

ANALISIS DAN PERANCANGAN SISTEM

Perspektif Bidang Akuntansi

Edisi 1
Tahun 2017

Agustinus MUJILAN

Sistem memiliki berbagai bentuk baik sistem manual maupun terkomputerisasi. Namun pada intinya, setiap pengembangan sistem memerlukan suatu analisis dan perancangan sehingga sistem yang dimaksud dapat berjalan dengan baik. Pengembang sistem perlu memahami metode-metode dalam pengembangan sistem, baik keunggulan maupun kelemahannya. Tidak kalah penting bagi pengelola organisasi/perusahaan/instansi maupun para profesi memerlukan pemahaman tentang metode pengembangan sistem sehingga dapat memilih metode yang paling efektif dan efisien, mendukung dalam penyediaan kebutuhan baik informasi maupun fasilitas, serta mendapat gambaran tentang alokasi sumber daya yang dibutuhkan.

Metode konvensional dan umum digunakan adalah *system life development cycle* (SDLC) dan metode-metode pengembangan alternatif lainnya. Mempelajari metode SDLC tidak hanya bermanfaat untuk pengembangan sistem berbasis teknologi, tapi juga dapat mendukung dalam mengembangkan sistem-sistem pengelolaan yang bersifat manual.

Demikian pula dalam buku ini dijelaskan beberapa cara perancangan menggunakan bahasa pemodelan antara lain: *entity relationship diagram*, *data flow diagram*, *UML use case*, *UML activity diagram*, dan *UML class diagram*. Diagram-diagram tersebut perlu dipahami bagi para perancang namun juga perlu diketahui oleh pihak pengelola organisasi, sehingga dapat mengetahui bentuk komunikasi dalam bentuk diagram.

ISBN: 978-602-61347-0-7



Penerbit:

Fakultas Ekonomi dan Bisnis
Universitas Katolik Widya Mandala Madiun
Jl. Manggis 15-17, Madiun 63131
<http://feb.widyamandala.ac.id>
feb@widyamandala.ac.id

FEB
UKMMI

Analisis dan Perancangan Sistem

Mujilan



Penerbit
Fakultas Ekonomi dan Bisnis
Universitas Katolik Widya Mandala Madiun

ANALISIS DAN PERANCANGAN SISTEM

Perspektif Bidang Akuntansi

Agustinus Mujilan



Penerbit Fakultas Ekonomi dan Bisnis Universitas Katolik Widya
Mandala Madiun

Judul: ANALISIS DAN PERANCANGAN SISTEM
Perspektif Bidang Akuntansi

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

Hak Cipta © 2017 pada penulis.

Hak cipta dilindungi undang-undang. Dilarang memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanik, termasuk memfotokopi, merekam, atau dengan menggunakan sistem penyimpanan lainnya, tanpa izin tertulis dari Penerbit.



**Penerbit Fakultas Ekonomi dan Bisnis Universitas Katolik
Widya Mandala Madiun**

Jl. Manggis 15-17, Madiun, Jawa Timur, Indonesia

Telp. : (0351) 453328

Website : <http://feb.widyamandala.ac.id>

Email : feb@widyamandala.ac.id

ISBN: 978-602-61347-0-7

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 agusmujilan@gmail.com. Terima kasih.

Madiun, 14 Maret 2017

Agustinus Mujilan, S.E., M.Sc.

Daftar Isi

KATA PENGANTAR	iii
Daftar Isi	v
BAB 1: PEMAHAMAN DASAR ANALISIS DAN PERANCANGAN SISTEM	1
A. Mengapa Dibutuhkan Pengembangan Sistem.....	1
B. Seputar Analisis dan Perancangan Sistem	2
C. Filosofi Analisis dan Perancangan Sistem	5
D. Aspek Kesuksesan Pengembangan Sistem	7
E. Efek Kepuasan Pengguna dalam Pengembangan .	9
Pertanyaan	10
Kasus.....	10
BAB 2: METODE PENGEMBANGAN SISTEM TEKNOLOGI INFORMASI	15
A. Pendahuluan	15
B. Konsep <i>Waterfall</i>	17
C. Metode Pengembangan Sistem Konvensional vs Alternatif	18
D. Metode SDLC.....	19
E. Metode Pengembangan Evolusioner.....	22
F. Metode Pengembangan Berorientasi Pemakaian Ulang (<i>Re-Usable</i>)	23
Kasus.....	24
BAB 3: Memahami Latar Belakang Organisasi	27
A. Pemahaman Awal	27
B. <i>Core Bisnis</i>	28
C. Tujuan Umum Organisasi.....	30
D. Strategi Organisasi	31
E. Struktur Organisasi	33
F. Budaya Organisasi.....	36
Pertanyaan	39

BAB 4: BAHASA PEMODELAN SISTEM	42
A. Mengapa perlu bahasa pemodelan?	42
B. Diagram Sistem	43
C. Unified Modeling Language (UML)	49
Pertanyaan	52
BAB 5: PEMODELAN DATABASE.....	54
A. Entity-Relationship (E-R) Diagram.....	54
B. Simbol Entity Relationship Diagram.....	58
C. PROGRAM APLIKASI	65
D. ERD Menggunakan Aplikasi MsVisio	66
Soal dan Latihan	68
BAB 6. DATA FLOW DIAGRAM (DFD).....	70
Pertanyaan	74
BAB 7. <i>UML: Use Case Diagram</i>	75
Latihan.....	79
BAB 8. <i>UML: ACTIVITY DIAGRAMS</i>	81
Soal/Latihan:	84
BAB 9. <i>UML: CLASS DIAGRAMS</i>	87
A. Definisi dan Konsep Klas	87
B. Aplikasi Diagram Klas (<i>Class Diagrams</i>)	91
Latihan Aplikasi	92
BAB 10. PROTOTYPING.....	95
A. <i>Prototyping</i>	95
B. Jenis-Jenis Prototipe	96
C. Kelemahan dan Kelebihan	99
Pertanyaan	100
DAFTAR PUSTAKA	101

BAB 1: PEMAHAMAN DASAR ANALISIS DAN PERANCANGAN SISTEM

Tujuan

1. Menjelaskan mengapa dibutuhkan analisis dan perancangan sistem.
2. Memahami Filosofi dalam analisis dan perancangan sistem.
3. Memahami aspek kesuksesan dalam pengembangan sistem.

A. Mengapa Dibutuhkan Pengembangan Sistem

Pada dunia usaha yang semakin kompetitif, keunggulan sekecil apapun akan mempengaruhi posisi perusahaan dalam jangka panjang (Al Fatta, 2007). Lebih lanjut dicontohkan, sebuah toko material yang terlalu lama menjawab berapa stok semen yang dimiliki ketika seorang konsumen membutuhkannya, maka akan ditinggalkan karena pelanggan menginginkan layanan yang cepat. Sebuah toko buku yang tidak dapat membantu secara cepat buku yang dicari pembeli dalam suatu tumpukan buku yang jumlahnya banyak, maka akan mengurangi kepuasan calon pembeli tersebut, dan lama-lama akan ditinggalkan pula.

Dari skala pengelolaan, ketika skala operasi perusahaan/organisasi meningkat dan pekerja semakin bertambah banyak (ratusan, bahkan ribuan),

maka manajer tidak dapat mengawasi kegiatan secara langsung, manajer akan lebih mengandalkan dan menggunakan informasi yang diterimanya. Peran informasi tersebut menjadi penting, dan manajer tidak dapat begitu saja mempercayai informasi yang dihasilkan jika tidak dapat memastikan bahwa data yang dimasukkan adalah benar dan akurat.

Dari skala pelanggan, peningkatan kegiatan dan peningkatan jumlah pelanggan juga mengakibatkan banyaknya layanan yang harus dikerjakan. Jika tidak dapat dikelola secara baik, antrian layanan akan semakin padat. Kepadatan antrian dan lambatnya respon dapat menurunkan tingkat kepuasan pelanggan, yang justru lama kelamaan pelanggan akan meninggalkan perusahaan.

Atas dasar contoh-contoh di atas, maka perusahaan/organisasi harus berbenah dan memperbaiki sistemnya sehingga perkembangan yang terjadi dapat diimbangi dengan makin baiknya layanan dan kemudahan dalam pengelolaan kegiatannya. Perusahaan/organisasi yang tanggap pada kondisi saat ini serta rencana pengembangan di masa yang akan datang tentu akan merespon pengembangan sistem dan melakukan pengembangan sistem secara terus menerus.

B. Seputar Analisis dan Perancangan Sistem

Pengembangan sistem (*system development*) akan selalu muncul pada organisasi yang masih hidup. Setiap organisasi tentu selalu mengupayakan

perbaikan proses bisnisnya. Untuk mencapai kesuksesan dalam perbaikan proses bisnis (tata kelola) itu kuncinya adalah pada pada pilihan: menyesuaikan sistem yang ada, merubah sistem yang ada, membuat sistem baru. Ketika organisasi merubah sistem itulah peran pengembangan sistem sangat penting. Pengembangan sistem dapat berupa membangun baru, merevisi sistem yang telah ada, atau merombak sistem yang telah ada. Berbagai alternatif pilihan tersebut perlu dikaji secara seksama agar pembenahan sistem yang dilakukan dapat berjalan dengan baik dan membawa dampak ke sistem yang lebih efektif dan efisien bukan sebaliknya justru membawa kehancuran, kerusakan, dan gangguan pada sistem yang berakibat malah menjadi buruknya proses bisnis.

Dalam proses pengembangan sistem yang berbasis teknologi informasi terdapat tiga komponen kunci yaitu analisis sistem, disain sistem /perancangan sistem, dan penulisan program. Analisis sistem merupakan kegiatan pendahuluan sebelum beranjak pada tahap pengembangan sistem selanjutnya. Maka, analisis sistem sangat dekat kaitannya dengan perancangan sistem dan selanjutnya adalah penulisan program untuk aplikasi sistem informasi. 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.

Mengapa sistem perlu dianalisis dan dirancang secara baik? Tentu saja agar sistem tepat sasaran dan berkualitas baik. Berikut disebutkan beberapa kriteria sistem yang baik:

1. Memenuhi kebutuhan pengguna
2. Performance baik
3. Reliabilitas baik
4. Kegunaan tinggi
5. Terjadwal secara baik
6. Terdapat rencana anggaran yang baik
7. Operasional tidak over anggaran atau over jadwal.

C. 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.

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

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 analisis.

Pertama, 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.

Ketiga, 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 terliabat, 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 analis, teknis menghitung biaya dan manfaat, dan sebagainya.

D. Aspek Kesuksesan Pengembangan Sistem

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. Dapat dimisalkan hal-hal yang perlu diperhatikan oleh analis dan pengembang sistem:

1. Apakah sistem yang dirancang sudah sesuai dengan proses bisnis, struktur organisasi, dan kebutuhan organisasi.
2. Apakah sistem yang dirancang dapat dioperasikan atau dijalankan secara mudah oleh penggunanya.
3. Apakah sistem yang dirancang akan menghasilkan informasi yang berguna.
4. Apakah sistem yang dirancang membawa efektifitas dan efisiensi.
5. Apakah sistem yang dirancang dapat dipelihara dengan mudah.
6. Apakah pengguna sistem memiliki kemampuan cukup untuk menjalankan sistem.

7. Apakah sistem yang dirancang tersebut dapat dikembangkan dengan lebih mudah di masa mendatang; dan sebagainya.

E. 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.

Pertanyaan

1. Dalam situasi yang bagaimana organisasi perlu mengembangkan sistemnya?
2. Bagaimana kaitan antara analisis, perancangan, dan penulisan program dalam pengembangan sistem berbasis teknologi informasi?
3. Sebutkan dan jelaskan aspek kesuksesan dalam pengembangan sistem?
4. Mengapa kepuasan pengguna perlu diperhatikan dalam pengembangan sistem?

Kasus

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 keluasan 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: METODE PENGEMBANGAN SISTEM TEKNOLOGI INFORMASI

Tujuan

1. Memahami konsep waterfall
2. Memahami adanya metode-metode dalam pengembangan sistem.
3. Memahami SDLC dan tahapannya.
4. Mengetahui metode pengembangan sistem alternatif atau evolusioner.

A. Pendahuluan

Sistem informasi (SI) semestinya dapat mendukung kebutuhan manajemen dalam menyediakan informasi yang akurat, sehingga sistem informasi perlu dirajut/dibentuk (*tailored*) 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 efisiensi 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. Konsep *Waterfall*

Waterfall dianalogikan seperti air terjun. Air akan mengalir dalam tahapan demi tahapan sampai akhirnya sampai tujuan. Masing-masing tahapan akan dilalui dan dapat terjadi jika tahapan sebelumnya telah dilampaui. Begitulah konsep air.

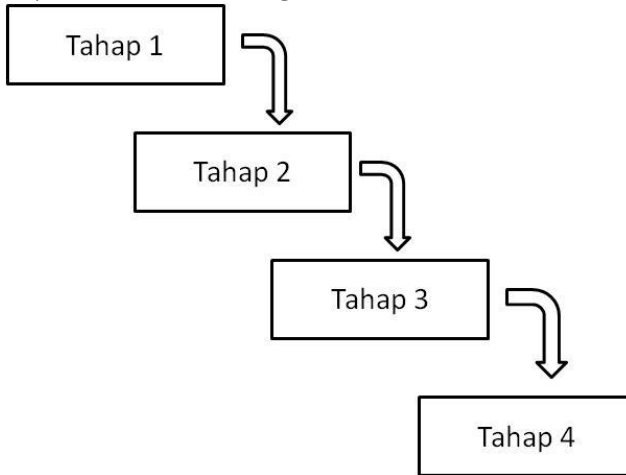


Gambar air terjun

Sumber: yogya-backpacker.com

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 *waterfall*

C. Metode Pengembangan Sistem Konvensional vs Alternatif

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*), *outsourcing*, metode pengembangan evolusioner, dan pengembangan berorientasi pemakaian ulang.

D. 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
2. Studi pendahuluan
3. Studi kelayakan
4. Mengidentifikasi permasalahan dan kebutuhan pemakai
5. Memahami sistem yang ada
6. Menganalisis hasil penelitian.
7. Perancangan sistem
8. Perancangan awal
9. Perancangan rinci

10. Implementasi sistem
11. 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.

Kelemahan SDLC

Namun demikian, terdapat pula kelemahan dalam metode SDLC (Al Fatta, 2007), antara lain:

1. Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses;
2. Dapat mengakibatkan sulitnya merespon perubahan kebutuhan pengguna
3. Model SDLC harus digunakan hanya ketika persyaratan dipahami dengan baik.

E. Metode Pengembangan Evolusioner

Metode pengembangan evolusioner diawali dengan pengembangan implementasi awal, memperlihatkan sistem awal kepada pengguna untuk dikomentari, kemudian memperbaiki versi demi versi sampai

sistem yang memenuhi persyaratan diperoleh (Al Fatta, 2007). Pengembangan evolusioner memiliki dua variasi: (1) pengembangan eksploratori, ditujukan untuk bekerja dengan pelanggan dalam menyelidiki persyaratan dan pengiriman sistem akhir. (2) Prototipe yang dapat dibuang (*throw-away-prototype*), berkonsentrasi pada eksperimen dengan pelanggan yang tidak dipahami dengan baik, cocok untuk jenis produk generik.

F. Metode Pengembangan Berorientasi Pemakaian Ulang (*Re-Usable*)

Permintaan dari satu klien dengan klien yang lain bisa jadi hampir sama dalam beberapa proses bisnis. Jika sudah terdapat suatu sistem untuk klien lain yang hampir sama proses bisnisnya tersebut, maka piranti lunak yang sudah ada dapat digunakan untuk membuat piranti lunak yang baru.

Fase pengembangan *re-usable* adalah sebagai berikut (Al Fatta, 2007):

1. Analisis komponen: spesifikasi persyaratan telah diketahui, mencari komponen-komponen sesuai spesifikasi,
2. Modifikasi persyaratan: komponen dimodifikasi untuk merefleksikan komponen yang dibutuhkan, jika modifikasi tidak dapat dilakukan maka dicari solusi alternatif.
3. Perancangan sistem dengan pemakaian ulang. Kerangka kerja sistem dirancang atau yang telah ada dipakai ulang.

4. Pengembangan dan integrasi. Piranti lunak yang tidak dapat dibeli akan dikembangkan dan komponen diintegrasikan ke dalam sistem.

Kasus

Konsultasi Awal CV Eksan

CV Eksan bergerak di bidang penjualan peralatan komputer. Usahanya telah berkembang dan telah mencapai lebih dari 5.000 pelanggan. Rata-rata per hari terdapat 70 transaksi meliputi penjualan komputer, asesoris, printer, tinta printer, dan sebagainya. Manajer CV Eksan, Andika merasakan beberapa bulan ini terjadi penurunan penjualan, namun ia tidak dapat memastikan berapa penurunan yang terjadi.

Andika menyampaikan permasalahan ini pada Ruby konsultan. Dalam suatu pertemuan awal terdapat pembicaraan sebagai berikut.

Ruby: “Baik tuan Andika. Saya telah sedikit memahami apa yang Anda maksudkan. Berapakah karyawan yang menangani penjualan saat ini?”

Andika: “Penjualan di konter ditangani oleh seorang selles wanita, Muti, ia mencatat penjualan dan menerima uang. Selain itu tugas utamanya dibantu oleh Wawan untuk mengambil barang atau mengecek fungsi barang.”

Ruby: “Apakah Anda memiliki daftar para pelanggan?”

Andika: “Kami hanya menghafal pelanggan kami yang sering datang, namun kami tidak memiliki catatan pasti. Dari mereka ada yang sering berkunjung, beberapa kali berkunjung, atau yang jarang berkunjung.”

Ruby: “Apakah tuan Andika menyadari bahwa di kota ini terdapat agen lain yang bergerak di bidang sama?”

Andika: “Betul, kami tahu itu. Ia memiliki *showroom* yang bagus. Pengunjung pun dapat mengecek persediaan dan harga secara mandiri melalui komputer yang disediakan. Bahkan ia membuat website yang dapat diakses di manapun untuk mencari informasi. Mungkin beberapa pelanggan kami juga berpindah ke tempat tersebut.”

Ruby: “Baiklah tuan Andika, kami bersedia membantu. Kami akan mempersiapkan usulan kami untuk menangani masalah di CV Anda. Nanti kami akan mengirim personal kami untuk memahami lebih jauh keperluan dan kondisi di tempat Anda.”

Andika: “Terima kasih pak Ruby, kami akan menunggu dan memberikan informasi yang diperlukan.”

Dari kasus singkat di atas:

1. Identifikasikan permasalahan, perkiraan alternatif sistem, dan analisis singkat tentang kondisi CV Eksan serta usulan Anda atas sistem.

2. Menurut Anda sudah tepatkah keputusan Andika mendatangi Ruby konsultan untuk menyampaikan permasalahan usahanya?

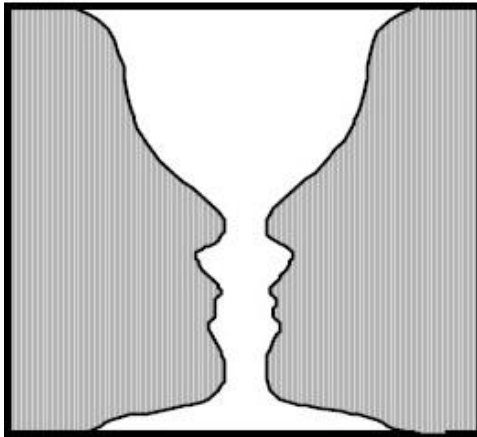
BAB 3: Memahami Latar Belakang Organisasi

Tujuan

1. Mengenali tujuan organisasi
2. Memahami *core business* dan kaitannya dengan pengembangan sistem.
3. Mengaitkan kebutuhan sistem dengan struktur organisasi.
4. Memahami budaya organisasi dan kaitannya dengan pengembangan sistem.

A. Pemahaman Awal

Dalam memandang suatu permasalahan, seseorang dapat memiliki persepsi masing-masing. Persepsi ini dapat berbeda antara satu dengan yang lain sehingga mengakibatkan tidak sejalannya keputusan yang diambil.



Gambar: persepsi

Sebagai contoh Anda dapat melihat dengan seksama gambar apa yang terlihat di atas. Masing-masing sudut pandang akan memberikan hasil yang berbeda dalam menterjemahkan gambar tersebut. Aplikasinya, permasalahan yang sama dapat dilihat dari persepsi biaya dan manfaat, hal lain adalah untung dan ruginya, positif atau negatif, peluang dan ancamannya.

Demikian juga, ketika kita ingin melakukan pengembangan sistem maka perlu memperhatikan dengan seksama hal-hal yang terkait dengan pengembangan tersebut. Untuk memberikan pemahaman kebutuhan dalam pengembangan sistem, maka pada awalnya perlu dilakukan analisis pendahuluan.

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

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

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.

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. Tujuan Umum Organisasi

Pada umumnya ketika didirikan, suatu organisasi atau perusahaan diharapkan untuk tetap dapat bertahan dan *survive*. Bertahan atau berada (*exist*) saja tidak cukup, tetapi organisasi/perusahaan tersebut perlu jaya/unggul (*survive*). Untuk dapat menjadi unggul 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).

D. 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.

E. Struktur Organisasi

Dalam melakukan desain 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: skop (*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.

F. 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 individu-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 ketaknyamanan 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.

Pertanyaan

1. Mengapa analis dan perancang sistem harus memahami latar belakang organisasi dalam pengembangan sistem?
2. Mengapa strategi organisasi perlu diseleraskan dengan pengembangan sistem?
3. Apa imbas adanya struktur organisasi yang telah ada pada sistem yang akan dibentuk?
4. Apa pengaruh budaya organisasi pada sistem yang akan dikembangkan?
5. Apa yang dimaksud dengan *core business*?

INFORMASI/PEMAHAMAN ORGANISASI

Nama Organisasi	
Alamat	

Hasil Analisis

1. Apa core bisnis organisasi?

2. Apa visi, misi organisasi?

-- berikan penjelasan mengenai visi dan misi organisasi --

3. Bagaimana struktur organisasi yang ada?

-- berikan penjelasan mengenai gambaran umum struktur organisasi --

4. Apa strategi Organisasi?

5. Bagaimana kondisi sistem informasi yang ada di organisasi?

-- berikan penjelasan keluasan jaringan, infrastruktur penting yang telah dimiliki organisasi --

6. Apa sistem informasi yang ingin dikembangkan pada organisasi?

-- berikan penjelasan pengembangan sistem yang direncanakan atau sistem yang ingin Anda jelaskan --

7. Unit apa saja yang terlibat dalam pengembangan sistem ini?

-- berikan keterangan mengenai unit-unit atau person yang terlibat atau dapat dimintai keterangan mengenai pengembangan sistem di dalam organisasi --

BAB 4: BAHASA PEMODELAN SISTEM

Tujuan

1. Memahami perlunya bahasa pemodelan dalam pengembangan sistem informasi berbasis teknologi.
2. Mengenali macam dan bentuk diagram sistem dalam perancangan sistem.
3. Mengetahui bahasa pemodelan *Unified Modeling Language* (UML).

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 diinterpretasikan, 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 harus didefinisikan arsitekturnya secara jelas, agar *bug* mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya.
- 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. Dapat dianalogikan bahwa suatu bangunan perlu dirancang terlebih dahulu sebelum dibangun: pondasi, rangka, tembok, atap, pintu, jendela, jalur listrik, jalur air, dan sebagainya. Apabila bangunan tidak dirancang terlebih dahulu sangat besar kemungkinan salah ukur, salah posisi, bahkan bisa roboh. Demikian pula ketika ingin mengembangkan gedung tersebut maka kekuatan dan rancangannya perlu dilihat kembali sehingga sesuai kekuatannya.

Sistem perlu dijelaskan secara mencukupi sehingga dapat dipahami oleh berbagai pihak terkait. Terdapat berbagai cara untuk menjelaskan sistem baik secara tekstual maupun secara grafis. Dikatakan cukup karena penjelasan baik tekstual maupun grafis tidak dapat mencakup secara sempurna dan menyeluruh melainkan mencukupi untuk dapat dipahami dan menjadi landasan yang baik dalam mengembangkan sistem.

B. Diagram Sistem

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

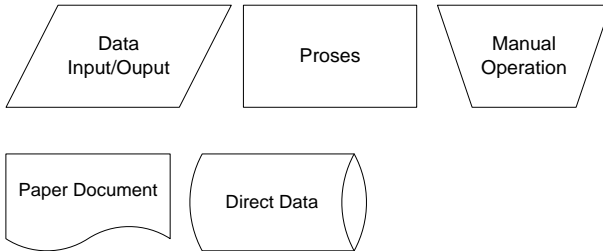
Banyak organisasi memberikan fokus pada usaha pendokumentasian sistem untuk memperoleh pemahaman akan peningkatan kekomplekan sistem informasi dan kontrol berdasar sistem yang terkait (Bradford et al., 2007). SD ini memegang peran penting dalam dokumentasi dengan memberikan tampilan secara grafis pada pengendalian internal, arus data, dan arus informasi yang terkait dengan proses pokok yang mendukung organisais. 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.

Contoh beberapa bentuk diagram *system flowchart*



Di kalangan akuntansi tentu sudah familiar dengan diagram model ini khususnya yang telah mempelajari sistem informasi akuntansi.

2. *Data Flow Diagrams (DFDs)*

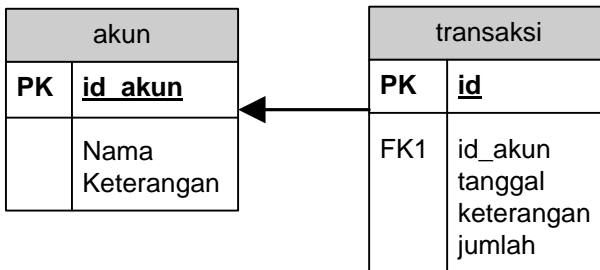
DFD berasal dari sistem informasi manajemen (SIM) dan digunakan untuk menggambarkan proses sistem, arus antar proses, dan sumber, tujuan, serta penyimpanan data. Dua tipe DFD adalah: *logical DFDs* yang berfokus pada aktifitas di dalam sistem, dan *physical DFDs* meliputi "who, where, and how" dari sistem.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

Contoh bentuk DFD



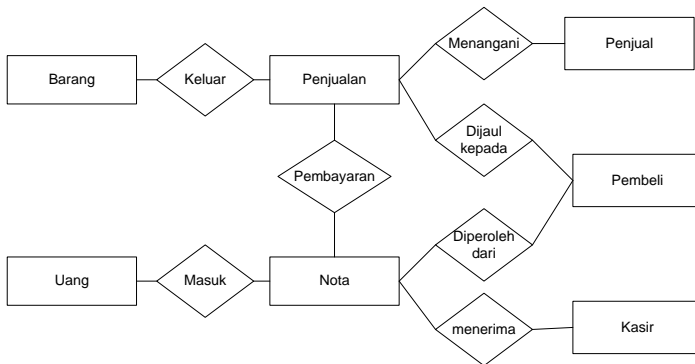
3. *Entity-Relationship Diagram (ERD)*

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*

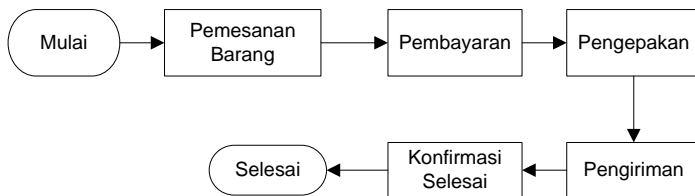
REA terdiri dari tiga komponen entitas: (1) sumber daya yang diperoleh dan digunakan organisasi, (2) kegiatan yang terkait dengan organisasi, dan (3) agen atau pihak yang berpartisipasi dalam kegiatan.



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.

Contoh:



6. *Unified Modeling Language (UML)*

UML merupakan 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. Unified berarti gabungan dari berbagai bahasa pemodelan berorientasi objek yang telah ada sebelumnya. Sehingga bentuk diagram yang tergabung dalam UML pun beragam jumlahnya. Pengenalan mengenai UML disampaikan bagian selanjutnya.

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. *Unified Modeling Language* (UML) merupakan 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 memelopori 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**, **keperilakuan** (*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, 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**.

Pertanyaan

1. Apa perlunya serta kegunaan dari bahasa pemodelan sistem?
2. Sebutkan beberapa bahasa pemodelan sistem?
3. Begitu banyaknya bahasa pemodelan sistem. Apakah seorang perancang harus menggunakan semua

diagram dalam suatu rancangan sistem? Mengapa demikian?

4. Apa yang dimaksud dengan methods wars sehingga memicu dibentuknya *unified modeling language*?
5. Sebutkan minimal 3 bahasa pemodelan di bawah UML dan apa kegunaannya?

BAB 5: PEMODELAN DATABASE

Tujuan

1. Mengenali cara pemodelan database
2. Mempraktikkan cara menyusun model ERD
3. Mengetahui piranti lunak penyusunan ERD.

A. Entity-Relationship (E-R) Diagram

E-R diagram (ERD) 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


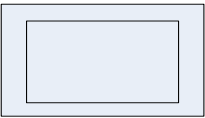
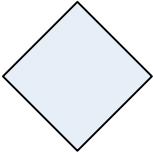
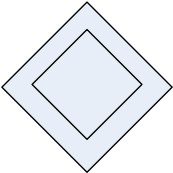
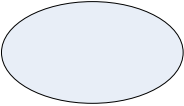

Informasi seputar entitas dalam set-entitas kemudian dapat diperlihatkan. Entitas direpresentasikan oleh nilai dari *primary key*.

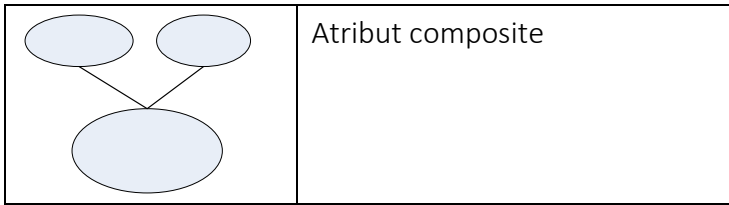
Dalam suatu hubungan antar entitas dimungkinkan terjadinya satu entitas terkait dengan beberapa entitas yang dapat disimbolkan dengan $1:n$. Angka n misalnya dapat kita contohkan n ($= 0, 1, 2, \dots$).

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
	entity
	weak entity
	relationship
	Identifying relationship
	Atribut
	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: contoh bentuk relationship

3. Atribut

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

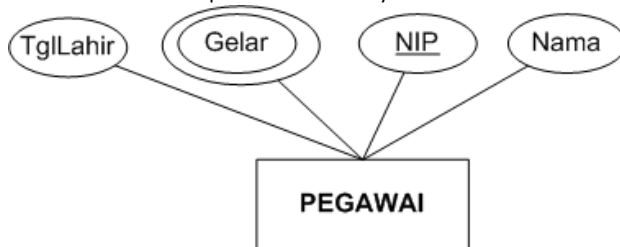
Nilai atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

Jenis-jenis atribut

Key: atribut yang digunakan untuk menentukan suatu entity secara unik.

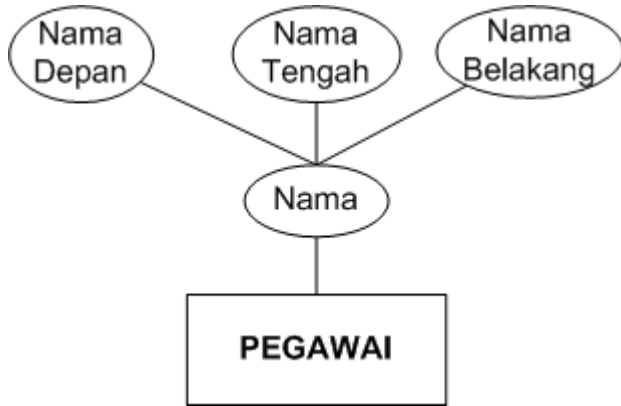
Atribut Simple: atribut yang bernilai tunggal.

Atribut Multivalued: atribut yang memiliki sekelompok nilai untuk setiap instan entity.



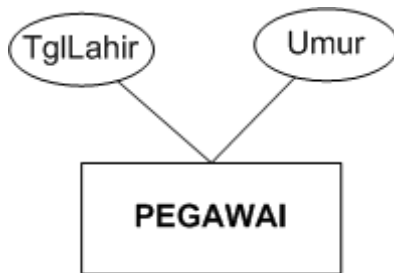
Gambar: contoh penulisan atribut (key, simple, multivalued)

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



Gambar: contoh atribut composite

Atribut Derivatif: suatu atribut yang dihasilkan dari atribut lain.



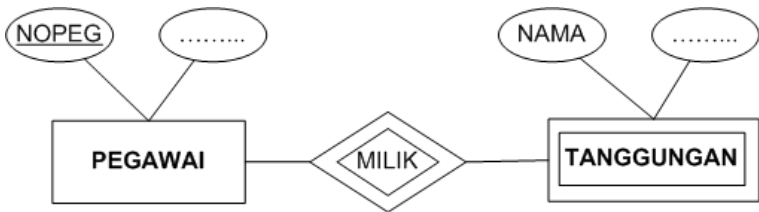
Gambar: contoh atribut derivatif

4. *Weak Entity*

Weak Entity adalah suatu *Entity* dimana keberadaan dari entitas tersebut tergantung dari keberadaan entitas lain.

Entitas yang merupakan induknya disebut *identifying owner* dan relationshipnya disebut *identifying relationship*.

Weak entity selalu mempunyai *total participation constraint* dengan *identifying owner*.



Gambar: 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 Key*nya adalah gabungan dari *Partial Key* dan *Primary Key* dari relasi induknya (*identifying owner*).

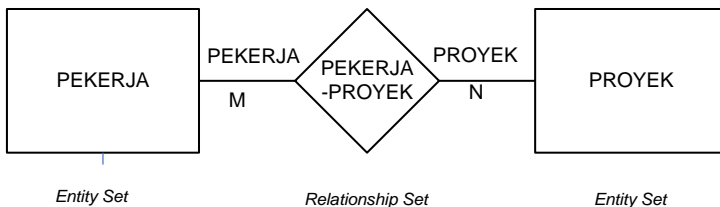
Contoh Hasil Transformasi dari Diagram ER ke Database Relational:

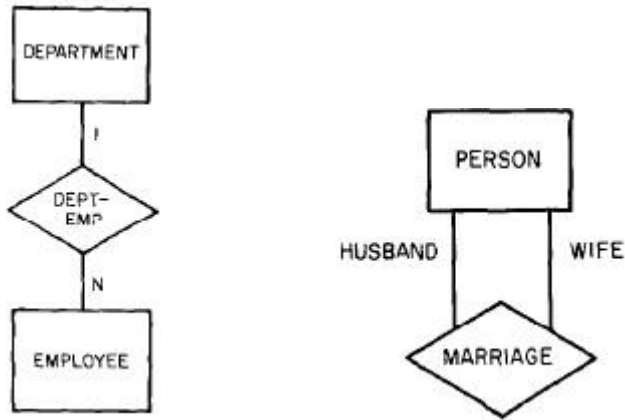
PEGAWAI : (nopeg, napeg, alm1, kdpos, tgllah, umur, supervisor-id, nobag)
 BAGIAN : (nobag, nabag, lokasi, manager)
 PROYEK : (nopro, napro, nobag)
 LOKASI_PROYEK : (nopro, lokapr)
 PEKERJAAN : (nopeg, nopro, jam)
 TANGGUNGAN : (nopeg, nama, jnkel, hubungan)

Tabel pegawai

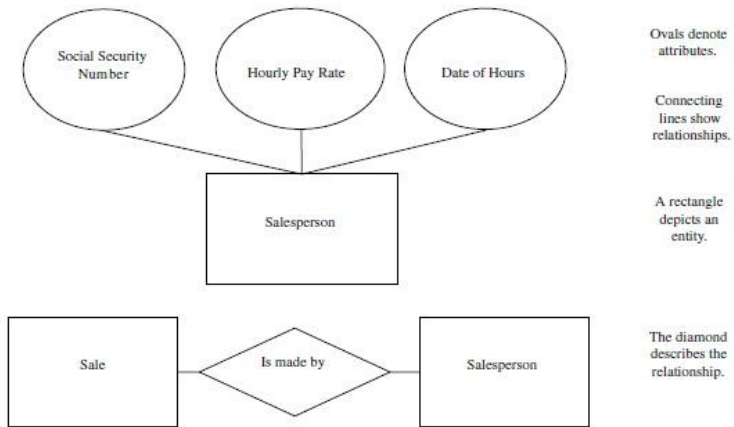
Fields	Type	Size
nopeg	Character, primary	15
napeg	Character	50
almt1	Character	100
kdpos	Character	15
tgllah	Date	
umur	numeric	5
Super_id	character	15
nobag	charcter	10

3. Contoh bentuk aplikasi diagram dalam E-R

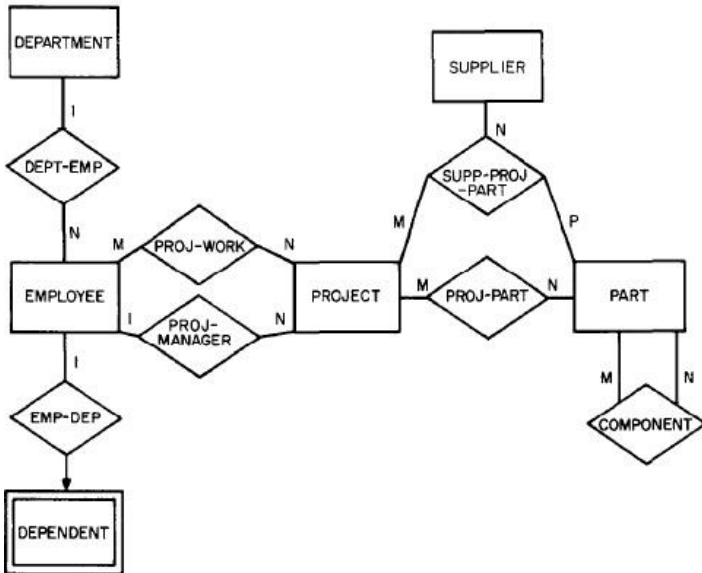




Gambar 1: contoh sederhana entity relationship diagram (sumber Chen, 1976)



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



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

C. PROGRAM APLIKASI

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

- Program berbasis *office* dengan cara menggambar biasa. *MsWord* menggunakan *shapes*, *power point* menggunakan *shapes*, atau *MsVisio* 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.smdraw.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.

D. 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

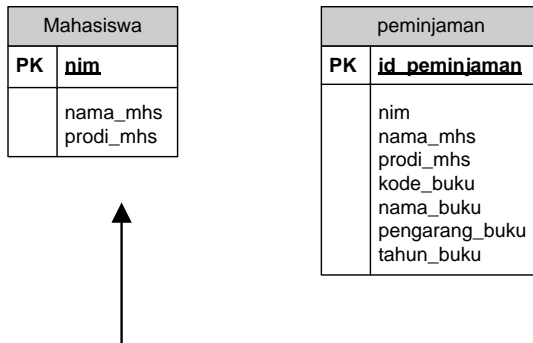
Categories:

- Definition
- Columns
- Primary ID
- Indexes
- Triggers
- Check
- Extended
- Notes

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"/>	

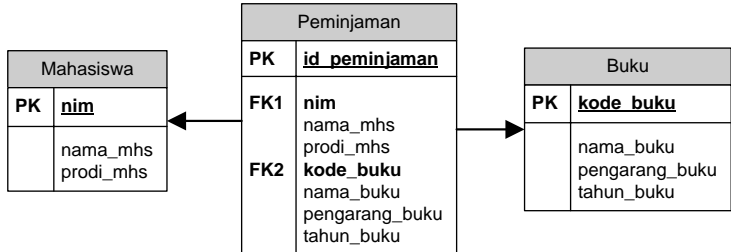
Kemudian buatlah entity lainnya seperti langkah di atas misalnya entity buku dan entity peminjaman. Lihat bagan di bawah untuk tujuannya.

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.

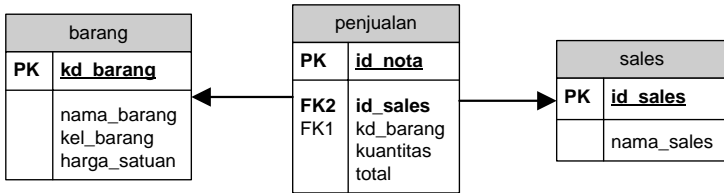


Dari bagan tersebut dapat ditransformasikan ke dalam database sebagai berikut:

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

Soal dan Latihan

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



2. Coba Anda cari contoh beberapa entitas yang dapat Anda temukan dalam keseharian serta sebutkan pula apa hubungan antar entitas tersebut?
3. Dasar *entity relationship diagram*: Dalam akuntansi mengenal berbagai rekening, silahkan buat diagram yang menjelaskan entitas rekening yang memiliki atribut sebagai berikut:
 - a. Kode rekening
 - b. Nama rekening
 - c. Nomoral D/K
4. Dirancang terdapat tabel database rek_akun, dan unit. Masing-masing mempunyai atribut: rek_akun terdiri dari rek_kode, rek_nama, kelompok, normal. Kemudian unit terdiri dari: unit_kode, unit_nama, unit_singkat. Kemudian perusahaan membuat tabel transaksi pengeluaran yang menggabungkan transaksi database tersebut dengan menambahkan jumlah pengeluaran yang dilakukan. Buatlah:
 - a. Gambar ERD yang menjelaskan hubungan antar tabel database.
 - b. Transformasikan dalam bentuk struktur database.

BAB 6. DATA FLOW DIAGRAM (DFD)

Tujuan

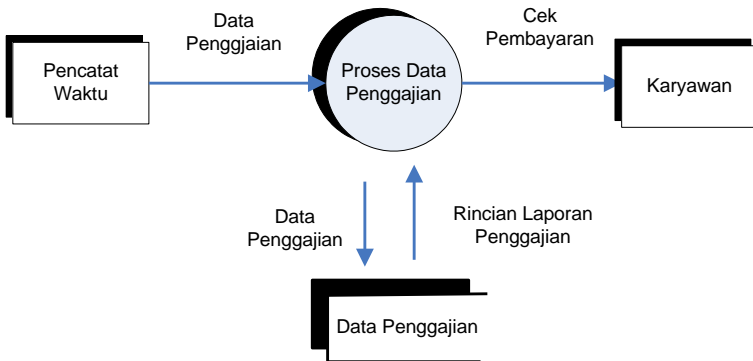
1. Mengetahui kegunaan DFD
2. Mengetahui bentuk dasar DFD
3. Mengaplikasikan penyusunan DFD

Diagram arus data (DAD) atau data flow diagram (DFD) lebih menunjukkan data yang mengalir dari satu entiti (entity) ke entiti yang lain (Jogiyanto, 2008: 457). Prinsip kerja DAD adalah dekomposisi yaitu memecah sistem yang kompleks menjadi beberapa modul-modul yang lebih mudah dipahami dan lebih terinci. DAD yang digambar pertama adalah diagram level atas (top level diagram) yang juga disebut diagram konteks (context diagram). Kemudian context diagram digambar lebih rinci dalam overview diagram atau diagram level 0. Diagram level 0 dapat dipecah lagi menjadi diagram level 1, diagram level 2 dan seterusnya.

Fungsi dari sistem informasi dapat diekspresikan dalam bentuk transformasi (proses) atas masukan (*input*) ke keluaran (*output*), data mana yang diperlukan ataupun harus diperbarui (Gangolly, 2000). *Dataflow diagrams* merupakan suatu model sistem informasi yang menunjukkan model fungsional atau model pemrosesan. Diagram arus data (*dataflow diagrams*) terdiri dari **entitas eksternal** (*external entities*) yang dilambangkan dengan kotak, **proses-proses** (*processes*) yang dilambangkan dengan

lingkaran, **penyimpanan data** (*data stores*) yang dilambangkan dengan kotak terbuka atau dua garis paralel, dan **arus data** (*dataflows*) yang dilambangkan dengan arah panah.

Contoh bentuk diagram arus data



DFD untuk Pemrosesan Penggajian

Contoh: penerapan DFDs untuk sistem input order penjualan dan pemrosesan (diadopsi dari Gangolly, 2000).

DFDs: Diagram konteks (*context diagram*)

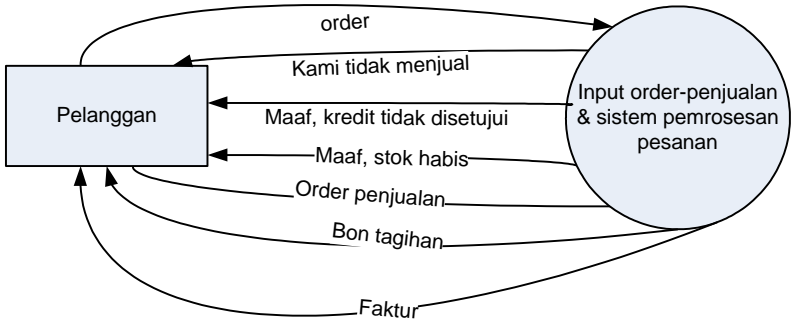
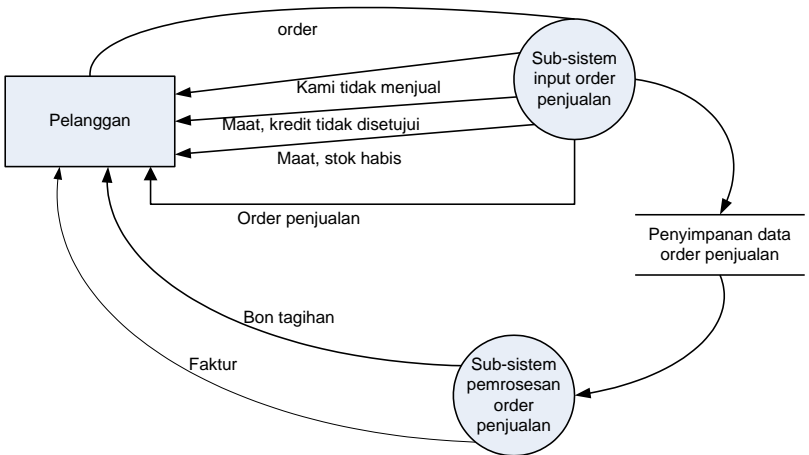
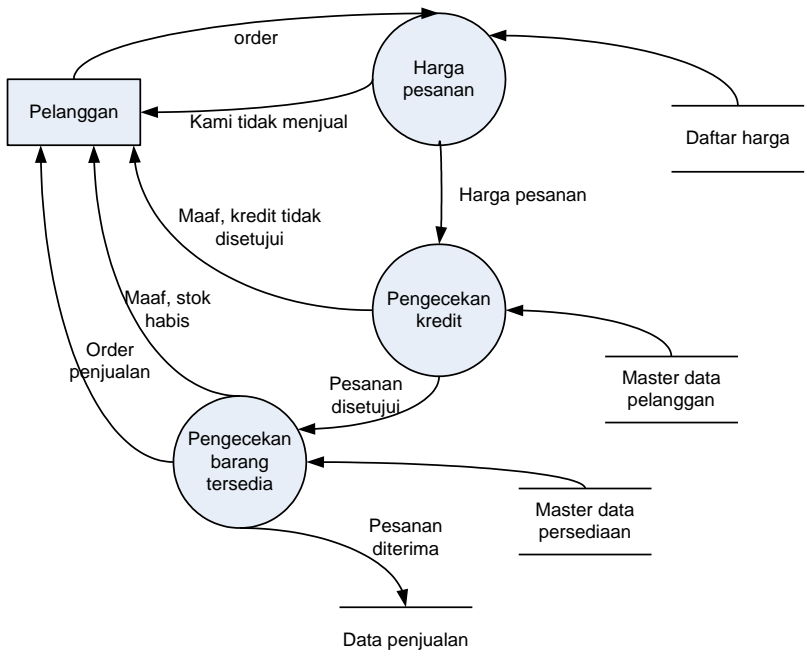


Diagram konteks untuk sistem input order penjualan dan pemrosesan

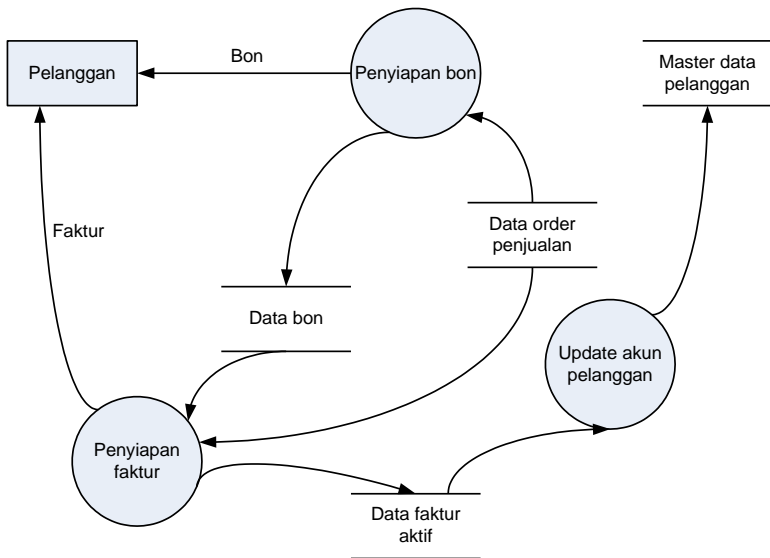
Level 0 Diagram arus data logis (*logical dataflow diagram*) sistem input order penjualan dan pemrosesan



Level 1 Diagram arus data logis (*logical dataflow diagram*): Sub-sistem input order penjualan



Level 1 Diagram arus data logis (*logical DFD*): Sub-sistem pemrosesan pesanan penjualan



Pertanyaan

1. Ceritakan dalam bentuk teks tentang sistem input pesanan penjualan dan pemrosesannya di atas mulai dari diagram konteks, level 0, level 1.
2. Buatlah diagram arus data konteks sistem transfer uang bank melalui internet banking. Nasabah akan menggunakan form input transfer penjualan. Output proses yang mungkin muncul pada nasabah antara lain:
 - Nomor rekening tujuan tidak dikenal
 - Saldo tidak mencukupi
 - Nomor PIN yang dimasukkan tidak cocok
 - Proses terlaksana dan informasi nota transfer

BAB 7. UML: Use Case Diagram

Tujuan

1. Memahami kegunaan *use case diagram*
2. Mengetahui bentuk dasar *use case diagram*
3. Mengaplikasikan pembuatan *use case diagram*

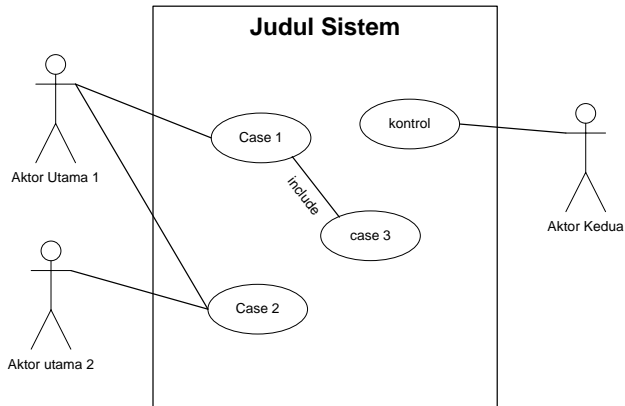
Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login 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*.

Berikut adalah komponen-komponen dalam menggambar suatu *use case*:

- Boundary system: berupa kotak sebagai batasan sistem.
- Judul/nama sistem.
- Aktor: pengguna, pihak yang terlibat dalam sistem. Aktor terdiri dari aktor utama dan aktor kedua. Aktor utama dapat ditempatkan di sebelah kiri dan aktor kedua di sebelah kanan.
- Case: kegiatan, fungsi.
- Garis komunikasi



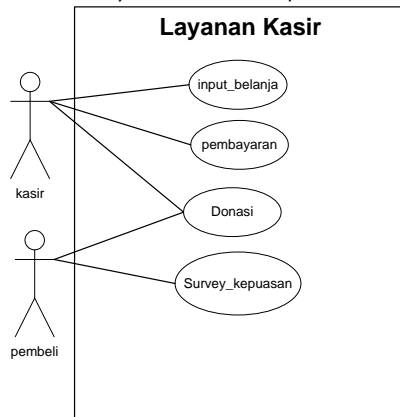
Tugas:

Perhatikan contoh-contoh kasus UML Use Case Diagram pada materi ini. Perhatikan pula bentuk bagan, antara lain: Boundary system, judul sistem, aktor, kegiatan. Kemudian, Anda diminta untuk merancang sebuah use case untuk kasus yang Anda

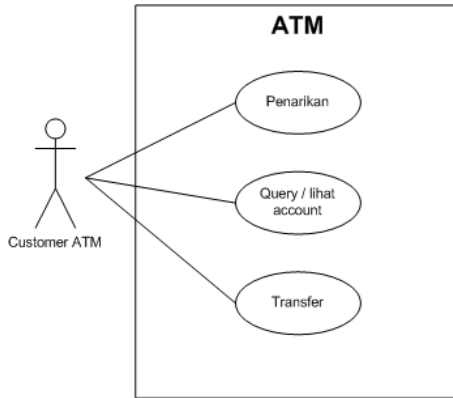
tentukan sendiri. Misalnya (alternatif): sistem penjualan, input transaksi jurnal keuangan, stok barang gudang, dsb.

Contoh diagram use case

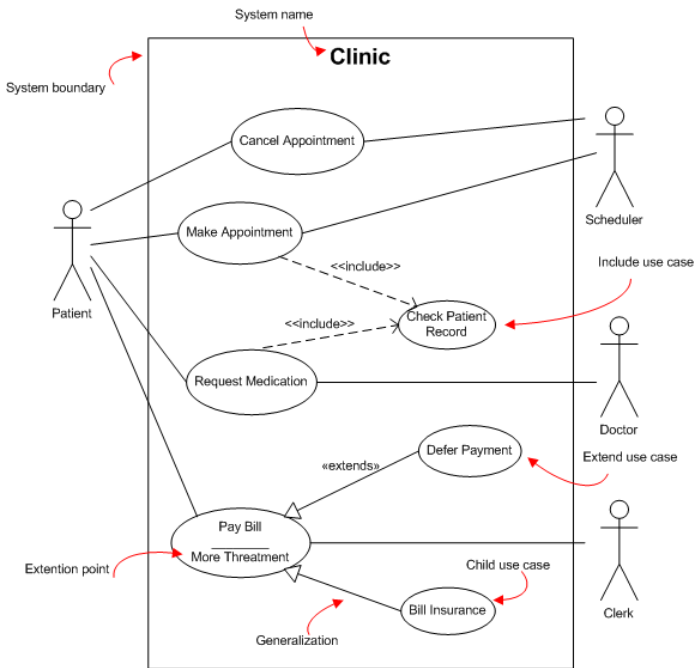
Contoh 1: use case layanan kasir. Selain untuk input harga dan menerima pembayaran, pada saat ini dapat kita lihat bahwa kasir-kasir pada swalayan tertentu disediakan layar komputer yang menghadap ke pembeli. Pembeli dapat melihat harga belanja. Selain itu ketika terjadi sisa, kasir menawarkan apakah sisa tersebut akan didonasikan? Jika ya kasir dapat meminta kepada pembeli untuk klik tombol donasi. Setelah transaksi selesai pembeli dapat melihat adanya pertanyaan survey dari toko, apakah Anda puas dengan layanan kami? Pertanyaan ini sebagai survey kepuasan konsumen. Pembeli dapat menyentuh tombol ya atau tidak pada survey.



Contoh 2: use case ATM



Contoh 3



Latihan

Latihan A: Berikut terdapat suatu penjelasan tekstual yang perlu Anda terjemahkan dalam diagram *use case*.

PEMBELIAN PRODUK

Skenario Keberhasilan Utama:

1. Pelanggan melihat-lihat katalog dan memilih barang untuk dibeli.
2. Pelanggan memeriksa daftar pesanan
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 B

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 8. UML: ACTIVITY DIAGRAMS

Tujuan

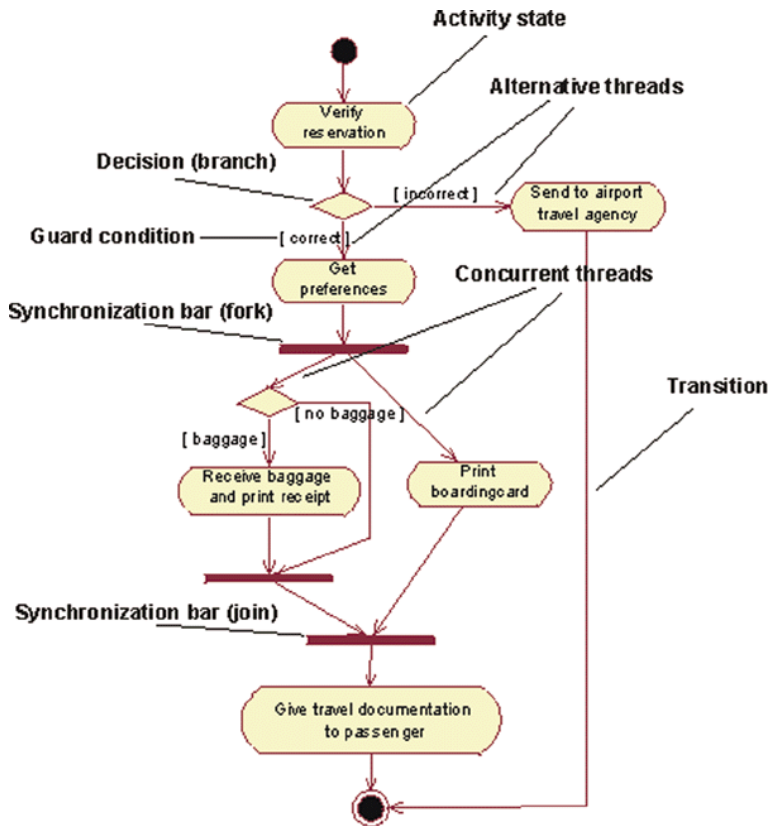
1. Mengetahui kegunaan *activity diagram*
2. Mengetahui simbol dasar *activity diagram*
3. Mengaplikasikan pembuatan *activity diagram*

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).

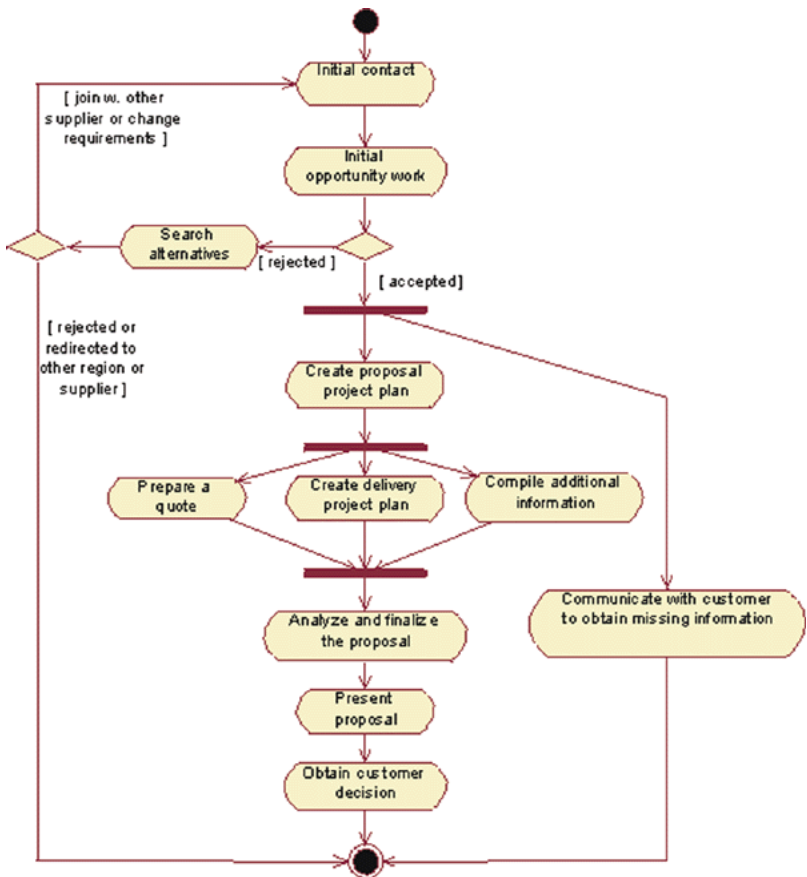
Bentuk simbol diagram dapat dijelaskan sebagai berikut:

1. *Activity states* (pernyataan aktivitas), merepresentasikan bentuk atas langkah yang akan dilaksanakan dalam aliran kerja.
2. *Transitions* (perpindahan), menunjukkan pernyataan aktivitas yang terjadi setelah suatu kegiatan tertentu.
3. *Decisions* (keputusan), merupakan suatu set pengendali (*a set of guard*) atas suatu kondisi yang ditentukan.
4. *Synchronization bars* (garis sinkronisasi). Dapat digunakan untuk menunjukkan sub aliran secara paralel.

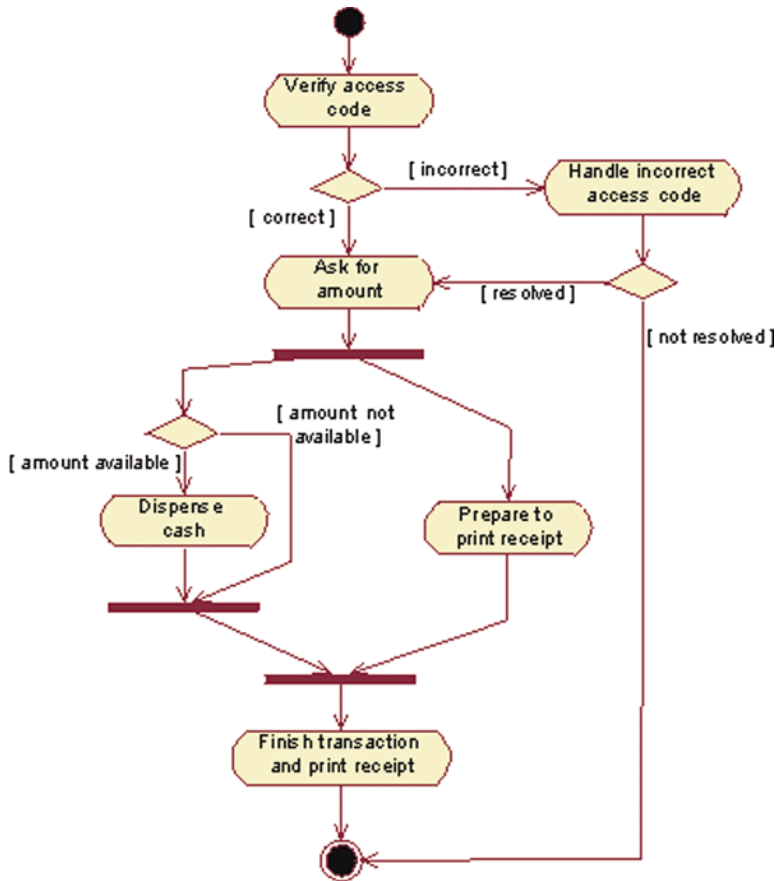
Berikut ini adalah contoh bentuk diagram dasar dalam *UML: Activity Diagram*. Dicontohkan pula suatu aktivitas diagram untuk kasus bisnis *airport check-in*.



Contoh aktivitas diagram untuk kasus pemrosesan proposal



Contoh aktivitas diagram dalam pengambilan uang di ATM



Soal/Latihan:

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 pengeluaran kas.

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

Buatlah aktivitas diagram layanan penjualan barang konsinyasi pada Departemen Store Jaya

BAB 9. UML: CLASS DIAGRAMS

Tujuan

1. Mengetahui penggunaan class dalam sistem
2. Mengetahui simbol dasar *class diagram*
3. Membaca *class diagram*

A. Definisi dan Konsep Klas

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 klas ini merupakan bagian dalam menjelaskan struktur-struktur objek dasar (*the fundamental object structures*). Suatu diagram klas (*class diagram*) adalah suatu gambar dengan simpul-simpul (*nodes*) yang menunjukkan klas-klas dan gambaran asosiasi (*asociations*), perumuman (*generalizations*), atau ketergantungan (*dependencies*) (Kuske et al., 2009). *Class* dapat direpresentasikan dalam sebuah *interface* atau sebaliknya merupakan implementasi dari sebuah

interface yang berupa klas abstrak. Klas abstrak tidak memiliki atribut namun hanya memiliki metode.

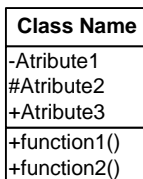
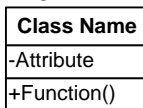
Sebuah *class* terdiri dari (Kuske et al., 2009):

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

Atribut dan metode 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.

Berikut disajikan contoh diagram dasar yang dibuat menggunakan piranti microsoft visio UML *class diagram*:



Gambar: bentuk dasar *class diagram* dibuat dari MsVisio

Dari gambar bentuk dasar *class diagram* dapat dilihat atribut memiliki simbol *visibility* di depannya berupa tanda - # dan +. Tanda (-) berarti atribut tersebut bersifat *private*, tanda # bersifat *protected*, dan tanda (+) bersifat *public*. Tanda tersebut mungkin akan berbeda ketika menggunakan piranti lunak lainnya dalam pembuatan diagram.

Banyak *platform* pemrograman dapat menggunakan klas dalam penulisannya. Setiap *platform* memiliki cara sedikit berbeda dalam pendefinisian klas, atribut, dan fungsi/metode. Namun secara umum akan memiliki konsep yang mirip dan membutuhkan sedikit penyesuaian.

Contoh: Skrip dasar klas

Class namaklas

```
{  
    //isi klas  
}
```

Contoh: Skrip dasar klas dengan fungsi

Class namaklas

```
{  
    Function fungsi1()  
    {  
        //isi fungsi  
    }  
    Function fungsi2()  
    {  
        //isi fungsi  
    }  
} // penutup klas
```


Berikut disajikan contoh penulisan klas dalam bahasa pemrograman php berdasarkan gambar diagram di atas. Contoh buatlah file bernama `classname.php` menggunakan program notepad atau program penulis *script*. Kemudian jalankan di komputer server yang memiliki program php.

```
<?php
Class class_name
{
    Private $attribute1="private";
    Protected $attribute2="Protected";
    Public $attribute3="Public";

    Function1()
    {
        Echo $this->$attribute1;
        Echo $this->$attribute2;
        Echo $this->$attribute3;
    }

    Function2()
    {
        Echo "Halo, selamat belajar";
    }
} // penutup klas
?> // penutup php
```

Klas yang ditulis dalam php tersebut tidak dapat tampil atau dieksekusi dengan sendirinya, melainkan harus dipanggil dan diperintah. Berikut contoh *script*

pemanggilnya dapat ditulis dibagian selanjutnya file di atas atau ditulis di file lainnya.

<?

```
$cls= new class_name();
```

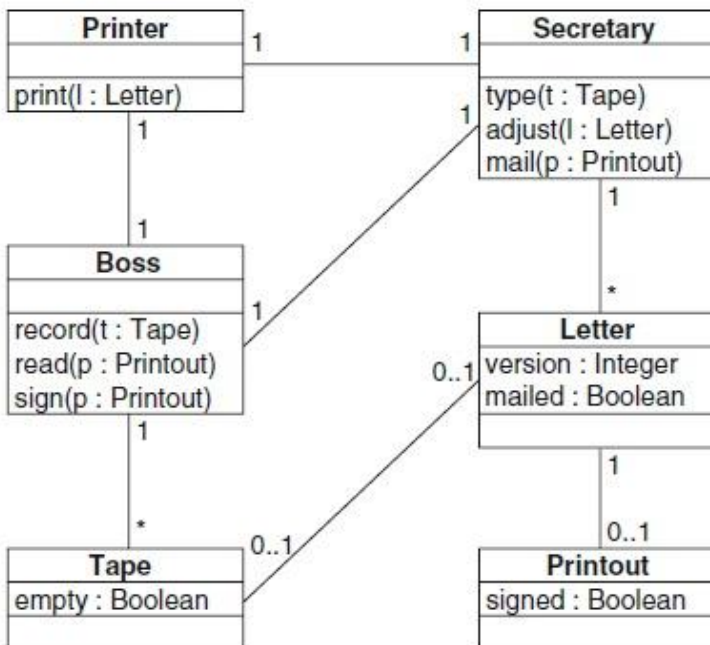
```
$cls->function1();
```

```
$cls->function2();
```

?>

B. Aplikasi Diagram Klas (*Class Diagrams*)

Dicontohkan sebuah diagram klas yang terdiri dari asosiasi biner klas.



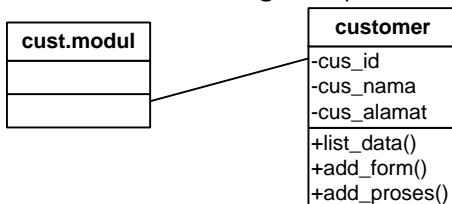
Gambar: Diagram klas dan asosiasinya (Sumber: Kuske et al., 2009)

Terdapat klas bernama *Printer*, *Secretary*, *Boss*, *Letter*, *Tape*, dan *Printout*. Beberapa klas berisi **operasi** yang menjelaskan aksi yang dapat dilakukan suatu objek klas. Misal seorang sekretaris dapat mengetik surat yang disimpan dalam media penyimpan atau mengeprintnya. Beberapa klas lain menggunakan **atribut**, contoh media penyimpan dapat kosong (*empty*) atau berisi yang diindikasikan dengan nilai *boolean*. **Asosiasi** klas 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 penyimpan (*tapes*) yang diwakili dengan tanda (*).

Latihan Aplikasi

Suatu class bernama customer memiliki fungsi `list_data`, `add_form`, `add_proses`. Variabel yang digunakan dalam class tersebut adalah `cus_id`, `cus_nama`, `cus_alamat`. Class customer tersebut dipanggil oleh suatu modul bernama `cust.modul`.

Gambarlah class diagramnya.



Sehingga contoh pembuatan program adalah sebagai berikut, program berikut disusun dengan bahasa pemrograman php.

```
<?php
class customer
{
    function list_data()
    {
        echo "Daftar Pelanggan";
    }
    function add_form()
    {
        echo "Tambah Data Pelanggan";
    }
    function add_proses()
    {
        echo "Tunggu Proses Simpan";
    }
}
?>
```

Contoh file yang digunakan untuk mengaktifkan klas, misal kita beri nama cust.modul.php

```
<?
// panggil file customer
include_once("customer.php");
//mengaktifkan klas
$cls= new customer();
//cara mengaktifkan fungsi
$cls->list_data();
echo "<br>";
```

```
$cls->add_form();  
echo "<br>";  
$cls->add_proses();  
?>
```

BAB 10. PROTOTYPING

Tujuan

1. Mengetahui metode pengembangan alternatif
2. Mengetahui manfaat prototipe

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*

Prototipe merupakan media komunikasi dan menunjukkan rancangan sistem dalam tahap awal. Dengan komunikasi tersebut diharapkan mendapat reaksi dari pihak-pihak pengguna:

- a. Reaksi awal dari pengguna
- b. Reaksi saat bekerja dengan prototipe
- c. Apakah fitur-fitur sudah sesuai dengan kebutuhan.
- d. Reaksi dikumpulkan dalam lembar observasi, wawancara, dan umpan balik.

- e. 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 pengarahan 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 pentransferan 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 dikembangkan menggunakan metode *prototyping*. Kondisi berikut perlu diperhatikan dan dipertimbangkan oleh pengembang sehingga kegiatan *prototyping* berjalan lancar.

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

Agar pengembangan sistem menggunakan metode prototipe berjalan lancar, maka berikut terdapat tips yang perlu diperhatikan:

1. Bekerja sesuai modul. Sistem dirancang berdasarkan modul-modul tertentu. Modul ini dapat mengarahkan sistem tertentu yang akan dikembangkan dan dikomunikasikan dengan pengguna. Modul ini sepertihalnya sub-sistem yang memiliki keterkaitan antar kegiatan sistem.
2. Membangun prototipe dengan cepat. Prototipe dibangun dengan cepat tujuannya untuk dapat segera ditunjukkan dan dibahas dengan pengguna.

Saat komunikasi dengan pengguna tersebutlah dapat digali bahan analisis dan perancangan selanjutnya.

3. Memodifikasi prototipe. Modifikasi prototipe didasarkan pada komunikasi dan usulan dari pengguna maupun pihak analis dan perancang sistem.
4. Menekankan antarmuka pengguna. Pengguna adalah bagian terpenting. Pengguna dapat dibagi menjadi pengguna input program dan pengguna informasi. Pengguna input (operator) perlu dipermudah dalam pengoperasian sistem. Sedangkan pengguna informasi diupayakan untuk disediakan informasi sesuai kebutuhan mereka.

C. Kelemahan dan Kelebihan

1. Kelebihan-Kelebihan Prototipe

Pengembangan prototipe memiliki keunggulan dalam menunjukkan gambaran sistem yang akan dikembangkan segera di awal pengembangan. Dari prototipe tersebut dapat dikomunikasikan rancangan sistem sehingga dapat diambil kebijakan/tindakan:

- a. Mengubah sistem sejak dini dalam masa perkembangan apabila terdapat usulan perubahan atau penyesuaian.
- b. Membatalkan sistem yang tidak diharapkan.
- c. Membantu dalam memberi rancangan sebuah sistem yang sesuai dengan kebutuhan dan harapan pengguna.

2. Kelemahan *Prototyping*

Selain terdapat keunggulan berikut adalah kelemahan dalam proyek pengembangan prototyping yang perlu diperhatikan dan perlu diatasi.

- a. Mengelola proyek pepanjangan prototipe secara tak terbatas menciptakan masalah tersendiri. Prototipe adalah fitur awal dan belum lengkap, jika pengembang hanya fokus mengembangkan prototipe maka sistem yang lengkap tidak pernah terbentuk dan proyek tidak dapat ditentukan kapan selesainya. Maka pengembang perlu menentukan target penyelesaian dari sistem tersebut.
- b. Memakai sistem yang belum selesai seolah-olah sebagai sistem yang selesai. Hal ini dapat mengakibatkan ketidakpuasan pengguna karena menganggap sistem dapat dioperasikan secara baik yang sebenarnya sistem baru dalam tahap awal pengembangan dan fiturnya belum lengkap.

Pertanyaan

1. Apa yang ingin dicari pengembang dalam komunikasi menggunakan prototipe?
2. Sebutkan jenis-jenis prototipe!
3. Apa keunggulan dari sistem prototipe?
4. Apa kelemahan pengembangan prototipe?

DAFTAR PUSTAKA


- Al Fatta, H. 2007. *Analisis & Perancangan Sistem Informasi untuk Keunggulan Bersaing Perusahaan & Organisasi Modern*. Yogyakarta: Andi Offset.
- Bradford, Marianne; Sandra B. Richtermeier; 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.
- 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.
- 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

- Ericsson, Maria. 2004. *Activity Diagrams: What They Are and How to Use Them*. IBM Software Group.
- Falkenberg, Eckhard D. et al. 1998. *A Framework of Information System Concepts*. The FRISCO Report (Web Edition). International Federation for Information Processing (IFIP). Department of Computer Science, University of Leiden, The Netherlands.
- Fowler Martin. 2005. "UML Distilled Panduan Singkat Bahasa Pemodelan Objek Standar" Edisi 3 dalam bahasa Indonesia. Andi: Yogyakarta
- Gangolly, Jagdis. 2000. *Lecture Note on Analysis & Design of Accounting Information Systems*. Departement of Accounting & Law, State University of New York.
- Habiburrochman and Hidayatin, D.A. (2012), The Influence of User Participation to User Satisfaction in Developing Information System, Proceeding: Airlangga Accounting International Conference & Doctoral Colloquium 2012, Nusa Dua Bali
- Hofstede, Geert. 1984. Cultural Dimensions in Management And Planning. *Asia Pacific Journal of Management*, January 1984.
- Jogiyanto, H.M., 2009. *Sistem Teknologi Informasi*. Edisi III. Andi Offset Yogyakarta.
- Jogiyanto, H.M. 2005. *Sistem Informasi Strategik: untuk keunggulan kompetitif*. Edisi 2, 2006. Penerbit Andi Offset. Yogyakarta.
- Jogiyanto, HM. 2008. *Sistem Informasi Keperilakuan*. Andi Offset. Yogyakarta.
- Jogiyanto, HM. 2008. Sistem Teknologi Informasi: Pendekatan terintegrasi konsep dasar, teknologi,

aplikasi, pengembangan, dan pengelolaan . Edisi III.
Penerbit Andi.

- Kumeiga, Andrew; and Ben Van Vliet. 2008. A Software Development Methodology for Research and Prototyping in Financial Market. *Unpublished*. Cornell University.
- 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
- 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.
- 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.

Biografi Singkat

	Nama	: Agustinus Mujilan, S.E., M.Sc.
	Juga dikenal	: Agus Mujilan, Mujilan
	TTL	: Sleman, 02 Juni 1977
	NIDN	: 0702067703
	ORCID	: http://orcid.org/0000-0003-3845-0966
	Alamat	: Bumi Mas Indah 2 Blok L No. 11, Madiun
	Email	: agusmujilan@gmail.com

Pendidikan formal

1. SD N Kwayuhan, Yogyakarta, Lulus 10 Juni 1989.
2. SMP N Sendangsari, Yogyakarta, Lulus 29 Mei 1992.
3. SMEA N Godean, Yogyakarta, Jurusan Keuangan, Lulus 22 Mei 1995.
4. S-1 Akuntansi, Universitas Widya Mandala Madiun, Lulus 25 September 2003.
5. S-2 Ilmu Akuntansi, Universitas Gadjah Mada Yogyakarta, Lulus 18 Januari 2012

Pendidikan Non-Gelar

1. Kursus Akuntansi Dasar Satu, Ijazah Nasional Dep. Dik. Bud., 1994
2. Kursus Bahasa Inggris Dasar Satu, Ijazah Nasional Dep. Dik. Bud., 1994.
3. Kursus Komputer Lotus 123, Ijazah Nasional Dep. Dik. Bud., 1995.
4. Program 1 Tahun IMKI (Pusat Pendidikan Komputer dan Manajemen) Yogyakarta, Program Profesi Akuntansi Keuangan dan Perbankan, 1996.
5. Pendidikan Bahasa Inggris Jarak Jauh, Cambridge English House (CEH) Yogyakarta 1996.

6. Kursus Operator Windows; Kusuma Bangsa Madiun, 1999.
7. Elementary Level of English for Active Communication, ELTI Gramedia Madiun, 2001.
8. Pre-Intermediate Level of The English for Accounting Departmen”, ELTI Gramedia Madiun, 2001.

Pengalaman Publikasi Ilmiah

A. Jurnal

Judul	Penerbit/Jurnal
Posisi Sistem Informasi sebagai Aktiva dan Biaya pada Pelaporan Keuangan	Jurnal Ilmiah Widya Warta; Juli 2009
Pengaruh Dukungan Teknologi Informasi untuk Pengelolaan Pengetahuan terhadap Berbagi Pengetahuan pada Perbankan di Jawa Timur	Ekonomika Vol. 4 No. 2 Desember 2011, Kopertis 7 Jawa Timur
Managerial Knowledge Sharing Behavior in the Commercial Banking Sector	The Indonesian Journal of Accounting Research (IJAR) Vol. 15 No. 3, Sept. 2012
Kualitas Layanan Internet Banking dan Karakteristiknya Berdasarkan Frekuensi Penggunaan	Widya Warta No. 01 Tahun XXXVII/Januari 2013
Konsep Keseuaian Sistem Anggaran dan Akuntansi Keuangan dalam Sistem Akuntansi Universitas	Widya Warta No. 01 Tahun XXXIX/Januari 2015, pp. 125 - 141
Pemotivasi Knowledge Sharing Manajerial Melalui Media Internet (Mujilan & Purbandari)	Widya Warta No. 02 Tahun XXXIX/Juli 2015; pp. 250 – 265

B. Pemakalah Seminar Ilmiah (*oral presentation*)

Nama Pertemuan Ilmiah	Judul	Waktu & Tempat
<i>Airlangga Accounting International Conference (AAIC) 2012,</i>	<i>Internet Banking Using Intention: a Model Modification Based on Quality and Risk Characteristics</i>	Juni 2012, Nusa Dua, Bali

Nama Pertemuan Ilmiah	Judul	Waktu & Tempat
Seminar Nasional Akuntansi (SNA) XV Banjarmasin	<i>The Organizational Roles of Knowledge Sharing and Information Technology in the Commercial Banking Sector</i>	September 2012, Banjarmasin
National Conference Faculty Business (NCFB) and Doctoral Colloquium	Pemotivasi Knowledge Sharing Manajerial melalui Media Internet	Oktober 2013, Fakultas Bisnis dan Pascasarjana Unika Widya Mandala Surabaya.
Konferensi Regional Akuntansi (KRA) Wilayah Jawa Timur	Disain Sistem Penanganan Transaksi pada Akuntansi Universitas	Universitas Kanjuruhan, Malang, 29 – 30 April 2015
The 9th NCFB and Doctoral Colloquium	Disain Sistem Data Nasabah dan Komunikasi Bisnis Lembaga Keuangan Kelurahan Kota Madiun	Unika Widya Mandala Surabaya, Fakultas Bisnis, 14 Oktober 2016