

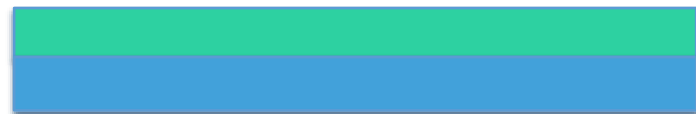
Membangun Sistem DSS vs GDSS

DECISION SUPPORT SYSTEM [D10K-5B01]





Decision Support System (DSS)



Agenda

- Pendahuluan
- Strategi Pengembangan DSS
- Proses Pengembangan DSS
- System Development Issues
- Pengembangan DSS Berbasis User
- Pengembangan DSS Berbasis Kelompok

Pendahuluan

- Membangun sebuah DSS, apalagi yang besar, merupakan proses yang rumit.
- Melibatkan hal-hal : teknis (hardware, software) dan perilaku (interaksi manusia-mesin, dampak DSS pada individu)

Strategi Pengembangan DSS

1. Tulis DSS dengan bahasa pemrograman umum : Pascal, Delphi, Java, C++, PHP dll.

2. Menggunakan 4GL : data-oriented language.

Bahasa generasi keempat (4GL) disebut juga dengan 'Bahasa produktifitas Tinggi', 'Bahasa non-prosedural', 'Pembuat aplikasi'. Bahasa 4GL ini berfungsi untuk menerima file yang berisi urutan-urutan statement program dan menghasilkan program yang dapat dijalankan. Contoh: Oracle, Informix.

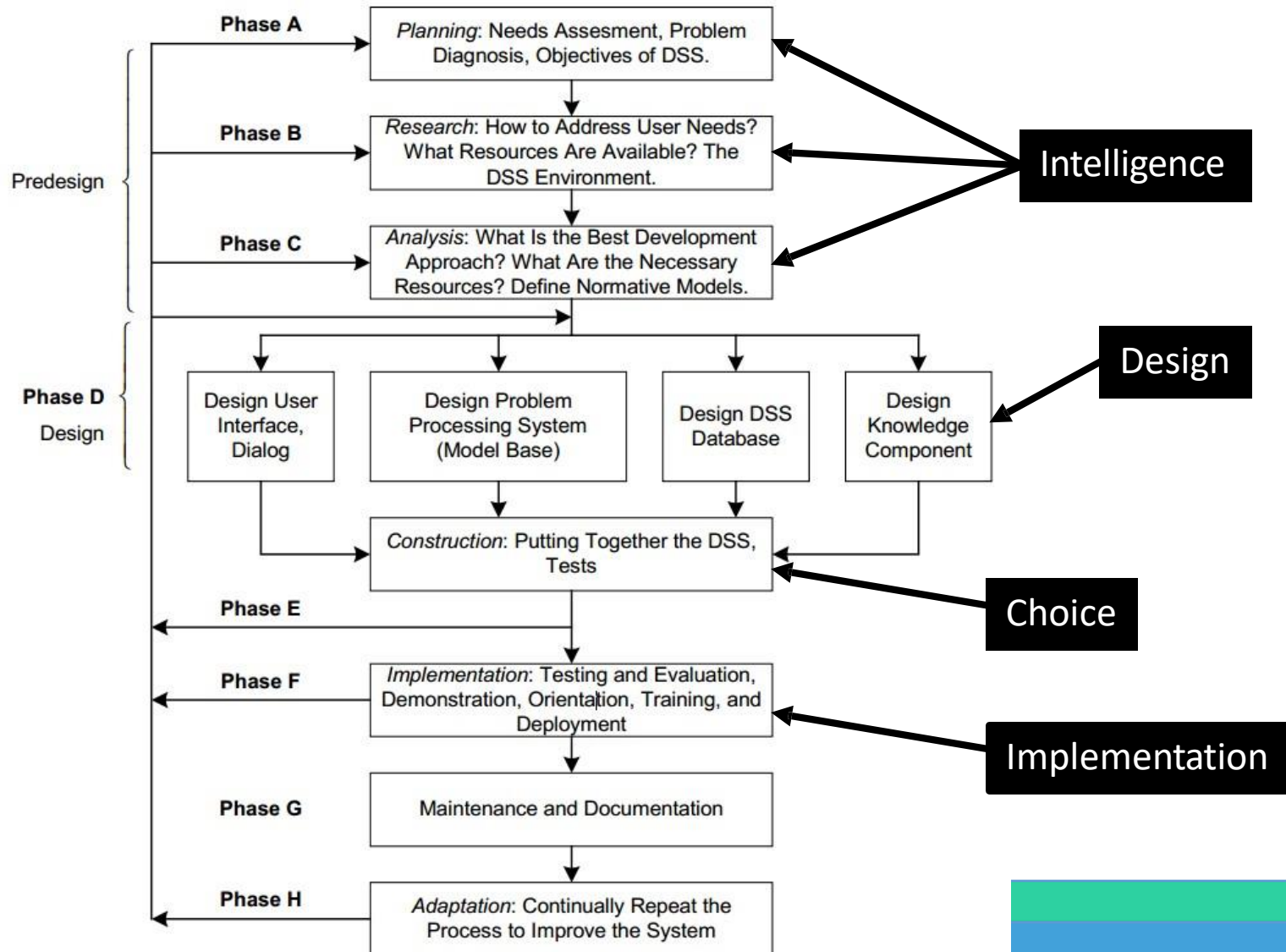
3. Menggunakan DSS Generator : Excell.

4. Menggunakan DSS Generator khusus (domain specific):
Commander FDC untuk budgeting & financial analysys, EFPM untuk kalangan perguruan tinggi

5. Mengembangkan DSS dengan metodologi CASE

6. Untuk DSS yang kompleks, bisa mengintegrasikan pendekatan-pendekatan di atas.

Proses Pengembangan DSS



Proses Pengembangan DSS

1. Perencanaan.

Merumuskan kerangka dan ruang lingkup SPK, persyaratan unjuk kerja, dan memilih konsep-konsep & menganalisis model pembuatan keputusan yang relevan dengan tujuan SPK. Langkah ini menentukan pemilihan jenis SPK yang akan dirancang dan metode pendekatan yang dipergunakan.

2. Penelitian.

Berhubungan dengan pencarian data serta sumber daya yang tersedia

3. Analisis & Perancangan konsep.

Penentuan teknik pendekatan yang akan dilakukan serta sumber daya yang dibutuhkan

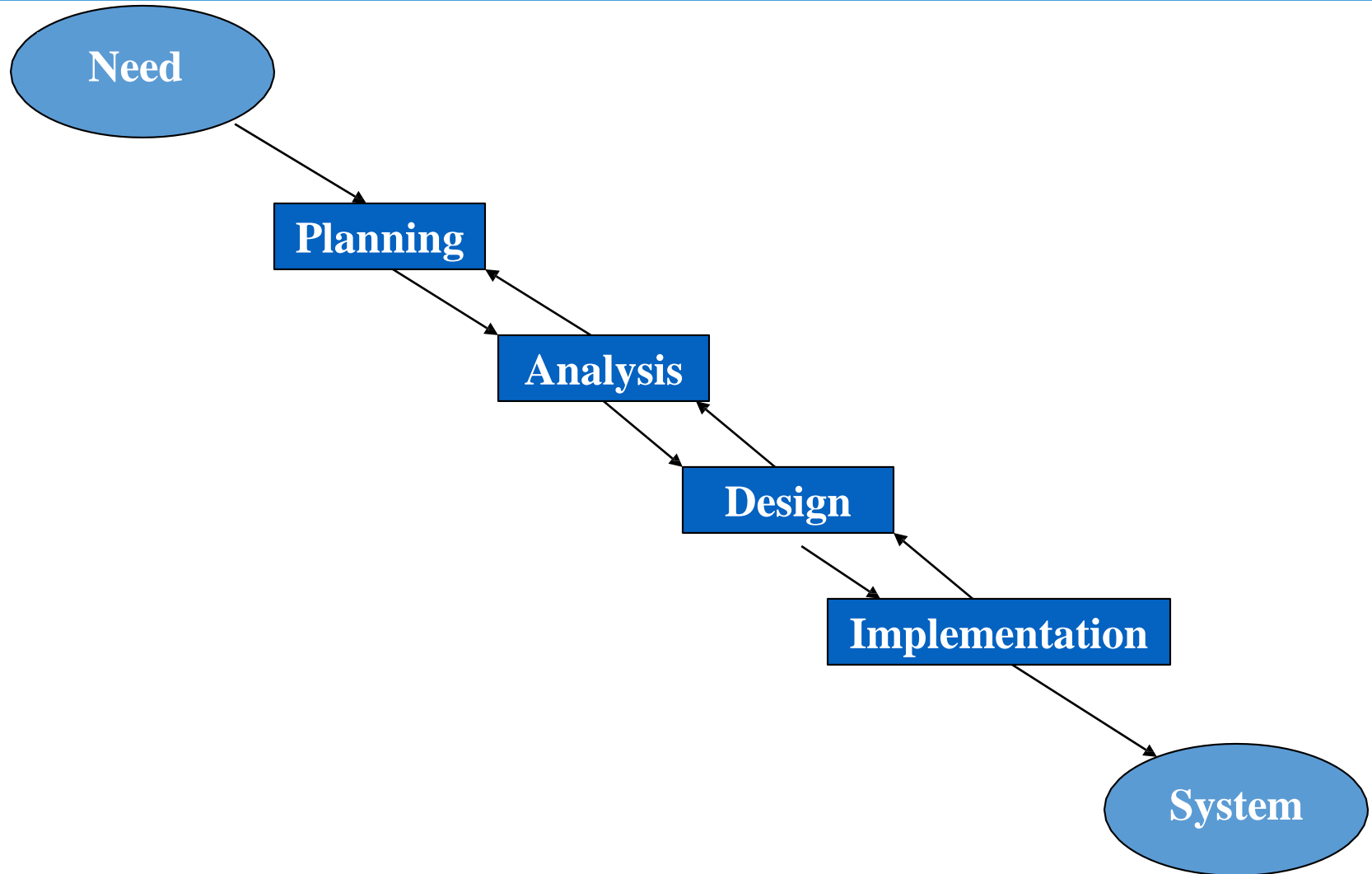
Proses Pengembangan DSS

4. Perancangan.
Melakukan perancangan ketiga subsistem utama SPK yaitu subsistem database, model dan Dialog.
5. Konstruksi.
Merupakan kelanjutan dari perancangan dimana ketiga subsistem yang telah dirancang digabungkan menjadi suatu SPK
6. Implementasi.
Menerapkan SPk yang dibangun. Pada tahap dilakukan testing, evaluasi, penampilan, orientasi, pelatihan dan penyebaran
7. Pemeliharaan.
Tahapan yang dilakukan terus menerus untuk mempertahankan keandalan sistem
8. Adaptasi.
Melakukan pengulangan terhadap tahapan diatas sebagai tanggapan terhadap perubahan kebutuhan "pemakai".

System Development Issues

- **System development life cycle (SDLC)**
- **Prototyping**
- **Forming the development team**
- **Complex process**
- **Technical issues**
- **Behavioral issues**
- **Different approaches**

Traditional Systems Development Life Cycle (SDLC) (Waterfall)



Traditional Systems Development Life Cycle (SDLC) (Waterfall)

TABLE 6.1 The Traditional System Development Life Cycle

<i>Major Phase</i>	<i>Minor Step</i>	<i>Deliverable</i>
Planning: Why build the system?	1. Identify business value	System request
	2. Analyze feasibility	Feasibility study
	3. Develop work plan	Work plan
	4. Staff project	Staffing plan
	5. Control and direct project	Project charter Project management tools CASE tool Standards list Project “binders” or files Risk assessment
Analysis: Who, what, when, and where will the system be?	6. Analyze problem	Analysis plan
	7. Gather information	Information
	8. Model process(es)	Process model
Design: How will the system work?	9. Model data	Data model
	10. Design physical system	Design plan
	11. Design architecture	Architecture design Infrastructure design
	12. Design interface	Interface design
	13. Design database and files	Data storage design
Implementation: System delivery	14. Design program(s)	Program design
	15. Construction	Test plan Programs Documentation
	16. Installation	Conversion plan Training plan

Source: Based on Dennis and Wixom (2000).

Planning

Why Build the System?

Minor Step

- 1. Identify business value**
- 2. Analyze feasibility**
- 3. Develop work plan**
- 4. Staff project**
- 5. Control and direct project**

Deliverable

System request
Feasibility study
Work plan
Staffing plan,
Project charter
Project management tools
CASE tool
Standards list
Project binders / files
Risk assessment

Analysis

Who, What, When, Where?

Minor Step

Deliverable

6. Analyze problem

Analysis plan

7. Gather information

Information

8. Model process(es)

Process model

9. Model data

Data model

Design

How Will the System Work?

Minor Step	Deliverable
10. Design physical system	Design plan
11. Design architecture	Architecture design, Infrastructure design
12. Design interface	Interface design
13. Design database and files	Data storage design
14. Design program(s)	Program design

Implementation

System Delivery

Minor Step

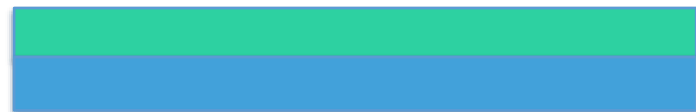
Deliverable

15. Construction

**Test plan,
Programs,
Documentation**

16. Installation

**Conversion plan,
Training plan**



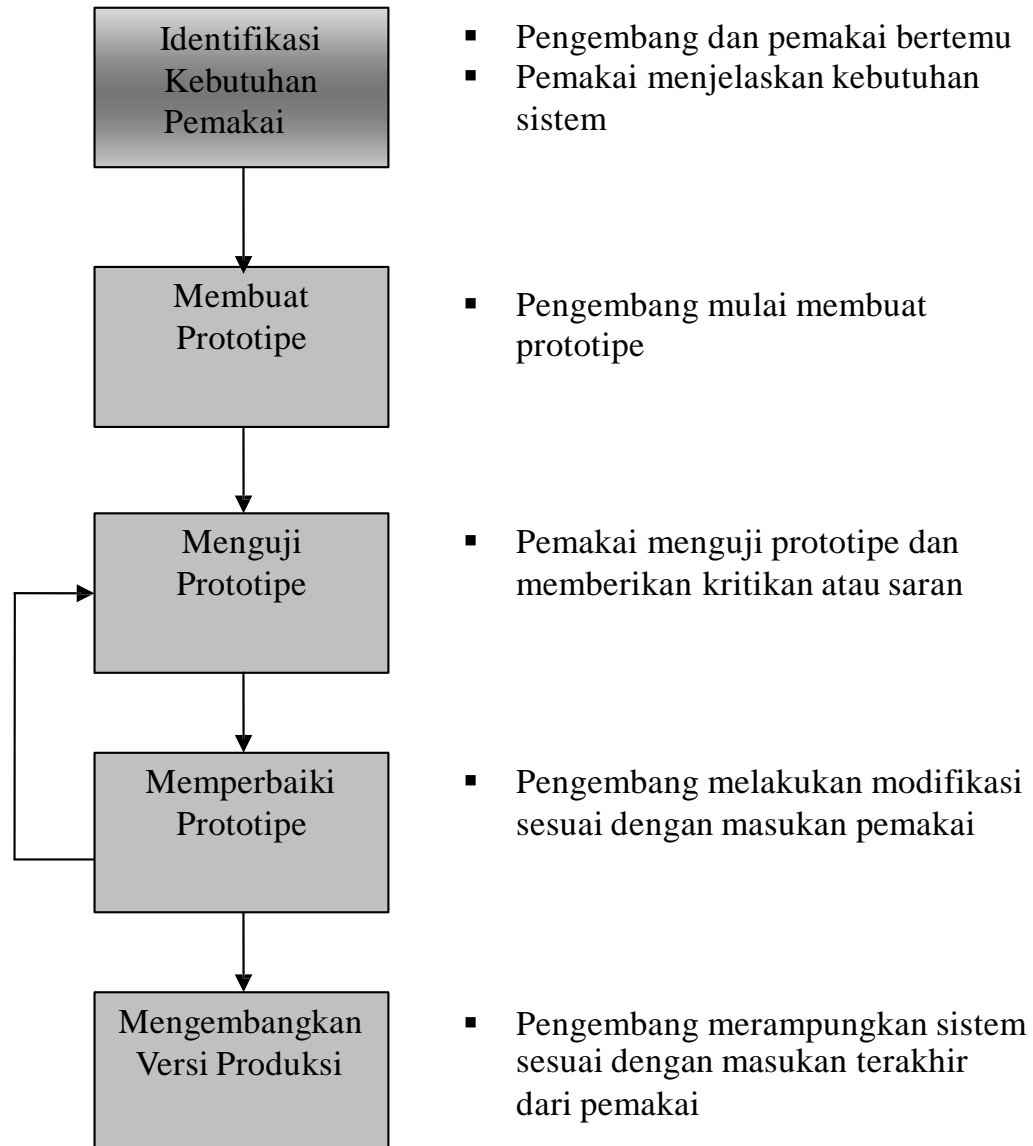
Common Issues

- Diperlukan manajemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
- Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
- Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidakpastian pada saat awal pengembangan.
- Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
- Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

Prototipe

- Suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai
- Hal ini berbeda dengan pendekatan SDLC tradisional (konvensional) yang lebih banyak menghabiskan waktu untuk menghasilkan spesifikasi yang sangat rinci sebelum pemakai dapat mengevaluasi sistem
- Mengingat kebanyakan pemakai mengalami kesulitan dalam memahami spesifikasi sistem berakibat bahwa pemakai tidak begitu paham sampai pengujian dilakukan

Pendekatan Prototipe



Kelebihan Prototipe

- Pendefinisian kebutuhan pengguna menjadi lebih baik karena keterlibatan pengguna yang lebih intensif
- Meningkatkan kepuasan pengguna dan mengurangi risiko pengguna tidak menggunakan sistem mengingat keterlibatan mereka yang sangat tinggi sehingga sistem memenuhi kebutuhan mereka dengan lebih baik
- Mempersingkat waktu pengembangan
- Memperkecil kesalahan disebabkan pada setiap versi prototipe, kesalahan segera terdeteksi oleh pengguna
- Pengguna memiliki kesempatan yang lebih banyak dalam meminta perubahan-perubahan
- Menghemat biaya (menurut penelitian, biaya pengembangan dapat mencapai 10% hingga 20% dibandingkan kalau menggunakan SDLC tradisional)

Kekurangan Prototipe

- Prototipe hanya bisa berhasil jika pemakai bersungguh-sungguh dalam menyediakan waktu dan pikiran untuk menggarap prototipe
- Kemungkinan dokumentasi terabaikan karena pengembang lebih berkonsentrasi pada pengujian dan pembuatan prototipe
- Mengingat target waktu yang pendek, ada kemungkinan sistem yang dibuat tidak lengkap dan bahkan sistem kurang teruji
- Jika terlalu banyak proses pengulangan dalam membuat prototipe, ada kemungkinan pemakai menjadi jenuh dan memberikan reaksi yang negatif
- Apabila tidak terkelola dengan baik, prototipe menjadi tak pernah berakhir. Hal ini disebabkan permintaan terhadap perubahan terlalu mudah untuk dipenuhi

Kondisi yang baik untuk Pembuatan Prototipe

- Sistem mempunyai resiko tinggi
 - Tidak jelas permasalahannya
 - Tidak jelas kebutuhan & keinginan
 - Tidak pasti ada yang ingin dilakukan
- Perancangan Dialog User - Komputer
 - Bagaimana membuat dialog yg. baik, ramah, mudah ?
- Sistem diminati oleh banyak pemakai
 - Mencari kesepakatan
 - Basis untuk menyamakan persepsi
- User ingin cepat selesai
 - User tidak sabar menunggu
 - Prototipe segera memperlihatkan bentuk kerja sistem

Kondisi yang cocok untuk Pembuatan Prototipe

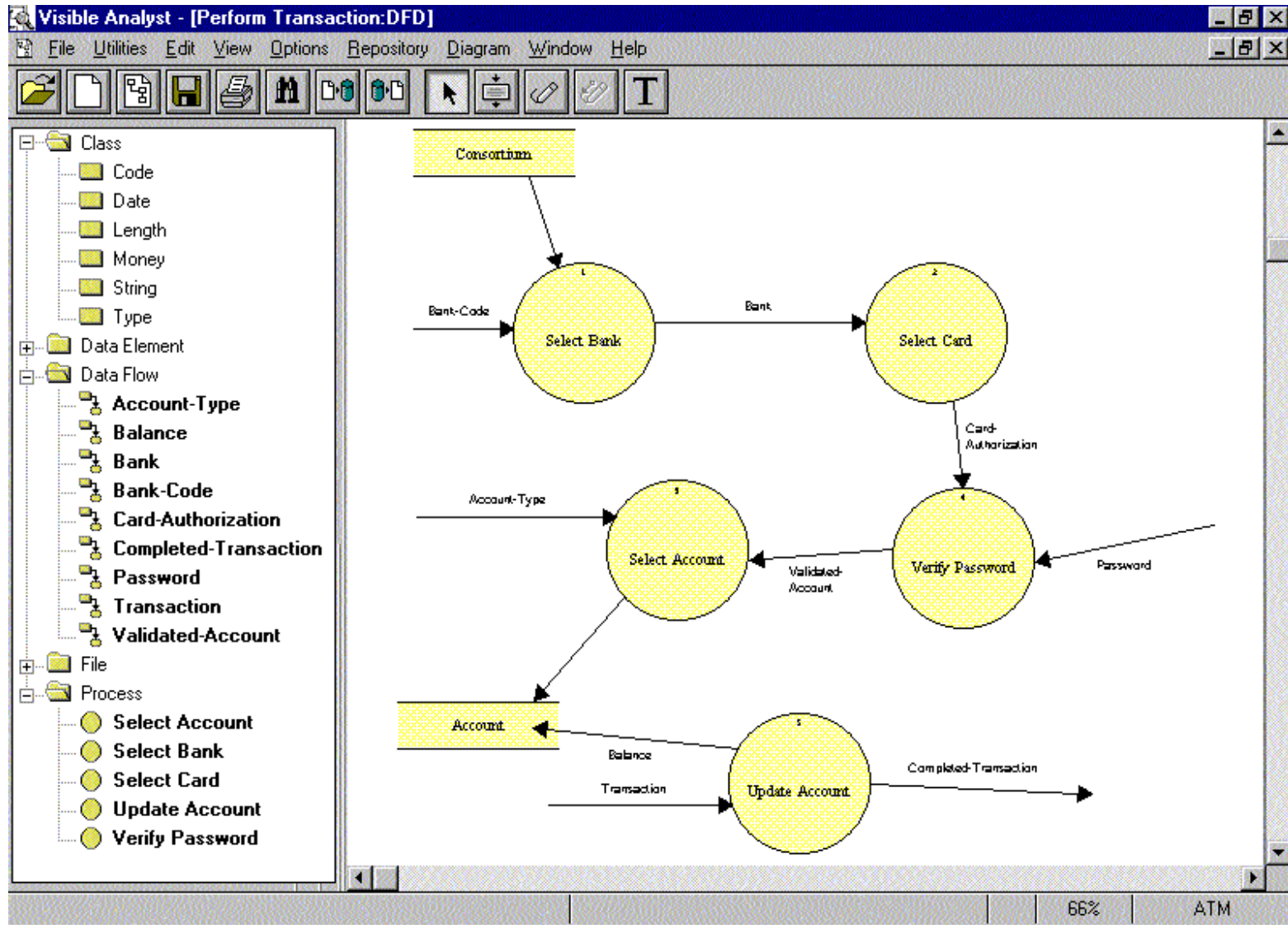
- Masa pakai singkat
 - Sistem hanya dipakai beberapa kali saja
- Ingin menunjukkan inovasi
 - Pengembangan dapat menunjukkan kecanggihan
 - Sistem cepat terlihat (mungkin juga cepat selesai)
- Kebutuhan berubah-ubah
 - User sulit menjelaskan kebutuhan
 - Menjadi keadaan yang paling umum untuk memakai prototyping

CASE Tools

- **Information systems for systems analysts**
- **Can help manage system development**
- **Upper CASE (assists in analysis)**
- **Lower CASE (manages diagrams and code generation)**
- **Integrated CASE (both)**

Visible Analyst

Courtesy of Visible System Corporation



DSS Technology Levels and Tools

Three Levels of DSS Technology

- **Specific DSS [the application]**

“Final Product” atau aplikasi DSS yang nyata-nya menyelesaikan pekerjaan

- **DSS integrated tools (generators)**

adalah software pengembangan terintegrasi yang menyediakan sekumpulan kemampuan untuk membangun specific DSS secara cepat, tak mahal, dan mudah. Contoh : Microsoft Excel

- **DSS primary tools [programming languages]**

Level terendah dari teknologi DSS adalah software utility atau tools. Elemen ini membantu pengembangan baik DSS generators

DSS Technology Levels and Tools

Three Levels of DSS Technology

- **Specific DSS [the application]**

“Final Product” atau aplikasi DSS yang nyata-nya menyelesaikan pekerjaan

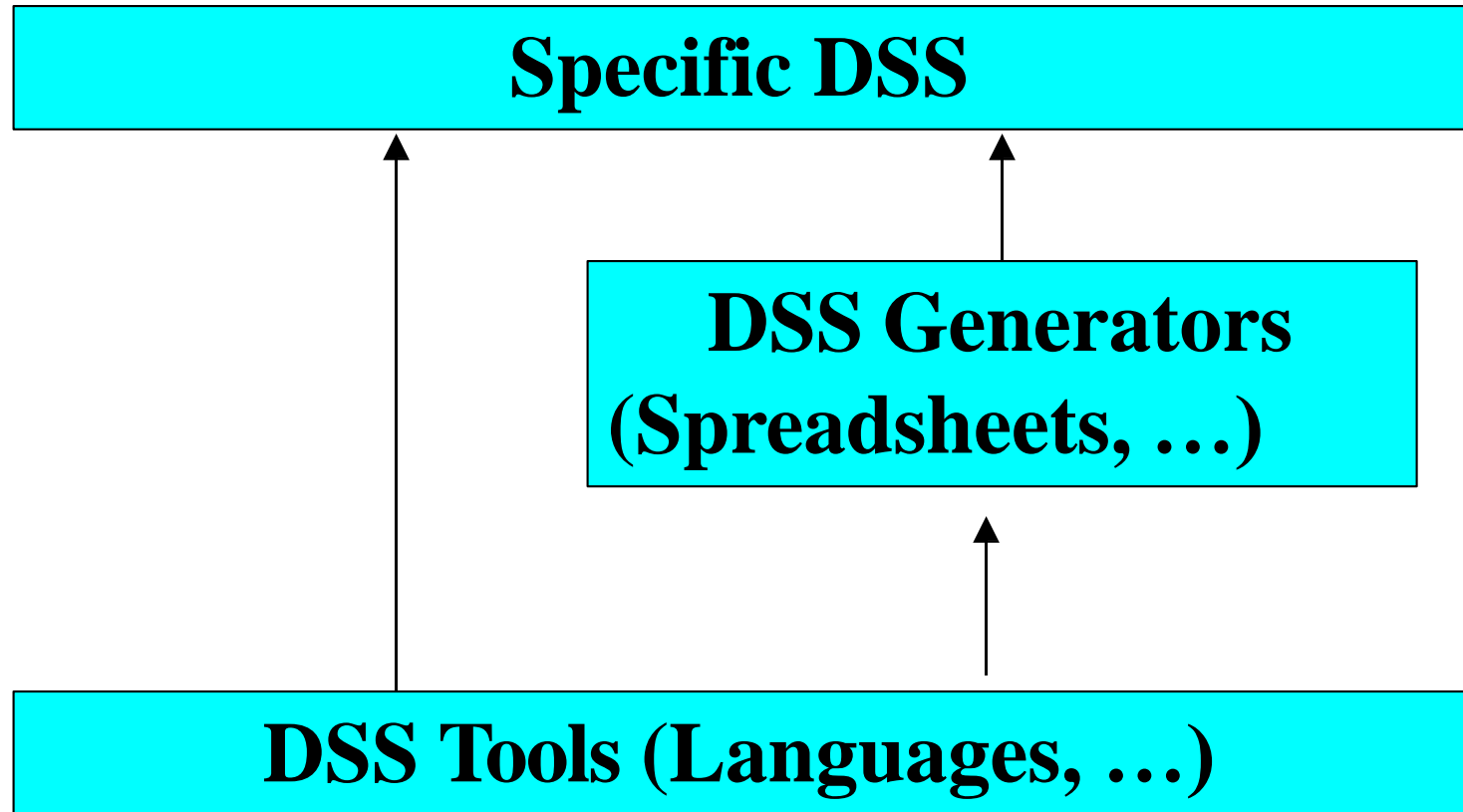
- **DSS integrated tools (generators)**

adalah software pengembangan terintegrasi yang menyediakan sekumpulan kemampuan untuk membangun specific DSS secara cepat, tak mahal, dan mudah. Contoh : Microsoft Excel

- **DSS primary tools [programming languages]**

Level terendah dari teknologi DSS adalah software utility atau tools. Elemen ini membantu pengembangan baik DSS generators

DSS Technology Level



Hardware Selection

- **PC**
- **Unix workstations**
- **Network of Unix workstations**
- **Web servers**
- **Mainframes**
- **Typically use existing hardware**

Software Selection

Complex because

- At start, information requirements, etc. are unknown
- Hundreds of packages
- Software updated rapidly
- Price changes
- Many people involved in decision
- Language capability problems

(More)

Pengembangan DSS Berbasis Tim

- Pengembangan DSS pada tahun 1970 dan 1980-an melibatkan skala yang besar, sistemnya kompleks, dan didesain utama untuk mendukung organisasi.
- Sistem ini didesain oleh tim yang terdiri dari user, penghubung, DSS builder, tenaga ahli, dan berbagai tools
- Secara organisasi, penempatan DSS Group bisa dimana-mana, umumnya pada lokasi:
 - Dalam departemen IS (Information Services)
 - Executive Staff Group
 - Dalam Wilayah keuangan ataupun fungsi lainnya
 - Dalam departemen rekayasa industri
 - Dalam kelompok manajemen pengetahuan (Management Science Group)
 - Dalam kelompok pusat informasi (Information Center Group)
- Dengan berbasis Tim, maka pembangunan DSS menjadi kompleks, lama dan prosesnya memakan biaya.

Pengembangan DSS Berbasis User

Pengembangan DSS berbasis user adalah pengembangan dan penggunaan sistem informasi berbasis komputer oleh orang-orang di luar wilayah sistem informasi formal.

- Keuntungan bila user sendiri yang membangun DSS :
 - Waktu penyelesaian singkat
 - Syarat-syarat spesifikasi kebutuhan sistem tak diperlukan
 - Masalah implementasi DSS dapat dikurangi
 - Biayanya sangat rendah
- Resikonya adalah :
 - Kualitasnya bisa tak terjaga.
 - Resiko potensial kualitas dapat diklasifikasi dalam 3 katagori: (a) tool dan fasilitas dibawah standar (b) resiko yang berhubungan dengan proses pengembangan (contoh : pengembangan sistem yang mnghasilkan hasil yang salah), dan (c) resiko manajemen data (misal : kehilangan data).



Group Decision Support System (GDSS)



Outline

- Pendahuluan
- Kelompok dan Sistem Pendukung Keputusan Kelompok
- Situasi Pengambilan Keputusan pada GDSS
- Clinical Group, Topologi, dan Ruang GDSS
- Karakteristik GDSS
- Level Group GDSS

Pendahuluan

- Banyak organisasi baik perseorangan maupun umum berusaha meningkatkan pijakannya pada rekayasa proses bisnis, penggunaan teknologi, dan pengenalan program peningkatan kualitas sebagai jalan untuk meningkatkan produktivitas dan penanganan yang lebih baik dengan kompetisi yang makin ketat, permintaan customer, penurunan anggaran, dan timbulnya pasar global.
- Di Manhattan - Amerika Serikat - manajemen dan karyawan dari IRS (Internal Revenue Service), dengan dibantu University of Minnesota mengimplementasikan program peningkatan kualitas berbasis pada manajemen partisipasi (tim kualitas), yang didukung oleh GDSS.
- Bagian utama dari program peningkatan kualitas ini adalah struktur tim kualitas, yang serupa dengan konsep daur ulang kualitas orang Jepang. Grup, terdiri dari manajer dan karyawan, bertemu sebagai unit kecil (dari 3 sampai 12 orang) untuk merumuskan metode bagi penyelesaian masalah dan menggunakan bermacam peluang untuk meningkatkan kualitas.

Pendahuluan

- Partisipan dalam tim kualitas sering datang dari pelbagai wilayah fungsional atau level penyeliaan yang berbeda, dan ini akan membawa pelbagai perspektif kedalam tim.
- Walaupun pelbagai perbedaan ini dapat memperkaya pertemuan, ia malahan dapat juga memperlambat pekerjaan.
- Sebagai tambahan, grup tergantung juga pada fenomena umum yang menghambat kesuksesan kerja tim. Misalnya, dominasi sebagian anggota, komunikasi antarpersonal yang jelek, dan ketakutan mengekspresikan ide-ide inovatif.
- Untuk mengurangi efek negatif tadi, diberikan pelatihan ekstensif dan bantuan profesional.
- Jika jumlah tim bertambah, pelatihan dan anggaran bantuan menjadi masalah dan juga sulit menemukan fasilitator berkualitas tinggi.

Solusi

- GDSS dimulai sebagai proyek riset. Laboratorium khusus yang dibangun selama akhir 1980-an di beberapa universitas, termasuk University of Minnesota, yang turut membangun IRS pada tahun 1988.
- Pada periode itu tidak ada hardware dan software komersial di pasaran.
- Software GDSS awal kali digunakan di University of Minnesota dengan sebutan **SAMM** (Software-aided Meeting Management).
- Pemimpin tim dan anggotanya lalu dilatih untuk menggunakan software, dan ditunjukkan bagaimana meningkatkan proses peningkatan kualitas menggunakan berbagai fitur **SAMM**.

Hasil

- Selama September 1989 sampai dengan Januari 1991, SAMM digunakan untuk ratusan pertemuan. SAMM digunakan oleh para anggota tim untuk:
 1. Peneluran ide dan evaluasi (19.4% dari pertemuan)
 2. Penggunaan tools bantuan keputusan yang canggih (59.4% dari pertemuan)
 3. Pembuatan dan pengaturan agenda (36.5% dari pertemuan).
 4. Penulisan dan pengelolaan record/catatan grup (15.3% dari pertemuan)

Pada skala 1 sampai 7 anggota tim memberikan nilai 5.5 untuk tingkat kepuasan mereka dan menyatakan: “merasa nyaman dengan teknologi ini” , “meningkatkan kerja tim”, “GDSS mudah digunakan dalam grup”.

Kelompok dalam GDSS

- **Turban:** sifat-sifat Group Decision Making
 - Perlu adanya pertemuan
 - Hasil pertemuan tergantung pada pengetahuan dan pendapat dari anggota kelompok.
 - Perlu adanya komposisi dan proses pengambilan keputusan.
 - Perlu adanya penyelesaian jika terjadi perbedaan pendapat.

Definisi GDSS

- *Group Decision Support System* (GDSS) adalah sistem yang digunakan oleh sekelompok orang untuk mencari penyelesaian masalah dengan menggunakan teknologi komunikasi, komputasi dan dukungan keputusan.

Tujuan GDSS

- Menurut **Desanctis** dan **Gallupe** (1987):
 - GDSS bertujuan untuk memperbaiki proses pembuatan keputusan kelompok dengan cara:
 - mengurangi kendala komunikasi,
 - menyediakan beragam cara untuk menyusun analisis keputusan, serta
 - memberikan arahan secara sistematis terhadap pola, jadwal dan isi diskusi.

Situasi Pengambilan Keputusan pada GDSS

- Proses pengambilan keputusan dalam suatu kelompok dapat terjadi pada situasi yang mana:
 - setiap individu dicirikan oleh persepsi, sikap, dan motivasi pengambil keputusan itu sendiri;
 - siapa yang akan mengakui atau menerima permasalahan secara umum (domain); dan
 - usaha untuk mendapatkan keputusan kolektif.



Situasi Pengambilan Keputusan pada GDSS

- Proses pengambilan keputusan dapat dilakukan dalam kerangka yang lebih fleksibel.
- Fleksibilitas ini salah satunya ditunjukkan dengan kemampuannya dalam mensimulasikan proses pengambilan keputusan dengan konsistensi yang bersifat kabur (Kickert, 1978).
- Metode-metode dalam pengambilan keputusan secara kelompok biasanya akan mengalami kendala ketika setiap pengambil keputusan memberikan preferensinya secara individual (Kwok, 2005).

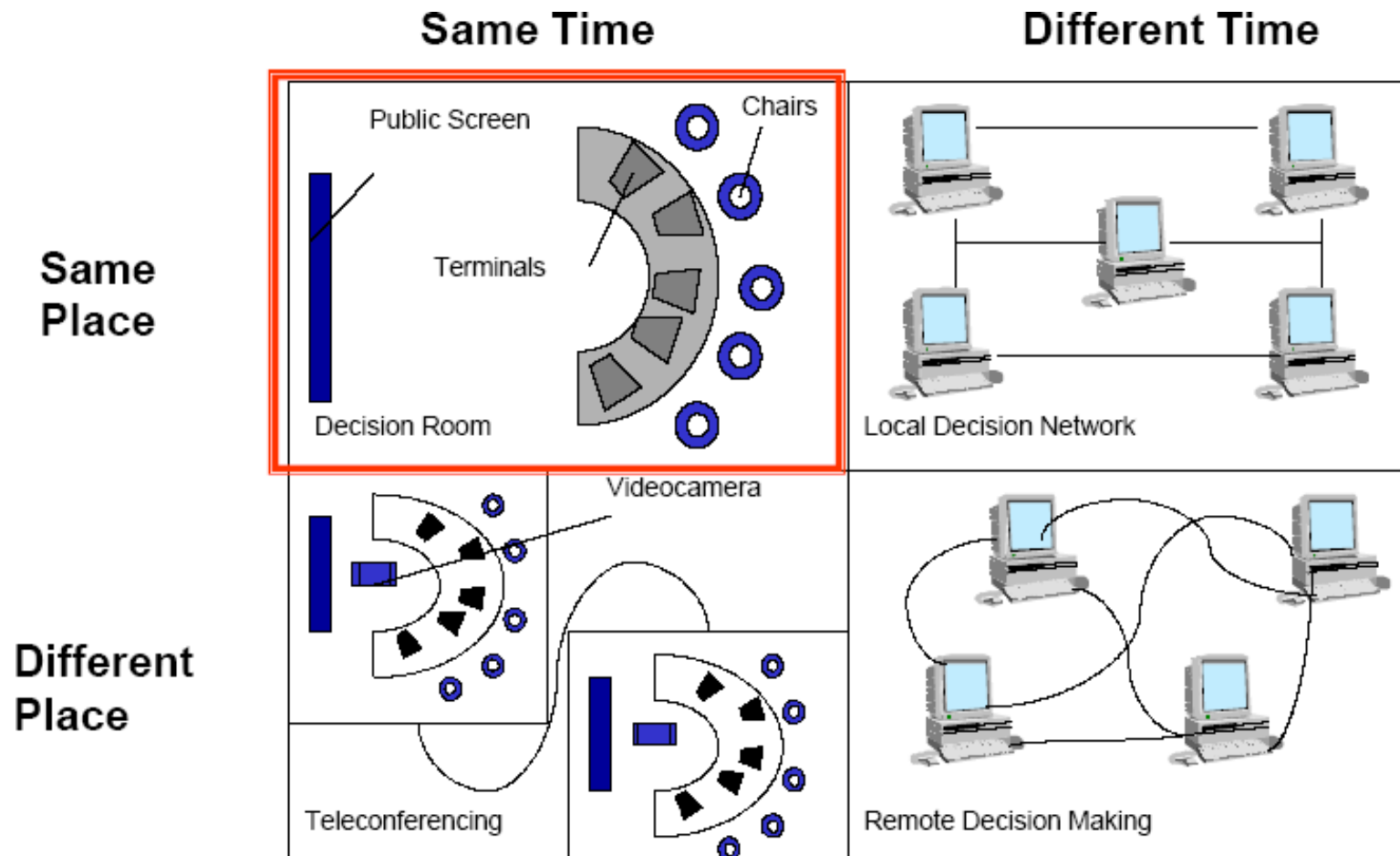
Situasi Pengambilan Keputusan pada GDSS

- **Secara umum**, ada 2 tahap yang harus dilakukan dalam *Group Support System* (GSS) yaitu:
 - membangkitkan preferensi pengambil keputusan secara terpisah; dan
 - melakukan agregasi kelompok terhadap setiap preferensi yang diberikan.

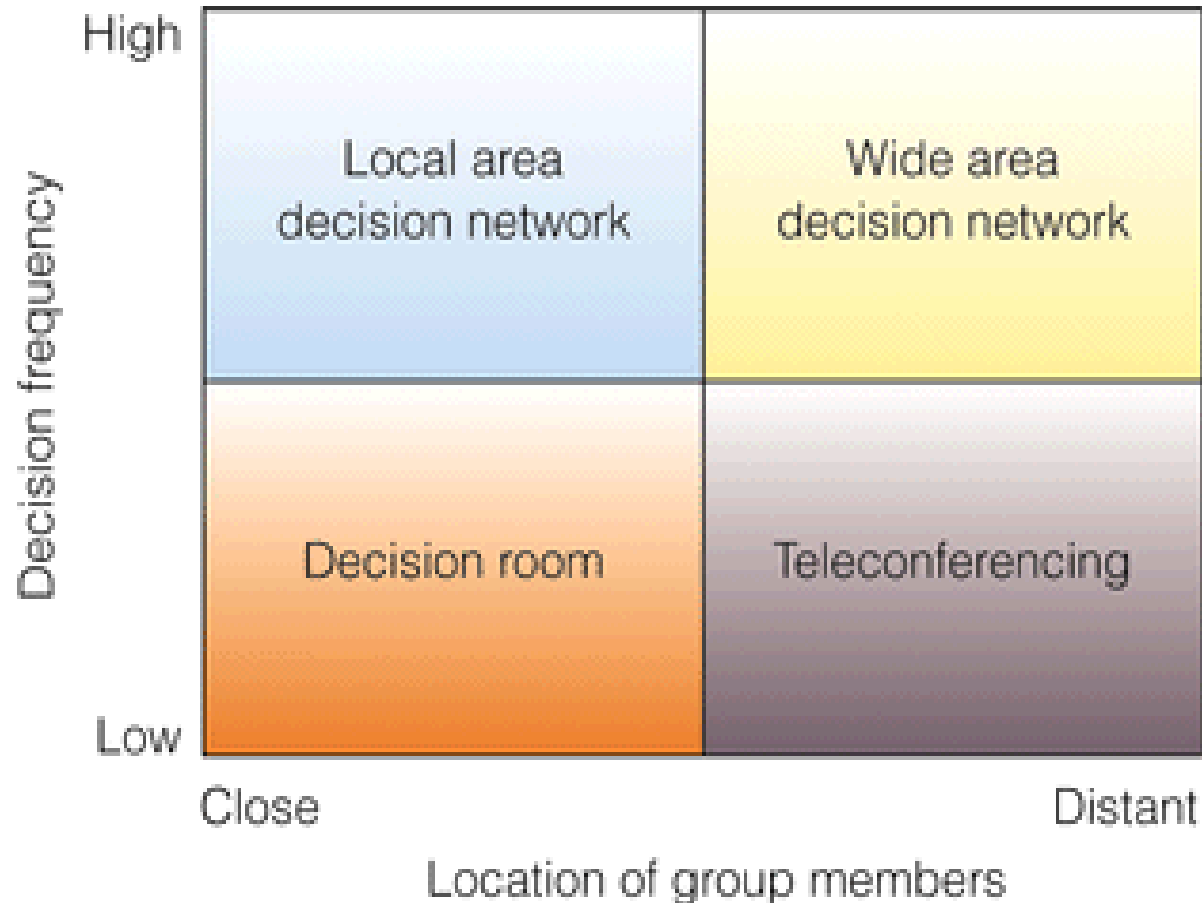
Clinical Group DSS

	Waktu sama	Waktu berbeda
Tempat sama	<ul style="list-style-type: none">• GSS dalam suatu ruang keputusan.• GSS berbasis web.• Presentasi berbasis multimedia.• Penggunaan whiteboard.• Berbagi dokumen.	<ul style="list-style-type: none">• GSS dalam suatu ruang keputusan.• GSS berbasis web.• Manajemen aliran kerja.• Berbagi dokumen.• Email, vmail.• <i>Video conferencing playback.</i>
Tempat berbeda	<ul style="list-style-type: none">• GSS berbasis web.• Penggunaan whiteboard.• Berbagi dokumen.• <i>Video conferencing.</i>• <i>Audio conferencing.</i>• <i>Computer conferencing.</i>• Email, vmail.	<ul style="list-style-type: none">• GSS berbasis web.• Penggunaan whiteboard.• Berbagi dokumen.• Email, vmail.• Manajemen aliran kerja.• <i>Computer conferencing with memory.</i>• <i>Video conferencing playback.</i>

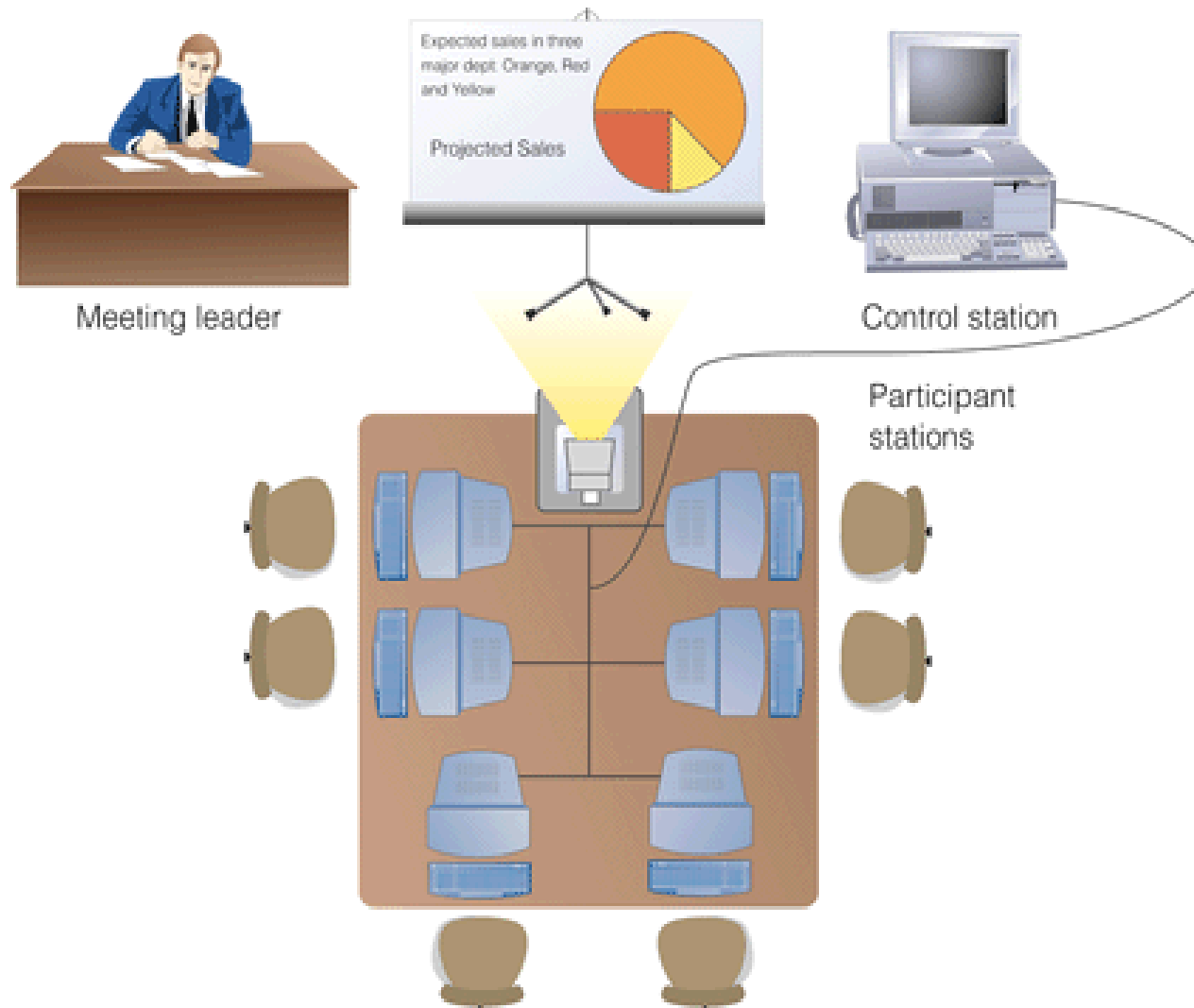
Topologi GDSS



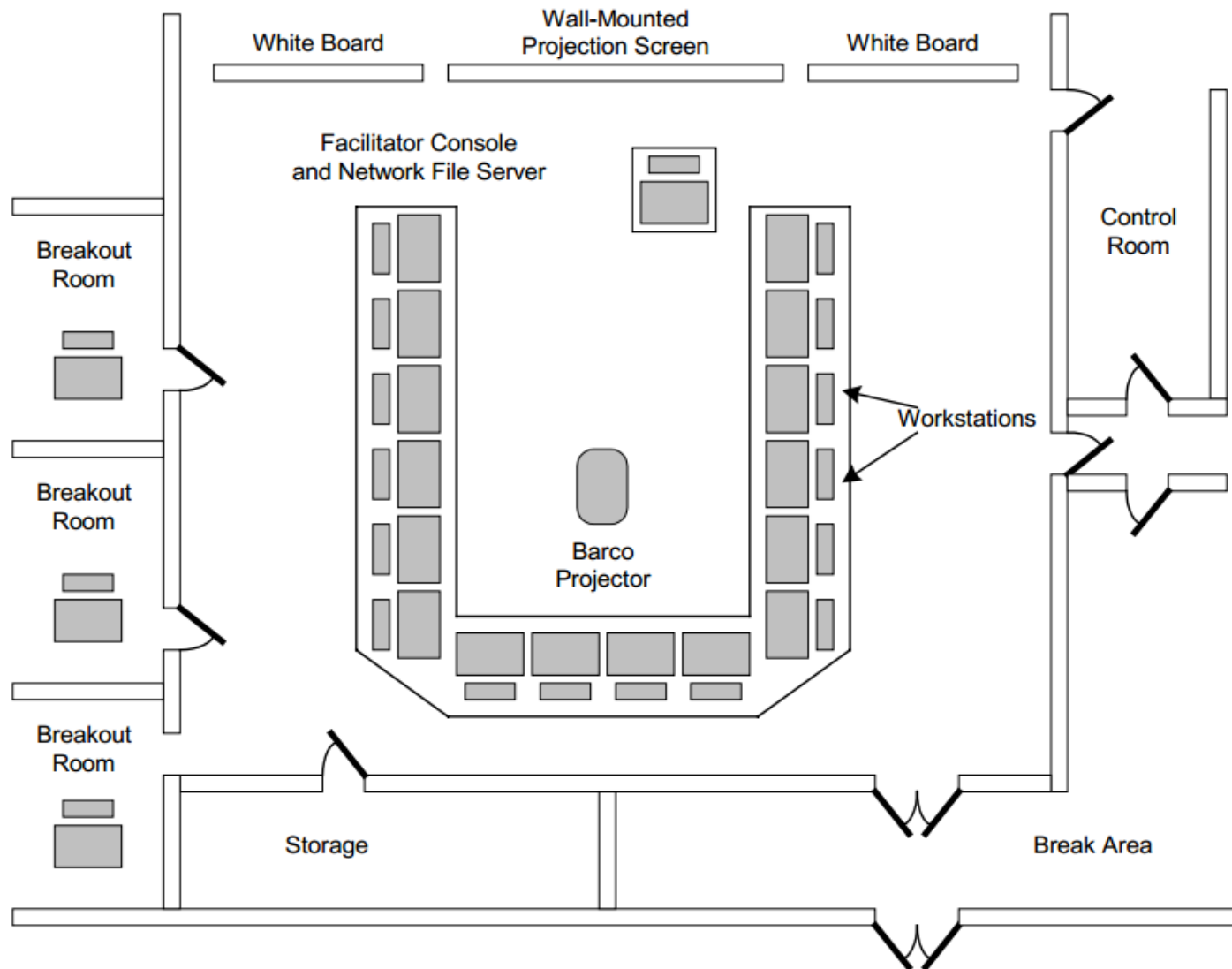
GDSS Alternatives



GDSS Decision Room



Ruang Pengambilan Keputusan



Contoh Ruang Pengambilan Keputusan



Karakteristik GDSS

- Karakteristik GDSS
 - Rancangan khusus: komunikasi yang efektif & group decision making
 - Mudah digunakan
 - Fleksibel: mengakomodasi perbedaan perspektif
 - Anonymous input: tidak mengekspos nama individu
 - Parallel communication
 - Decision-making support
 - Mengurangi kelakuan kelompok yang bersifat negatif: dibutuhkan fasilitator.
 - Automated record keeping

Keuntungan GDSS

- Adanya komunikasi secara paralel antar anggota kelompok.
- Menawarkan kesempatan yang sama untuk memberikan ide dan opini.
- Mengeliminasi adanya dominasi dari sebagian anggota kelompok.
- Dapat diketahui dengan cepat adanya persetujuan dan ketidaksetujuan terhadap opini dalam anggota kelompok.
- Membantu mengatur jadwal pertemuan
- Menyediakan kapabilitas dokumentasi otomatis secara efektif.

Pengaruh GDSS dalam Kelompok

- Pengaruh GDSS dalam kelompok
 - GDSS menitikberatkan pada perbaikan kerja anggota kelompok atau penyelesaian masalah dalam kelompok.
 - GDSS meningkatkan kualitas dalam perbaikan proses pengambilan keputusan kelompok.
 - GDSS meningkatkan pencapaian konsensus.

Level Group Decision Support

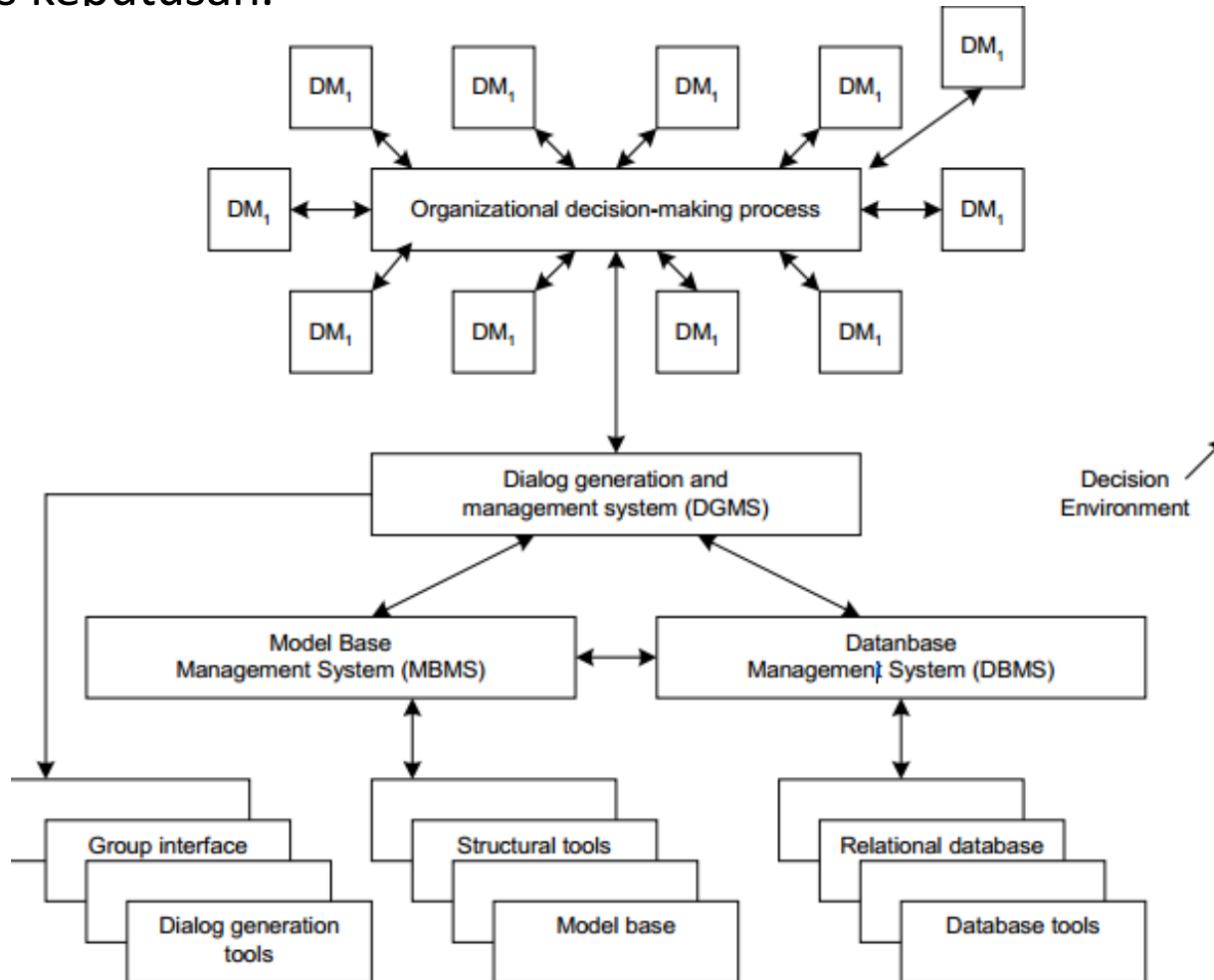
- **Level 1 GDSS (Proses)** – menyediakan fitur-fitur teknis yang bertujuan untuk mengurangi adanya kendala komunikasi seperti voting dan pertukaran pesan.
- **Level 2 GDSS (Pengambilan Keputusan)** menyediakan pemodelan keputusan dan teknik-teknik pengambilan keputusan kelompok yang bertujuan untuk mengurangi adanya ketidakpastian yang terjadi dalam proses pengambilan keputusan kelompok.
- **Level 3 GDSS (Aturan)** – berisi saran-saran dalam menyeleksi dan menyusun aturan yang dapat diaplikasikan selama pertemuan.

Level 1 (Dukungan Proses)

- Pengiriman pesan/message elektronik diantara para anggota grup
- Hubungan jaringan ke setiap PC anggota pada anggota grup yang lain, fasilitator, database.
- Layar publik tersedia di setiap terminal anggota grup.
- Masukan dalam hal pemungutan suara dan ide yang terlindungi siapa pencetusnya untuk meningkatkan partisipasi anggota grup.
- Pentimpulan dan penampilan ide dan opini, termasuk ringkasan secara statistic dan tampilan jalannya pemungutan suara.

Level 2 Dukungan Pengambilan Keputusan

- Pada level ini software dibekali kemampuan dalam pemodelan dan analisis keputusan.



Perancangan GDSS

- Ada 3 faktor yang perlu diperhatikan dalam merancang GDSS:
 - Ukuran kelompok
 - Adanya atau ketiadaannya interaksi secara tatap muka.
 - Tugas yang akan dihadapi oleh kelompok.

Fitur-fitur Dasar GDSS

- Pembentukan, modifikasi dan penyimpanan file data & teks bagi anggota kelompok.
- Pemrosesan kata (word processing) untuk memformat dan mengedit teks.
- Fasilitas pembelajaran.
- Fasilitas Help secara online
- Worksheets, spreadsheets, decision trees, dan lain-lain yang membutuhkan display secara grafis.
- Manajemen basisdata

Fitur-Fitur Kelompok

- Rangkuman ide atau hasil pemungutan suara anggota kelompok baik secara numeris maupun grafis.
- Menu untuk input teks, data atau pemungutan suara untuk anggota kelompok.
- Program untuk prosedur-prosedur kelompok tertentu.
- Metode untuk menganalisis interaksi atau pendapat awal.
- Transmisi teks dan data dalam anggota kelompok

Clinical Group DSS - Konsensus

- Cara memperoleh konsensus:
 - Voting
 - Agregasi
 - *Ordered Weighted Averaging (OWA)*
 - *Induced Ordered Weighted Averaging (IOWA)*
 - *Importance Induced Ordered Weighted Averaging (I-IOWA)*
 - *Consistency Induced Ordered Weighted Averaging (C-IOWA)*

Clinical Group DSS - Konsensus

Alternatif	Pengambil Keputusan			Keputusan Akhir	Nilai Konsensus
	P ₁	P ₂	P ₃		
A	√	√		√	66,67%
B			√		
C					

Clinical Group DSS - Konsensus

Alternatif	Pengambil Keputusan			Keputusan Akhir	Nilai Konsensus
	P ₁	P ₂	P ₃		
A	sangat yakin	sangat yakin	yakin	√	100%
B	yakin	yakin	sangat yakin		
C	tidak yakin	sangat tidak yakin	tidak yakin		