

Pengembangan Sistem Informasi

Sistem Informasi
Universitas Gunadarma
2012/2013

- Menjelaskan definisi pengembangan sistem dan tahapan pada *System Development Life Cycle (SDLC)*.
- Menjelaskan macam-macam metodologi pengembangan sistem baik yang secara tradisional maupun yang non tradisional.
- Menjelaskan macam-macam pendekatan di dalam pengembangan sistem.
- Menjelaskan yang terlibat didalam pengembangan sistem.
- Menggambarkan 2 pendekatan yang digunakan untuk mengembangkan sistem informasi : metode terstruktur dan metode *object-oriented*.
- Menggambaran beberapa variasi dari *System Development Life Cycle (SDLC)*.

Tujuan

- Pengembangan sistem informasi sering disebut sebagai **proses pengembangan sistem (*System Development*)**.
- **Pengembangan sistem** didefinisikan sebagai :
 1. Aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk **menyelesaikan persoalan (problem)** organisasi atau **memanfaatkan kesempatan (opportunities)** yang timbul.
 2. Menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang ada.

Pengembangan Sistem

- Alasan pengembangan sistem adalah :
 1. Adanya permasalahan : adanya ketidakberesan dan pertumbuhan organisasi.
 2. Untuk meraih kesempatan-kesempatan.
 3. Adanya instruksi (pimpinan dan pemerintah).

Pengembangan Sistem

- Untuk menghasilkan sistem informasi tersebut terdiri dari :
 1. ***System Analysis*** : upaya mendapatkan gambaran bagaimana sistem bekerja dan masalah-masalah apa saja yang ada pada sistem.
 2. ***System Development*** : langkah-langkah mengembangkan sistem informasi yang baru berdasarkan gambaran cara kerja sistem & permasalahan yang ada.

Pengembangan Sistem

- Sumberdaya (*resource*) yang dibutuhkan untuk pengembangan sistem informasi adalah :

1. *Development Team* (Tim Pengembang)

2. *Development Toolkit*

- * *Methods*

- * *Technique*

- * *Tools*

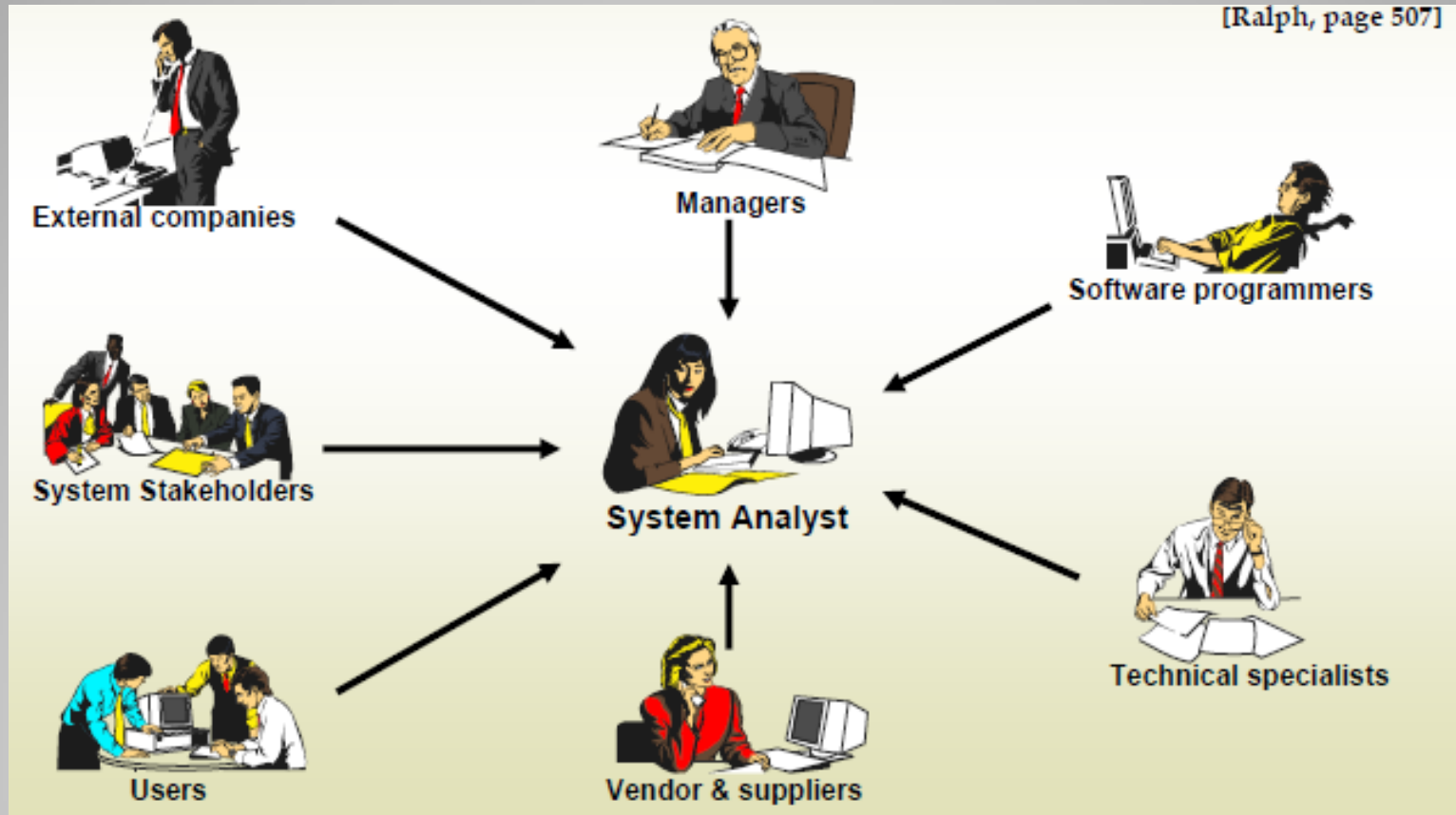
- * *IT Resources : hardware, software, communications technology*

Pengembangan Sistem

- Untuk mengembangkan sistem informasi perlu dibentuk **Tim Pengembang** (*Development Team*) yang terdiri dari :
 1. *Project Manager*
 2. *Business Analyst*
 3. *System Analyst*
 4. *Programmer*
 5. *Change Manager*

Pengembangan Sistem

- Peranan System Analyst



Pengembangan Sistem

- *Development Toolkit* terdiri dari :
 1. *Methods (Metoda)* : suatu kerangka kerja (*framework*) untuk membangun suatu sistem secara menyeluruh.
 2. *Technique (Teknik)* : bagian dari metoda untuk melaksanakan bagian dari tahapan pengembangan sistem.
 3. *Tools (Alat bantu)* : suatu perangkat yang dapat digunakan untuk membantu pelaksanaan pengembangan sistem. Tools harus dapat mendukung (*support*) metoda dan teknik yang digunakan dalam proses pengembangan sistem.

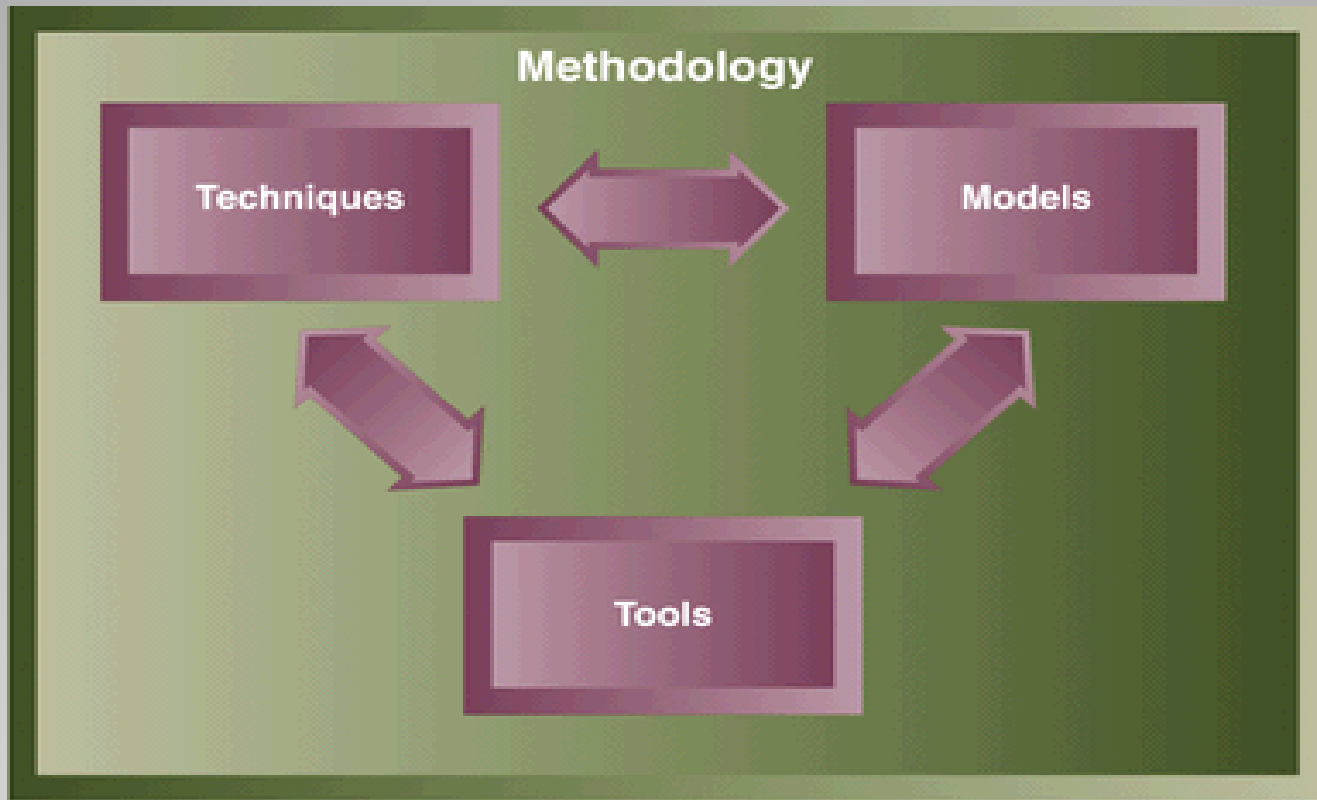
Pengembangan Sistem

Methodology

Techniques

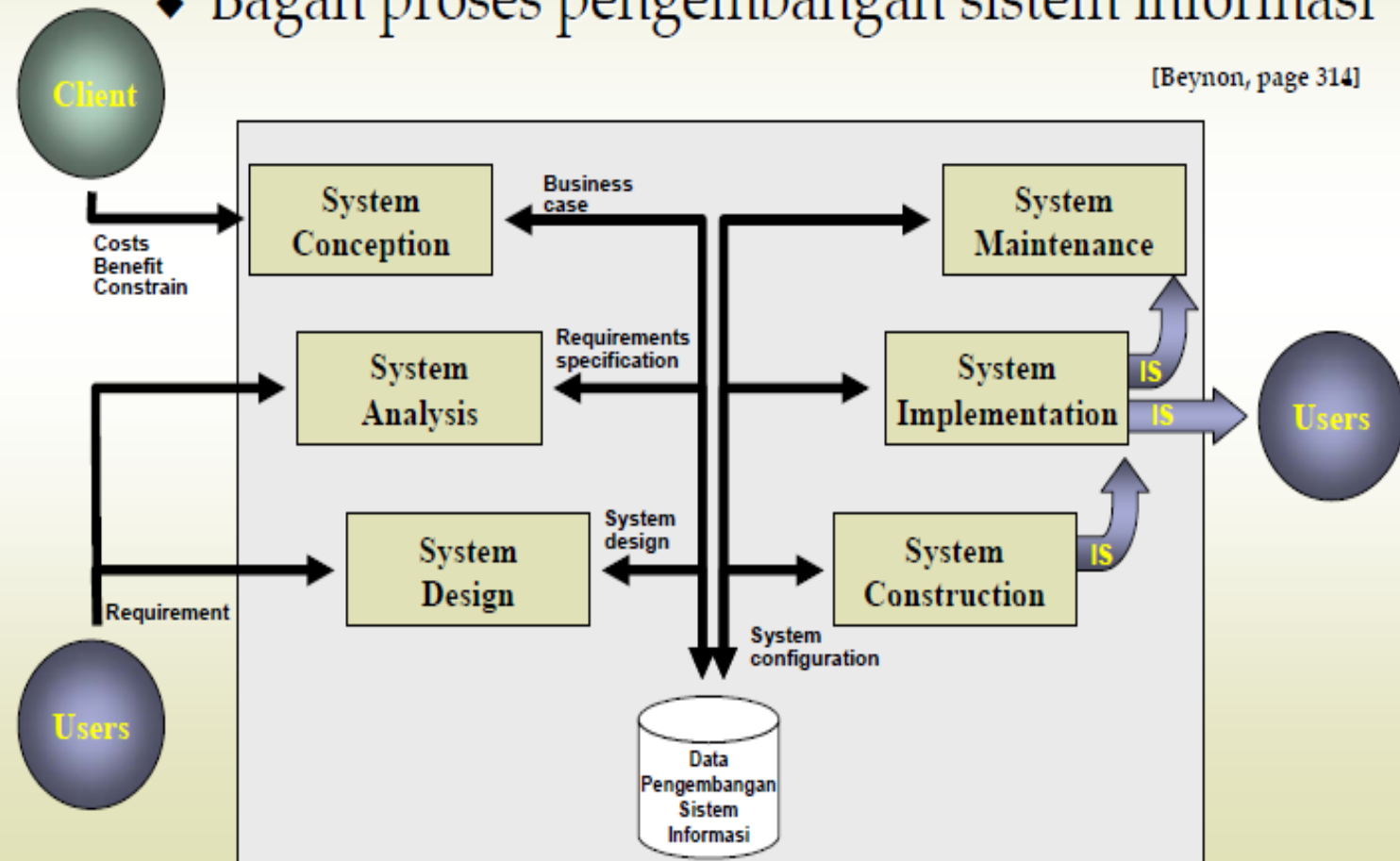
Models

Tools



◆ Bagan proses pengembangan sistem informasi

[Beynon, page 314]



Pengembangan Sistem

- **Metodologi pengembangan sistem** adalah suatu proses pengembangan sistem yang formal dan presisi yang mendefinisikan serangkaian aktivitas, metode, *best practices* dan *tools* yang terautomasi bagi para pengembang dan manager proyek dalam rangka mengembangkan dan merawat sebagai keseluruhan sistem informasi atau *software*.

Metodologi Pengembangan Sistem

- Alasan perlunya metodologi pengembangan sistem adalah :
 1. Menjamin adanya konsistensi proses.
 2. Dapat diterapkan dalam berbagai jenis proyek.
 3. Mengurangi resiko kesalahan dan pengambilan jalan pintas.
 4. Menuntut adanya dokumentasi yang konsisten yang bermanfaat bagi personal baru dalam tim proyek.

Metodologi Pengembangan Sistem

1. *System Development Life Cycle (SDLC)*
2. *Prototyping*
3. *Computer-Aided Software Engineering (CASE) Tools*
4. *Rapid Application Development (RAD)*
5. *Joint Application Design (JAD)*
6. *Agile Methodologies*
7. *eXtreme Programming (XP)*
8. *Model-Driven Development (MDD)*
9. *Framework for the Application of System Thinking (FAST)*
10. *Pembelian Software Aplikasi*

Metode Pengembangan Sistem

- SDLC merupakan metode pengembangan sistem paling tua dan sangat cocok untuk pengembangan sistem yang besar.
- Tidak sesuai atau tidak terlalu disarankan untuk *small scale project* karena :
 1. Banyak memerlukan sumber daya.
 2. Tidak fleksibel.
 3. Sulit untuk melakukan perubahan aplikasi dengan pengambilan keputusan yang cepat.
- SDLC lebih dari sekedar fase
 1. Prinsip Manajemen
 2. Perencanaan dan Pengawasan
 3. Pengorganisasian dan Penjadwalan
 4. Penyelesaian Masalah

System Development Life Cycle (SDLC)

- **Keuntungan :**

1. Adanya laporan setiap akhir fase sehingga memudahkan adanya kontrol/pengawasan.
2. Mudah melakukan dokumentasi.
3. Dokumentasi secara formal sehingga memudahkan penelusuran kembali terhadap kebutuhan bisnis.

- **Kelemahan :**

1. Pengguna mendapatkan produk sesuai dengan pemahaman pengembang, yang belum tentu sesuai kebutuhan.
2. Dokumentasi mahal dan menghabiskan waktu dalam pembuatan, dan selalu berubah/mengalami perbaikan.

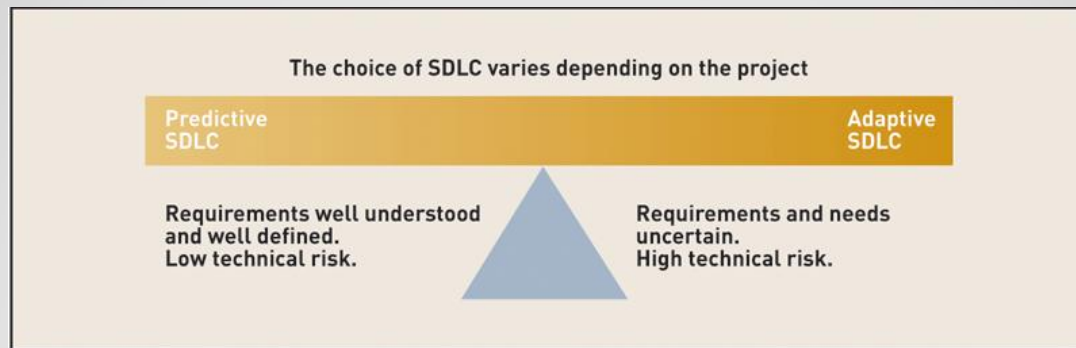
System Development Life Cycle (SDLC)

- **System development life cycle (SDLC)**

Menyediakan keseluruhan framework untuk mengelola proses pengembangan sistem, for managing systems development process.

- **2 Pendekatan Pengembangan SDLC**

1. **Predictive approach** – assumes project can be planned out in advance.
2. **Adaptive approach** – more flexible, assumes project cannot be planned out in advance.



System Development Life Cycle (SDLC)

- SDLC terdiri dari 5 fase

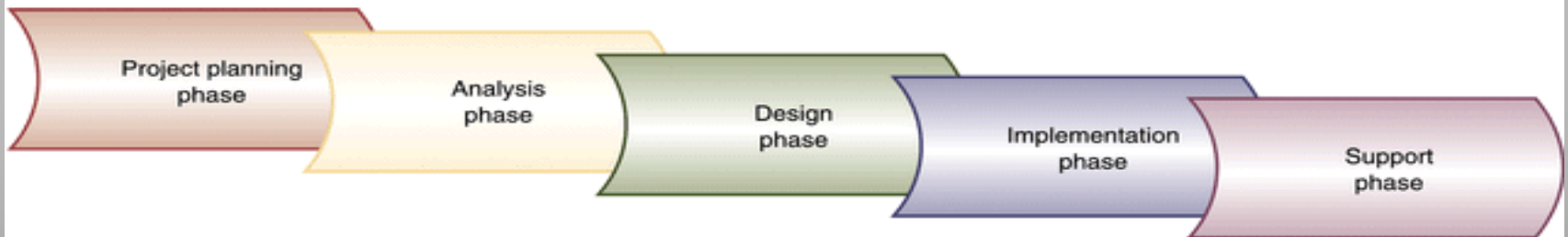
Masing masing fase terdiri dari aktivitas yang saling terkait/berhubungan.

- 3 aktivitas utama

1. **Analisa** : memahami kebutuhan bisnis.
2. **Desain** : membuat konsep solusi pengembangan sistem berbasis komputer.
3. **Implementasi** : konstruksi/pembuatan, testing, dan instalasi.

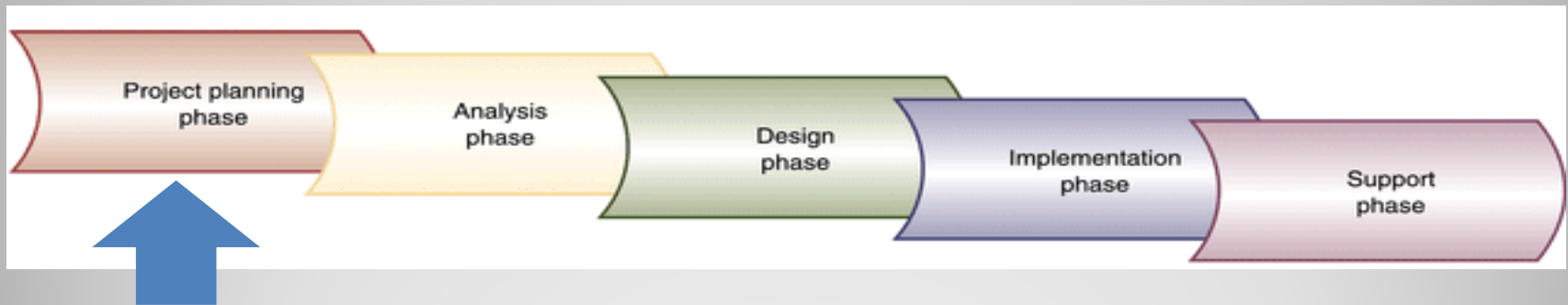
- 2 aktivitas tambahan

1. Perencanaan proyek.
2. Support/dukungan.



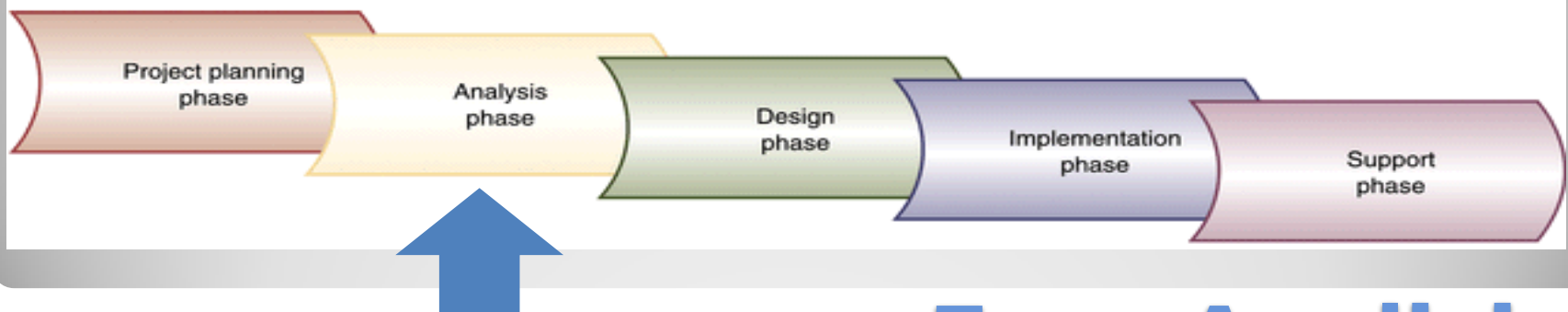
System Development Life Cycle (SDLC)

- Mendefinisikan masalah.
- Mengkonfirmasi kelayakan proyek.
- Membuat jadwal proyek.
- Menentukan staff yang terlibat dalam proyek.
- Memulai proses pengembangan proyek.



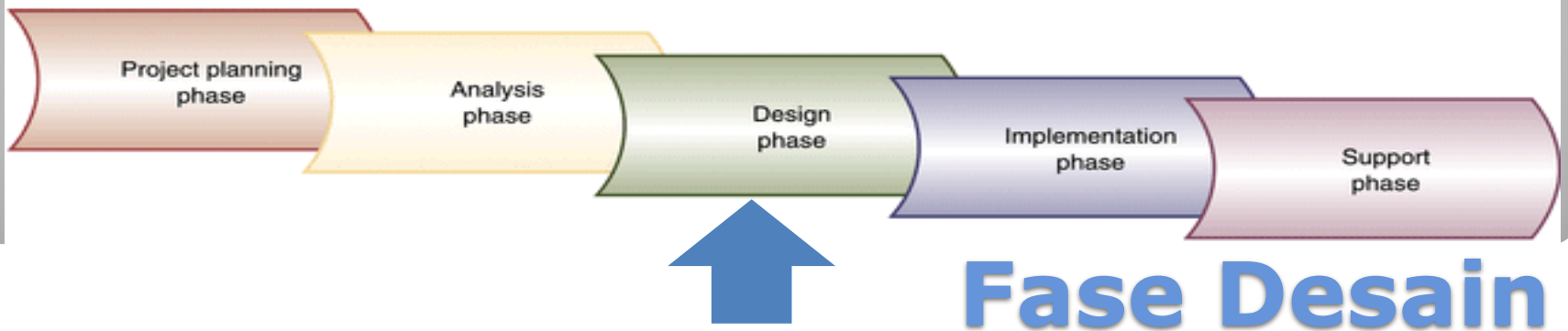
Fase Perencanaan

- Mengumpulkan informasi.
- Mendefinisikan kebutuhan - kebutuhan sistem.
- Membangun prototipe yang sesuai atau memenuhi kebutuhan sistem.
- Menentukan prioritas kebutuhan sistem.
- Membuat prototipe atas prioritas dan melakukan evaluasi terhadap alternatif yang dipilih.
- Mereview rekomendasi terhadap pihak manajemen.

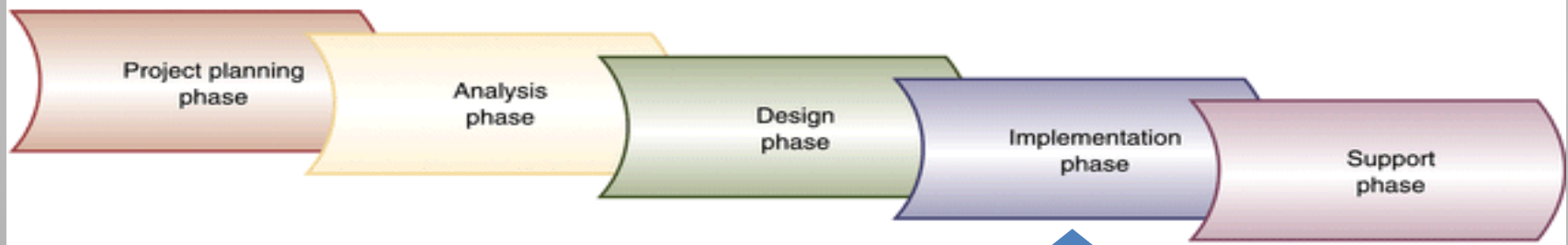


Fase Analisis

- Desain Level Tinggi (Arsitektur Sistem)
 1. Desain dan integrasi jaringan
 2. Desain arsitektur aplikasi
- Desain Level Rendah
 1. Desain user interface
 2. Desain sistem interface
 3. Desain dan integrasi database
 4. Prototype desain secara lengkap
 5. Desain dan integrasi pengawasan sistem

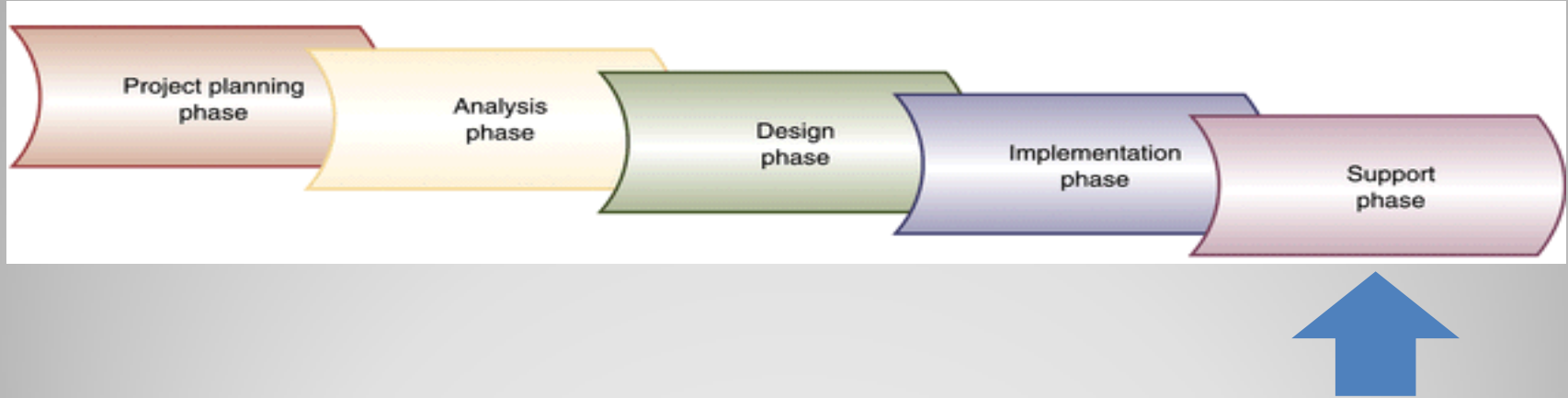


- Membangun komponen - komponen perangkat lunak.
- Melakukan verifikasi dan pengujian.
- Mengkonversi data.
- Melakukan training user dan mendokumentasikan sistem.
- Menginstall sistem.



Fase Implementasi

- Memelihara sistem.
- Memperbaiki sistem.
- Mendukung pengguna

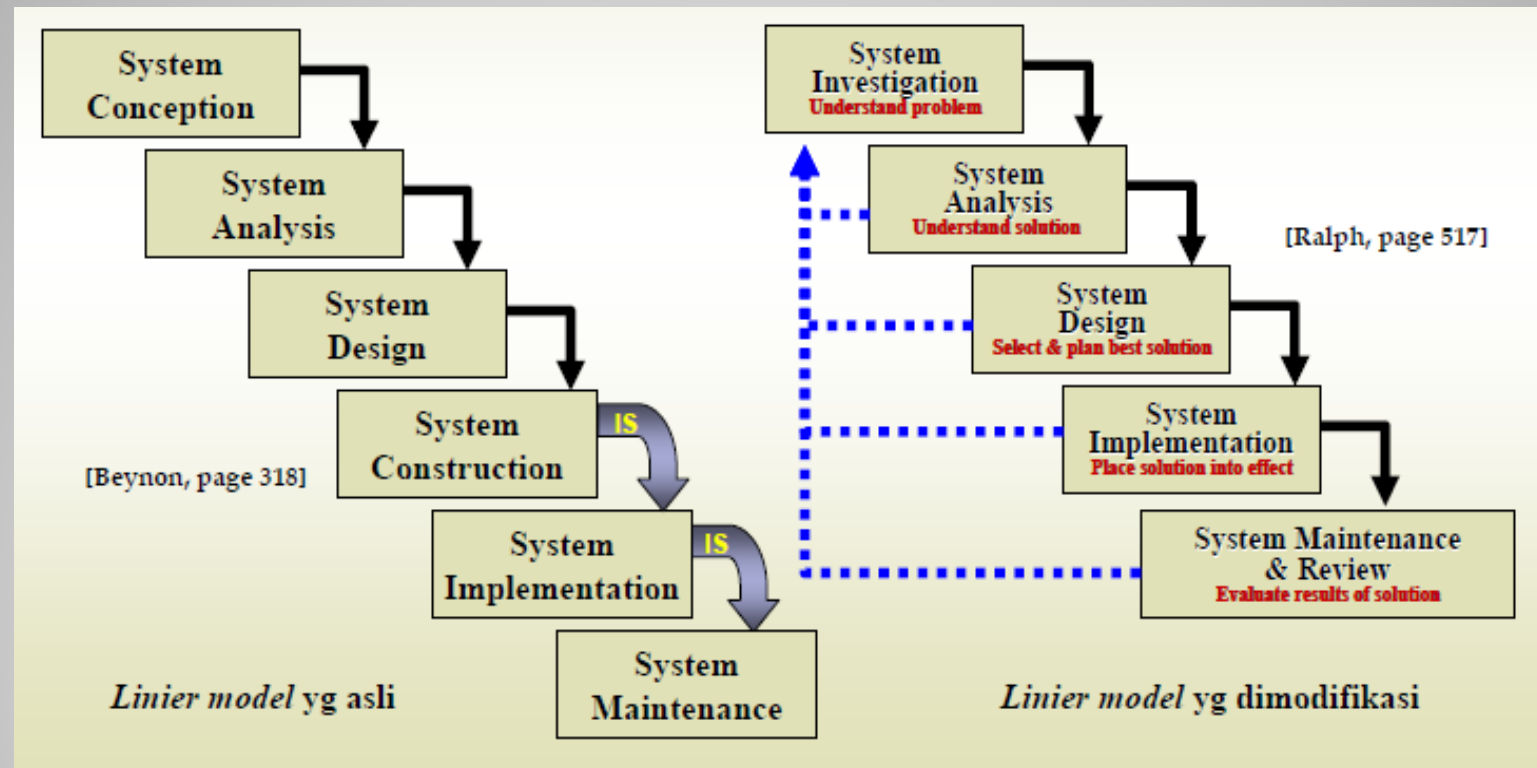


Fase Support/Dukungan

Table 1-2 Products of SDLC Phases

<i>Phase</i>	<i>Products, Outputs, or Deliverables</i>
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems; Detailed steps, or work plan, for project; Specification of system scope and planning and high-level system requirements or features; Assignment of team members and other resources; System justification or business case
Analysis	Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system; Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs); Technical, detailed specifications of all system elements (programs, files, network, system software, etc.); Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

- Setiap tahapan harus diselesaikan terlebih dahulu sebelum meneruskan ke tahapan berikutnya.



Model Waterfall (Linear Model of Development)

- Keuntungan :

1. Mudah dalam penyusunan perencanaan dan melakukan estimasi tahapan pelaksanaan maupun biaya.
2. Mudah dalam *managed* pelaksanaan tahapan pengembangan sistem Informasi.

- Kelemahan :

1. *Client & user* baru dapat mengetahui wujud dari sistem informasi yang dikembangkan pada tahap-tahap akhir (untuk *linear model asli*).
2. Pembuatan dokumen memerlukan biaya yang mahal dan waktu yang lama.
3. *User* seringkali kesulitan dalam *review intermediated product* (dokumen-dokumen).

Model Waterfall (Linear Model of Development)

- Setiap tahapan dapat dilaksanakan berulang-ulang sampai mendapatkan hasil yang diinginkan.
- Setiap loop mewakili tahap proses rekayasa terbagi menjadi 4 sektor.

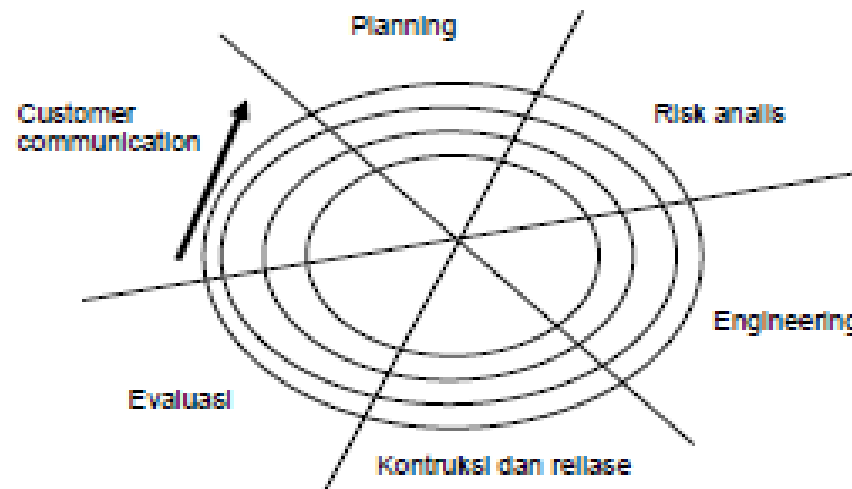


Model Spiral/Iteratif

- Metode spiral dirancang secara revolusioner dengan tahapan yang jelas, tetapi terbuka bagi partisipasi pemesan untuk ikut serta menentukan pemodelan sistem.
- Metode spiral lambat dan mahal karena setiap tahapan yang dilalui harus mengikutsertakan pemesan.
- Model spiral merupakan perbaikan dari model waterfall dan prototype. Menggabungkan keuntungan waterfall dan prototype dan memasukkan analisis resiko.

Model Spiral/Iteratif

- Spiral melibatkan proses iterasi dimana setiap iterasi bekerja pada satu level produk dimulai dari level prototype awal sampai pada level yang diinginkan.
- Setiap perpindahan level didahului analisa resiko.



Model Spiral/Iteratif

- **Metode prototyping** sebagai suatu paradigma baru dalam pengembangan sistem informasi, tidak hanya sekedar suatu evolusi dari metode pengembangan sistem informasi yang sudah ada, tetapi sekaligus merupakan revolusi dalam pengembangan sistem informasi manajemen.

Prototyping (Iterative Model of Development)

- **Keuntungan :**

1. *End user* dapat berpartisipasi aktif.
2. Penentuan kebutuhan lebih mudah diwujudkan.
3. Mempersingkat waktu pengembangan sistem informasi dan relatif lebih mudah dibangun.
4. Kesalahan dan kelalaian dalam pengembangan dapat segera diketahui.

- **Kelemahan :**

1. Proses analisis dan perancangan terlalu singkat.
2. Mengesampingkan alternatif pemecahan masalah.
3. Biasanya kurang fleksible dalam menghadapi perubahan.
4. *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
5. *Prototype* terlalu cepat selesai.
6. Dokumentasi seringkali tidak lengkap.

Prototyping (Iterative Model of Development)

- Karakteristik metode prototyping meliputi langkah-langkah :

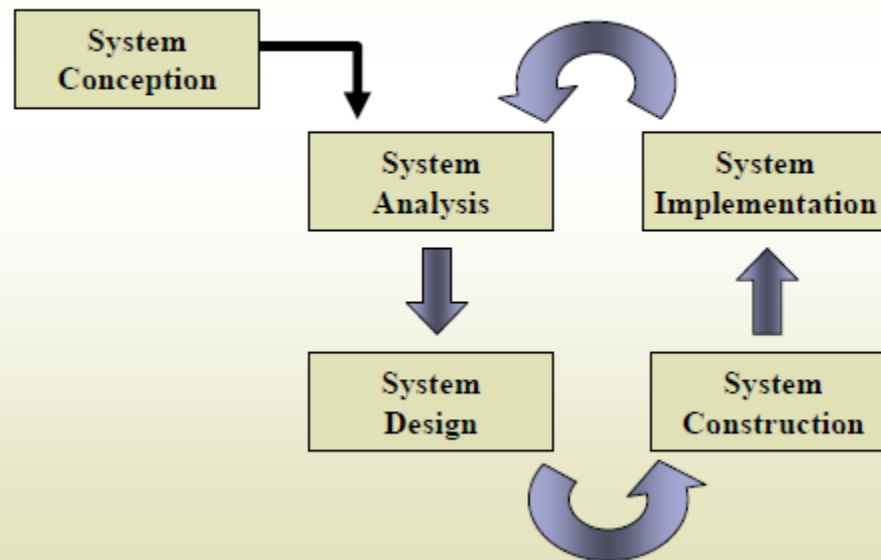
1. Pemilahan fungsi
2. Penyusunan Sistem Informasi
3. Evaluasi
4. Penggunaan Selanjutnya

- Jenis-jenis prototyping meliputi :

1. *Feasibility prototyping*
2. *Requirement prototyping*
3. *Design Prototyping*
4. *Implementation prototyping*

Prototyping (Iterative Model of Development)

[Beynon, page 319]



Prototyping (Iterative Model of Development)

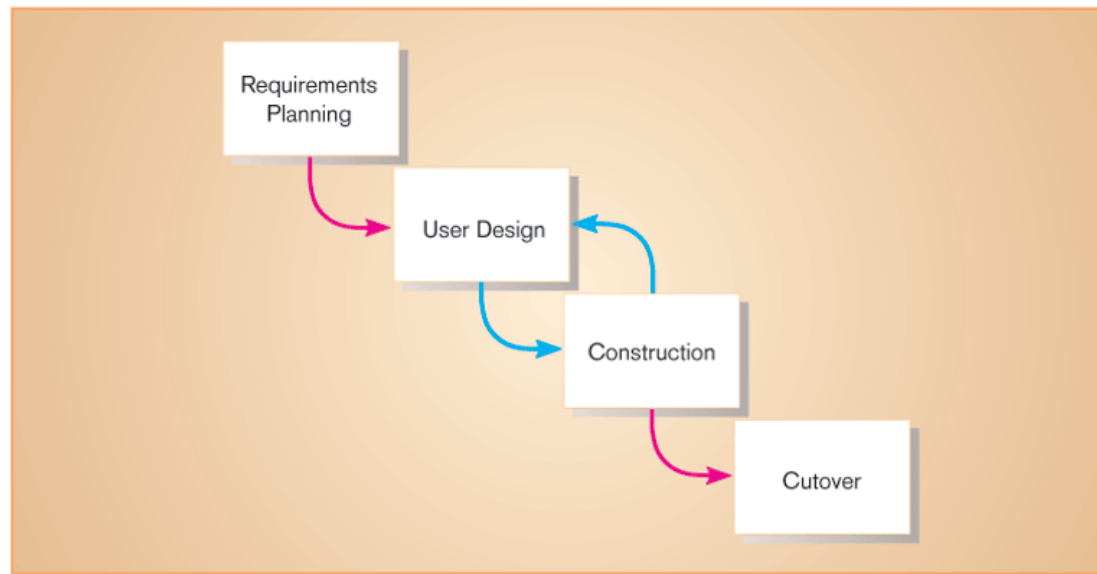
- Penggunaan software untuk membantu pengembangan dalam merencanakan, menganalisa, mendesain, memprogram dan memelihara sistem informasi.
- Software CASE dapat menghasilkan kode program dan dukungan secara otomatis untuk pengembangan sistem.
- Proyek kamus/buku kerja: sistem deskripsi dan spesifikasi diagram alat.
- Contoh produk: Oracle Designer, Rational Rose

- CASE disebut juga dengan *Computer-Aided Information System Engineering (CAISE)*.
- Banyak CASE yang dapat digunakan untuk membantu pengembangan sistem, mulai yang support *linear method* sampai yang *object-oriented*.
- CASE Tools ada 3 yaitu :
 1. *Front-end CASE Tools* : *analysis & design stage*
 2. *Back-end CASE Tools* : *construction, implementasion, testing & maintenace stage*
 3. *Integrated CASE Tools* : *kombinasi front dan back end*

Computer-Aided Software Engineering (CASE) Tools

- Metodologi untuk mengurangi waktu desain dan implementasi.
- **RAD** adalah penggabungan beberapa metode atau teknik terstruktur. RAD menggunakan metode **prototyping** dan **teknik terstruktur lainnya** untuk menentukan kebutuhan user dan perancangan sistem informasi. RAD dapat mencakup prototyping, JAD, CASE tools, dan generator kode.

Figure 1-12 RAD life cycle



Rapid Application Development (RAD)

- Proses pengembangan, meliputi :
 1. Mempelajari apakah proyek pengembangan sistem memenuhi kriteria.
 2. Mempelajari aktivitas bisnis perusahaan, menentukan area bisnis serta fungsi yang menjadi prioritas.
 3. Membuat model dari fungsi-fungsi yang menjadi prioritas.
 4. Memilih prototype mana yang direview.
 5. Implementasi Sistem Informasi

Rapid Application Development (RAD)

- **JAD** merupakan suatu kerjasama yang terstruktur antara pemakai sistem informasi, manajer dan ahli sistem informasi untuk menentukan dan menjabarkan permintaan pemakai, teknik yang dibutuhkan dan unsur rancangan eksternal.
- Proses terstruktur yang melibatkan pengguna, analis, dan manajer. Membutuhkan beberapa hari sesi workgroup intensif.
- Tujuan JAD adalah memberi kesempatan kepada user dan manajemen untuk berpartisipasi secara luas dalam siklus pengembangan sistem informasi dan untuk menentukan serta meninjau persyaratan sistem.

Joint Application Design (JAD)

- Termotivasi oleh pengakuan pengembangan perangkat lunak sebagai sesuatu yang tak terduga dan dinamis.
- Tiga prinsip kunci :
 1. Adaptif dibanding prediksi.
 2. Menekankan pada orang dibanding peran.
 3. Self-adaptif proses.

Agile Methodologies

- Siklus pengembangan incremental, digunakan untuk siklus pendek.
- Otomatis tes.
- Dua orang tim pemrograman.
- Pengkodean dan pengujian beroperasi bersama-sama.
- **Keuntungan :**
 1. Komunikasi antara pengembang.
 2. Tinggi tingkat produktivitas.
 3. Tinggi kualitas kode.

eXtreme Programming (XP)

- Suatu teknik yang menekankan penggambaran model untuk memvisualisasikan dan menganalisis masalah, mendefinisikan proses bisnis dan merancang sistem informasi.
- Pendekatan dalam teknik pemodelan MDD adalah :
 1. *Structured Analysis Design* : berorientasi proses.
 2. *Information Engineering* : berorientasi data.
 3. *Object-Oriented Analysis and Design* : menggabungkan orientasi proses dan data ke dalam bentuk objek.

Model-Driven Development (MDD)

- Teknik yang digunakan membangun *behaviour* dari suatu sistem informasi.
- Tool yang digunakan adalah *DFD (Data Flow Diagrams), Data dictionary, Process description.*

Structured Analysis Design

- Teknik yang digunakan membangun (sistem) informasi yang terstruktur.
- Tools yang digunakan adalah *ERD (Entity-Relationship diagrams)*, *Normalisasi database*.

Information Engineering

- Teknik yang digunakan untuk membangun obyek-obyek yang ada di dalam sistem informasi.
- Tools yang digunakan adalah *UML (Unified Modelling Language)*, *STD (State transition diagrams)*.

Object-Oriented Analysis and Design

- **FAST** adalah sebuah contoh *framework* proses pengembangan sistem.
- Urutan FAST adalah :
Scope Definition – Problem Analysis – Requirement Analysis – Decision Analysis – Logical Design – Physical Design & Integration – Construction & Testing – Installation & Delivery.

(Framework for the Application of System Thinking) FAST

- Alternatif lain dalam metode pengembangan sistem adalah dengan membeli software aplikasi yaitu paket software yang sudah jadi. Misalnya seperti SAP, MSProject dan lain-lain.
- Digunakan untuk aplikasi yang sifatnya umum seperti payroll dan akunting. Selain itu software berbasis enterprise sudah tersedia seperti Oracle dan SAP.

Pembelian Software Aplikasi

- Pendekatan pengembangan sistem yang dapat digunakan antara lain :
 1. Metodologi Yang Digunakan
 2. Sasaran Yang Dicapai
 3. Cara Menentukan Kebutuhan Dari Sistem
 4. Cara Mengembangkannya
 5. Teknologi Yang Digunakan

Pendekatan Pengembangan Sistem

- Pendekatan Klasik (*Classical Approach*) atau Pendekatan Tradisional (*Traditional Approach*) atau Pendekatan Konvensional (*Conventional Approach*)
Metodologi pengembangan sistem dengan mengikuti tahapan-tahapan dalam *System Development Life Cycle*.
- Pendekatan Terstruktur (*Structured Approach*)
Metodologi pengembangan sistem dengan menggunakan *tools* dan teknik yang dibutuhkan didalam pengembangan sistem.

Metodologi Yang Digunakan

- Pendekatan Sepotong (*Piecerneal Approach*)
Pendekatan pengembangan sistem dengan menekankan pada satu aplikasi tertentu tanpa memperhatikan posisi atau sasaran sistem informasinya secara global.
- Pendekatan Sistem (*Systems Approach*)
Pendekatan pengembangan sistem dengan memperhatikan sistem informasi sebagai suatu kesatuan terintegrasi dan menekankan sasarannya secara global.

Sasaran Yang Dicapai

- Pendekatan Dari Bawah Ke Atas (*Bottom-up Approach*)
Pendekatan pengembangan sistem dimulai dari level paling bawah organisasi yaitu level operasional. Termasuk dalam pendekatan klasik, dalam tahapan analisis dikenal dengan *data analysis*.
- Pendekatan Dari Atas Ke Bawah (*Top-down Approach*)
Pendekatan pengembangan sistem dimulai dari level paling atas organisasi yaitu level perencanaan. Termasuk dalam pendekatan terstruktur, dalam tahapan analisis dikenal dengan *decision analysis*.

Cara Menentukan Kebutuhan Dari Sistem

- Pendekatan Sistem Menyeluruh
Pendekatan pengembangan sistem secara serentak tanpa dibagi-bagi sehingga menjadi sulit untuk dikembangkan. Sama seperti pendekatan klasik.
- Pendekatan Modular
Pendekatan pengembangan sistem dengan memecah-mecah suatu sistem yang kompleks menjadi suatu modul yang sederhana, sehingga akan menjadi lebih mudah dipahami dan dikembangkan. Sama seperti pendekatan terstruktur.

Cara Mengembangkannya

- Pendekatan Lompatan Jauh (*Great Loop Approach*)
Pendekatan pengembangan sistem yang melakukan perubahan secara menyeluruh serentak dengan teknologi canggih, sehingga muncul resiko yang terjadi.
- Pendekatan Berkembang (*Evolution Approach*)
Pendekatan pengembangan sistem yang menerapkan teknologi canggih hanya pada aplikasi yang memerlukan saja dan dilakukan pengembangan setiap waktunya.

Teknologi Yang Digunakan

- Analisis sistem perlu memilih pendekatan khusus untuk mengembangkan suatu sistem informasi - ada banyak alternatif.
- Analisis perlu untuk memahami metodologi istilah, model, alat dan teknik.
- Analisis menggunakan alat untuk menyelesaikan kegiatan dan untuk menghasilkan model.

Yang Perlu Dilakukan Analisis Sistem

- Metodologi merupakan kumpulan teknik yang digunakan untuk menyelesaikan semua kegiatan SDLC.
- Model merupakan representasi aspek penting dari dunia nyata - abstraksi atau penyederhanaan.
- Teknik menghasilkan model (antara lain).
- Tools merupakan perangkat lunak pendukung yang membantu membuat model atau komponen proyek lainnya.

Metodologi, Model, Teknik dan Tools

Some models of system components

- Flowchart
- Data flow diagram (DFD)
- Entity-relationship diagram (ERD)
- Structure chart
- Use case diagram
- Class diagram
- Sequence diagram

- Project management application
- Drawing/graphics application
- Word processor/text editor
- Computer-aided system engineering (CASE) tools
- Integrated development environment (IDE)
- Database management application
- Reverse-engineering tool
- Code generator tool

Some models used to manage the development process

- PERT chart
- Gantt chart
- Organizational hierarchy chart
- Financial analysis models – NPV, ROI

- Strategic planning techniques
- Project management techniques
- User interviewing techniques
- Data-modeling techniques
- Relational database design techniques
- Structured analysis technique
- Structured design technique
- Structured programming technique
- Software-testing techniques
- Object-oriented analysis and design techniques

Terima Kasih