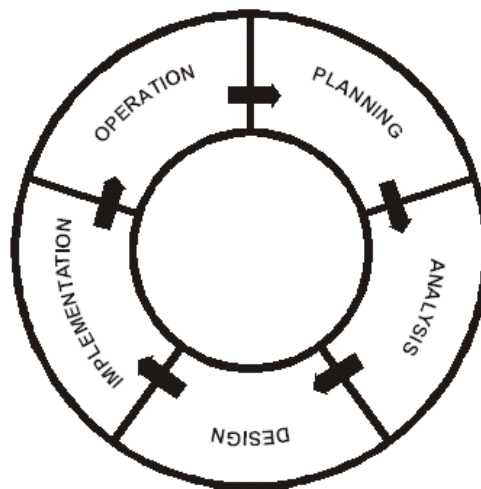


## 6.4 Siklus Hidup Pengembangan Sistem

Pengembangan sistem informasi merupakan suatu proses yang kompleks, mahal karena menggunakan banyak sumberdaya dan membutuhkan waktu yang lama dari berbulan-bulan sampai bertahun-tahun. Untuk memastikan keberhasilan pengembangan sistem informasi ini digunakan tahapan-tahapan yang dikenal dengan nama Siklus Hidup Pengembangan Sistem (System Development Life Cycle-SDLC). SDLC adalah suatu aplikasi dari pendekatan sistem untuk pengembangan sistem informasi, siklus ini meliputi tahap-tahap perencanaan, analisis, desain, implementasi, dan penggunaan. SDLC tradisional berbentuk air terjun, pada perkembangannya SDLC menjadi berbentuk lingkaran.



Gambar 9 Siklus hidup pengembangan sistem

Sumber: Raymond McLeod Jr. dan George Shell dalam, *Sistem Informasi Manajemen (terjemahan)*, 9/E (New Jersey, Prentice Hall, 2001) hal. 185

## 6.5 Pendekatan Pengembangan Sistem

Wilkinson dalam buku *accounting and information system* yang dikutip oleh Jogiyanto menyatakan dalam pengembangan sistem menggunakan SDLC terdapat beberapa pendekatan, yaitu

- Pendekatan klasik dan pendekatan terstruktur
- Pendekatan sepotong dan pendekatan sistem
- Pendekatan bawah-naik dan pendekatan atas-turun
- Pendekatan menyeluruh dan pendekatan modular
- Pendekatan lompatan dan pendekatan berkembang

### 6.5.1 Pendekatan Klasik dan Pendekatan Terstruktur

Pendekatan klasik adalah pendekatan pengembangan yang menggunakan tahap-tahap SDLC namun tidak dibekali teknik yang memadai, dengan pendekatan ini proses pengembangan relative mudah dan cepat karena disesuaikan dengan kebutuhan yang teridentifikasi selama proses berjalan namun pengembangan software menjadi sulit dan tidak terarah karena tidak ada acuan yang disepakati

untuk digunakan sebagai alat komunikasi antar anggota tim, walaupun biaya perencanaan, analisis, dan desain bias ditekan namun biaya pemeliharaan menjadi mahal akibat ketidak-sempurnaan sistem yang dikembangkan akan menyulitkan pemeliharaan. Kemungkinan terjadi kesalahan akibat tidak dilakukannya pengujian selama proses pengembangan, keberhasilan kurang terjamin karena banyak asumsi yang diterapkan, dan terjadi kesulitan dalam penerapan sistem karena ketidak-sesuaian dengan pemakai.

Pendekatan terstruktur berkembang sejak tahun 1970an sebagai jawaban dari kegagalan sistem klasik, yaitu dengan mengembangkan penggunaan alat-alat, teknik-teknik, dan dokumentasi yang digunakan sebagai acuan dalam pengembangan dan komunikasi. Pendekatan terstruktur ini menghasilkan beberapa metodologi pengembangan sistem. Dengan pendekatan ini masalah yang kompleks di dekomposisi menjadi masalah yang lebih sederhana dan terkendali sehingga hasil pengembangannya menjadi mudah dipelihara, fleksibel, sesuai dengan pemakai, tepat waktu, dan kualitas yang terjamin.

### **6.5.2 Pendekatan Sepotong dan Pendekatan Sistem**

Pendekatan sepotong (piecemeal) adalah pendekatan yang menekankan pada satu titik permasalahan tertentu yang terjadi dalam subsistem perusahaan. Pendekatan ini efektif jika ditinjau dari pencapaian tujuan subsistem itu namun bisa terjadi ketidak-sesuaian dengan tujuan perusahaan secara keseluruhan.

Pendekatan sistem melihat sebagai suatu integrasi, sehingga setiap bagian yang dikembangkan harus memperhatikan bagian-bagian yang lain dan tujuan perusahaan secara keseluruhan, dengan pendekatan sistem diharapkan terjadi efisiensi yang menyeluruh.

### **6.5.3 Pendekatan Bawah-Naik dan Pendekatan Atas-Turun**

Pendekatan bawah-naik atau Bottom-Up dimulai dari level bawah organisasi, yaitu level operasional tempat dilakukannya transaksi. Menekankan pada data, sedang informasi yang akan dihasilkan menyusul ketersediaan data, pendekatan ini disebut data analysis.

Pendekatan atas-turun atau Top-Down dimulai dari level atas organisasi dengan menetapkan rencana strategis, dimulai dengan mendefinisikan sasaran dan kebijakan organisasi. Setelah kebutuhan informasi ditetapkan maka turun ke penentuan output, input, basis data, prosedur operasi, dan control, pendekatan ini disebut sebagai decision analysis.

### **6.5.4 Pendekatan Menyeluruh dan Pendekatan Modular**

Pendekatan menyeluruh merupakan pendekatan pengembangan sistem secara serentak dan menyeluruh. Pendekatan ini dapat digunakan untuk sistem kecil atau sederhana. Untuk sistem yang kompleks pendekatan ini menjadi tidak implementatif.

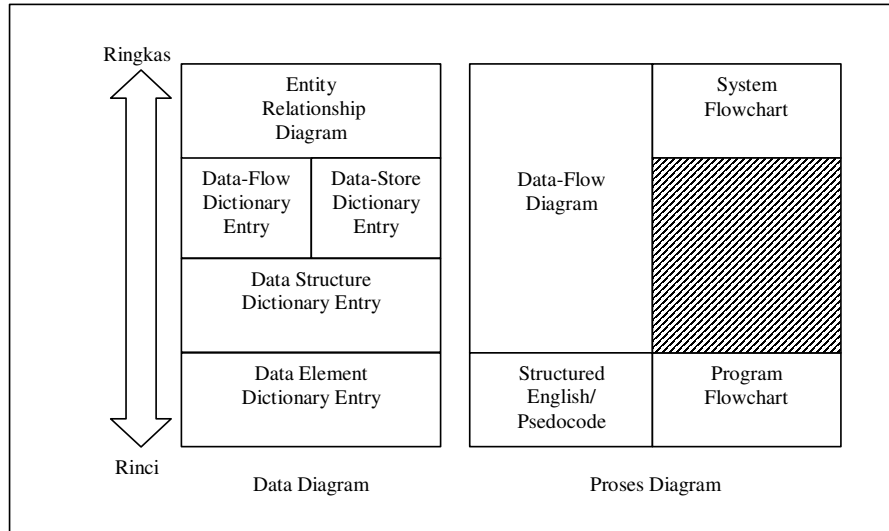
Pendekatan modular berusaha memecah sistem yang kompleks menjadi beberapa bagian yang sederhana, sehingga sistem akan lebih mudah untuk dikembangkan.

## **6.6 Metodologi Pengembangan Sistem**

Metodologi pengembangan sistem dibuat dan diusulkan oleh para peneliti, penulis buku, dan pengembang sistem. Dalam matakuliah ini metodologi yang akan dibahas adalah beberapa metodologi yang digunakan dalam pengembangan sistem informasi yaitu Structured Analysis Design Technique (SADT) dan object modeling technique (OMT).

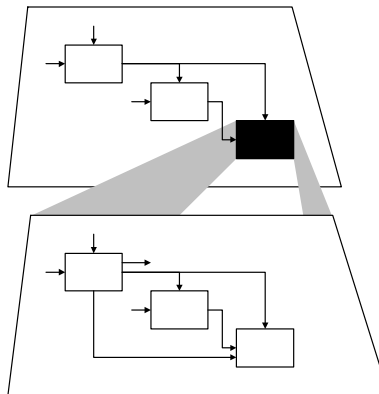
### 6.6.1 Metodologi Structured Analysis and Design Technique (SADT)

SADT merupakan representasi pendekatan arus data (*data flow approach*) merupakan pendekatan yang mudah diaplikasikan dan memiliki dokumentasi yang baik. SADT menekankan pengembangan pada dekomposisi fungsional. Sistem dilihat sebagai penyedia satu atau banyak fungsi bagi pengguna akhir (*end user*). Secara umum gambar berikut menunjukkan alat-alat yang digunakan dalam SADT.



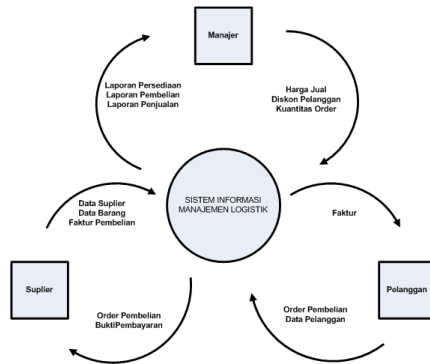
Gambar 10 Diagram Dalam SADT

Model diagram arus data (*data flow diagrams*) mentransformasi data sebagai aliran yang melalui sistem. Diagram arus data meliputi proses, arus data, aktor, dan tempat penyimpanan. Dimulai dari tingkat teratas, lalu secara rekursi membagi proses yang kompleks menjadi subdiagram-subdiagram, sampai dengan proses-proses kecil yang mudah diimplementasikan. Ketika proses telah cukup sederhana maka dekomposisi dihentikan, dan spesifikasi proses ditulis untuk setiap proses terendah.

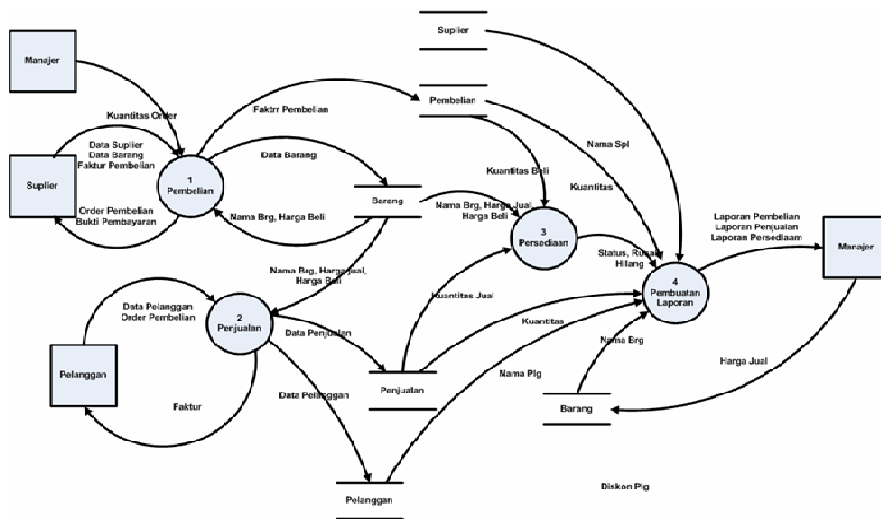


Gambar 11 Proses Dekomposisi Diagram Arus Data

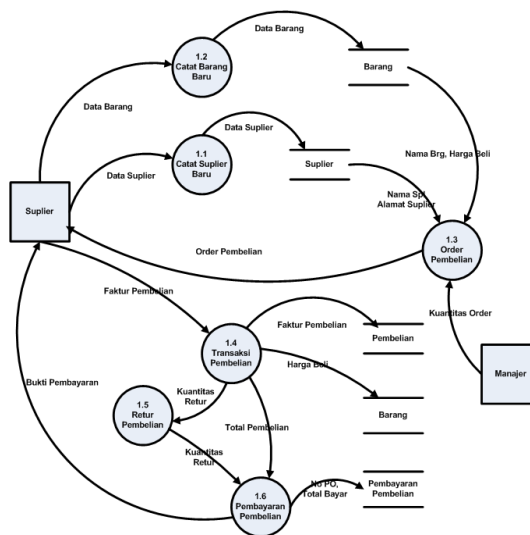
Gambar berikut adalah contoh dari dekomposisi diagram arus data



Gambar 11 Contoh Diagram Arus Data Level 0 (Diagram Konteks)



Gambar 12 Contoh Diagram Arus Data Level 1



Gambar 12 Contoh Diagram Arus Data Level 2

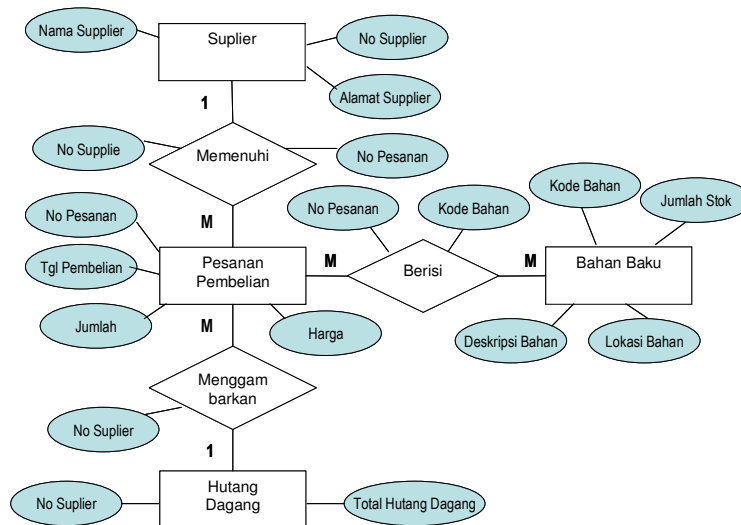
Kemudian digunakan kamus data untuk menampilkan rincian yang hilang dari diagram arus data. Kamus data mendefinisikan arus data dan penyimpanan data dan pengertian berbagai nama.

Entry penyimpanan data	
Kegunaan:	Untuk menjelaskan tiap penyimpanan data dalam diagram arus data
Nama penyimpanan data	: Tabel pesanan penjualan
Penjelasan	: Tabel yang berisi data historis pemesanan penjualan setelah data dimasukkan kedalam sistem komisi penjualan
Struktur data	: Catatan pemesanan penjualan
Volume data	: Rata-rata 140 per hari
Akses	: Pegawai departemen penjualan
Catatan	:

Gambar 11 Contoh Kamus Data

Model diagram status transisi digunakan untuk menggambarkan pengendalian proses atau waktu dari eksekusi fungsi dan akses data melalui *event*.

Diagram lain yang digunakan dalam SADT adalah diagram hubungan entitas (*Entity-relationship Diagram*), diagram ini menunjukkan bagaimana hubungan diantara pihak-pihak yang terlibat dalam dan data apa saja dari entitas itu yang digunakan oleh sistem informasi.

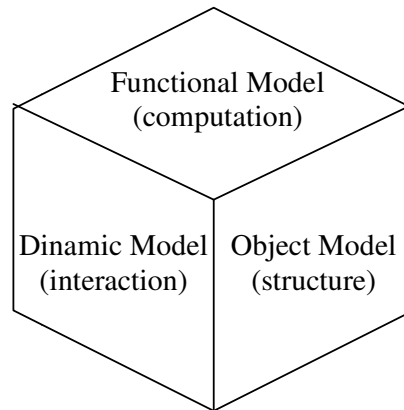


Gambar 13 Contoh Diagram Hubungan Entitas (ERD)

### 6.6.2 Metodologi Object Modeling Technique (OMT)

OMT adalah pendekatan utilisasi model berorientasi objek. OMT menekankan pada kepentingan model dan penggunaannya untuk mencapai suatu abstraksi. Pendekatan ini membedakan antara tahap analisis yang berfokus pada dunia nyata dan tahap desain yang umumnya untuk ditujukan merinci sumber daya komputer. OMT model dibangun diantara deskripsi struktur data, perilaku, dan batasan-batasan. OMT ditujukan untuk sistem pemrosesan transaksi (volume besar, pemrosesan sederhana) dan sistem pendukung keputusan (beragam *query* yang tidak dapat diprediksi).

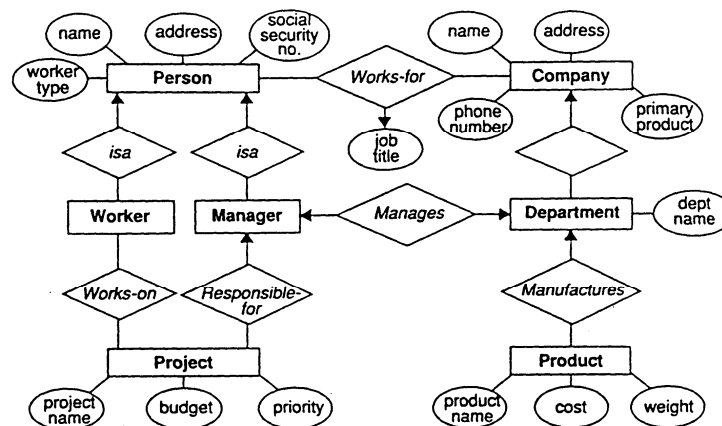
SADT dan OMT memiliki banyak persamaan. Keduanya menggunakan model bangunan yang sama dan mendukung tiga sudut pandang *orthogonal* dari system. Perbedaan antara SADT dan OMT hanya pada gaya dan penekanannya. Dalam pendekatan SADT model fungsional lebih mendominasi, kemudian model dinamis menjadi yang penting berikutnya, dan model objek menjadi yang paling sedikit penting. Sedangkan pada OMT model objek dianggap yang paling penting, kemudian model dinamis, dan terakhir model fungsional..



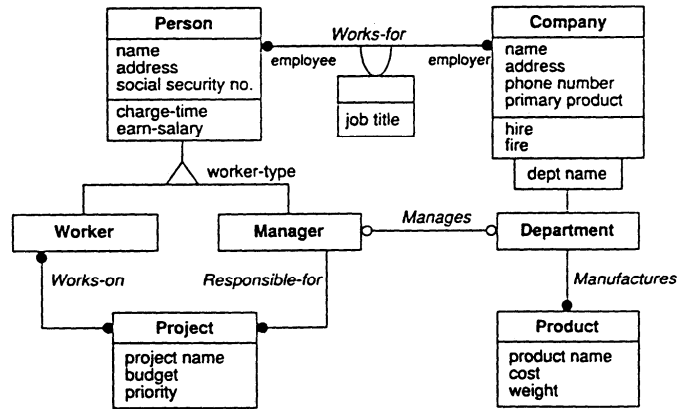
Gambar 3.5 Tiga Sudut Pandang *Orthogonal* Sistem

SADT mengorganisasi sistem di sekeliling prosedur. Sebaliknya OMT mengorganisasi sistem di sekeliling objek dunia nyata, atau objek konseptual yang ada dalam pandangan pengguna. Perubahan dalam prosedur dapat menyebabkan bencana bagi desain berbasis prosedur. Sebaliknya perubahan tersebut siap diakomodasi dalam desain berbasis objek melalui penambahan atau perubahan operasi, dan meninggalkan struktur objek tidak berubah. SADT berguna untuk masalah yang fungsi-fungsinya lebih kompleks dari data.

Desain SADT memiliki batas sistem yang didefinisikan dengan jelas, melalui bagaimana prosedur *software* berkomunikasi dengan dunia nyata. Struktur desain SADT menjadi sulit untuk ditingkatkan dengan batasan yang baru. Lebih mudah meningkatkan desain berorientasi objek yang tinggal menambahkan objek dan relasinya saja. Desain berorientasi-objek lebih lentur terhadap perubahan.



(a)



(b)

Gambar 3.6 Perbandingan ERD (a) dan OMT Model (b)

Dalam perkembangannya, OMT menggunakan banyak diagram yang dikenal dengan nama Unified Modeling Language-UML yang merupakan bahasa standar yang dikontrol oleh konsorsium dari banyak perusahaan yang dinamakan Object Management Group (OMG). Jenis Diagram UML adalah

Nama Diagram	Kegunaan	Generasi UML
Activity	Behavioral prosedural dan paralel	1
Class	Class, Fitur, dan Relasi	1
Communication	Interaksi antar objek, penekanan jalur	1
Component	Struktur dan koneksi komponen	1
Composite structure	Dekomposisi runtime suatu class	2
Deployment	Pemindahan artefak ke node	1
Interaction overview	Campuran sequence dan activity	2
Object	Konfigurasi	1
Package	Struktur hierarki	1
Sequence	Interaksi antar objek; penekanan pada sequence	1
State machine	Bagaimana event mengubah objek	1
Timing	Interaksi antar objek; penekanan pada waktu	2
Use case	Bagaimana pengguna berinteraksi dengan sistem	1