PERTEMUAN 1

SISTEM

A. TUJUAN PEMBELAJARAN

Pada pertemuan ini dijelaskan tentang apa itu pengertian sistem, karakteristik sistem, klasifikasi sistem, juga pendekatan dan metodologi pengembangan sistem. Setelah pertemuan ini, mahasiswa mampu mendeskripsikan klasifikasi dan metodologi pengembangan sistem.

B. URAIAN MATERI

1. Pengertian Sistem

Istilah "sistem" diambil dari bahasa Yunani: syst¯ema, yang berarti "menempatkan bersama." Berbagai domain bisnis dan teknik memiliki definisi sistem. Teks ini mendefinisikan sistem sebagai:

“Sistem adalah sebuah set elemen yang dapat dioperasikan dan saling terintegrasi, masing-masing ditentukan dengan secara eksplisit dan kemampuan terbatas, bekerja secara sinergis untuk melakukan pemrosesan nilai tambah yang memungkinkan Pengguna untuk memenuhi kebutuhan berorientasi misi dalam lingkungan operasi yang ditentukan dengan hasil tertentu dan kemungkinan sukses”.

Untuk membantu memahami dasar pemikiran definisi ini, mari periksa setiap bagian secara mendetail.

2. Dasar Pemikiran Definisi Sistem

Definisi di atas menangkap sejumlah poin diskusi utama tentang sistem. Mari periksa dasar untuk setiap frasa dalam definisi.

a. Yang dimaksud dengan "himpunan terintegrasi" adalah bahwa sistem, menurut definisi, terdiri dari hierarki tingkat elemen fisik, entitas, atau komponen.

b. Yang dimaksud dengan "elemen yang dapat dioperasikan", yang dimaksud adalah bahwa elemen dalam struktur sistem haruslah kompatibel satu sama lain dalam bentuk, fit, dan fungsi, misalnya. Elemen sistem termasuk peralatan (misalnya, perangkat keras dan perangkat lunak, personel, fasilitas, kendala operasi, dukungan), pemeliharaan, persediaan, suku cadang, pelatihan, sumber daya, data prosedural, sistem eksternal, dan apa pun yang mendukung pencapaian misi.
c. Dengan setiap elemen yang memiliki "kemampuan yang ditentukan dan dibatasi secara eksplisit", yang dimaksud adalah setiap elemen harus bekerja untuk mencapai beberapa tujuan tingkat yang lebih tinggi atau misi yang bertujuan. Sistem kontribusi elemen terhadap kinerja sistem secara keseluruhan harus ditentukan secara eksplisit. Ini membutuhkan kemampuan kinerja operasional dan fungsional untuk setiap elemen sistem dapat diidentifikasi dan secara eksplisit dibatasi ke tingkat kota tertentu yang memungkinkan elemen tersebut berada dianalisis, dirancang, dikembangkan, diuji, diverifikasi, dan divalidasi baik secara berdiri sendiri atau sebagai bagian dari sistem terintegrasi.
d. Yang dimaksud dengan “bekerja secara sinergis” adalah tujuan mengintegrasikan himpunan elemen adalah memanfaatkan kapabilitas kapabilitas elemen individu untuk mencapai level yang lebih tinggi kemampuan yang tidak dapat dicapai sebagai elemen yang berdiri sendiri.
e. Dengan "pemrosesan nilai tambah", maksudnya adalah faktor-faktor seperti biaya operasional, utilitas, kesesuaian, ketersediaan, dan efisiensi menuntut agar setiap operasi sistem dan tugas menambah nilai inputnya ketersediaan, dan menghasilkan keluaran yang berkontribusi pada pencapaian misi sistem secara keseluruhan hasil dan tujuan kinerja.
f. Dengan "memungkinkan pengguna untuk secara terprediksi memenuhi kebutuhan berorientasi misi", maksudnya adalah setiap sistem memiliki tujuan (yaitu, alasan keberadaan) dan nilai bagi pengguna. Nilainya mungkin laba atas investasi (ROI) relatif untuk memenuhi kebutuhan operasional atau untuk memuaskan misi dan tujuan sistem.
g. Dengan "dalam lingkungan operasi yang ditentukan", maksudnya adalah untuk ekonomi, hasil, dan untuk alasan kelangsungan hidup, setiap sistem harus memiliki operasi yang ditentukan yaitu, terikat lingkungan Hidup.
h. Yang dimaksud dengan "dengan hasil yang ditentukan" adalah pemangku

kepentingan sistem (Pengguna, pemegang saham, pemilik, dll.) mengharapkan sistem memberikan hasil. Perilaku yang diamati, produk, oleh produk, atau layanan, misalnya, harus berorientasi pada hasil, dapat diukur, dapat diukur, dan bisa dibuktikan.

i. Yang dimaksud dengan "dan kemungkinan sukses" adalah pencapaian hasil tertentu melibatkan tingkat ketidakpastian atau risiko. Jadi, derajat keberhasilan ditentukan oleh berbagai macam faktor kinerja seperti keandalan, ketergantungan, ketersediaan, pemeliharaan, keberlanjutan kemampuan, mematikan, dan bertahan hidup.

3. Karakteristik Sistem

Dalam mengkarakterisasi sistem, terutama untuk pemasaran atau analisis, ada empat tipe dasar karakteristik antara lain adalah:

a. Karakteristik Umum

Fitur tingkat tinggi dari suatu sistem adalah karakteristik umumnya. Karakter umum sering dinyatakan dalam brosur pemasaran di mana fitur-fitur utama ditekankan untuk menangkap klien atau minat pelanggan. Karakteristik umum sering kali memiliki beberapa kesamaan di beberapa contoh atau model sistem. Perhatikan contoh berikut:

1) Karakteristik Umum Mobil Tersedia dalam model dua pintu atau empat pintu; convertible atau sedan; AC nyaman; penangguhan independen; jendela berwarna, kota 22mpg, jalan raya 30mpg.
2) Karakteristik Umum Pesawat Fanjet, 50 penumpang, jangkauan 2000 mil laut, kemampuan IFR.
3) Karakteristik Umum Perusahaan atau Organisasi 200 karyawan; staf dengan 20 PhD, 50 Master, dan 30 derajat BS; penjualan tahunan sebesar $ 500 M per tahun.
4) Karakteristik Umum Jaringan Arsitektur server-klien, platform PC dan Unix, keamanan firewall, akses dial-up jarak jauh, tulang punggung Ethernet, struktur file jaringan (NFS).

b. Karakteristik Operasi atau Perilaku

Pada tingkat kerincian di bawah karakteristik umum, sistem memiliki karakteristik operasi itu menjelaskan fitur sistem yang terkait dengan

kegunaan, survivabilitas, dan kinerja untuk lingkungan operasi yang ditentukan. Perhatikan contoh berikut:

1) Karakteristik Operasi Mobil Kemampuan manuver, radius putar 18 kaki, 0 hingga 60 mph dalam 6 detik, dll.
2) Karakteristik Operasi Pesawat Aplikasi segala cuaca, kecepatan, dll.
3) Otorisasi Karakteristik Operasi Jaringan, waktu akses, latensi, dll.

c. Karakteristik fisik

Setiap sistem digambarkan oleh karakteristik fisik yang berhubungan dengan atribut nonfungsional tersebut sebagai atribut ukuran, berat, warna, kapasitas, dan antarmuka. Perhatikan contoh berikut:

1) Karakteristik Fisik Mobil 2000 lbs, berat trotoar volume kargo 14,0 cu ft, 43,1 inci (maks). ruang kaki depan, kapasitas bahan bakar 17,1 gals, mesin 240 tenaga kuda pada 6250 rpm, turbo, tersedia dalam 10 warna.
2) Karakteristik Fisik Perusahaan atau Organisasi Ruang kantor seluas 5.000 kaki persegi, 15 komputer jaringan, 100.000 kaki persegi gudang.
3) Karakteristik Fisik Jaringan tulang punggung Ethernet 1.0 Mb, topografi, router, gateway.

d. Karakteristik Estetika Sistem

Karakteristik umum, operasi, dan fisik adalah parameter kinerja yang objektif. Namun, bagaimana dengan karakteristik subjektif? Disebut juga sebagai karakteristik estetika sistem karena mereka berhubungan dengan "tampilan dan nuansa" dari suatu sistem. Jelas, ini termasuk psikologis, sosiologis, dan perspektif budaya yang terkait dengan daya tarik untuk preferensi Pengguna, Acquirer, atau Pemilik Sistem. Dengan demikian, beberapa pembeli membuat keputusan independen, sementara yang lain dipengaruhi oleh eksternal sistem (yaitu, pembeli lain) dalam hal-hal yang berkaitan dengan komunitas atau status perusahaan, citra, dan kesukaan.

Definisi tentang sistem menunjukkan beberapa karakteristik yang ada di semua sistem, antara lain: organisasi (urutan), interaksi, saling ketergantungan, integrasi, dan tujuan sentral.

1) Organisasi

Organisasi menyiratkan struktur dan ketertiban. Ini adalah pengaturan komponen yang membantu mencapai tujuan. Dalam desain sistem bisnis, misalnya, hubungan hierarki yang dimulai dengan presiden di atas dan mengarah ke bawah hingga pekerja kerah biru merepresentasikan struktur organisasi. Pengaturan seperti itu menggambarkan hubungan sistem-subsistem, mendefinisikan struktur otoritas, menentukan aliran formal komunikasi dan memformalkan rantai komando. Seperti bijaksana, sistem komputer dirancang di sekitar perangkat input, unit pemrosesan pusat, perangkat output dan satu atau lebih unit penyimpanan. Ketika dihubungkan bersama, mereka bekerja sebagai satu kesatuan sistem untuk menghasilkan informasi.

2) Interaksi

Interaksi mengacu pada cara setiap komponen berfungsi dengan komponen lain dari sistem. Dalam sebuah organisasi, misalnya, pembelian harus berinteraksi dengan produksi, periklanan dengan penjualan, dan penggajian dengan personel. Dalam sistem komputer, unit pengolah pusat harus berinteraksi dengan perangkat input untuk memecahkan masalah. Pada gilirannya, memori utama menyimpan program dan data yang digunakan unit aritmatika untuk komputasi. Keterkaitan antara komponen ini memungkinkan komputer untuk bekerja.

3) Saling ketergantungan

Saling ketergantungan berarti bahwa bagian-bagian dari organisasi atau sistem komputer saling bergantung satu sama lain. Mereka dikoordinasikan dan dihubungkan bersama menurut sebuah rencana. Satu subsistem bergantung pada masukan dari subsistem lain agar dapat berfungsi dengan benar: yaitu, keluaran dari satu subsistem adalah masukan yang diperlukan untuk subsistem lainnya. Saling ketergantungan ini sangat penting dalam kerja sistem.

Sistem informasi terintegrasi dirancang untuk melayani kebutuhan pengguna yang berwenang (kepala departemen, manajer, dll.) Untuk akses dan pengambilan cepat melalui terminal jarak jauh. Saling

ketergantungan antara subsistem personel dan pengguna organisasi terlihat jelas.

Singkatnya, tidak ada subsistem yang dapat berfungsi secara terpisah karena bergantung pada data (input) yang diterimanya dari subsistem lain untuk melakukan tugas yang diperlukan. Saling ketergantungan selanjutnya diilustrasikan oleh aktivitas dan dukungan dari analis sistem, pemrogram, dan staf operasi di pusat komputer. Keputusan untuk mengkomputerisasi aplikasi dimulai oleh pengguna, dianalisis dan dirancang oleh analis, diprogram dan diuji oleh pemrogram, dan dijalankan oleh operator komputer. Tak satu pun orang dapat melakukan properti tanpa masukan yang diperlukan dari orang lain di subsistem pusat komputer.

4) Integrasi

Integrasi mengacu pada holisme sistem. Sintesis mengikuti analisis untuk mencapai tujuan utama organisasi. Integrasi berkaitan dengan bagaimana suatu sistem terikat bersama. Ini lebih dari sekadar berbagi bagian atau lokasi fisik. Ini berarti bahwa bagian-bagian dari sistem bekerja bersama di dalam sistem meskipun setiap bagian menjalankan fungsinya yang unik. Integrasi yang berhasil biasanya akan menghasilkan efek sinergis dan dampak total yang lebih besar daripada jika setiap komponen bekerja secara terpisah.

5) Tujuan Utama

Karakteristik terakhir dari suatu sistem adalah tujuan utamanya. Tujuan mungkin nyata atau dinyatakan. Meskipun tujuan yang dinyatakan mungkin merupakan tujuan yang sebenarnya, tidak jarang organisasi menyatakan satu tujuan dan beroperasi untuk mencapai tujuan yang lain. Poin pentingnya adalah bahwa pengguna harus mengetahui tujuan utama dari aplikasi komputer sejak awal analisis untuk desain dan konversi yang sukses. Pertimbangan politik dan organisasi seringkali mengaburkan tujuan sebenarnya. Ini berarti bahwa analis harus mengatasi hambatan tersebut untuk mengidentifikasi tujuan sebenarnya dari perubahan yang diusulkan.

4. Klasifikasi Sistem

Kerangka acuan di mana seseorang memandang suatu sistem terkait dengan penggunaan pendekatan sistem untuk analisis. Sistem telah diklasifikasikan dengan cara yang berbeda. Klasifikasi umum antara lain adalah: (1) fisik atau abstrak, (2) terbuka atau tertutup, dan (3) sistem informasi "buatan manusia".

a. Sistem fisik atau abstrak

Sistem fisik adalah entitas berwujud yang mungkin statis atau dinamis dalam operasi. Sebagai contoh, bagian fisik pusat komputer adalah petugas, meja, dan kursi yang memudahkan pengoperasian komputer. Mereka bisa dilihat dan dihitung; mereka statis. Sebaliknya, komputer yang diprogram adalah sistem dinamis. Data, program, keluaran, dan aplikasi berubah sesuai permintaan pengguna atau prioritas informasi yang diminta berubah.

Sistem abstrak adalah entitas konseptual atau non-fisik. Mereka mungkin sesederhana rumus hubungan antara set variabel atau model konseptualisasi abstrak dari situasi fisik. Model adalah representasi dari sistem nyata atau terencana. Penggunaan model memudahkan analis untuk memvisualisasikan hubungan dalam sistem yang diteliti. Tujuannya adalah untuk menunjukkan elemen penting dan keterkaitan kunci dari sistem yang kompleks.

b. Sistem Terbuka atau Tertutup

Klasifikasi lain dari sistem didasarkan pada tingkat kemandiriannya. Sistem terbuka memiliki banyak antarmuka dengan lingkungannya. Ini memungkinkan interaksi melintasi batasnya; ia menerima masukan dari dan mengirimkan keluaran ke luar. Sistem informasi termasuk dalam kategori ini, karena harus beradaptasi dengan perubahan tuntutan pengguna. Sebaliknya, sistem tertutup diisolasi dari pengaruh lingkungan. Pada kenyataannya, sistem yang sepenuhnya tertutup jarang terjadi. Dalam analisis sistem, organisasi, aplikasi, dan komputer selalu terbuka, sistem dinamis dipengaruhi oleh lingkungannya.

Fokus pada karakteristik sistem terbuka sangat tepat waktu mengingat masalah bisnis saat ini dengan penipuan komputer, pelanggaran privasi, kontrol keamanan, dan etika dalam komputasi. Sedangkan aspek teknis dari

analisis sistem berhubungan dengan rutinitas internal dalam area aplikasi pengguna, analisis sistem sebagai sistem terbuka cenderung memperluas cakupan analisis untuk hubungan antara area pengguna dan pengguna lain dan faktor lingkungan yang harus dipertimbangkan sebelum yang baru. sistem akhirnya disetujui. Lebih lanjut, bersikap terbuka terhadap saran menyiratkan bahwa analis harus fleksibel dan sistem yang dirancang harus responsif terhadap perubahan kebutuhan pengguna dan lingkungan.

Lima karakteristik penting dari sistem terbuka dapat diidentifikasi sebagai berikut:

1) **Masukan dari luar**: Sistem terbuka menyesuaikan diri dan mengatur sendiri. Saat berfungsi dengan baik, sistem terbuka mencapai kondisi stabil atau keseimbangan. Dalam perusahaan ritel, misalnya, kondisi mapan terjadi ketika barang dibeli dan dijual tanpa stok atau kelebihan stok. Kenaikan harga pokok memaksa kenaikan harga yang sebanding atau penurunan biaya operasi. Ini respons memberi perusahaan kondisi mapannya.
2) **Entropi**: Semua sistem dinamis cenderung menurun seiring waktu, mengakibatkan entropi atau hilangnya energi. Sistem terbuka menolak entropi dengan mencari input baru atau memodifikasi proses untuk kembali ke kondisi mapan. Dalam contoh kita, tidak ada reaksi terhadap kenaikan harga barang dagangan yang membuat bisnis tidak menguntungkan yang dapat memaksanya menjadi bangkrut - keadaan disorganisasi.
3) **Proses, keluaran, dan siklus**: Sistem terbuka menghasilkan keluaran yang berguna dan beroperasi dalam siklus, mengikuti jalur aliran kontinu.
4) **Diferensiasi**: Sistem terbuka memiliki kecenderungan ke arah peningkatan spesialisasi fungsi dan diferensiasi komponen yang lebih besar. Dalam bisnis, peran orang dan mesin cenderung ke arah spesialisasi yang lebih besar dan interaksi yang lebih besar. Karakteristik ini menawarkan alasan yang kuat untuk meningkatkan nilai konsep sistem dalam pemikiran analis sistem.
5) **Ekuifinalitas**: Istilah ini menyiratkan bahwa tujuan dicapai melalui tindakan yang berbeda dan jalur yang berbeda. Dalam kebanyakan sistem, ada lebih banyak konsensus tentang tujuan daripada pada jalur

untuk mencapai tujuan.

Memahami karakteristik sistem membantu analis untuk mengidentifikasi peran mereka dan menghubungkan aktivitas mereka dengan pencapaian tujuan perusahaan saat mereka melakukan proyek sistem. Analis sendiri adalah bagian dari organisasi. Mereka memiliki peluang untuk menyesuaikan organisasi dengan perubahan melalui aplikasi terkomputerisasi sehingga sistem tidak "rusak". Kunci dari proses ini adalah umpan balik informasi dari pengguna utama sistem baru serta dari manajemen puncak.

Tema proses perancangan sistem informasi banyak meminjam dari pengetahuan umum tentang teori sistem. Tujuannya adalah membuat sistem lebih efisien dengan memodifikasi tujuannya atau mengubah keluarannya.

c. Sistem Informasi Buatan Manusia

Informasi dapat mengurangi ketidakpastian akan suatu keadaan ataupun peristiwa. Misalnya, informasi bahwa dilaut angin sedang tenang dapat mengurangi keraguan bahwa perjalanan perahu akan menyenangkan. Sistem informasi merupakan dasar interaksi antara pengguna dan system analis. Ini memberikan instruksi, perintah dan *feedback*. Ini menentukan sifat hubungan di antara pemangku kebijakan. Bahkan, bisa dipandang sebagai pusat keputusan bagi setiap orang di semua *level*. Dari dasar ini, sistem informasi dapat didefinisikan sebagai seperangkat prosedur dan sistem Operasi yang dirancang di sekitar kriteria berbasis pengguna untuk menghasilkan informasi dan mengkomunikasikannya kepada pengguna untuk perencanaan, pengendalian dan kinerja. Dalam analisis sistem, penting untuk diingat bahwa mempertimbangkan sistem alternatif berarti meningkatkan satu atau lebih kriteria ini.

5. Pendekatan dan Metodologi Pengembangan Sistem

Ada banyak metodologi pengembangan sistem yang berbeda, dan masing-masing unik, berdasarkan urutan dan fokusnya ditempatkan pada setiap fase SDLC. Beberapa metodologi adalah standar formal yang digunakan oleh instansi pemerintah, sedangkan yang lain telah dikembangkan oleh perusahaan konsultan untuk dijual kepada klien. Banyak organisasi memiliki

metodologi internal itu telah diasah selama bertahun-tahun, dan mereka menjelaskan dengan tepat bagaimana setiap fase SDLC dilakukan di perusahaan itu.

Faktor penting lainnya dalam mengkategorikan metodologi adalah pengurutan SDLC fase dan jumlah waktu dan upaya yang dicurahkan untuk masing-masing. Pada hari-hari awal komputasi, programmer tidak memahami perlunya metode siklus hidup yang formal dan terencana dengan baiki. Mereka cenderung beralih langsung dari tahap perencanaan yang sangat sederhana ke dalam konstruksi tahap implementasi dengan kata lain, dari yang sangat kabur, tidak baik permintaan sistem pemikiran ke dalam menulis kode. Ini adalah pendekatan yang sama yang terkadang Anda lakukan digunakan saat menulis program untuk kelas pemrograman. Dapat bekerja untuk program kecil itu hanya memerlukan satu programmer, tetapi jika persyaratannya rumit atau tidak jelas, Anda mungkin saja kehilangan aspek penting dari masalah dan harus memulai dari awal lagi, membuang sebagian program (dan waktu serta tenaga yang dihabiskan untuk menulisnya). Pendekatan ini juga membuat kerja tim sulit karena anggota memiliki sedikit gagasan tentang apa yang perlu diselesaikan dan bagaimana caranya bekerja sama untuk menghasilkan produk akhir. Pada bagian ini, dijelaskan tiga kelas yang berbeda dari metodologi pengembangan sistem: *structured design* (desain terstruktur), *rapid application development* (pengembangan aplikasi cepat), dan pengembangan *agile*.

a. Structured Design (Desain Terstruktur)

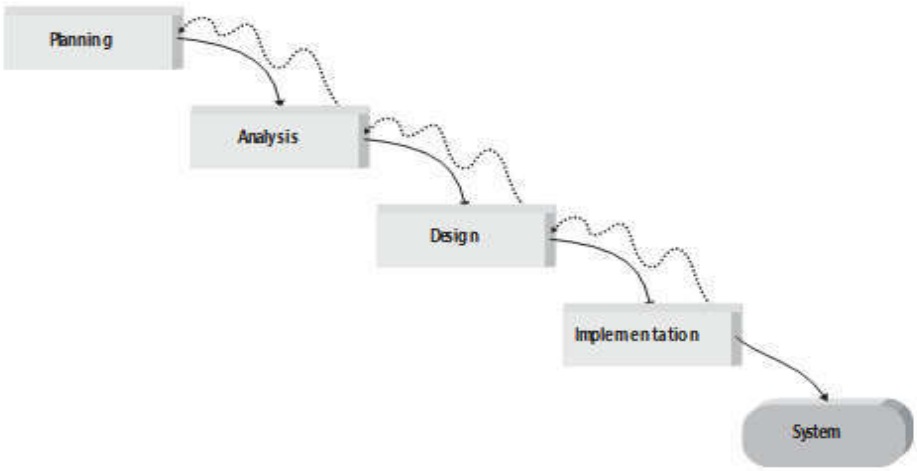
Kategori pertama dari metodologi pengembangan sistem disebut desain terstruktur. Metodologi ini menjadi dominan pada 1980-an, menggantikan ad hoc dan pendekatan yang tidak disiplin. Metodologi *structured design* mengadopsi pendekatan *step by step* dalam siklus pengembangan sistem, berpindah dari satu tahap ke tahap berikutnya secara logis dan beraturan. Berikut adalah dua metodologi yang termasuk dalam *structured design*:

1) Metodologi Pengembangan Waterfall

Metodologi desain terstruktur asli (masih digunakan sampai sekarang) adalah pengembangan *waterfall*. Dengan menggunakan pendekatan pengembangan berbasis *waterfall*, pengembangan sistem dilakukan secara berurutan dari satu tahap ke tahap berikutnya (lihat

Gambar 1). Kiriman utama dari setiap tahap biasanya sangat panjang (biasanya ratusan halaman) dan dikirimkan ke sponsor proyek untuk persetujuan saat proyek berpindah dari satu tahap ke tahap berikutnya.

Desain terstruktur juga memperkenalkan penggunaan pemodelan formal atau teknik diagram untuk menggambarkan proses bisnis dasar dan data yang mendukungnya. Tradisional desain terstruktur menggunakan satu set diagram untuk merepresentasikan proses dan satu set diagram untuk merepresentasikan proses dan satu set terpisah diagram untuk merepresentasikan data. Karena dua set diagram digunakan, analis sistem harus memutuskan himpunan mana yang akan dikembangkan pertama kali dan digunakan sebagai inti dari sistem: diagram model proses atau diagram model data.



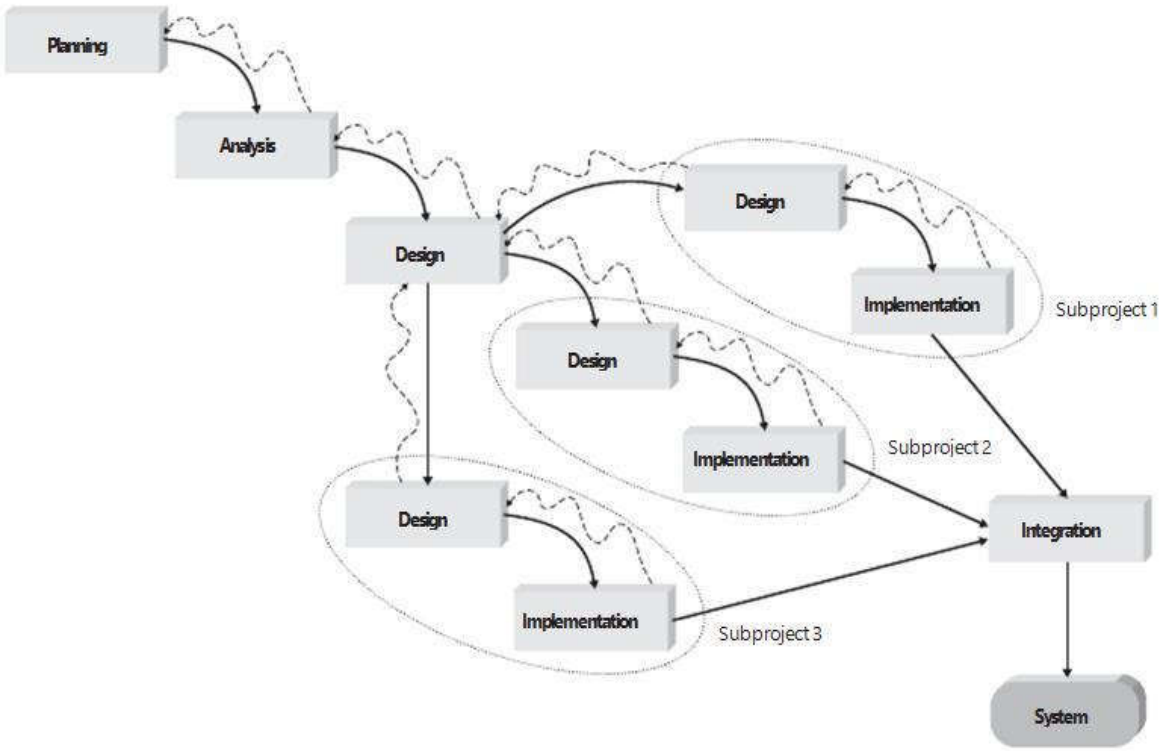
Gambar 1 Metodologi Pengembangan Waterfall

Dua keuntungan utama dari pendekatan waterfall adalah memenuhi semua persyaratan sistem jauh sebelum dimulainya pemrograman dan meminimalkan perubahan pada persyaratan sebagai kelanjutan sebuah proyek. Dua kelemahan utamanya adalah desain harus ditentukan secara lengkap sebelum dimulainya pemrograman dan butuh waktu yang lama antara penyelesaian proposal sistem dan pengiriman sistem ke pengguna (biasanya berbulan-bulan atau bertahun-tahun). Jika tim proyek melewatkan persyaratan penting, mahal pemrograman pasca implementasi mungkin diperlukan (bayangkan diri Anda mencoba merancang sebuah mobil di kertas;

seberapa besar kemungkinan Anda untuk mengingat lampu interior yang menyala saat pintu buka atau untuk menentukan jumlah katup yang tepat pada mesin?). Sebuah sistem juga bisa membutuhkan pengerjaan ulang yang signifikan karena lingkungan bisnis telah berubah dari waktu ketika fase analisis terjadi.

2) Metodologi Pengembangan Paralel

Pendekatan pengembangan paralel berusaha untuk mengatasi penundaan yang lama antara fase analisis dan pengiriman sistem. Alih-alih merancang dan mengimplementasikannya secara berurutan, ia merancang seluruh sistem dan kemudian membagi proyek menjadi serangkaian subproyek berbeda yang dapat dirancang dan dilaksanakan secara paralel. Setelah semua subproyek selesai, bagian-bagian yang terpisah diintegrasikan dan sistem disampaikan (lihat Gambar 2).



Gambar 2 Metodologi Pengembangan Paralel

Keuntungan utama dari metodologi ini adalah dapat mengurangi waktu pengiriman sistem; dengan demikian, kecil kemungkinan terjadinya perubahan yang menyebabkan pengerjaan ulang. Namun, terkadang subproyek tidak sepenuhnya independen; keputusan desain dibuat dalam satu subproyek dapat mempengaruhi yang lain, dan akhir proyek dapat membutuhkan signifikan upaya integrasi.

b. Rapid Application Development (RAD)

Kategori kedua dari metodologi adalah metodologi pengembangan aplikasi cepat / *Rapid Application Development* (RAD). Metodologi ini merupakan kelas terbaru dari metodologi pengembangan sistem yang muncul di tahun 1990-an. Metodologi ini berupaya untuk mengatasi dua kekurangan metodologi perancangan terstruktur dengan menyesuaikan tahapan SDLC, sehingga bagian-bagian tertentu dari sistem dapat berkembang dengan cepat dan menjangkau pengguna. Dengan cara ini, pengguna dapat lebih memahami sistem dan mengusulkan amandemen untuk membuat sistem lebih dekat dengan konten yang dibutuhkan. Namun, pendekatan berbasis RAD mungkin memiliki masalah kecil: mengelola ekspektasi pengguna. Karena penggunaan alat dan teknologi yang dapat meningkatkan kecepatan dan kualitas pengembangan sistem, ekspektasi pengguna dapat berubah secara dramatis. Sebagai pengguna yang lebih memahami teknologi informasi (TI), persyaratan sistem cenderung berkembang. Ini bukan masalah saat menggunakan metodologi yang menghabiskan banyak waktu untuk mendokumentasikan persyaratan secara menyeluruh.

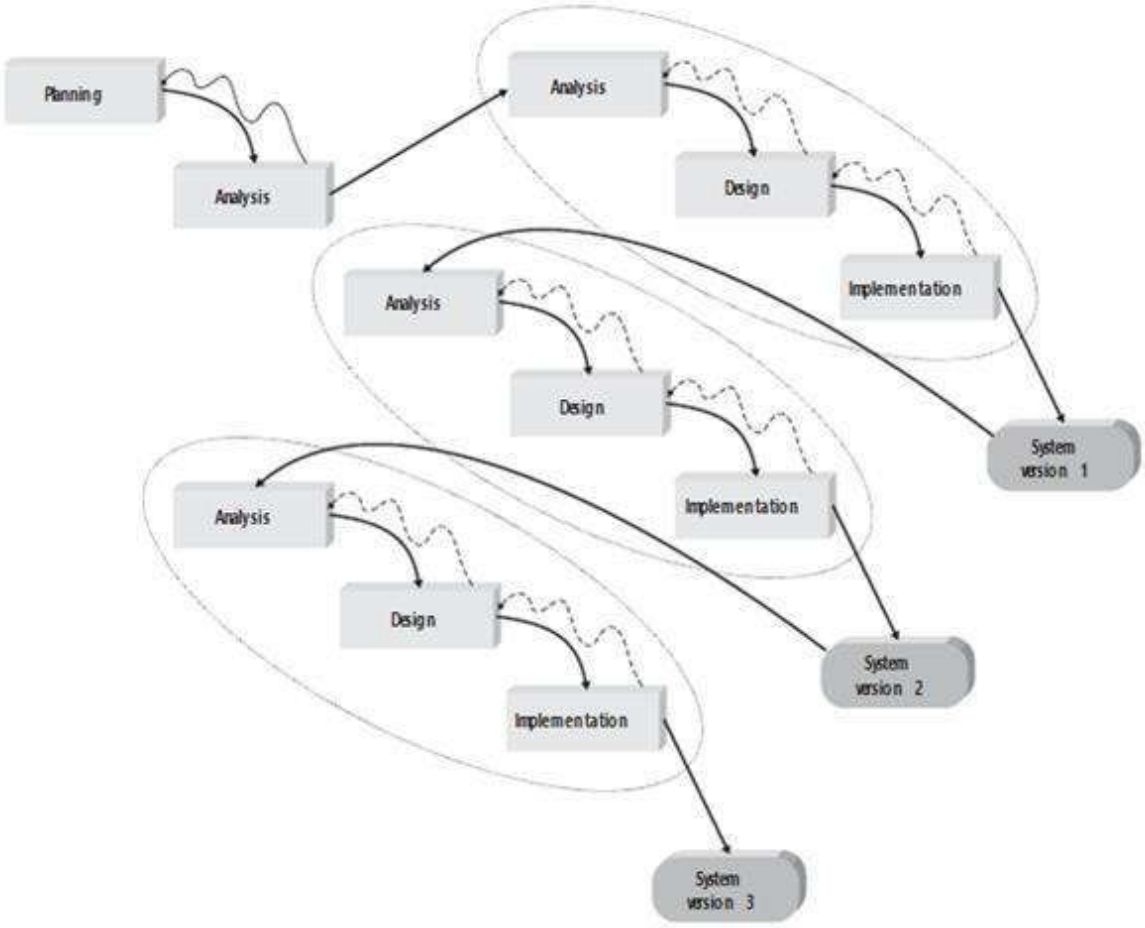
1) Metodologi Pengembangan Bertahap

Berdasarkan metode pengembangan inkremental, seluruh sistem diuraikan menjadi serangkaian versi pengembangan sekuensial. Fase analisis menentukan konsep sistem secara keseluruhan, dan kemudian tim proyek, pengguna, dan sponsor sistem mengklasifikasikan persyaratan ke dalam serangkaian versi. Persyaratan paling penting dan dasar telah dimasukkan ke dalam versi pertama sistem. Kemudian, tahap analisis akan dirancang dan diimplementasikan, tetapi hanya seperangkat persyaratan yang akan diidentifikasi untuk versi satu (lihat

Gambar 3).

Setelah versi satu diimplementasikan, pekerjaan dimulai pada versi dua. Berdasarkan persyaratan yang diidentifikasi sebelumnya serta ide dan masalah baru yang disebabkan oleh pengalaman pengguna versi satu, analisis tambahan dilakukan. Versi dua kemudian dirancang dan diimplementasikan, dan bekerja segera dimulai pada versi berikutnya. Proses ini berlanjut hingga sistem selesai atau tidak lagi digunakan.

Metodologi berbasis pengembangan bertahap memiliki keuntungan untuk segera mendapatkan sistem yang berguna ke tangan pengguna. Meskipun sistem tidak menjalankan semua fungsi yang dibutuhkan pengguna pada awalnya, sistem mulai memberikan nilai bisnis lebih cepat daripada jika sistem dikirimkan setelah selesai, seperti halnya dengan metodologi waterfall dan paralel. Demikian juga, karena pengguna mulai bekerja dengan sistem lebih cepat, mereka cenderung mengidentifikasi persyaratan tambahan penting lebih cepat daripada dengan situasi desain terstruktur.



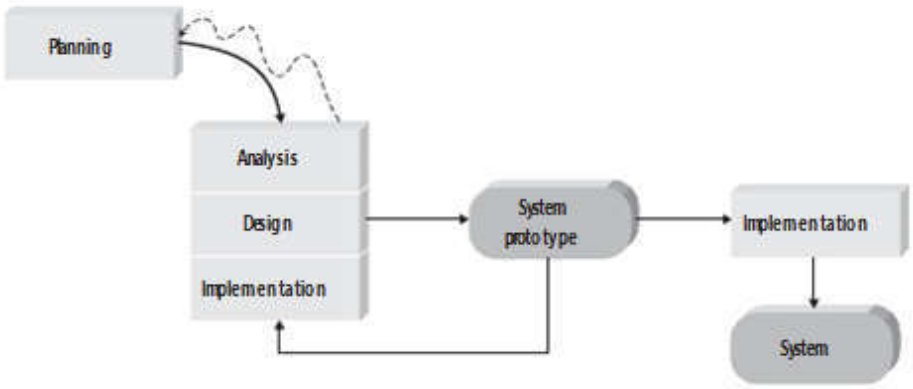
Gambar 3 Metodologi Pengembangan Bertahap

Kelemahan utama dari pengembangan bertahap adalah bahwa pengguna mulai bekerja dengan sistem yang sengaja tidak lengkap. Sangat penting untuk mengidentifikasi fitur yang paling penting dan berguna dan menyertakannya di versi pertama dan untuk mengelola ekspektasi pengguna selama prosesnya.

2) Metodologi Prototipe

Dalam metodologi prototipe tahap analisa, desain, dan implementasi dikerjakan secara bersamaan, dan dilakukan pengulangan pada ketiga tahap tersebut hingga didapatkan sistem yang sesuai dengan keinginan. Dengan menggunakan metode ini, dapat dibuat dasar analisa dan desain, dan segera mulai mengerjakan prototipe sistem, yang merupakan program cepat dan kotor yang hanya

menyediakan fungsionalitas minimal. Setelah prototipe (sekarang disebut "sistem") dipasang, perbaikan dilakukan hingga diterima sebagai sistem baru (lihat Gambar 4).



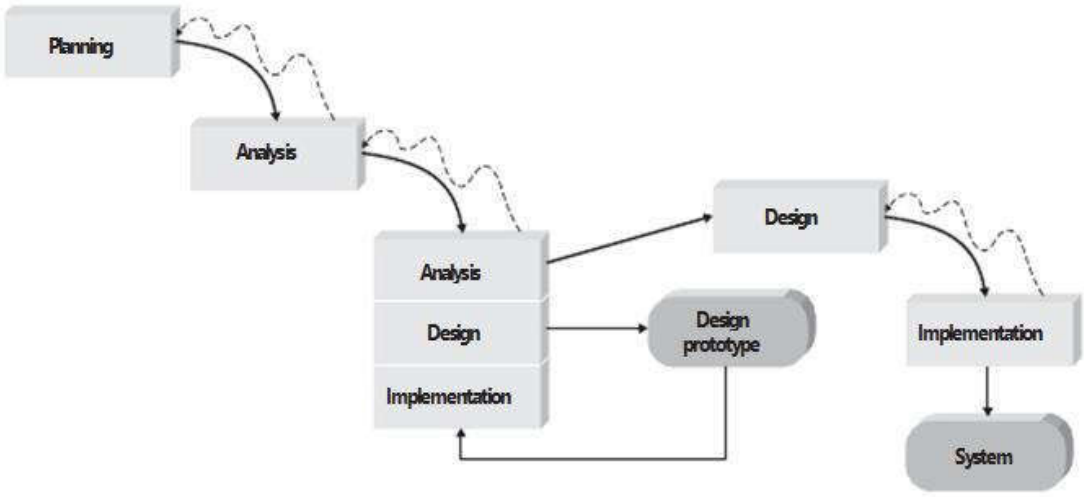
Gambar 4 Metodologi Prototipe

Keuntungan utama dari metodologi berbasis prototipe adalah bahwa metodologi ini sangat cepat menyediakan sistem yang dapat digunakan pengguna untuk berinteraksi, bahkan jika belum siap untuk digunakan organisasi secara luas pada awalnya. pembuatan prototipe meyakinkan pengguna bahwa tim proyek sedang mengerjakan sistem (tidak ada penundaan besar ketika pengguna melihat sedikit kemajuan), dan pembuatan prototipe membantu menyesuaikan kebutuhan nyata dengan lebih cepat.

Masalah utama dengan pembuatan prototipe adalah bahwa sistemnya yang bergerak cepat melepaskan tantangan upaya untuk melakukan analisis metodis yang cermat. Seringkali prototipe mengalami perubahan signifikan sehingga banyak keputusan desain awal menjadi salah.

3) Metodologi Prototipe Sekali Pakai

Metodologi berbasis prototipe sekali pakai mirip dengan metodologi berbasis prototipe yang mencakup pengembangan prototipe; namun, prototipe sekali pakai dilakukan pada titik yang berbeda di SDLC. Prototipe ini digunakan untuk tujuan yang sangat berbeda dari yang telah dibahas sebelumnya, dan memiliki tampilan yang sangat berbeda (lihat Gambar 5).



Gambar 5. Metologi Prototipe Sekali Pakai

Metodologi yang didasarkan pada prototipe sekali pakai memiliki tahap analisis yang relatif komprehensif, yang digunakan untuk mengumpulkan informasi dan mengembangkan ide untuk konsep sistem.

Namun, pengguna mungkin tidak sepenuhnya memahami banyak fitur yang mereka sarankan, dan mungkin ada masalah teknis yang sulit untuk diselesaikan. Masing-masing masalah ini diselidiki dengan menganalisis, merancang, dan membangun prototipe. Prototipe desain bukanlah sistem yang berfungsi. Ini adalah produk yang merupakan bagian dari sistem yang membutuhkan perbaikan tambahan dan hanya berisi detail yang cukup bagi pengguna untuk memahami masalah yang dihadapi. Misalnya, pengguna tidak sepenuhnya jelas tentang cara kerja sistem entri pesanan.

Sebuah sistem yang dikembangkan menggunakan metodologi jenis ini mengandalkan beberapa prototipe desain selama tahap analisis dan desain. Setiap prototipe digunakan untuk meminimalkan risiko yang terkait dengan sistem dengan memastikan bahwa masalah penting dipahami sebelum yang sebenarnya sistem dibangun. Setelah masalah terselesaikan, proyek beralih ke desain dan implementasi. Pada titik ini, prototipe desain tidak digunakan lagi, yang merupakan perbedaan penting antara metodologi lain dan metodologi pembuatan prototipe ini, di mana prototipe berkembang menjadi sistem akhir.

Metodologi berbasis prototipe sekali pakai menyeimbangkan manfaat fase analisis dan desain yang dipikirkan matang-matang dengan keuntungan menggunakan prototipe untuk memperbaiki masalah utama sebelum sistem dibangun. Diperlukan waktu lebih lama untuk menyampaikan sistem akhir dibandingkan dengan metodologi berbasis prototipe, tetapi jenis metodologi ini biasanya menghasilkan sistem yang lebih stabil dan andal.

c. Pengembangan Agile

Kategori ketiga dari metodologi pengembangan sistem masih muncul saat ini: pengembangan agile. Semua metodologi pengembangan agile didasarkan pada manifesto tangkas dan seperangkat dua belas prinsip. Penekanan dari manifesto adalah untuk memfokuskan pengembang pada kondisi kerja pengembang, perangkat lunak yang berfungsi, pelanggan, dan menangani persyaratan yang berubah alih-alih berfokus pada proses pengembangan sistem terperinci, alat, semua dokumentasi inklusif, kontrak hukum, dan rencana terperinci . Metodologi yang berpusat pada pemrograman ini memiliki sedikit aturan dan praktik, yang semuanya cukup mudah diikuti. Metodologi ini biasanya hanya didasarkan pada dua belas prinsip perangkat lunak tangkas. Prinsip-prinsip ini termasuk yang berikut ini:

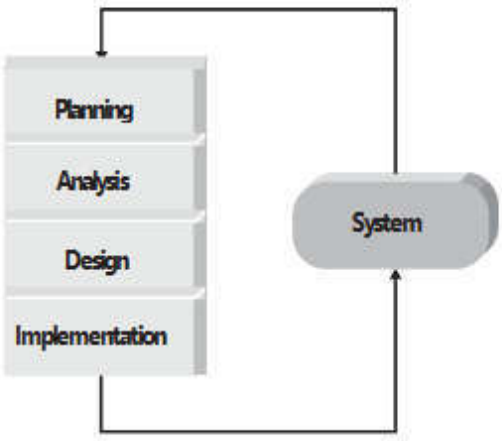
1) Perangkat lunak dikirimkan lebih awal dan terus menerus melalui proses pengembangan, untuk memuaskan pelanggan.
2) Persyaratan yang berubah diterapkan terlepas dari kapan hal itu terjadi dalam proses pengembangan.
3) Perangkat lunak yang berfungsi sering dikirimkan ke pelanggan.
4) Pelanggan dan pengembang bekerja sama untuk memecahkan masalah bisnis.
5) Individu yang termotivasi menciptakan solusi; memberi mereka alat dan lingkungan yang mereka butuhkan, dan percayai mereka untuk memberikannya.
6) Komunikasi tatap muka dalam tim pengembangan adalah metode pengumpulan persyaratan yang paling efisien dan efektif.
7) Ukuran utama kemajuan bekerja, menjalankan perangkat lunak.
8) *User* maupun pengembang harus bekerja dengan kecepatan yang berkelanjutan. Artinya, tingkat pekerjaan bisa dipertahankan tanpa batas

waktu tanpa ada pekerja yang mengalami kejenuhan.

9) Ketangkasan ditingkatkan melalui perhatian pada keunggulan teknis dan desain yang baik.
10) Kesederhanaan, penting untuk menghindari pekerjaan yang tidak perlu.
11) Tim mengembangkan arsitektur, persyaratan, dan desain terbaik.
12) Pengembang secara teratur merefleksikan bagaimana meningkatkan proses pengembangan mereka.

Berdasarkan prinsip-prinsip ini, metodologi agile berfokus pada penyederhanaan proses pengembangan sistem dengan meniadakan banyak overhead pemodelan dan dokumentasi serta waktu yang dibutuhkan untuk tugas-tugas tersebut. Sebaliknya, proyek tersebut menitikberatkan pada pengembangan aplikasi berulang yang sederhana.

Semua metodologi pengembangan agile mengikuti siklus sederhana melalui fase tradisional dari proses pengembangan sistem (lihat Gambar 6). Hampir semua metodologi agile digunakan dalam hubungannya dengan teknologi berorientasi objek.



Gambar 6. Pengembangan Agile

Namun, metodologi agile memang mendapat kritik. Salah satu kritik utama berkaitan dengan lingkungan bisnis, di mana sebagian besar pengembangan sistem informasi aktual di luar batas, dialihdayakan, dan / atau disubkontrakkan. Mengingat metodologi pengembangan agile yang membutuhkan lokasi bersama tim pengembangan, ini tampaknya merupakan asumsi yang sangat tidak realistis. Kritik besar kedua adalah

bahwa jika pengembangan tangkas tidak dikelola dengan hati-hati, dan menurut definisi tidak, proses pengembangan dapat beralih ke pendekatan prototipe yang pada dasarnya menjadi lingkungan "programmer menjadi liar" di mana programmer mencoba meretas solusi bersama. Kritik utama ketiga, yang didasarkan pada kurangnya dokumentasi aktual yang dibuat selama pengembangan perangkat lunak, menimbulkan masalah terkait kemampuan audit dari sistem yang dibuat. Tanpa dokumentasi yang memadai, baik sistem maupun proses pengembangan sistem tidak dapat dijamin. Kritik besar keempat didasarkan pada apakah pendekatan tangkas dapat menghasilkan sistem misi kritis yang besar.

Bahkan dengan kritik ini, mengingat potensi pendekatan tangkas untuk mengatasi backlog aplikasi dan memberikan solusi tepat waktu untuk banyak masalah bisnis, pendekatan tangkas harus dipertimbangkan dalam beberapa keadaan. Selain itu, banyak teknik yang didorong dengan memperhatikan tujuan yang mendasari manifesto tangkas dan sekumpulan dua belas prinsip tangkas sangat berguna dalam pengembangan sistem berorientasi objek. Dua dari contoh metodologi pengembangan agile yang lebih populer adalah program ekstrim (XP) dan Scrum.

1) Pemrograman Ekstrim/ Extreme Programming (XP)

Pemrograman Ekstrim memiliki empat nilai inti antara lain: komunikasi, kesederhanaan, *feedback*, dan keberanian. Keempat nilai ini merupakan dasar bagi pengembang XP untuk membangun sistem apa pun. Pertama, pengembang harus terus memberikan *feedback* kepada pengguna. Kedua, XP mengharuskan pengembang mengikuti prinsip KISS. Ketiga, pengembang harus dapat melakukan perubahan tambahan untuk memperluas sistem. Tidak hanya harus melakukan perubahan, mereka juga harus menerima perubahan. Keempat, developer harus memiliki pola pikir yang mengutamakan kualitas. XP juga mendukung anggota tim untuk mengembangkan keterampilan mereka.

Tiga prinsip utama keberhasilan penggunaan sistem XP adalah pengujian berkelanjutan, pengkodean sederhana yang dilakukan oleh sepasang pengembang, dan interaksi erat dengan pengguna akhir untuk membangun sistem dengan sangat cepat. Pengujian dan coding yang efisien adalah inti dari XP. Kode diuji setiap hari dan ditempatkan ke

dalam lingkungan pengujian integratif. Jika ada bug, kode tersebut dicadangkan hingga benar-benar bebas dari kesalahan.

Proyek XP dimulai dengan cerita pengguna yang menjelaskan apa yang perlu dilakukan sistem. Kemudian, programmer membuat kode dalam modul kecil dan sederhana dan menguji untuk memenuhi kebutuhan tersebut. Pengguna diharuskan tersedia untuk menjernihkan pertanyaan dan masalah yang muncul. Standar sangat penting untuk meminimalkan kebingungan, jadi tim XP menggunakan sekumpulan nama, deskripsi, dan praktik pengkodean yang umum. Proyek XP memberikan hasil lebih cepat daripada pendekatan RAD, dan mereka jarang macet dalam mengumpulkan persyaratan untuk sistem.

Penganut XP mengklaim banyak kekuatan yang terkait dengan pengembangan perangkat lunak menggunakan XP. Programmer bekerja erat dengan semua pemangku kepentingan, dan komunikasi di antara semua pemangku kepentingan ditingkatkan. Pengujian berkelanjutan dari sistem yang berkembang didorong. Sistem ini dikembangkan secara evolusioner dan bertahap, yang memungkinkan persyaratan berkembang seiring dengan pemahaman para pemangku kepentingan terhadap potensi yang dimiliki teknologi dalam memberikan solusi untuk masalah mereka. Estimasi didorong oleh tugas dan dilakukan oleh programmer yang akan mengimplementasikan solusi untuk tugas yang sedang dipertimbangkan. Karena semua pemrograman dilakukan berpasangan, tanggung jawab bersama untuk setiap komponen perangkat lunak berkembang di antara programmer. Akhirnya, kualitas produk akhir meningkat selama setiap iterasi.

Untuk proyek kecil dengan tim yang sangat termotivasi, kohesif, stabil, dan berpengalaman, XP seharusnya berfungsi dengan baik. Namun, jika proyeknya tidak kecil atau timnya tidak senang, 9 keberhasilan upaya pengembangan XP diragukan. Hal ini cenderung meragukan seluruh gagasan membawa kontraktor luar ke lingkungan tim yang ada menggunakan XP. Kemungkinan orang luar bercengkerama dengan orang dalam mungkin terlalu optimis. XP membutuhkan disiplin yang tinggi, jika tidak, proyek akan menjadi tidak fokus dan kacau. XP disarankan hanya untuk sekelompok kecil pengembang (tidak lebih dari sepuluh pengembang) dan tidak

disarankan untuk aplikasi penting yang besar. Karena kurangnya dokumentasi analisis dan desain, hanya ada dokumentasi kode yang terkait dengan XP, jadi memelihara sistem besar yang dibangun dengan XP mungkin mustahil. Dan karena sistem informasi bisnis mission-critical cenderung ada untuk waktu yang lama, kegunaan XP sebagai metodologi pengembangan sistem informasi bisnis diragukan. Akhirnya, metodologi membutuhkan banyak masukan pengguna di tempat, sesuatu yang tidak dapat dilakukan oleh banyak unit bisnis. Namun, beberapa teknik yang terkait dengan XP berguna dalam pengembangan sistem berorientasi objek. Misalnya, cerita pengguna, pemrograman pasangan, dan pengujian berkelanjutan adalah alat yang sangat berharga dari mana pengembangan sistem berorientasi objek bisa mendapatkan keuntungan.

2) Scrum

Scrum adalah istilah yang sangat dikenal oleh para penggemar rugby. Dalam rugby, scrum digunakan untuk memulai kembali permainan. Singkatnya, pencipta metode *scrum* percaya bahwa tidak peduli seberapa banyak Anda merencanakan, segera setelah perangkat lunak mulai dikembangkan, kekacauan muncul dan rencana tersebut keluar dari jendela. Hal terbaik yang dapat Anda lakukan adalah bereaksi terhadapnya. di mana bola rugby pepatah menyemprot keluar. Anda kemudian melakukan sprint dengan bola hingga *scrum* berikutnya. Dalam kasus metodologi Scrum, sprint berlangsung selama tiga puluh hari kerja. Di akhir sprint, sebuah sistem dikirimkan ke pelanggan.

Di permukaan, Scrum adalah metode pengembangan sistem yang paling membingungkan. Untuk mengontrol kekacauan bawaan tertentu, pengembangan Scrum berfokus pada beberapa praktik utama. Tim ini mengatur diri sendiri dan mengarahkan diri sendiri. Tidak seperti metode lain, tim Scrum tidak memiliki ketua tim yang ditunjuk. Sebaliknya, tim mengatur secara simbiosis dan menetapkan tujuan sendiri untuk setiap sprint (iterasi). Setelah sprint dimulai, tim Scrum tidak mempertimbangkan persyaratan tambahan apa pun. Persyaratan baru apa pun yang ditemukan ditempatkan di simpanan persyaratan yang masih perlu ditangani. Di awal setiap hari kerja, pertemuan Scrum berlangsung. Di akhir setiap sprint, tim mendemonstrasikan perangkat

lunak tersebut kepada klien. Berdasarkan hasil sprint, rencana baru dimulai untuk sprint berikutnya.

Rapat scrum adalah salah satu aspek paling menarik dari proses pengembangan Scrum. Anggota tim menghadiri rapat, tetapi siapa pun dapat hadir. Namun, dengan sedikit pengecualian, hanya anggota tim yang boleh berbicara. Satu pengecualian yang menonjol adalah manajemen memberikan umpan balik tentang relevansi bisnis dari pekerjaan yang dilakukan oleh tim tertentu. Dalam pertemuan ini, semua anggota tim berdiri membentuk lingkaran dan melaporkan apa yang telah mereka capai pada hari sebelumnya, menyampaikan apa yang akan dilakukan hari ini, dan menjelaskan apa saja kemajuan yang diblokir hari sebelumnya. Untuk mengaktifkan kemajuan berkelanjutan, setiap blok yang diidentifikasi ditangani dalam waktu satu jam. Dari sudut pandang Scrum, lebih baik membuat keputusan "buruk" tentang suatu blok pada saat ini dalam pengembangan daripada tidak membuat keputusan. Karena pertemuan berlangsung setiap hari, keputusan yang buruk dapat dengan mudah dibatalkan. Larman menyarankan agar setiap anggota tim harus melaporkan persyaratan tambahan apa pun yang telah ditemukan selama sprint dan apa pun yang dipelajari anggota tim yang dapat berguna untuk diketahui anggota tim lainnya.

Salah satu kritik utama dari Scrum, seperti halnya semua metodologi tangkas, adalah bahwa masih dipertanyakan apakah Scrum dapat berkembang untuk mengembangkan sistem misi-kritis yang sangat besar. Ukuran tim Scrum biasanya tidak lebih dari tujuh anggota. Satu-satunya prinsip pengorganisasian yang dikemukakan oleh pengikut Scrum untuk menjawab kritik ini adalah dengan mengatur scrum dari scrum. Setiap tim bertemu setiap hari, dan setelah rapat tim berlangsung, seorang perwakilan (bukan pemimpin) dari setiap tim menghadiri rapat scrum-of-scrums. Ini berlanjut sampai kemajuan seluruh sistem telah ditentukan. Bergantung pada jumlah tim yang terlibat, pendekatan untuk mengelola proyek besar ini diragukan. Namun, seperti di XP dan pendekatan pengembangan tangkas lainnya, banyak ide dan teknik yang terkait dengan pengembangan Scrum berguna dalam pengembangan sistem berorientasi objek, seperti fokus pertemuan Scrum, pendekatan evolusioner dan inkremental untuk

mengidentifikasi persyaratan, dan inkremental. dan pendekatan iteratif untuk pengembangan sistem.

C. SOAL LATIHAN/ TUGAS

1. Carilah definisi pengertian sistem dari para ahli selain yang dijelaskan di atas!
2. Definisikan pengertian sistem menurut pendapatmu!
3. Apa yang kalian ketahui tentang klasifikasi sistem? Jelaskan!
4. Apa yang kalian ketahui tentang metodologi pengembangan sistem? Jelaskan!
5. Jelaskan secara singkat alur metodologi pengembangan sistem!

D. REFERENSI

Alan Dennis, Barbara Haley Wixom, David Tegarden (2015). *Systems Analysis & Design: an object-oriented approach with UML.* 5th edition. New Jersey: John Wiley & Sons, Inc.

Charles S. Wasson (2006). *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: John Wiley & Sons, Inc.

Dr. Jawahar. *Overview of System Analysis & Design*. Diakses dari: http://www.ddegjust.ac.in/studymaterial/pgdca/ms-04.pdf

GLOSARIUM

Application atau Program Aplikasi adalah perangkat lunak yang dikembangkan oleh perusahaan komputer agar dapat menyelesaikan tugas tertentu, seperti Microsoft-Word dan Microsoft-Excel.

Data merupakan kumpulan angka dan karakter yang tidak mempunyai arti. Data tersebut dapat diolah untuk menghasilkan sebuah informasi.

Ethernet adalah Standar perangkat keras LAN (Local Area Network) untuk kabel dan spesifikasi transmisi.

Fitur dari kata *feature*, adalah fungsi atau kemampuan khusus yang ada pada suatu alat.

Hacker atau peretas adalah pakar komputer terampil yang menggunakan pengetahuan teknis mereka untuk memecahkan masalah. Meskipun "peretas" dapat merujuk pada pemrogram komputer yang terampil, istilah ini telah dikaitkan dengan "peretas keamanan" dalam budaya populer, dan "peretas keamanan" mengandalkan pengetahuan teknis mereka untuk menggunakan kesalahan atau celah untuk menyerang sistem komputer.

Java merupakan bahasa pemrograman yang digunakan untuk membuat konten aktif di halaman web, yang juga dapat dijalankan di komputer mana pun.

Komputer adalah peralatan elektronik yang dapat mengolah berbagai data secara sistematis dan cermat. Misalnya data digital, suara dan gambar.

Soft Real-Time merupakan sistem yang bila tidak berhasil diselesaikan dalam waktu satu hari maka tidak menyebabkan kegagalan sistem.

Wizard adalah sebuah fungsi dalam perangkat lunak, termasuk perkantoran, dapat mendesain dokumen dengan mudah dan langkah demi langkah.