

2

TINJAUAN UMUM PENGEMBANGAN SISTEM

2.1. PERLUNYA PENGEMBANGAN SISTEM

Pengembangan sistem (*systems development*) dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki atau diganti disebabkan karena beberapa hal, yaitu sebagai berikut ini.

1. Adanya permasalahan-permasalahan (*problems*) yang timbul di sistem yang lama. Permasalahan yang timbul dapat berupa:
 - a. Ketidakberesan.
Ketidakberesan dalam sistem yang lama menyebabkan sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan. Ketidak beresan ini dapat berupa:
 - kecurangan-kecurangan disengaja yang menyebabkan tidak amannya harta kekayaan perusahaan dan kebenaran dari data menjadi kurang terjamin;
 - kesalahan-kesalahan yang tidak disengaja yang juga dapat menyebabkan kebenaran dari data kurang terjamin;
 - tidak efisiennya operasi;
 - tidak ditaatinya kebijaksanaan manajemen yang telah ditetapkan.
 - b. Pertumbuhan organisasi.
Pertumbuhan organisasi yang menyebabkan harus disusunnya sistem yang baru. Pertumbuhan organisasi diantaranya adalah kebutuhan informasi yang semakin luas, volume pengolahan data semakin meningkat, perubahan prinsip akuntansi yang baru. Karena adanya perubahan ini, maka menye-

babkan sistem yang lama tidak efektif lagi, sehingga sistem yang lama sudah tidak dapat memenuhi lagi semua kebutuhan informasi yang dibutuhkan manajemen.

2. Untuk meraih kesempatan-kesempatan (*opportunities*).

Teknologi informasi telah berkembang dengan cepatnya. Perangkat keras komputer, perangkat lunak dan teknologi komunikasi telah begitu cepat berkembang. Organisasi mulai merasakan bahwa teknologi informasi ini perlu digunakan untuk meningkatkan penyediaan informasi sehingga dapat mendukung dalam proses pengambilan keputusan yang akan dilakukan oleh manajemen. Dalam keadaan pasar bersaing, kecepatan informasi atau efisiensi waktu sangat menentukan berhasil atau tidaknya strategi dan rencana rencana yang telah disusun untuk meraih kesempatan-kesempatan yang ada. Bila pesaing dapat memanfaatkannya, sedang perusahaan tidak dapat memanfaatkan teknologi ini, maka kesempatan-kesempatan akan jatuh ke tangan pesaing. Kesempatan-kesempatan ini dapat berupa peluang-peluang pasar, pelayanan yang meningkat kepada langganan dan lain sebagainya.

3. Adanya instruksi-instruksi (*directives*).

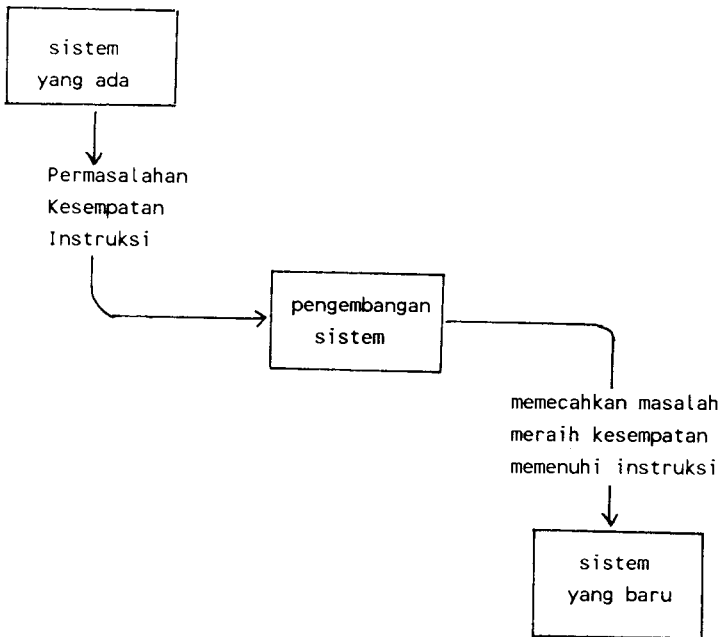
Penyusunan sistem yang baru dapat juga terjadi karena adanya instruksi-instruksi dari atas pimpinan ataupun dari luar organisasi, seperti misalnya peraturan pemerintah.

Berikut ini dapat digunakan sebagai indikator adanya permasalahan-permasalahan dan kesempatan-kesempatan yang dapat diraih, sehingga menyebabkan sistem yang lama harus diperbaiki, ditingkatkan bahkan diganti keseluruhannya. Indikator-indikator ini diantaranya adalah sebagai berikut:

- keluhan dari langganan;
- pengiriman barang yang sering tertunda;
- pembayaran gaji yang terlambat;
- laporan yang tidak tepat waktunya;
- isi laporan yang sering salah;
- tanggung jawab yang tidak jelas;
- waktu kerja yang berlebihan;
- ketidak beresan kas;
- produktivitas tenaga kerja yang rendah;
- banyaknya pekerja yang menganggur;
- kegiatan yang tumpang tindih;
- tanggapan yang lambat terhadap langganan;
- kehilangan kesempatan kompetisi pasar;
- kesalahan-kesalahan manual yang tinggi;
- persediaan barang yang terlalu tinggi;
- pemesanan kembali barang yang tidak efisien;
- biaya operasi yang tinggi;
- file-file yang kurang teratur;

- keluhan dari supplier karena tertundanya pembayaran;
- bertumpuknya *back-order* (tertundanya pengiriman karena kurangnya persediaan barang);
- investasi yang tidak efisien;
- peramalan penjualan dan produksi tidak tepat;
- kapasitas produksi yang menganggur (*idle capacities*);
- pekerjaan manajer yang terlalu teknis;

Karena adanya permasalahan, kesempatan atau instruksi, maka sistem yang baru perlu dikembangkan untuk memecahkan permasalahan-permasalahan yang timbul, meraih kesempatan-kesempatan yang ada atau memenuhi instruksi yang diberikan.



Dengan telah dikembangkannya sistem yang baru, maka diharapkan akan terjadi peningkatan-peningkatan di sistem yang baru. Peningkatan-peningkatan ini berhubungan dengan PIECES (merupakan singkatan untuk memudahkan

mengingatnya), yaitu sebagai berikut ini ¹.

- *Performance* (kinerja), peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* dan *response time*. *Throughput* adalah jumlah dari pekerjaan yang dapat dilakukan suatu saat tertentu. *Response time* adalah rata-rata waktu yang tertunda diantara dua transaksi atau pekerjaan ditambah dengan waktu *response* untuk menanggapi pekerjaan tersebut.
- *Information* (informasi), peningkatan terhadap kualitas informasi yang disajikan.
- *Economy*(ekonomis), peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan atau penurunan-penurunan biaya yang terjadi.
- *Control* (pengendalian), peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan-kecurangan yang dan akan terjadi.
- *Efficiency* (efisiensi), peningkatan terhadap efisiensi operasi. Efisiensi berbeda dengan ekonomis. Bila ekonomis berhubungan dengan jumlah sumber daya yang digunakan, efisiensi berhubungan dengan bagaimana sumber daya tersebut digunakan dengan pemborosan yang paling minimum. Efisiensi dapat diukur dari outputnya dibagi dengan inputnya.
- *Services* (pelayanan), peningkatan terhadap pelayanan yang diberikan oleh sistem.

2.2. PRINSIP PENGEMBANGAN SISTEM

Sewaktu Anda melakukan proses pengembangan sistem, beberapa prinsip harus tidak boleh dilupakan. Prinsip-prinsip ini adalah sebagai berikut ini ².

1. Sistem yang dikembangkan adalah untuk manajemen.
Setelah sistem selesai dikembangkan, maka yang akan menggunakan informasi dari sistem ini adalah manajemen, sehingga sistem harus dapat mendukung kebutuhan yang diperlukan oleh manajemen. Pada waktu Anda mengembangkan sistem, maka prinsip ini harus selalu diingat.
2. Sistem yang dikembangkan adalah investasi modal yang besar.
Sistem informasi yang akan Anda kembangkan membutuhkan dana modal yang tidak sedikit, apalagi dengan digunakannya teknologi yang mutakhir.

¹Jeffrey L. Whitten, Lonnie D. Bentley, Thomas I.M. Ho, Systems Analysis & Design Methods, (St. Louis: Times Mirror/Mosby College Publishing, 1986), hal. 107-113.

²Ibid., hal. 117-123.

Sistem yang dikembangkan ini merupakan investasi modal yang besar. Seperti halnya dengan investasi modal lainnya yang dilakukan oleh perusahaan, maka setiap investasi modal harus mempertimbangkan 2 hal berikut ini.

a. Semua alternatif yang ada harus diinvestigasi.

Bila alternatif yang ada diabaikan dan sudah terlanjur menanamkan dana ke suatu proyek investasi tertentu, maka *investor* akan kehilangan kesempatan untuk menanamkan dananya ke investasi yang lain. Ekonom menyebut hal ini dengan istilah biaya kesempatan (*opportunity cost*). Misalnya Anda mempunyai dana sebesar Rp X,- dan bila diinvestasikan ke proyek A akan mendapatkan keuntungan sebesar Rp A,-, maka Rp A,- ini yang disebut dengan *opportunity cost*. Bila Anda tidak menginvestasikan dana Anda sebesar Rp X,- tersebut ke proyek A, tetapi ke proyek B, maka proyek B harus memberikan hasil lebih besar dari *opportunity cost* yang hilang akibat tidak diinvestasikan ke proyek A. Oleh karena itu dari beberapa alternatif investasi yang ada harus diinvestigasi untuk menentukan alternatif yang terbaik atau yang paling menguntungkan.

b. Investasi yang terbaik harus bernilai.

Belum tentu alternatif terbaik merupakan investasi yang menguntungkan. Investasi terbaik ini memang menguntungkan dibandingkan dengan alternatif yang lainnya, tetapi untuk investasi terbaik ini sendiri harus juga diukur. Investasi ini baru dikatakan menguntungkan bila bernilai yang artinya manfaat (*benefit*) atau hasil baliknya lebih besar dari biaya untuk memperolehnya (*cost*). *Cost-benefit analysis* atau *cost-effectiveness analysis* dapat digunakan untuk menentukan apakah proyek investasi tersebut bernilai atau tidak.

3. Sistem yang dikembangkan memerlukan orang yang terdidik.

Manusia merupakan faktor utama yang menentukan berhasil tidaknya suatu sistem, baik dalam proses pengembangannya, penerapannya, maupun dalam proses operasinya. Oleh karena itu orang yang terlibat dalam pengembangan maupun penggunaan sistem ini harus merupakan orang yang terdidik tentang permasalahan-permasalahan yang ada dan terhadap solusi-solusi yang mungkin dilakukan. Terdidik disini bukan berarti harus secara formal duduk di perguruan tinggi, tetapi dapat dilakukan secara latihan kerja (*on-the-job training*). Analis sistem harus mempunyai pendidikan terhadap masalah yang dihadapinya. Tidaklah mungkin seorang analis sistem akan mengembangkan suatu sistem informasi bisnis tanpa mempunyai pengetahuan sedikitpun tentang bisnis atau akan mengembangkan sistem informasi akuntansi tanpa mengetahui pengetahuan sedikitpun tentang akuntansi dan teknologi komputer. Bagaimana mungkin nantinya analis sistem ini akan berkomunikasi dengan manajemen dan *programmer* yang akan membuat programnya. Demikian juga dengan pemakai sistem harus merupakan orang yang terdidik tentang sistem ini dan dapat dilakukan dengan memberikan *on-the-job training* kepada mereka tentang cara menggunakan sistem yang diterapkan.

4. Tahapan kerja dan tugas-tugas yang harus dilakukan dalam proses pengem-

bangun sistem.

Proses pengembangan sistem umumnya melibatkan beberapa tahapan kerja dan melibatkan beberapa personil dalam bentuk suatu *team* untuk mengerjakannya. Pengalaman menunjukkan bahwa tanpa adanya perencanaan dan koordinasi kerja yang baik, maka proses pengembangan sistem tidak akan berhasil dengan memuaskan. Untuk maksud ini sebelum proses pengembangan sistem dilakukan, maka harus dibuat terlebih dahulu skedul kerja yang menunjukkan tahapan-tahapan kerja dan tugas-tugas pekerjaan yang akan dilakukan, sehingga proses pengembangan sistem dapat dilakukan dan selesai dengan berhasil sesuai dengan waktu dan anggaran yang direncanakan. Siklus atau daur hidup pengembangan sistem (*Systems Development Life Cycle* atau SDLC) umumnya menunjukkan tahapan-tahapan kerja dan tugas-tugas kerja yang harus dilakukan. Beberapa *methodology* pengembangan sistem juga menyediakan lebih terinci konsep kerja yang harus dilakukan dalam proses pengembangan sistem.

5. Proses pengembangan sistem tidak harus urut.

Prinsip ini kelihatannya bertentangan dengan prinsip nomor 4, tetapi tidaklah demikian. Tahapan kerja dari pengembangan sistem di prinsip nomor 4 menunjukkan langkah-langkah yang harus dilakukan dan langkah-langkah ini dapat saja tidak harus urut, tetapi dapat dilakukan secara bersama-sama. Ingatlah waktu adalah uang. Misalnya di dalam pengembangan sistem, perancangan output merupakan tahap yang harus dilakukan sebelum melakukan perancangan file. Ini tidak berarti bahwa semua output harus dirancang semuanya terlebih dahulu baru dapat melakukan perancangan file, tetapi dapat dilakukan secara serentak, yaitu sewaktu proses perancangan output masih dilakukan, hasil perancangan output yang telah selesai dapat digunakan untuk merancang file. Contoh yang lainnya, yaitu sewaktu proses pengadaan *hardware* dilakukan, tidak berarti proses pengembangan sistem yang lainnya dihentikan hanya untuk menanti datangnya *hardware* yang dibutuhkan, tetapi dapat dilakukan perancangan lainnya yang tidak tergantung dengan keberadaan *hardware*.

6. Jangan takut membatalkan proyek.

Umumnya hal ini merupakan pantangan untuk membatalkan suatu proyek yang sedang berjalan. Keputusan untuk meneruskan suatu proyek atau membatalkannya memang harus dievaluasi dengan cermat. Untuk kasus-kasus yang tertentu, dimana suatu proyek terpaksa harus dihentikan atau dibatalkan karena sudah tidak layak lagi, maka harus dilakukan dengan tegas. Keraguan-raguan untuk terus melanjutkan proyek yang tidak layak lagi karena sudah terserapnya dana ke dalam proyek ini hanya akan membuang dana yang sia-sia. Ekonom menyebut dana yang sudah terserap ini dengan istilah *sunk cost* dan *sunk cost* ini tidak relevan untuk digunakan sebagai dasar pengambilan keputusan, karena biaya ini sudah tidak dapat ditarik kembali. Jika proyek yang tidak layak masih terus dilanjutkan lagi, maka dana berikutnya yang terserap akan sia-sia.

7. Dokumentasi harus ada untuk pedoman dalam pengembangan sistem. Kegagalan untuk membuat suatu dokumentasi kerja adalah salah hal yang sering terjadi dan merupakan kesalahan kritis yang dibuat oleh analis sistem. Banyak analis sistem yang membicarakan pentingnya dokumentasi. Mereka membuat dokumentasi hasil dari analisis setelah mereka selesai mengembangkan sistemnya dan bahkan ada yang tidak membuat dokumentasi ini. Dokumentasi ini seharusnya dibuat pada waktu proses dari pengembangan sistem itu sendiri masih dalam proses, karena dokumentasi ini dapat dihasilkan dari hasil kerja tiap-tiap langkah di pengembangan sistem. Dokumentasi yang dibuat dan dikumpulkan selama proses dari pengembangan sistem dapat digunakan untuk bahan komunikasi antara analis sistem dengan pemakai sistem dan dapat digunakan untuk mendorong keterlibatan pemakai sistem.

2.3. SIKLUS HIDUP PENGEMBANGAN SISTEM

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun-tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari 'mulai' sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Bila operasi sistem yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap yang pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah di dalam tahapan tersebut dalam proses pengembangannya.

Ide dari *systems life cycle* adalah sederhana dan masuk akal. Di *systems life cycle*, tiap-tiap bagian dari pengembangan sistem dibagi menjadi beberapa tahapan kerja. Tiap-tiap tahapan ini mempunyai karakteristik tersendiri. Tahapan utama siklus hidup pengembangan sistem dapat terdiri dari tahapan perencanaan sistem (*systems planning*), analisis sistem (*systems analysis*), desain sistem (*systems design*), seleksi sistem (*systems selection*), implementasi sistem (*systems implementation*) dan perawatan sistem (*systems maintenance*). Tahapan-tahapan seperti ini sebenarnya merupakan tahapan di dalam pengembangan sistem teknik (*engineering systems*). Pengembangan konstruksi dari gedung, jaringan transmisi tenaga listrik, mesin-mesin dan pabrik-pabrik kimia merupakan contoh dari pengembangan *engineering systems*. Tampak bahwa daur hidup dari sistem teknik dan sistem informasi dapat sama atau mirip. Hal ini tidaklah kebetulan, karena proses pengembangan sistem informasi adalah proses teknik dan proses semacam ini harus mengikuti langkah-langkah yang sama serta prinsip-prinsip umum dari sistem teknik. Istilah *software engineering* merupakan proses pengembangan perangkat lunak yang merupakan subsistem dari pengembangan sistem informa-

si. Jika ditinjau kembali siklus pengembangan sistem sejak tahun 1970 yang diusulkan oleh beberapa penulis, maka akan didapatkan kesamaan dengan siklus di sistem teknik³.

Sebelum dibahas siklus pengembangan sistem yang akan digunakan di buku ini, terlebih dahulu akan disajikan beberapa siklus pengembangan sistem dari beberapa penulis sejak tahun 1970. Penyajian beberapa siklus pengembangan sistem ini akan berguna sebagai perbandingan dengan yang digunakan di buku ini.

1970. J. F Kelly, *Computerized Management Information Systems*, (MacMilan, 1970), dikutip oleh A. Ziya Aktas, *Structured Analysis & Design of Information Systems*, (NJ: Prentice-Hall, 1987):

1. Penelitian sistem (systems survey).
 - a. Definisi ruang-lingkup (scope definition).
 - b. Studi penelitian (survey study).
2. Analisis dan desain sistem (system analysis and design).
 - a. Studi penelitian (survey study).
 - b. Pengumpulan data dan analisis (data collection and analysis).
 - c. Desain sistem (system design).
 - d. Rencana implementasi (implementation planning).
3. Pengembangan sistem (system development).
 - a. Pengembangan (development).
 - b. Pengetesan (testing).
 - c. Pengoperasian sistem baru (cutover).
 - d. Perawatan (maintenance).

1970. Martin L. Rubin, Thomas Harrel (editor), *Introduction to The System Life Cycle* (Volume 1; London: Brandon/Systems Press):

1. Konsepsi sistem (system conception).
2. Analisis pendahuluan (preliminary analysis).
 - a. Pendefinisian masalah pendahuluan (preliminary problem definition).
 - b. Investigasi (investigation).
 - c. Persiapan usulan sistem (system proposal preparation).
3. Desain sistem (system design).
 - a. Analisis terinci (detailed analysis).
 - b. Mendesain keputusan (design decision).

³Ziya A. Aktas, *Structured Analysis and Design of Information Systems*, (New Jersey: Prentice-Hall, 1987), hal. 13-18.

- c. Mendesign sasaran (design objective).
- d. Rancang bangun sistem (system specifications).
4. Pemrograman (programming).
 - a. Memecahkan kembali rancang bangun (resolve specification).
 - b. Mengembangkan bagan alir secara garis besar (develop macro flowchart).
 - c. Menulis instruksi program (write program instructions).
 - d. Merakit program (assemble program).
 - e. Mempersiapkan data untuk tes (prepare test data).
 - f. Melakukan pengetesan (make test run).
 - g. Mengecek hasil (check result).
 - h. Mendiagnosa kesalahan (diagnose errors).
 - i. Membetulkan program (correct program).
 - j. Memulai pengetesan sistem (begin system testing).
5. Dokumentasi (documentation).
6. Instalasi sistem (system installation).
7. Operasi sistem (system operation).

1978. Robert G. Murdick , Thomas C. Fuller, Joel E. Ross, Frank J. Winnermark, *Accounting Information Systems*, (NJ: Prentice-hall):

1. Tahap studi (study phase).
 - a. Studi tujuan dan masalah organisasi (study organization goals and problems).
 - b. Studi sumber daya dan kesempatan yang dimiliki perusahaan (study company resources and opportunities).
 - c. Studi kemampuan komputer yang dimiliki (study computer capabilities).
 - d. Mempersiapkan usulan untuk studi design SIM (prepare proposal for MIS design study).
2. Tahap desain kotor (gross design phase).
 - a. Mengenal subsistem-subsistem dibutuhkan (identify required subsystems).
3. Tahap desain terinci (detailed design phase).
 - a. Menyebarkan kepada organisasi tentang sifat dari proyek yang mempunyai masa depan ini (disseminate to the organization the nature of the prospective project).
 - b. Mengenal kriteria prinsip dan dominan yang perlu dipertimbangkan untuk SIM (identify dominant and principal trade-off criteria for the MIS).
 - c. Mendefinisikan kembali subsistem-subsistem dengan lebih rinci (redefine the subsystems in great detail).
 - d. Tentukan tingkat otomatis yang memungkinkan untuk masing-masing kegiatan atau transaksi (determine the degree of automation possible for each activity or transaction).
 - e. Desain basis data atau file induk (design the data base or mater file).
 - f. Membuat model dari sistem secara kuantitatif (model the system quantitatively).
 - g. Kembangkan dukungan untuk komputer (develop computer support).
 - h. Terapkan format dari input dan output (establish input and output formats).
 - i. Mengetes sistem (test the system).

- j. Usulkan struktur organisasi formal untuk mengoperasikan sistem (propose the formal organization structure to operate the system).
- k. Dokumentasikan desain terinci ini (document the detailed design).
- 4. Tahap implementasi (implementation phase).
 - a. Rencanakan urutan implementasi (plan the implementation sequence).
 - b. Mengatur implementasi (organize for implementation).
 - c. Mengembangkan prosedur untuk proses instalasi (develop the procedures for the installation process).
 - d. Melatih personil operasi (train operating personnel).
 - e. Mendapatkan perangkat keras (obtain hardware).
 - f. Mengembangkan perangkat lunak (develop the software).
 - g. Mendapatkan formulir yang telah ditentukan di desain terinci atau mengembangkan formulir bila diperlukan (obtain forms specified in detail design or develop forms as necessary).
 - h. Mendapatkan data dan membuat file induk (obtain data and construct the master file).
 - i. Mengetes sistem sebagian-bagian (test the system by parts).
 - j. Mengetes sistem secara keseluruhan (test the complete system).
 - k. Memulai mengoperasikan SIM yang baru (cut over to the new MIS).
 - l. Mencari kesalahan sistem (debug the system).
 - m. Mendokumentasikan SIM yang dioperasikan (document the operational MIS).
 - n. Mengevaluasi sistem yang sedang dioperasikan (evaluate the system in operation).

1980. Charles L. Biggs, Evan G. Birks, William Atkins, *Managing the Systems Development Process*, (NJ: Prentice-hall):

- 1. Perencanaan sistem (systems planning).
 - a. Investigasi awal (initial investigation).
 - b. Studi kelayakan (feasibility study).
- 2. Kebutuhan-kebutuhan sistem (systems requirements).
 - a. Operasi dan analisis sistem (operations and system analysis).
 - b. Kebutuhan-kebutuhan pemakai (user requirements).
 - c. Pendekatan dukungan secara teknik (technical support approach).
 - d. Design secara konsep dan kaji ulang paket (conceptual design and package review).
 - e. Penilaian alternatif dan perencanaan (alternatives evaluation and planning).
- 3. Pengembangan sistem (systems development).
 - a. Rancang bangun sistem secara teknis (system technical specifications).
 - b. Rancang bangun aplikasi (applications specifications).
 - c. Pemrograman aplikasi dan pengetesan (applications programming and testing).
 - d. Prosedur pemakai dan pengendalian (user procedures and controls).
 - e. Latihan untuk pemakai (user training).
 - f. Perencanaan implementasi (implementation planning).

- g. Perencanaan konversi (conversion planning).
- h. Pengetesan sistem (systems test).
- 4. Implementasi sistem (systems implementation).
 - a. Konversi dan tahap implementasi (conversion and phase implementation).
 - b. Perbaikan dan pembedulan (refinement and tuning).
 - c. Kaji ulang setelah implementasi (post implementation review).
- 5. Perawatan sistem (systems maintenance).

1980. Enid Squire, *Introducing Systems Design*, (Massachusetts: Addison-Wesley):

- 1. Mengajukan desain dan implementasi proyek (request for project design & implementation).
- 2. Studi kelayakan (feasibility study).
- 3. Merencanakan dan memproyeksikan perkiraan biaya (planning and project cost estimation).
- 4. Desain dan pengembangan sistem (system design & development).
- 5. Implementasi sistem (system implementation).
- 6. Perawatan sistem (system maintenance).

1982. Joseph W. Wilkinson, *Accounting and Information Systems*, (New York: John Wiley & Sons):

- 1. Perencanaan sistem (systems planning).
- 2. Analisis sistem (systems analysis).
 - a. Penelitian terhadap sistem yang ada (survey of present system).
 - b. Analisis dari temuan penelitian (analysis of survey findings).
 - c. Identifikasi kebutuhan-kebutuhan informasi (identification of information needs).
 - d. Identifikasi kebutuhan sistem (identification of system requirements).
 - e. Penyerahan laporan analisis sistem (submission of system analysis report).
- 3. Desain sistem (systems design).
 - a. Evaluasi dari alternatif desain (evaluation of design alternatives).
 - b. Persiapan rancang bangun desain (preparation of design specifications).
 - c. Penyerahan laporan desain sistem (submission of system design report).
- 4. Justifikasi dan seleksi sistem (systems justification and selection).
 - a. Penentuan kelayakan desain (determination of design feasibility).
 - b. Permintaan usulan pengadaan perangkat keras dan perangkat lunak (solicitation of hardware and software proposals).
 - c. Evaluasi usulan-usulan sistem (evaluation of system proposals).
 - d. Penyeleksian perangkat lunak dan perangkat keras untuk sistem (selection of system hardware and software).
- 5. Implementasi sistem (systems implementation).
 - a. Penerapan rencana dan pengendalian untuk implementasi (establishment of implementation plans and controls).
 - b. Eksekusi dari kegiatan-kegiatan implementasi (execution of implementation activi-

- ties).
 - c. Pengawasan dan penilaian dari sistem baru (follow up and evaluation of new system).
 - 6. Operasi dan penilaian sistem (systems operation and evaluation).
1982. John Page, Paul Hooper, *Accounting and Information Systems*, (NJ: Prentice-Hall):
- 1. Analisis sistem (system analysis).
 - 2. Pernyataan dari sasaran (statement of objective).
 - 3. Desain sistem (system design).
 - 4. Rancang bangun sistem (system specification).
 - 5. Pemrograman (programming).
 - 6. Implementasi (implementation).
 - 7. Penilaian (evaluation).
1982. Robert J. Verzello/John Reuter III, *Data Processing: Systems and Concepts*, (Tokyo: McGraw-Hill Kogakusha):
- 1. Analisis (analysis).
 - a. Studi sistem (system studi).
 - b. Usulan pemecahan (proposal of solution).
 - c. Studi kelayakan (feasibility study).
 - 2. Desain (design).
 - a. Pendefinisian kebutuhan-kebutuhan fungsional (definition of functional requirement).
 - b. Persiapan rancang bangun implementasi (preparation of implementation specifications).
 - 3. Implementasi (implementation).
 - a. Pemrograman (programming).
 - b. Instalasi (installation).
 - c. Konversi (conversion).
1982. Henry C. Lucas, *Information Systems Concepts for Management*, (Tokyo: McGraw-Hill Kogakusha):
- 1. Penelitian pendahuluan (*preliminary survey*).
 - 2. Studi kelayakan (feasibility study).
 - 3. Analisis sistem (systems analysis).
 - 4. Desain (design).
 - 5. Rancang bangun (specifications).
 - a. Logika pengolahan (processing logic).
 - b. Desain file (file design).
 - c. Masukan/Keluaran (input/output).
 - d. Kebutuhan pemrograman (programming requirement).

- e. Prosedur-prosedur manual (manual procedures).
6. Pemrograman (programming).
7. Pengetesan (testing).
8. Pelatihan (training).
9. Konversi dan instalasi (conversion and installation).
10. Operasi (operation).
 - a. Perawatan (maintenance).
 - b. Perbaikan dan peningkatan-peningkatan (enhancement).

1983. Robert A. Leitch/K. Roscoe Davis, *Accounting Information Systems*, (NJ: Prentice-Hall):

1. Analisis sistem (system analysis).
 - a. Definisi kebutuhan dan pendekatan alternatif (requirement definition & alternative approaches).
 - b. Desain sistem secara umum atau secara konsep (general system design / conceptual design).
2. Desain dan implementasi sistem (system design and implementation).
 - a. Desain sistem terinci (detailed system design).
 - b. Rancang bangun program dan perencanaan implementasi (program specification and implementation planning).
 - c. Pemrograman dan pengetesan (programming and testing).
 - d. Pengetesan sistem (system testing).
 - e. Konversi (conversion).
 - f. Implementasi (implementation).
3. Penilaian sistem (system evaluation).

1983. William S. Davis, *Systems Analysis and Design*, (Massachusetts: Addison-Wesley):

1. Definisi masalah (problem definition).
2. Studi kelayakan (feasibility study).
3. Analisis (analysis).
4. Desain sistem (system design).
5. Desain terinci (detailed design).
6. Implementasi (implementation).
7. Perawatan (maintenance).

1984. Frederick H. Wu, *Accounting Information Systems Theory and Practice*, (Tokyo: McGraw-Hill Kogakusha):

1. Awal dari proyek sistem (initiation of systems project).
2. Analisis dan penilaian sistem (systems analysis and evaluation).
3. Desain dan seleksi sistem (systems design and selection).

4. Implementasi sistem (systems implementation).
5. Pemeriksaan setelah implementasi (post audit).
6. Perawatan sistem (systems maintenance).

1985. Gordon B. Davis, *Management Information System: Conceptual Foundations, Structure, and Development*, (Tokyo: McGraw-Hill Kogakusha):

1. Tingkat definisi (definiton stage).
 - a. Definisi usulan (proposal definition).
 - b. Penilaian kelayakan (feasibility assesment).
 - c. Analisis kebutuhan informasi (information requirement analysis).
 - d. Desain secara konsep (conceptual design).
2. Tingkat pengembangan (development stage).
 - a. Desain sistem secara fisik (physical system design).
 - b. Desain basis data secara fisik (physical database design).
 - c. Pengembangan program (program development).
 - d. Pengembangan prosedur (procedure development).
3. Instalasi dan operasi (installation and operation).
 - a. Konversi (conversion).
 - b. Operasi dan perawatan (operation and maintenance).
 - c. Pemeriksaan setelah implementasi (post audit).

1985. Donald H. Sander, *Computer Today*, (New York: McGraw-Hill):

1. Definisi masalah (problem definition).
2. Analisis sistem (system analysis).
3. Desain sistem (system design).
4. Implementasi sistem (system implementation).

1985. Robert H. Blismer, *Computer Annual, An Introduction to Information Systems*, (New York: John Wiley & Sons):

1. Memahami sistem yang ada (understanding the existing system).
 - a. Mengumpulkan informasi (collecting information).
 - b. Menganalisis sistem yang ada (analyzing the existing system).
2. Mendefinisikan kebutuhan sistem baru (defining new system requirement).
 - a. Pertimbangan-pertimbangan perencanaan (planning considerations).
 - b. Kebutuhan keluaran, masukan, simpanan dan pengolahan (output, input, storage and processing requirement).
 - c. Mengidentifikasi kriteria penilaian (identify evaluation criteria).
3. Proses desain sistem (the system design process).
 - a. Desain keluaran (output design).
 - b. Desain masukan (input design).
 - c. Desain file (file design).

- d. Desain pengolahan sistem (system processing design).
- e. Pengendalian sistem (system control).
- f. Dokumentasi rinci (detail documentation).
- 4. Pengembangan dan implementasi sistem (system development and implementation).
 - a. Menilai perangkat lunak paket (evaluating packaged software).
 - b. Pengembangan perangkat lunak (software development).
 - c. Dokumentasi sistem dan pelatihan (system documentation and training).
 - d. Pengetesan sistem (system testing).
 - e. Implementasi sistem (system implementation).

1986. John Burch, Gary Grudnitski, *Information Systems, Theory and Practice*, (New York: John Wiley & Sons):

- 1. Kebijakan dan perencanaan sistem (system policy and planning).
- 2. Pengembangan sistem (system development).
 - a. Analisis sistem (system analysis).
 - b. Desain sistem secara umum (general system design).
 - c. Penilaian sistem (system evaluation).
 - d. Desain sistem terinci (detailed system design).
 - e. Implementasi sistem (system implementation).
- 3. Manajemen sistem dan operasi (system management and operation).

1986. Jeffrey L. Whitten, Lonnie D. Bentley, Thomas I.M. Ho, *Systems Analysis & Design Methods*, (St. Louis: Times Mirror/Mosby College Publishing):

- 1. Analisis sistem (systems analysis).
 - a. Penelitian situasi/investigasi pendahuluan/studi kelayakan (survey the situation/preliminary investigation or feasibility study).
 - b. Mempelajari sistem yang ada (study the current system).
 - c. Mendefinisikan kebutuhan-kebutuhan pemakai (define user requirement).
 - d. Menilai pemecahan-pemecahan alternatif (evaluate alternative solutions).
- 2. Desain sistem (systems design).
 - a. Memilih peralatan komputer baru dan perangkat lunak (select new computer equipment and software).
 - b. Desain sistem yang baru (design the new system).
 - Desain keluaran (design output).
 - Desain file/basis data (design file/database).
 - Desain masukan (design input).
 - Desain dialog terminal (design terminal dialogue).
 - Desain metode dan prosedur (design methods and procedures).
 - Desain program (design programs).
- 3. Implementasi sistem (systems implementation).
 - A. Mengkonstruksi sistem informasi baru (construct the new information system).
 - a. Menginstall perangkat keras dan perangkat lunak sistem (install hardware

- and system software).
- b. Merencanakan untuk pemrograman (plan for programming).
- c. Membangun data tes dan mengetes file/basis data (build test data and test files/databases).
- d. Menulis dan mengetes program (write and test custom program).
- e. Menginstal perangkat lunak aplikasi (install applications software).
- f. Memodifikasi perangkat lunak aplikasi (modify applications software).
- B. Mengirim sistem informasi baru untuk operasi (deliver the new information system to operation).
 - a. Menginstal file dan basis data (install files and databases).
 - b. Melatih pemakai sistem untuk menggunakan sistem baru (train users to use the new system).
 - c. Konversi ke sistem baru (convert to the new system).
 - d. Kaji ulang setelah implementasi (post implementation review).

1986. George M. Scott, *Principles of Management Information Systems*, (New York: McGraw-Hill):

1. Studi pendahuluan (preliminary study).
 - a. Penemuan masalah (problem discovery).
 - b. Studi pendahuluan (preliminary study).
 - c. Laporan singkat studi pendahuluan (preliminary study brief).
2. Analisis sistem (systems analysis).
 - a. Perencanaan proyek (project planning).
 - b. Penelitian sistem yang ada (survey existing system).
 - c. Mendefinisikan masalah (define problems).
 - d. Analisis sistem (systems analysis).
 - e. Kebutuhan-kebutuhan sistem (systems requirements).
 - f. Laporan rancang bangun (specification report).
3. Desain sistem (system design).
 - a. Desain secara makro (macro design).
 - b. Desain terinci (detailed design).
 - c. Rancang bangun desain (design specifications).
4. Implementasi (implementation).
 - a. Pelatihan (training).
 - b. Penyeleksian program dan peralatan (selection programs & equipment).
 - c. Pemrograman (programming).
 - d. Persiapan tempat (site preparation).
 - e. Instalasi (installation).
 - f. Konversi (conversion).
 - g. Penerimaan (acceptance).

1987. A. Ziya Aktas, *Structured Analysis & Design of Information Systems*, (NJ: Prentice-Hall):

1. Perencanaan (planning).
 - a. Permintaan untuk studi suatu sistem (request for a system studi).
 - b. Investigasi awal (initial investigation).
 - c. Studi kelayakan (feasibility study).
2. Analisis (analysis).
 - a. Mendefinisikan kembali masalah (redefine the problem).
 - b. Memahami sistem yang ada (understand the existing system).
 - c. Menentukan kebutuhan-kebutuhan pemakai dan hambatan-hambatan pada suatu sistem baru (determine user requirements and constraints on a new system).
 - d. Model logika dari pemecahan yang direkomendasi (logical model of the recommended solution).
3. Desain secara fisik (physical design).
 - a. Desain sistem atau desain secara umum atau rancang bangun sistem (system design or general design or system specifications).
 - b. Desain terinci atau desain khusus (detailed design or specific design).
4. Implementasi atau konstruksi (implementation or construction).
 - a. Pembangunan sistem (system building).
 - b. Pengetesan (testing).
 - c. Instalasi/konversi (installation/conversion).
 - d. Operasi (operation).
 - e. Kaji ulang setelah implementasi (post-implementation review).
5. Perawatan (maintenance).
 - a. Perawatan dan peningkatan-peningkatan (maintenance and enhancement).

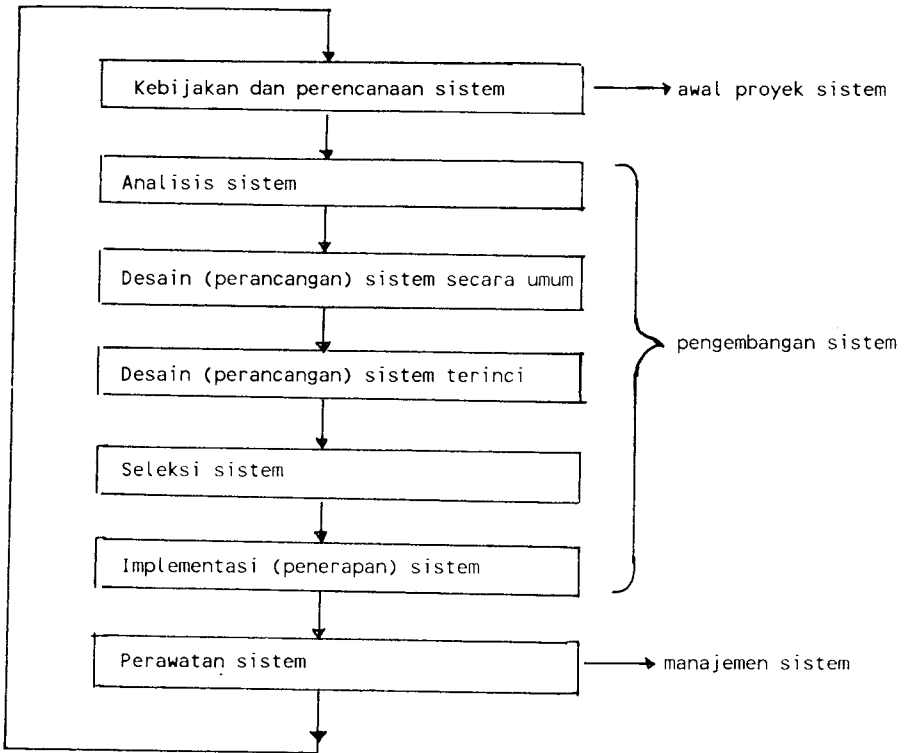
Dari beberapa siklus pengembangan sistem yang disajikan ini, maka proses dari pengembangan sistem yang terutama adalah analisis sistem, desain sistem dan implementasi sistem. Beberapa penulis juga memasukkan proses kebijakan dan perencanaan sistem dalam tahapan pengembangan sistem. Sebenarnya proses ini merupakan tahapan sebelum dilakukan pengembangan sistem. Beberapa penulis menyebut tahap ini sebagai awal terjadinya proyek sistem (*initiation of system project*).

Beberapa penulis juga memisahkan desain sistem menjadi dua tahapan yang terpisah, yaitu desain sistem secara umum atau desain sistem secara konsep atau secara makro atau secara kotor atau secara logika atau secara khusus (*general design or conceptual design or macro design or gross design or logical design or specific design*) dengan desain sistem secara rinci atau secara fisik (*detailed design or physical design*). Di buku ini, tahap desain juga akan dipisahkan ke dalam dua tahap tersebut. Beberapa buku juga ada yang memasukkan tahap desain secara umum atau secara konseptual tersebut ke dalam tahap analisis.

Tahap perawatan sistem (*systems maintenance*) sebenarnya juga merupakan tahapan setelah pengembangan sistem selesai dilakukan dan sistem telah

dioperasikan. Beberapa penulis menyebut juga tahap ini sebagai tahap manajemen sistem, karena yang melakukan proses ini sudah bukan analisis sistem, tetapi manajemen.

Siklus hidup pengembangan sistem dengan langkah-langkah utamanya yang akan digunakan di buku ini adalah sebagai berikut:



Gambar. 2.1. Siklus hidup pengembangan sistem.

2.4. PENDEKATAN PENGEMBANGAN SISTEM

Terdapat beberapa pendekatan untuk mengembangkan sistem, yaitu sebagai berikut ini.⁴

⁴Lihat Wilkinson, *Accounting and Information Systems*, (New York: John Wiley & Sons, 1982), hal. 547-549.

1. Pendekatan klasik lawan pendekatan terstruktur (dipandang dari metodologi yang digunakan).
2. Pendekatan sepotong lawan pendekatan sistem (dipandang dari sasaran yang akan dicapai).
3. Pendekatan bawah-naik lawan pendekatan atas-turun (dipandang dari cara menentukan kebutuhan dari sistem).
4. Pendekatan sistem-meyeluruh lawan pendekatan moduler (dipandang dari cara mengembangkannya).
5. Pendekatan lompatan jauh lawan pendekatan berkembang (dipandang dari teknologi yang akan digunakan).

2.4.1. Pendekatan klasik lawan pendekatan terstruktur

Pendekatan klasik (*classical approach*) merupakan lawan dari pendekatan terstruktur (*structured approach*). Metodologi pendekatan klasik mengembangkan sistem dengan mengikuti tahapan-tahapan di *systems life cycle*. Pendekatan ini menekankan bahwa pengembangan sistem akan berhasil bila mengikuti tahapan di *systems life cycle*. Akan tetapi sayangnya didalam praktek, hal ini tidaklah cukup, karena pendekatan ini tidak memberikan pedoman lebih lanjut tentang bagaimana melakukan tahapan-tahapan tersebut dengan terinci. Orang yang mengembangkan sistem masih memerlukan alat-alat dan teknik-teknik untuk mengembangkan sistem tersebut. Mulai awal tahun 1970 muncul suatu pendekatan baru disebut dengan pendekatan terstruktur. Pendekatan ini pada dasarnya mencoba menyediakan kepada analis sistem tambahan alat-alat dan teknik-teknik untuk mengembangkan sistem disamping tetap mengikuti ide dari *systems life cycle*.

2.4.1.1. Pendekatan klasik

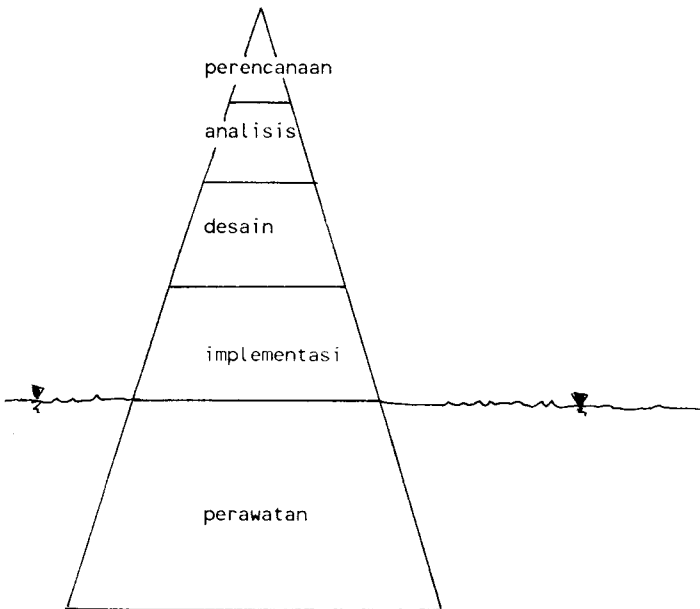
Pendekatan klasik (*classical approach*) disebut juga dengan pendekatan tradisional (*traditional approach*) atau pendekatan konvensional (*conventional approach*) adalah pendekatan didalam pengembangan sistem yang mengikuti tahapan-tahapan di *systems life cycle* tanpa dibekali dengan alat-alat dan teknik-teknik yang memadai. Karena sifat dari sistem informasi sekarang menjadi lebih kompleks, pendekatan klasik tidak cukup digunakan untuk mengembangkan suatu sistem informasi yang sukses dan akan menimbulkan beberapa permasalahan. Permasalahan-permasalahan yang dapat timbul di pendekatan klasik adalah sebagai berikut ini.

1. Pengembangan perangkat lunak akan menjadi sulit.
Pendekatan klasik kurang memberikan alat-alat dan teknik-teknik di dalam mengembangkan sistem dan sebagai akibatnya proses pengembangan perangkat lunak menjadi tidak terarah dan sulit untuk dikerjakan oleh pemrogram.

Lain halnya dengan pendekatan terstruktur yang memberikan alat-alat seperti diagram arus data (*data flow diagram*), kamus data (*data dictionary*), tabel keputusan (*decision table*), diagram IPO dan bagan terstruktur (*structured chart*) dan lain sebagainya yang memungkinkan pengembangan perangkat lunak lebih terarah berdasarkan alat-alat dan teknik-teknik tersebut.

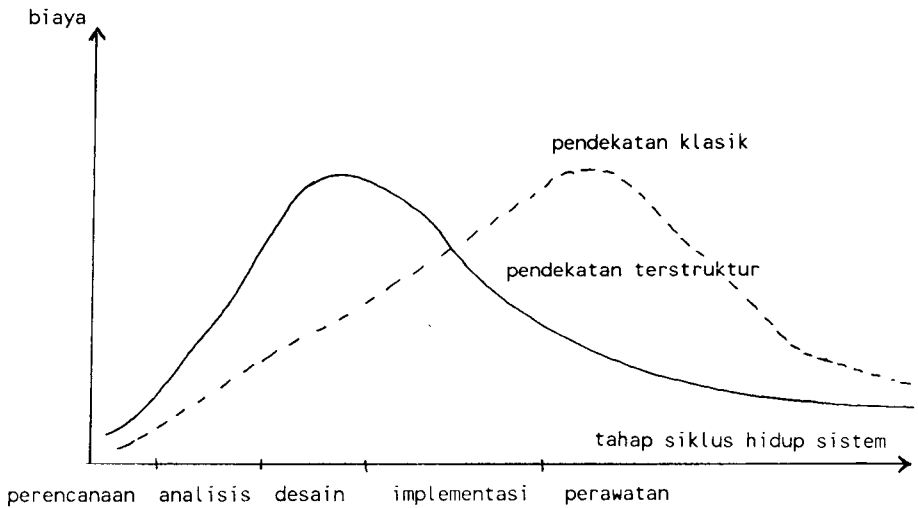
2. Biaya perawatan atau pemeliharaan sistem akan menjadi lebih mahal.

Biaya pengembangan sistem yang termahal adalah terletak di tahap perawatannya seperti tampak pada gambar biaya gunung es (*cost iceberg*) di gambar 2.2. Biaya perawatan terdapat di bagian terbawah dari gunung es. Karena tidak terlihat dari permukaan, maka banyak orang tidak menyadari bahwa biaya perawatan ini yang terbesar.



Gambar. 2.2. Gunung es biaya pengembangan sistem.

Dibandingkan dengan pendekatan terstruktur, biaya pemeliharaan sistem untuk pendekatan klasik akan lebih mahal seperti tampak pada gambar berikut ini.



Mahalnya biaya perawatan di pendekatan klasik disebabkan karena dokumentasi sistem yang dikembangkan kurang lengkap dan kurang terstruktur. Dokumentasi ini merupakan hasil dari alat-alat dan teknik-teknik yang digunakan. Karena pendekatan klasik kurang didukung dengan alat-alat dan teknik-teknik, maka dokumentasi menjadi tidak lengkap dan walaupun ada tetapi strukturnya kurang kelas, sehingga pada waktu pemeliharaan sistem menjadi kesulitan.

3. Kemungkinan kesalahan sistem besar.

Di pendekatan terstruktur, pengembangan sistem dilakukan dalam bentuk modul-modul yang terstruktur. Modul-modul ini akan lebih mudah dites secara terpisah dan kemudian pengujian dapat dilakukan pada integrasi semua modul untuk meyakinkan bahwa interaksi antar modul telah berfungsi semestinya. Pengujian sistem sebelum diterapkan merupakan hal yang kritis karena koreksi kesalahan sistem setelah diterapkan akan mengakibatkan pengeluaran biaya yang lebih besar. Beberapa penelitian menunjukkan bahwa sistem yang tidak dites selama tahap pengembangannya merupakan sumber utama dari kesalahan-kesalahan sistem. Pendekatan klasik tidak menyediakan kepada analis sistem cara untuk melakukan pengujian sistem, sehingga kemungkinan kesalahan-kesalahan sistem akan menjadi lebih besar.

4. Keberhasilan sistem kurang terjamin.

Penekanan dari pendekatan klasik adalah kerja dari personil-personil pengembang sistem, bukan pada pemakai sistem, padahal sekarang sudah disadari bahwa dukungan dan pemahaman dari pemakai sistem terhadap sistem yang sedang dikembangkan merupakan hal yang vital untuk keberhasilan proyek pengembangan sistem pada akhirnya. Salah satu dari kontribusi utama pendekatan terstruktur adalah partisipasi dan dukungan dari pemakai sistem.

Pendekatan klasik mengasumsikan bahwa analis sistem telah mengerti akan kebutuhan-kebutuhan pemakai sistem dengan jelas dan benar. Pengalaman telah menunjukkan bahwa di beberapa kasus, kebutuhan-kebutuhan pemakai sistem tidaklah selalu jelas dan benar menurut analis sistem. Pendekatan klasik karena kurang melibatkan pemakai sistem dalam pengembangan sistem, maka kebutuhan-kebutuhan pemakai sistem menjadi kurang sesuai dengan yang diinginkan dan sebagai akibatnya sistem yang diterapkan menjadi kurang berhasil.

5. Masalah dalam penerapan sistem.

Karena kurangnya keterlibatan pemakai sistem dalam tahapan pengembangan sistem, maka pemakai sistem hanya akan mengenal sistem yang baru pada tahap diterapkan saja. Sebagai akibatnya pemakai sistem akan menjadi kaget dan tidak terbiasa dengan sistem baru yang tiba-tiba dikenalkan. Sebagai akibat lebih lanjut, pemakai sistem akan menjadi frustrasi karena tidak dapat mengoperasikan sistem dengan baik.

2.4.1.2. Pendekatan terstruktur

Karena banyak terjadi permasalahan-permasalahan di pendekatan klasik, maka kebutuhan akan pendekatan pengembangan sistem yang lebih baik mulai terasa dibutuhkan. Sayangnya sampai sekarang masih banyak orang yang tidak menyadari bahwa hanya dengan mengikuti tahapan di *life cycle* saja tidak akan membuat pengembangan sistem informasi menjadi berhasil. Oleh karena itu diperlukan suatu pendekatan pengembangan sistem yang baru yang dilengkapi dengan beberapa alat dan teknik supaya membuatnya berhasil. Pendekatan ini yang dimulai dari awal tahun 1970 disebut dengan pendekatan terstruktur (*structured approach*). Pendekatan terstruktur dilengkapi dengan alat-alat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang strukturnya didefinisikan dengan baik dan jelas. Beberapa metodologi pengembangan sistem yang terstruktur telah banyak yang dikenalkan baik dalam buku-buku maupun oleh perusahaan-perusahaan konsultan pengembang sistem (lihat bab 2.5). Metodologi-metodologi ini mengenalkan penggunaan alat-alat dan teknik-teknik untuk mengembangkan sistem yang terstruktur. Kebutuhan dari suatu metodologi di pengembangan sistem informasi juga diungkapkan oleh C.H.P. Brooks dan kawan-kawan sebagai berikut:⁵

⁵C.H.P Brooks, P.J. Grouse, D.R. Jeffery dan M.J. Lawrence, *Information Systems Design*, (New Jersey: Prentice-Hall, 1982); dikutip oleh A. Ziya Aktas, *Structured Analysis & Design of Information Systems*, (New Jersey: Prentice-Hall, 1978), hal. 30.