

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Membicarakan tentang sistem sebenarnya bukan hal yang asing lagi bagi kita. Dalam kehidupan sehari-hari banyak contoh dari sistem yang kita temukan, contohnya sistem komputer, sistem pembayaran, sistem penjualan, sistem pembelian dan sebagainya.

Setiap sistem terdiri dari bagian-bagian yang lebih kecil yang disebut dengan subsistem. Dalam penulisan laporan Tugas Akhir ini, penulis mengambil salah satu contoh subsistem dari sistem akuntansi yaitu sistem penjualan.



2.1.1. Pengertian Sistem

Menurut Sutabri (2012:10) “sedara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling bergantung satu sama lain, dan terpadu”.

Menurut Gordon B. Davis dalam Sutabri (2012:12) “menyatakan, sistem bisa berupa abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling bergantung. Sedangkan sistem yang bersifat fisis adalah serangkaian unsur yang bekerjasama untuk mencapai suatu tujuan”.

2.1.2. Karakteristik Sistem

Menurut Tata Sutabri (2012:13) “Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem”. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batas Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.



3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan juga dapat merugikan sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari suatu sub sistem ke sub sistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk suatu kesatuan.

5. Masukan Sistem (*Input*)

Energy yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh didalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah *signal input* yang akan diolah menjadi informasi.

6. Keluaran Sistem (*output*)

Hasil dari energy yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Seperti contoh sistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolah Sistem (*procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengelola data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.



8. Sasaran Sistem (*objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.3. Klasifikasi Sistem

Menurut Sutabri (2012:15) “sistem dapat diklasifikasikan dari beberapa sudut pandangan” diantaranya sebagai berikut:

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sedangkan, sistem buatan manusia adalah sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machinsystem*.

3. Sistem *deterministic* dan sistem *probalistik*

Sistem yang beroperasi dengan tingkah laku yang dapat prediksi disebut sistem *deterministic*. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat *probalistik* adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur *probabilitas*.



4. Sistem terbuka dan sistem tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh sistem lingkungan luar nya. Sistem ini bekerja secara otomatis tanpa ada campuran tangan dari pihak luar. Sedangkan sistem terbuka adalah sistem yang berhubungan dan dapat dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

2.1.4. Pengertian Informasi

McLeod dalam bukunya Yakub yang berjudul Pengantar Sistem Informasi (2012:9) mendefinisikan bahwa, “Informasi adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya”. Informasi juga disebut data yang diproses atau data yang memiliki arti. Informasi merupakan data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan informasi tersebut. Para pembuat keputusan memahami bahwa informasi menjadi faktor kritis dalam menentukan kesuksesan atau kegagalan dalam suatu bidang usaha.



2.1.5. Pengertian Sistem Informasi

Menurut O'Brian dalam Yakub (2012:17) “sistem informasi (*informasi system*) merupakan kombinasi teratur dari orang-orang, perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

Orang tergantung pada sistem informasi untuk berkomunikasi antara satu dengan yang lain dengan menggunakan berbagai jenis alat fisik, perintah dan prosedur pemrosesan informasi, saluran telekomunikasi atau jaringan, dan data yang disimpan atau sumber daya data.

2.1.6. Pengertian Sistem Informasi Manajemen

Menurut McLeod dan Schell (2008:40) “Sistem Informasi manajemen merupakan suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai kebutuhan yang serupa”. Informasi tersebut tersedia dalam bentuk laporan periodik, laporan khusus, dan output dari simulasi informasi yang ada. Para pemakai biasanya membentuk suatu entitas organisasi formal instansi atau sub-unit di bawahnya. Informasi digunakan oleh pengelola maupun staf lainnya pada saat mereka membuat keputusan untuk memecahkan masalah.



2.1.7. Pengertian Akuntansi

Menurut Pura (2013:4) “Akuntansi adalah Seperangkat pengetahuan yang mempelajari perekayasaan dalam penyediaan jasa, yang berupa informasi keuangan kuantitatif dari suatu unit organisasi dan cara penyampaian (pelaporan) informasi tersebut kepada pihak yang berkepentingan untuk dijadikan dasar pengambilan keputusan ekonomi”.

Sedangkan menurut Kieso (2008:4) “akuntansi adalah suatu sistem informasi yang mengidentifikasi, mencatat dan mengkomunikasikan peristiwa-peristiwa ekonomi dari suatu organisasi kepada para pengguna yang berkepentingan”.

2.1.8. Pengertian Sistem Informasi Akutansi

Menurut Gelinas dan Dull (2012:13) “Sistem informasi akuntansi merupakan sistem yang digunakan untuk mengumpulkan, mengolah, dan melaporkan informasi terkait dengan aspek keuangan akan kegiatan bisnis”.

Menurut Romney dan Steinbart (2015:10) “Sistem informasi akuntansi adalah sistem yang digunakan untuk mengumpulkan, mencatat, menyimpan dan mengolah data untuk menghasilkan suatu informasi untuk pengambilan keputusan”. Sistem ini meliputi orang, prosedur dan instruksi data perangkat lunak, infrastruktur teknologi informasi serta pengendalian internal dan ukuran keamanan.



2.1.9. Pengertian Penjualan

Menurut Leny Sulistiyowati (2010:270) “mengartikan penjualan adalah Pendapatan yang berasal dari penjualan produk perusahaan, disajikan setelah dikurangi potongan penjualan dan retur penjualan”. Aktivitas penjualan dalam perusahaan dapat dilakukan baik secara tunai ataupun kredit. Penjualan tunai merupakan penjualan yang dilakukan dengan cara menerima uang tunai/cash pada saat barang diserahkan pada pembeli.

Berikut ini jurnal transaksi penjualan tunai pada PT.Langgeng Multi Jaya Jakarta:

Pada Transaksi Penjualan Tunai :

Kas	Rp.XXX	-
Penjualan	-	Rp.XXX

Pada Transaksi Retur Penjualan dan Pengurangan Harga :

Retur Penjualan dan Pengurangan Harga	Rp.XXX	-
Kas/Piutang	-	Rp.XXX

2.2. Peralatan Pendukung (Tools System)

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukan secara tepat arti dan fungsinya. Adapun peralatan pendukung (*tools system*) yang dijelaskan sebagai model sistem yang akan dirancang adalah sebagai berikut :



2.2.1. UML (Unified Modelling Language)

Menurut Whitten dan Bentley (2009: 371) “UML adalah sekumpulan ketentuan pemodelan yang digunakan untuk menetapkan atau menggambarkan sistem *software* dalam terminologi objek.”

Berdasarkan *International Journal of Advanced Research in Computer Science and Software Engineering* (Vol.2, No.7, Juli 2012: 241) “UML adalah satu pemodelan standar yang dapat digunakan untuk melaksanakan tugas yaitu

menganalisa, memvisualisasikan, membangun dan mendokumentasikan perubahan dalam proses bisnis dan *software*.”

Berdasarkan *International Journal of Computer Applications* (Vol.1, No.1, September 2012: 17) “UML adalah kelompok notasi grafis, didukung oleh *meta-model* tunggal yang membantu dalam menggambarkan desain sistem *software*, terutama sistem perangkat lunak yang dibangun dengan menggunakan gaya berorientasi objek.”

Berdasarkan pendapat ahli diatas, maka dapat disimpulkan bahwa UML merupakan bahasa pemodelan yang digunakan untuk menganalisa, menggambarkan, membangun, dan mendokumentasikan proses bisnis dan *software* yang berkaitan dengan objek.



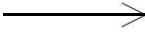
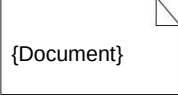
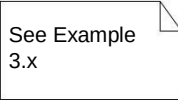
2.2.2. Activity Diagram

Menurut Jones dan Rama (2008: 79), yang diterjemahkan oleh M. Slamet Wibowo, “*Activity diagram* adalah diagram yang memainkan peran seperti sebuah peta dalam memahami proses bisnis dengan menunjukkan urutan aktivitas di dalam proses”.

2.2.3 Notasi Activity Diagram

Menurut Jones dan Rama (2008: 82-84), terdapat beberapa notasi dalam *activity diagram*:

Tabel II.1 Notasi Activity Diagram

Bentuk Notasi	Nama	Keterangan
	<i>Initial state</i>	Menunjukkan titik mulainya sebuah proses dalam <i>activity diagram</i> .
	<i>State</i>	Menunjukkan kejadian, aktivitas, atau <i>trigger</i> .
	<i>Control flow</i>	Menunjukkan urutan dari satu kejadian atau aktivitas ke kejadian atau aktivitas berikutnya.
	<i>Object flow</i>	Menunjukkan aliran informasi antar kejadian.
	<i>Document</i>	Menunjukkan dokumen sumber atau laporan.
	<i>Decision</i>	Menunjukkan percabangan.
	<i>Table</i>	Menunjukkan <i>file</i> komputer di mana data dibaca atau dicatat selama kejadian bisnis terjadi.
	<i>Note</i>	Menunjukkan referensi bagi pembaca ke diagram atau dokumen lain untuk informasi yang lebih rinci.

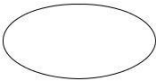
	<i>Final state</i>	Menunjukkan titik akhir dari sebuah proses.
	<i>Swimlane</i>	Menunjukkan pemisahan aktivitas atau kejadian menurut orang atau departemen yang bertanggung jawab atas kejadian atau aktivitas tertentu.

Sumber : Jones dan Rama (2008: 82-84)

2.2.4. Use case Diagram

Menurut Satzinger (2012: 78) “Use case diagram adalah model UML yang digunakan untuk menunjukkan use case dan hubungannya dengan aktor secara grafis.”

Tabel II.2 Use Case Diagram

Notasi	Nama	Keterangan
	<i>Use case</i>	Menunjukkan aktivitas yang dilakukan oleh sistem, biasanya sebagai respon terhadap <i>request</i> dari <i>user</i> terhadap sistem untuk mencapai tujuan <i>user</i> .
	<i>Actor</i>	Menunjukkan orang yang secara aktual berinteraksi dengan sistem komputer.

	<i>Connecting line</i>	Menghubungkan antara aktor dan <i>use case</i> dan mengindikasikan <i>actor</i> tertentu yang berhubungan <i>use case</i> tertentu.
	<i>Include</i>	Menggambarkan bahwa jalannya suatu usecase memicu jalannya usecase lain. Misalnya usecase login diinclude oleh usecase memilih menu, artinya usecase memilih menu akan memicu dijalankannya usecase login
	<i>Extend</i>	menggambarkan bahwa suatu usecase dijalankan karena ada persyaratan tertentu dari usecase lain. Misal, dalam sebuah sistem user tidak bisa menjalankan login sebelum dia mendaftar akun.

Sumber : Satzinger (2009: 242-244)

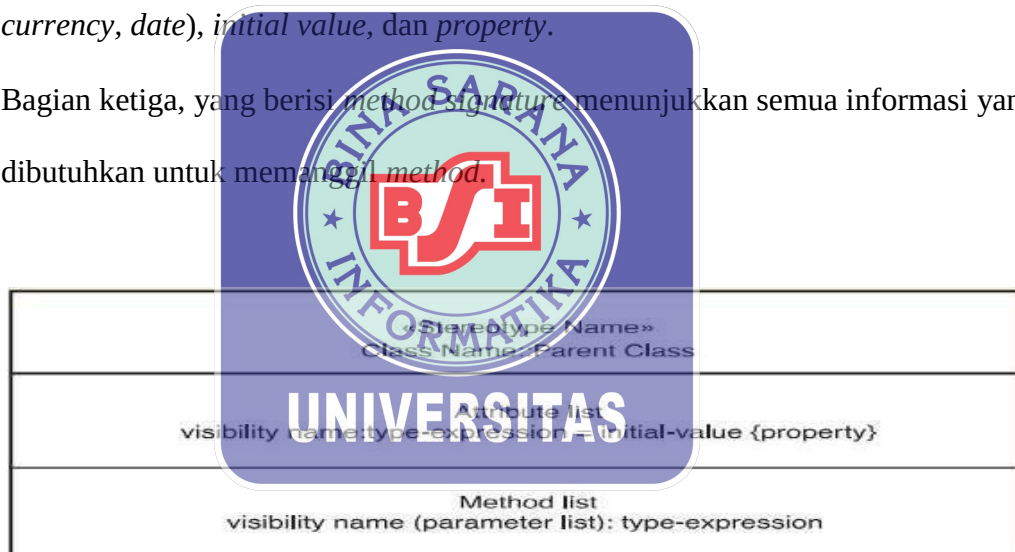
2.2.5. *Class Diagram*

Menurut Satzinger (2012: 101) “*Class diagram* adalah diagram yang terdiri dari kelas-kelas (contohnya kumpulan objek) dan asosiasi di antara kelas”.

2.2.6. Notasi Class Diagram

Menurut Satzinger (2009: 187) “sebuah *domain class* disimbolkan dengan menggunakan bentuk persegi empat yang dibagi menjadi tiga bagian”, antara lain:

1. Bagian pertama, yang berisi nama *class* dan informasi *stereotype*.
2. Bagian kedua, yang berisi nama atribut yang terdiri dari *attribute visibility* (menunjukkan kemampuan objek yang dapat secara langsung mengakses atribut; tanda + berarti atribut *visible* sedangkan tanda – berarti atribut tidak *visible*), nama atribut, *type expression* (seperti *character*, *string*, *integer*, *number*, *currency*, *date*), *initial value*, dan *property*.
3. Bagian ketiga, yang berisi *method signature* menunjukkan semua informasi yang dibutuhkan untuk memanggil *method*.



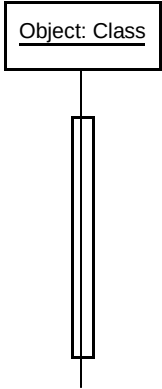
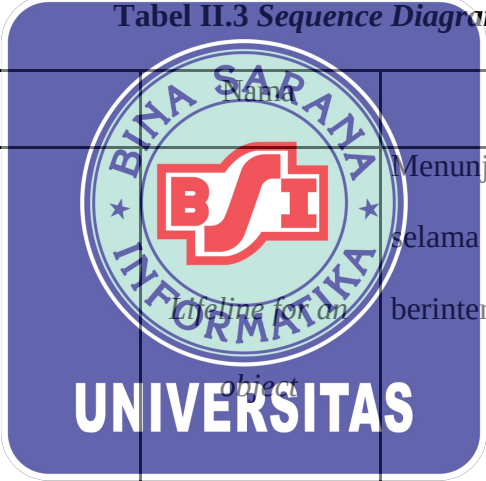
Sumber : Satzinger (2009: 411)

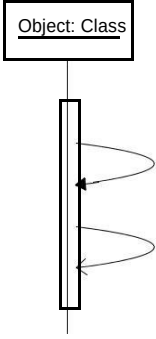
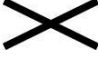
2.2.7. Package Diagram

Menurut Rosa dan Shalahudin (2013:153) mengatakan bahwa “ *Package Diagram* menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML”.

2.2.8. Sequence Diagram

Menurut Mathiasen (2008: 340) “*Sequence diagram* menjelaskan tentang interaksi di antara beberapa objek dalam jangka waktu tertentu. *Sequence diagram* melengkapi *class diagram*, yang menjelaskan situasi umum dan statis. Sebuah *sequence diagram* dapat mengumpulkan situasi kompleks dan dinamis melibatkan beberapa dari kebanyakan *object* yang digeneralisasikan dari *class* pada *class diagram*”.

Tabel II.3 Sequence Diagram		
Notasi	Nama	Keterangan
		Menunjukkan garis hidup selama objek tersebut masih berinteraksi.

	<i>Recursive call & return</i>	Menunjukkan urutan interaksi yang dilakukan berulang-ulang.
	<i>Message in the form of event</i>	Menunjukkan pesan yang dikirim dalam <i>event</i> .
	<i>Procedure call</i>	Menunjukkan urutan interaksi yang dilakukan dengan konsisi tertentu.
	<i>Return</i>	Menunjukkan respon pesan terhadap <i>message</i> yang dikirimkan.
	<i>Destruction of an object</i>	Apabila objek di <i>destroy</i> .

Sumber : Mathiassen (2008: 340)

2.2.9. Component Diagram

Menurut Rosa dan Shalahudin(2013:148) mengatakan bahwa “ *component diagram* di buat untuk menunjukan organisasi dan ketergantungan diantara komponen dalam sebuah sistem “.

2.2.10. *Deployment Diagram*

Menurut Rosa dan Shalahudin (2013:154) mengatakan bahwa “ *Diagram Deployment* menunjukan konfigurasi komponen dalam proses eksekusi aplikasi”. Ada 4 (empat) macam *relationship* dalam *Unified Modeling Language* (UML) yaitu:

1. Pengklasifikasian (*Classifier*) pada prinsipnya merupakan konsep diskret dalam model yang memiliki identitas (*identity*), state, perilaku (*behavior*), serta relasi dengan mengklasifikasi yang lainnya (*relationship*).
2. Asosiasi (*asociation*) pada dasarnya mendeskripsikan koneksi diskret antara objek atau antar instance lain dalam sistem atau perangkat lunak yang sedang dikembangkan.
3. Generalisasi menggambarkan hubungan antara *use case* yang bersifat umum dengan *use case-use case* yang bersifat lebih spesifik.
4. Realisasi Relasi (*realization*) menghubungkan elemen-elemen model, misalnya kelas, ke elemen-elemen model lainnya, seperti suatu antarmuka, yang menyediakan spesifikasi perilaku tetapi bukan strukturnya atau implementasinya.



2.2.11. *Data dan Database*

Menurut Mc Fadden dalam Nugroho (2011:5), mendefinisikan data adalah “fakta-fakta tentang segala sesuatu didunia nyata yang dapat direkam dan disimpan pada media komputer”

Menurut Nugroho (2011:5) mendefinisikan *database* adalah “sebagai kumpulan terorganisasi dari data-data yang berubungan sedemikian rupa sehingga mudah disimpan, dimanipulasi, serta dipanggil penggunaanya”

2.2.12. *Logical Record Structure (LRS)*

Menurut Hasugian dan Shidiq (2012:608) memberikan batasan bahwa *Logical Record Structure (LRS)* adalah “Sebuah model sistem yang digambarkan dengan sebuah *diagram-ER* akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konvensi ke LRS”. Menentukan kardinalitas, jumlah tabel dan *Foreign Key* sebagai berikut:

1. *One-to-one*

2. *One-to-many*

Satu entitas dapat berhubungan dengan lebih dari satu entitas lain.

3. *Many-to-many*

Beberapa entitas dapat berhubungan dengan beberapa entitas lain.



2.2.13. *Entity Relationship Diagram (ERD)*

Menurut Sutanta (2011:91) “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek.” *Entity Relationship Diagram (ERD)* digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. Ada beberapa simbol dari ERD, diantaranya :

1. *Entitas*

Kumpulan dari beberapa *atribut/field*.

2. *Relasi*

Suatu hubungan yang terjalin diantara beberapa *entitas*.

3. *Atribut*

Notasi yang memberikan informasi mengenai jenis entitas dan jugarelasinya.

4. Garis Penghubung

Notasi yang digunakan untuk merangkai beberapa *notasi* yang digunakan seperti : *entitas*, *relasi* dan *atribut*.

2.2.14. Desain User Interface

User Interface merupakan suatu system yang membantu pengguna (*user*) untuk berkomunikasi dengan sistem komputer dan sistem aplikasi. Desain *User Interface* untuk tipe penggna ada dua yaitu: ahli dan pemula. Umumnya desain *user interface* saat ini berasumsi pemakai adalah pemula yang sedang dalam proses menjadi ahli.

Desain-desain antarmuka perlu memperhatikan:

1. Faktor pemakai
2. Faktor *human engineering*
3. Dialog dan istilah



Permasalahan-permasalahan yang sering timbul dalam Desain *User Interface*:

1. Penggunaan jargon atau singkatan-singkatan komputer yang berlebihan
2. Desain yang tidak jelas
3. Ketidakmampuan membedakan tindakan-tindakan alternative yang harus dipilih pemakai
4. Pendekatan *problem-solving* yang tidak konsisten
5. Desain yang tidak konsisten

Dari permasalahan-permasalahan yang diatas ada beberapa cara untuk pemecahan masalah desain user interface:

1. Pahami *user* dan tugas mereka
2. Lihatkan *user* dalam desain antarmuka
3. Uji sistem dengan melibatkan user
4. Lakukan proses desain secara

iterative Syarat-syarat desain *user*

interface:

1. Layar harus di format agar berbagai info, perintah, pesan, muncul di area layar yang konsisten
2. Pesan, perintah, informasi jangan terlalu panjang
3. Jangan terlalu sering memakai atribut display yang mengganggu seperti *blinking, highlight* dan sebagainya
4. Nilai default di buat jelas
5. Antipasti eror yang mungkin di buat user
6. Jika ada eror, user mestinya tidak boleh melanjutkan tanpa memperbaiki eror tersebut
7. Jika user melakukan sesuatu yang memebahayakan sistem, keyboard harus terkunci dan pesan untuk meminta bantuan teknisi harus di munculkan

