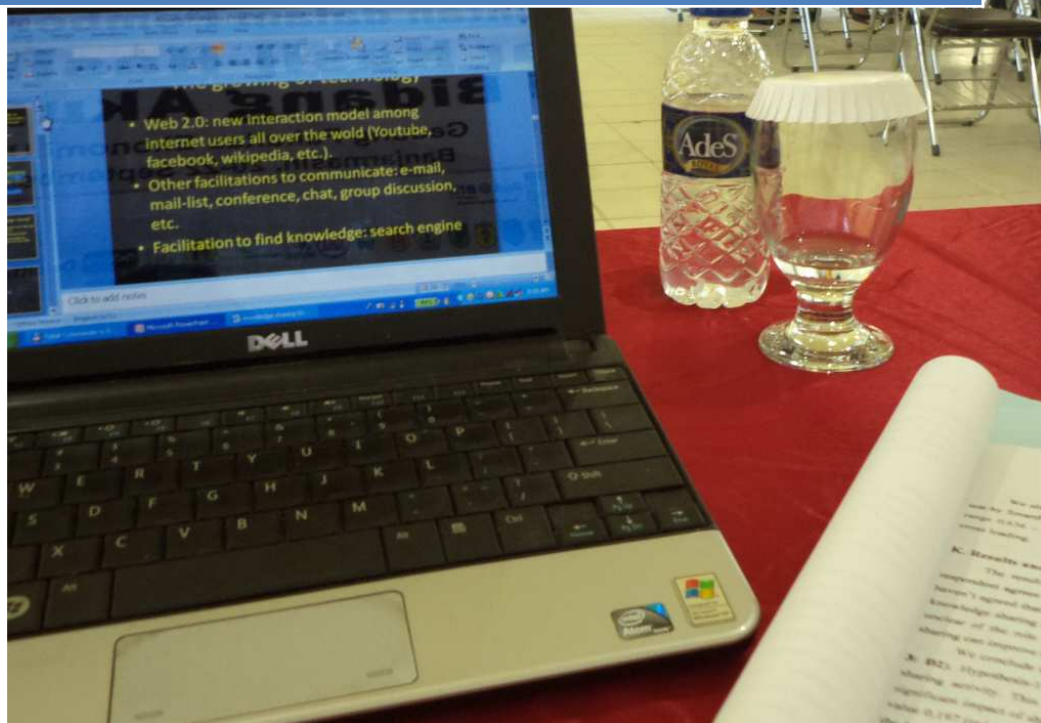


2013

EDISI 1

ANALISIS DAN PERANCANGAN SISTEM Perspektif Kompetensi Akuntansi



Agustinus MUJILAN, S.E., M.Sc.
Akuntansi – Univ. Widya Mandala
Madiun

ANALISIS DAN PERANCANGAN SISTEM

Perspektif kompetensi akuntansi

Agustinus MUJILAN, S.E., M.Sc.

Edisi 1

ISBN: --

v, 59 hal, 182 mm x 257 mm (B5)

Lay out : Mujilan

Disain cover : Mujilan

Edisi 1: 2013

Prodi Akuntansi

Universitas Widya Mandala Madiun

© 2013 Hak cipta ada pada penulis

Copyright

Buku ini dapat digunakan atau dibagikan secara bebas asalkan dalam ranah untuk penyebaran ilmu dan bersifat disebarikan gratis. Komersialisasi tanpa seijin dari penulis merupakan pelanggaran terhadap hak cipta.

Daftar Isi

Daftar Isi	iii
KATA PENGANTAR.....	v
BAB 1: PENDAHULUAN ANALISIS DAN PERANCANGAN SISTEM.....	1
A. Aspek Penting dalam Kesuksesan Pengembangan Sistem.....	1
B. Filosofi Analisis dan Perancangan Sistem	2
Referensi:	3
BAB 2: PENGEMBANGAN SISTEM (<i>System Development</i>).....	6
A. Pendahuluan	6
B. Metode Waterfall.....	6
C. Efek Kepuasan Pengguna dalam Pengembangan	7
Referensi:	8
BAB 3: METODE PENGEMBANGAN SISTEM TEKNOLOGI INFORMASI.....	9
A. Metode Pengembangan Sistem	9
B. Metode SDLC.....	9
1. Analisis Sistem	10
2. Perancangan Sistem	10
3. Implementasi sistem	11
4. Operasi dan Perawatan.....	11
C. Aplikasi: SSADM.....	11
Referensi:	15
BAB 4: Memahami Latar Belakang Organisasi	17
A. Tujuan Organisasi	17
B. <i>Core Bisnis</i>	17
C. Strategi Organisasi	18
D. Struktur Organisasi.....	19
E. Budaya Organisasi	20
Referensi:	22
BAB 5: KUALITAS LAYANAN	24
BAB 6: BAHASA PEMODELAN SISTEM	27
A. Mengapa perlu bahasa pemodelan?.....	27
B. DIAGRAM SISTEM.....	27
C. UNIFIED MODELING LANGUAGE (UML)	30
Referensi:	32
BAB 7: PEMODELAN DATABASE.....	34
A. Entity-Relationship (E-R) Diagram.....	34
1. Informasi terkait <i>Entities and Relationship</i> (Level 1)	34
2. Struktur Informasi (<i>Information Structure</i>) (Level 2)	35
B. Simbol Entity Relationship Diagram.....	36
3. Contoh bentuk aplikasi diagram dalam E-R	39
C. ERD Menggunakan Aplikasi MsVisio	41
D. PROGRAM APLIKASI	43

Referensi:	44
[BAB 8] <i>UML: Use Case Diagram</i>	45
[BAB 9] <i>UML: ACTIVITY DIAGRAMS</i>	48
[BAB 10] <i>UML: CLASS DIAGRAMS</i>	52
A. Kelas (Class)	52
B. Diagram Kelas (Class Diagrams)	53
Referensi:	54
[BAB 11] PROTOTYPING & PENGEMBANGAN APLIKASI CEPAT	55
A. PROTOTYPING	55
B. Jenis-Jenis Prototipe	56
C. Kelemahan dan Kelebihan	57
D. PAC (PENGEMBANGAN APLIKASI CEPAT)	57
PROJECT 1: Membangun sistem penerimaan pendapatan	59
A. Keterangan Umum	59
B. Administrasi User	59
C. Interaksi Pengguna Sistem	59
D. Pengoperasian Penerimaan Kas	59

KATA PENGANTAR

Setiap elemen dalam organisasi saat ini semakin dituntut untuk mengenal teknologi informasi. Sedapat mungkin setiap bagian dalam organisasi termasuk manajerial tidak hanya berperan sebagai pengguna yang pasif namun juga dapat terlibat di dalam pengembangan sistemnya. Analisis dan perancangan sistem mengasah kemampuan untuk mengenali proses pengembangan suatu sistem sehingga dapat mengenali tentang proses pengembangannya, demikian juga dapat menginspirasi daya pikir dan inovasi untuk mengusulkan atau memperbaiki sistem yang diterapkan.

Buku ini memberi pengenalan dan aplikatif sederhana sebagai dasar pemahaman dalam pengembangan sistem. Pemahaman atas dasar penganalisisan dan perancangan sistem memberi manfaat ketika harus berkomunikasi dengan pengembang sistem. Pembaca yang ingin lebih dalam memahami konsep dalam pengembangan sistem dapat mengembangkan wawasannya dengan mencari literatur lain yang lebih spesifik dalam analisis dan perancangan sistem.

Bagian penting dalam buku ini adalah dasar-dasar pemahaman kebutuhan analisis sistem; konsep dan aplikasi analisis sistem; dasar-dasar perancangan (*design*) sistem; dan aplikasi dasar perancangan sistem. Bagian-bagian tersebut merupakan konsep-konsep penting dalam kegiatan pengembangan sistem. Demikian juga, ditambahkan konsep-konsep penting yang dapat mempengaruhi keberhasilan pengembangan sistem sebagai bahan pengembangan wawasan.

Semoga buku ini dapat dimanfaatkan sebagaimana mestinya. Masukan dan kritisi diterima dan penulis sangat berterima kasih. Silahkan kontak di agus_muji@yahoo.com. Terima kasih

Madiun, 2 Juni 2013

Penulis

BAB 1: PENDAHULUAN ANALISIS DAN PERANCANGAN SISTEM

A. Aspek Penting dalam Kesuksesan Pengembangan Sistem

Analisis sistem akan sangat dekat kaitannya dengan perancangan sistem dan selanjutnya adalah penulisan program untuk aplikasi sistem informasi. Analisis sistem akan mendahului kegiatan sebelum dilakukannya perancangan sistem. Dan sebetulnya analisis masih akan terus berlangsung dan diperlukan dalam tahap perancangan maupun dalam proses penulisan program. Analisis akan terus dilakukan sampai diperkirakan sistem informasi yang dihasilkan layak untuk diimplementasikan.

Analisis ini diperlukan dalam rangka pengembangan sistem. Pengembangan sistem itu sendiri dapat terdiri dari pengembangan sistem baru ataupun memperbaiki dan merevisi sistem yang telah ada. Sistem dapat dikatakan sebagai baru apabila bentuk atau model sistem yang akan dibangun merupakan sesuatu yang belum pernah diterapkan sebelumnya. Sistem baru ini dapat terjadi misalnya karena adanya teknologi baru, sistem operasi baru, penggunaan jaringan baru, atau bahkan pada organisasi yang belum pernah menerapkan sistem informasi. Sementara memperbaiki atau merevisi sistem dapat dilakukan ketika sistem yang sudah berjalan membutuhkan suatu perbaikan atau penambahan menu yang akan digunakan untuk menangani transaksi.

Permasalahan dalam pengembangan sistem adalah bagaimana agar sistem yang akan dirancang dapat berhasil diterapkan atau kesuksesan sistem informasi (*information system success*). Kesuksesan sistem informasi dapat dipengaruhi oleh dua hal:

- Aspek teknis
- Aspek Keperilakuan

Aspek teknis adalah bagaimana membuat sistem yang dibangun berkualitas secara teknis. Kualitas secara teknis adalah bagaimana membuat sistem minim kesalahan (*errors*), informasi yang dihasilkan berkualitas (tepat bagi pengguna, mudah dimengerti, tepat waktu, efisien, tersedia ketika digunakan, dan aman). Namun demikian, kesuksesan secara teknis tersebut belum tentu membuat sistem tersebut sukses diterapkan pada organisasi. Kesalahan-kesalahan yang mungkin timbul dalam aspek teknis antara lain kesalahan sintak, kesalahan logik, dan kesalahan informasi. Namun demikian, banyak kasus bahwa sistem informasi telah baik secara teknis namun gagal dalam penerapannya di organisasi, maka kemudian perlu dilihat aspek keperilakuan (Jogiyanto, 2008).

Aspek perilaku dari para pengguna di organisasi tersebut menjadi faktor penting pula. Manusia sebagai aktor yang menjalankan sistem menentukan pula apakah sistem yang diterapkan akan berhasil atau tidak. Maka, perlu pula diperhitungkan adanya aspek keperilakuan dalam pengembangan sistem.

Aspek keperilakuan ini sering dilihat dari sisi penerimaan pengguna atas sistem informasi. Aspek keperilakuan ini telah dipelajari misalnya dengan *Technology Acceptance Model (TAM)* (Davis, 1989) yang membahas dua hal penting yaitu kemudahan dan kegunaan yang kemudian mempengaruhi niat penggunaan sistem informasi. Contoh lain adalah model DeLone & McLane (DM) yang membahas masalah kualitas sistem dan kepuasan pengguna.

Berbagai hal terkait dengan bagaimana mengusahakan kesuksesan sistem informasi semestinya menjadi perhatian dan sensitifitas bagi para analis dan pihak-pihak yang terlibat dalam pengembangan sistem.

B. Filosofi Analisis dan Perancangan Sistem

Whitten et al. (1986) mengungkapkan filosofi dalam analisis dan perancangan sistem untuk memahami karakteristik dasar dalam kegiatan analisis dan perancangan sistem. Filosofi ini juga mengindikasikan lingkup perhatian dalam analisis sistem.

- Analisis dan perancangan sistem bukan merupakan kegiatan mekanis.
- Penulis program mungkin sangat membutuhkan ketrampilan, namun pada analisis dan perancangan sistem juga sangat membutuhkan daya seni (*art*).
- Analisis dan perancangan sistem adalah bidang praktis.

Kita dapat mencoba memahami apa yang disampaikan di atas. Pertama bahwa analisis dan perancangan sistem merupakan kegiatan yang tidak dapat ditentukan langkahnya secara mekanis. Di dalamnya memerlukan berbagai pertimbangan yang sesuai dengan permasalahan yang dihadapi. Namun demikian, bukan berarti kegiatan analisis dan perancangan sistem tidak dapat dipelajari dan ditentukan prosesnya. Proses yang umum dilakukan dapat menjadi acuan untuk mendukung kesuksesan analisis sistem. Namun demikian bukan berarti semua langkah tersebut harus diikuti tanpa pertimbangan. Dalam hal ini, juga memerlukan ide yang lebih baik bagi setiap analis.

Kedua, istilah daya seni berarti seorang analis dan perancang sistem juga sangat membutuhkan kemampuan pengembangan daya pikir dan sentuhan seni dalam menjalankan kegiatannya. Sentuhan seni karena mereka harus berhadapan pula dengan orang-orang yang terlibat, kemudian harus menggambarkan pula ide-ide yang ada di benak analis baik atas hasil temuan ataupun perencanaannya. Sehingga kemampuan dan keterampilan teknis diperlukan namun belum cukup. Kemampuan teknis ini misalnya adalah kemampuan membuat bagan-bagan alir perancangan, teknis membuat pelaporan analisis, teknis menghitung biaya dan manfaat, dan sebagainya.

Ketiga, analisis dan perancangan dipandang sebagai bidang praktis. Dapat kita sadari bahwa ilmu yang kita pelajari dapat berupa teori ataupun praktis. Bidang praktis akan mendekatkan pemecahan masalah pada aplikasi atau hal yang terkait dengan keputusan dalam perusahaan atau organisasi. Secara lebih mudah berarti mendekatkan pada aplikasi untuk organisasi. Sudut

pandang ini beralasan, sebab hasil dari analisis dan perancangan sistem tersebut akan diterapkan pada organisasi tertentu. Kecocokan dan kesesuaian dengan organisasi yang akan menjadi objek analisis merupakan hal yang sangat penting dalam melakukan analisis dan perancangan sistem.

Referensi:

- Davis, Fred D. 1989. Perceive Usefulness, Perceive Ease of Use, and User Accpetance of Information Technology. *MIS Quarterly*. September 1989.
- DeLone, William H; Ephraim R. McLean. 1992. Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research*. March 1992. 3: 1
- Jogiyanto HM, Prof. Dr., MBA, Akt. 2009. *Sistem Teknologi Informasi*. Edisi III. Andi Offset Yogyakarta.
- Jogiyanto, HM. 2008. *Sistem Informasi Keperilakuan*. Andi Offset. Yogyakarta.
- Kumeiga, Andrew; and Ben Van Vliet. 2008. A Software Development Methodology for Research and Prototyping in Financial Market. *Unpublished*. Cornell University.
- Royce, Winston W. 1970. Managing the Development of Large Software Systmes. *Proceedings IEEE Wescon*. August 1970. P 1-9.
- Whitten, Jeffrey L.; Lonnie D Bentley; Thomas I.M. Ho, PdD), “*System Analysis & Design Methods*”, First Edition, Times Mirror/Mosby College Publising, St. Louis, 1986.

The Social Network

Film Tahun 2010.

Film ini memenangkan berbagai penghargaan diantaranya Academy Award, Golden Globe Awards, Critics' Choice Awards, Writers Guild of America, American Cinema Editors, dsb. Film menceritakan sebuah proses bagaimana mengkreasi adanya situs jejaring sosial. Tokoh utama adalah Mark Zuckerberg, seorang mahasiswa S1 pemrograman komputer Harvard.

Pada musim gugur 2003, Mark mulai mengerjakan ide barunya dalam konteks pemrograman blog. Ia meretas ke berbagai facebook orang-orang ketika mereka menjadi murid di Harvard dan membuat web site Facemash. Situs ini ditujukan agar pengguna dapat meranking mahasiswa perempuan yang cantik berdasar foto yang didapat dari berbagai sistem universitas. Kunjungan ke situs ini sangat banyak hingga melumpuhkan jaringan kampus. Oleh dewan universitas tindakan Mark ini dianggap bersalah. Ia mendapat hukuman percobaan selama 6 bulan.

FaceMash menjadi populer dan menarik perhatian Cameron dan Tyler Winklevoss beserta rekan bisnisnya Divya Narendra. Dalam pertemuan di sebuah pesta, Mark menerima tawaran sebagai pemrogram dalam situs kencan yang mereka sebut "Harvard Connection" yang akan digunakan secara eksklusif untuk alumni Harvard.

Melihat perkembangan gagasannya, Mark mendekati temannya, Eduardo Saverin, dan mengatakan tentang idenya yang dia sebut "Thefacebook", sebuah website jejaring sosial online eksklusif untuk mahasiswa Universitas Harvard. Situs ini diperkirakan akan meniru kesuksesan FaceMash. Namun dalam Thefacebook akan dibuat akun yang pengguna harus mendaftar terlebih dahulu untuk menghindari permasalahan etika seperti yang terjadi pada website sebelumnya. Seseorang yang memiliki akun dapat mengatur sendiri apa yang akan dia tampilkan, informasi apa yang dapat diakses publik, gambar apa yang akan dia pasang. Ia dapat mengundang temannya untuk ikut serta bergabung ke dalam web dengan membuat akun. Eduardo setuju membantu \$1.000 untuk memulai situs.

Setelah diluncurkan pada 4 Februari 2004, situs ini menjadi populer dikalangan para siswa. Namun demikian, si kembar Winklevoss dan Narendra merasa bahwa idenya telah dicuri oleh Mark. Mereka mengajukan tuntutan atas pelanggaran hak kekayaan intelektual. Sidang tuntutan terus berlangsung, namun Mark juga tetap aktif mengembangkan sistemnya.

Ide-ide terus berdatangan, ia menemukan ide dari diskusi dengan teman. Pengembangan fitur dan isi facebook ia dapatkan dari obrolan, diskusi, serta melihat

dan menyadari permasalahan. Ketika terjadi permasalahan dalam pengkodean dan rumus algoritma, ia pun bertanya pada teman yang pakar di bidangnya.

Mark berencana untuk mengembangkan jaringan situsnya ke Universitas Yale, Universitas Columbia. Eduardo menambah usulan untuk perluasan jaringan ke Stanford. Ia beranggapan bahwa Stanford adalah area penting untuk ekspansi. Stanford berlokasi di California, sebelah kanan kota Palo Alto dan San Jose. Area ini dikenal dengan istilah “Silicon Valley” karena sejumlah perusahaan besar komputer dan teknologi berada di sana. Dalam benak Eduardo berpikir bahwa dengan memperluas ke Universitas Stanford dapat menarik kapitalis di Silicon Valley dengan menginvestasikan pada usahanya.

Pada suatu kesempatan Mark bertemu dengan Sean Parker, pendiri Napster. Eduardo menjadi skeptik setelah mengingat sejarah permasalahan personal dan profesi dari Sean. Namun demikian, Sean menyampaikan pendapatnya pada Mark, bahwa Facebook dapat memberikan kekaguman dengan cepat. Dalam bagian pendapatnya, Sean memberi saran agar kata “The” ditanggalkan dari “Thefacebook”.

Berdasarkan saran Sean, Mark memindahkan kantornya ke Palo Alto sementara Eduardo ke New York untuk mencari dukungan iklan. Sean menyarankan agar Mark mempertahankan kepemilikannya atas Facebook agar ia tidak kehilangan kendali atas potensi bisnis yang menguntungkan. Setelah Sean menjanjikan akan mengembangkan Facebook ke dua benua, Mark mengundang Sean untuk tinggal di rumahnya yang digunakan pula sebagai perkantoran.

**

Pada tahun 2010, Mark Elliot Zuckerberg mendapatkan penghargaan dari majalah *Time* dan dinobatkan sebagai *person of the Year*. Penghargaan ini diberikan karena telah mengkoneksikan lebih dari setengah milyar orang dan memetakan hubungan sosial diantara mereka, untuk penciptaan sistem baru dalam bertukar informasi dan untuk perubahan dalam cara hidup.

Pertanyaan:

1. Apa ide utama dalam pengembangan sistem jejaring sosial Facebook?
2. Darimana ide perancangan muncul?
3. Tuntutan hukum apa yang ditujukan dalam kasus di atas?
4. Apa potensi pendapatan dari adanya pembangunan situs jejaring sosial?
5. Bagaimana cara pemilik situs merancang peningkatan keluasaan jaringan?

Referensi:

<http://www.thesocialnetwork-movie.com/>

<http://www.imdb.com/title/tt1285016/>

http://en.wikipedia.org/wiki/The_Social_Network

Lev Grossman. 2010. Person of the Year 2010 Mark Zuckerberg. *Time*, December 15, 2010.

BAB 2: PENGEMBANGAN SISTEM (*System Development*)

A. Pendahuluan

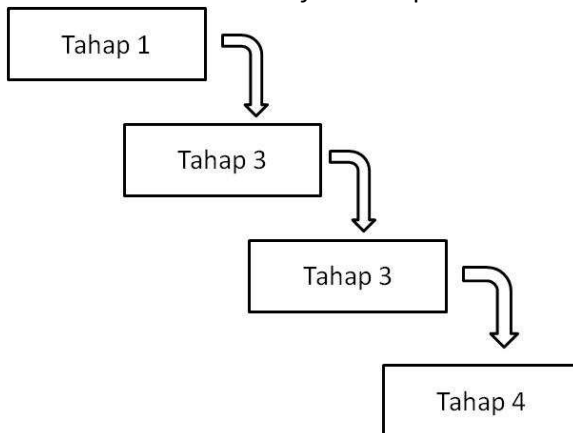
Sistem informasi (SI) semestinya dapat mendukung kebutuhan manajemen dalam menyediakan informasi yang akurat, sehingga sistem informasi perlu dirajut sesuai kebutuhan informasi bagi manajemen juga mendukung perubahan yang berkelanjutan Habiburrochman & Hidayatin (2012). Konsekuensinya bahwa hal penting yang perlu dicatat dalam proyek pengembangan sistem informasi (*information system development project*) adalah perlunya sistem informasi dibangun berdasarkan kebutuhan dan keinginan dari para penggunanya (*users*). Pengembangan SI perlu mempertimbangkan berbagai faktor yang dapat mendorong kesuksesan pengembangannya, mengingat SI adalah hal yang mahal dalam membangunnya, sehingga perlu hati-hati dalam pembangunannya agar tidak justru menimbulkan kerugian yang besar. Faktor yang perlu diperhatikan antara lain analisis sistem, pengguna, sponsor, dan pelanggan.

Analisis sistem merupakan hal yang baik dalam mempersiapkan penerapan pengembangan sistem. Tersedia berbagai metode dalam pengembangan sistem. Namun perlu diingat bahwa metode-metode tersebut harus disesuaikan dengan kondisi dan kebutuhan pembangunan sistem dalam organisasi tertentu. Satu metode tidak dapat dipaksakan kebenarannya pada semua organisasi. Analisis dan perancangan sistem membutuhkan waktu yang cukup panjang dalam pengembangan sistem, dalam kondisi perusahaan atau organisasi harus segera menerapkan sistem, maka langkah-langkah dalam metode tertentu akan menjadi tidak efisien untuk dipaksakan penggunaannya. Maka pemilihan metode analisis yang tepat dalam pengembangan sistem informasi dalam organisasi tertentu juga perlu dipertimbangkan efesiensi dan efektivitasnya.

Perlu diingat kembali bahwa analisis dan perancangan sistem selain membutuhkan kemampuan teknis juga membutuhkan kemampuan daya seni. Dalam hal ini pula sangat dibutuhkan komunikasi pada pengguna sistem. Analisis dan perancangan sistem bukan ditujukan untuk penulisan program aplikasi, namun lebih banyak digunakan untuk membuat konsep secara lebih luas tentang pengaplikasian sistem informasi pada suatu organisasi. Ia mempersiapkan penulisan program dan pengimplementasian program. Berbagai faktor perlu dipertimbangkan termasuk aspek teknis dan non teknis. Tujuannya adalah kesuksesan penerapan sistem pada suatu organisasi. Maka, konsep analisis dan disain sistem ini sangat dekat dengan bidang praktis.

B. Metode Waterfall

Dapat dikatakan bahwa metode ini menggunakan tahap demi tahap dalam proses pengembangan sistem. Misalnya setelah mengetahui kebutuhan sistem dan kebutuhan piranti lunak, maka dapat dilakukan analisis sistem, setelah selesai analisis sistem dapat dilakukan disain sistem, dan seterusnya sampai pada implementasi (Royce, 1970; Kumeiga & Vliet, 2008). Secara sederhana metode *waterfall* ini dapat dilukiskan sebagai berikut.



Gambar: konsep metode *waterfall*

C. Efek Kepuasan Pengguna dalam Pengembangan

Kepuasan pengguna dalam hal ini adalah ungkapan rasa senang atau peningkatan peningkatan koneksi para pengguna selama proses pembangunan sistem. Kepuasan pengguna dapat mendukung kesuksesan dalam implementasi sistem yang dibangun. Habiburrochman & Hidayatin (2012) menyatakan bahwa kepuasan pengguna salah satunya dapat diupayakan melalui partisipasi pengguna (*user participation*) dalam proses pengembangan sistem. Partisipasi pengguna merupakan perilaku, pekerjaan, aktivitas yang dapat memberi manfaat dalam proses pengembangan sistem dari para calon pengguna langsung informasi yang akan dihasilkan.

Dalam banyak hal kepuasan merupakan hal penting. Misalnya kepuasan pelanggan dapat mendukung pembelian ulang produk dan jasa. Kepuasan pengguna dalam media online dapat mendukung kunjungan kembali dan menggunakan fasilitas atau layanan yang diberikan. Dengan demikian, kepuasan pengguna sistem ketika dalam proses pembangunan sistem akan memicu rasa suka dan aman dalam operasionalnya kelak. Perasaan suka dan aman ini dapat terjadi karena pengguna merasa kebutuhan serta bentuk informasi yang ia inginkan telah diperhatikan.

Referensi:

- Davis, Fred D. 1989. Perceive Usefulness, Perceive Ease of Use, and User Accpetance of Information Technology. *MIS Quarterly*. September 1989.
- DeLone, William H; Ephraim R. McLean. 1992. Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research*. March 1992. 3: 1
- Habiburrochman and Hidayatin, D.A. (2012), The Influence of User Participation to User Satisfaction in Developing Information System, Proceeding: Airlangga Accounting International Conference & Doctoral Colloqium 2012, Nusa Dua Bali
- Jogiyanto HM, Prof. Dr., MBA, Akt. 2009. Sistem Teknologi Informasi. Edisi III. Andi Offset Yogyakarta.
- Jogiyanto, HM. 2008. Sistem Informasi Keperilakuan. Andi Offset. Yogyakarta.
- Kumeiga, Andrew; and Ben Van Vliet. 2008. A Software Development Methodology for Research and Prototyping in Financial Market. Unpublished. Cornell University.
- Royce, Winston W. 1970. Managing the Development of Large Software Systmes. *Proceedings IEEE Wescon*. August 1970. P 1-9.
- Whitten, Jeffrey L.; Lonnie D Bentley; Thomas I.M. Ho, PdD), "System Analysis & Design Methods", First Edition, Times Mirror/Mosby College Publising, St. Louis, 1986.

BAB 3: METODE PENGEMBANGAN SISTEM TEKNOLOGI INFORMASI

A. Metode Pengembangan Sistem

Pengembangan sistem teknologi informasi (STI) dapat dilakukan dengan beberapa cara. Secara garis besar metode pengembangan sistem dapat dikategorikan menjadi dua, yaitu 1) metode konvensional dan 2) metode alternatif (Hartono, 2009). Disebut metode konvensional karena telah ada sebelum metode alternatif. Pengembangan STI konvensional dapat menggunakan metode “**siklus hidup pengembangan sistem**” atau “**system development life cycle (SDLC)**”. Sementara metode alternatif misalnya dapat menggunakan: paket (*package*), pembuatan prototip (*prototyping*), pengembangan oleh pemakai akhir (*end user development* atau *end user computing*) dan *outsourcing*.

B. Metode SDLC

Metode siklus hidup pengembangan sistem (*system development life cycle / SDLC*) memiliki beberapa tahapan. Tahap utama dapat dikategorikan menjadi (Hartono, 2009):

1. Analisis sistem
2. Perancangan sistem
3. Implementasi sistem
4. Operasi dan perawatan sistem

Disebut siklus karena pengembangan sistem selanjutnya dapat dimulai lagi dari awal tahap sampai dengan tahap terakhir. Tahapan-tahapan tersebut dapat meliputi pula sub-sub kegiatan, yaitu:

1. Analisis sistem
 - a. Studi pendahuluan
 - b. Studi kelayakan
 - c. Mengidentifikasi permasalahan dan kebutuhan pemakai
 - d. Memahami sistem yang ada
 - e. Menganalisis hasil penelitian.
2. Perancangan sistem
 - a. Perancangan awal
 - b. Perancangan rinci
3. Implementasi sistem
4. Operasi dan perawatan sistem



Gambar 3. 1: Siklus hidup pengembangan sistem (SDLC)

1. Analisis Sistem

Studi Pendahuluan: merupakan kegiatan awal dari analisis sistem. Studi ini meliputi: jenis, ruang lingkup dan pemahaman awal dari proyek pengembangan sistem. Hasilnya adalah: pemahaman awal dan perkiraan biaya

Studi Kelayakan (*feasibility study*): terdiri dari lima macam kelayakan yang disebut TELOS yang berupa kelayakan Teknologi, Ekonomi, Legal, Operasi, dan Sosial. Layak secara teknologi jika teknologi yang dibutuhkan tersedia atau dapat diperoleh. Layak secara ekonomi jika manfaat yang diperoleh lebih besar daripada biaya yang dikeluarkan serta dana tersedia. Layak secara legal jika tidak melanggar peraturan dan hukum. Layak secara operasi jika sistem dapat dioperasikan dan dijalankan. Layak secara sosial jika tidak mempunyai pengaruh negatif terhadap lingkungan sosial.

Mengidentifikasi Permasalahan dan Kebutuhan Informasi Pemakai: mengidentifikasi masalah dilakukan dengan mengidentifikasi penyebab masalahnya yang merupakan sumber permasalahan yang harus diperbaiki. Kemudian dapat dilakukan penelitian terkait dengan data dan sistem yang telah ada.

Menganalisis hasil penelitian: menganalisis kelemahan dan kebutuhan informasi pemakai. Menganalisis kelemahan dimaksudkan untuk menemukan penyebab tidak berfungsinya sistem. Menganalisis kebutuhan informasi dimaksudkan agar sistem dapat menghasilkan informasi yang relevan.

2. Perancangan Sistem

Perancangan sistem mempunyai dua tujuan utama: 1) memberikan gambaran umum kebutuhan informasi kepada pemakai, dan 2) memberi

gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya.

3. Implementasi sistem

Tahap ini merupakan tahap meletakkan sistem agar siap digunakan. Pada tahap ini dapat dilakukan: rancangan implementasi, memilih dan melatih personil, mempersiapkan tempat dan lokasi sistem, mengetes sistem, serta melakukan konversi sistem.

4. Operasi dan Perawatan

Kegiatan perawatan perlu dilakukan antara lain karena: 1) keperluan memperbaiki kesalahan 2) adanya perubahan karena permintaan pengguna sistem, 2) adanya perubahan lingkungan luar, 3) keperluan peningkatan sistem.

C. Aplikasi: SSADM

SSADM merupakan salah satu model aplikasi dalam pengembangan sistem. Biro Komputer dan Telekomunikasi Pusat (*Central Computer and Telecommunications Agency*) atau yang sekarang dikenal dengan Kantor Bisnis Pemerintah (*Office of Government Commerce*) di Pemerintah Inggris, sejak tahun 1980 mulai membangun cara melakukan analisis dan disain sistem yang saat ini dikenal dengan sebutan Metode Analisis dan Disain Terstruktur (*Structured Systems Analysis and Design Method / SSADM*). SSADM menggunakan metode *waterfall* dalam melakukan analisis dan disain sistem informasi.

Tiga hal penting yang digunakan dalam teknik SSADM adalah:

1. Pemodelan Data Logik (*Logical Data Modeling*).
Merupakan proses identifikasi, pemodelan dan dokumentasi data yang diperlukan atas sistem yang didisain. Data dibedakan menjadi *entities* (sesuatu menyangkut kebutuhan bisnis untuk mencatat informasi) dan *relationships* (hubungan antar entitas).
2. Pemodelan Arus Data (*Data Flow Modeling*)
Merupakan proses identifikasi, pemodelan dan dokumentasi bagaimana data berpindah seputar sistem informasi. Pemodelan arus data menguji proses (aktivitas yang merubah data dari satu bentuk ke bentuk lain), penyimpanan data (area penyimpan data), entitas eksternal (apa data yang dikirim ke sistem lain dan apa data yang diterima), dan aliran data (jalur yang dapat dilalui aliran data)
3. Pemodelan Perilaku Entitas (*Entity Behaviour Modeling*)
Merupakan proses identifikasi, pemodelan dan dokumentasi atas kejadian yang berpengaruh pada setiap entitas dan rangkaian (*sequence*) dari kejadian yang muncul.

Kemudian SSADM mengaplikasikan tahapan dalam melakukan analisis, sebagai berikut:

Tahap 0 – Studi Kelayakan (*Feasibility Study*)

Digunakan untuk menentukan apakah proyek layak atau tidak, terdapat beberapa bentuk investigasi atas tujuan dan implikasi dari proyek. Untuk proyek skala kecil, mungkin hal ini tidak diperlukan karena lingkup proyek telah mudah untuk dipahami. Untuk proyek yang lebih besar, studi kelayakan mungkin dapat dilakukan meskipun secara informal karena tidak cukup dengan waktu formal atau karena proyek harus sudah dilaksanakan.

Produk dari tahapan ini adalah dokumen studi kelayakan formal. Ketika melakukan studi kelayakan, terdapat empat area yang perlu dipertimbangkan.

- *Technical* – apakah proyek mungkin secara teknik?
- *Financial* – dapatkah organisasi menanggung proyek tersebut?
- *Organizational* – akankah sistem baru sesuai dengan praktik yang telah ada?
- *Ethical* – apakah dampak dari sistem baru terhadap penerimaan sosial?

Tahap 1 – Investigasi lingkungan saat ini (*Investigation of the current environment*)

Tahap ini merupakan tahapan yang paling penting dalam tahapan SSADM. Pengembang SSADM memahami bahwa dalam tugas dan tujuan sistem baru mungkin tidak akan secara radikal berbeda dari sistem yang lama, data dasar mungkin hanya akan berubah sedikit. Dalam banyak kasus terdapat beberapa format sistem yang berlaku saat ini yang berbentuk manusia atau kertas. Melalui kombinasi antara wawancara dengan pekerja, kuesioner, observasi dan melihat dokumentasi, analis mencoba memahami secara utuh sistem ketika proyek akan dilakukan. Tujuan dari tahap ini:

- Analis mempelajari terminologi bisnis, apa yang pengguna (*users*) lakukan dan bagaimana mereka melakukannya.
- Sistem yang lama menyediakan lingkup kebutuhan untuk keperluan sistem baru.
- Kegagalan, kesalahan, dan area yang tidak efisien digarisbawahi dan ditambahkan koreksi kebutuhan.
- Model data dapat disusun
- Pengguna yang akan dilibatkan dan mempelajari teknik serta model analisis
- Batasan (*boundaries*) sistem dapat didefinisikan.

Produk dalam tahap ini adalah:

- Katalog pengguna yang menjabarkan semua pengguna sistem dan bagaimana mereka berinteraksi dengan sistem.
- Katalog kebutuhan secara rinci atas semua kebutuhan sistem yang baru

- Penjabaran Layanan saat ini (*current services description*)
- Struktur data logik saat ini (ERD / *Entity relationship diagram*)
- Context diagram (DFD)
- Leveled set of DFDs untuk sistem logik saat ini
- Direktori data lengkap termasuk hubungan antar penyimpanan data (*stores data*) dan entitas.

Tahap 2 – Pilihan sistem bisnis (*Business system options*)

Setelah menginvestigasi sistem berjalan, analis sistem harus memutuskan disain sistem baru secara keseluruhan. Untuk melakukannya, ia, menggunakan hasil dari tahap sebelumnya, mengembangkan rangkaian pilihan sistem bisnis. Analis mungkin juga akan melakukan *brainstorming* sehingga banyak ide yang mungkin didapat. Ide-ide tersebut kemudian dikumpulkan dalam dua atau tiga pilihan yang akan disampaikan pada pengguna, pilihan yang dapat dipertimbangkan:

- Tingkat otomatisasi
- Batasan antara sistem dan pengguna
- Distribusi sistem, contoh apakah tersentral pada satu kantor atau dipecah ke dalam beberapa bagian?
- Biaya dan manfaat
- Dampak/pengaruh dari sistem yang baru.

Jika mungkin, pilihan/alternatif tersebut didokumentasikan dengan struktur data logik dan diagram aliran data level 1. Diagram level 1 semacam gambaran umum aliran data. Kemudian antara pengguna dan analis akan bersama-sama menentukan satu pilihan. Hasil dari tahap ini adalah pilihan bisnis yang dilakukan bersama berdasarkan hasil dari analisis tahap 1.

Tahap 3 – Spesifikasi Kebutuhan (*requirements specification*)

Mungkin hal ini adalah tahap yang kompleks dalam SSADM. Menggunakan kebutuhan yang diidentifikasi dalam tahap 1 dan bekerja dalam rerangka pilihan alternatif bisnis, analis harus mengembangkan spesifikasi logik secara lengkap atas apa yang akan dilaksanakan dalam sistem baru. Spesifikasi ini harus bebas dari kesalahan, ambiguitas dan ketakonsistenan. Dengan logikal, dimaksudkan bahwa spesifikasi bukan untuk menyampaikan apa yang akan diimplementasikan sistem namun lebih pada penjabaran apa yang akan dilakukan sistem.

Untuk membuat spesifikasi logik, analis membangun model kebutuhan logik dengan *data-flow diagrams* (DFDs) dan *the entity relationship diagrams* (ERDs). Hal ini digunakan untuk membuat definisi setiap fungsi dan apa yang akan dibutuhkan pengguna sistem, *entity life-histories* (ELHs) dan *effect correspondence diagrams*.

Produk dari tahap ini adalah

- Katalog kebutuhan yang terupdate
- Spesifikasi proses:
 - Peran pengguna / matrik fungsi
 - Definisi fungsi
 - Model data logik kebutuhan
 - Sejarah hidup entitas
 - Diagram pengaruh koresponden

Tahap 4 – Pilihan sistem teknis (*Technical system options*)

Tahap ini merupakan tahap pertama menuju implementasi sistem secara fisik. Sepertihalnya pilihan sistem bisnis, dalam tahap ini sejumlah pilihan untuk implementasi sistem baru dibuat. Kemudian akan disampaikan kepada pengguna dengan dua atau tiga pilihan yang selanjutnya pilihan akhir akan ditentukan dan disintesiskan.

Pertimbangan yang dilakukan adalah:

- Arsitektur *hardware*
- *Software* yang akan digunakan
- Biaya implementasi
- Staf yang dibutuhkan
- Keterbatasan fisik seperti *space* antar sistem
- Distribusi termasuk jaringan yang diperlukan
- Format antarmuka (*interface*)

Semua aspek ini harus dikonfirmasi dengan berbagai keterbatasan organisasi seperti kemampuan dana dan standarisasi *hardware* dan *software*. Sehingga hasil dari tahap ini adalah keputusan pemilihan sistem teknik.

Tahap 5 – Disain Logik (*logical design*)

Pada tahap sebelumnya menspesifikasikan rincian implementasi, sementara pada tahap ini menghasilkan implementasi yang independen dan konsen pada kebutuhan manusia akan antarmuka (*interface*). Disain logik menspesifikasikan metode-metode utama interaksi dalam bentuk struktur menu (*menu structures*) dan struktur perintah (*command structures*).

Salah satu area aktivitas adalah mendefinikan dialog pengguna. Ini merupakan antarmuka utama dimana pengguna akan berinteraksi dengan sistem. Aktivitas lain adalah konsen pada analisis efek kejadian dalam pembaruan sistem dan kebutuhan untuk membuat permintaan data pada sistem. Diagram penggunaan, deskripsi fungsi dan efek koresponden yang dibuat dalam tahap 3 ditentukan bagaimana cara mengupdate dan membaca data secara konsisten dan aman.

Produk dalam tahap ini adalah:

- Katalog data
- Struktur kebutuhan data logik
- Model proses logik
- Titik berat dan tikungan peristiwa (*stress & Bending moment*).

Tahap 6 – Disain Fisik (*Physical design*)

Merupakan tahap akhir setelah semua spesifikasi logik sistem dikonversi ke dalam deskripsi sistem dalam bentuk *hardware* dan *software* nyata. Ini merupakan tahap yang sangat teknis. Struktur data logik dikonversi ke dalam arsitektur fisik dalam bentuk struktur database. Struktur sesungguhnya atas fungsi dan bagaimana cara implementasi dispesifikasikan. Produk tahap ini adalah disain fisik secara lengkap yang dapat menjelaskan pembuat perangkat lunak dan bagaimana membangun sistem dalam spesifikasi rinci atas perangkat keras dan perangkat lunak dan standar yang mencukupi.

Enam langkah dalam analisis dan perancangan sistem dengan SSADM merupakan salah satu contoh acuan dalam proses pengembangan sistem. Dalam praktiknya, di setiap perusahaan belum tentu menerapkan semua langkah tersebut, hal ini akan tergantung pada kondisi dan kemampuan perusahaan serta sistem yang akan dikembangkan sehingga perlu mempertimbangkan efek efektifitas dan efisiensinya. Namun demikian, acuan ini akan menjadi sangat berharga bagi para pengembang sistem.

Contoh metode lain dalam pengembangan sistem dipaparkan oleh Jogiyanto (2009). Pengembangan dapat dilakukan dengan metode **siklus hidup pengembangan sistem (*system development life cycle / SDLC*)**, dan **metode-metode alternatif (*alternative methods*)**. Metode alternatif yang tersedia antara lain paket (*package*), pembuatan prototip (*prototyping*), pengembangan oleh pemakai akhir (*end user development* atau *end user computing*), dan *outsourcing*. Siklus hidup pengembangan sistem (SDLC) terdiri dari proses *analisis sistem, perancangan sistem, implementasi sistem, operasi dan perawatan sistem*.

Referensi:

Jogiyanto HM, Prof. Dr., MBA, Akt. 2009. Sistem Teknologi Informasi. Edisi III. Andi Offset Yogyakarta.

Referensi Internet

Structured Systems Analysis and Design Method (SSADM).

http://en.wikipedia.org/wiki/Structured_Systems_Analysis_and_Design_Method. Diakses 4 Agustus 2011.

BAB 4: Memahami Latar Belakang Organisasi

A. Tujuan Organisasi

Pada umumnya ketika didirikan, suatu organisasi atau perusahaan diharapkan untuk tetap dapat bertahan dan *survive*. Untuk dapat bertahan banyak faktor yang perlu diperhatikan antara lain keunggulan kompetitif. Keunggulan kompetitif dapat dicapai misalnya dengan perbaikan kualitas atau peningkatan kinerja. Atau di sisi lain dengan memperkuat daya inovasi.

Banyak pula yang menyampaikan bahwa untuk mencapai peningkatan dalam organisasi perlu memperhatikan masukan dari pelanggan maupun mengusahakan memberi kepuasan pada para pelanggan. Serta masih banyak faktor terkait dengan kegiatan manajemen yang perlu diusahakan untuk mendukung kemajuan organisasi.

Banyak hal tersebut membutuhkan sensitifitas dan kemampuan organisasi untuk meramu dan menggunakan sumber daya yang ada dengan efisien dan efektif. Sistem informasi dapat dipandang dan ditempatkan fungsinya sebagai alat bantu manajemen, operasional, membangun hubungan dengan stakeholder. Bahkan pada tingkat yang lebih tinggi sistem informasi digunakan sebagai senjata strategis (*strategic weapon*) untuk mencapai keunggulan kompetitif dan memenangkan persaingan (Jogiyanto, 2005).

B. Core Bisnis

Core bisnis atau bisnis utama atau usaha pokok. Salah satu kunci sukses bagi pengusaha, perusahaan, atau organisasi adalah jangan melupakan bisnis utamanya. Memahami core bisnis¹ atau bisnis utama menjadi penting untuk melakukan ide pengembangan.

Misalnya, dalam suatu organisasi pendidikan dapat menyatakan bahwa bisnis utamanya adalah melayani masyarakat dengan cara menyediakan pengetahuan dan pendidikan yang bermutu. Jika organisasi tersebut memahami bisnis utamanya ini, maka segala upaya dan pengembangan akan ditujukan untuk mewujudkan dan meningkatkan peran tersebut. Misalnya kita ambil kata kuncinya yaitu: melayani, pengetahuan, pendidikan, dan mutu. Jika ditujukan untuk melayani maka segala bentuk pelayanan harus ditata dengan baik. Jika yang disampaikan adalah pengetahuan maka organisasi harus berupaya untuk mendapatkan pengetahuan sebanyak banyaknya dan menyebar luaskan pengetahuan tersebut dengan sebaik baiknya. Karena terdapat kata pendidikan, maka segala upaya dan kegiatan harus disertakan pula konteks mendidik. Demikian juga dengan mutu, maka apa saja yang dilakukan diusahakan mempunyai mutu yang baik.

¹ Ada dapat membaca contoh ulasan yang membahas core bisnis:
<http://news.fajar.co.id/read/93874/127/index.php> ; http://www.thecocacolacompany.com/ourcompany/wn20061110_bsr.html ;

Sekarang kita kaitkan dengan analisis dan perancangan sistem. Seorang perancang yang tidak memperhatikan bisnis utama dari organisasi atau perusahaan yang sistemnya dia rancang, mungkin saja kurang sensitif dengan hal-hal yang sebenarnya mempunyai kaitan atau benang merah dengan bisnis utama organisasi tersebut. Misalnya, perancangan website untuk organisasi yang terkait dengan pendidikan ditampilkan dengan bahasa gaul atau tidak menggunakan bahasa Indonesia yang tepat. Ketika hal itu terjadi, maka unsur mendidiknya berkurang. Kaitannya dengan pengetahuan, seorang yang tidak memahami unsur penyebaran pengetahuan dalam dunia pendidikan mungkin tidak merancang adanya ruang untuk penyebaran informasi atau pengetahuan yang bermanfaat bagi masyarakat umum.

Dengan memahami bisnis utama baik secara menyeluruh ataupun per kata kunci, maka seorang analis atau perancang akan menjabarkan dalam sub-sub bagian dalam rencana pengembangan sistemnya. Bahkan jika perlu seorang analis dapat mengusulkan dan memberi masukan kepada pihak manajemen tentang pandangan jauh ke depan atas perancangan sistem dan pengembangannya di masa mendatang.

C. Strategi Organisasi

Memahami strategi dapat memberikan pemahaman dalam menentukan sistem informasi yang cocok atas strategi yang diterapkan organisasi. Setiap organisasi mempunyai strategi untuk mencapai keunggulan kompetitif. Jika analis dari luar organisasi, mungkin perlu mengkomunikasikan strategi yang diterapkan organisasi. Namun tidak menutup kemungkinan pula bahwa analis intern perlu mengkomunikasikan strategi organisasi kepada pihak manajemen. Keselarasan antara strategi dan disain sistem informasi sangat diperlukan. Karena, sistem informasi dapat difungsikan untuk mendukung strategi yang ada.

Porter (1996) menyatakan bahwa keefektifan operasional bukanlah strategi. Suatu organisasi harus fleksibel dalam merespon perubahan kompetisi dan pasar. Keefektifan operasional dinyatakan sebagai hal yang perlu namun belum cukup (*necessary but not sufficient*). Keefektifan operasional dan strategi keduanya merupakan hal penting untuk keunggulan kompetitif dan dapat menjadi tujuan utama organisasi.

Suatu organisasi dapat mengungguli pesaingnya jika dapat memberikan sesuatu yang berbeda. Ia harus memberikan nilai yang lebih tinggi pada pelanggannya atau menciptakan nilai sebanding pada harga lebih rendah, atau keduanya. Perbedaan kos dan harga disebabkan karena banyak aktivitas dalam mencipta, memproduksi, menjual, mengirim produk atau jasa, telepon pelanggan, assembling produk akhir, dan pelatihan karyawan.

Strategi yang dikemukakan Porter tersebut yang sangat dikenal saat ini adalah strategi diferensiasi dan harga rendah. Diferensiasi memberikan kekhususan atau keunikan pada produk atau jasa. Sementara harga rendah mengusahakan penekanan harga dalam aktivitas organisasi pada biaya-biaya

strategis. Pemilihan strategi harga rendah tidak berarti bahwa perusahaan kemudian menjadi pelit dalam berbagai urusan misalnya pada karyawan dan pengeluaran-pengeluaran kecil. Pengurangan biaya yang strategis diusahakan misalnya dengan mencari informasi atas harga yang kompetitif dalam pasokan persediaan, membuat alternatif atau teknologi dalam cara memproduksi, dan lain sebagainya.

Dua strategi tersebut dapat dilengkapi dengan strategi yang mendukung misalnya keefektifan operasional, posisi pasar. Keefektifan operasional ini penting sebagai strategi kinerja (Porter, 1997). Strategi lain misalnya karakteristik entrepreneur.

Kemudian Porter (2001) membaca bahwa internet sudah mulai berkembang di kalangan organisasi. Teknologi internet memberikan peluang lebih baik bagi perusahaan untuk menempatkan posisi strategik daripada yang dapat dilakukan dari generasi teknologi informasi sebelumnya. Banyak perusahaan yang sukses karena menggunakan internet sebagai komplemen atas cara tradisional dalam berkompetisi.

D. Struktur Organisasi

Dalam melakukan disain sistem memerlukan kesesuaian dengan struktur organisasi dan berbagai faktor lainnya. Misalnya dalam mendisain suatu sistem akuntansi manajemen perlu juga menyesuaikan dengan faktor struktur organisasi, aspek ketidakpastian lingkungan, interdependensi organisasi (Chenhall & Morris, 1986). Faktor-faktor ini merupakan latar belakang organisasi yang mempengaruhi aktivitas dan proses manajemen.

Pihak manajemen akan merasa bahwa informasi yang dihasilkan oleh sistem mempunyai manfaat baginya. Manfaat tersebut akan lebih optimal jika dapat menyesuaikan dengan berbagai karakteristik dalam organisasi. Karakteristik yang disebutkan Chenhall antara lain: skup (*scope*), tepat waktu (*timeliness*), tingkat agregasi (*level of agregation*), dan informasi yang mendukung integrasi (*information which assists integration*).

Skup atau keluasan informasi dapat dipahami apakah cukup diberikan informasi detail untuk operasional atau informasi yang lebih singkat namun menjangkau ke berbagai organisasi bahkan sampai ke luar organisasi. Tepat waktu, dapat dipahami bahwa informasi akan bermanfaat jika diberikan pada waktu saat dibutuhkan. Jenis informasi dapat secara historikal ataupun informasi saat ini. Tingkat agregasi menyiratkan informasi yang dikumpulkan atau dijumlahkan. Misalnya dari berbagai unit bisnis kemudian dijadikan satu informasi yang teragregat. Sementara integrasi merupakan kombinasi dari berbagai informasi yang terintegrasi sehingga dapat digunakan untuk kepentingan tertentu atau dapat memiliki nilai manfaat lebih.

Struktur organisasi sering dipahami dalam dua kategori yaitu **desentralisasi** dan **sentralisasi**. **Desentralisasi** memberikan distribusi

kewenangan kepada manajemen-manajemen di tingkat lebih bawah atau unit organisasi yang berada dalam suatu organisasi. Misalnya juga apabila suatu organisasi mempunyai pusat pertanggungjawaban, kewenangan dapat dilimpahkan dari manajemen puncak kepada manajer di tingkat pusat pertanggung jawaban. Hal lain misalnya ketika organisasi mempunyai kantor cabang atau bahkan cabang-cabang di luar negeri dengan memberikan kewenangan kepada manajer cabang. Sementara **sentralisasi** dapat dipahami bahwa kewenangan untuk mengambil keputusan berada di tingkat pusat atau manajemen puncak. Manajemen di tingkat lebih rendah hanya berwenang untuk menjalankan kebijakan atau keputusan. Dalam hal ini dapat dipahami bahwa manajemen tingkat lebih rendah hanya berwenang menjalankan operasional dari keputusan atasannya.

Apa implikasi dari dua sistem dalam struktur organisasi ini? Misalnya dalam penyusunan anggaran, ketika organisasi memilih sistem desentralisasi maka bawahan (manajer pertanggungjawaban) diperkenankan untuk mengusulkan dan terlibat dalam pengambilan keputusan anggaran. Hal ini sering dikenal dengan sebutan partisipasi anggaran. Manajer pusat pertanggungjawaban hanya diijinkan untuk mengusulkan saja tanpa dilibatkan dalam pengambilan keputusan dapat dikatakan partisipasi ini merupakan partisipasi semu. Implikasi lain terkait dengan struktur organisasi misalnya akan melibatkan siapa yang dapat mengambil keputusan.

Dalam mendisain sistem informasi maka simpul-simpul ini perlu diperhatikan. Jika manajer pemasaran misalnya diperkenankan untuk memutuskan harga jual, maka sebaiknya fasilitas untuk pengambilan keputusan harga jual juga disediakan baginya. Tidak hanya sebatas memutuskan, namun kemudian manajer tersebut diberi kesempatan untuk memasukkan harga yang telah menjadi keputusannya ke dalam sistem informasi yang kemudian informasi ini dapat digunakan oleh unit lainnya.

E. Budaya Organisasi

Jika kita mengingat bahwa dalam kesuksesan pengembangan sistem informasi selain kesuksesan teknis juga perlu memperhatikan kesuksesan dari aspek pengguna atau yang sering dikenal dengan aspek keperilakuan. Di dalam organisasi terdapat individ-individu yang berperan. Karakteristik umum perilaku dari anggota organisasi dapat dipahami sebagai budaya dalam organisasi.

Chow et al. (2001) memandang dari sisi budaya nasional dari berbagai negara. Ia merumuskan bahwa pengaruh budaya dalam suatu negara akan mempengaruhi implementasi dari adanya penentuan standar kinerja yang diberlakukan. Budaya yang dimaksud adalah karakteristik perilaku individu dalam suatu negara. Jika dipersempit, maka cara pandang ini dapat digunakan untuk meninjau budaya dalam suatu organisasi.

Budaya akan berdampak terhadap sistem informasi melalui pengaruhnya pada reaksi pekerja (Chow et al. 2001). Dalam konteks budaya nasional, budaya didefinisikan oleh Hofstede (1984) sebagai “program terkoleksi atas pikiran yang membedakan anggota grup atau masyarakat dari yang satu ke lainnya”. Hofstede (1984) mengidentifikasi dimensi budaya nasional yaitu:

- *Individualism vs Collectivism*, bagaimana orientasi secara individual (individu dan keluarga) atau orientasi sosial.
- *Large vs Small Power Distance*, penerimaan kekuasaan (*power*) dalam institusi dan organisasi, apakah terdistribusi tidak sama atau hampir sama.
- *Strong vs Weak Uncertainty Avoidance*, bagaimana anggota atau masyarakat merasakan ketidaknyamanan atas ketidakpastian dan ambiguitas.
- *Masculinity vs Femininity*, maskulin merupakan preferensi masyarakat atas prestasi, kepahlawanan, ketegasan, dan kesuksesan materi. Feminim berpreferensi pada kemitraan, kesederhanaan dan kerendahatian, perhatian pada yang lemah, dan kualitas hidup.

Hofstede (1984) mengungkapkan bahwa orang-orang membangun organisasi berdasarkan nilai mereka, dan masyarakat dibentuk dari institusi-institusi dan organisasi-organisasi yang merefleksikan nilai dominan dalam budaya mereka.

Misalnya kita kaitkan konsep tersebut dengan pengembangan sistem:

- Individualisme dan kolektivisme akan mempengaruhi proses penyusunan anggaran dan pengolahan informasi terkait dengan capaian individu dan capaian organisasi serta di sisi lain adalah pengaruhnya pada kemakmuran sosial/bersama.
- Power distance akan mempengaruhi bagaimana proses penyusunan anggaran, bagaimana prosedur otorisasi.
- Maskulinitas akan mempengaruhi bagaimana sistem harus menyediakan informasi terkait dengan kinerja.

- Sikap atas Ketakpastian, akan mempengaruhi bagaimana cara menyediakan informasi sehingga dapat digunakan untuk memprediksi masa depan atau menentukan keputusan.

Chow et al. (2001) mempelajari budaya nasional dari sisi perbedaan orientasi atas *power distance* dan *colectivism/individualism*. Mereka mempelajari bagaimana sikap berdasarkan budaya dalam kaitannya dengan adanya peningkatan standar kinerja yang harus dicapai. Dalam penentuan standar kinerja dapat digunakan partisipasi dari anggota. Masyarakat yang kolektifisme akan cenderung untuk menerima keputusan dari atasan. Demikian juga jika kita hubungkan dengan power distance maka semakin kuat power distance, anggota akan makin cenderung menerima keputusan.

Memahami karakteristik budaya akan berimplikasi bahwa penerapan kebijakan manajemen serta proses dalam sistem informasi harus disesuaikan agar menunjukkan kinerja yang paling optimal. Keberhasilan metode dalam negara tertentu jika diadopsi begitu saja dengan cara yang sama, belum tentu akan menunjukkan kinerja yang sama.

Referensi

- Jogiyanto, H.M. 2005. *Sistem Informasi Strategik: untuk keunggulan kompetitif*. Edisi 2, 2006. Penerbit Andi Offset. Yogyakarta.
- Porter, Michael E. 1996. What is Strategy? *Harvard Business Review*. November-December 1996. p. 61-78
- Porter, Michael E. 1997. Replies: What is Strategy? *Harvard Business Review*. January – February 1997.
- Porter, Michael E. 2001. Strategy and the Internet. *Harvard Business Review*. March 2001.
- Chenhall, Robert H; and Deigan Morris. 1986. The Impact of Structure, Environment, and Interdependence on the Perceived Usefulness of Management Accounting Systems. *The Accounting Review*. Vol LXI, No 1, January 1986.
- Chow, Chee W; Tim M Lindquist; Anne Wu. 2001. National culture and the Implementation of High-Stretch Performance Standards: An Exploratory Study. *Behavioral Research in Accounting*. Vol 13, 2001.

Hofstede, Geert. 1984. Cultural Dimensions in Management And Planning. *Asia Pacific Journal of Management*, January 1984.

BAB 5: KUALITAS LAYANAN

Penting bagi seorang analis dan perancang untuk menyadari adanya kualitas dalam sistem. Salah satu tinjauannya adalah kualitas sistem dalam memberikan layanan yang disebut kualitas layanan (*SERVQUAL*). *SERVQUAL* dapat digunakan sebagai peralan diagnosa untuk melihat area keunggulan dan kegagalan pendek layanan (Kettinger & Lee, 1997). Apabila ingin melakukan perbandingan dengan melakukan *benchmarking* maka nilai *SERVQUAL* ini dapat digunakan sebagai pembanding antar organisasi atau dengan 'industri yang sama'.

Kualitas layanan ini dapat diidentifikasi ke dalam dimensi sebagai berikut (Parasuraman et al. 1991):

1. *Tangible*, yaitu penampilan fasilitas fisik, perlengkapan, personal, dan materi komunikasi.
2. *Reliability*, yaitu kemampuan memberikan layanan yang dijanjikan dapat diharapkan dan akurat.
3. *Responsiveness*, yaitu keinginan untuk membantu pelanggan dan menyediakan layanan.
4. *Assurance*, yaitu pengetahuan dan kesopanan dan pekerja dan kemampuan mereka untuk membentuk kepercayaan dan keyakinan.
5. *Empathy*, yaitu sikap perhatian untuk menarik perhatian pelanggan secara individu.

Memahami dimensi dalam *SERVQUAL* dimungkinkan dapat mempelajari lebih dalam tentang kualitas layanan dalam sistem informasi dan melakukan diagnosa seperlunya. Misalnya analisis dalam kemampuan reliabilitas sistem ternyata membutuhkan lebih banyak staf.

Ketika memperhatikan kualitas, faktor pelanggan atau pengguna menjadi suatu hal yang penting. Perusahaan harus memperhatikan masukan dan informasi dari para pelanggannya untuk meningkatkan kualitas (Garvin, 1987). Dalam hal kualitas, Garvin (1987) mengajukan delapan dimensi atau kategori penting yang dapat memberikan kerangka analisis strategik: *performance*, *features*, *reliability*, *comformance*, *durability*, *serviceability*, *aesthetics*, dan *perceived quality*.

Dalam suatu produk atau jasa/layanan dimensi tertentu mungkin memiliki ranking tinggi dan yang lain rendah. Suatu perusahaan tidak diharuskan mengusahakan semua dari kedelapan dimensi kualitas secara

bersama. Perusahaan dapat memilih kualitas yang menjadi celah kualitas perusahaan lain atau yang menjadi dimensi kualitas kritis konsumen. Hal yang pasti adalah: kualitas tinggi berarti memuaskan pengguna, tidak hanya memproteksi mereka dari gangguan. Bahkan, komplain pengguna memainkan peran baru karena memberikan sumberdaya yang berharga atas informasi produk. Hal yang perlu hati-hati dalam membandingkan kualitas atau kinerja suatu produk atau layanan adalah kemungkinan adanya preferensi yang berbeda dari pengguna.

1. *Performance*, mengacu pada karakteristik operasi utama produk. Misalnya di bidang otomotif, kinerja akan dilihat menurut akselerasi, handling, kecepatan, dan kenyamanan. Untuk peralatan televisi, kinerja berarti suara dan kejelasan gambar, warna, dan kemampuan menerima dari stasiun. Untuk bisnis jasa, misalnya makanan cepat saji dan pesawat udara, kinerja diterjemahkan sebagai bentuk layanan.
2. *Feature*. Fitur merupakan aspek kedua kinerja. Fitur disebut sebagai “lonceng dan peluit” atas produk dan jasa. Karakteristiknya merupakan pelengkap dari fungsi dasarnya. Contoh minuman gratis pada pesawat, alat permanen dalam siklus pencucian mesin, *automatic tuners* pada televisi berwarna.
3. *Reliability*. Reliabilitas merefleksikan kemungkinan produk tidak berfungsi atau gagal dalam spesifikasi waktu tertentu. Misalnya dapat diukur dengan kerusakan pertama, rata-rata waktu sampai kerusakan, dan rata-rata waktu kerusakan dalam satuan waktu. Secara normal, hal ini menjadi penting bagi pelanggan jika waktu macet (*downtime*) dan pembenahannya menjadi sesuatu yang mahal.
4. *Conformance*, dapat juga disebut sebagai tingkat kesesuaian antara disian produk beserta karakteristik operasinya dengan standar yang ditentukan. Setiap produk mempunyai spesifikasi. Spesifikasi ini dapat diekspresikan sebagai target pusat, perbedaan dengan target pusat yang diijinkan adalah dalam jarak tertentu. Pendekatan pengukuran kualitas seperti tersebut dapat dilihat pada konsep yang diprakarsai oleh pakar misalnya Juran.
5. *Durability*. Merupakan ukuran dari umur produk, mempunyai dua dimensi yaitu ekonomis dan teknis. Durasi secara teknis dapat didefinisikan sebagai jumlah penggunaan produk sampai keadaannya kurang baik. Misalnya bolam lampu perlu diganti setelah sekian jam penggunaan.
6. *Serviceability*. Sebutan lain dari *serviceability* adalah kecepatan (*speed*), kesopanan (*courtesy*), kompetensi (*competence*), dan kemudahan

direparasi (*ease of repair*). Pelanggan tidak hanya konsen pada kerusakan atau kemacetan produk namun juga waktu reparasinya, ketepatan waktu, kaitan dengan personelnnya, dan frekuensi pemanggilan reparasi.

7. *Aesthetics*. Berkaitan dengan bagaimana penampilan produk, perasaan, suara, rasa. Hal ini tergantung dari selera dan pertimbangan preferensi individu. Namun demikian dapat dicari ranking dari pelanggan.
8. *Perceived Quality*. Kualitas persepsian merupakan kualitas berdasarkan persepsi dari pelanggan. Pelanggan tidak selalu mempunyai informasi lengkap tentang atribut produk atau jasa. Sehingga kadang terpengaruh oleh gambar, iklan, dan nama merek.

Referensi:

- Kettinger, William J.; Choong C. Lee. 1997. Pragmatic Perspectives on the Measurement of Information Systems Service Quality. *MIS Quarterly*. June 1997.
- Parasuraman, A; Leonard L. Berry; Valarie A Zeithaml. 1991. Refinement and Reassessment of the SERVQUAL Scale.
- Garvin, David A. 1987. Competing on the eight dimension of quality. *Harvard Business Review*. Nov – Dec 1987.

BAB 6: BAHASA PEMODELAN SISTEM

A. Mengapa perlu bahasa pemodelan?

Dapat kita pahami bahwa model merupakan cara untuk menjelaskan suatu keadaan atau kondisi yang kompleks ke dalam konsep yang lebih sederhana sehingga mudah dipahami. Sementara bahasa merupakan cara untuk berkomunikasi sehingga mudah dipahami oleh orang lain atau pihak yang berkepentingan. Sehingga jika dikumpulkan, maka bahasa pemodelan sistem adalah cara untuk mengkomunikasikan keadaan atau fenomena yang kompleks dari adanya sistem.

Mengapa diperlukan pemodelan sistem? Berikut ini adalah hal-hal terkait dengan filosofi pemodelan sistem.

- Piranti lunak, arsitekturnya harus didefinisikan dengan jelas, agar *bug* mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya.
- Keuntungan lain adalah, dimungkinkannya penggunaan kembali modul atau komponen untuk aplikasi piranti lunak lain yang membutuhkan fungsionalitas yang sama.
- Pemodelan (*modeling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*).
- Model piranti lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan gedung.

B. DIAGRAM SISTEM

Sebenarnya terdapat berbagai cara untuk menjelaskan sistem baik secara tekstual atau grafis. Diagram sistem menggunakan pendekatan secara grafis. Di dalam pembuatan diagram sistem pun terdapat banyak cara yang telah dikenal di kalangan analis atau pengembang sistem.

Diagram sistem (*System Diagrams / SD*) adalah komponen dalam dokumentasi sistem dan menjadi lebih penting sebagai respon atas meningkatnya kesadaran terjadinya peningkatan proses dan keperluan dokumentasi sistem (Bradford et al., 2007). Ia mengidentifikasi tipe metode diagram sistem sebagai berikut:

- *System flowcharts* atau bagan sistem.
- *Entity-relationship (E-R) diagrams* atau diagram hubungan entitas

- *Data flow diagrams* (DFD) atau diagram arus data
- *Resource-event-agent* (REA) *model* atau model sumber-kejadian-agen
- *Process maps* atau peta proses
- *Unified Modeling Language* (UML) atau bahasa pemodelan gabungan

Banyak organisasi memberikan fokus pada usaha pendokumentasian sistem untuk memperoleh pemahaman akan peningkatan kekomplekan sistem informasi dan kontrol berdasar sistem yang terkait (Bradford et al., 2007). SD ini memegang peran penting dalam dokumentasi dengan memberikan tampilan secara grafis pada pengendalian internal, arus data, dan arus informasi yang terkait dengan proses pokok yang mendukung organisasi. Berbagai macam praktisi akuntansi yang bekerja baik di dalam maupun di luar organisasi harus dapat membaca dan menyiapkan SD. Pemahaman ini dapat digunakan ketika melakukan analisis kebutuhan, analisis kelemahan, penataan ulang, dan pemodelan data atau bisnis.

1. System Flowcharts

Bagan sistem digunakan untuk menunjukkan proses informasi seperti arus logik (*logic flows*), *inputs*, *outputs*, penyimpanan data (*data storage*), dan proses operasional (*operational processes*) seperti arus fisik, aktifitas, dan entitas.

2. Data Flow Diagrams (DFDs)

DFD berasal dari sistem informasi manajemen (SIM) dan digunakan untuk menggambarkan proses sistem, arus antar proses, dan sumber, tujuan, serta penyimpanan data. Dua tipe DFD adalah: *logical DFDs* yang berfokus pada aktifitas di dalam sistem, dan *physical DFDs* meliputi “*who, where, and how*” dari sistem. DFD ini merupakan teknik yang populer dalam penggunaannya di SIM sampai dikembangkannya diagram berorientasi objek (*object oriented*).

3. Entity-Relationship (E-R) Diagrams

Metode ini dikembangkan oleh Chen pada tahun 1976. Merupakan teknik grafis yang digunakan untuk menggambarkan skema database. *ER diagrams* mengilustrasikan struktur logik atas database dengan memperhatikan entitas-entitas dalam sistem.

4. Resource, Event, and Agent (REA) Models

REA model adalah peralatan konseptual khususnya dalam disain untuk memberikan petunjuk dan struktur dalam mendisain keterhubungan dalam sistem informasi akuntansi. Dalam REA, entitas dibedakan dalam tiga kategori: sumber daya yang diperoleh dan digunakan organisasi, kegiatan yang terkait dengan organisasi, dan agen atau pihak yang berpartisipasi dalam kegiatan.

5. Process Maps

Merupakan teknik diagram sistem yang dikembangkan oleh General Electric pada tahun 1980an dan digunakan di berbagai organisasi untuk melakukan dokumentasi, analisis, pelangsingan, dan mendisain ulang aktivitas bisnis. Peta proses ini digunakan untuk menunjukkan bagaimana pekerjaan dilakukan oleh organisasi dan bagaimana meningkatkannya.

6. Unified Modeling Language (UML)

UML juga mengeluarkan diagram sistem menggunakan orientasi objek (*object oriented*) dalam analisis dan perancangan sistem. UML saat ini menjadi standar dalam berbagai tipe solusi piranti lunak dalam pengembangan sistem.

Penggunaan Diagram Sistem

Bradford et al. (2007) mengidentifikasi penggunaan berbagai diagram tersebut dan mengkategorikan penggunaannya dalam:

- Menjelaskan proses bisnis
- Mengevaluasi sistem berjalan
- Disain atau merubah sistem
- Lingkungan pengendalian internal

Dalam surveinya, ia menyebutkan bahwa teknik *system flowcharts* paling banyak digunakan, kemudian *process maps*, *data flow diagrams*, *REA models*, *E-R diagrams*, dan yang paling sedikit adalah *UML*.

UML memang paling sedikit digunakan, mungkin karena masih baru dan belum populer di kalangan praktisi dan pendidik sistem informasi akuntansi. Namun menurut Fowler, walau belum dibuktikan secara empiris (*anecdotal evidence*), menyatakan bahwa UML akan meningkat penggunaannya sejalan dengan kematangannya sebagai metode diagram sistem.

C. UNIFIED MODELING LANGUAGE (UML)

Mengingat perkembangan UML yang banyak digunakan untuk perancangan sistem khususnya dalam disain piranti lunak. Maka berikut ini akan disampaikan tentang bahasa pemodelan gabungan (*unified modeling language*) ini. Perlu diingat kembali bahwa UML hanyalah merupakan salah satu cara dalam menjelaskan perancangan sistem, masih terdapat berbagai cara yang dapat digunakan seperti yang antara lain telah disebutkan di atas dalam diagram sistem.

Pengenalan UML

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML mendefinisikan notasi dan syntax/semantik.

Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Notasi UML diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented-Design), Jim Rumbaugh OMT (Object Modeling Technique) dan Ivar Jacobson OOSE (Object-Oriented Software Engineering). Sampai era tahun 1990 terdapat puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya tiga notasi tersebut di atas. Masa itu dikenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Karena setiap metodologi membawa notasi sendiri-sendiri, mengakibatkan masalah ketika bekerjasama dengan group/perusahaan lain.

Mulai Oktober 1994 Booch, Rumbaugh dan Jacobson mempelopori usaha untuk menyatukan metodologi pendesainan berorientasi objek. Tahun 1995 direlease draft pertama dari UML. Pada tahun 1999 ketiga tokoh tersebut menyusun tiga buku serial tentang UML, sejak saat itulah

UML menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek.

Unified Modeling Language (UML) adalah bagian/salah satu bentuk notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO) (Fowler: 2005).

UML dapat diibaratkan sebagai suatu *blueprint* pada suatu bangunan. Desain dengan UML digunakan oleh seorang programmer untuk melakukan pengkodean. Bisa terjadi seorang desainer adalah juga seorang programmer.

Kuske et al. (2009) menyatakan bahwa dewasa ini, UML telah diterima secara luas sebagai standar bahasa untuk pemodelan dan dokumentasi perangkat lunak. UML menawarkan sejumlah diagram yang dapat digunakan untuk menjelaskan sejumlah aspek dalam objek piranti lunak. Kuske menyebutkan bahwa UML dapat digunakan untuk menjelaskan tiga aspek penting yaitu aspek **struktural**, **keperilakukaan** (*behavioral*), dan **kolaborasi**.

Fowler (2005) memberikan daftar UML 2 terdiri dari 13 jenis diagram resmi seperti dalam tabel berikut ini:

Jenis Diagram Resmi UML

Diagram	Kegunaan
<i>Activity</i>	<i>Behaviour</i> prosedural dan paralel
<i>Class</i>	<i>Class</i> , <i>fitur</i> , dan hubungan-hubungan
<i>Communication</i>	Interaksi antar objek; penekanan pada jalur
<i>Component</i>	Struktur dan koneksi komponen
<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
<i>Deployment</i>	Pemindahan artifak ke <i>node</i>
<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity diagram</i>
<i>Object</i>	Contoh konfigurasi
<i>Package</i>	Struktur hirarki <i>compile-time</i>
<i>Sequence</i>	Interaksi antar objek; penekanan pada <i>sequence</i>
<i>State machine</i>	Bagaimana <i>even</i> mengubah objek selama aktif
<i>Timing</i>	Interaksi antar objek; penekanan pada <i>timing</i>
<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sebuah sistem

UML (*Unified Modeling Language*) dapat digunakan untuk menjelaskan beberapa hal yang penting dalam sistem. Tidak semua diagram UML harus dipakai, dan tidak semua sistem dijelaskan dengan UML karena terdapat model penjelasan lain, atau dengan kata lain harus dipilih yang penting dalam penjelasan sistem sesuai kemampuan perancang dan pemahaman penulis program (Fowler, 2005). Bagi perancang pemula disarankan oleh Fowler (2005) untuk memusatkan perhatian pada bentuk-bentuk diagram dasar *class diagram* dan *sequence diagram*, diagram tersebut adalah bentuk yang paling umum dan diagram yang paling berguna.

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*.

Sequence diagram, secara khusus, menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek di dalam *use case*.

Kuske et al. (2009) dalam artikelnya juga menyampaikan bahwa cara yang cukup berarti dalam mengaplikasikan UML adalah menggunakan diagram *class*, *state*, dan *interaction* sebagai dasar penjelasan sistem. *Class diagrams* menjelaskan dasar penjelasan **struktur objek**, *state diagram* untuk menjelaskan **object behaviour**, dan *interaction diagrams* menjelaskan bagaimana objek-objek berinteraksi dalam suatu **kolaborasi**.

Referensi:

- Bradford, Marianne; Sandra B. Richtermeyer; Douglas F. Roberts. 2007. System Diagramming Techniques: An Analysis of Methods Used in Accounting Education and Practices. *Journal of Information Systems*. Vol. 21, No. 1, Spring, pp. 173-212.
- Fowler Martin. 2005. "UML Distilled Panduan Singkat Bahasa Pemodelan Objek Standar" Edisi 3 dalam bahasa Indonesia. Andi: Yogyakarta

- Kuske, Sabine; Gogolla, Martin; Kreowski, Hans-Jörg; & Ziemann, Paul. 2009. Towards an integrated graph-based semantics for UML. *Softw Syst Model* (2009) 8:403–422
- Chen, Peter Pin-Shan. 1976. The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, Pages 9-36.

BAB 7: PEMODELAN DATABASE

A. Entity-Relationship (E-R) Diagram

E-R diagram dikenalkan oleh Chen (1976) yang ia sebut sebagai *entity-relationship model*. E-R model menggunakan teknik diagram khusus sebagai peralatan dalam mendisain database. Pada saat itu terdapat tiga model utama yang digunakan untuk menggambarkan data logik yaitu: *the network model*, *the relational model*, dan *the entity set model*. E-R model dimaksudkan untuk menanggapi kelebihan dan kelemahan dari ketiga model tersebut. Diharapkan E-R model mampu menjabarkan keadaan senyatanya yang terjadi di dalam kepentingan penjelasan entitas dan hubungannya.

Di dalam pemodelan data, kita sebaiknya mengidentifikasi tingkat pandangan logika data (*logical views of data*) (Chen, 1976):

1. Informasi tentang entitas (*entities*) dan hubungannya (*relationships*) yang muncul dalam benak kita.
2. Struktur organisasi informasi (*information structure-organization*) yang direpresentasikan oleh data pada entitas dan hubungannya.
3. Struktur data jalur-akses-independen (*access-path-independent*). Yaitu struktur data yang tidak termasuk dalam skema pencarian, skema indeks, dan sebagainya.
4. Struktur data jalur-akses-dependen (*access-path-dependent*)

Jika kita runut kembali ke dalam tiga model sebelumnya, maka level 1 dan 2 merupakan bagian dari *the entity set model*. Level 2 dan 3 merupakan bagian dari *the relationship model*. Level 4 berdasarkan *the network model*.

1. Informasi terkait *Entities and Relationship* (Level 1)

Pada level ini kita mempertimbangkan entitas dan hubungannya. Suatu entitas (*entity*) adalah sesuatu (*thing*) yang dapat diidentifikasi secara terpisah (*distinctly indentified*). Seseorang, perusahaan, atau kejadian (*event*) adalah contoh dari entitas. Suatu hubungan (*relationship*) adalah asosiasi antar entitas. Contoh adalah hubungan antara “ayah-anak” yang merupakan hubungan antara dua entitas “orang”.

Dalam database perusahaan akan berisi berbagai informasi, namun pada pemodelan ini kita hanya mempertimbangkan entitas dan hubungannya. Namun demikian, mungkin tidak semua hubungan antar entitas yang terjadi dalam perusahaan dapat semuanya terekam dalam database. Yang dijabarkan

pada model diagram ini hanyalah entitas yang dirasa akan masuk ke dalam disain database.

Entity and Entity Set

Entitas dapat diklasifikasikan ke dalam set entitas (*entity set*) misalnya PEKERJA, PROYEK, dan DEPARTEMEN. Di dalam suatu entitas dimungkinkan masih terdapat subset entitas. Misalnya ORANG-LAKI adalah subset dari ORANG.

Relationship, Role, and Relationship Set

Untuk menjelaskan relationship, kita misalkan pernikahan, pernikahan merupakan hubungan antara dua entitas dalam set entitas ORANG. Aturan (*role*) yang membentuk hubungan misalnya adalah “Suami” dan “Istri”.

Attribute, Value, and Value Set.

Informasi tentang entitas atau hubungannya dibentuk dari observasi ataupun pengukuran, dan diekspresikan dengan suatu set pasangan nilai atribut (*a set of attribute-value pairs*). Nilai (*value*) misalnya dicontohkan dari beberapa nilai “170”, “hitam”, “Peter”, “Johan”. Nilai-nilai tersebut diklasifikasikan dalam set nilai, misalnya TINGGI, WARNA, NAMA-DEPAN, NAMA-BELAKANG. Tinggi misalnya menjelaskan dalam ukuran cm, hitam menjelaskan warna rambut.

Atribut merupakan penjelas dalam set entitas. Beberapa atribut akan mendefinisikan set entitas. Misalnya atribut NAMA yang dapat dipetakan nilainya (*maps*) dari NAMA-DEPAN, dan NAMA-BELAKANG yang merupakan bagian dari set entitas ORANG.

2. Struktur Informasi (*Information Structure*) (Level 2)

Pada level ini kita mempertimbangkan representasi dari objek konseptual. Kita mengasumsikan bahwa terdapat representasi langsung dari nilai.

Primary Key

Primary key (PK) digunakan untuk mengidentifikasi entitas dalam set entitas. Misalnya atribut NO-PEKERJA dapat digunakan untuk mengidentifikasi entitas personal pekerja dari set entitas PEKERJA. Setiap orang pekerja mempunyai nomor yang berbeda. Dalam kasus lain dimungkinkan bahwa suatu identitas membutuhkan lebih dari satu atribut.


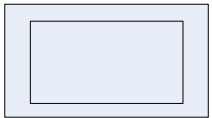
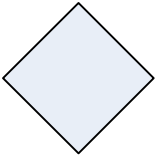
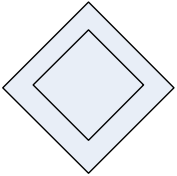
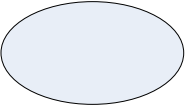
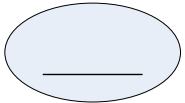
Entity/Relationship Relations

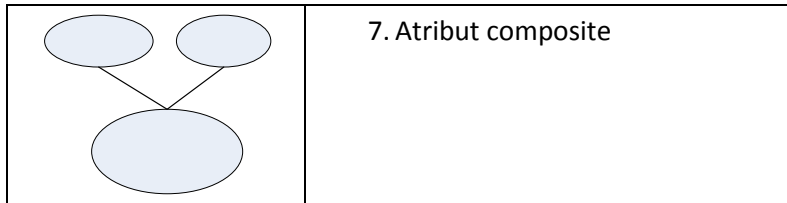
Informasi seputar entitas dalam set-entitas kemudian dapat diperlihatkan. Entitas direpresentasikan oleh nilai dari *primary key*.

Dalam suatu hubungan antar entitas dimungkinkan terjadinya satu entitas terkait dengan beberapa entitas yang dapat disimbolkan dengan $1:n$. Angka n misalnya dapat kita contohkan $n (= 0, 1, 2, \dots)$. Dimisalkan satu orang dapat memiliki beberapa motor.

B. Simbol Entity Relationship Diagram

Simbol ERD dapat dilihat pada gambar-gambar di bawah ini.

Simbol/Notasi	Keterangan
	1. entity
	2. weak entity
	3. relationship
	4. Identifying relationship
	5. Atribut
	6. Atribut primary key



Berikut ini adalah penjelasan dan contoh penggunaan simbol ERD.

1. Entity

Entity adalah obyek yang dapat dibedakan dalam dunia nyata.

Entity set adalah kumpulan dari entity yang sejenis.

Entity set dapat berupa:

- Obyek Secara fisik : Rumah, Kendaraan, Peralatan.
- Obyek secara konsep : Pekerjaan, Perusahaan, Rencana.

2. Relationship

Relationship adalah hubungan yang terjadi antara satu atau lebih entity.

Relationship set adalah kumpulan relationship yang sejenis.



Gambar 1: contoh bentuk relationship

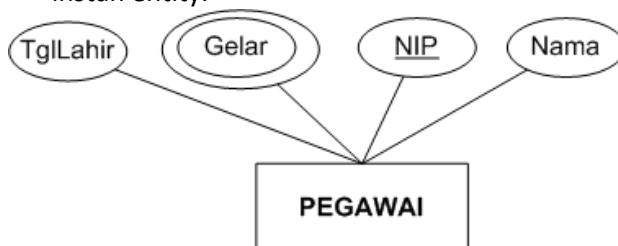
3. Atribut

Atribut adalah karakteristik dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut.

Nilai atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

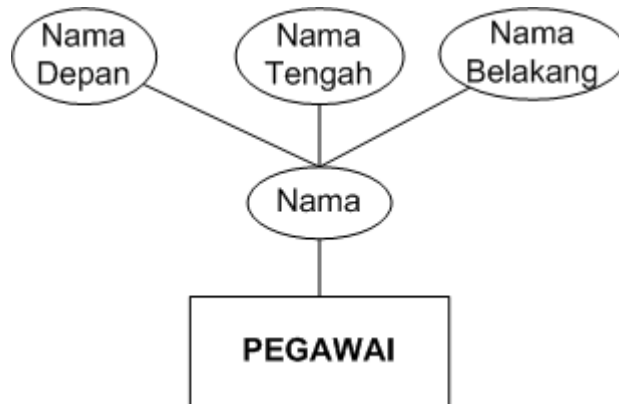
Jenis-jenis atribut

- Key: atribut yang digunakan untuk menentukan suatu entity secara unik.
- Atribut Simple: atribut yang bernilai tunggal.
- Atribut Multivalue: atribut yang memiliki sekelompok nilai untuk setiap instan entity.



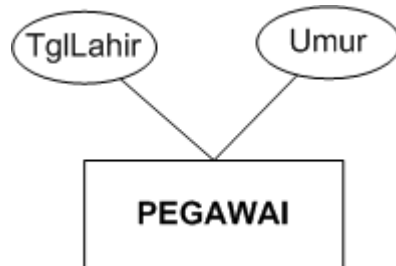
Gambar 2: contoh penulisan atribut (key, simple, multivalue)

- Atribut Composite: Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



Gambar 3: contoh atribut composite

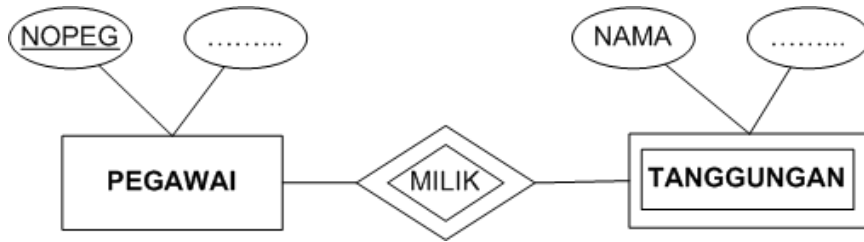
- Atribut Derivatif: suatu atribut yang dihasilkan dari atribut lain.



Gambar 4: contoh atribut derivatif

4. Weak Entity

- Weak Entity adalah suatu Entity dimana keberadaan dari entity tersebut tergantung dari keberadaan entity lain.
- Entity yang merupakan induknya disebut *Identifying Owner* dan relationshipnya disebut *Identifying Relationship*.
- Weak Entity selalu mempunyai *Total Participation constraint* dengan *Identifying Owner*.



Gambar 5: contoh penulisan weak entity

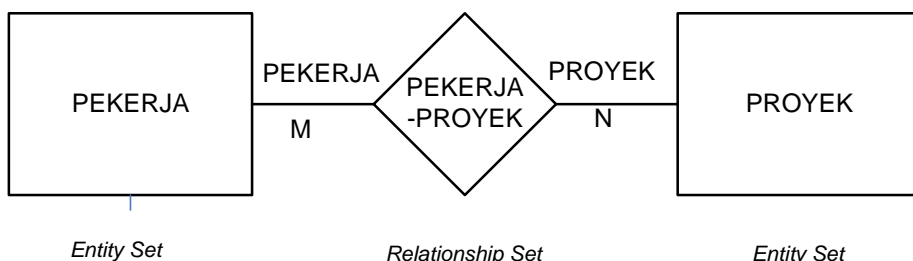
5. Transformasi dari ERD ke Database Relational

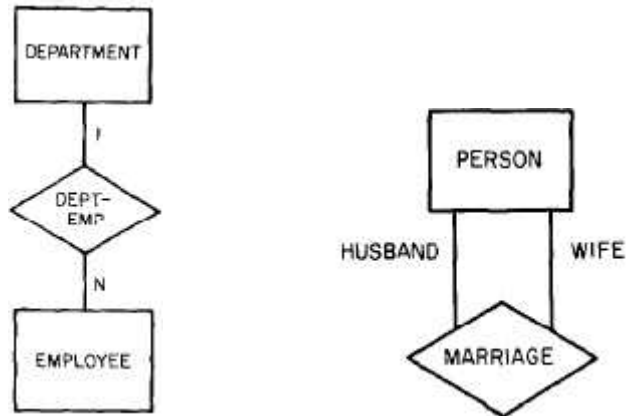
- Setiap tipe Entity dibuat suatu relasi yang memuat semua atribut simple, sedangkan untuk atribut composite hanya dimuat komponen-komponennya saja.
- Setiap relasi yang mempunyai atribut multivalued, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key dari relasi tersebut dengan atribut multivalued.
- Setiap tipe Weak Entity, dibuat suatu relasi yang memuat semua atributnya dimana Primary Keynya adalah gabungan dari Partial Key dan Primary Key dari relasi induknya (identifying owner).

Contoh Hasil Transformasi dari Diagram ER ke Database Relational:

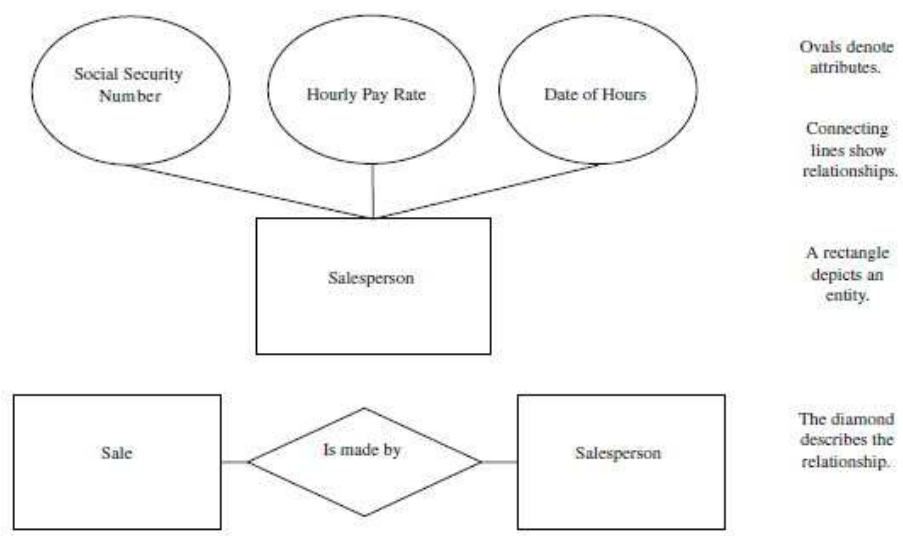
- PEGAWAI : (nopeg, napeg, alm1, kdpos, tglah, umur, supervisor-id, nobag)
- BAGIAN : (nobag, nabag, lokasi, manager)
- PROYEK : (nopro, napro, nobag)
- LOKASI_PROYEK : (nopro, lokapr)
- PEKERJAAN : (nopeg, nopro, jam)
- TANGGUNGAN : (nopeg, nama, jnkel, hubungan)

3. Contoh bentuk aplikasi diagram dalam E-R

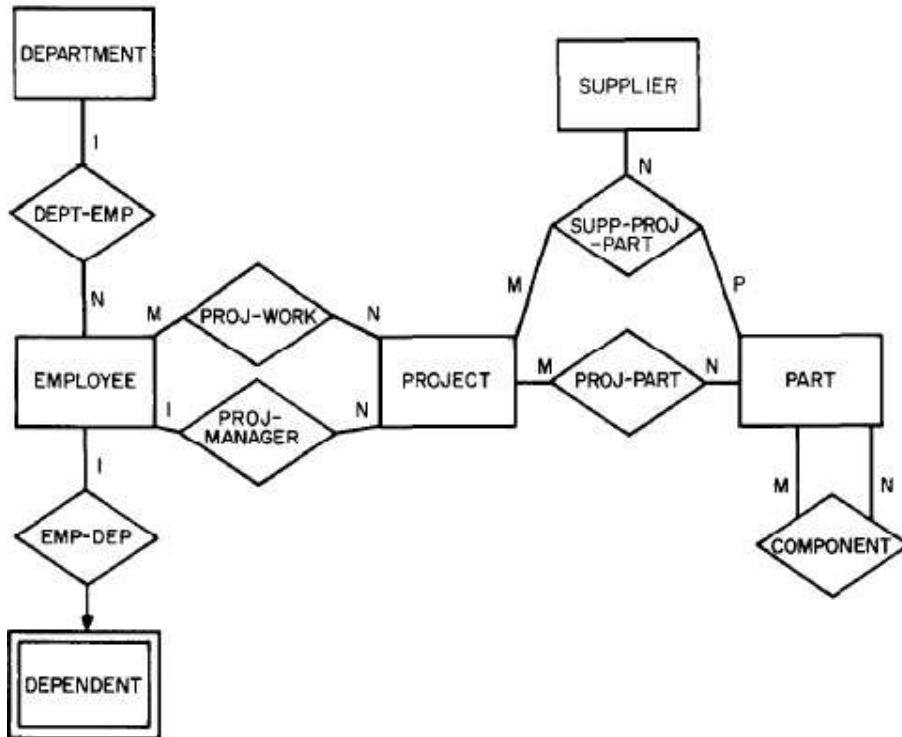




Gambar 1: contoh sederhana entity relationship diagram (sumber Chen, 1976)



Gambar 2: Contoh sederhana ER diagram untuk menjelaskan atribut (Sumber Bradford et al. 2007)



Gambar 3: ERD dalam analisis perusahaan manufaktur (Chen, 1976)

C. ERD Menggunakan Aplikasi MsVisio

MsVisio juga menyediakan fasilitas untuk menggambar bagan ERD, namun bentuk modelnya agak berbeda dengan bagan simbol di atas. Jika Anda ingin menggambar ERD menggunakan MsVisio, maka langkahnya adalah sebagai berikut:

1. Aktifkan aplikasi MsVisio
2. Pilih Shapes [DATABASE], kemudian pilih [Entity Relationship]
3. Klik dan tahan simbol entity dari panel kemudian masukkan ke lembar kerja.

Table1	

4. Jika Anda klik kotak tersebut maka dibawahnya akan muncul properties objek yang dapat Anda isi: definition [Physical Name], column [Physical Name], centang untuk pilihan Primary Key-nya.

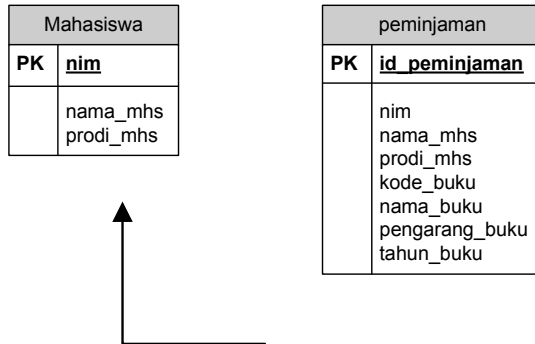
Misal isi definition physical name dengan: mahasiswa

Column: nim (primary key), nama_mhs, prodi_mhs

Physical Name	Data Type	Req'd	PK	
nim	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	nim identifies Mahasiswa
nama_mhs	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	nama_mhs is of Mahasiswa
prodi_mhs	CHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	prodi_mhs is of Mahasiswa
		<input type="checkbox"/>	<input type="checkbox"/>	

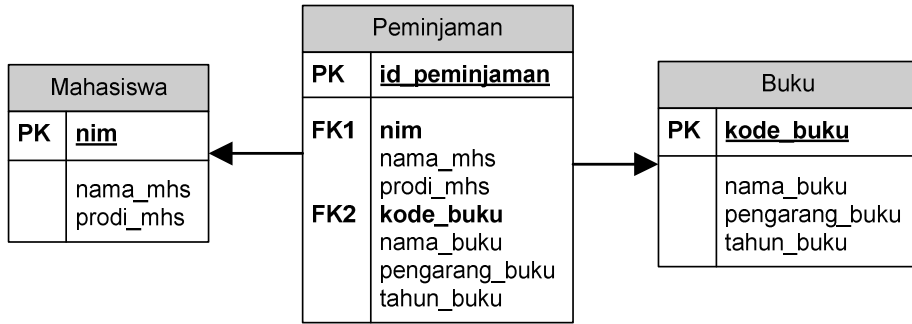
Kemudian buatlah entity lainnya seperti langkah di atas misalnya entity buku dan entity peminjaman. Lihat bagan di bawah untuk tujuan pembuatannya.

5. Ambil dari panel garis relationship, kemudian letakkan di lembar di luar entity. Klik dan tahan awal garis relationship dan letakkan pada tabel transaksinya (peminjaman) sampai muncul kotak merah (pertanda hubungan telah pas dan siap), setelah muncul kotak merah kemudian lepaskan. Jika berhasil maka akan muncul foreign key (FR) apabila pada data transaksi memiliki field yang sama dengan primary key (PK) tabel master.



Kemudian lakukan hal yang sama untuk ujung panah untuk peletakkannya pada tabel master (mahasiswa).

6. Lakukan dan buat entity relationship yang hasil akhirnya seperti di bawah ini.

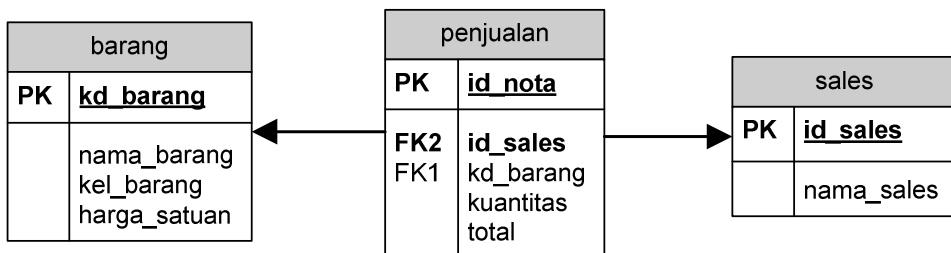


Dari bagan tersebut dapat ditransformasikan ke dalam database sebagai berikut:

- Mahasiswa : nim, nama_mhs, prodi_mhs
- Buku : kode_buku, nama_buku, pengarang_buku, tahun_buku
- Peminjaman : id_peminjaman, nim, kode_buku, nama_mhs, prodi_mhs, nama_buku, pengarang_buku, tahun_buku

Latihan

Buat entity relationship menggunakan MsVisio sehingga sesuai dengan gambar barikut ini. Buat juga penjelasan transformasi ERD ke dalam struktur databasenya.



D. PROGRAM APLIKASI

Untuk membuat Chen diagram E-R di komputer, Anda dapat memanfaatkan berbagai program aplikasi, misalnya:

- Program berbasis office dengan cara menggambar biasa. Word menggunakan shapes, power point menggunakan shapes, atau Visio menggunakan block diagram, demikian pula Anda dapat menggunakan program Visio di shapes database dan memilih entity relationship.
- SmartDraw (berlisensi) yang dapat Anda kunjungi di web <http://www.smardraw.com>

- Toad Data Modeller yang dapat Anda kunjungi di <http://www.casestudio.com>
- Edraw Soft yang dapat Anda kunjungi di <http://www.edrawsoft.com>
- Atau mencari software lain yang mendukung pembuatan ER diagram.

Pertanyaan

1. Coba Anda cari contoh beberapa entitas yang dapat Anda temukan dalam keseharian serta sebutkan pula apa hubungan antar entitas tersebut?
2. Identifikasikan suatu entitas yang memiliki subset entitas.
3. Identifikasikan suatu atribut dan pemetaan nilainya.

Referensi:

- Bradford, Marianne; Sandra B. Richtermeyer; Douglas F. Roberts. 2007. System Diagramming Techniques: An Analysis of Methods Used in Accounting Education and Practices. *Journal of Information Systems*. Vol. 21, No. 1, Spring, pp. 173-212.
- Chen, Peter Pin-Shan. 1976. The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, Pages 9-36.

[BAB 8] UML: Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal.

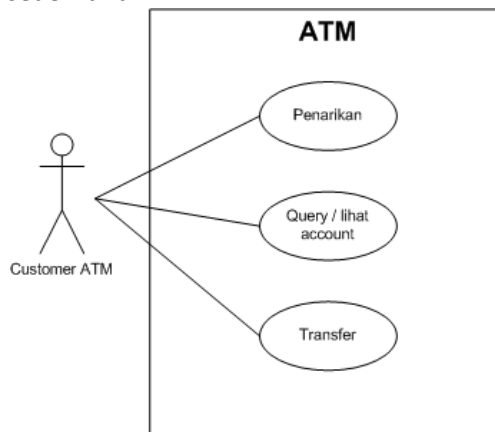
Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Tugas:

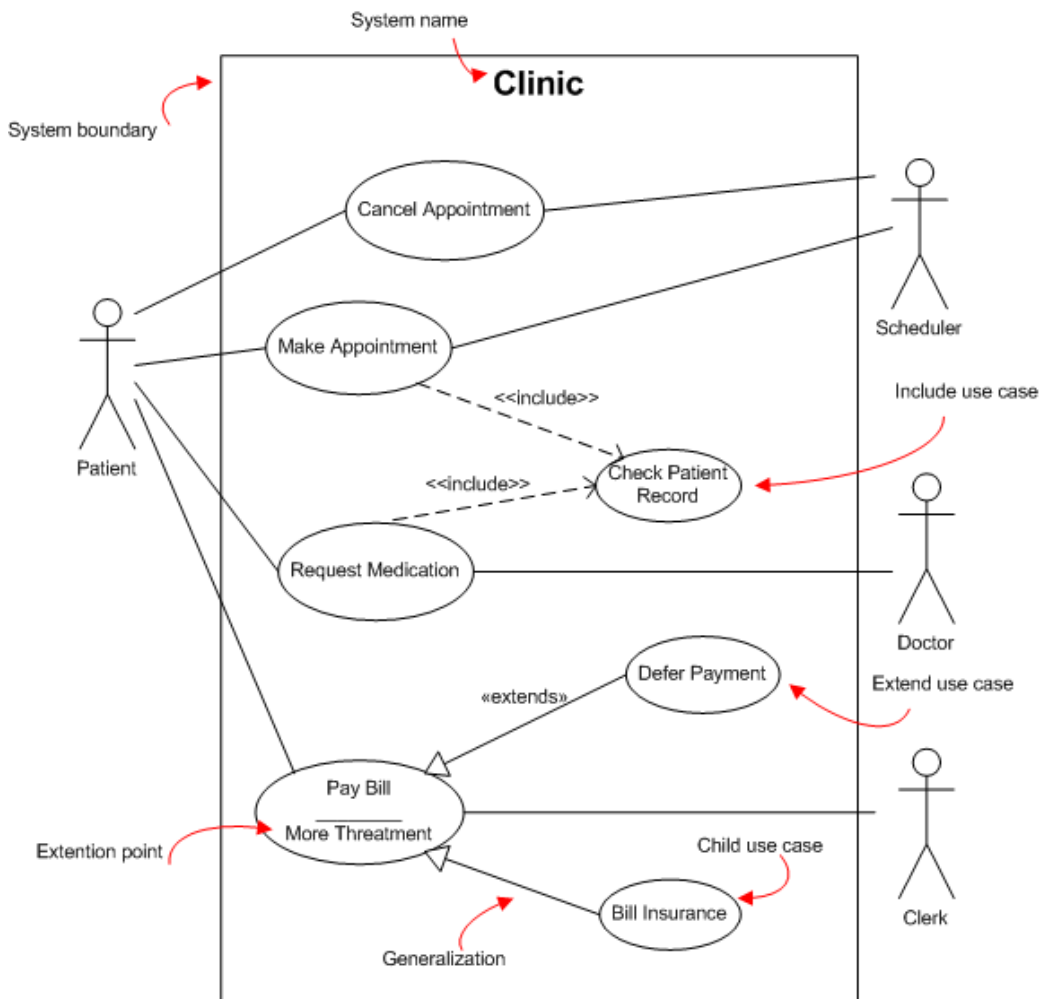
Perhatikan contoh-contoh kasus UML Use Case Diagram pada materi ini. Perhatikan pula bentuk bagan, antara lain: Boundary system, judul sistem, user, kegiatan. Kemudian, Anda diminta untuk merancang sebuah use case untuk kasus yang Anda tentukan sendiri. Misalnya (alternatif): sistem penjualan, input transaksi jurnal keuangan, stok barang gudang, dsb.

Contoh diagram use case

Contoh 1: *use case sederhana*



Contoh 2



Contoh:

Text Use Case

PEMBELIAN SEBUAH PRODUK

Skenario Keberhasilan Utama:

1. Pelanggan melihat-lihat katalog dan memilih barang untuk dibeli.
2. Pelanggan memeriksa
3. Pelanggan mengisi informasi pengiriman barang (alamat, pengiriman sehari atau 3 hari).
4. Sistem menampilkan informasi seluruh harga, termasuk pengiriman
5. Pelanggan mengisi informasi kartu kredit.
6. Sistem mengotorisasi pembelian
7. Sistem mengkonfirmasi penjualan secara langsung
8. Sistem mengirim email konfirmasi ke pelanggan

Ekstensi

3.a. Konsumen adalah langganan

1. Sistem menampilkan informasi pengiriman barang, harga, dan tagihan saat ini.
2. Pelanggan dapat menerima atau menghiraukan default ini, kembali ke skenario utama langkah 6.

6.a. Sistem gagal mengotorisasi pembelian kredit

1. Pelanggan dapat memasukkan kembali informasi kartu kredit atau membatalkan.

Latihan

Kasus 1

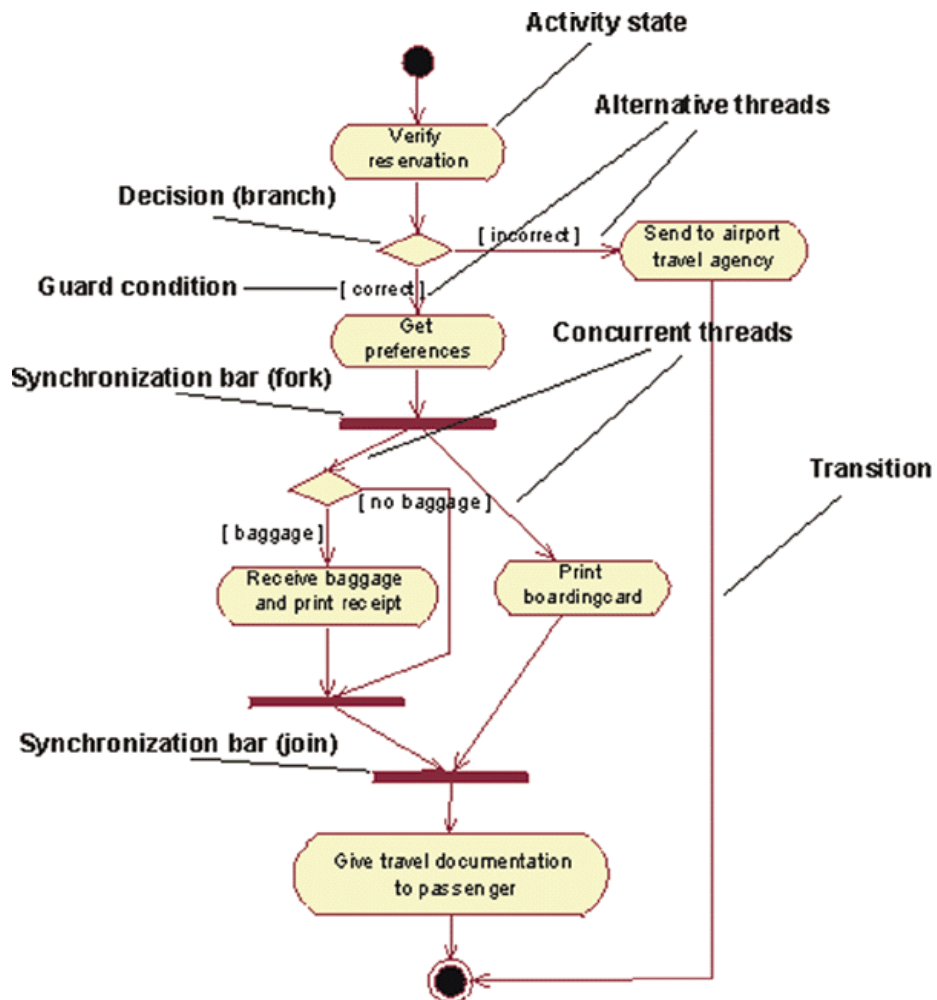
PT Aba yang bergerak dalam bidang perjalanan travel Madiun – Surabaya, Madiun – Jogjakarta, dan Madiun – Jakarta sedang membangun suatu sistem berbasis internet untuk melayani pemesanan tiket. Kegiatan-kegiatan yang dapat dilakukan melalui sistem adalah seperti berikut ini. Dari keterangan kegiatan tersebut, buatlah Use Case Diagramnya.

1. Pengguna bisa melakukan cek jadwal keberangkatan, melihat daftar tarif.
2. Pengguna memesan tiket secara online.
3. Pengguna mengkonfirmasi pembayaran melalui transfer, atau datang dengan membawa cash.
4. Petugas mencatat pemesanan tiket.
5. Pengguna bisa melihat daftar tempat duduk yang telah terisi.

[BAB 9] UML: ACTIVITY DIAGRAMS

Pada dasarnya *activity diagram* adalah suatu ilustrasi sederhana akan apa yang terjadi dalam aliran kerja, aktivitas apa yang dapat dilakukan secara paralel, dan apakah terdapat jalur alternatif dalam aliran kerja (Ericsson, 2004).

Berikut ini adalah contoh bentuk diagram dasar dalam *UML: Activity Diagram*. Dicontohkan pula suatu aktivitas diagram untuk kasus bisnis *airport check-in*.



Bentuk simbol diagram dapat dijelaskan sebagai berikut:

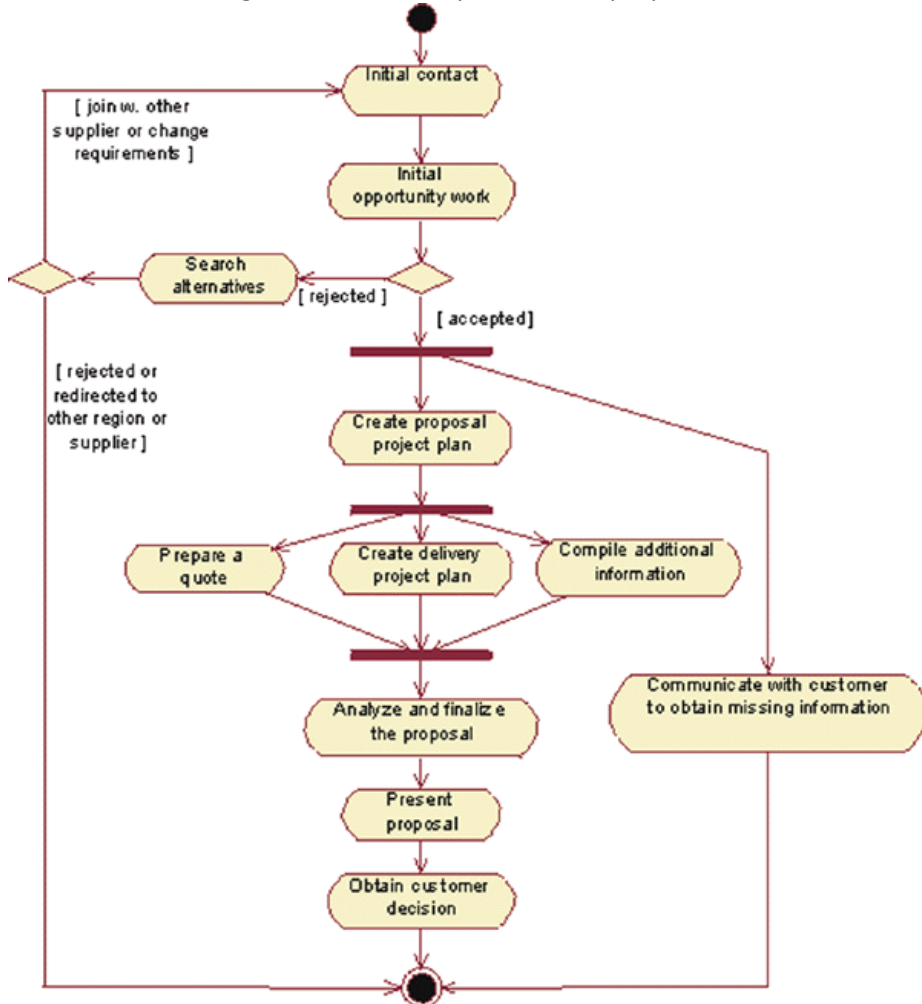
Activity states (pernyataan aktivitas), merepresentasikan bentuk atas langkah yang akan dilaksanakan dalam aliran kerja.

Transitions (perpindahan), menunjukkan pernyataan aktivitas yang terjadi setelah suatu kegiatan tertentu.

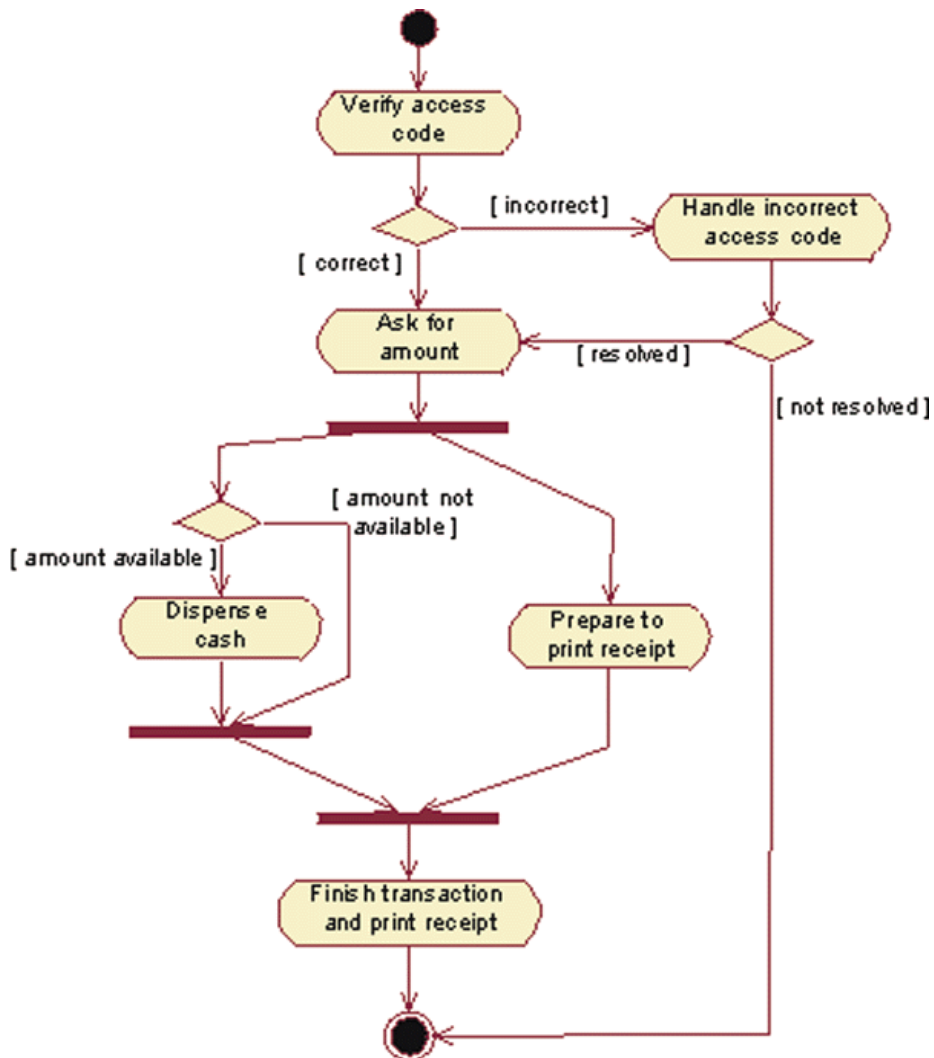
Decisions (keputusan), merupakan suatu set pengendali (*a set of guard*) atas suatu kondisi yang ditentukan.

Synchronization bars (garis sinkronisasi). Dapat digunakan untuk menunjukkan sub aliran secara paralel.

Contoh aktivitas diagram untuk kasus pemrosesan proposal



Contoh aktivitas diagram dalam pengambilan uang di ATM



Pertanyaan:

1. Perhatikan diagram aktivitas dalam pengambilan uang di ATM. Kemudian ceritakan proses tersebut dalam bentuk tekstual.
2. Berikut ini adalah kasus singkat dalam sistem pengajuan pengeluaran kas berdasarkan anggaran tahunan.
 - a. Kegiatan ini disebut dengan proses “pengajuan pengeluaran kas”
 - b. Aktivitas dimulai dari unit bisnis mengajukan permohonan pengeluaran kas, dan masuk dalam daftar antrian.

- c. Manajer keuangan mengecek apakah saldo anggaran untuk unit bisnis tersebut masih tersedia atau tidak, jika tersedia manajer keuangan akan mengotorisasi dan proses dilanjutkan ke direktur, jika kurang atau sudah habis maka pengajuan dikembalikan ke unit bisnis untuk direvisi atau dibatalkan.
- d. Direktur memberikan otorisasi yang akan masuk ke daftar permohonan pada direktur.
- e. Direktur mempertimbangkan. Jika setuju maka akan memberi otorisasi, jika tidak maka akan membatalkan permohonan tersebut dan memberikan keterangan pada kotak memo.
- f. Permohonan yang disetujui direktur akan masuk dalam daftar permohonan pengeluaran kas disetujui di bagian administrasi.
- g. Bagian administrasi mencetak nota persetujuan.
- h. Unit bisnis mengambil nota persetujuan tersebut untuk ditunjukkan dan dicairkan di bagian kasir.
- i. Proses selesai.

Dari aliran kegiatan yang disebutkan, buatlah diagram aktivitas permohonan pengeluarannya kas.

3. Suatu Departemen Store memiliki proses layanan penjualan barang konsinyasi kepada pelanggan sebagai berikut.
 - a. Persetujuan barang dibeli pelanggan
 - b. Sales membuat faktur konsinyasi
 - c. Kegiatan pembayaran
 - i. Kegiatan A
 1. Konsumen menunjukkan nota ke kasir
 2. Kasir mengecek kecocokan harga
 3. Kasir menerima uang
 - ii. Kegiatan B
 1. Sales membawa barang ke kasir
 - d. Barang dan nota diserahkan kepada pembeli.

Referensi:

- Fowler Martin. 2005. *“UML Distilled Panduan Singkat Bahasa Pemodelan Objek Standar”* Edisi 3 dalam bahasa Indonesia. Andi: Yogyakarta
- Ericsson, Maria. 2004. *Activity Diagrams: What They Are and How to Use Them*. IBM Software Group.

[BAB 10] UML: CLASS DIAGRAMS

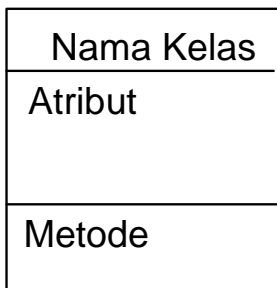
Fowler (2005) menyebutkan bahwa *class diagram* mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. UML menggunakan istilah fitur sebagai istilah umum yang meliputi properti dan operasi sebuah *class*.

Kuske et al. (2009) menyatakan bahwa *class diagrams* digunakan untuk merepresentasikan struktur statis (*static structure*) dari sistem berorientasi objek (*object-oriented systems*). Diagram kelas ini merupakan bagian dalam menjelaskan struktur-struktur objek dasar (*the fundamental object structures*).

A. Kelas (Class)

Sebuah *class* terdiri dari (Kuske et al., 2009):

1. Nama (*a name*)
2. Set atribut (*a set of attributes*)
3. Metode atau set operasi (*a set of operations*)

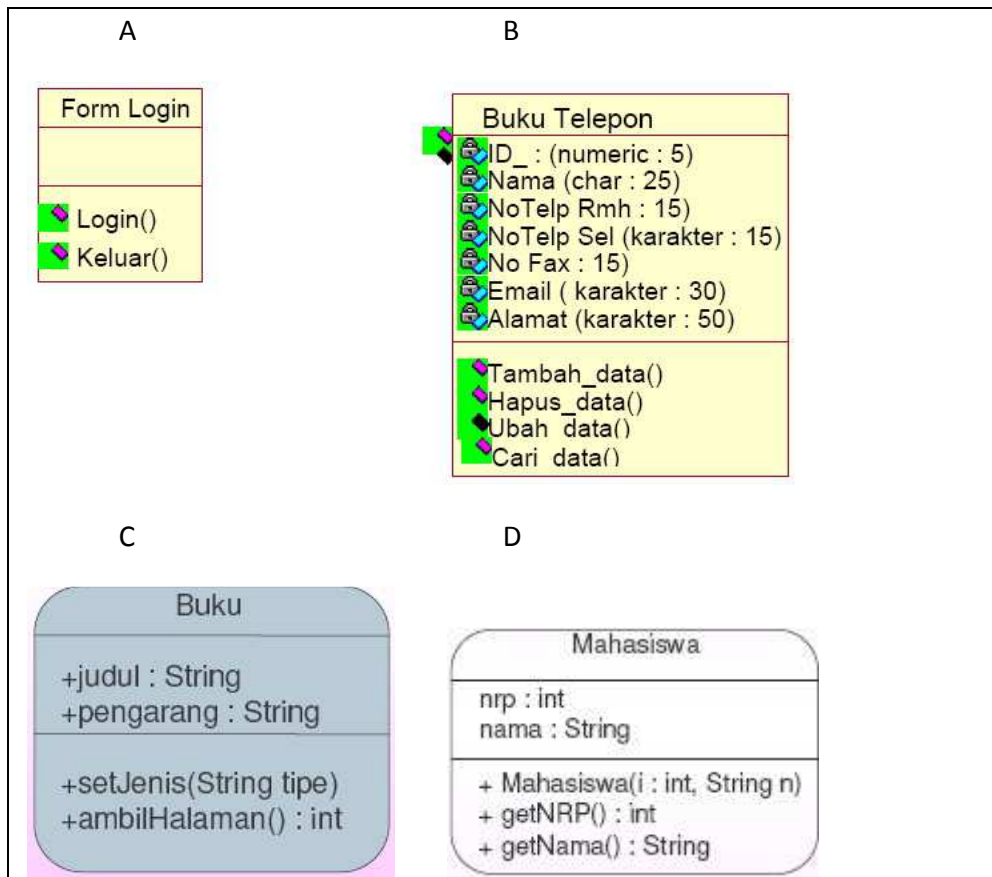


Gambar: Bentuk umum *Class Diagram*

Atribut dan metoda dalam *class diagram* dapat memiliki salah satu sifat seperti berikut di bawah ini :

- *Private*, hanya dapat diakses oleh *class* itu sendiri.
- *Protected*, hanya dapat diakses oleh *class* itu sendiri dan turunan dari *class* tersebut.
- *Public*, dapat diakses oleh *class* selain dari *class* yang bersangkutan.

Class dapat direpresentasikan dalam sebuah interface atau sebaliknya merupakan implementasi dari sebuah interface yang berupa kelas abstrak. Kelas abstrak tidak memiliki atribut namun hanya memiliki metode.

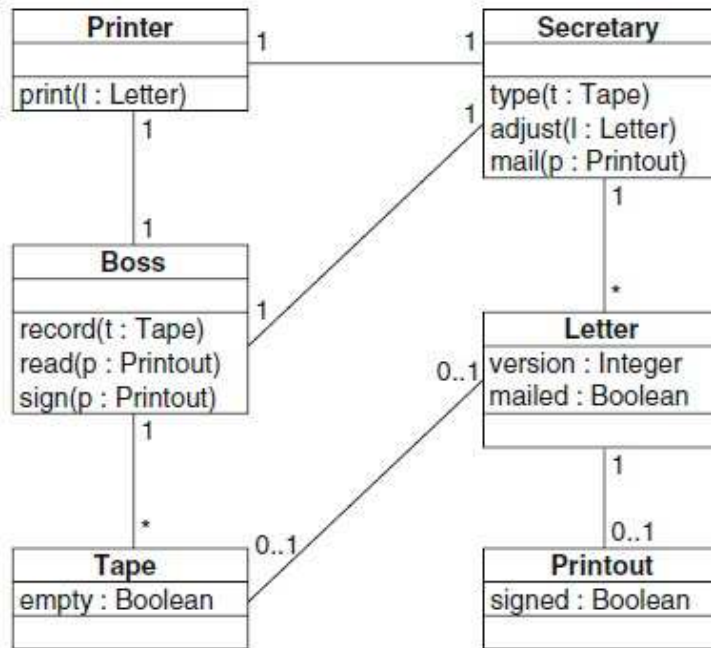


Gambar: contoh penggambaran klas

B. Diagram Kelas (Class Diagrams)

Suatu diagram kelas (*class diagram*) adalah suatu gambar dengan simpul-simpul (*nodes*) yang menunjukkan kelas-kelas dan gambaran asosiasi (*asociations*), perumuman (*generalizations*), atau ketergantungan (*dependencies*) (Kuske et al., 2009).

Dicontohkan sebuah diagram kelas yang terdiri dari asosiasi biner kelas.



Gambar: Diagram kelas dan asosiasinya (Sumber: Kuske et al., 2009)

Terdapat kelas bernama *Printer*, *Secretary*, *Boss*, *Letter*, *Tape*, dan *Printout*. Beberapa kelas berisi **operasi** yang menjelaskan aksi yang dapat dilakukan suatu objek kelas. Misal seorang sekretaris dapat mengetik surat yang disimpan dalam media penyimpanan atau mengeprintnya. Beberapa kelas lain menggunakan **atribut**, contoh media penyimpan dapat kosong (*empty*) atau berisi yang diindikasikan dengan nilai *boolean*. **Asosiasi** kelas dapat digambarkan dengan angka yang tertera pada garis hubungan, misal satu printer dapat digunakan oleh seorang sekretaris dan seorang sekretaris dapat menggunakan satu printer. Dapat dilihat pula bahwa seorang boss dapat menggunakan beberapa media penyimpanan (*tapes*) yang diwakili dengan tanda (*).

Referensi:

- Fowler Martin. 2005. "UML Distilled Panduan Singkat Bahasa Pemodelan Objek Standar" Edisi 3 dalam bahasa Indonesia. Andi: Yogyakarta
- Kuske, Sabine; Gogolla, Martin; Kreowski, Hans-Jörg; & Ziemann, Paul. 2009. Towards an integrated graph-based semantics for UML. *Softw Syst Model* (2009) 8:403–422

[BAB 11] PROTOTYPING & PENGEMBANGAN APLIKASI CEPAT

A. PROTOTYPING

Prototyping sistem informasi adalah suatu teknik yang sangat berguna untuk mengumpulkan informasi tertentu mengenai syarat-syarat informasi pengguna secara cepat. Umumnya prototyping yang efektif dilakukan pada awal siklus pengembangan sistem, yakni pada fase penetapan syarat-syarat. Dengan prototyping penganalisis sistem berusaha memperoleh reaksi awal dari pengguna atau manajemen, saran-saran, sehingga memungkinkan dilakukan inovasi, rencana revisi.

1. Jenis informasi yang dicari saat melakukan prototyping

Reaksi awal dari pengguna

- Reaksi saat bekerja dengan prototype
- Apakah fitur-fitur sudah sesuai dengan kebutuhan.
- Reaksi dikumpulkan dalam lembar observasi, wawancara, dan umpan balik.
- Penganalisis berusaha menemukan apakah pengguna merasa senang atau kesulitan.

2. Saran-saran dari pengguna

- Saran-saran diperoleh dari pengalaman saat bekerja dengan prototipe selama periode waktu tertentu.
- Saran-saran memberi petunjuk untuk menganalisis cara-cara memperbaiki, mengubah, atau menghentikan prototipe.

3. Inovasi

- Inovasi adalah kemampuan-kemampuan sistem baru yang tidak dianggap berhubungan saat pengguna mulai berinteraksi dengan prototipe.
- Inovasi memberi nilai tambah terhadap fitur-fitur.

4. Rencana Revisi

- Rencana revisi membantu mengidentifikasi prioritas yang diprototipekan selanjutnya.
- Informasi yang terkumpul memungkinkan penganalisis memberi pengarahannya kembali rencana tersebut dengan lebih efisien.

B. Jenis-Jenis Prototipe

Prototipe Patched-up

Berkaitan dengan penyusunan sistem yang bekerja namun patch atau patch bersama sama. Contoh dalam sistem informasi adalah model kerja yang memiliki semua fitur yang diperlukan namun tidak efisien. Program-programnya ditulis secara cepat dengan tujuan agar bisa berfungsi. Contoh lain adalah suatu SI yang memiliki semua prototipe yang diajukan, tetapi menjadi model dasar yang sebenarnya akan ditingkatkan.

Prototipe Non-Operasional

Prototipe dari model skala nganggur yang disusun untuk menguji beberapa rancangan tertentu. Contoh adalah model skala penuh untuk mobil yang digunakan dalam ujicoba terowongan angin, tetapi mobil tidak bisa dioperasikan. Model skala nganggur dipertimbangkan apabila aplikasi terlalu mahal untuk diprototipekan.

Prototipe First of Series

Prototyping dengan melibatkan penciptaan suatu model skala lengkap pertama dari sistem, yang disebut pilot. Prototipe benar-benar bisa dioperasikan dan direalisasikan dengan fitur-fiturnya identik.

Contoh:

- Prototipe pesawat terbang dengan fitur-fitur identik.
- Instalasi perbankan untuk penransferan dana, pertama dipasang di dua lokasi, bila berhasil dipasang duplikatnya di semua lokasi.

Prototipe Fitur-fitur terpilih

Pembangunan suatu model operasional yang mencakup beberapa, tetapi tidak semua, fitur-fitur yang dimiliki sistem final.

Contoh:

Sebuah sistem menu bisa muncul fitur penambahan, pembaharuan, penghapusan, pembuatan daftar, pencarian. Pada protitipe hanya tiga yang bisa digunakan, yaitu penambahan, penghapusan, dan pembuatan daftar.

Faktor penentu apakah sebuah sistem lebih sesuai atau kurang untuk prototyping

Kurang sesuai	<-- Kondisi -->	Lebih Sesuai
Beberapa kali sebelum	Pengalaman design	Sedikit sebelum
Khusus dan stabil	Lingkungan	Tidak khusus dan tidak stabil
Terstruktur	Pembuatan Keputusan	Tidak terstruktur Dan semiterstruktur

Petunjuk untuk mengembangkan sebuah prototipe

- Bekerja sesuai modul
- Membangun prototipe dengan cepat
- Memodifikasi prototipe
- Menekankan antarmuka pengguna

C. Kelemahan dan Kelebihan

1. KELEMAHAN PROTOTYPING

- Mengelola proyek, pepanjangan prototipe secara tak terbatas menciptakan masalah tersendiri.
- Memakai sistem yang belum selesai seolah-olah sebagai sistem yang selesai.

2. KELEBIHAN-KELEBIHAN PROTOTIPE

- Mengubah sistem sejak dini dalam masa perkembangan
- Membatalkan sistem yang tidak diharapkan.
- Merancang sebuah sistem yang sesuai dengan kebutuhan dan harapan pengguna.

D. PAC (PENGEMBANGAN APLIKASI CEPAT)

PAC (Rapid Application Development) adalah pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak.

FASE-FASE PAC

- Fase Perencanaan Syarat
- Workshop Desain PAC
- Fase Implementasi

PROJECT 1: Membangun sistem penerimaan pendapatan

A. Keterangan Umum

Sistem ini dirancang untuk menangani proses penerimaan pendapatan yang dilakukan oleh kasir pendapatan.

B. Administrasi User

User perlu teregister dan memiliki password.

User yang teregister perlu mendapat hak akses pada sesi tertentu dalam sistem.

Hak akses tersebut terdiri dari:

- Kasir
- Supervisor
- Bagian Akuntansi

Dengan demikian perlu dibuat dua tabel database yaitu **user**: user_ide, user_nama, user_password, user_approve

Kemudian tabel database **user_akses**: id, user_id, akses.

Buatlah: penjelasan mengenai administrasi user kemudian buatlah database relationship-nya bermodel visio.

C. Interaksi Pengguna Sistem

Kasir

- Melakukan input pendapatan
- Mengeprint kuitansi
- Membuat laporan harian kertas

Supervisor

- Melihat laporan harian

Bagian Akuntansi

- Melihat laporan harian

Buatlah: penjelasan interaksi pengguna sistem dan buat pula diagram use case-nya.

D. Pengoperasian Penerimaan Kas

Halaman login

Halaman utama kasir: meliputi pula menu transaksi penerimaan, transaksi pengeluaran (setoran), dan pelaporan.

Transaksi penerimaan akan dimulai dengan membuka formulir input transaksi penerimaan. Pada halaman input ini terdapat pula dua fungsi yaitu proses, batal dan kembali ke utama.

Setelah proses selesai dilaksanakan maka layar akan kembali ke halaman formulir input dan menunggu perintah selanjutnya.

Buat class diagram dan penjelasannya.