

PENGEMBANGAN SISTEM PENDUKUNG KEPUTUSAN

Pendahuluan.

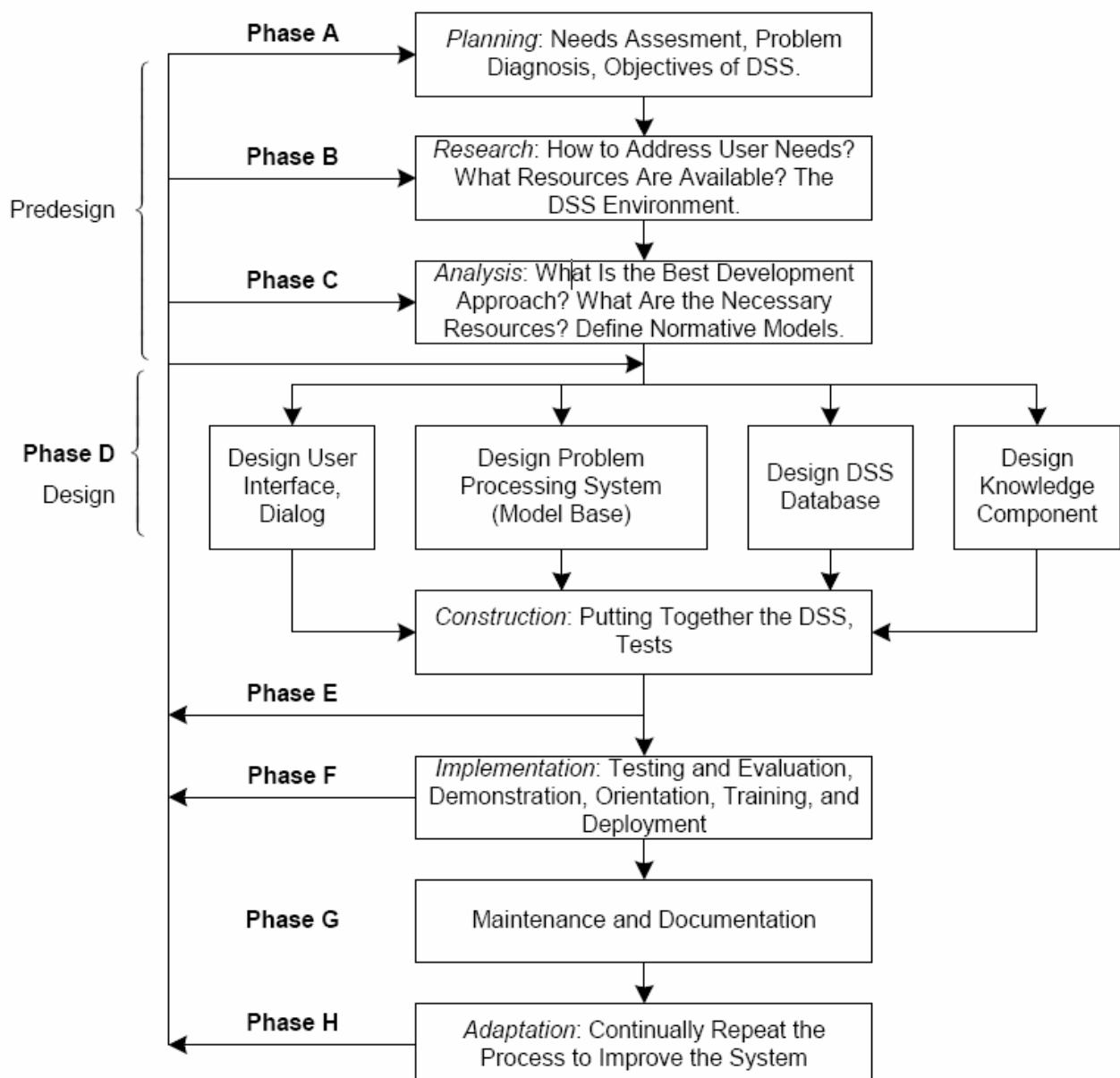
- Membangun sebuah DSS, apalagi yang besar, merupakan proses yang rumit.
- Melibatkan hal-hal: teknis (hardware, jaringan) dan perilaku (interaksi manusia-mesin, dampak DSS pada individu).
- Agar lebih mudah membangun DSS bisa digunakan bahasa khusus (misal CASE TOOLS).

Strategi Pengembangan.

1. Tulis DSS dengan bahasa pemrograman umum: Pascal, Delphi, C, C++, C#, Java, dan lainnya.
2. Menggunakan 4GL: data-oriented language, spreadsheets, dan financial-oriented language.
3. Menggunakan DSS Generator: Excel, Lotus 1-2-3, Quattro Pro, Express. Generator lebih efisien dari 4GL tapi ini tergantung juga pada batasannya.
4. Menggunakan DSS Generator khusus (domain specific): Commander FDC untuk budgeting & financial analysis, EFPM untuk kalangan perguruan tinggi.
5. Mengembangkan DSS dengan metodologi CASE. Memiliki jaminan kualitas yang memadai.
6. Untuk DSS yang kompleks, bisa mengintegrasikan pendekatan-pendekatan di atas.

Proses Pengembangan DSS.

- Pengembangan suatu DSS terkait juga dengan struktur permasalahan: tak terstruktur, semi terstruktur, ataupun terstruktur. Berturut-turut ES/DSS, DSS, MIS bisa digunakan untuk ini.
- Di bawah ini disajikan bagan fase-fase pengembangan suatu DSS:



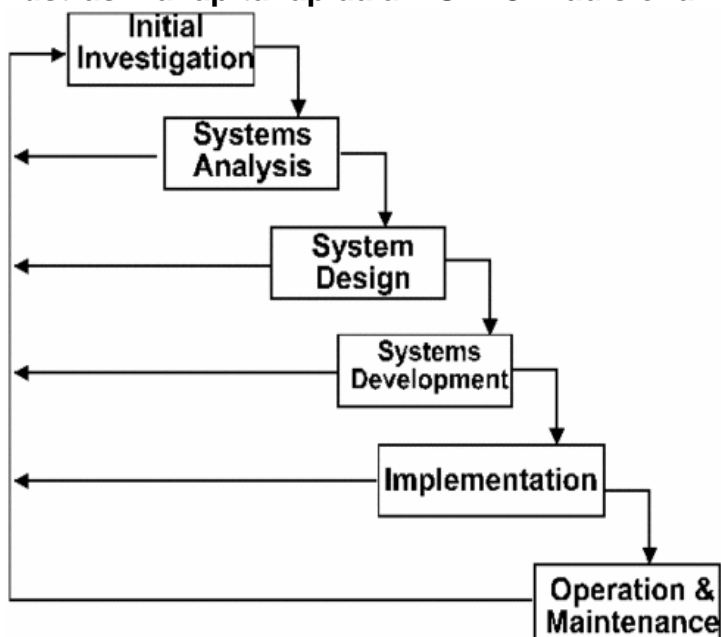
Proses Pengembangan: Life Cycle vs Prototyping.

Pembangunan DSS dilakukan dalam berbagai cara. Dibedakan antara pendekatan life cycle (daur hidup) dan iterative process (proses berulang).

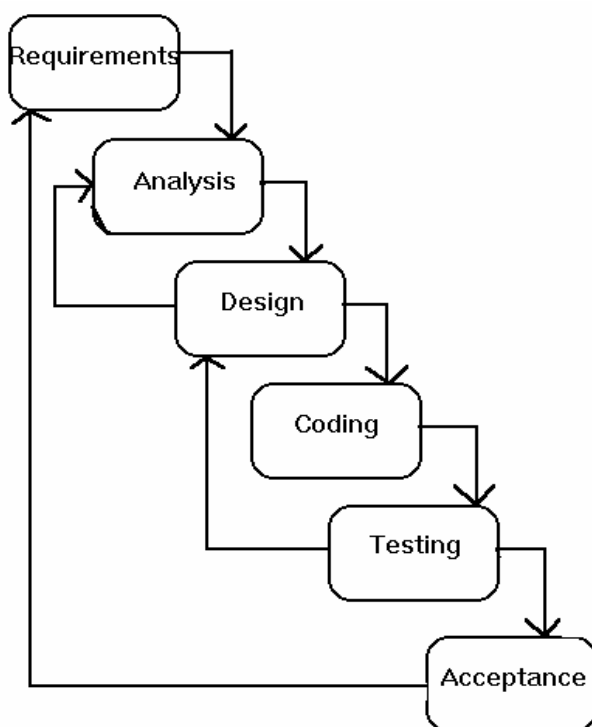
Pendekatan SDLC (System Development Life Cycle).

- Asumsi dasarnya adalah kebutuhan informasi dari suatu sistem dapat ditentukan sebelumnya.
- IRD (Information Requirements Definition) adalah pendekatan formal yang digunakan oleh sistem analis.
- IRD secara tradisional merupakan kombinasi analisis logik dengan pengamatan perilaku pemrosesan informasi.
- IRD bisa juga melibatkan CSF (Critical Success Factors).
- DSS didesain untuk membantu pengambilan keputusan para manajer pada masalah yang jelek strukturnya. Di satu sisi memahami kebutuhan user adalah hal yang sulit. Sehingga perlu diterapkan adanya bagian pembelajaran dalam desain atau proses kita. Dari situ, diharapkan user belajar mengenai masalah atau lingkungannya sehingga dapat mengidentifikasi kebutuhan informasi baru dan yang tak diantisipasi sebelumnya.

Ilustrasi Tahap-tahap dalam SDLC Tradisional



http://exonous.typepad.com/mis/2004/02/systems_develop.html



The waterfall model (Systems Development Life Cycle)

<http://www.albany.edu/acc/courses/fall97/acc681/ch7.html>

Pendekatan Prototyping Evolusioner.

- Pendekatan prototyping disebut juga proses evolusioner (evolutionary process), proses berulang (iterative process), atau cukup disebut prototyping saja. Nama lainnya adalah middle-out process (proses sementara), adaptive design (desain adaptif) dan incremental design (desain berkelanjutan).
- Proses desain berulang ini mengkombinasikan 4 fase utama SDLC tradisional (analisis, desain, konstruksi, dan implementasi) ke dalam 1 langkah yang diulang-ulang.
- Proses berulang terdiri dari 4 tugas, seperti di bawah ini:
 1. Memilih submasalah penting yang akan dibangun pertama kali.
 2. Mengembangkan sistem yang kecil, tapi berguna, dalam membantu pengambil keputusan.
 3. Mengevaluasi sistem terus menerus.
 4. Menghaluskan, mengembangkan, dan memodifikasi sistem secara berulang.

Pengembangan DSS Berbasis Tim dan Berbasis User.

- 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 (intermediaries), DSS builder, tenaga ahli, dan berbagai tool. Berbagai individu dalam setiap kategori tadi, sehingga ukuran tim menjadi besar dan komposisinya sering berubah seiring berjalannya waktu. Intinya dengan berbasis tim, maka pembangunan DSS menjadi kompleks, lama, dan prosesnya memakan biaya.
- Pendekatan lainnya adalah membangun DSS berbasiskan user. Dimulai mulai tahun 1980-an, seiring pesatnya perkembangan di bidang PC (Personal Computer), jaringan komunikasi komputer, berkurangnya biaya hardware maupun software.
- Enterprise-wide computing serta kemudahan akses data dan pemodelan berarsitektur client/server juga mendukung pengembangan DSS berbasis user.
- Tentu saja kedua pendekatan ini bisa dikombinasikan, untuk mendapat kinerja yang diinginkan.

Pengembangan DSS Berbasis Tim.

Menentukan DSS Group.

Secara organisasi penempatan DSS Group bisa dimana-mana, umumnya pada lokasi:

1. Dalam departemen IS (Information Services).
2. Executive Staff Group.
3. Dalam wilayah keuangan atau fungsi lainnya.
4. Dalam departemen rekayasa industri.
5. Dalam kelompok manajemen pengetahuan (Management Science Group).
6. Dalam kelompok pusat informasi (Information Center Group).

Komputasi End-User dan Pengembangan DSS Berbasis User.

Komputasi End-User.

Pengembangan DSS berbasis user berelasi secara langsung kepada komputasi end-user. Definisinya adalah: pengembangan dan penggunaan sistem informasi berbasis komputer oleh orang-orang di luar wilayah sistem informasi formal. Definisi ini melibatkan juga manajer dan profesional yang menggunakan komputer pribadi, pengolah kata yang digunakan oleh sekretaris, e-mail yang digunakan oleh CEO, dan sistem time-sharing yang digunakan oleh ilmuwan dan peneliti.

Pengembangan DSS Berbasis User: Keuntungan dan Resikonya.

Berbagai keuntungan yang bisa didapat user bila dia sendiri yang membangun DSS:

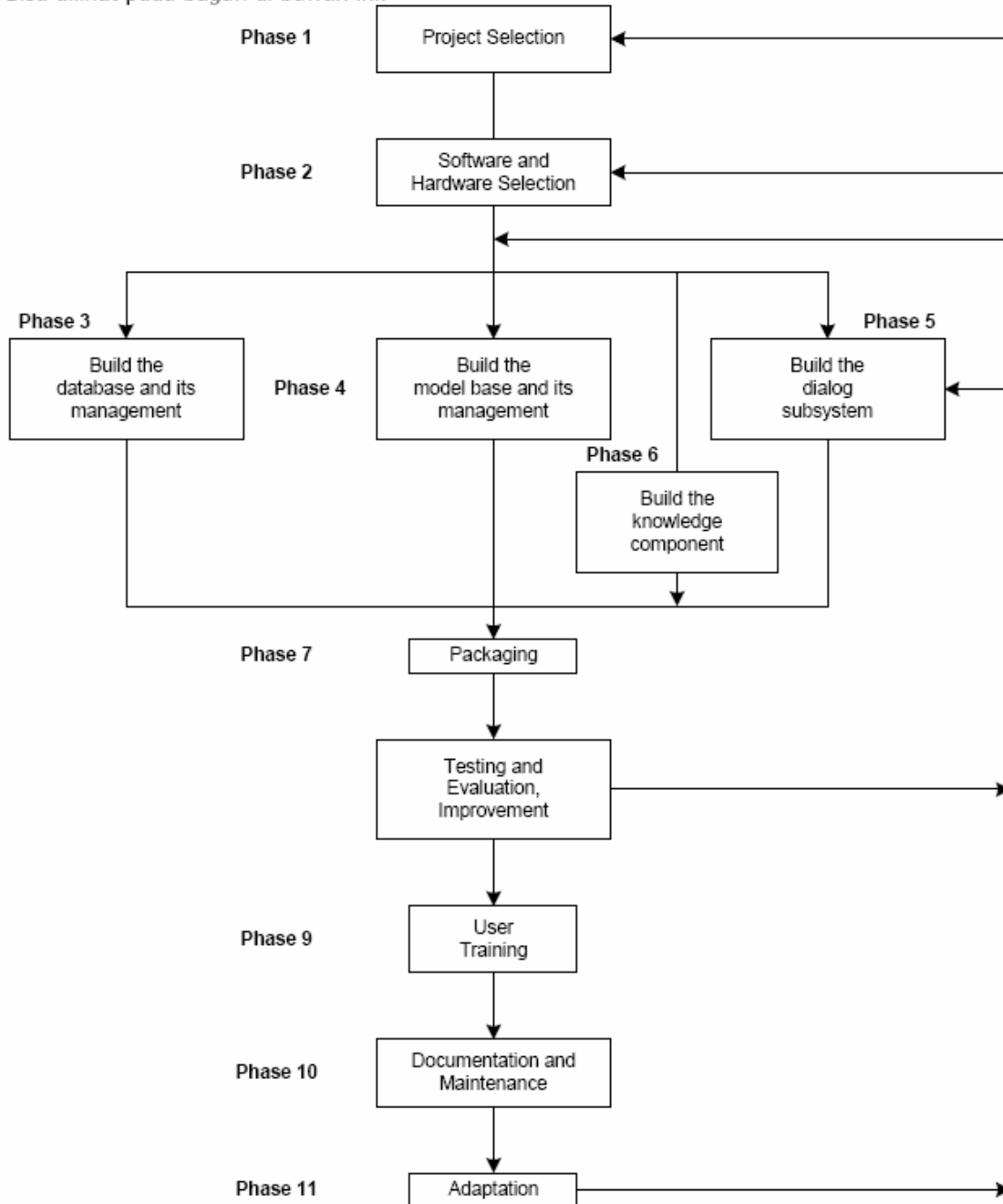
1. Waktu penyelesaiannya singkat.
2. Syarat-syarat spesifikasi kebutuhan sistem tak diperlukan.
3. Masalah implementasi DSS dapat dikurangi.
4. Biayanya sangat rendah.

Resikonya adalah:

1. Kualitasnya bisa tak terjaga.
2. Resiko potensial kualitas dapat diklasifikasikan dalam 3 kategori: (a) tool dan fasilitas di bawah standar, (b) resiko yang berhubungan dengan proses pengembangan (contoh: ketidakmampuan mengembangkan sistem yang bisa bekerja, pengembangan sistem yang menghasilkan hasil yang salah), dan (c) resiko manajemen data (misal: kehilangan data).

Proses Pembuatan.

Bisa dilihat pada bagan di bawah ini:



DSS Generator.

DSS Generator mengkombinasikan kemampuan berbagai aplikasi umum dalam 1 program.

Di bawah ini adalah program-program yang merupakan “bahan baku” bagi paket terintegrasi:

- Spreadsheet.
- Manajemen Data.
- Pengolah kata.
- Komunikasi.
- Grafis bisnis.
- Kalender (manajemen waktu).
- Desk management.
- Manajemen projek.

Contoh dari paket terintegrasi ini adalah: Lotus 1-2-3, Microsoft Excel.

Pemilihan DSS Generator dan Tool Software Lainnya.

Berbagai pertanyaan yang harus dijawab oleh suatu organisasi yang akan menggunakan DSS Generator: (1) generator seperti apa yang akan digunakan, (2) hardware seperti apakah yang dipakai untuk menjalankannya, (3) sistem operasi seperti apa yang akan digunakan, (4) jaringan seperti apakah yang akan dipakai untuk menjalankannya.

Dengan kemampuan PC yang luar biasa sekarang ini, software DSS lebih banyak ditemui pada jenis komputer mikro. Kemudian dengan adanya program-program berbasis Windows, membuat DSS menjadi lebih disukai karena kemudahan penggunaannya.

Pemilihan Software.

Tool software dasar yang patut dipertimbangkan adalah:

- Fasilitas database relasional dengan fasilitas pembuatan laporan yang baik dan fasilitas pemilihan data setiap saat.
- Bahasa penghasil grafis.
- Bahasa pemodelan.
- Bahasa analisis data statistikal umum.
- Bahasa khusus yang lain (misal: untuk membangun simulasi).
- Bahasa pemrograman (generasi ketiga).
- Tool pemrograman berorientasi objek.
- Tool pembangun ES.
- Jaringan.
- CASE tools.

Kesimpulan.

- DSS dikembangkan dengan proses pengembangan yang unik berdasarkan prototyping.
- Langkah utamanya adalah: perencanaan, riset, analisis, desain, konstruksi, implementasi, perawatan, dan adaptasi.
- Pendekatan berulang (prototyping) paling umum digunakan dalam DSS, karena kebutuhan informasi tak dapat diketahui dengan tepat pada awal proses.
- DSS dapat dibangun oleh tim maupun individu.
- Pembangunan DSS dengan tim mengikuti proses terstruktur, termasuk perencanaan, pemilihan software yang sesuai (generator jika dibutuhkan), dan hardware.
- Bagian utama dari komputasi end-user adalah pembangunan DSS untuk dukungan personal yang dilakukan oleh individu.
- Keuntungan utama orang-orang yang membangun DSS-nya sendiri adalah: waktu penyelesaiannya singkat, familiar dengan kebutuhannya, biaya rendah, dan implementasinya lebih mudah.
- Pengembangan DSS berbasis user bisa juga berkualitas rendah, karena itu kontrol yang cukup dapat memperbaiki situasi tersebut.
- Kebanyakan DSS dibangun dengan generator pengembangan DSS atau dengan tool-tool pengembangan 4GL tak terintegrasi.
- Terdapat banyak sekali tool dan generator di pasaran. Pemilihan yang sesuai untuk membangun DSS tertentu haruslah didesain dengan cermat.
- Banyak DSS dibangun dalam lingkungan Windows. Windows membuatnya mungkin untuk membangun DSS dengan cepat dan murah.