan organisasi. MIS dapat mendukung posisi kompetitif perusahaan. Semua perusahaan yang sukses mempunyai satu atau dua fungsi bisnis yang dilakukan lebih baik dari pada persaingan. Hal ini disebut sebagai sebuah kompetensi inti (*core competencies*). Kompetensi inti ini ditujukan untuk menjadi keunggulan kompetitif berkelanjutan.

Kecepatan perubahan yang cepat telah membuat akses ke informasi yang tepat waktu dan terkini menjadi kritis dalam lingkungan persaingan sistem informasi. Berikut ini adalah kunci keberhasilan penerapan MIS, yakni:

- a. Sebuah visi dan strategi yang diartikulasikan secara jelas untuk bisnis.
- b. Dukungan yang kuat dari manajemen senior.
- c. Berbagi informasi melalui semua tingkat bisnis.
- d. Penghasilan ide dan partisipasi karyawan dalam pengambilan keputusan.
- e. Men-*transfer* dari pengambilan keputusan yang nyata.
- f. Keterampilan pelatihan untuk mendukung ide-ide baru dan peran-peran.

- g. Pembinaan yang berkelanjutan dari pemimpin tim.
- h. Sistem untuk mendukung tim.

### **2.1.3 Sistem Dinamis**

#### **2.1.3.1 Gambaran Umum Sistem Dinamis**

Menurut Nielsen dan Nielsen (2008, p172), sistem dinamis (*system dynamic*) adalah sebuah perspektif dan satu set alat konseptual yang memungkinkan untuk memahami struktur dan dinamika sistem yang kompleks. Ini juga merupakan metode pemodelan cermat yang memungkinkan untuk membangun simulasi komputer formal dari sistem yang kompleks dan menggunakannya untuk merancang kebijakan dan organisasi yang lebih efektif.

Menurut Sehlke dan Jacobson (2005, pp722-724), sistem dinamis (*system dynamic*) adalah suatu pendekatan dengan bantuan komputer untuk mengevaluasi hubungan timbal balik berbagai komponen dan aktivitas dalam sistem yang kompleks. Saat ini, model sistem dinamis telah dikembangkan di berbagai bidang seperti rancangan kebijakan, pemodelan biologis dan medis, analisis energi dan lingkungan dan di berbagai area mengenai alam dan ilmu sosial. Sistem dinamis juga merupakan alat yang dapat membantu *stakeholders* untuk lebih memahami dan menyampaikan betapa sistem yang kompleks berperilaku melalui visualisasi dan simulasi komputer.

Model sistem dinamis adalah deskriptif secara alami. Unsur-unsur atau elemen-elemen dalam model harus sesuai dengan entitas aktual di dunia nyata. Aturan keputusan dalam model harus sesuai dengan praktek sebenarnya. Model sistem dinamis berguna untuk merancang kebijakan bukan keputusan. Kebijakan yang direpresentasikan dalam persamaan model sistem dinamis adalah aturan yang dipengaruhi oleh keputusan.

Kebijakan yang direkomendasikan menyatakan bagaimana membuat aliran keputusan dari waktu ke waktu.

Menurut Muhammadi dan Aminulla (2001, pp3-7), sistem adalah *keseluruhan interaksi* antarunsur dari sebuah *obyek* dalam *batas lingkungan* tertentu yang bekerja mencapai *tujuan*. Penjelasan dari pengertian sistem di atas, yakni:

- a. *keseluruhan* → mencakup bagian-bagian atau divisi,
- b. *interaksi* → pengikat atau penghubung antarunsur,
- c. *unsur* → benda, baik konkrit atau abstrak, yang menyusun obyek sistem,
- d. *obyek* → sistem yang menjadi perhatian,
- e. *batas* → sistem yang memiliki batasan disebut sistem tertutup. Sistem tertutup adalah sebuah sistem dengan batas yang dianggap kedap (tidak tembus) terhadap pengaruh lingkungan. Sistem tertutup itu hanya ada dalam anggapan (untuk analisis), karena pada kenyataan sistem selalu berinteraksi dengan lingkungan atau sebagai sebuah sistem terbuka,
- f. dan *tujuan* → unjuk kerja sistem yang teramati dan diinginkan.

Selain itu, Muhammadi dan Aminulla (2001, p11) juga berpendapat bahwa ketika adanya kesadaran untuk mengapresiasi dan memikirkan suatu kejadian sebagai suatu sistem disebut berpikir sistemik (*systemic approach*). Ada lima langkah yang dapat ditempuh untuk menghasilkan bangunan pemikiran (model) yang bersifat sistemik, yaitu:

- a. Identifikasi proses menghasilkan kejadian nyata.
- b. Identifikasi kejadian yang diinginkan.
- c. Identifikasi kesenjangan antara kenyataan dengan keinginan.
- d. Identifikasi dinamika menutup kesenjangan.

e. Analisis kebijakan.

Sistem dinamis pertama kali diperkenalkan oleh Jay W. Forrester di Massachusetts Institute of Technology (MIT) pada tahun 1950-an, merupakan suatu metode pemecahan masalah-masalah kompleks yang timbul karena adanya kecenderungan sebab-akibat dari berbagai macam variabel di dalam sistem. Metode sistem dinamik ini pertama kali diterapkan pada permasalahan manajemen seperti fluktuasi persediaan, ketidakstabilan tenaga kerja dan penurunan pangsa pasar suatu perusahaan.

Menurut Forrester karakteristik dari pemodelan sistem dinamis, yakni:

- a. Menggambarkan hubungan sebab akibat dari sistem.
- b. Sederhana dalam *mathematical nature*.
- c. Sinonim dengan terminologi dunia industri, ekonomi, dan sosial dalam tatanama.
- d. Dapat melibatkan banyak variabel.
- e. Dapat menghasilkan perubahan yang tidak kontinyu jika dalam keputusan memang dibutuhkan.

Kemudian Muhammadi et al. (2001, p28) menjelaskan bahwa sistem dinamis adalah sebuah sistem tertutup. Pengaruh faktor lingkungan terhadap sistem dimungkinkan terjadi dan perubahan eksternal itu dianggap sebagai variabel eksogen.

Jadi dapat disimpulkan bahwa sistem dinamis (*system dynamic*) adalah sebuah sistem tertutup yang merupakan pendekatan atau perspektif yang memungkinkan untuk memahami struktur masalah dan dinamika sistem yang kompleks yang timbul karena adanya kecenderungan sebab-akibat.

### 2.1.3.2 Struktur Sistem dan Diagram Simpal Kausal

Menurut Muhammadi et al. (2001, p23), struktur sistem adalah interaksi atau mekanisme. Setiap gejala apapun, baik fisik ataupun non-fisik, bagaimanapun kerumitannya, dapat disederhanakan menjadi struktur dasar yaitu mekanisme dari masukan, proses, keluaran dan umpan balik. Untuk memudahkan pekerjaan berpikir sistemik ini, struktur sistem dinamis disederhanakan ke dalam diagram simpal kausal. Ciri sistem tertutup dari sistem dinamis dalam hal ini ditunjukkan simpal umpan balik dari struktur.

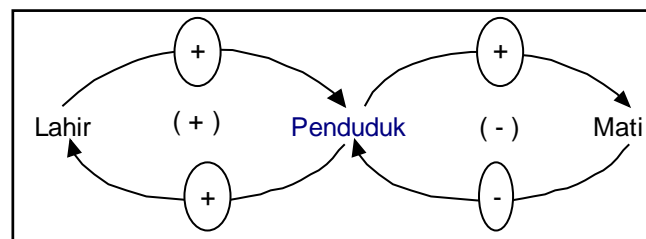
Menurut Kiani, Shirouyehzad dan Fouladgar (2009, p688), sistem dinamis berfokus pada struktur dan perilaku sistem yang terdiri dari interaksi simpal umpan balik. Diagram simpal kausal (*causal loop diagram*) adalah alat yang mudah, yang membantu para analisis untuk mengkonseptualisasi sistem dunia nyata dalam bentuk simpal umpan balik. Dalam diagram simpal kausal, panah-panah menunjukkan arah dari pengaruh, dan tanda tambah dan tanda kurang menunjukkan jenis dari pengaruh. Jika perubahan dalam satu variabel menghasilkan perubahan dalam arah yang sama dalam variable kedua, relatif terhadap nilai sebelumnya, hubungan antara dua variabel disebut sebagai positif. Jika perubahan dalam variabel kedua berlangsung di arah yang berlawanan, hubungan antara dua variabel disebut sebagai negatif.

Menurut Mehta dan Shah (2005, p17), diagram simpal kausal (*causal loop diagram*) digunakan untuk mendapatkan implikasi teoritis dan memperoleh arah hubungan hipotesis variabel-variabel dalam kerangka konseptual. Diagram simpal kausal merupakan alat yang penting untuk merepresentasikan struktur umpan balik dari sistem.

Menurut Viswanadhan (2004, p77), diagram simpal kausal (*causal loop diagram*) merupakan metode kualitatif dalam sistem dinamis. Diagram simpal kausal terdiri dari variabel yang dihubungkan dengan panah yang menunjukkan pengaruh kausal antara variabel-variabel. Simpal umpan balik yang penting juga diidentifikasi dalam diagram. Kelemahan dari diagram simpal kausal adalah ketidakmampuan untuk meng-*capture stock* dan struktur aliran (*flow structures*) dari sistem.

Menurut Muhammadi et al. (2001, pp28-30), diagram simpal kausal adalah pengungkapan tentang kejadian hubungan sebab-akibat (*causal relationships*) ke dalam bahasa gambar tertentu. Di sini bahasa gambar tersebut adalah panah yang saling mengait, sehingga membentuk diagram simpal (*causal loop*), di mana hulu panah mengungkapkan sebab dan ujung panah mengungkapkan akibat.

Keduanya, baik unsur sebab maupun akibat, atau salah satu diantaranya (sebab saja atau akibat saja) harus merujuk keadaan yang terukur, baik secara kualitatif untuk keadaan yang dirasakan (*perceived*) maupun secara kuantitatif untuk keadaan nyata (*actual*). Harus diingat logikanya adalah proses (*rate*) sebagai sebab yang menghasilkan keadaan (*level*) sebagai akibat, ataupun sebaliknya. Informasi tentang keadaan sebagai sebab menghasilkan pengaruh pada proses sebagai akibat. Contoh diagram simpal kausal dapat dilihat pada Gambar 2.2.



**Gambar 2.2 Diagram Simpal Kausal sederhana**  
Sumber : Muhammadi et al., 2001, p33



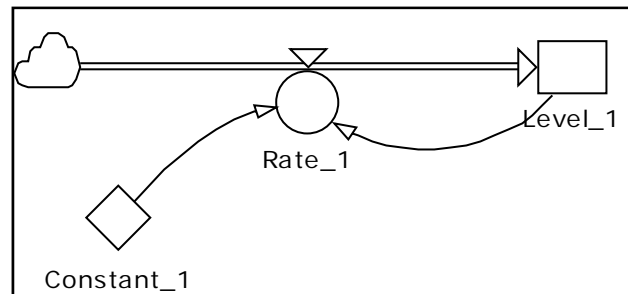
Pada Gambar 2.2 terdapat tulisan yang berwarna hitam adalah proses (*rate*) dan tulisan yang berwarna biru adalah keadaan (*level*). Tanda “( + )” menjelaskan bahwa simpal antara “Lahir” dan “Penduduk” adalah simpal positif, mempunyai perilaku percepatan atau perlambatan. Sedangkan tanda “( - )” menjelaskan bahwa simpal antara “Penduduk” dan “Mati” adalah simpal negatif, mempunyai perilaku menuju sasaran atau limit.

Jadi dapat disimpulkan bahwa diagram simpal kausal (*causal loop diagram*) merupakan sebuah metode kualitatif untuk menggambarkan sistem dinamis berupa hubungan timbal balik (sebab-akibat) dalam kerangka konseptual.

### **2.1.3.3 Diagram Alir**

Menurut Viswanadhan (2004, p77), diagram alir (*flow diagram*) merupakan metode kuantitatif dalam sistem dinamis. *Stock* dan *flows* bersamaan dengan umpan balik adalah merupakan dua pusat konsep dari teori sistem dinamis. *Stock* adalah akumulasi, yang mencirikan keadaan sistem dan menghasilkan informasi di mana keputusan dan tindakan itu didasarkan.

Menurut Soesilo (2001, p66), diagram alir adalah suatu sistem yang menggambarkan hubungan antara variabel-variabel. Variabel-variabel tersebut akan digambarkan dengan beberapa simbol.



**Gambar 2.3 Diagram Alir sederhana**

Sumber : Soesilo, 2001, p66

Pada Gambar 2.3 dapat dilihat beberapa simbol. Simbol pertama adalah simbol aliran “Rate\_1” yang selalu dihubungkan dengan simbol level “Level\_1” melalui simbol panah tebal untuk proses aliran. Aliran benda yang dapat mengalir di sini adalah barang, uang, orang dan lain-lain, yang diamati dan diukur penambahannya dan pengurangannya dalam level. Dalam pemodelan level adalah mewakili pokok persoalan yang menjadi perhatian. Selanjutnya panah halus (*information link*) yang menghubungkan antara *level* dengan aliran adalah proses informasi umpan balik. Simbol lainnya adalah simbol konstanta “Constant\_1”. Simbol ini merupakan simbol yang memiliki nilai konstan. Diagram alir menggambarkan struktur dari model. Sedangkan hasil simulasi berupa gambar atau grafik menggambarkan perilaku (*behavior*) dari sistem.

Jadi dapat disimpulkan bahwa diagram alir (*flow diagram*) merupakan sebuah metode kuantitatif untuk menggambarkan sistem dinamis berupa variabel-variabel yang digambarkan ke dalam simbol-simbol, yang mana setiap simbol memiliki nilai kuantitatif.

#### 2.1.3.4 Simulasi

Menurut Taylor et al. (2009, p69), pemodelan simulasi, atau pemodelan dan simulasi adalah bidang multi-disiplin yang mengagumkan yang mengacu pada berbagai

teknik dan alat. Pemodelan simulasi digunakan untuk penelitian perilaku sistem dunia nyata dalam berbagai bidang seperti komunikasi, pertahanan, kesehatan, manufaktur dan transportasi.

Menurut Muhammadi dan Soesilo (2001, pp51-52), simulasi adalah peniruan perilaku suatu gejala atau proses. Simulasi bertujuan untuk memahami gejala atau proses tersebut, membuat analisis dan peramalan perilaku gejala atau proses tersebut di masa depan. Simulasi dilakukan melalui tahap-tahap seperti berikut: penyusunan konsep, pembuatan model, simulasi dan validasi hasil simulasi.

Jadi dapat disimpulkan bahwa simulasi adalah penelitian perilaku sistem dunia nyata dengan melakukan peniruan perilaku suatu gejala atau proses.

## **2.1.4 *Object Oriented Analysis and Design (OOAD)***

### **2.1.4.1 Pengertian OOAD**

#### **2.1.4.1.1 *Object***

Menurut Mathiassen et al. (2000, p4), *object* adalah suatu entitas yang memiliki *identity*, *state* dan *behavior*.

Menurut Whitten, Bentley dan Dittman (2004, p179), *object* adalah pembungkusan data yang mendeskripsikan orang, obyek, tempat, kejadian, atau sesuatu yang berlainan, dengan semua proses (disebut *metode*) yang diizinkan untuk menggunakan atau memperbaiki data dan properti-properti tersebut.

Jadi dapat disimpulkan bahwa *object* adalah entitas dengan sebuah proses yang diizinkan untuk menggunakan atau memperbaiki data dan properti-properti.

#### **2.1.4.1.2 *Object Oriented Analysis***

Menurut Whitten, Bentley dan Dittman (2004, p179), *object-oriented analysis* (OOA) adalah sebuah teknik yang *model-driven* yang mengintegrasikan data dan proses ke dalam konstruksi yang disebut obyek.

Menurut McLeod dan Schell (2004, p138-140), langkah-langkah dalam analisis sistem adalah sebagai berikut:

1. Mengumumkan penelitian sistem.
2. Mengorganisasikan Tim Proyek.
3. Mendefinisikan kebutuhan informasi.
4. Mendefinisikan kriteria kinerja sistem.
5. Menyiapkan usulan rancangan.
6. Menyetujui atau menolak rancangan proyek.

Jadi dapat disimpulkan bahwa *object-oriented analysis* (OOA) adalah teknik mengintegrasikan data dan proses ke dalam sebuah obyek.

#### **2.1.4.1.3 *Object Oriented Design***

Menurut Whitten, Bentley dan Dittman (2004, p648), *object-oriented design* (OOD) adalah pendekatan yang digunakan untuk menentukan solusi perangkat lunak dalam hal menggabungkan obyek, atribut dan metodenya.

Menurut McLeod dan Schell (2004, p140-143), langkah-langkah dalam perancangan sistem adalah sebagai berikut:

1. Menyiapkan rancangan sistem yang rinci.
2. Mengidentifikasi berbagai alternatif konfigurasi sistem.
3. Mengevaluasi berbagai alternatif konfigurasi sistem tersebut.

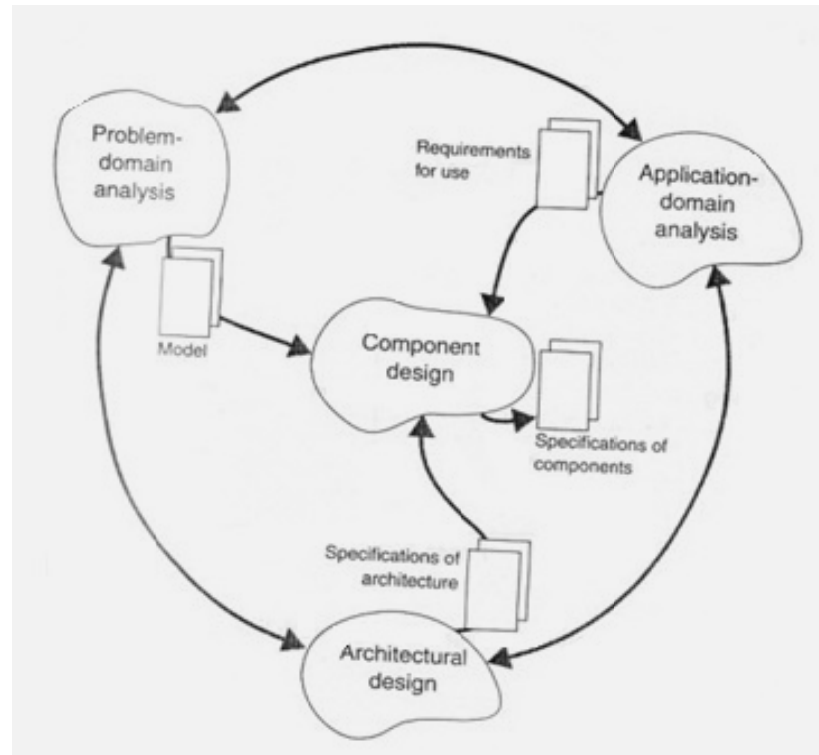
4. Memilih konfigurasi yang terbaik.
5. Menyiapkan usulan penerapan.
6. Menyetujui atau menolak penerapan sistem.

Jadi dapat disimpulkan bahwa *object-oriented design* (OOD) adalah proses lanjutan dari analisis sistem yang berjalan saat ini untuk menemukan kebutuhan untuk menghasilkan rancangan sistem baru yang diusulkan.

#### **2.1.4.1.4 *Object Oriented Analysis and Design (OOAD)***

Menurut Whitten, Bentley dan Dittman (2004, p27), *object-oriented analysis and design* (OOAD) adalah kumpulan alat dan teknik untuk pengembangan sistem yang akan menggunakan teknologi obyek untuk mengkonstruksi sebuah sistem dan perangkat lunaknya.

Menurut Mathiassen et al.(2000, p15), *object-oriented analysis and design* (OOAD) memiliki empat aktivitas utama yang tergambar pada Gambar 2.4.



**Gambar 2.4 Aktivitas utama dalam OOAD**

Sumber : Mathiassen et al., 2000, p15

Jadi dapat disimpulkan bahwa *object-oriented analysis and design* (OOAD) kumpulan alat dan teknik untuk pengembangan sistem yang memiliki empat aktivitas utama yakni: *problem-domain analysis*, *application-domain analysis*, *architectural design* dan *component design*.

#### **2.1.4.2 System Definition**

Menurut Mathiassen et al. (2000, pp23-24), *system definition* adalah deskripsi singkat dari sebuah sistem terkomputerisasi yang dinyatakan dalam bahasa alami (*natural language*). *System definition* menggambarkan *properties* yang mendasar untuk pengembangan dan penggunaan sistem. *System definition* dapat membantu dalam menampung pandangan umum dari pilihan yang berbeda-beda dan bisa digunakan untuk

perbandingan alternatif. *System definition* akan dinyatakan dalam *rich picture* dan *FACTOR criterion*.

Jadi dapat disimpulkan bahwa *system definition* adalah suatu cara untuk menggambarkan sebuah sistem yang terkomputerisasi dengan menggunakan bahasa alami seperti *rich picture* dan *FACTOR criterion*.

#### **2.1.4.2.1 Rich Picture**

Menurut Mathiassen et al. (2000, p26-27), *rich picture* adalah penggambaran informal yang memberikan pemahaman mengenai situasi kepada ilustrator. *Rich picture* berfokus kepada aspek-aspek utama dari situasi yang dideterminasikan oleh ilustrator. *Rich picture* harus memberikan penjelasan-penjelasan dari situasi yang memungkinkan beberapa alternatif pemahaman.

Jadi dapat disimpulkan bahwa *rich picture* merupakan salah satu cara untuk menggambarkan sebuah sistem yang terkomputerisasi dengan bahasa alami yang memberikan pemahaman mengenai situasi kepada ilustrator.

#### **2.1.4.2.2 FACTOR Criterion**

Menurut Mathiassen et al. (2000, p39-40), kriteria *FACTOR* memiliki enam elemen, yakni:

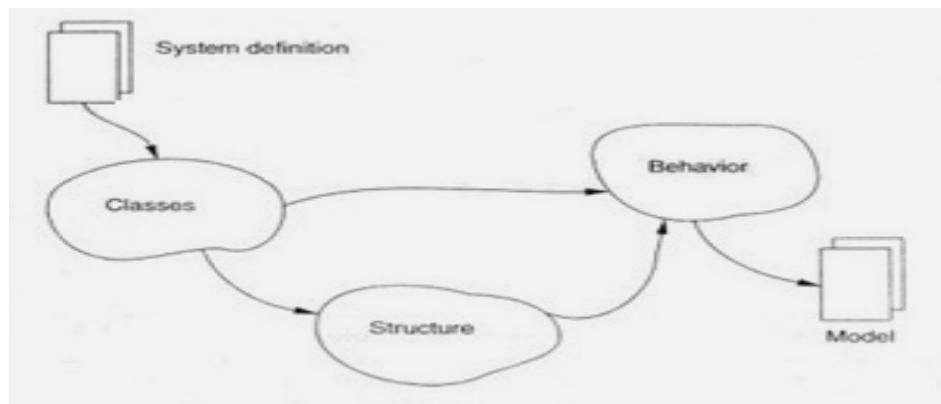
- a. *Funtionality* → fungsi sistem yang mendukung tugas-tugas *application-domain*.
- b. *Application domain* → bagian dari organisasi yang mengatur, memonitor dan mengendalikan *problem domain*.
- c. *Conditions* → kondisi dimana suatu sistem akan dikembangkan dan digunakan.

- d. *Technology* → teknologi yang digunakan untuk mengembangkan sistem dan teknologi dimana sistem akan dijalankan.
- e. *Objects* → obyek-obyek utama dalam *problem domain*.
- f. *Responsibility* → tanggung jawab seluruh sistem dalam hubungannya dengan konteks.

Jadi dapat disimpulkan bahwa *FACTOR criterion* merupakan salah satu cara untuk menggambarkan sebuah sistem yang terkomputerisasi dengan bahasa alami yang terbagi ke dalam enam elemen, yakni: *Functionality*, *Application domain*, *Conditions*, *Technology*, *Objects* dan *Responsibility*.

#### 2.1.4.3 Problem Domain Analysis

Menurut Mathiassen et al. (2000, p45) menyatakan *problem domain* adalah bagian dari keadaan yang akan diatur, dipantau dan dikontrol oleh sebuah sistem. *Model* adalah sebuah deskripsi dari *classes*, *objects*, *structures* dan *behavior* dalam sebuah *problem domain*. Tujuan dari *problem domain analysis* adalah untuk mengidentifikasi dan memodelkan sebuah *problem domain*. Hasil dari *problem domain analysis* adalah sebuah model yang koheren (masuk akal) dari sebuah *problem domain*.



**Gambar 2.5** Aktivitas dalam *Problem Domain Analysis*

Sumber : Mathiassen et al., 2000, p46



Gambar 2.5 menunjukkan aktivitas dalam *problem domain analysis*. Menurut Mathiassen et al. (2000, p46) terdapat tiga aktivitas dalam *problem domain analysis*, yakni:

a. *Classes*

Menurut Mathiassen et al. (2000, p49), *class* adalah sebuah deskripsi dari sekumpulan *object* yang berbagi *structure*, *behavioral pattern* dan *attributes*. *Event* adalah kejadian yang bersifat instant yang melibatkan satu atau lebih *object*. Tujuan dari *classes* adalah untuk memilih elemen-elemen dari *problem domain model*. Hasil dari *classes* adalah sebuah *event table* dengan *classes* dan *events* yang berhubungan.

b. *Structure*

Menurut Mathiassen et al. (2000, p69), terdapat dua jenis *structure*, yakni:

1. *Class structure* → menggambarkan hubungan konseptual yang statis antar-*class*.

Ada dua bentuk *class structure*, yakni:

a. *Generalization* → sebuah *general class* (*super class*) yang mendeskripsikan *properties* umum dari sekelompok *specialized classes* (*subclasses*).

b. *Cluster* → sekelompok koleksi dari *classes* yang berhubungan.

2. *Object structure* → menggambarkan hubungan dinamis dan konkrit antarobyek.

Ada dua bentuk *object structure*, yakni:

a. *Aggregation* → *superior object* (*the whole*) yang terdiri dari sejumlah obyek-obyek (*the parts*).

b. *Association* → hubungan yang berarti di antara sejumlah obyek-obyek.

Tujuan dari *structure* adalah untuk mendeskripsikan hubungan struktural antara *classes* dan *objects* dalam sebuah *problem domain*. Hasil dari *structure* adalah sebuah *class diagram* dengan *classes* dan *structures*.

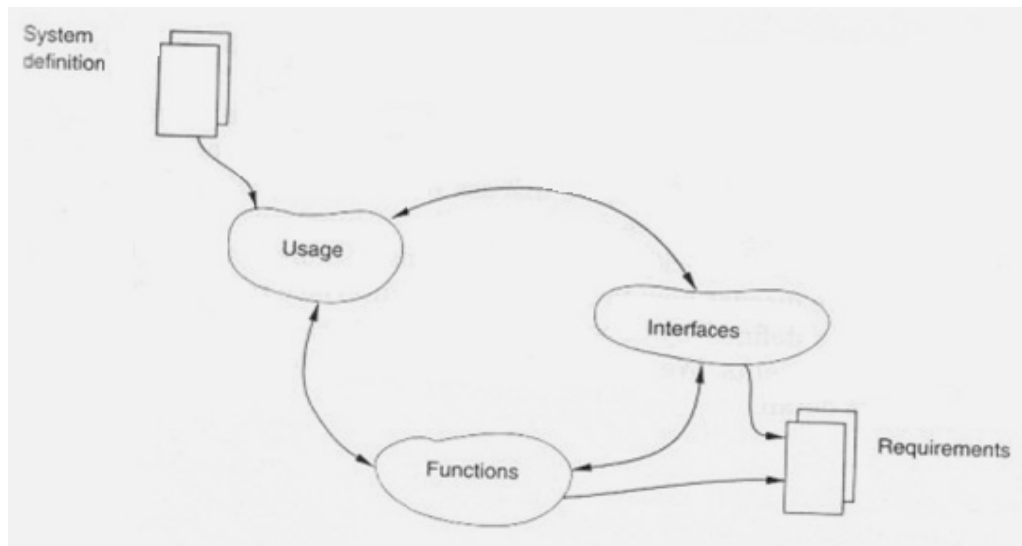
c. *Behavior*

Menurut Mathiassen et al. (2000, p89), *behavioral pattern* adalah sebuah deskripsi dari kemungkinan penemuan *event* untuk semua *objects* dalam *class* dan *event trace* adalah sebuah *sequence* dari *events* yang melibatkan *specific objects*. Tujuan dari *behavior* adalah untuk memodelkan dinamika dari sebuah *problem domain*. Hasil dari *behavior* adalah sebuah *behavior pattern* dengan atribut-atribut untuk setiap *class* dalam sebuah *class diagram*.

Jadi dapat disimpulkan bahwa *problem domain analysis* adalah proses mengidentifikasi dan memodelkan bagian dari keadaan yang akan diatur, dipantau dan dikontrol oleh sebuah sistem (*problem domain*) yang mencakup tiga aktivitas yakni: *classes*, *structure* dan *behavior*.

#### **2.1.4.4 Application Domain Analysis**

Menurut Mathiassen et al. (2000, p115), *application domain* adalah organisasi yang mengatur, memantau atau mengendalikan sebuah *problem domain*. *Requirements* adalah perilaku sebuah sistem eksternal yang dapat diamati. Tujuan dari *application domain analysis* adalah untuk menentukan persyaratan kebutuhan dari sebuah sistem. Hasil dari *application domain analysis* adalah sebuah daftar lengkap dari keseluruhan persyaratan kebutuhan dari sebuah sistem.



**Gambar 2.6** Aktivitas dalam *Application Domain Analysis*

Sumber : Mathiassen et al., 2000, p117

Gambar 2.6 menunjukkan aktivitas dalam *application domain analysis*. Menurut Mathiassen et al. (2000, p117) terdapat tiga aktivitas dalam *application domain analysis*, yakni:

a. *Usage*

Menurut Mathiassen et al. (2000, p119), *actor* adalah sebuah abstraksi dari pengguna-pengguna atau sistem lain yang berinteraksi dengan target sistem. *Use case* adalah sebuah pola untuk interaksi antara sistem dan *actor* dalam *application domain*. Tujuan dari *usage* adalah untuk menentukan bagaimana *actor* berinteraksi dengan sistem. Hasil dari *usage* adalah untuk mendeskripsikan keseluruhan dari *use cases* dan *actor*.

b. *Function*

Menurut Mathiassen et al. (2000, p137), *function* adalah sebuah fasilitas untuk membuat sebuah model yang berguna bagi *actors*. Tujuan dari *function* adalah untuk menentukan kemampuan pemrosesan informasi dari sistem. Hasil dari *function*

adalah sebuah daftar lengkap dari fungsi-fungsi dengan spesifikasi dari fungsi yang kompleks. Selain itu, menurut Mathiassen et al. (2000, p138) terdapat empat tipe *function*, yakni:

1. *Update functions* → diaktifkan oleh sebuah kejadian pada *problem-domain* dan hasilnya dalam sebuah perubahan dalam *model's state*.
2. *Signal functions* → diaktifkan oleh sebuah perubahan dalam *model's state* dan hasilnya dalam sebuah reaksi dalam konteks.
3. *Read functions* → diaktifkan oleh kebutuhan akan informasi dalam tugas kerja *actor* dan hasilnya dalam sistem menampilkan bagian dari model yang relevan.
4. *Compute functions* → diaktifkan oleh kebutuhan akan informasi dalam tugas kerja *actor* dan hasilnya terdiri dari perhitungan yang melibatkan informasi yang disediakan oleh *actor* dalam model.

c. *Interface*

Menurut Mathiassen et al. (2000, p137), *interface* adalah sebuah fasilitas yang membuat model dari sistem dan fungsi dapat digunakan oleh *actors*. *User interface* adalah tampilan untuk *user*. *System interface* adalah tampilan untuk sistem lain. Tujuan dari *interface* adalah untuk menentukan *interfaces* dari sistem. Hasil dari *user interface* adalah *dialogue styles* dan *presentation forms*, sebuah daftar lengkap dari elemen-elemen *user interface*, *selected windows diagrams* dan *navigation diagram*.

Ketika *interface* telah dibuat, langkah selanjutnya adalah membuat *sequence diagram*. Menurut Mathiasse et al. (2000, p340), *sequence diagram* menggambarkan interaksi di antara beberapa obyek dalam jangka waktu tertentu. Sebuah *sequence*

*diagram* dapat menampung rincian situasi yang kompleks dan dinamis yang melibatkan beberapa dari banyak obyek yang digeneralisasikan dari *class* pada *class diagram*.

Menurut Bennett, McRobb dan Farmer (2006, pp252-253), *sequence diagram* adalah *semantically equivalent* pada sebuah diagram komunikasi untuk interaksi-interaksi sederhana. Sebuah interaksi menjelaskan pola komunikasi di antara satu set dari *objects* atau sistem-sistem yang terlibat dalam sebuah kolaborasi. *Sequence diagram* menunjukkan sebuah interaksi antara *objects* yang ditata dalam urutan waktu.

Menurut Bennet, et al. (2006, p270) ada dua belas tipe *interaction operator* yang digunakan dalam *fragment*, yakni: *alt*, *opt*, *break*, *par*, *seq*, *strict*, *neg*, *critical*, *ignore*, *consider*, *assert* dan *loop*. Ada tiga *fragment* yang sering digunakan, yakni:

- a. *Alt* → *alternative* ini mewakili *alternative behaviors* yang ada, setiap *behavior* ditampilkan dalam operasi yang terpisah.
- b. *Opt* → *option* ini merupakan pilihan tunggal atas operasi yang hanya akan dieksekusi bila batasan interaksi bernilai *true*.
- c. *Loop* → *loop* digunakan untuk mengindikasikan sebuah operasi yang diulang berkali-kali sampai batasan interaksi untuk pengulangan berakhir.

Menurut Bell (2004, p2) tujuan *sequence diagram*, yakni:

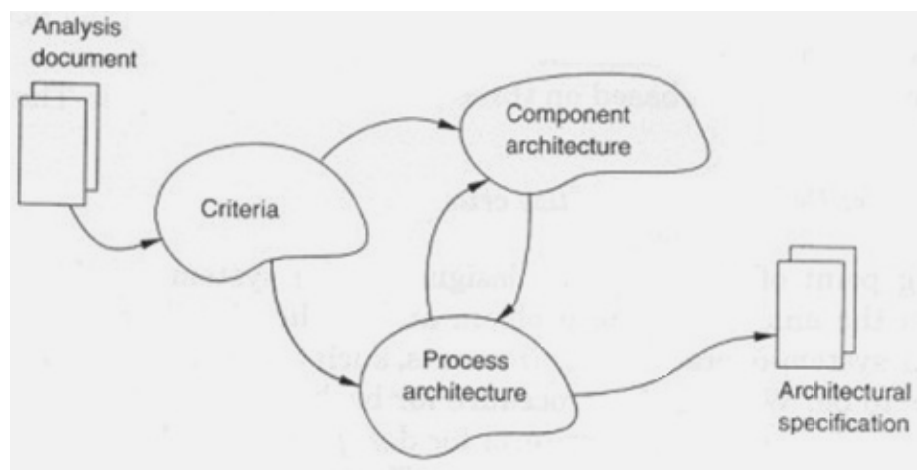
- a. Menggambarkan interaksi di antara obyek-obyek dalam suatu urutan *sequential* ketika interaksi itu terjadi.
- b. Bagi staf organisasi dapat menemukan kegunaan *sequence diagram* untuk mengkomunikasikan bagaimana sebuah bisnis berjalan dengan menunjukkan bagaimana obyek-obyek bisnis yang berbeda berinteraksi.
- c. Bagi staf teknis, *sequence diagram* berguna untuk dokumentasi bagaimana sistem yang akan datang akan dibuat.

Selain itu, Bell (2004, p20) memberikan kesimpulan tentang *sequence diagram* yakni: *sequence diagram* adalah sebuah diagram yang digunakan untuk mendokumen sebuah persyaratan dari sistem dan *flush out* dari rancangan sistem. *Sequence diagram* sangat berguna karena *sequence diagram* menunjukkan logika interaksi antara obyek-obyek dalam sistem pada urutan waktu terjadinya interaksi.

Jadi dapat disimpulkan bahwa *application domain analysis* adalah suatu proses menentukan persyaratan kebutuhan dari sebuah *problem domain* yang mencakup tiga aktivitas yakni: *usage*, *function* dan interface (pada aktivitas *interface* dilengkapi dengan *sequence diagram*).

#### 2.1.4.5 Architectural Design

Menurut Mathiassen et al. (2000, p173), tujuan dari *architectural design* adalah untuk menstruktur sistem yang terkomputerisasi. Hasil dari *architectural design* adalah struktur-struktur untuk komponen dan proses dari sebuah sistem.



**Gambar 2.7** Aktivitas dalam *Architectural Design*

Sumber : Mathiassen et al., 2000, p176

Gambar 2.7 menunjukkan aktivitas dalam *architectural design*. Penjelasan tiga aktivitas pada *architectural design*, yakni:

a. *Criteria*

Menurut Mathiassen et al. (2000, p177), *criteria* adalah properti yang lebih dipilih dari sebuah *architecture*. *Condition* adalah teknik, organisir dan peluang manusia dan keterbatasan yang terlibat dalam menjalankan tugas. Tujuan dari *criteria* adalah untuk mengatur prioritas rancangan. Hasil dari *criteria* adalah sekumpulan koleksi dari *prioritized criteria*.

**Tabel 2.1 *Criteria* klasik untuk kualitas *software***

Sumber : Mathiassen et al., 2000, p178

No.	<i>Criteria</i>	Pengukuran dari
1	<i>Usable</i>	Kemampuan adaptasi sistem dalam konteks organisasi, hubungan kerja dan teknikal.
2	<i>Secure</i>	Suatu pencegahan melawan akses yang tidak terotorisasi terhadap data dan fasilitas-fasilitas.
3	<i>Efficient</i>	Eksplorasi secara ekonomis dari fasilitas <i>technical platform</i> .
4	<i>Correct</i>	Pemenuhan terhadap <i>requirements</i> .
5	<i>Reliable</i>	Pemenuhan terhadap eksekusi <i>function</i> yang tepat.
6	<i>Maintainable</i>	Besarnya usaha untuk mengalokasikan dan memperbaiki kecatatan sistem.
7	<i>Testable</i>	Besarnya usaha untuk memastikan bahwa sistem menampilkan fungsi-fungsi yang telah ditentukan.
8	<i>Flexible</i>	Besarnya usaha untuk memodifikasi sistem.
9	<i>Comprehensible</i>	Usaha yang dibutuhkan untuk mendapatkan pemahaman yang masuk akal terhadap sistem.
10	<i>Reusable</i>	Kemungkinan penggunaan bagian dari sistem dalam sistem lain yang terhubung.
11	<i>Portable</i>	Besarnya usaha untuk memindahkan sistem ke <i>technical platform</i> lain.
12	<i>Interoperable</i>	Besarnya usaha untuk menggabungkan suatu sistem ke sistem lain.

b. *Component architecture*

Menurut Mathiassen et al. (2000, p189), *component architecture* adalah sebuah struktur sistem dari komponen yang saling berhubungan. *Component* adalah

sebuah koleksi dari bagian *program* yang merupakan keseluruhan dan memiliki tanggung jawab yang jelas. Tujuan dari *components* adalah untuk menciptakan struktur sistem yang dapat dipahami dan fleksibel. Hasil dari *components* adalah sebuah *class diagram* dengan spesifikasi dari *complex components*.

**Tabel 2.2 Bentuk distribusi dalam *client-server architecture***

Sumber : Mathiassen et al., 2000, p200

No.	<i>Client</i>	<i>Server</i>	<i>Architecture</i>
1	U	U + F + M	<i>Distributed presentation</i>
2	U	F + M	<i>Local presentation</i>
3	U + F	F + M	<i>Distributed functionality</i>
4	U + F	M	<i>Centralized data</i>
5	U + F + M	M	<i>Distributed data</i>

#### c. *Process architecture*

Menurut Mathiassen et al. (2000, p209), tujuan dari *process* adalah untuk mendefinisikan struktur fisik dari sebuah sistem. Hasil dari *process architecture* biasanya berupa *deployment diagram*.

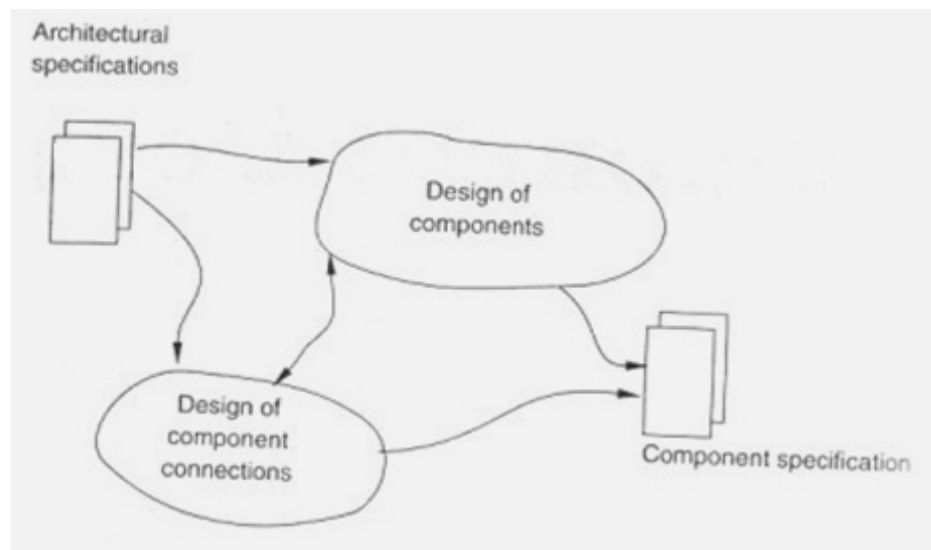
Jadi dapat disimpulkan bahwa *architectural design* adalah suatu proses untuk menstruktur sistem yang terkomputerisasi yang mencakup tiga aktivitas yakni: *criteria*, *component architecture* dan *process architecture*.

#### 2.1.4.6 *Component Design*

Menurut Mathiassen et al. (2000, p231), *Component* adalah sebuah koleksi dari bagian *program* yang merupakan keseluruhan dan memiliki tanggung jawab yang jelas. Tujuan dari *component design* adalah untuk menentukan sebuah penerapan dari persyaratan yang terdapat pada sebuah kerangka kerja *architectural*. Selain itu, menurut



Mathiassen et al. (2000, p232) terdapat dua aktivitas dalam *component design*, yakni: *design of components* dan *connecting components*.



**Gambar 2.8** Aktivitas dalam *Component Design*

Sumber : Mathiassen et al., 2000, p232

Gambar 2.8 menunjukkan aktivitas dalam *component design*. Penjelasan dua aktivitas pada *component design*, yakni:

a. *Design of components*

Pada aktivitas *design of components* terdapat dua subaktivitas, yakni: *model component* dan *function component*. Menurut Mathiassen et al. (2000, p235), *model component* adalah bagian dari sistem yang mengimplementasikan model *problem domain*. Tujuan dari *model component* adalah untuk merepresentasikan sebuah model dari sebuah *problem domain*. Hasil dari *model component* adalah sebuah *class diagram* dari *model component*. Menurut Mathiassen et al. (2000, p251), *function component* adalah bagian dari sistem yang mengimplementasikan persyaratan fungsional. Tujuan dari *function component* adalah untuk menentukan implementasi dari fungsi-fungsi. Hasil dari *function component* adalah sebuah *class diagram* dengan operasi-operasi dan spesifikasi dari operasi-operasi yang kompleks.

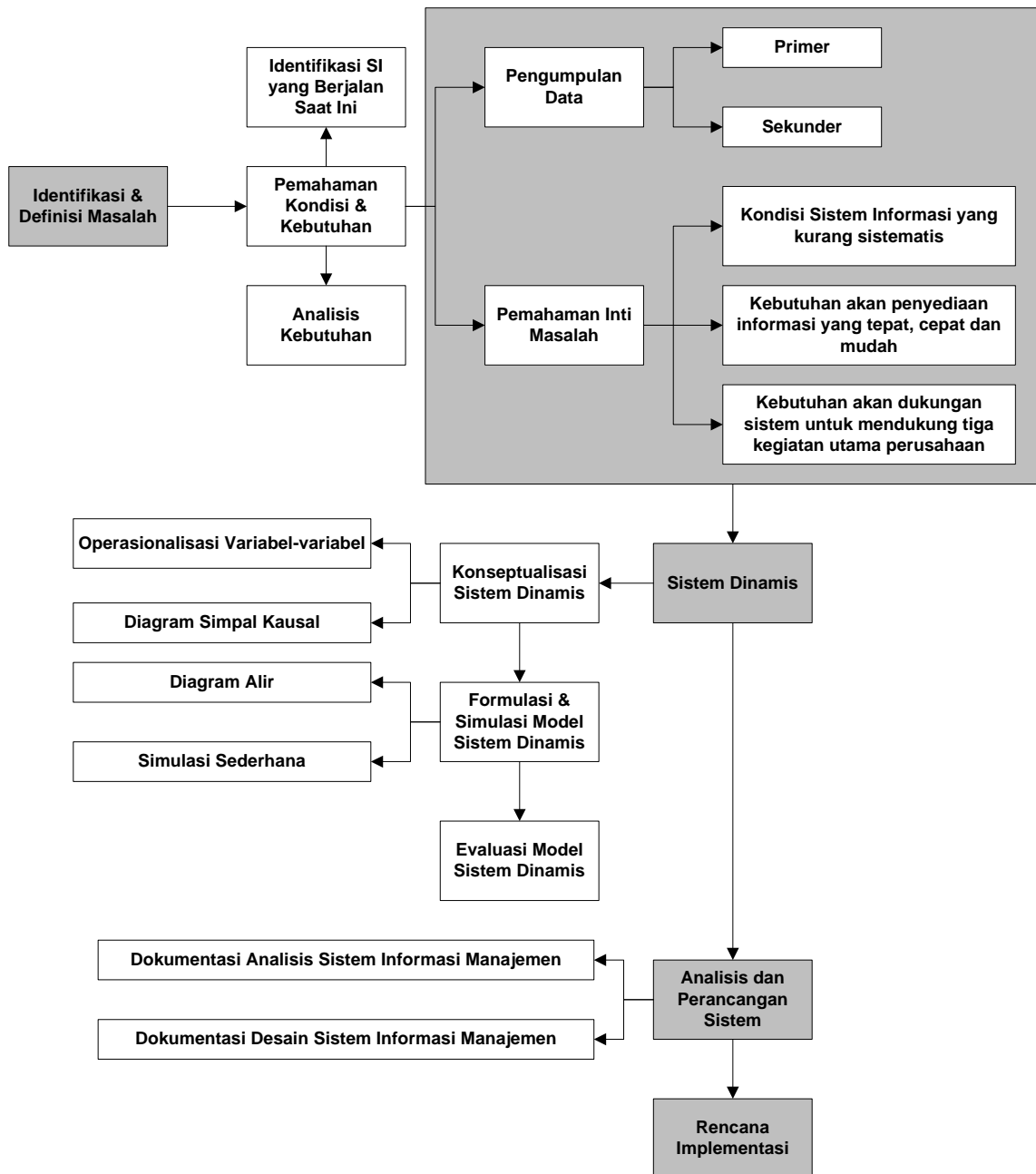
*b. Connecting components*

Menurut Mathiassen et al. (2000, p271) terdapat dua jenis *connecting components*, yakni: *coupling* adalah sebuah ukuran dari seberapa dekat dua *class* atau *components* terhubung dan *cohesion* adalah sebuah ukuran dari seberapa baik sebuah *class* atau *component* terikat secara bersama-sama.

Jadi dapat disimpulkan bahwa *component design* adalah suatu proses untuk menentukan sebuah penerapan dari persyaratan yang terdapat pada sebuah kerangka kerja *architectural* yang mencakup dua aktivitas yakni: *design of component* dan *connecting components*.

## 2.2 Kerangka Pikir

Berikut Gambar 2.9 adalah bagan kerangka pikir:



Gambar 2.9 Bagan kerangka pikir