**Sesi ke-12**

***Building Information System***

SISTEM SEBAGAI PERUBAHAN ORGANISASI YANGDIRENCANAKAN

Penggunaan sistem informasi baru adalah salah satu jenis perubahan organisasi yang direncanakan. Pengenalan sistem informasi baru melibatkan lebih dari sekedar perangkat keras dan perangkat lunak. Ini juga termasuk perubahan dalam pekerjaan, keterampilan, manajemen, dan organisasi. Ketika mendesain sistem informasi baru, maka organisasi juga didesain ulang. Pembangun sistem harus memahami bagaimana sistem akan mempengaruhi proses bisnis tertentu dan organisasi secara keseluruhan.

**SISTEM PENGEMBANGAN DAN PERUBAHAN ORGANISASI**

Teknologi informasi dapat mempromosikan berbagai tingkatan perubahan organisasi, mulai dari peningkatan hingga jangkauan luas. Ada empat jenis perubahan organisasi struktural yang dimungkinkan oleh teknologi informasi:

(1) otomatisasi, (2) rasionalisasi, (3) perancangan ulang proses bisnis, dan (4) pergeseran paradigma. Masing-masing membawa risiko dan imbalan yang berbeda.

Bentuk perubahan organisasi TI yang paling umum adalah otomatisasi. Aplikasi teknologi informasi pertama yang melibatkan membantu karyawan dengan melakukan tugas mereka secara lebih efisien dan efektif. Menghitung gaji dan daftar gaji, memberikan akses langsung kepada teller bank ke catatan setoran pelanggan, dan mengembangkan jaringan reservasi nasional untuk agen tiket pesawat adalah semua contoh otomatisasi risiko awal. Strategi-strategi yang relatif lamban ini memberikan hasil yang sedikit. Perubahan yang lebih cepat dan lebih komprehensif — seperti desain ulang dan perubahan paradigma — membawa imbalan yang tinggi tetapi menawarkan peluang kegagalan yang substansial.

Bentuk perubahan organisasi yang lebih dalam — yang segera menyusul dari otomatisasi awal — adalah rasionalisasi prosedur. Otomatisasi sering mengungkapkan kemacetan baru dalam produksi dan membuat pengaturan prosedur dan struktur yang ada sangat rumit. Rasionalisasi prosedur adalah penyederhanaan prosedur operasi standar. Sebagai contoh, sistem CIMB Bank untuk menangani rekening perbankan ritel efektif tidak hanya karena menggunakan teknologi komputer tetapi juga karena perusahaan menyederhanakan proses bisnis untuk membuka akun pelanggan. CIMB menyederhanakan alur kerjanya untuk memanfaatkan antarmuka dan perangkat lunak baru dari sistem untuk mengimpor data pribadi dari MyKad.

Rasionalisasi prosedur sering ditemukan dalam program untuk membuat serangkaian peningkatan kualitas berkelanjutan dalam produk, layanan, dan operasi, seperti manajemen kualitas total (TQM) dan enam sigma. Total quality management ment (TQM) membuat pencapaian kualitas menjadi tujuan itu sendiri dan tanggung jawab semua orang dan fungsi dalam suatu organisasi. TQM berasal dari konsep yang dikembangkan oleh ahli kualitas Amerika seperti W. Edwards Deming dan Joseph Juran, tetapi dipopulerkan oleh Jepang. Six sigma adalah ukuran kualitas yang spesifik, mewakili 3,4 cacat per juta peluang. Sebagian besar perusahaan tidak dapat mencapai tingkat kualitas ini, tetapi menggunakan enam sigma sebagai tujuan untuk mendorong program peningkatan kualitas.

Jenis perubahan organisasi yang lebih kuat adalah perancangan ulang proses bisnis, di mana proses bisnis dianalisis, disederhanakan, dan dirancang ulang. Redesain ulang proses bisnis mengatur ulang alur kerja, menggabungkan langkah-langkah untuk memotong limbah dan menghilangkan tugas-tugas yang berulang dan padat kertas. (Kadang-kadang desain baru menghilangkan pekerjaan juga.) Ini jauh lebih ambisius daripada rasionalisasi prosedur, membutuhkan visi baru tentang bagaimana proses itu harus diatur.

Contoh desain ulang proses bisnis yang banyak dikutip adalah pemrosesan tanpa henti dari Ford Motor Company, yang mengurangi jumlah pegawai dalam organisasi Akunting Pelanggan Amerika Utara sebesar 500 orang sebanyak 75 persen. Pegawai yang dibayar hutang digunakan untuk menghabiskan sebagian besar waktu mereka untuk menyelesaikan perbedaan antara pesanan pembelian, menerima dokumen, dan faktur. Ford mendesain ulang proses hutangnya sehingga departemen pembelian memasukkan pesanan pembelian ke database online yang dapat diperiksa oleh departemen penerima ketika barang pesanan tiba. Jika barang yang diterima sesuai dengan pesanan pembelian, sistem secara otomatis menghasilkan cek untuk hutang untuk dikirim ke vendor. Vendor tidak perlu mengirim faktur.

Merasionalisasi prosedur dan mendesain ulang proses bisnis terbatas pada bagian-bagian tertentu dari suatu bisnis. Sistem informasi baru pada akhirnya dapat mempengaruhi desain seluruh organisasi dengan mengubah cara organisasi menjalankan bisnisnya atau bahkan sifat bisnisnya. Misalnya, perusahaan pengangkutan dan transportasi jarak jauh Schneider National menggunakan sistem informasi baru untuk mengubah model bisnisnya. Schneider menciptakan bisnis baru yang mengelola logistik untuk perusahaan lain. Bentuk perubahan bisnis yang lebih radikal ini disebut pergeseran paradigma. Pergeseran paradigma melibatkan pemikiran ulang sifat bisnis dan sifat organisasi.

**PROSES REDUKSI PROSES BISNIS**

Seperti CIMB Group, dijelaskan dalam kasus bab pembukaan, banyak bisnis saat ini mencoba menggunakan teknologi informasi untuk meningkatkan proses bisnis mereka. Beberapa sistem ini memerlukan perubahan proses tambahan, tetapi yang lain memerlukan desain ulang proses bisnis yang lebih luas. Untuk menghadapi perubahan ini, organisasi beralih ke manajemen proses bisnis. **Manajemen proses bisnis** menyediakan berbagai alat dan metodologi untuk menganalisis proses yang ada, merancang proses baru, dan mengoptimalkan proses tersebut. BPM tidak pernah disimpulkan karena perbaikan proses membutuhkan perubahan terus-menerus. Perusahaan yang mempraktikkan manajemen proses bisnis melalui langkah-langkah berikut:

1**. Identifikasi proses untuk perubahan:** Salah satu keputusan strategis terpenting yang dapat dibuat oleh perusahaan adalah tidak memutuskan bagaimana menggunakan komputer untuk meningkatkan proses bisnis, tetapi memahami proses bisnis apa yang perlu ditingkatkan. Ketika sistem digunakan untuk memperkuat model bisnis yang salah atau proses bisnis, bisnis dapat menjadi lebih efisien dalam melakukan apa yang seharusnya tidak dilakukan. Akibatnya, perusahaan menjadi rentan terhadap pesaing yang mungkin telah menemukan model bisnis yang tepat. Waktu dan biaya yang besar juga dapat digunakan untuk meningkatkan proses bisnis yang berdampak kecil terhadap kinerja dan pendapatan perusahaan secara keseluruhan. Manajer perlu menentukan proses bisnis apa yang paling penting dan bagaimana meningkatkan proses ini akan membantu kinerja bisnis.

2. **Analisis proses yang ada:** Proses bisnis yang ada harus dimodelkan dan didokumentasikan, mencatat input, output, sumber daya, dan urutan kegiatan. Tim desain proses mengidentifikasi langkah-langkah redundan, tugas-tugas penting, kemacetan, dan inefisiensi lainnya.

**3. Merancang proses baru:** Setelah proses yang ada dipetakan dan diukur dalam hal waktu dan biaya, tim desain proses akan mencoba memperbaiki proses dengan merancang yang baru. Proses baru yang "mudah dilakukan" akan didokumentasikan dan dimodelkan untuk dibandingkan dengan proses lama.

4. **Menerapkan proses baru:** Setelah proses baru telah benar-benar dimodelkan dan dianalisis, itu harus diterjemahkan ke dalam satu set prosedur baru dan aturan kerja. Sistem informasi baru atau perangkat tambahan untuk sistem yang ada mungkin harus diimplementasikan untuk mendukung proses yang dirancang ulang. Proses baru dan sistem pendukung diluncurkan ke dalam organisasi bisnis. Ketika bisnis mulai menggunakan proses ini, masalah terungkap dan ditangani. Karyawan yang bekerja dengan proses ini dapat merekomendasikan perbaikan.

5. **Pengukuran berkelanjutan:** Setelah proses telah dilaksanakan dan dioptimalkan, perlu diukur secara terus-menerus. Mengapa? Proses dapat menurun seiring waktu ketika karyawan kembali menggunakan metode lama, atau mereka dapat kehilangan keefektifannya jika bisnis mengalami perubahan lain.

Meskipun banyak perbaikan proses bisnis yang inkremental dan berkelanjutan, ada kesempatan ketika perubahan yang lebih radikal harus terjadi. Contoh kami tentang toko buku fisik yang mendesain ulang proses pembelian buku sehingga dapat dilakukan secara online adalah contoh dari jenis perubahan radikal yang jauh menjangkau. Ketika diterapkan dengan benar, perancangan ulang proses bisnis menghasilkan peningkatan dramatis dalam produktivitas dan efisiensi, dan bahkan dapat mengubah cara bisnis dijalankan. Dalam beberapa kasus, itu mendorong "pergeseran paradigma" yang mentransformasikan sifat dari bisnis itu sendiri.

Ini benar-benar terjadi dalam penjualan buku ketika Amazon menantang toko-toko buku fisik tradisional dengan model ritel online-nya. Dengan secara radikal memikirkan kembali cara buku dapat dibeli dan dijual, Amazon dan toko buku online lainnya telah mencapai efisiensi luar biasa, pengurangan biaya, dan cara baru dalam berbisnis.

BPM menimbulkan tantangan. Eksekutif melaporkan bahwa penghalang tunggal terbesar untuk perubahan proses bisnis yang sukses adalah budaya organisasi. Karyawan tidak menyukai rutinitas yang tidak biasa dan sering mencoba menolak perubahan. Ini terutama berlaku untuk proyek-proyek di mana perubahan organisasi sangat ambisius dan jauh jangkauannya. Mengelola perubahan tidaklah sederhana dan juga tidak intuitif, dan perusahaan yang berkomitmen pada peningkatan proses yang ekstensif membutuhkan strategi manajemen perubahan yang baik (lihat Bab 14).

**Alat untuk Manajemen Proses Bisnis**

Lebih dari 100 perusahaan perangkat lunak menyediakan alat untuk berbagai aspek BPM, termasuk IBM, Oracle, dan TIBCO. Alat-alat ini membantu bisnis mengidentifikasi dan mendokumentasikan proses yang membutuhkan perbaikan, membuat model proses yang ditingkatkan, menangkap dan menegakkan aturan bisnis untuk melakukan proses, dan mengintegrasikan sistem yang ada untuk mendukung proses baru atau yang dirancang ulang. Perangkat lunak BPM juga menyediakan analitik untuk memverifikasi bahwa kinerja proses telah dilakukan ditingkatkan dan untuk mengukur dampak perubahan proses pada indikator kinerja bisnis utama.

Beberapa alat BPM mendokumentasikan dan memantau proses bisnis untuk membantu perusahaan mengidentifikasi inefisiensi, menggunakan perangkat lunak untuk terhubung dengan masing-masing sistem yang digunakan perusahaan untuk proses tertentu untuk mengidentifikasi titik masalah. Perusahaan reksadana Kanada AIC menggunakan perangkat lunak pemantauan Sajus BPM untuk memeriksa inkonsistensi dalam prosesnya untuk memperbarui akun setelah setiap transaksi klien. Sajus mengkhususkan diri dalam manajemen proses “berbasis tujuan”, yang berfokus pada menemukan penyebab masalah organisasi melalui pemantauan proses sebelum menerapkan alat untuk mengatasi masalah tersebut.

Kategori alat lainnya mengotomatisasi beberapa bagian dari proses bisnis dan menegakkan aturan bisnis sehingga karyawan melakukan proses itu secara lebih konsisten dan efisien.

Misalnya, American National Insurance Company (ANCO), yang menawarkan asuransi jiwa, asuransi kesehatan, asuransi kecelakaan properti, dan layanan investasi, menggunakan perangkat lunak alur kerja BPOM Pegasystems untuk merampingkan proses layanan pelanggan di empat kelompok bisnis. Perangkat lunak ini membangun aturan untuk memandu perwakilan layanan pelanggan melalui satu pandangan tunggal dari informasi pelanggan yang dikelola dalam berbagai sistem. Dengan menghilangkan kebutuhan untuk menyulap beberapa aplikasi secara bersamaan untuk menangani permintaan pelanggan dan agen, proses yang ditingkatkan meningkatkan kapasitas beban kerja perwakilan layanan pelanggan sebesar 192 persen.

Kategori alat ketiga membantu bisnis mengintegrasikan sistem yang ada untuk mendukung peningkatan proses. Mereka secara otomatis mengelola proses di seluruh bisnis, mengekstrak data dari berbagai sumber dan database, dan menghasilkan transaksi dalam berbagai sistem terkait. Sebagai contoh, Star Alliance dari 15 maskapai penerbangan, termasuk United dan Lufthansa, menggunakan BPM untuk membuat proses umum yang dibagikan oleh semua anggotanya dengan mengintegrasikan sistem yang ada. Satu proyek menciptakan layanan baru untuk frequent flyer pada maskapai anggota dengan mengkonsolidasikan 90 proses bisnis terpisah di sembilan maskapai dan 27 sistem warisan. Perangkat lunak BPM mendokumentasikan bagaimana setiap maskapai memproses informasi frequent flyer untuk membantu manajer maskapai memodelkan proses bisnis baru yang menunjukkan cara membagi data di antara berbagai sistem.

Sesi Interaktif di Organisasi menjelaskan bagaimana beberapa perusahaan menggunakan alat serupa untuk program manajemen proses bisnis mereka. Ketika Anda membaca kasus ini, pikirkan tentang jenis perubahan yang digunakan perusahaan dengan alat BPM ini dalam cara mereka menjalankan bisnis mereka.

IKHTISAR PENGEMBANGAN SISTEM

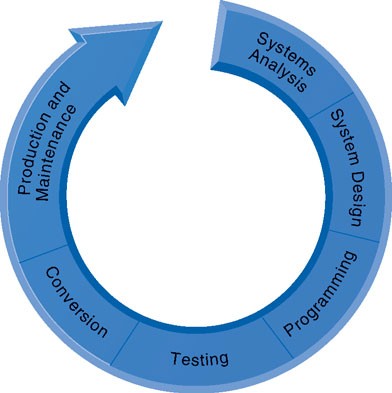
Sistem informasi baru adalah hasil dari proses pemecahan masalah organisasi. Sistem informasi baru dibangun sebagai solusi untuk beberapa jenis masalah atau serangkaian masalah yang dirasakan oleh organisasi. Masalahnya mungkin salah satu di mana manajer dan karyawan menyadari bahwa organisasi tidak berkinerja sebaik yang diharapkan, atau bahwa organisasi harus memanfaatkan peluang baru untuk melakukan lebih berhasil.

Kegiatan yang masuk ke dalam menghasilkan solusi sistem informasi untuk masalah atau peluang organisasi disebut pengembangan sistem. Pengembangan sistem adalah jenis masalah terstruktur yang diselesaikan dengan aktivitas yang berbeda. Kegiatan ini terdiri dari analisis sistem, desain sistem, pemrograman, pengujian, konversi, dan produksi dan pemeliharaan.

**ANALISIS SISTEM**

Analisis sistem adalah analisis masalah yang berusaha diselesaikan perusahaan dengan sistem informasi. Ini terdiri dari mendefinisikan masalah, mengidentifikasi penyebabnya, menentukan solusi, dan mengidentifikasi persyaratan informasi yang harus dipenuhi oleh solusi sistem.

**FIGURE THE SYSTEMS DEVELOPMENT PROCESS**



Building a system can be broken down into six core activities.

Analis sistem membuat peta jalan dari organisasi dan sistem yang ada, mengidentifikasi pemilik utama dan pengguna data bersama dengan perangkat keras dan perangkat lunak yang ada. Analis sistem kemudian merinci masalah sistem yang ada. Dengan memeriksa dokumen, kertas kerja, dan prosedur; mengamati operasi sistem; dan mewawancarai pengguna kunci dari sistem, analis dapat mengidentifikasi area masalah dan sasaran yang akan dicapai solusi. Seringkali solusinya membutuhkan membangun sistem informasi baru atau meningkatkan yang sudah ada.

Analisis sistem juga termasuk studi kelayakan untuk menentukan apakah solusi itu layak, atau dapat dicapai, dari sudut pandang finansial, teknis, dan organisasi. Studi kelayakan menentukan apakah sistem yang diusulkan diharapkan menjadi investasi yang baik, apakah teknologi yang dibutuhkan untuk sistem tersedia dan dapat ditangani oleh spesialis sistem informasi perusahaan, dan apakah organisasi dapat menangani perubahan yang diperkenalkan oleh sistem.

Biasanya, proses analisis sistem mengidentifikasi beberapa alternatif solusi yang organisasi dapat mengejar dan menilai kelayakan masing-masing. Laporan proposal sistem tertulis menggambarkan biaya dan manfaat, dan keuntungan dan kerugian, dari setiap alternatif. Terserah kepada manajemen untuk menentukan campuran biaya, manfaat, fitur teknis, dan dampak organisasi merupakan alternatif yang paling diinginkan.

**Menetapkan Persyaratan Informasi**

Mungkin tugas yang paling menantang dari analis sistem adalah untuk menentukan persyaratan informasi spesifik yang harus dipenuhi oleh solusi sistem yang dipilih. Pada tingkat yang paling dasar, persyaratan informasi dari sistem baru melibatkan identifikasi siapa yang membutuhkan informasi apa, di mana, kapan, dan bagaimana. Analisis persyaratan secara hati-hati menentukan tujuan dari sistem yang baru atau yang dimodifikasi dan mengembangkan deskripsi mendetail tentang fungsi-fungsi yang harus dilakukan oleh sistem baru. Analisis kebutuhan yang salah adalah penyebab utama kegagalan sistem dan biaya pengembangan sistem yang tinggi (lihat Bab 14). Sebuah sistem dirancang di sekitar seperangkat persyaratan yang salah akan harus dibuang karena kinerja yang buruk atau perlu menjalani modifikasi besar. Bagian 13.3 menjelaskan pendekatan alternatif untuk memunculkan persyaratan yang membantu meminimalkan masalah ini.

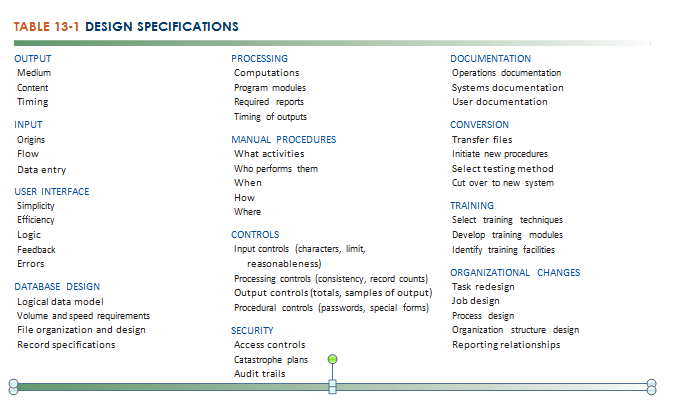
Beberapa masalah tidak memerlukan solusi sistem informasi tetapi membutuhkan penyesuaian dalam manajemen, pelatihan tambahan, atau penyempurnaan prosedur organisasi yang ada. Jika masalahnya terkait dengan informasi, analisis sistem masih diperlukan untuk mendiagnosis masalah dan sampai pada solusi yang tepat.

**DESAIN SISTEM**

Analisis sistem menggambarkan apa yang harus dilakukan sistem untuk memenuhi persyaratan informasi, dan desain sistem menunjukkan bagaimana sistem akan memenuhi tujuan ini. Rancangan sistem informasi adalah keseluruhan rencana atau model untuk sistem itu. Seperti cetak biru bangunan atau rumah, itu terdiri dari semua spesifikasi yang memberikan sistem bentuk dan strukturnya.

Perancang sistem merinci spesifikasi sistem yang akan memberikan fungsi yang diidentifikasi selama analisis sistem. Spesifikasi ini harus mengatasi semua komponen manajerial, organisasi, dan teknologi dari solusi sistem. Tabel 13-1 daftar jenis-jenis spesifikasi yang akan dihasilkan selama perancangan sistem.

Seperti rumah atau bangunan, sistem informasi mungkin memiliki banyak desain yang mungkin. Setiap desain mewakili perpaduan unik dari semua komponen teknis dan organisasi. Apa yang membuat satu desain lebih unggul dari yang lain adalah kemudahan dan efisiensi yang memenuhi persyaratan pengguna dalam satu set spesifik kendala teknis, organisasi, keuangan, dan waktu.



**Peran Pengguna Akhir**

Persyaratan informasi pengguna mendorong upaya pengembangan sistem secara keseluruhan. Pengguna harus memiliki kontrol yang cukup atas proses desain untuk memastikan bahwa sistem mencerminkan prioritas bisnis dan kebutuhan informasi mereka, bukan bias staf teknis. Bekerja pada desain meningkatkan pemahaman dan penerimaan pengguna dari sistem. Seperti yang kami jelaskan di Bab 14, keterlibatan pengguna yang tidak mencukupi dalam upaya desain merupakan penyebab utama kegagalan sistem. Namun, beberapa sistem memerlukan partisipasi pengguna yang lebih dalam desain daripada yang lain, dan Bagian 13.3 menunjukkan bagaimana metode pengembangan sistem alternatif mengatasi masalah partisipasi pengguna.

**MENYELESAIKAN PROSES PENGEMBANGAN SISTEM**

Langkah-langkah yang tersisa dalam proses pengembangan sistem menerjemahkan spesifikasi solusi yang ditetapkan selama analisis sistem dan desain menjadi sistem informasi yang beroperasi penuh. Langkah-langkah menyimpulkan ini terdiri dari pemrograman, pengujian, konversi, produksi, dan pemeliharaan.

**Pemrograman**

Selama tahap pemrograman, spesifikasi sistem yang disiapkan selama tahap desain diterjemahkan ke dalam kode program perangkat lunak. Saat ini, banyak organisasi tidak lagi melakukan pemrograman mereka sendiri untuk sistem baru. Sebaliknya, mereka membeli perangkat lunak yang memenuhi persyaratan untuk sistem baru dari sumber eksternal seperti paket perangkat lunak dari vendor perangkat lunak komersial, layanan perangkat lunak dari penyedia layanan aplikasi, atau perusahaan outsourcing yang mengembangkan perangkat lunak aplikasi kustom untuk klien mereka (lihat Bagian 13.3 ).

**Pengujian**

Pengujian menyeluruh dan menyeluruh harus dilakukan untuk memastikan apakah sistem menghasilkan hasil yang tepat. Pengujian menjawab pertanyaan, “Apakah sistem akan menghasilkan hasil yang diinginkan dalam kondisi yang diketahui?” Seperti disebutkan Bab 5, beberapa perusahaan mulai menggunakan layanan komputasi awan untuk pekerjaan ini.

Jumlah waktu yang diperlukan untuk menjawab pertanyaan ini secara tradisional diremehkan dalam perencanaan proyek sistem (lihat Bab 14). Pengujian memakan waktu: Data uji harus disiapkan dengan hati-hati, hasil yang ditinjau, dan koreksi yang dibuat dalam sistem. Dalam beberapa kasus, bagian dari sistem mungkin harus didesain ulang. Risiko yang diakibatkan oleh upaya mengoleskan langkah ini sangat besar.

Pengujian suatu sistem informasi dapat dibagi menjadi tiga jenis kegiatan: pengujian unit, pengujian sistem, dan pengujian penerimaan. Pengujian unit, atau pengujian program, terdiri dari pengujian setiap program secara terpisah dalam sistem. Dipercaya secara luas bahwa tujuan pengujian tersebut adalah untuk menjamin bahwa program itu bebas dari kesalahan, tetapi tujuan ini tidak mungkin secara realistis. Pengujian harus dilihat sebagai alat untuk menemukan kesalahan dalam program, berfokus pada menemukan semua cara untuk membuat program gagal. Begitu mereka menunjuk, masalah dapat diperbaiki.

**Pengujian sistem** menguji fungsi sistem informasi secara keseluruhan, mencoba untuk menentukan apakah modul diskrit akan berfungsi bersama seperti yang direncanakan dan apakah ada perbedaan antara cara sistem benar-benar bekerja dan cara itu dikandung. Di antara area yang diperiksa adalah waktu kinerja, kapasitas penyimpanan file dan penanganan beban puncak, kemampuan pemulihan dan restart, dan prosedur manual.

**Pengujian penerimaan** memberikan sertifikasi akhir bahwa sistem sudah siap untuk digunakan dalam pengaturan produksi. Pengujian sistem dievaluasi oleh pengguna dan ditinjau oleh manajemen. Ketika semua pihak puas bahwa sistem baru tersebut memenuhi standar mereka, sistem tersebut secara resmi diterima untuk pemasangan.

Tim pengembangan sistem bekerja dengan pengguna untuk menyusun rencana pengujian yang sistematis. Rencana pengujian mencakup semua persiapan untuk serangkaian tes yang baru saja kami jelaskan.

Gambar 13-5 menunjukkan contoh rencana pengujian. Kondisi umum yang sedang diuji adalah perubahan catatan. Dokumentasi ini terdiri dari serangkaian layar rencana uji yang disimpan pada database (mungkin database PC) yang cocok untuk aplikasi semacam ini.

**Konversi** adalah proses perubahan dari sistem lama ke sistem yang baru. Empat strategi konversi utama dapat digunakan: strategi paralel, strategi cutover langsung, strategi studi percontohan, dan strategi pendekatan bertahap.

Dalam **strategi paralel,** baik sistem lama maupun pengganti potensinya berjalan bersama untuk sementara waktu sampai semua orang yakin bahwa yang baru berfungsi dengan benar. Ini adalah pendekatan konversi yang paling aman karena, jika terjadi kesalahan atau gangguan pemrosesan, sistem lama masih dapat digunakan sebagai cadangan. Namun, pendekatan ini sangat mahal, dan staf atau sumber daya tambahan mungkin diperlukan untuk menjalankan sistem ekstra.

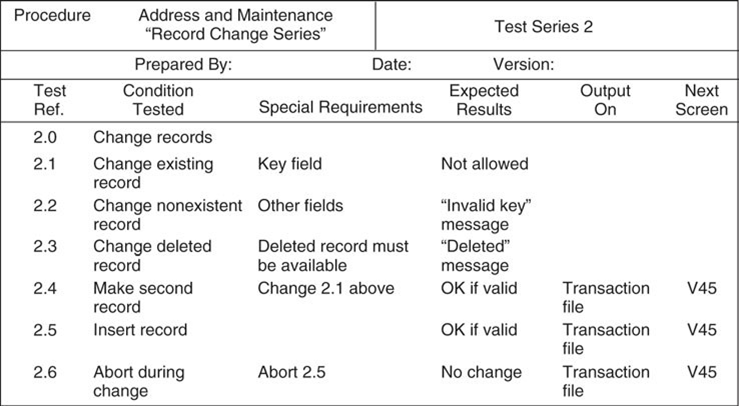
**Strategi cutover langsung** menggantikan sistem lama sepenuhnya dengan yang baru sistem pada hari yang ditentukan. Ini adalah pendekatan yang sangat berisiko yang berpotensi lebih mahal daripada menjalankan dua sistem secara paralel jika masalah serius dengan sistem baru ditemukan. Tidak ada sistem lain untuk jatuh kembali. Dislokasi, gangguan, dan biaya koreksi bisa sangat besar.

**Strategi studi percontohan** memperkenalkan sistem baru hanya pada area terbatas organisasi, seperti satu departemen atau unit operasi. Ketika versi percontohan ini selesai dan bekerja dengan lancar, ini dipasang di seluruh organisasi, baik secara simultan atau bertahap.

Strategi pendekatan bertahap memperkenalkan sistem baru secara bertahap, baik oleh fungsi atau oleh unit organisasi. Jika, misalnya, sistemnya. diperkenalkan oleh fungsi, sistem penggajian baru mungkin dimulai dengan pekerja per jam yang dibayar mingguan, diikuti enam bulan kemudian dengan menambahkan karyawan yang digaji (yang dibayar bulanan) ke sistem. Jika sistem diperkenalkan oleh unit organisasi, kantor pusat perusahaan mungkin dikonversi pertama, diikuti oleh unit operasi yang terpencil empat bulan kemudian.

Pindah dari sistem lama ke sistem baru mengharuskan pengguna akhir dilatih untuk menggunakan sistem baru. Dokumentasi rinci menunjukkan bagaimana sistem bekerja baik dari sudut pandang teknis dan pengguna akhir diselesaikan selama waktu konversi untuk digunakan dalam pelatihan dan operasi sehari-hari. Kurangnya pelatihan dan dokumentasi yang tepat berkontribusi pada kegagalan sistem, sehingga bagian dari proses pengembangan sistem ini sangat penting.

**FIGURE 13-5 A SAMPLE TEST PLAN TO TEST A RECORD CHANGE**

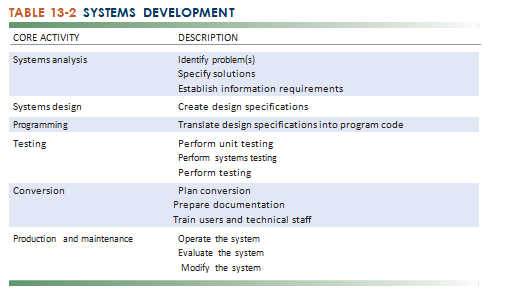


Ketika mengembangkan rencana uji, sangat penting untuk memasukkan berbagai kondisi yang akan diuji, persyaratan untuk setiap kondisi yang diuji, dan hasil yang diharapkan. Rencana pengujian memerlukan masukan dari pengguna akhir dan spesialis sistem informasi.

**Produksi dan Pemeliharaan**

Setelah sistem baru dipasang dan konversi selesai, sistem dikatakan dalam tahap produksi. Selama tahap ini, sistem akan ditinjau oleh pengguna dan spesialis teknis untuk menentukan seberapa baik mereka telah memenuhi tujuan awalnya dan untuk memutuskan apakah ada revisi atau modifikasi yang sesuai. Dalam beberapa kasus, dokumen audit pasca implementasi formal disiapkan. Setelah sistem telah diperbaiki, itu harus dipertahankan ketika sedang dalam produksi untuk memperbaiki kesalahan, memenuhi persyaratan, atau meningkatkan efisiensi pemrosesan. Perubahan dalam perangkat keras, perangkat lunak, dokumentasi, atau prosedur untuk sistem produksi untuk memperbaiki kesalahan, memenuhi persyaratan baru, atau meningkatkan efisiensi pemrosesan disebut perawatan.

Sekitar 20 persen waktu yang digunakan untuk pemeliharaan digunakan untuk debugging atau memperbaiki masalah produksi darurat. 20 persen lainnya berkaitan dengan perubahan data, file, laporan, perangkat keras, atau perangkat lunak sistem. Tetapi 60 persen dari semua pekerjaan pemeliharaan terdiri dari membuat peningkatan pengguna, meningkatkan dokumentasi, dan mengkode ulang komponen sistem untuk efisiensi pemrosesan yang lebih besar. Jumlah pekerjaan dalam kategori ketiga masalah pemeliharaan dapat dikurangi secara signifikan melalui analisis sistem dan praktik desain yang lebih baik. Tabel 13-2 merangkum kegiatan pengembangan sistem.



**PEMODELAN DAN PERANCANGAN SISTEM: METODOLOGI STRUKTUR DAN OBJEK-ORIENTASI**

Ada metodologi alternatif untuk pemodelan dan perancangan sistem. Metodologi terstruktur dan pengembangan berorientasi objek adalah yang paling menonjol.

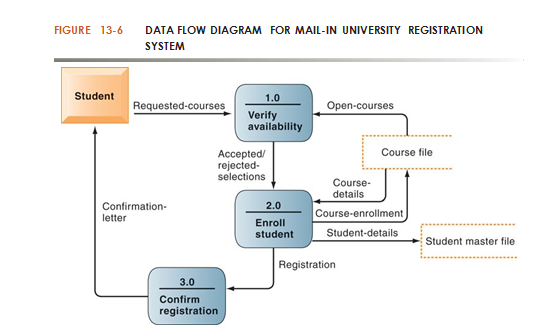
**Metodologi Terstruktur**

Metodologi terstruktur telah digunakan untuk mendokumentasikan, menganalisis, dan merancang sistem informasi sejak tahun 1970-an. Terstruktur mengacu pada fakta bahwa teknik ini selangkah demi selangkah, dengan setiap langkah membangun pada yang sebelumnya. Metodologi terstruktur adalah top-down, berkembang dari level tertinggi, paling abstrak ke level terendah detail — dari yang umum hingga yang spesifik.

Metode pengembangan terstruktur berorientasi pada proses, berfokus terutama pada pemodelan proses, atau tindakan yang menangkap, menyimpan, memanipulasi, dan mendistribusikan data sebagai aliran data melalui sistem. Metode-metode ini memisahkan data dari proses. Prosedur pemrograman terpisah harus ditulis setiap kali seseorang ingin mengambil tindakan pada bagian data tertentu. Prosedur bertindak berdasarkan data yang dilewatkan oleh program.

Alat utama untuk mewakili proses komponen sistem dan aliran data di antaranya adalah diagram alur data (DFD). Diagram alur data menawarkan model grafik logis dari arus informasi, mempartisi suatu sistem ke dalam modul yang menunjukkan tingkat detail yang dapat dikelola. Ini secara ketat menentukan proses atau transformasi yang terjadi dalam setiap modul dan antarmuka yang ada di antara mereka.

Gambar 13-6 menunjukkan diagram aliran data sederhana untuk sistem registrasi kursus universitas mail-in. Kotak bulat mewakili proses, yang menggambarkan



Sistem ini memiliki tiga proses: Verifikasi ketersediaan (1.0), Daftarkan siswa (2.0), dan Konfirmasikan pendaftaran (3.0). Nama dan konten masing-masing arus data muncul berdekatan dengan setiap panah. Ada satu entitas eksternal dalam sistem ini: siswa. Ada dua toko data: file master siswa dan file kursus.

transformasi data. Kotak persegi mewakili entitas eksternal, yang merupakan penggagas atau penerima informasi yang terletak di luar batas-batas sistem yang dimodelkan. Persegi panjang terbuka mewakili penyimpanan data, yang merupakan persediaan data manual atau otomatis. Panah mewakili arus data, yang menunjukkan pergerakan antara proses, entitas eksternal, dan penyimpanan data. Mereka berisi paket data dengan nama atau konten dari setiap aliran data yang tercantum di samping panah.

Diagram alur data ini menunjukkan bahwa siswa mengirimkan formulir pendaftaran dengan nama mereka, nomor identifikasi, dan jumlah mata kuliah yang ingin mereka ambil. Dalam proses 1.0, sistem memverifikasi bahwa setiap kursus yang dipilih masih terbuka dengan merujuk file kursus universitas. File tersebut membedakan kursus yang terbuka dari yang telah dibatalkan atau diisi. Proses 1.0 kemudian menentukan pilihan siswa mana yang dapat diterima atau ditolak. Proses 2.0 mendaftarkan siswa dalam kursus yang telah diterima. Ini memperbarui file program universitas dengan nama siswa dan nomor identifikasi dan menghitung ulang ukuran kelas. Jika pendaftaran maksimum telah tercapai, nomor kursus ditandai sebagai ditutup. Proses 2.0 juga memperbarui file master siswa universitas dengan informasi tentang siswa baru atau perubahan alamat. Proses 3.0 kemudian mengirimkan setiap pelamar mahasiswa surat konfirmasi pendaftaran yang berisi daftar mata kuliah dimana dia terdaftar dan mencatat seleksi kursus yang tidak dapat dipenuhi.

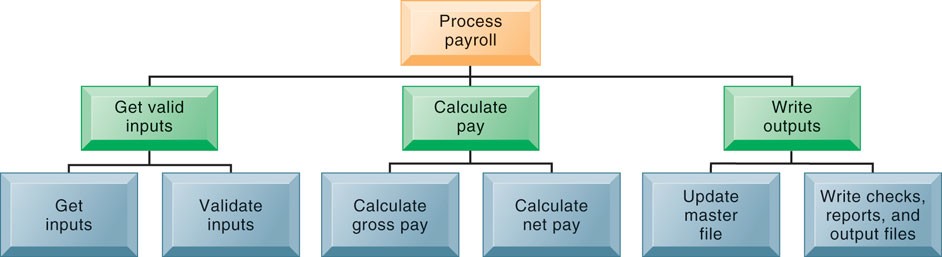
Diagram dapat digunakan untuk menggambarkan proses tingkat yang lebih tinggi serta rincian tingkat yang lebih rendah. Melalui diagram alir data yang diratakan, proses yang kompleks dapat dipecah menjadi tingkat detail yang berurutan. Seluruh sistem dapat dibagi menjadi subsistem dengan diagram aliran data tingkat tinggi. Setiap subsistem, pada gilirannya, dapat dibagi menjadi subsistem tambahan dengan diagram aliran data tingkat kedua, dan subsistem tingkat bawah dapat dipecah lagi sampai tingkat terendah dari detail telah tercapai.

Alat lain untuk analisis terstruktur adalah kamus data, yang berisi informasi tentang setiap bagian data dan pengelompokan data dalam suatu sistem (lihat Bab 6). Kamus data mendefinisikan isi aliran data dan penyimpanan data sehingga pembangun sistem memahami dengan tepat potongan data apa yang dikandungnya. Spesifikasi proses menggambarkan transformasi yang terjadi dalam tingkat terendah diagram aliran data. Mereka mengekspresikan logika untuk setiap proses.

Dalam metodologi terstruktur, desain perangkat lunak dimodelkan menggunakan bagan struktur hirarkis. Bagan struktur adalah bagan top-down, yang menunjukkan setiap tingkat desain, hubungannya dengan tingkat lain, dan tempatnya dalam struktur desain keseluruhan. Desain pertama mempertimbangkan fungsi utama dari suatu program atau sistem, kemudian memecah fungsi ini menjadi subfungsi, dan menguraikan setiap subfungsi sampai tingkat terendah detail telah tercapai. Gambar 13-7 menunjukkan bagan struktur tingkat tinggi untuk sistem penggajian. Jika sebuah desain memiliki terlalu banyak level untuk masuk ke dalam satu struktur bagan, itu bisa dipecah lebih jauh pada bagan struktur yang lebih rinci. Bagan struktur dapat mendokumentasikan satu program, satu sistem (satu set program), atau bagian dari satu program.

**Pengembangan Berorientasi Objek**

Metode terstruktur berguna untuk proses pemodelan, tetapi tidak menangani pemodelan data dengan baik. Mereka juga memperlakukan data dan proses sebagai entitas yang secara logis terpisah, sedangkan di dunia nyata pemisahan seperti itu tampaknya tidak alami. Konvensi pemodelan yang berbeda digunakan untuk analisis (diagram aliran data) dan untuk desain (bagan struktur).

FIGURE 13-7 HIGH-LEVEL STRUCTURE CHART FOR A PAYROLL

This structure chart shows the highest or most abstract level of design for a payroll system, providing an overview of the entire system.

**Pengembangan berorientasi objek** mengatasi masalah ini. Pengembangan berorientasi objek menggunakan objek sebagai unit dasar dari analisis dan desain sistem. Objek menggabungkan data dan proses khusus yang beroperasi pada data tersebut. Data yang dienkapsulasi dalam suatu objek dapat diakses dan dimodifikasi hanya dengan operasi, atau metode, yang terkait dengan objek itu. Alih-alih mengirimkan data ke prosedur, program mengirim pesan untuk objek untuk melakukan operasi yang sudah tertanam di dalamnya. Sistem dimodelkan sebagai kumpulan objek dan hubungan di antara mereka. Karena logika pemrosesan berada di dalam objek, bukan dalam program perangkat lunak terpisah, objek harus berkolaborasi satu sama lain untuk membuat sistem bekerja.

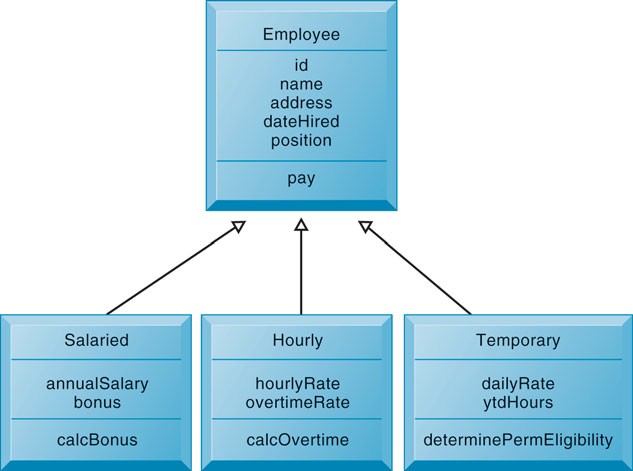
Pemodelan berorientasi objek didasarkan pada konsep kelas dan warisan.

Objek milik kelas tertentu, atau kategori umum objek serupa, memiliki fitur dari kelas itu. Kelas objek pada gilirannya dapat mewarisi semua struktur dan perilaku kelas yang lebih umum dan kemudian menambahkan variabel dan perilaku yang unik untuk setiap objek. Kelas-kelas baru objek dibuat dengan memilih kelas yang ada dan menentukan bagaimana kelas baru berbeda dari kelas yang ada, bukan mulai dari awal setiap kali.

Kita dapat melihat bagaimana kelas dan pewarisan bekerja di Gambar 13-8, yang menggambarkan hubungan di antara kelas tentang karyawan dan bagaimana mereka dibayar. Karyawan adalah leluhur yang sama, atau superclass, untuk tiga kelas lainnya. Gaji, Jam, dan Sementara adalah subclass dari Karyawan. Nama kelas berada di kompartemen atas, atribut untuk setiap kelas berada di bagian tengah setiap kotak, dan daftar operasi berada di bagian bawah setiap kotak. Fitur-fitur yang dimiliki oleh semua karyawan (id, nama, alamat, tanggal yang disewa, posisi, dan pembayaran) disimpan di superclass Karyawan, sedangkan setiap fitur toko subkelas khusus untuk tipe karyawan tertentu. Khusus untuk karyawan per jam, misalnya, adalah tarif per jam dan tarif lembur. Garis yang solid dari subkelas ke superclass adalah jalur generalisasi yang menunjukkan bahwa subclass Gari, Jam, dan Sementara memiliki fitur umum yang dapat digeneralisasikan ke dalam Karyawan superclass.

Pembangunan berorientasi objek lebih bersifat iteratif dan inkremental daripada pengembangan terstruktur tradisional. Selama analisis, pembangun sistem mendokumentasikan persyaratan fungsional sistem, menentukan sifat-sifatnya yang paling penting dan apa yang harus dilakukan oleh sistem yang diusulkan. Interaksi antara sistem dan penggunanya dianalisis untuk mengidentifikasi objek, yang meliputi data dan proses. Fase desain berorientasi objek menggambarkan bagaimana objek akan

**FIGURE 13-8 CLASS AND INHERITANCE**



This figure illustrates how classes inherit the common features of their superclass.

berperilaku dan bagaimana mereka akan berinteraksi dengan yang lain. Objek serupa dikelompokkan bersama untuk membentuk kelas, dan kelas dikelompokkan ke dalam hierarki di mana subkelas mewarisi atribut dan metode dari superclass-nya.

Sistem informasi diimplementasikan dengan menerjemahkan desain ke dalam kode program, menggunakan kembali kelas yang sudah tersedia di perpustakaan objek perangkat lunak yang dapat digunakan kembali dan menambahkan yang baru dibuat selama fase desain berorientasi objek. Implementasi juga dapat melibatkan pembuatan database berorientasi objek. Sistem yang dihasilkan harus diuji dan dievaluasi secara menyeluruh.

Karena objek dapat digunakan kembali, pengembangan berorientasi objek berpotensi mengurangi waktu dan biaya perangkat lunak penulisan karena organisasi dapat menggunakan kembali objek perangkat lunak yang telah dibuat sebagai blok bangunan untuk aplikasi lain. Sistem baru dapat dibuat dengan menggunakan beberapa objek yang ada, mengubah yang lain, dan menambahkan beberapa objek baru. Kerangka berorientasi objek telah dikembangkan untuk menyediakan aplikasi yang dapat digunakan kembali, semicomplete yang dapat disesuaikan oleh organisasi ke dalam aplikasi yang telah selesai.

**Teknik Perangkat Lunak Computer-Aided**

Computer-aided software engineering (CASE) —sekali disebut rekayasa sistem dibantu komputer — menyediakan perangkat lunak untuk mengotomatisasi metodologi yang baru saja kami jelaskan untuk mengurangi jumlah pekerjaan berulang yang perlu dilakukan pengembang. Perangkat CASE juga memfasilitasi pembuatan dokumentasi yang jelas dan koordinasi upaya pengembangan tim. Anggota tim dapat membagikan pekerjaan mereka dengan mudah dengan mengakses file masing-masing untuk meninjau atau mengubah apa yang telah dilakukan. Manfaat produktivitas yang sederhana juga dapat dicapai jika alat digunakan dengan benar.

Alat KASUS menyediakan fasilitas grafis otomatis untuk menghasilkan diagram dan diagram, layar dan pembuat laporan, kamus data, fasilitas pelaporan yang luas, alat analisis dan pengecekan, generator kode, dan generator dokumentasi. Secara umum, alat CASE mencoba meningkatkan produktivitas dan kualitas dengan:

* Menerapkan metodologi pengembangan standar dan disiplin desain
* Meningkatkan komunikasi antara pengguna dan spesialis teknis
* Mengatur dan menghubungkan komponen desain dan menyediakan akses cepat ke mereka menggunakan repositori desain
* Mengotomasi bagian-bagian analisis dan desain yang membosankan dan rawan kesalahan
* Mengotomasi pembuatan dan pengujian kode dan mengontrol peluncuran

Alat CASE berisi fitur untuk memvalidasi diagram desain dan spesifikasi. Alat CASE dengan demikian mendukung desain iteratif dengan mengotomatisasi revisi dan perubahan dan menyediakan fasilitas prototyping. Sebuah penyimpanan informasi KASUS menyimpan semua informasi yang ditentukan oleh para analis selama proyek. Repositori termasuk diagram aliran data, bagan struktur, diagram hubungan entitas, definisi data, spesifikasi proses, layar dan format laporan, catatan dan komentar, dan hasil tes.

Agar dapat digunakan secara efektif, alat CASE memerlukan disiplin organisasi. Setiap anggota proyek pengembangan harus mematuhi seperangkat konvensi penamaan dan standar serta metodologi pengembangan. Alat CASE terbaik menegakkan metode dan standar umum, yang dapat mengurangi penggunaannya dalam situasi di mana disiplin organisasi kurang.

**PENDEKATAN ALTERNATIF SISTEM**

Sistem berbeda dalam hal ukuran dan kompleksitas teknologinya dan dalam hal masalah organisasi yang harus dipecahkan. Sejumlah pendekatan pembangunan sistem telah dikembangkan untuk menangani perbedaan-perbedaan ini. Bagian ini menjelaskan metode alternatif ini: siklus hidup sistem tradisional, prototipe, paket perangkat lunak aplikasi, pengembangan pengguna akhir, dan alih daya.

SIKLUS HIDUP SISTEM TRADISIONAL

Siklus hidup sistem adalah metode tertua untuk membangun sistem informasi. Metodologi siklus hidup adalah pendekatan bertahap untuk membangun sistem, membagi pengembangan sistem menjadi tahap formal. Spesialis pengembangan sistem memiliki pendapat yang berbeda tentang bagaimana mempartisi tahap pengembangan sistem, tetapi mereka kira-kira sesuai dengan tahapan pengembangan sistem yang baru saja kita jelaskan.

Metoda siklus hidup sistem mempertahankan pembagian kerja yang sangat formal antara pengguna akhir dan spesialis sistem informasi. Spesialis teknis, seperti analis sistem dan programer, bertanggung jawab untuk banyak analisis sistem, desain, dan pekerjaan implementasi; pengguna akhir terbatas untuk memberikan persyaratan informasi dan meninjau pekerjaan staf teknis. Siklus hidup juga menekankan spesifikasi formal dan dokumen, sehingga banyak dokumen dihasilkan selama proyek sistem.

Siklus hidup sistem masih digunakan untuk membangun sistem kompleks besar yang memerlukan analisis persyaratan yang ketat dan formal, spesifikasi yang telah ditentukan, dan kontrol ketat atas proses pembangunan sistem. Namun, pendekatan siklus hidup sistem bisa mahal, memakan waktu, dan tidak fleksibel. Meskipun pembangun sistem dapat bolak-balik di antara tahap di mana tugas dalam satu tahap selesai sebelum bekerja untuk tahap berikutnya dimulai. Kegiatan dapat diulang, tetapi volume dokumen baru harus dihasilkan dan langkah-langkah kembali jika persyaratan dan spesifikasi perlu direvisi. Ini mendorong pembekuan spesifikasi relatif awal dalam proses pengembangan. Pendekatan siklus hidup juga tidak cocok untuk banyak sistem desktop kecil, yang cenderung kurang terstruktur dan lebih individual.

**PROTOTYPING**

Prototyping terdiri dari membangun sistem eksperimental dengan cepat dan tidak jelas bagi pengguna akhir untuk dievaluasi. Dengan berinteraksi dengan prototipe, pengguna bisa mendapatkan ide yang lebih baik dari persyaratan informasi mereka. Prototipe yang didukung oleh pengguna dapat digunakan sebagai template untuk membuat sistem akhir.

Prototipe adalah versi kerja dari suatu sistem informasi atau bagian dari sistem, tetapi itu dimaksudkan hanya menjadi model awal. Setelah operasional, prototipe akan lebih disempurnakan hingga sesuai dengan kebutuhan pengguna. Setelah desain selesai, prototipe dapat dikonversi ke sistem produksi yang dipoles.

Proses membangun desain awal, mencoba, memperbaikinya, dan mencoba lagi disebut proses iteratif pengembangan sistem karena langkah-langkah yang diperlukan untuk membangun suatu sistem dapat diulang terus-menerus. Prototipe lebih bersifat iteratif daripada siklus hidup konvensional, dan secara aktif mempromosikan perubahan desain sistem. Telah dikatakan bahwa prototipe mengganti pengerjaan ulang yang tidak direncanakan dengan iterasi terencana, dengan setiap versi lebih akurat mencerminkan kebutuhan pengguna.

**Langkah-langkah dalam Prototyping**

Gambar 13-9 menunjukkan model empat langkah dari proses prototyping, yang terdiri dari hal-hal berikut:

Langkah 1: Identifikasi persyaratan dasar pengguna. Perancang sistem (biasanya spesialis sistem informasi) bekerja dengan pengguna hanya cukup lama untuk menangkap kebutuhan informasi dasar pengguna.

Langkah 2: Kembangkan prototipe awal. Perancang sistem membuat prototipe kerja dengan cepat, menggunakan alat untuk menghasilkan perangkat lunak dengan cepat.

Langkah 3: Gunakan prototipe. Pengguna didorong untuk bekerja dengan sistem untuk menentukan seberapa baik prototipe memenuhi kebutuhannya dan membuat saran untuk meningkatkan prototipe.

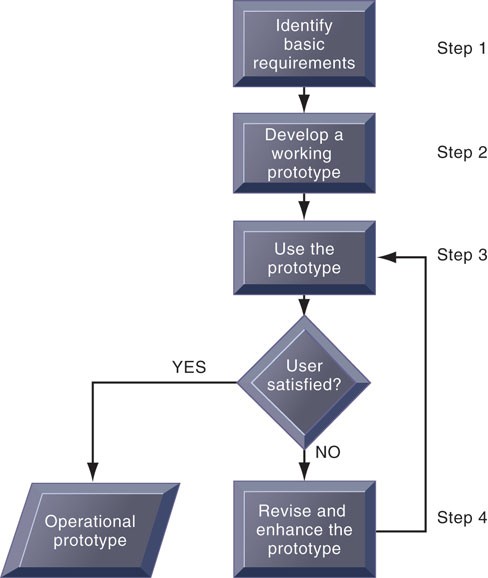
Langkah 4: Merevisi dan meningkatkan prototipe. Pembuat sistem mencatat semua perubahan yang diminta pengguna dan memurnikan prototipe yang sesuai. Setelah prototipe direvisi, siklus kembali ke Langkah 3. Langkah 3 dan 4 diulang sampai pengguna puas.

Ketika tidak ada lagi iterasi yang diperlukan, prototipe yang disetujui kemudian menjadi prototipe operasional yang melengkapi spesifikasi akhir untuk aplikasi. Kadang-kadang prototipe diadopsi sebagai versi produksi dari sistem.

**Keuntungan dan Kerugian dari Prototyping**

Prototyping paling berguna ketika ada ketidakpastian tentang persyaratan atau solusi desain dan sering digunakan untuk merancang antarmuka pengguna akhir sistem informasi (bagian dari sistem yang digunakan pengguna akhir, seperti tampilan online dan layar entri data, laporan, atau Halaman web). Karena proto-typing mendorong keterlibatan pengguna akhir yang intens di seluruh sistem

**FIGURE 13-9 THE PROTOTYPING PROCESS**



Proses mengembangkan prototipe dapat dipecah menjadi empat langkah. Karena prototipe dapat dikembangkan dengan cepat dan murah, pembuat sistem dapat melalui beberapa iterasi, mengulangi langkah 3 dan 4, untuk memperbaiki dan meningkatkan prototipe sebelum tiba di operasi akhir. Siklus hidup pengembangan, itu lebih cenderung menghasilkan sistem yang memenuhi persyaratan pengguna.

Namun, prototipe cepat dapat mengabaikan langkah-langkah penting dalam pengembangan sistem. Jika prototipe yang diselesaikan berfungsi dengan baik, manajemen mungkin tidak melihat kebutuhan untuk pemrograman ulang, mendesain ulang, atau dokumentasi dan pengujian lengkap untuk membangun sistem produksi yang dipoles. Beberapa dari sistem yang tergesa-gesa ini mungkin tidak mudah menampung data dalam jumlah besar atau sejumlah besar pengguna dalam lingkungan produksi.

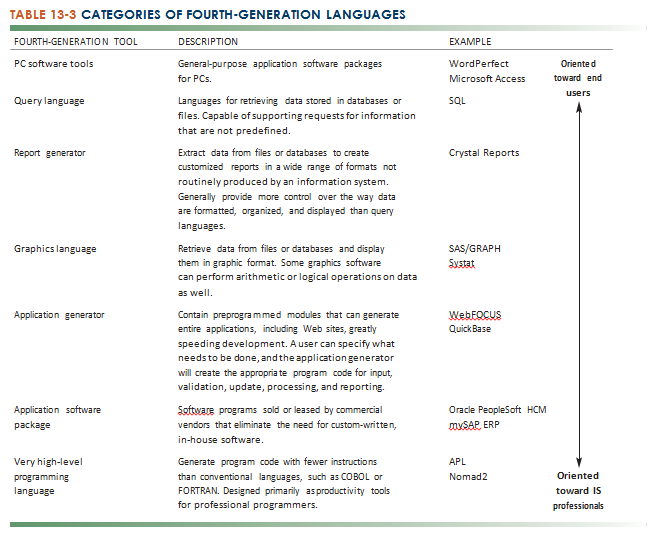
**PENGEMBANGAN PENGGUNA**

Beberapa jenis sistem informasi dapat dikembangkan oleh pengguna akhir dengan sedikit atau tanpa bantuan formal dari spesialis teknis. Fenomena ini disebut end-user development. Serangkaian alat perangkat lunak yang dikategorikan sebagai bahasa generasi keempat memungkinkan hal ini. Bahasa generasi keempat adalah perangkat lunak yang memungkinkan pengguna akhir untuk membuat laporan atau mengembangkan aplikasi perangkat lunak dengan bantuan teknis minimal atau tanpa bantuan teknis. Beberapa alat generasi keempat ini juga meningkatkan produktivitas programmer profesional.

Bahasa generasi keempat cenderung tidak prosedural, atau kurang prosedural, daripada bahasa pemrograman konvensional. Bahasa prosedural memerlukan spesifikasi urutan langkah, atau prosedur, yang memberi tahu komputer apa yang harus dilakukan dan bagaimana melakukannya. Bahasa nonprocedural hanya perlu menentukan apa yang harus diselesaikan daripada memberikan rincian tentang bagaimana melaksanakan tugas.

Tabel 13-3 menunjukkan bahwa ada tujuh kategori bahasa generasi keempat: perangkat lunak PC, bahasa query, pembuat laporan, bahasa grafis, generator aplikasi, paket perangkat lunak aplikasi, dan bahasa pemrograman tingkat tinggi. Tabel menunjukkan alat yang diperintahkan dalam hal kemudahan penggunaan oleh pengguna akhir non-pemrograman. Pengguna akhir kemungkinan besar bekerja dengan perangkat lunak PC dan bahasa query. Bahasa pertanyaan adalah alat perangkat lunak yang menyediakan jawaban online langsung atas permintaan informasi yang tidak ditentukan sebelumnya, seperti “Siapa perwakilan penjualan berkinerja tertinggi?” Bahasa kueri sering dikaitkan dengan perangkat lunak manajemen data dan sistem manajemen basis data (lihat Bab 6).

Secara keseluruhan, sistem yang dikembangkan oleh pengguna akhir dapat diselesaikan lebih cepat daripada yang dikembangkan melalui siklus kehidupan sistem konvensional. Mengizinkan pengguna untuk menentukan kebutuhan bisnis mereka sendiri meningkatkan pengumpulan persyaratan dan sering mengarah ke tingkat keterlibatan pengguna yang lebih tinggi dan kepuasan dengan sistem. Namun, alat generasi keempat masih tidak dapat menggantikan alat-alat konvensional untuk beberapa aplikasi bisnis karena mereka tidak dapat dengan mudah menangani pemrosesan sejumlah besar transaksi atau aplikasi dengan logika prosedural dan pembaruan persyaratan yang luas.



Komputasi pengguna akhir juga menimbulkan risiko organisasi karena terjadi di luar mekanisme tradisional untuk manajemen dan pengendalian sistem informasi. Ketika sistem dibuat dengan cepat, tanpa metodologi pengembangan formal, pengujian dan dokumentasi mungkin tidak memadai. Kontrol atas data dapat hilang dalam sistem di luar departemen sistem informasi tradisional. Untuk membantu organisasi memaksimalkan manfaat pengembangan aplikasi pengguna akhir, manajemen harus mengontrol pengembangan aplikasi pengguna akhir dengan mewajibkan pembenaran biaya proyek sistem informasi pengguna akhir dan dengan menetapkan perangkat keras, perangkat lunak, dan standar kualitas untuk aplikasi yang dikembangkan pengguna.

**PAKET PERANGKAT LUNAK APLIKASI DAN OUTSOURCING**

Bab 5 menunjukkan bahwa banyak perangkat lunak saat ini tidak dikembangkan sendiri tetapi dibeli dari sumber eksternal. Perusahaan dapat menyewa perangkat lunak dari penyedia layanan perangkat lunak, mereka dapat membeli paket perangkat lunak dari vendor komersial, atau mereka dapat memiliki aplikasi khusus yang dikembangkan oleh perusahaan pemasok luar.

**Paket Perangkat Lunak Aplikasi**

Selama beberapa dekade terakhir, banyak sistem telah dibangun di atas fondasi paket perangkat lunak aplikasi. Banyak aplikasi yang umum untuk semua organisasi bisnis — misalnya, pembayaran gaji, piutang, buku besar umum, atau kontrol inventaris. Untuk fungsi universal dengan proses standar yang tidak berubah banyak dari waktu ke waktu, sistem umum akan memenuhi persyaratan banyak organisasi.

Jika paket perangkat lunak dapat memenuhi sebagian besar persyaratan organisasi, perusahaan tidak perlu menulis perangkat lunaknya sendiri. Perusahaan dapat menghemat waktu dan uang dengan menggunakan program perangkat lunak pra-cetak, pradesain, pretested dari paket. Vendor paket menyediakan banyak pemeliharaan dan dukungan berkelanjutan untuk sistem, termasuk peningkatan untuk menjaga sistem sejalan dengan perkembangan teknis dan bisnis yang berkelanjutan.

Jika sebuah organisasi memiliki persyaratan unik yang tidak dikenali oleh paket, banyak paket menyertakan kemampuan untuk penyesuaian. Fitur kustomisasi memungkinkan paket perangkat lunak dimodifikasi untuk memenuhi persyaratan unik organisasi tanpa merusak integritas perangkat lunak yang dikemas. Jika banyak penyesuaian diperlukan, pemrograman tambahan dan pekerjaan kustomisasi dapat menjadi sangat mahal dan memakan waktu sehingga mereka meniadakan banyak keuntungan dari paket perangkat lunak.

Ketika sistem dikembangkan menggunakan paket perangkat lunak aplikasi, analisis sistem akan menyertakan upaya evaluasi paket. Kriteria evaluasi yang paling penting adalah fungsi yang disediakan oleh paket, fleksibilitas, persahabatan pengguna, sumber daya perangkat keras dan perangkat lunak, persyaratan basis data, upaya pemasangan dan pemeliharaan, dokumentasi, kualitas vendor, dan biaya. Proses evaluasi paket sering didasarkan pada Permintaan untuk Proposal (RFP), yang merupakan daftar rinci pertanyaan yang diajukan ke vendor perangkat lunak yang dikemas.

Ketika paket perangkat lunak dipilih, organisasi tidak lagi memiliki kontrol penuh atas proses desain sistem. Alih-alih menyesuaikan spesifikasi desain sistem secara langsung dengan kebutuhan pengguna, upaya desain akan terdiri dari mencoba untuk mencetak persyaratan pengguna untuk menyesuaikan dengan fitur paket. Jika persyaratan organisasi bertentangan dengan cara kerja paket danpaket tidak dapat dikustomisasi, organisasi harus menyesuaikan dengan paket dan mengubah prosedurnya.

Sesi Interaktif tentang Teknologi menggambarkan pengalaman Zimbra, perusahaan perangkat lunak yang memilih solusi paket perangkat lunak untuk sistem otomasi pemasarannya yang baru. Ketika Anda membaca kasus ini, perhatikan masalah manajemen, teknologi, dan organisasi yang harus ditangani ketika perusahaan memilih solusi paket perangkat lunak baru.

**Outsourcing**

Jika suatu perusahaan tidak ingin menggunakan sumber daya internalnya untuk membangun atau mengoperasikan sistem informasi, ia dapat mengalihkan pekerjaannya ke organisasi eksternal yang berspesialisasi dalam menyediakan layanan ini. Cloud computing dan penyedia SaaS, yang kami jelaskan di Bab 5, adalah salah satu bentuk alih daya. Perusahaan berlangganan akan menggunakan perangkat lunak dan perangkat keras komputer yang disediakan oleh layanan sebagai platform teknis untuk sistem mereka. Dalam bentuk lain dari outsourcing, perusahaan dapat menyewa vendor eksternal untuk merancang dan menciptakan perangkat lunak untuk sistemnya, tetapi perusahaan itu akan mengoperasikan sistem pada komputernya sendiri. Vendor outsourcing mungkin domestik atau di negara lain. Pengalihan domestik didorong terutama oleh fakta bahwa perusahaan outsourcing memiliki keterampilan, sumber daya, dan aset yang tidak dimiliki klien mereka. Instalasi sistem manajemen rantai pasokan baru di perusahaan yang sangat besar mungkin memerlukan mempekerjakan tambahan 30 hingga 50 orang dengan keahlian khusus dalam perangkat lunak manajemen rantai suplai, berlisensi dari vendor. Daripada mempekerjakan karyawan baru permanen, yang sebagian besar membutuhkan pelatihan ekstensif dalam paket perangkat lunak, dan kemudian merilisnya setelah sistem baru dibangun, lebih masuk akal, dan sering lebih murah, untuk mengalihkan pekerjaan ini untuk Periode 12 bulan.

Dalam kasus outsourcing lepas pantai, keputusannya cenderung lebih digerakkan oleh biaya. Seorang programmer yang terampil di India atau Rusia menghasilkan sekitar USD $ 9.000 per tahun, dibandingkan dengan $ 65.000 per tahun untuk programmer yang sebanding di Amerika Serikat. Internet dan teknologi komunikasi berbiaya rendah telah secara drastis mengurangi biaya dan kesulitan koordinasi kerja tim global di lokasi yang jauh. Selain penghematan biaya, banyak perusahaan outsourcing luar negeri menawarkan aset dan keterampilan teknologi kelas dunia. Inflasi upah di luar Amerika Serikat baru-baru ini mengikis sebagian dari keuntungan ini, dan beberapa pekerjaan telah kembali ke Amerika Serikat.

Namun demikian, ada peluang yang sangat kuat bahwa di beberapa titik dalam karir Anda, Anda akan bekerja dengan agen outsourcing lepas pantai atau tim global. Perusahaan Anda kemungkinan besar akan mendapat manfaat dari outsourcing jika membutuhkan waktu untuk mengevaluasi semua risiko dan memastikan outsourcing sesuai untuk kebutuhan khususnya. Setiap perusahaan yang meng-outsource aplikasinya harus benar-benar memahami proyek, termasuk persyaratannya, metode penerapan, manfaat yang diantisipasi, komponen biaya, dan metrik untuk mengukur kinerja.

Banyak perusahaan meremehkan biaya untuk mengidentifikasi dan mengevaluasi vendor layanan teknologi informasi, untuk transisi ke vendor baru, untuk meningkatkan metode pengembangan perangkat lunak internal untuk mencocokkan vendor outsourcing, dan untuk memantau vendor untuk memastikan mereka memenuhi kewajiban kontrak mereka. Perusahaan akan perlu mengalokasikan sumber daya untuk mendokumentasikan persyaratan, mengirim RFP, biaya perjalanan, negosiasi kontrak, dan manajemen proyek. Klaim para ahli dibutuhkan dari tiga bulan hingga setahun penuh untuk sepenuhnya mentransfer pekerjaan ke mitra lepas pantai dan memastikan vendor benar-benar memahami bisnis Anda.

**PENGEMBANGAN BERBASIS KOMPONEN DAN LAYANAN WEB**

Telah menjelaskan beberapa manfaat dari pengembangan berorientasi objek untuk membangun sistem yang dapat merespon lingkungan bisnis yang berubah dengan cepat, termasuk aplikasi Web. Untuk lebih mempercepat pembuatan perangkat lunak, kelompok objek telah dirakit untuk menyediakan komponen perangkat lunak untuk fungsi umum seperti antarmuka pengguna grafis atau kemampuan pemesanan online yang dapat dikombinasikan untuk membuat aplikasi bisnis skala besar. Pendekatan pengembangan perangkat lunak ini disebut pengembangan berbasis komponen, dan memungkinkan sistem yang akan dibangun dengan merakit dan mengintegrasikan komponen perangkat lunak yang sudah ada. Semakin banyak, komponen perangkat lunak ini berasal dari layanan cloud. Bisnis menggunakan pengembangan berbasis komponen untuk membuat aplikasi e-commerce mereka dengan menggabungkan komponen yang tersedia secara komersial untuk keranjang belanja, otentikasi pengguna, mesin pencari, dan katalog dengan potongan perangkat lunak untuk kebutuhan bisnis unik mereka sendiri.

**Web Ser Vces dan Service-Oriented Computing**

Bab 5 memperkenalkan layanan Web sebagai komponen perangkat lunak yang dapat digabungkan secara longgar dan dapat digunakan kembali yang disampaikan menggunakan Extensible Markup Language (XML) dan protokol dan standar terbuka lainnya yang memungkinkan satu aplikasi untuk berkomunikasi dengan yang lain tanpa pemrograman khusus yang diperlukan untuk berbagi data dan layanan. Selain mendukung integrasi internal dan eksternal dari sistem, layanan Web dapat digunakan sebagai alat untuk membangun aplikasi sistem informasi baru atau meningkatkan sistem yang ada. Karena layanan perangkat lunak ini menggunakan seperangkat standar universal, mereka berjanji untuk menjadi lebih murah dan lebih sulit untuk ditenun bersama daripada komponen kepemilikan.

Layanan web dapat menjalankan fungsi tertentu sendiri, dan mereka juga dapat melibatkan layanan Web lainnya untuk menyelesaikan transaksi yang lebih kompleks, seperti memeriksa kredit, pengadaan, atau memesan produk. Dengan membuat komponen perangkat lunak yang dapat berkomunikasi dan berbagi data terlepas dari sistem operasi, bahasa pemrograman, atau perangkat klien, layanan Web dapat memberikan penghematan biaya yang signifikan dalam membangun sistem sambil membuka peluang baru untuk berkolaborasi dengan perusahaan lain.