

Pengadaan Sistem Informasi

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

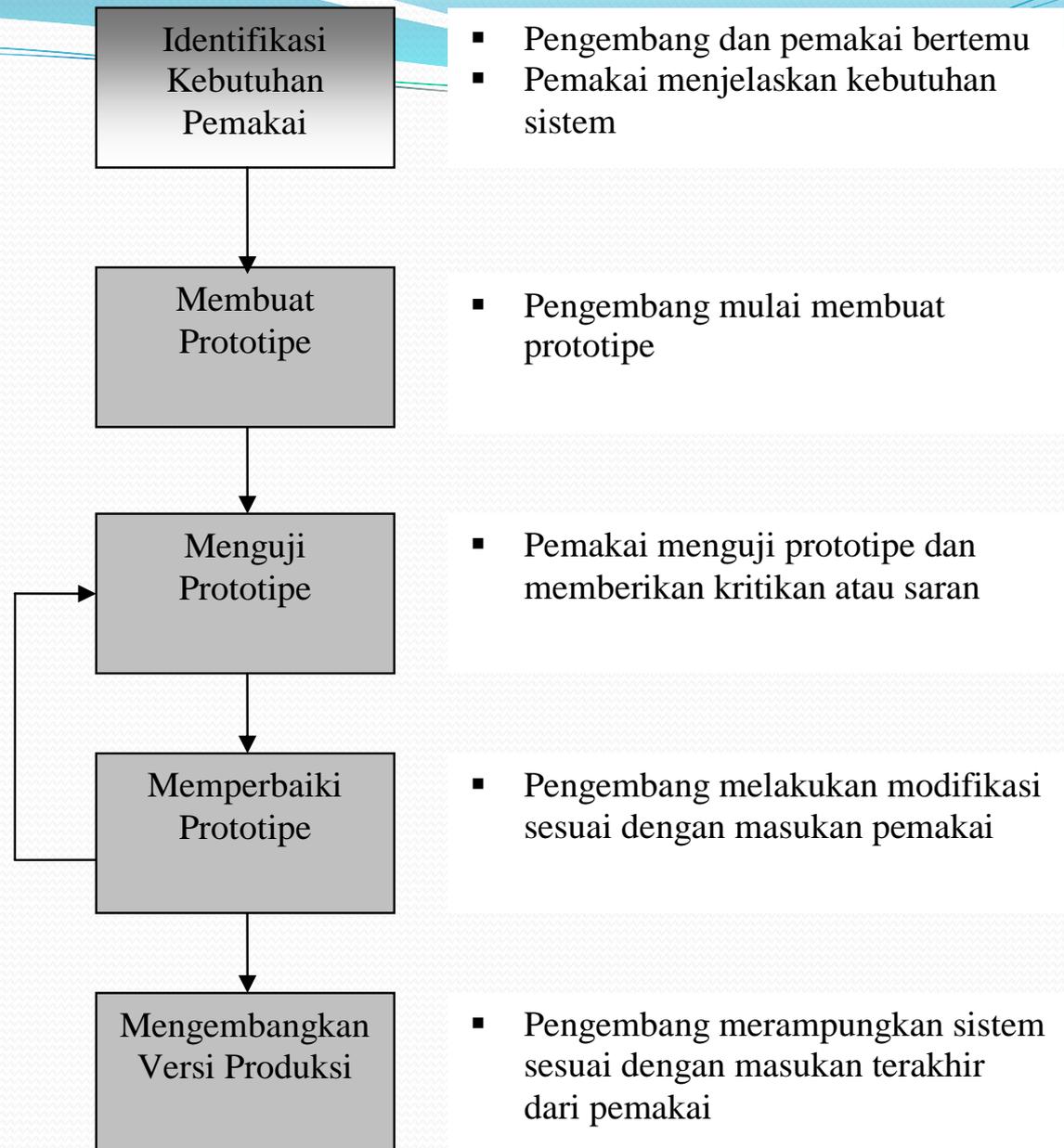
Prototipe (Lanjutan...)

- Selain itu, prototipe membuat proses pengembangan sistem informasi menjadi lebih cepat dan lebih mudah, terutama pada keadaan kebutuhan pemakai sulit untuk diidentifikasi
- Prototipe kadangkala disebut juga RAD (*Rapid Application Development*)

Sasaran Prototipe (Lucas, 2000)

1. Mengurangi waktu sebelum pemakai melihat sesuatu yang konkret dari usaha pengembangan sistem
2. Menyediakan umpan balik yang cepat dari pemakai kepada pengembang
3. Membantu menggambarkan kebutuhan pemakai dengan kesalahan yang lebih sedikit
4. Meningkatkan pemahaman pengembang dan pemakai terhadap sasaran yang seharusnya dicapai oleh sistem
5. Menjadikan keterlibatan pemakai sangat berarti dalam analisis dan desain sistem

Pendekatan Prototipe



Prototipe

- Prototipe dapat berdiri sendiri sebagai metode pengembangan tersendiri, tetapi juga dapat menjadi bagian dari SDLC yang telah dibahas di depan
- Beberapa versi SDLC yang lebih baru seringkali menyertakan prototipe sebagai alternatif atau suplemen dalam tahapan analisis dan desain sistem (Turban, McLean, dan Wetherbe, 1999)
- Dalam banyak kasus, prototipe lebih digunakan untuk mendukung SDLC daripada untuk menggantikannya (Romney, Steibart, dan Cushing, 1977)
- Prototipe dapat dibuat dengan menggunakan perangkat-perangkat RAD (misalnya Visual BASIC dan PowerBuilder), ataupun DBMS (*Database Management System*) seperti Microsoft Access, sehingga pembuatan program dapat dilakukan dengan cepat

Kelebihan Prototipe

- Pendefinisian kebutuhan pemakai menjadi lebih baik karena keterlibatan pemakai yang lebih intensif
- Meningkatkan kepuasan pemakai dan mengurangi risiko pemakai 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 pemakai
- Pemakai 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)

Kelemahan 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 dikelola dengan baik, prototipe menjadi tak pernah berakhir. Hal ini disebabkan permintaan terhadap perubahan terlalu mudah untuk dipenuhi

CASE Tool

- *Kepanjangannya Computer-Aided Software Engineering atau Computer-Assisted Software Engineering*
- Perangkat lunak yang berguna bagi para pengembang dalam merencanakan, menganalisa, merancang, melakukan pemrograman, dan memelihara sistem sistem informasi
- Tujuan utama CASE dibentuk adalah untuk mengalihkan sejumlah beban yang biasanya dipikul oleh pengembang sistem kepada komputer

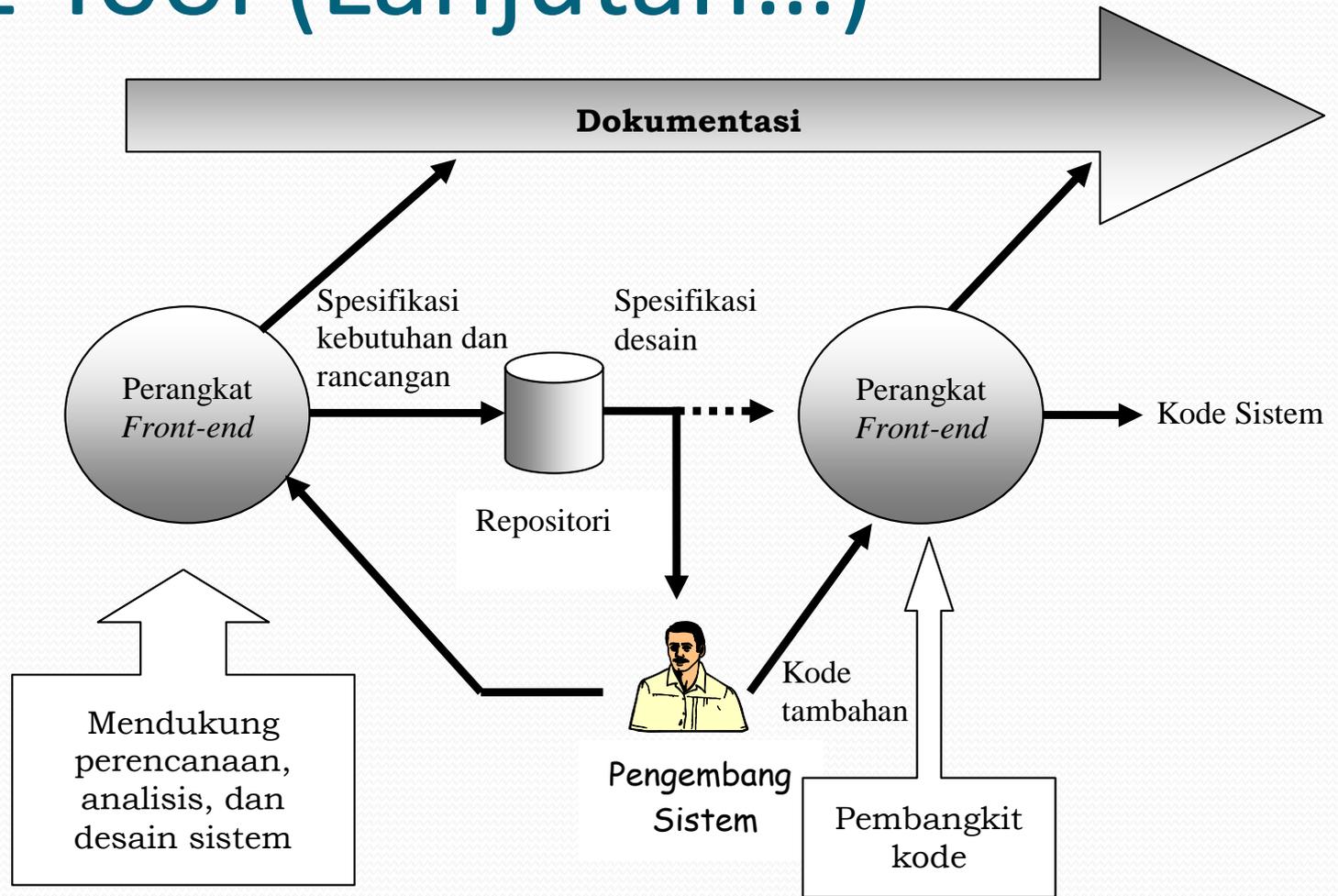
CASE Tool (Lanjutan...)

- Seringkali dikatakan bahwa CASE berguna untuk mendukung otomatisasi dalam pengembangan dan pemeliharaan sistem informasi, disebabkan perangkat ini dapat menghasilkan kode program secara otomatis
- Sejumlah studi menunjukkan bahwa CASE dapat meningkatkan produktivitas pengembangan perangkat lunak kira-kira 10% hingga 15% (Bodnar dan Hopwood, 1993)

CASE Tool (Lanjutan...)

- Perangkat CASE menggabungkan beberapa teknologi:
 - Metodologi pengembangan sistem, misalnya pengembangan sistem terstruktur,
 - Bahasa generasi keempat (4GL), yang menggunakan pendekatan nonprosedural, dan
 - Antarmuka grafis
- Perangkat CASE yang lengkap mengandung perangkat *front-end* dan *back-end*

CASE Tool (Lanjutan...)



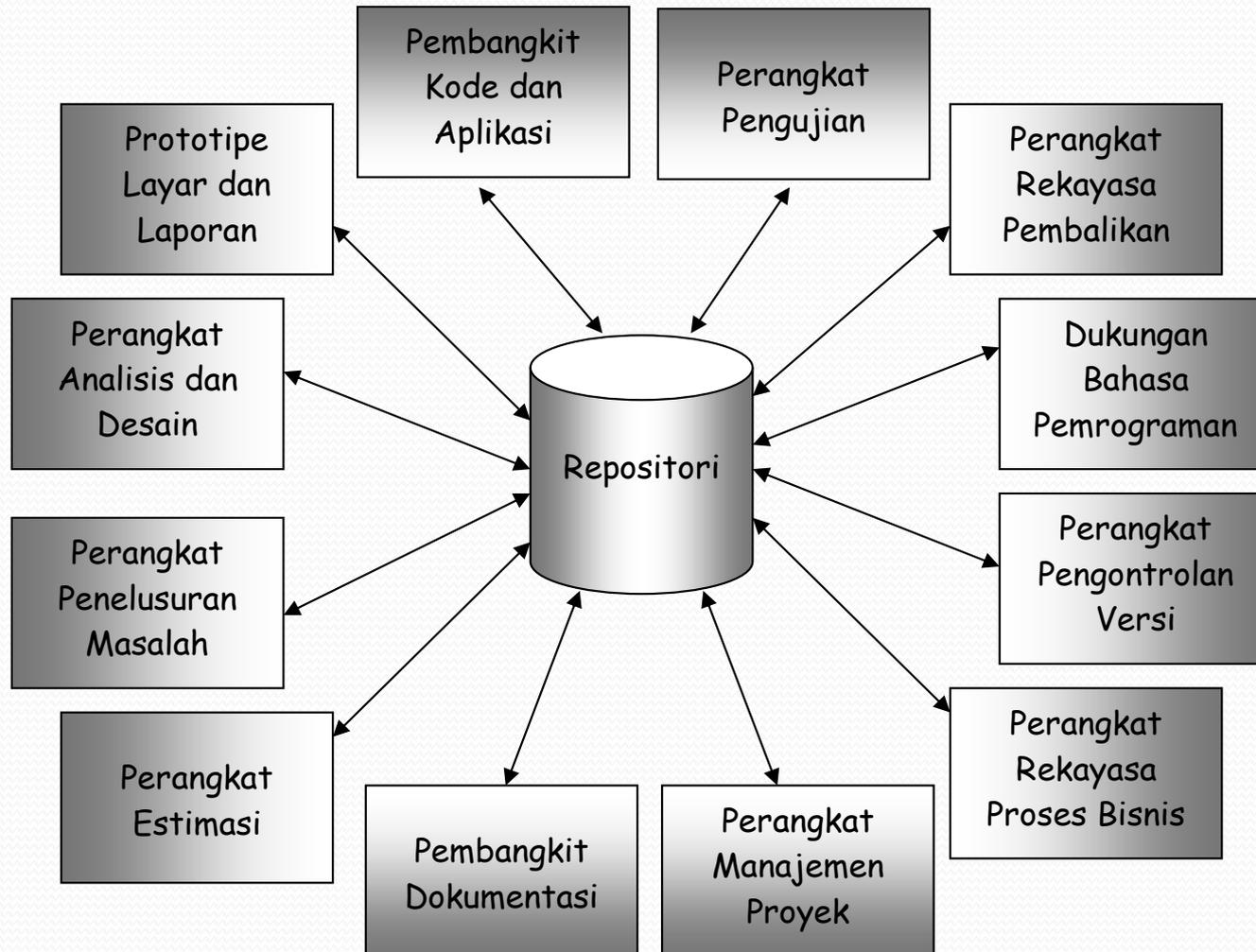
CASE Tool : Front-end

- Perangkat *front-end* merupakan perangkat CASE yang mendukung tahapan analisis dan desain
- Perangkat ini terkadang disebut sebagai **Upper CASE**
- Dukungan yang tersedia berupa:
 - Perangkat pembuat diagram (*diagramming tool*). Perangkat ini berguna untuk membuat DFD (*Data Flow Diagram*)
 - Pembangkit layar dan laporan (*screen and report generator*). Perangkat ini dapat dipakai untuk membuat prototipe laporan dengan cara memasukkan kolom atau *field* data pada layar

CASE Tool : Back-end dan i-CASE

- Perangkat *back-end* merupakan perangkat CASE yang berfungsi sebagai pembangkit kode (*code generator*), yakni menghasilkan kode program berdasarkan suatu spesifikasi rancangan. Perangkat ini biasa juga disebut **Lower CASE**
- Adapun perangkat CASE yang menggabungkan kedua kemampuan *front-end* maupun *back-end* biasa disebut **i-CASE (Integrated CASE)**

Komponen Utama CASE Tool



Komponen Utama CASE

Komponen	Keterangan
Repositori	Gudang data
Perangkat Analisis dan Desain	<ul style="list-style-type: none">• Membuat diagram aliran data, diagram entitas-hubungan (E-R), dll.• Metodologi perancangan sistem
Prototipe Layar dan Laporan	<ul style="list-style-type: none">• Pembangkit layar dan menu• Pembangkit laporan
Pembangkit Kode dan Aplikasi	<ul style="list-style-type: none">• Mengonversi spesifikasi menjadi kode program• Mendukung kemampuan drag-and-drop untuk membuat aplikasi atau antarmuka

Komponen Utama CASE

Dukungan Bahasa Pemrograman	<ul style="list-style-type: none">• Memiliki cetakan (<i>template</i>) untuk kode yang bersifat umum pada bahasa tertentu• Pustaka subrutin untuk fungsi-fungsi yang umum
Perangkat Pengujian	<ul style="list-style-type: none">• Menghasilkan data untuk pengujian• Memantau eksekusi program• Analisis kelengkapan dan konsistensi
Perangkat Rekayasa Pembalikan	<ul style="list-style-type: none">• Membantu mengamati kode sistem yang sudah ada
Perangkat Pengontrolan Versi	<ul style="list-style-type: none">• Dukungan manajemen perubahan sehingga mampu menampung lebih dari satu versi kode• Memungkinkan pengaksesan hanya pada personil yang berwenang

Komponen Utama CASE

Perangkat Rekayasa Proses Bisnis	<ul style="list-style-type: none">• Menganalisa dan memperbaiki proses-proses pada sistem sekarang• Merancang proses baru
Perangkat Manajemen Proyek	<ul style="list-style-type: none">• Mendukung bagan PERT yang digunakan untuk menangani lintasan kritis dalam jadwal proyek• Mendukung duagram Grantt• Penelusuran waktu dan pengeluaran
Pembangkit Dokumentasi	<ul style="list-style-type: none">• Menciptakan bagan alir (<i>flowchart</i>) dan dokumentasi sistem
Perangkat Estimasi	<ul style="list-style-type: none">• Memperkirakan kebutuhan personil dan biaya untuk proyek pengembangan sistem
Perangkat Penelusuran Masalah	<ul style="list-style-type: none">• Mengidentifikasi kesalahan program

Keuntungan CASE

- Meningkatkan produktivitas, dengan meningkatkan kecepatan dan efisiensi SDLC
- Membuat prototipe dapat dilakukan dengan lebih mudah, sehingga pemakai dapat melihat kemajuan proses pengembangan lebih cepat
- Membuat perubahan-perubahan rancangan sistem dapat dilakukan dengan lebih mudah
- Memungkinkan pembuatan sistem yang bekerja pada berbagai platform. Sebagai contoh, Anda dapat mendokumentasikan pada suatu sistem operasi (misalnya Windows) dan kemudian membangkitkannya pada sistem operasi yang lain (misalnya UNIX). Hal ini dimungkinkan karena kebanyakan perangkat CASE dapat menghasilkan kode dalam berbagai bahasa pemrograman, termasuk C dan C++

Kelemahan CASE

- Jika dukungan manajemen terhadap penggunaan CASE kurang maka akan menimbulkan masalah
- Harga CASE sangat mahal. Biaya untuk pelatihan bagi pengembang sistem juga mahal. Karena itu, apabila tidak termanfaatkan dengan baik, dapat dipastikan bahwa organisasi yang menggunakannya akan mengalami kerugian yang besar

Perhatian Kalau Memakai CASE

- Kemampuan analisis dan desain bagi personil yang menggunakan perangkat CASE tetap diperlukan. Sebagai apapun perangkat CASE tidak akan menghasilkan sistem yang bagus kalau para spesialis teknologi informasi yang menggunakannya tidak memiliki dasar analisis dan desain
- Perlu penyediaan waktu yang lebih banyak pada proses analisis dan desain. Hal ini disebabkan perangkat CASE tak dapat menghasilkan kode program kalau spesifikasi tidak lengkap, ambigu, dan tak benar

Contoh CASE Tool

Perangkat CASE	Vendor
Design/1	Andersen Consulting
Designer/2000	Oracle
Excelerator	Intersolv
IEF (Information Engineering Facility)	Texas Instrument
Teamwork	Cadre Technologies
Tlon	Panzophic Systems
Visible Analyze	Visible Systems

Pengadaan Sistem Informasi

1. Mengembangkan sendiri
2. Membeli perangkat lunak paket
3. Melakukan *outsourcing*

Mengembangkan Sendiri

- Kelebihan :
 - Sistem dapat diatur sesuai dengan kebutuhan
 - Dapat diintegrasikan dengan lebih baik terhadap sistem yang sudah ada
 - Proses pengembangan sistem dapat dikelola dan dikontrol
 - Dapat dijadikan sebagai keunggulan kompetitif

Mengembangkan Sendiri

- Kelemahan :
 - Perlu waktu yang lama untuk mengembangkan sistem karena harus dimulai dari nol
 - Kemungkinan program mengandung *bug* sangat besar
 - Kesulitan para pemakai dalam menyatakan kebutuhan dan kesukaran pengembangan memahami mereka dan seringkali hal ini membuat para pengembang merasa putus asa

Menggunakan Paket

- Kelebihan :
 - Memerlukan waktu pengembangan yang jauh lebih singkat karena secara prinsip perangkat lunak paket siap untuk dioperasikan. Yang diperlukan adalah penyesuaian sistem berdasarkan kebutuhan pemakai
 - Pemakai dapat memilih paket yang paling sesuai dengan kebutuhan
 - Umumnya paket merupakan perangkat lunak yang berkualitas tinggi karena sudah teruji beberapa kali di tempat lain, sehingga terbebas dari *bug*
 - Pemakai dapat melakukan uji coba terlebih dulu sebelum membeli
 - Dokumentasi lengkap

Menggunakan Paket

- Kelemahan :
 - Ada kemungkinan paket tidak mendukung fungsi-fungsi yang spesifik dalam perusahaan
 - Ada kemungkinan harganya sangat mahal
 - Perangkat lunak tidak seefisien kalau menggunakan sistem buatan sendiri (karena umumnya paket dibuat seumum mungkin sehingga dapat dipakai oleh perusahaan mana saja)
 - Evaluasi paket menyita waktu dan menuntut biaya
 - Ada kemungkinan paket hanya jalan pada jenis perangkat keras tertentu (tidak kompatibel dengan perangkat yang sudah tersedia)

Outsourcing

1. Menyerahkan pengembangan ke pihak luar
2. Menyerahkan pengembangan dan pengoperasian ke pihak luar

Hal-hal yang perlu diperhatikan dalam Outsourcing

- Menentukan pengembang yang ditunjuk untuk membangun sistem informasi dengan hati-hati. Sebaiknya, pihak luar yang dipilih memang benar-benar telah berpengalaman
- Menandatangani kontrak. Kontrak dimaksudkan sebagai pengikat tanggung jawab dan dapat dijadikan sebagai pegangan dalam melanjutkan atau menghentikan proyek jika terjadi masalah selama masa pengembangan
- Merencanakan dan memonitor setiap langkah dalam pengembangan agar keberhasilan proyek benar-benar tercapai. Kontrol perlu diterapkan pada setiap aktivitas dengan maksud agar pemantauan dapat dilakukan dengan mudah
- Menjaga komunikasi yang efektif antara personil dalam perusahaan dengan pihak pengembang dengan tujuan agar tidak terjadi konflik atau hambatan selama proyek berlangsung
- Mengendalikan biaya dengan tepat dengan misalnya memperhatikan proporsi pembayaran berdasarkan persentasi tingkat penyelesaian proyek.

Kelebihan Outsourcing

- Perusahaan dapat mengonsentrasikan diri pada bisnis yang ditangani
- Dapat digunakan untuk meningkatkan kas dalam aset perusahaan karena tak perlu ada aset untuk teknologi informasi
- Mendapatkan kepekaan yang lebih baik dan teknologi yang lebih maju
- Lebih menghemat biaya. Sebagai contoh, American Standard melaporkan bahwa dalam setahun dapat menghemat \$2 juta karena melakukan *outsourcing* terhadap operasi keuangan dan penggajian (Laudon & Laudon, 1998)

Kelebihan Outsourcing

- Meningkatkan waktu pengembangan
- Menghilangkan penyediaan sarana saat beban puncak terjadi (yakni ketika terjadi masa-masa pembeli membanjir) dan cukup melakukan pengeluaran biaya sesuai dengan tambahan layanan yang diberikan oleh pihak luar
- Memfasilitasi *downsizing*, sehingga perusahaan tak perlu memikirkan pengurangan pegawai

Kelemahan Outsourcing

- Kehilangan kendali terhadap sistem dan data karena bisa saja pihak *outsourcer* menjual data ke pesaing
- Mengurangi keunggulan kompetitif karena pihak *outsourcer* tidak dapat diharapkan untuk menyediakannya karena juga harus memikirkan klien lain
- Menjadi sangat bergantung pada pihak luar sehingga sangat sulit bagi perusahaan untuk mengambil alih kembali sistem yang sedang berjalan