

BAB II

TINJAUAN PUSTAKA

Pada bagian ini akan dibahas mengenai keputusan, DSS sebagai konsep dasar CDSS, panduan klinis sebagai basis pengetahuan keputusan klinis, GASTON *framework* untuk mengembangkan panduan klinis berbasis komputer yang memfasilitasi CDSS, konsep adaptabilitas sistem, dan beberapa teori penting mengenai metode perancangan interaksi, konsep *usability*, dan konsep *state transition*.

II.1 Keputusan

Keputusan tidak dapat dipisahkan dari setiap aktivitas yang dilakukan manusia sepanjang hidupnya. Menurut Zeez Maoz, definisi keputusan adalah:

“A process where in an organism (individual, group, organization, state) selects a given course of action among several alternatives in order to solve a specific problem or set of problems.” [MAO07]

Suatu keputusan bukan merupakan aktivitas tunggal, melainkan serangkaian proses untuk memilih salah satu dari berbagai alternatif solusi. Menurut Hansson, terdapat empat fase pembuatan keputusan [HAN94] yang dikutip dari Simon, sebagai berikut:

1. **Intelligence:** mengidentifikasi, mengklarifikasi, dan memperjelas isu permasalahan yang membutuhkan suatu keputusan,
2. **Design:** mencari, mengembangkan, atau memodifikasi alternatif solusi keputusan,
3. **Choice:** mengevaluasi dan memilih alternatif solusi yang dihasilkan pada fase sebelumnya, dan
4. **Implementation:** menyetujui dan mengaktualisasikan opsi solusi yang dipilih sebagai suatu bentuk keputusan.

II.2 Sistem Pendukung Keputusan

Seiring perkembangan zaman, manusia dituntut membuat berbagai keputusan yang tepat dalam menghadapi permasalahan yang semakin kompleks. Oleh karena itu, dikembangkan DSS (*Decision Support System*) untuk dukungan pembuatan keputusan. Pada awalnya, DSS dimanfaatkan pada level manajerial. Namun, karena pertumbuhan pengetahuan dan kebutuhan dukungan keputusan yang tidak berasal dari dunia manajerial saja, DSS tumbuh sebagai konsep yang dapat digunakan pada berbagai bidang.

II.2.1 Definisi dan Klasifikasi DSS

Definisi DSS terus-menerus mengalami pergeseran mulai dari persepsi mengenai hal apa saja yang dapat dilakukan oleh suatu DSS hingga ide mengenai bagaimana DSS dapat mencapai objektif yang diinginkan [TUR05]. Secara umum, Finlay mendefinisikan DSS sebagai sistem berbasis komputer untuk mendukung pengambilan keputusan [FIN94]. Sedangkan secara spesifik, Turban dan Aronson menyimpulkan definisi DSS sebagai berikut:

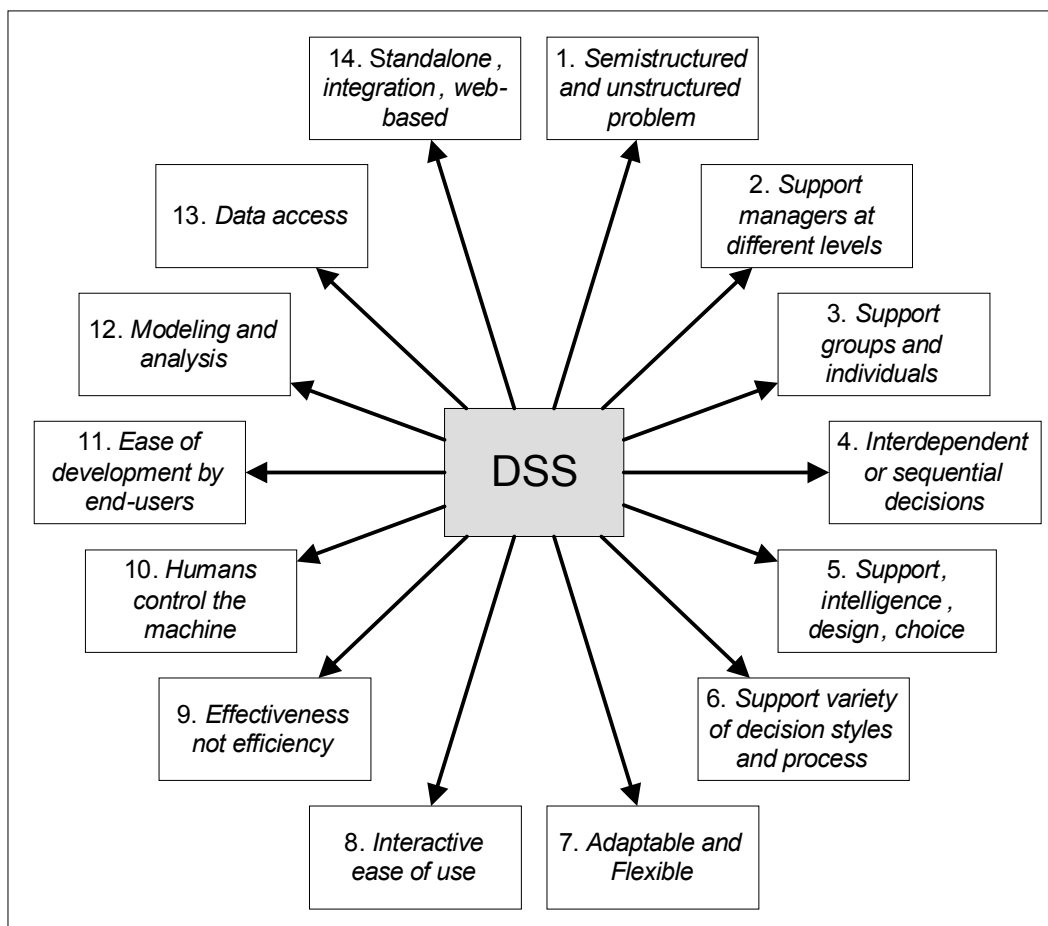
“A DSS is an approach (or methodology) for supporting decision-making. It uses an interactive, flexible, adaptable CBIS (Computer Based Information System) especially developed for supporting the solution to a specific nonstructured management problem. It uses data, provides an easy user interface, and can incorporate the decision maker’s own insight.” [TUR05]

Seperti juga dalam hal pendefinisian, tidak ada klasifikasi DSS yang berlaku universal. Menurut Turban seperti yang dikutip dari Holsapple dan Whinston, DSS dapat diklasifikasikan berdasarkan strukturnya (*framework*) [TUR05], sebagai berikut:

1. ***Text-oriented DSS***: menggunakan pengelolaan, perolehan, dan manipulasi informasi tidak terstruktur dalam format dokumen elektronik untuk dukungan keputusan.
2. ***Database-oriented DSS***: menggunakan pengorganisasian struktur basis data yang umumnya dengan konfigurasi relasional dengan kapabilitas *query* dan *report generation* untuk dukungan keputusan.
3. ***Spreadsheet-oriented DSS***: merupakan sistem pemodelan yang menggunakan perhitungan kompleks, statistik, simulasi, atau model optimisasi untuk membantu pembuat keputusan menganalisis situasi.
4. ***Solver-oriented DSS***: menggunakan struktur algoritma atau prosedur dengan suatu program komputer yang dapat dieksekusi untuk dukungan keputusan.
5. ***Rule-oriented DSS***: menyediakan keahlian penyelesaian persoalan (sistem pakar) yang disimpan dalam bentuk fakta, aturan, prosedur, atau struktur tertentu.
6. ***Compound-oriented DSS***: DSS hibrida yang mengkombinasikan dua atau lebih dari lima klasifikasi DSS yang telah diuraikan.

II.2.2 Karakteristik dan Kapabilitas DSS

Permasalahan akibat berbagai interpretasi DSS adalah sulitnya dalam pemanfaatan istilah DSS. Oleh karena itu, Turban mendefinisikan karakteristik dan kapabilitas DSS seperti yang ditunjukkan pada Gambar II-1 sebagai berikut [TUR05]:



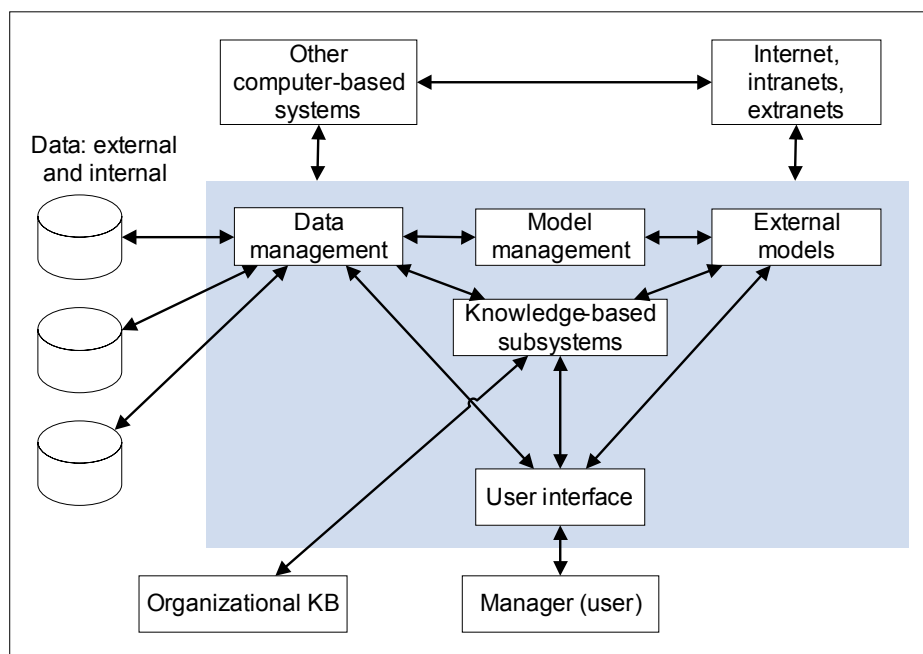
Gambar II-1 Karakteristik dan Kapabilitas DSS [TUR05]

1. DSS mendukung permasalahan semi-terstruktur (permasalahan yang rutin berulang, tetapi masih dibutuhkan *human judgement* dalam penetapan solusinya) atau tidak terstruktur (permasalahan yang belum jelas dan kompleks sehingga tidak ada solusi yang langsung dapat digunakan).
2. DSS mendukung keputusan bagi berbagai lapisan manajer.
3. DSS mendukung keputusan bagi kelompok maupun individu.
4. DSS mendukung keputusan yang bersifat interdependen dan/atau runtun.
5. DSS mendukung seluruh fase pembuatan keputusan seperti yang telah dibahas pada sub-bab II.1, yaitu: *intelligence, design, choice, dan implementation*.
6. DSS mendukung berbagai cara dan gaya pembuatan keputusan.

7. DSS bersifat fleksibel (pengguna dapat menambahkan, menghapus, dan mengubah elemen dasar yang dikelola DSS) dan dapat diadaptasikan (pengguna mengadaptasikan sistem agar dapat menghadapi kondisi yang cepat berubah).
8. DSS bersifat *user friendly* sehingga mudah diadaptasi oleh pengguna yang belum berpengalaman dengan pemanfaatan komputer.
9. Tujuan pemanfaatan DSS adalah meningkatkan efektivitas pembuatan keputusan (waktu dan kualitas), bukan pada efisiensi (meminimalisasi biaya).
10. DSS dimanfaatkan untuk mendukung pembuat keputusan, bukan menggantikannya.
11. DSS harus mudah dikonfigurasi, fleksibel dalam penggunaannya, dan mudah dimodifikasi untuk memenuhi berbagai kebutuhan setiap pembuat keputusan.
12. DSS dapat menggunakan pemodelan untuk analisis situasi dan permasalahan yang membutuhkan keputusan. Kapabilitas pemodelan memungkinkan pengguna mencoba berbagai strategi aksi pada sirkumstansi dan konfigurasi yang berbeda.
13. Memungkinkan akses berbagai bentuk sumber data (format, tipe, dan lain-lain).
14. DSS dapat dikembangkan sebagai kakas *standalone*, diintegrasikan dengan aplikasi lain, dan didistribusikan melalui jaringan atau teknologi Internet.

II.2.3 Komponen DSS

Jika dilihat dari sudut pandang komponen, DSS dapat dibagi menjadi tiga buah subsistem utama seperti yang ditunjukkan pada Gambar II-2, antara lain [SPR89, TUR05]:



Gambar II-2 Komponen Sistem Pendukung Keputusan [TUR05]

1. Subsistem Data

Subsistem data terdiri dari basis data yang relevan dengan lingkungan sistem dan dikelola oleh sebuah sistem perangkat lunak DBMS (*Database Management System*). Sumber data DSS dapat berasal dari internal maupun eksternal.

2. Subsistem Model

DSS memiliki kemampuan mengintegrasikan akses data dan model dengan cara menanamkannya pada sistem informasi dengan menggunakan *database* sebagai mekanisme integrasi dan komunikasi antar model. Subsistem ini seringkali disebut sebagai MBMS (*Model Base Management System*). Proses kreasi model harus bersifat fleksibel dengan bahasa pemodelan yang kuat dan sekumpulan *building blocks* yang dapat dibentuk pada proses pemodelan.

3. Subsistem Antarmuka Pengguna

Karakteristik fleksibilitas dan *usability* dari DSS ditentukan oleh kapabilitas antarmuka sistem. Pengguna berinteraksi dan memberikan perintah-perintah kepada DSS melalui subsistem ini yang dikenal sebagai DGMS (*Dialogue Generation and Management System*). Subsistem menyediakan struktur antarmuka grafis yang konsisten dan mudah diadaptasi oleh pengguna.

Pada pengembangannya, DSS juga dapat memiliki subsistem pengetahuan [TUR05]. Subsistem ini digunakan untuk mendukung pengelolaan subsistem DSS lainnya atau berfungsi sebagai komponen independen dalam menyediakan kemampuan kecerdasan tambahan bagi pembuat keputusan dengan menggunakan sistem pakar atau sistem inteligensi lainnya. DSS dengan kemampuan kecerdasan seringkali dikenal dengan *intelligent DSS* [TUR05].

II.3 Sistem Pendukung Keputusan Klinis

Sistem pendukung keputusan klinis (CDSS atau *Clinical Decision Support System*) muncul pertama kali pada sekitar tahun 1980 sebagai reaksi perkembangan pengetahuan klinis. CDSS merupakan bentuk pengaplikasian DSS dalam dunia kesehatan yang berfungsi menghubungkan observasi atau tindakan medis yang dilakukan oleh petugas medis dengan suatu basis pengetahuan untuk meningkatkan kualitas pelayanan kesehatan.

II.3.1 Definisi CDSS

Agar lebih memahami dan dapat mengadopsi terminologi informatika medis ini, berikut adalah definisi CDSS menurut Hritovski:

“A medical decision-support system is any computer program designed to help health professionals make clinical decisions. In a sense, any computer system that deals with clinical data or medical knowledge is intended to provide decision support.” [HRI96]

CDSS merupakan sistem perangkat lunak dalam konteks dunia kesehatan untuk membimbing pasien, perawat, atau penyedia layanan kesehatan dalam hal pengambilan keputusan klinis. Keputusan klinis tersebut dapat direpresentasikan dalam bentuk peringatan, saran, atau interpretasi spesifik berdasarkan status klinis pasien [CAL06]. Sistem ini menyediakan pengetahuan klinis berbasis elektronik [BEI05] yang dibutuhkan, tetapi tidak dapat mengaplikasikan informasi tersebut sebagai *task* keputusan tertentu terhadap pasien spesifik.

II.3.2 Latar Belakang CDSS

CDSS dikembangkan untuk mengatasi permasalahan-permasalahan berikut [MIN07]:

1. Keterbatasan daya ingat manusia dalam pemanfaatan pengetahuan klinis untuk diagnosa dan perawatan pasien,
2. Terus bertambahnya pengetahuan klinis seiring dengan penelitian bidang kesehatan,
3. Bertambahnya ragam gejala (*symptoms*) untuk diobservasi dalam diagnosa penyakit,
4. Bertumbuhnya ragam jenis penyakit,
5. Terbatasnya pengetahuan spesialis terhadap domain spesialisasi lain.

Selain hal-hal di atas, CDSS memiliki peran yang semakin penting karena ketidakseimbangan antara sumber daya terbatas dengan permintaan kualitas pelayanan kesehatan yang terus meningkat. CDSS dapat membantu para ahli kesehatan untuk mengorganisasikan, menyimpan, dan mengaplikasikan pengetahuan klinis; dan juga meningkatkan kualitas pelayanan kesehatan dengan menyediakan diagnosa dan perawatan kesehatan yang lebih akurat, efektif, dan handal; serta menghindari kemungkinan terjadinya kesalahan akibat kurangnya pengetahuan dari ahli kesehatan.

II.4 Panduan Klinis

Pada praktik perawatan kesehatan, telah terjadi perubahan *trend* menuju pelayanan berbasis bukti klinis terbaik. Panduan klinis menyediakan kerangka kerja bagi profesional kesehatan untuk menawarkan standar perawatan yang setinggi mungkin kepada pasien. Meskipun saat ini dampaknya masih kurang signifikan, pemanfaatan panduan klinis terus dirasakan semakin penting dari tahun ke tahun.

II.4.1 Definisi Panduan Klinis

Panduan klinis mendefinisikan pertanyaan-pertanyaan penting terkait praktik klinis dan juga mengidentifikasi seluruh kemungkinan keputusan beserta hasil dari setiap keputusan yang diambil. Menurut IOM (*Institute of Medicine*), definisi panduan klinis adalah:

“Clinical practice guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances.” [FIE90]

Meskipun tidak dituliskan secara eksplisit dalam definisinya, dapat dinyatakan bahwa panduan klinis merupakan suatu basis pengetahuan berdasarkan bukti klinis terbaik yang digunakan oleh petugas medis dalam pengambilan keputusan klinis yang sesuai dengan status kesehatan pasien spesifik.

II.4.2 Tujuan Panduan Klinis

Berikut adalah tujuan pemanfaatan panduan klinis pada pelayanan kesehatan [OPE07]:

1. Mendeskripsikan perawatan kesehatan yang tepat berdasarkan bukti sains terbaik,
2. Mengurangi berbagai variasi yang tidak diinginkan pada perawatan kesehatan,
3. Menyediakan referensi atau basis pengetahuan yang lebih rasional,
4. Menyediakan suatu fokus terhadap edukasi yang berkelanjutan,
5. Mengefisiensikan penggunaan sumber daya yang ada dan terbatas,
6. Bertindak sebagai pengendali kualitas, termasuk di dalamnya proses audit, dan
7. Menandakan hal-hal yang masih kurang atau kesalahan-kesalahan untuk pengkajian ulang atau perbaikan.

II.4.3 Masalah Panduan Klinis

Panduan klinis telah menunjukkan kemampuannya untuk mendukung perbaikan kualitas dan konsistensi perawatan kesehatan. Meskipun demikian, masih terdapat banyak permasalahan dalam pemanfaatan panduan klinis yang dihadapi oleh berbagai organisasi kesehatan. Menurut De Clerq [DEC04], pengembangan panduan klinis sangat menghabiskan waktu dan sumber daya. Banyak panduan klinis telah didistribusikan ke berbagai organisasi pelayanan kesehatan, tetapi sulit digunakan karena umumnya berbentuk teks naratif. Berbagai usaha telah dilakukan dalam pengembangan panduan klinis, tetapi petugas medis masih seringkali mengalami kesulitan untuk beradaptasi dengan pemanfaatan panduan klinis sehingga tidak dapat mengaplikasikannya dengan tepat pada perawatan kesehatan pasien aktual.

II.4.4 Panduan Klinis Berbasis Komputer

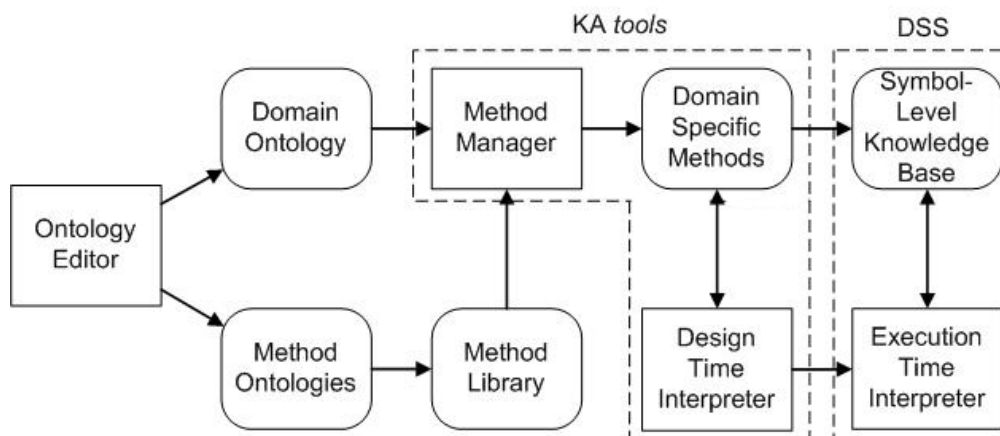
Panduan klinis berbasis komputer mengkodekan rekomendasi berdasarkan bukti klinis terbaik. Pengimplementasian panduan klinis berbasis komputer pada CDSS merupakan strategi jangka panjang dalam meningkatkan pengaplikasiannya pada praktik rutin. Berbeda dengan panduan klinis berbasis dokumen tertulis, sifat natural panduan klinis berbasis komputer dalam bentuk multimedia juga dapat mempermudah pengaplikasian dan konektivitas terhadap basis data perawatan kesehatan pasien [OPE07].

Pengimplementasian panduan klinis berbasis komputer pada CDSS dapat memberikan banyak keuntungan [OPE07], antara lain:

1. Kemudahan akses referensi dan pengetahuan klinis spesifik sesuai kebutuhan,
2. Kemudahan pencarian kesalahan pada isi panduan klinis,
3. peningkatan kejelasan makna panduan klinis,
4. pendeskripsian status pasien yang lebih baik,
5. pemberian saran atau rekomendasi sebagai dukungan keputusan klinis berdasarkan keadaan pasien spesifik.

II.5 GASTON Framework [DEC01, LIM05]

Perancangan model untuk merepresentasikan panduan dan pemilihan metodologi untuk mengkonversikannya ke dalam bentuk elektronik yang dapat dieksekusi komputer membutuhkan usaha yang besar. Oleh karena itu, beberapa grup penelitian informatika kesehatan mengembangkan sebuah *framework* dan standardisasi CDSS untuk meningkatkan pengaplikasian panduan klinis berbasis komputer. GASTON *framework* merupakan kerangka arsitektur generik dalam pengembangan panduan klinis berbasis komputer untuk memfasilitasi CDSS [LIM05].



Gambar II-3 GASTON Framework [DEC01]

Proses pengembangan aplikasi panduan klinis berbasis komputer dengan GASTON *framework* dapat dilihat pada Gambar II-3. Ide utama dari *framework* ini adalah pemanfaatan kakas yang memungkinkan *domain experts* (pakar kesehatan) berinteraksi langsung dengan sistem untuk menyediakan pengetahuan dasar dan terminologi domain spesifik melalui suatu proses akuisisi pengetahuan. KA *Tools* (*Knowledge Acquisition Tools*) digunakan untuk mengekstraksi pengetahuan klinis langsung dari pakar kesehatan sebagai model pengetahuan panduan klinis dari CDSS dalam bentuk kode atau aturan-aturan yang dapat dieksekusi oleh sistem berbasis komputer [VEN08].

Secara umum, GASTON *framework* terdiri dari empat buah tahap utama [DEC01], yaitu:

1. Pendefinisian ontologi

Ontologi adalah spesifikasi formal eksplisit yang mendefinisikan representasi objek, konsep, atau entitas lain pada suatu domain pengetahuan serta relasi antar entitas tersebut [HOW07]. GASTON memanfaatkan ontologi mengenai konsep-konsep yang relevan dengan domain panduan klinis (contoh: penyakit dan obat) dan konsep-konsep penalaran petugas medis pada proses perawatan kesehatan pasien, antara lain:

- a. *Primitives* adalah himpunan blok yang tidak dapat didekomposisi lagi untuk merepresentasikan *clinical tasks*, langkah tunggal (*decision* atau *action*), atau struktur internal PSM (*Problem Solving Method*) panduan klinis [LIM05].
- b. PSM merepresentasikan strategi generik untuk menyelesaikan *tasks* yang independen terhadap domain sistem aplikasi [LIM05].

2. Pengembangan *method library*

Method library merupakan kumpulan PSM dan *primitives* yang dibutuhkan untuk mengembangkan panduan klinis. *Knowledge engineer* menggunakan *method manager* untuk merancang PSM yang spesifik terhadap domain aplikasi.

3. Pengembangan panduan klinis

Pengembangan panduan klinis dilakukan dengan memanfaatkan KA *Tools* (*Knowledge Acquisition Tools*) yang terdiri dari *design time interpreter* untuk memanggil *primitives* dan *domain specific methods* yang dibutuhkan, serta membangkitkan antarmuka untuk memfasilitasi *domain experts* dalam memformulasikan dan menstrukturisasi panduan klinis secara efisien dalam lingkungan visual yang *user-friendly*.

4. Eksekusi panduan klinis

Ketika diinstruksikan, *KA Tools* mengkombinasikan struktur kendali dari tiap panduan klinis dan PSM serta membentuk jaringan *primitives*. Dengan cara ini, *symbol level knowledge base* dibentuk sehingga dapat diproses oleh *execution time engine* yang mengeksekusi modul-modul implementasi setiap *primitive*.

II.6 Adaptabilitas Sistem

Suatu panduan klinis terus mengalami penambahan atau perubahan karena bertumbuhnya pengetahuan dunia penelitian kesehatan atau perkembangan bukti-bukti klinis baru dari praktik perawatan medis di lapangan. Oleh karena itu, CDSS berbasis panduan juga harus dapat mengakomodasi permasalahan tersebut sebagai sistem yang mudah diadaptasikan sesuai kebutuhan perubahan panduan klinis. Dalam konteks sistem HCI, Fischer mengklasifikasikan dua jenis kategori adaptabilitas sistem [FIS01], yaitu:

1. ***Adaptable system***, jika sistem tersebut diubah untuk menyesuaikan terhadap perubahan yang terjadi dengan dukungan substansi dari sistem melalui interaksinya dengan pengguna (inisiasi perubahan fungsionalitas sistem oleh pengguna).
2. ***Adaptive system***, jika sistem tersebut mengalami penyesuaian tingkah laku secara dinamis dan otomatis menanggapi perubahan yang terjadi di dalam lingkungan interaksinya dengan pengguna (inisiasi perubahan fungsionalitas sistem oleh dirinya sendiri).

Menurut Gerhard Fischer, ciri khas pada sistem adaptasi yang membedakannya dari sistem adaptif [FIS01] antara lain:

1. Pengguna mengubah fungsionalitas sistem dengan dukungan substansi dari sistem,
2. sifat pengetahuan ditambahkan,
3. memungkinkan tercapainya keberhasilan model pengetahuan, dan
4. membutuhkan kerja pengguna untuk mempelajari komponen adaptasi.

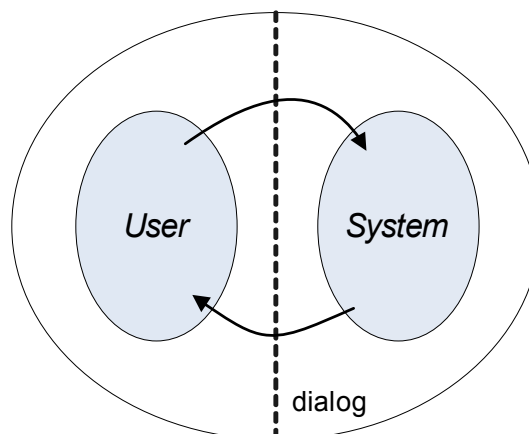
II.7 Metode Perancangan Interaksi

Menurut Jenny Preece [PRE02], hampir seluruh produk (termasuk perangkat lunak (sistem)) yang melibatkan interaksi pengguna (interaktif) dalam mendukung pelaksanaan aktivitas *tasks*-nya tidak dirancang dengan keterlibatan faktor pengguna. Pada umumnya produk tersebut hanya direkayasa sebagai sistem aplikasi untuk melakukan serangkaian fungsi. Oleh karena itu, perancangan interaksi membawa sebuah paradigma baru dalam

siklus pengembangan perangkat lunak (sistem) interaktif dengan melibatkan aspek *usability* dalam perancangan. Dengan demikian, sistem HCI (*Human Computer Interaction*) yang terdiri dari manusia, komputer, dan interaksi dapat dipandang sebagai solusi yang utuh [SAS99], tidak hanya dilihat berdasarkan aspek fungsionalitas aplikasi.

II.7.1 Sistem HCI

HCI adalah suatu studi, perencanaan, dan perancangan mengenai aktivitas bekerja bersama antara manusia (*user*) dan suatu sistem berbasis komputer [HCI07]. HCI merupakan bagian dari konsep rekayasa interaksi yang merupakan fundamental dari seluruh disiplin, bidang, dan pendekatan mengenai penelitian dan perancangan sistem berbasis komputer untuk penggunaannya oleh manusia [PRE02].



Gambar II-4 Sistem HCI Sebagai Solusi [SAS99]

Dalam konteks rekayasa interaksi, sistem HCI dipandang sebagai solusi, yaitu proses interaksi yang terjadi di antara pengguna dengan sistem interaktif [SAS99]. Suatu sistem HCI terdiri dari tiga komponen utama seperti yang ditunjukkan pada Gambar II-4, yaitu:

1. **Manusia** merepresentasikan pengguna individu atau kelompok yang menginginkan tujuan pekerjaannya tercapai dengan pemanfaatan teknologi.
2. **Komputer** merepresentasikan teknologi yang digunakan mulai dari komputer berbasis *desktop* hingga sistem komputer dalam skala besar.
3. **Interaksi** adalah pengalaman respon yang dialami oleh aktor dan reaktor pada suatu kejadian mutualisme [SHE07].

Pada sistem HCI, aktor dan reaktor yang dimaksudkan adalah pengguna atau sistem dalam suatu media interaktif yang memungkinkan terjadinya dialog. Dialog adalah pertukaran instruksi dan informasi antara manusia dan komputer [PRE93]. Konsep dialog inilah yang kemudian membentuk antarmuka sebagai media interaksi perangkat lunak (sistem) dengan pengguna [SAS99].

II.7.2 Perancangan Interaksi

Proses perancangan interaksi bersifat *user-centered* yaitu dengan mencari cara untuk melibatkan pengguna selama proses perancangan [PRE02]. Menurut Sastramihardja [SAS99], terdapat dua buah fase utama dalam perancangan interaksi, yaitu:

1. Fase kreatif

Fase ini diawali dengan tahap pencarian domain yang lebih spesifik dari *universe of discourse* yang luas. Kemudian dilakukan pengidentifikasian (*what* dan *why*) permasalahan sehingga dapat didefinisikan solusinya; atau prospek sehingga dapat didefinisikan *future* yang dapat dicapai dengan pengembangan suatu produk interaktif. Pada tahap ini kemudian diidentifikasi juga peran dari setiap pengguna (*who*) yang berinteraksi dengan produk interaktif yang dirancang.

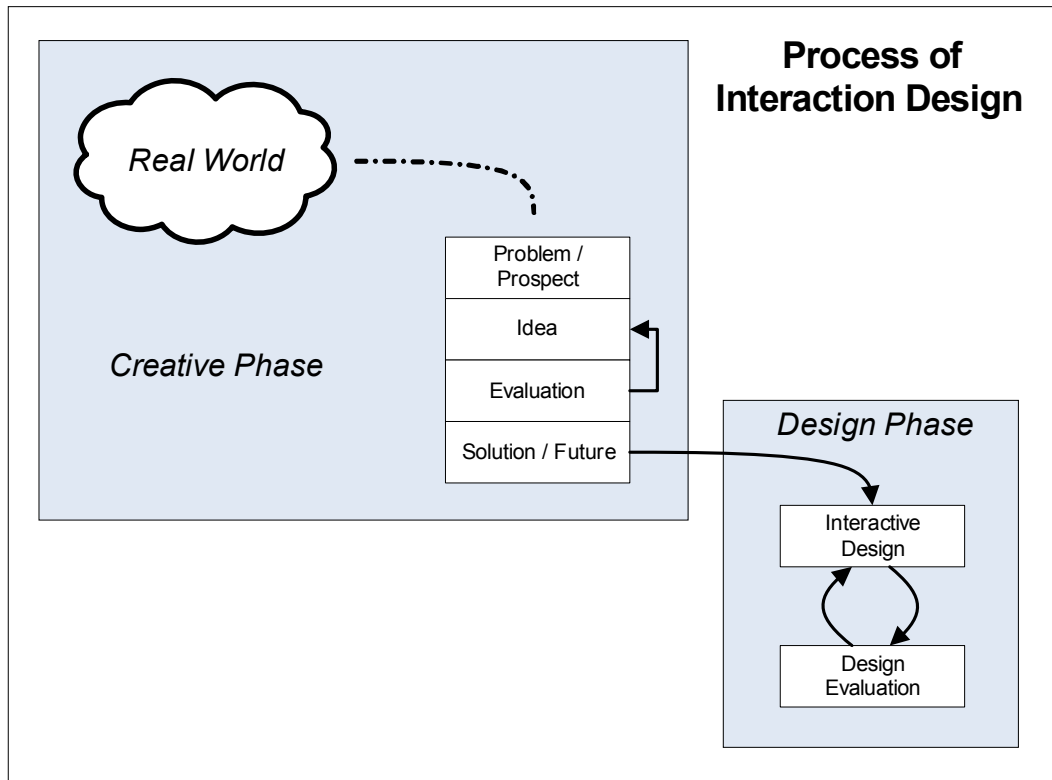
2. Fase perancangan

Fase ini terdiri dari dua tahap utama pemodelan *task* (komponen aktivitas), yaitu: (*user*) *task analysis* dan *task design*. Menurut IEEE STD 610, definisi *task* adalah urutan instruksi sebagai satuan unit kerja [SPA98].

Sedangkan menurut Preece [PRE02], perancangan interaksi terdiri dari empat buah aktivitas dasar, yaitu:

1. Mengidentifikasi kebutuhan pengguna dan mengembangkan kebutuhan solusi.
2. Membuat rancangan alternatif hingga rancangan solusi akhir.
3. Mengembangkan rancangan solusi dalam bentuk interaktif sehingga hasilnya dapat dikomunikasikan kepada pihak pengguna.
4. Mengevaluasi rancangan selama proses perancangannya.

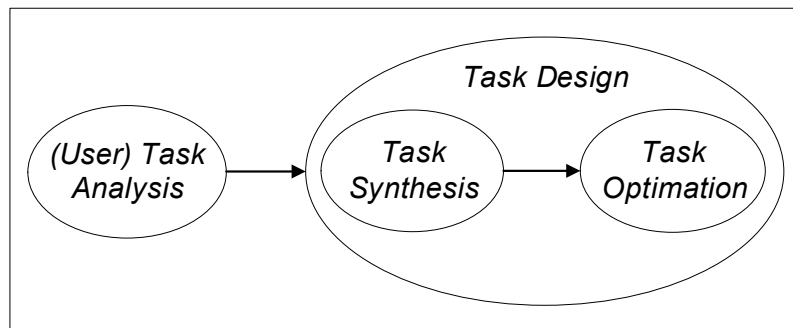
Proses evaluasi tidak hanya dilakukan pada langkah akhir, melainkan selama proses perancangan interaksi secara iteratif untuk memastikan produk interaktif telah memiliki *usability* yang sesuai dengan kebutuhan pengguna. Proses evaluasi dapat diperoleh dari opini pakar *usability* atau evaluasi interaksi pengguna terhadap hasil rancangan model [BEV91]. Proses perancangan interaksi yang diungkapkan oleh Sastramihardja dan Preece memiliki kesamaan pola. Proses pemodelan *tasks* [SAS99] dapat berperan pada perancangan model interaktif solusi [PRE02]. Dengan demikian, proses perancangan interaksi secara lengkap dapat ditunjukkan seperti pada Gambar II-5.



Gambar II-5 Metode Perancangan Interaksi

II.7.3 Pemodelan Task [SAS99]

Proses perancangan interaksi yang bersifat *user-centered* mendasari pemodelan *task* interaksi sistem HCI yang diawali dengan tahap (*user*) *task analysis*. Di dalam suatu proses rekayasa, aktivitas tidak berhenti sampai proses analisis saja. Oleh karena itu, dikembangkanlah konsep *task design* [SAS99]. Proses pemodelan *task* pada perancangan interaksi sistem HCI dapat dilihat pada Gambar II-6, dengan detail penjelasan sebagai berikut:



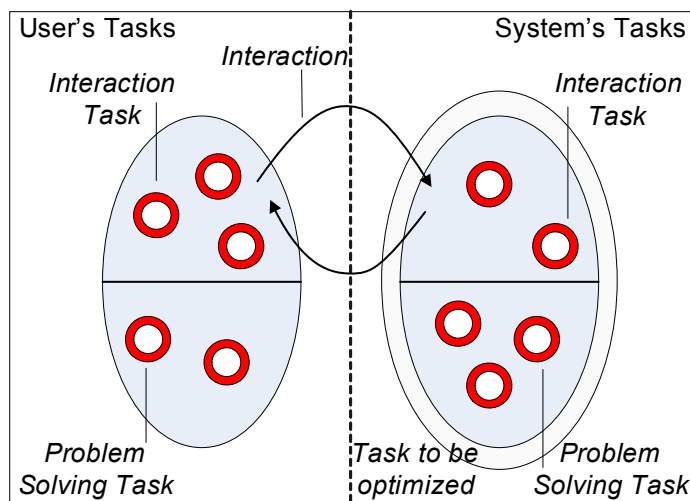
Gambar II-6 Tahapan Pemodelan Tasks pada Sistem HCI

1. **(User) task analysis** merupakan suatu proses analisis mengenai cara manusia (*user*) melakukan suatu pekerjaan (*tasks*). Berikut adalah teknik yang digunakan untuk melakukan analisis *task*:
 - a. Untuk setiap pengguna, lakukan analisis mengenai pekerjaan apa saja yang dilakukan terkait dengan persoalan dan untuk mencapai tujuan pekerjaan.
 - b. Untuk setiap pekerjaan pengguna, lakukan dekomposisi (penguraian) pekerjaan ke dalam *tasks* dan *subtasks*; klasifikasi pengetahuan mengenai setiap *task* (aksi dan objek); dan organisasi atas pengetahuan tersebut.

Model *task* direpresentasikan sebagai:

task <*verb, noun*>,
verb menunjukkan aksi yang dilakukan *user* atau *system*
noun menunjukkan objek dari aksi

2. **Task design** merupakan perancangan untuk memperoleh model *tasks* yang representatif terhadap rancangan perangkat lunak (sistem) dan kontribusi pengguna. Proses ini terdiri dari dua tahap utama, yaitu: *task synthesis* (pemetaan, reduksi, dan alokasi terhadap *tasks* pengguna dan sistem) dan *task optimization* (mengelola *task* sebagai komponen sistem ke dalam klasifikasi *generic, specific, atau parametric tasks* dengan harapan model sistem optimal). Pada tahap ini berarti telah mulai dilakukan perancangan *tasks* untuk sistem. *Tasks* yang sudah mulai dipetakan tersebut berorientasi pada penyelesaian persoalan (*problem solving tasks*) dan interaksi (*interaction tasks*) seperti yang ditunjukkan pada Gambar II-7.



Gambar II-7 Keterhubungan *User's Tasks* dan *System's Tasks* [SAS99]

II.8 Usability

Usability melibatkan proses optimasi interaksi antara pengguna dengan sistem interaktif sehingga memungkinkan pengguna menyelesaikan aktivitas penyampaian atau penerimaan informasi [SAS99]. Agar suatu aplikasi menjadi efektif, efisien, dan memuaskan, maka aplikasi tersebut harus dapat memungkinkan pengguna untuk menyelesaikan aktivitasnya sebaik mungkin. Menurut Landauer [LAN95], sebagian besar total biaya pengembangan perangkat lunak digunakan untuk perawatan karena permasalahan interaksi (*usability*) pengguna dengan sistem dan bukan permasalahan teknis. Situasi ini menggambarkan pentingnya *usability* dan mempertegas kebutuhan untuk melibatkan kriteria *usability* sebelum, pada saat, dan sesudah proses pengembangan perangkat lunak [GON04].

II.8.1 Definisi Usability

Kata "*usability*" digunakan secara luas dan memiliki definisi yang beragam. Berikut adalah salah satu definisi yang diberikan terhadap *usability* (kemudahan penggunaan [NIE03]) menurut ISO9241-11:

"[Usability refers to] the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." [USA07]

Berdasarkan definisi tersebut, terdapat tiga aspek penting dari *usability* [ABR03], yaitu:

1. **Effectiveness**: akurasi dan kelengkapan yang dilakukan pengguna untuk mencapai tujuan yang diinginkan.
2. **Efficiency**: sumber daya yang dikeluarkan untuk akurasi dan kelengkapan yang dilakukan pengguna untuk mencapai tujuan yang diinginkan.
3. **Satisfaction**: kebebasan dari perasaan tidak nyaman dan sikap positif terhadap penggunaan suatu produk.

II.8.2 Ukuran Kriteria Usability

Untuk memenuhi kebutuhan penggunanya, dibutuhkan objektif yang jelas bagi perancang produk interaktif [PRE02]. *Usability* merupakan suatu ukuran atribut kualitas yang dapat didefinisikan dengan jelas kepada pengguna. Ukuran tersebut seringkali dikenal sebagai kriteria *usability* (matriks PLUME) [COR99], yang terdiri dari:

1. **Productivity**, ukuran jumlah kerja yang dapat diselesaikan dalam waktu tertentu.
2. **Learnability**, ukuran pelatihan untuk memahami pemanfaatan sistem.

3. *User satisfaction*, ukuran subjektif pengguna terhadap sistem.
4. *Memorability*, ukuran seberapa kuat hubungan pembelajaran dan performansi.
5. *Error rates*, ukuran akurasi kerja yang dilakukan dalam serangkaian *tasks* lengkap.

Matriks tersebut dapat dijadikan sebagai kebutuhan perancangan dan digunakan selama perancangan sistem perangkat lunak untuk mengarahkan hasil rancangan agar sesuai dengan objektif yang telah didefinisikan dan disepakati.

II.8.3 Prinsip Perancangan dan *Usability* [PRE02]

Cara lain untuk mengkonseptualisasi *usability* adalah dengan menggunakan prinsip perancangan. Menurut Jenny Preece, prinsip perancangan merupakan kombinasi pengetahuan berbasis teori, pengalaman, dan nalar. Prinsip perancangan tidak menspesifikasikan bagaimana merancang bentuk antarmuka aktual (seperti merancang bentuk ikon atau merancang struktur portal *Web*), melainkan bertindak sebagai serangkaian pengingat bagi perancang, untuk meyakinkan ketersediaan hal-hal tertentu dan hal-hal yang harus dihindari dalam perancangan produk interaktif.

Menurut Norman, prinsip perancangan yang dapat digunakan untuk aspek *usability* dalam suatu produk interaktif [NOR88], antara lain:

1. *Visibility*, peningkatan visibilitas fungsi memudahkan pemahaman mengenai hal-hal yang dapat dilakukan oleh pengguna.
2. *Feedback*, pengiriman kembali informasi mengenai aksi yang telah dilakukan sehingga memungkinkan pengguna untuk melanjutkan aktivitasnya.
3. *Constraints*, membatasi interaksi pengguna pada waktu tertentu.
4. *Mapping*, relasi antara kendali dengan efek yang diberikan terhadap dunia nyata.
5. *Consistency*, merancang antarmuka operasi yang serupa; dan elemen serupa untuk *tasks* yang serupa.
6. *Affordance*, atribut objek yang memungkinkan pengguna mengetahui penggunaannya.

II.8.4 Prinsip Heuristik dan *Usability* [PRE02]

Ketika prinsip perancangan digunakan dalam praktiknya, seringkali diacu sebagai heuristik. Heuristik menekankan pada artian sesuatu yang harus dilakukan terhadap prinsip perancangan ketika diaplikasikan pada suatu permasalahan. Menurut Jacob Nielsen, terdapat sepuluh prinsip *usability* utama yang harus diperhatikan dalam perancangan produk interaktif [NIE05], antara lain:

1. **Visibility of system status**, menginformasikan pengguna mengenai apa yang terjadi dalam proses internal sistem dengan cara dan waktu yang tepat.
2. **Match between system and real world**, menggunakan konsep yang dikenali oleh pengguna dalam lingkungan nyata dan tidak dengan istilah-istilah yang berorientasi pada sistem.
3. **User control and freedom**, memungkinkan pengguna pindah dari status yang tidak diinginkan tanpa melalui serangkaian dialog.
4. **Consistency and standards**, mencegah pengguna menduga-duga perbedaan kata, situasi, atau aksi menunjuk pada suatu hal yang sama.
5. **Error prevention**, mencegah pengguna melakukan kesalahan sejak dari hasil rancangan dengan mengeliminasi kondisi yang rawan kesalahan, atau memberikan pilihan konfirmasi sebelum aksi dilakukan.
6. **Recognition rather than recall**, meminimalisasi beban daya ingat (*memory*) pengguna dengan memperhatikan visibilitas objek, aksi, dan opsi. Instruksi-instruksi harus mudah dilihat dan dicapai pada saat dibutuhkan.
7. **Flexibility and efficiency of use**, meningkatkan kecepatan proses interaksi bagi pengguna awam maupun mahir dengan memungkinkan pengguna melakukan aksi yang sering dilakukan.
8. **Aesthetic and minimalist design**, dialog sebaiknya tidak mengandung informasi yang tidak relevan atau jarang dibutuhkan.
9. **Help users recognize, diagnose, and recover from errors**, pesan kesalahan harus disampaikan dengan bahasa yang jelas, mengindikasikan permasalahan, dan menyarankan penyelesaian masalah.
10. **Help and documentation**, ketersediaan bantuan dan dokumentasi harus mudah untuk dicari, fokus terhadap *task* pengguna, menuliskan langkah-langkah untuk dilakukan, dan tidak terlalu besar.

II.9 State Transition [MOD07]

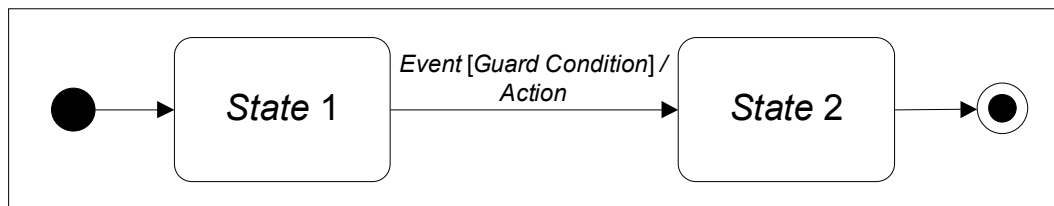
Suatu sistem memiliki *behaviour* yang dinamik berdasarkan respon terhadap kejadian. *State transition* dapat digunakan untuk merefleksikan urutan fungsi, urutan aksi internal sebuah fungsi tunggal, atau urutan fungsi entitas eksternal spesifik. *State transition* umumnya ditunjukkan dalam bentuk diagram pada perancangan perangkat lunak (sistem) untuk memodelkan sistem *real-time* dan mengeksplorasi *behaviour* kompleks dari kelas, aktor, subsistem, atau komponen.

II.9.1 State Transition Diagram

State transition diagram merupakan deskripsi spesifikasi seluruh kemungkinan *behaviours* dari suatu elemen model dinamik [MOD07]. *Behaviour* dimodelkan sebagai grafik *state* traversal dengan interkoneksi *nodes* pada satu atau lebih busur transisi yang dipicu oleh serangkaian instan kejadian (*event*). Selama proses traversal, *state machine* mengeksekusi serangkaian aksi yang berasosiasi dengan berbagai elemen tersebut.

Status (*states*) (dan aksi yang berasosiasi) menspesifikasikan respon dari *resource* atau fungsi, terhadap suatu kejadian. Ketika suatu kejadian berlangsung, status selanjutnya dapat beragam tergantung pada status saat ini (dan aksi yang berasosiasi), kejadian, dan *guard conditions* (kondisi yang harus terpenuhi untuk dapat melakukan suatu transisi). Perubahan status disebut dengan transisi. Setiap transisi menspesifikasikan respon berdasarkan kejadian spesifik dan status saat ini. Aksi dapat berasosiasi dengan status atau transisi antar status.

II.9.2 Semantik State Transition Diagram



Gambar II-8 *State Transition Diagram* Generik [MOD07]

Semantik *state transition diagram* generik ditunjukkan pada Gambar II-8. Titik hitam menunjukkan *initial states*, sedangkan *terminal states* digambarkan sebagai lingkaran titik hitam. *States* digambarkan dengan kotak bulat dan diberi label nama, nomor, dan aksi (opsional) yang berasosiasi dengan *state* tersebut. Transisi ditunjukkan dengan panah satu arah yang diberi label notasi *event* yang mengakibatkan transisi, *guard condition* yang harus dipenuhi untuk suatu transisi, dan aksi (jika ada) yang berasosiasi dengan transisi.