

BAB 2

LANDASAN TEORI

2.1. Sistem Informasi Akuntansi

2.1.1. Pengertian Sistem Informasi Akuntansi

Mulyadi (1997, h.2) mendefinisikan bahwa sistem adalah sekelompok unsur yang erat berhubungan satu dengan lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

O'Brien (2003, p.7) berpendapat bahwa sebuah sistem informasi dapat berupa kombinasi terorganisasi dari orang, *hardware*, *software*, jaringan komunikasi, dan sumber data yang mengumpulkan, mentransformasi, dan menyebarkan informasi di dalam sebuah organisasi.

Selanjutnya menurut Romney dan Steinbart (2003), sistem informasi akuntansi adalah sumber daya manusia dan modal dalam organisasi yang bertanggung jawab untuk mempersiapkan informasi keuangan dan informasi yang diperoleh dari mengumpulkan dan memproses berbagai transaksi perusahaan.

Jadi dapat disimpulkan bahwa sistem informasi akuntansi adalah kombinasi antara orang, *hardware*, *software*, jaringan komunikasi, dan sumber data yang erat berhubungan satu dengan lainnya, yang bertanggung jawab mempersiapkan informasi keuangan dan informasi yang diperoleh dari mengumpulkan dan memproses transaksi perusahaan.

2.1.2. Tujuan Umum Pengembangan Sistem Akuntansi

Menurut Mulyadi (1997, h.19), tujuan umum pengembangan sistem akuntansi adalah:

- Untuk menyediakan informasi bagi pengelolaan kegiatan usaha baru.

Kebutuhan pengembangan sistem akuntansi terjadi jika perusahaan baru didirikan atau suatu perusahaan menciptakan usaha baru yang berbeda dengan usaha yang telah dijalankan selama ini.

- Untuk memperbaiki informasi yang dihasilkan oleh sistem yang sudah ada.

Adakalanya sistem akuntansi yang berlaku, tidak dapat memenuhi kebutuhan manajemen, baik dalam hal mutu, ketepatan penyajian maupun struktur informasi yang terdapat dalam laporan. Hal ini kemungkinan disebabkan oleh perkembangan usaha perusahaan sehingga menuntut sistem akuntansi untuk dapat menghasilkan laporan dengan mutu informasi yang lebih baik dan tepat penyajiannya dengan struktur informasi yang sesuai dengan tuntutan kebutuhan manajemen.

- Untuk memperbaiki pengendalian akuntansi dan pengecekan intern.

Akuntansi merupakan alat pertanggung jawaban kekayaan suatu organisasi. Pengembangan sistem akuntansi seringkali ditunjukkan untuk memperbaiki perlindungan terhadap kekayaan organisasi sehingga dipertanggung jawabkan terhadap penggunaan kekayaan organisasi dapat dilaksanakan dengan baik. Pengembangan sistem akuntansi, dapat pula ditujukan untuk memperbaiki pengecekan intern agar informasi yang dihasilkan oleh sistem tersebut dapat dipercaya.

- Untuk mengurangi biaya klerikal dalam penyelenggaraan catatan akuntansi. Pengembangan sistem akuntansi seringkali ditujukan untuk menghemat biaya. Untuk memperolehnya, diperlukan pengorbanan sumber ekonomi yang lain. Oleh karena itu dalam menghasilkan informasi, perlu dipertimbangkan besarnya manfaat yang diperoleh dengan pengorbanan yang dilakukan.

2.2. Lembaga Keuangan Bank

2.2.1. Pengertian Bank

Pengertian bank menurut UU No. 7 Tahun 1992 tentang Perbankan sebagaimana telah diubah dengan UU No. 10 Tahun 1998 adalah:

... Bank adalah badan usaha yang menghimpun dana dari masyarakat dalam bentuk simpanan, dan menyalurkannya kepada masyarakat dalam bentuk kredit dan atau bentuk-bentuk lainnya, dalam rangka meningkatkan taraf hidup masyarakat banyak.

2.2.2. Bank Asing

Sedangkan menurut Dahlan Siamat (2005, h.56) yang termasuk dengan bank asing adalah kantor cabang dari suatu bank di luar Indonesia yang saat ini hanya diperkenankan beroperasi di Jakarta dan membuka kantor cabang pembantu di beberapa ibukota propinsi selain Jakarta yaitu: Semarang, Surabaya, Bandung, Denpasar, Ujung Pandang, Medan, dan Batam.

Bank asing, yang sejak awal tahun 1970-an tidak diizinkan membuka kantor cabang di Indonesia, sejak pertengahan tahun 1999 diberi kembali kesempatan membuka kantor cabangnya dengan memenuhi persyaratan yang ditetapkan. Bank asing yang dapat membuka cabangnya tersebut harus termasuk

bank yang memiliki *asset* 200 terbesar dunia dan memiliki rating minimal A dari lembaga peringkat (*rating agency*) internasional. Selama ini bank asing hanya diperkenankan membuka kantor perwakilan di Jakarta.

Pemberian pelayanan jasa-jasa dalam kegiatan operasional bank asing pada prinsipnya tidak memiliki perbedaan signifikan dengan bank-bank umum swasta nasional, kecuali dalam hal pembatasan pembukaan kantor di wilayah tertentu di Indonesia. Selain itu, bank asing tidak diperkenankan menerima simpanan dari masyarakat dalam bentuk tabungan. Di samping itu, segmen usaha bank asing yang ditekuni terutama adalah segmen korporasi atau *corporate banking*. Ciri lain dari kegiatan bank asing ini adalah penyediaan jasa di bidang *investment bank* yang menawarkan jasa di bidang pasar modal.

Ketentuan-ketentuan lain yang berkaitan dengan pengaturan bank berlaku juga bagi bank asing antara lain: *net open position*, Giro Wajib Minimum, *legal lending limit*, kewajiban penyediaan modal minimum (*Capital Adequacy Ratio*), *Loan to Deposit Ratio*, dan termasuk ketentuan tingkat kesehatan bank. Bentuk hukum cabang bank asing yang beroperasi di Indonesia sampai akhir 2004 berjumlah 10 bank. Salah satu diantaranya adalah Bank XYZ.

2.3. Transaksi Penerimaan dan Pengeluaran Dana Nasabah (Incoming and Outgoing Transaction)

2.3.1. Jasa Pengiriman Uang (Transfer)

Menurut Faud dan Rustan (2005, h.91), transfer adalah salah satu jasa yang diberikan oleh Bank Umum untuk melayani pengiriman uang dari satu tempat ke tempat lain.

Menurut Taswan, S.E., M.Si. (2005, h.271), pengiriman uang adalah perpindahan dana antar rekening dari suatu tempat (bank) ke tempat lain (cabang bank sendiri/bank lain) baik untuk kepentingan nasabah maupun kepentingan bank itu sendiri.

Sedangkan menurut Indra Bastian dan Suhardjono (2006, h.11), transfer atau pengiriman uang merupakan kegiatan penyelesaian permohonan pemindahan uang/dana dari satu kantor cabang bank ke kantor cabang lainnya atau bank korespondennya di luar negeri yang dilakukan melalui sarana komunikasi yang telah dilengkapi dengan berbagai alat pengaman, diawali dengan permohonan transfer dari nasabah, diteruskan bank dengan instruksi untuk membayar sejumlah tertentu kepada orang yang disebutkan namanya dalam transfer tersebut serta pembayaran kepada nasabah.

Jadi, transfer atau pengiriman uang yaitu jasa kiriman uang antar bank baik antar bank yang sama maupun bank yang berbeda. Pengiriman uang dapat dilakukan untuk dalam kota, luar kota maupun luar negeri.

Menurut Taswan (2005, h.272), berdasarkan lalu lintas dananya, transfer dibedakan menjadi:

- a. Transfer keluar (*outgoing transfer*) yaitu pengiriman uang atas perintah nasabah/bagian bank tertentu untuk keuntungan pihak lain pada bank lain atau cabang bank sendiri.
- b. Transfer masuk (*incoming transfer*) yaitu pengiriman uang yang diterima dari cabang lain bank sendiri atau dari bank lain untuk keuntungan nasabah sendiri atau penerima dana pada bank sendiri.

2.3.2. Jasa Kliring

Kliring berasal dari bahasa Inggris yakni “*Clear*” yang berarti penyelesaian. Menurut Faud dan Rustan (2005, h.77), kliring adalah suatu proses penyelesaian hutang piutang antar satu bank dengan bank lain dalam suatu wilayah tertentu.

Menurut Kasmir (2003, h.137), kliring merupakan jasa penyelesaian hutang piutang antar bank dengan cara saling menyerahkan warkat-warkat yang akan dikliringkan di lembaga kliring. Lembaga kliring ini dibentuk dan dikoordinir oleh Bank Indonesia setiap hari kerja. Peserta kliring adalah bank yang sudah memperoleh izin dari Bank Indonesia.

Menurut Taswan (2005, h.67), kliring merupakan sarana atau cara perhitungan hutang piutang dalam bentuk surat-sutar berharga atau surat dagang dari suatu bank peserta yang diselenggarakan oleh Bank Indonesia atau pihak lain yang ditunjuk. Dalam perkembangannya, kliring tidak hanya dilakukan secara manual tapi juga secara otomatis maupun elektronik. Oleh karena itu kliring didefinisikan juga sebagai pertukaran warkat atau data keuangan elektronik antar bank baik atas nama bank maupun nasabah yang hasil perhitungannya diselesaikan pada waktu tertentu.

Disebutkan dalam Outlook Sistem Kliring Elektronik Edisi Desember 2002, pengertian umum kliring adalah pertukaran warkat atau data keuangan elektronik antar bank baik atas nama Bank maupun nasabah yang hasil perhitungannya diselesaikan pada waktu tertentu.

Jadi, kliring adalah suatu cara penyelesaian utang piutang dalam bentuk warkat atau surat-surat berharga antara bank-bank peserta kliring di suatu tempat

tertentu. Dengan mekanisme kliring penyelesaian utang piutang antara bank-bank peserta kliring dipermudah, dipercepat, dan lebih efisien. Proses pelaksanaan kliring dilaksanakan oleh lembaga kliring Bank Indonesia dengan menyediakan tempat pertemuan antara bank-bank peserta.

2.3.3. Sistem Bank Indonesia-Real Time Gross Settlement

Jasa transfer saat ini semakin canggih, perkembangan terkini Bank Indonesia telah menyelenggarakan Bank Indonesia Real Time Gross Settlement (BI-RTGS). Sistem ini sangat cepat dalam menangani transfer antar bank. Menurut Taswan (2005, h.271), BI-RTGS adalah sistem transfer/kliring antar bank seketika. Nasabah yang menggunakan fasilitas ini akan dapat mentransfer dalam waktu sangat cepat, dalam hitungan menit. Namun demikian biayanya relatif lebih mahal dan belum semua bank menyelenggarakan sistem ini. Bank Indonesia memang belum mewajibkan.

Menurut Outlook Sistem BI-RTGS Edisi April 2005, Sistem Real Time Gross Settlement (RTGS) adalah proses penyelesaian akhir transaksi (*settlement*) pembayaran yang dilakukan per transaksi (*individually processed / gross settlement*) dan bersifat real time (*electronically processed*), dimana rekening peserta dapat didebit/dikredit berkali-kali dalam sehari sesuai dengan perintah pembayaran dan penerimaan pembayaran. Sistem Bank Indonesia Real Time Gross Settlement (BI-RTGS) adalah sistem RTGS yang diimplementasikan dalam Bank Indonesia.

Sedangkan yang dimaksud dengan *net settlement* menurut Outlook Sistem BI-RTGS adalah proses penyelesaian akhir transaksi-transaksi pembayaran yang dilakukan pada akhir suatu periode dengan melakukan

offsetting antara kewajiban-kewajiban pembayaran dengan hak-hak penerimaan sehingga hanya ada 1 net hak atau kewajiban yang akan di-settle untuk masing-masing rekening bank. Kliring menggunakan metode *net settlement* dalam rangka penyelesaian akhir.

Adapun tujuan BI-RTGS adalah untuk:

1. Menyediakan sarana transfer dana antar peserta yang lebih cepat, efisien, andal dan aman.
2. Kepastian *settlement* dapat diperoleh dengan lebih segera (*irrevocable* dan *unconditional*).
3. Menyediakan informasi rekening peserta secara *real time* dan menyeluruh.
4. Meningkatkan disiplin dan profesionalisme peserta dalam mengelola likuiditasnya.
5. Mengurangi resiko-resiko *settlement*.

2.4. Sistem Pengendalian Intern

2.4.1. Pengertian Sistem Pengendalian Intern

Menurut Mulyadi (2001, h.163), sistem pengendalian intern meliputi organisasi, metode dan ukuran-ukuran yang dikoordinasikan untuk menjaga kekayaan organisasi, mengecek ketelitian dan keandalan data akuntansi, mendorong efisiensi dan mendorong dipatuhinya kebijakan manajemen.

Menurut Wilkinson (2000, p.234), *internal control* adalah sebuah sistem, struktur, atau proses, yang diimplementasikan oleh dewan direksi perusahaan, manajemen, dan personel lainnya, dengan maksud untuk menyediakan kepastian

yang masuk akal tentang pencapaian tujuan pengendalian dalam kategori sebagai berikut:

1. Efektifitas dan efisiensi dari operasi.
2. Keandalan dari pelaporan keuangan.
3. Kepatuhan terhadap hukum dan peraturan yang dapat diterapkan.

Menurut Romney dan Steinbart (2003, p.195), pengendalian intern adalah rencana dari organisasi dan metode yang digunakan suatu bisnis untuk mengamankan harta, menyediakan informasi yang akurat dan dapat diandalkan, menghasilkan dan meningkatkan efisiensi operasional, dan mendorong keutuhan kebijakan manajerial.

Struktur pengendalian intern terdiri dari peraturan dan prosedur yang dibuat untuk menyediakan sebuah tingkat kepastian yang masuk akal dimana tujuan spesifik organisasi akan dicapai. Sistem hanya menyediakan kepastian yang masuk akal, karena seseorang yang menyediakan kepastian yang lengkap akan sangat sulit untuk merancang dan akan sangat mahal.

Pengendalian internal menampilkan 3 fungsi penting.

1. ***Preventive controls*** (pengendalian pencegahan) mencegah masalah sebelum terjadi. Mempekerjakan personel akuntansi yang berkualifikasi tinggi, memisahkan dengan baik tugas karyawan, dan secara efektif mengontrol akses fisik terhadap *assets*. Karena tidak semua kontrol masalah dapat dicegah,
2. ***Detective controls*** (pengendalian penyelidikan) diperlukan untuk menemukan masalah segera setelah mereka timbul. Contoh dari *detective*

controls adalah memeriksa duplikasi dari perhitungan dan menyiapkan rekonsiliasi bank dan neraca saldo bulanan.

3. ***Corrective controls*** (pengendalian perbaikan) memperbaiki masalah yang ditemukan oleh *detective controls*. Mereka termasuk prosedur yang diambil untuk mengidentifikasi penyebab dari masalah, memperbaiki penyebab masalah dan kesulitan, dan modifikasi sistem sehingga masalah yang akan datang dapat diminimalisasi atau dihilangkan. Contohnya termasuk mengelola *backup copies* dari transaksi kunci dan *master files*, menyesuaikannya dengan prosedur untuk perbaikan *data entry errors*.

2.4.2. Unsur Sistem Pengendalian Intern

Mulyadi (2001, h.165) menuliskan, unsur pokok sistem pengendalian intern adalah:

1. Struktur organisasi yang memisahkan tanggung jawab fungsional secara tegas.

Pembagian tanggung jawab fungsional dalam organisasi ini didasarkan pada prinsip-prinsip berikut ini:

- a. Fungsi-fungsi operasi dan penyimpanan harus dipisahkan dari fungsi akuntansi. Fungsi operasi adalah fungsi yang memiliki wewenang untuk melaksanakan suatu kegiatan. Fungsi penyimpanan adalah fungsi yang memiliki wewenang untuk menyimpan aktiva perusahaan. Fungsi akuntansi adalah fungsi yang memiliki wewenang untuk mencatat peristiwa keuangan perusahaan.

- b. Suatu fungsi tidak boleh diberi tanggung jawab penuh untuk melaksanakan semua tahap suatu transaksi.
2. Sistem wewenang dan prosedur pencatatan memberikan perlindungan yang cukup terhadap kekayaan, utang, pendapatan dan biaya.
3. Praktik yang sehat dalam melaksanakan tugas dan fungsi setiap unit organisasi.
 - a. Penggunaan formulir bernomor urut tercetak yang pemakaiannya harus dipertanggungjawabkan oleh yang berwenang.
 - b. Pemeriksaan mendadak (*surprised audit*).
 - c. Setiap transaksi tidak boleh dilaksanakan dari awal sampai akhir oleh satu orang atau satu unit organisasi, tanpa ada campur tangan dari orang atau unit organisasi lain.
 - d. Perputaran jabatan (*job rotation*) diadakan secara rutin akan dapat menjaga independensi pejabat dalam melakukan tugasnya.
 - e. Keharusan pengambilan cuti bagi karyawan yang berhak.
 - f. Secara periodik diadakan rekonsiliasi atau pencocokan fisik kekayaan dengan catatannya.
 - g. Pembentukan unit organisasi yang bertugas untuk mengecek efektivitas unsur-unsur sistem pengendalian intern yang lain.
4. Karyawan yang mutunya sesuai dengan tanggung jawabnya.

2.4.3. COSO Internal Control Framework

Committee of Sponsoring Organizations (COSO) adalah kelompok ikatan profesi yang terdiri dari American Accounting Association, AICPA, Institute of

Internal Auditors, Institute of Management Accountants, dan Financial Executives Institute. Di tahun 1992, COSO menerbitkan hasil studi untuk mengembangkan sebuah definisi dari *internal controls* dan menyediakan panduan untuk mengevaluasi sistem pengendalian intern. Lapornya telah diterima secara luas sebagai otoritas atas pengendalian internal.

COSO mendefinisikan pengendalian intern sebagai proses yang diimplementasi oleh dewan direksi, manajemen, dan mereka yang bertanggung jawab menyediakan kepastian yang masuk akal yang mengontrol tujuan-tujuan untuk mengikuti:

1. Efektivitas dan efisiensi operasi.
2. Keandalan dari pelaporan keuangan
3. Kepatuhan terhadap hukum dan regulasi yang ada.

Menurut COSO yang disebutkan oleh Romney dan Steinbart di dalam bukunya (2003, p.196), pengendalian intern adalah suatu proses karena memenuhi sebuah aktivitas operasional organisasi dan adalah bagian integral dari dasar aktivitas manajemen. Pengendalian intern menyediakan kepastian yang masuk akal daripada absolut, karena kemungkinan kesalahan manusia, kolusi, dan kesalahan pengelolaan manajemen membuat proses ini sebagai yang tidak sempurna.

Model *internal control* COSO memiliki 5 komponen penting, yaitu:

1. ***Control environment***

Pusat dari segala bisnis terletak pada orang-orangnya – atribut individualnya, termasuk integritas, nilai-nilai etis, dan kompetensi – dan lingkungan dimana mereka beroperasi.

2. *Control activities*

Peraturan pengendalian dan prosedur harus diterapkan dan dieksekusi untuk membantu memastikan bahwa tindakan yang diidentifikasi manajemen adalah perlu untuk mengarahkan resiko menuju pencapaian dari tujuan organisasi.

3. *Risk Assessment*

Organisasi harus waspada dan berkepatutan dengan resiko yang akan dihadapinya. Dia harus menetapkan tujuan, mengintegrasikannya dengan penjualan, produksi, pemasaran, keuangan, dan aktivitas lain sehingga organisasi beroperasi dengan benar. Juga harus diterapkan mekanisme untuk mengidentifikasi, menganalisa, dan mengelola resiko yang berhubungan. Di dalam bukunya, Idroes dan Sugiarto (2006, p.131) menyebutkan bahwa kesepakatan Basel II secara spesifik mendefinisikan resiko operasional sebagai resiko dari kerugian atau ketidakcukupan dari proses intern, manusia dan sistem yang gagal atau dari peristiwa eksternal.

4. *Information and communication*

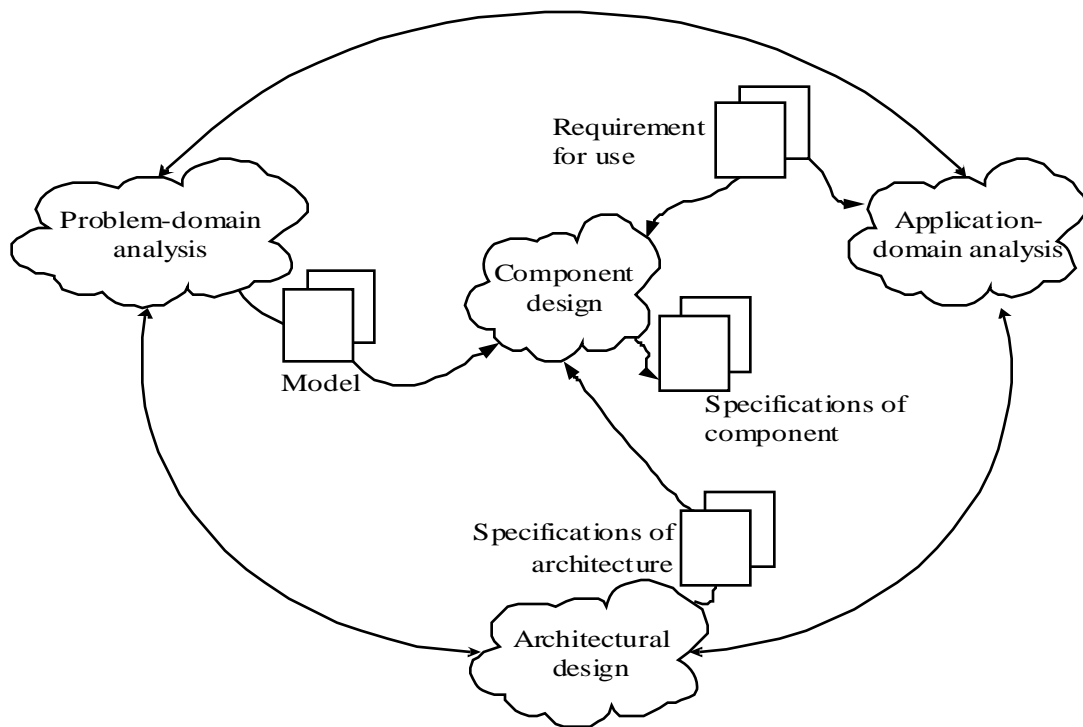
Yang mengelilingi aktivitas kontrol adalah sistem informasi dan komunikasi. Mereka memungkinkan orang-orang di dalam organisasi menangkap dan menukarkan informasi yang diperlukan untuk mengadakan, mengelola, dan mengendalikan operasinya.

5. *Monitoring*

Keseluruhan proses harus dimonitor dan modifikasi dibuat bila diperlukan. Dalam hal ini sistem dapat bereaksi secara dinamis, berubah sesuai kondisi.

2.5. Konsep Analisis dan Perancangan Berorientasi Objek

Mathiassen et al. (2000, p.14) mengutarakan pendapatnya bahwa *Object Oriented Analysis and Design* terbagi kedalam empat aktivitas, antara lain: analisis *problem-domain*, analisis *application domain*, *architecture design*, dan *component design* seperti dapat dilihat pada **Gambar 2.1**.



Gambar 2.1. Kegiatan utama dan hasil dari analisa dan perancangan orientasi objek

2.5.1. Object

Menurut Mathiassen et al. (2000, p.4), objek adalah suatu entitas dengan identitas, keadaan dan sifat tertentu. Jadi dapat disimpulkan *object* adalah sesuatu yang dapat dilihat, disentuh, atau dapat dirasakan dimana *user* dapat menyimpan data dan berasosiasi dengan *behaviour*.

2.5.2. Rich Picture

Mengacu pada Mathiassen et al. (2000, p.26), *rich picture* adalah sebuah gambaran informal yang digunakan oleh pengembang sistem untuk menyatakan pemahaman mereka terhadap situasi dari sistem yang sedang berlangsung. *Rich picture* juga dapat digunakan sebagai alat yang berguna untuk memfasilitasi komunikasi yang baik antara pengguna dalam sistem.

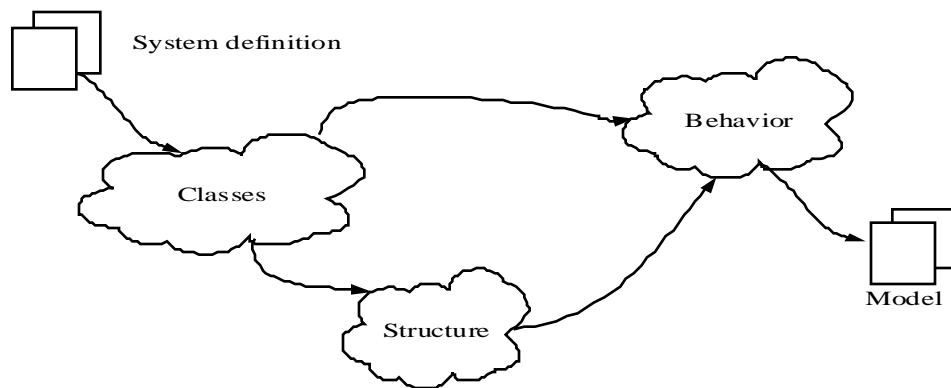
2.5.3. System Definition

Mathiassen et al. (2000, p.38) menyebutkan bahwa, sebuah definisi sistem mendeskripsikan suatu solusi terkomputerisasi di dalam *context*. Kriteria FACTOR (Functionality, Application Domain, Conditions, Technology, Objects, and Responsibility) dapat digunakan untuk mendukung pengembangan *system definition*.

2.5.4. Problem Domain Analysis

Mathiassen et al. (2000, p.45) mengartikan bahwa *problem domain* merupakan bagian dari *context* yang diadministrasikan, dimonitor, dikontrol oleh sebuah sistem dengan tujuan untuk mengembangkan sebuah model. Sedangkan,

dapat didefinisikan bahwa model adalah deskripsi dari *class-class*, *object-object*, *structure-structure*, dan *behaviour* di dalam sebuah *problem domain*. Seperti yang terlihat pada **Gambar 2.2.** dibawah ini.



Gambar 2.2 Aktivitas pada *problem domain analysis*

2.5.4.1. *Classes*

Classes disini akan menggambarkan tentang *object-object* dan *event-event* yang mana saja yang akan menjadi bagian dari *problem domain*. Menurut Mathiassen et al. (2000, p.53), kelas adalah kumpulan dari objek yang berbagi atribut dan behavior yang sama.

Mengacu pada Mathiassen et al. (2000, p.49), kegiatan kelas akan menghasilkan *event table*. Dalam tabel ini dimensi horizontal berisi kelas-kelas yang terpilih, dimensi vertikal berisi *event-event* terpilih, dan tanda cek digunakan untuk mengidentifikasi objek-objek dari kelas yang berhubungan dalam *event* tertentu.

2.5.4.2. *Structure*

Structure di sini harus mencerminkan bagaimana *class-class* dan *object-object* secara konseptual saling terkait secara bersamaan.

Menurut Mathiassen et al. (2000, p.69) ada empat tipe hubungan struktural dimana keempatnya dibagi ke dalam dua bagian yaitu:

1. *Class structure*, yang meliputi:

a. *Generalization*

Generalisasi adalah suatu kelas yang umum (kelas super) yang menggambarkan properti umum untuk suatu grup yang memiliki kelas khusus (sub kelas).

b. *Cluster*

Cluster adalah suatu koleksi dari kelas yang berhubungan.

2. *Object structure*, yang meliputi:

a. *Aggregation*

Agregasi adalah suatu objek superior (keseluruhan) yang berisi jumlah dari objek atau bagiannya.

b. *Association*

Asosiasi adalah hubungan yang berarti antar sejumlah objek.

Hasil dari kegiatan stuktur ini adalah *class diagram*. *Class Diagram* menghasilkan ringkasan model *problem-domain* yang jelas dengan menggambarkan semua struktur hubungan statik antar kelas dan objek yang ada dalam model dari sistem yang berubah-ubah.

2.5.4.3. Behavior

Behavior di sini menggambarkan mengenai suatu tujuan, yaitu untuk memberi model dinamis yang harus dipunyai oleh *object-object* pada *problem domain*. Tugas utama dalam kegiatan ini adalah menggambarkan pola perilaku

(*behavior pattern*) dan atribut dari setiap kelas. Hasil dari kegiatan ini adalah *statechart diagram*.

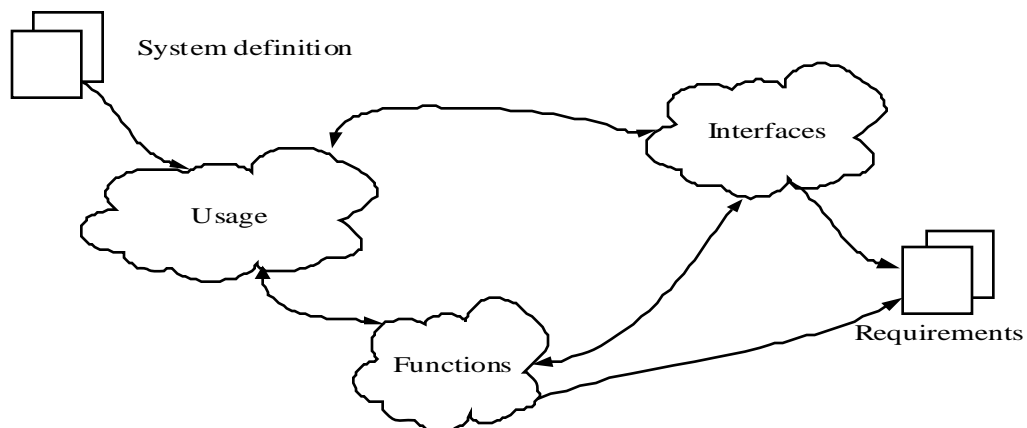
Menurut Mathiassen et al. (2000, p.93) ada 3 notasi untuk *behavior pattern* yaitu:

- a. *Sequence*, dimana *event* muncul satu per satu secara berurutan.
- b. *Selection*, dimana terjadi pemilihan satu *event* dari sekumpulan *event* yang muncul.
- c. *Iteration*, dimana sebuah *event* muncul sebanyak nol atau berulang kali.

2.5.5. *Application Domain Analysis*

Mathiassen et al. (2000, p.115) berpendapat bahwa *application domain* adalah suatu organisasi yang mengadministrasikan, memantau atau mengendalikan *problem domain*. Tujuan dari *application domain* adalah untuk menganalisis kebutuhan dari pengguna sistem.

Pada *application domain* terdapat tiga aktivitas utama seperti yang ditunjukkan pada **Gambar 2.3.** dibawah ini:



Gambar 2.3. Aktivitas *application-domain*

2.5.5.1. Usage

Di dalam *usage* harus mencerminkan bagaimana sistem berinteraksi dengan *actor* di dalam sebuah *context*. Definisi *actor* itu sendiri menurut Mathiassen et al. (2000, p.119) adalah suatu abstraksi dari pengguna atau sistem lain yang berhubungan dengan sasaran dari sistem, sedangkan pengertian *use case* menurut Mathiassen et al. (2000, p.120) adalah suatu pola dari interaksi antara sistem dan *actor* dari *application domain*.

Hasil dari analisis kegiatan *usage* ini adalah deskripsi lengkap dari semua *use case* dan *actor* yang digambarkan dalam *actor table* dan *use case diagram*.

Menurut Bennet (2006, p.148) *use case diagram* mempunyai dua jenis hubungan (*relationship*) yaitu: *extend* dan *include*. Hubungan *extend* digunakan ketika ingin menunjukkan bahwa sebuah *use case* menyediakan fungsi tambahan yang mungkin digunakan oleh *use case* lain, sedangkan hubungan *include* digunakan ketika terdapat urutan *behavior* yang sering kali digunakan oleh sejumlah *use case* dan ingin dihindari pengkopian deskripsi yang sama ke setiap *use case* yang akan menggunakan perilaku tersebut.

Sequence diagram adalah diagram yang menggambarkan bagaimana antar objek berinteraksi dalam pelaksanaan sebuah *use case*. *Sequence diagram* menggambarkan bagaimana pesan atau *message* dikirim dan diterima antar objek dalam *sequence* tertentu.

2.5.5.2. Function

Menurut Mathiassen et al. (2000, p.138), *function* adalah suatu fasilitas untuk membuat suatu model berguna bagi *actors*. *Function* memfokuskan pada

bagaimana cara sebuah sistem dapat membantu aktor dalam melaksanakan pekerjaan mereka. *Function* memiliki 4 tipe berbeda yaitu:

- a. *Update*, fungsi ini disebabkan oleh *event problem domain* dan menghasilkan perubahan dalam *state* atau keadaan dari model tersebut.
- b. *Signal*, fungsi ini disebabkan oleh perubahan keadaan atau *state* dari model yang dapat menghasilkan reaksi pada *context*.
- c. *Read*, fungsi ini disebabkan oleh kebutuhan informasi dalam pekerjaan aktor dan mengakibatkan sistem menampilkan bagian yang berhubungan dengan informasi dalam model.
- d. *Compute*, fungsi ini disebabkan oleh kebutuhan informasi dalam pekerjaan aktor dan berisi perhitungan yang melibatkan informasi yang disebabkan oleh aktor atau model, hasil dari fungsi ini adalah tampilan dari hasil komputasi.

Tujuan dari kegiatan *function* adalah untuk menentukan kemampuan sistem memproses informasi. Hasil dari kegiatan ini adalah sebuah *function list* yang kompleks. Daftar *function* harus lengkap, menyatakan kebutuhan kolektif dari *actor* serta harus konsisten dengan *use case*.

2.5.5.3. User Interface

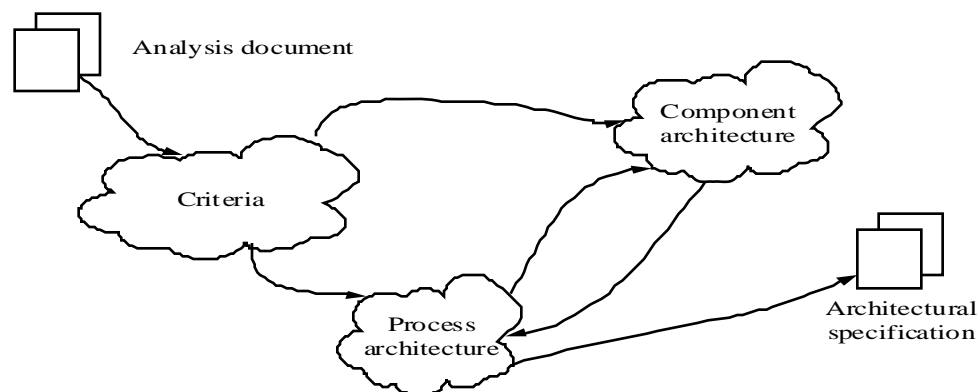
Mathiassen et al. (2000, p.151) menuliskan *interfaces* adalah fasilitas yang membuat suatu model dan fungsi yang dapat dipakai oleh pengguna. *Interface* menghubungkan sistem dengan semua aktor yang berhubungan dalam *context*. Kualitas *user interface* ditentukan oleh kegunaan atau *usability interface* tersebut bagi *user*.

Hasil dari kegiatan ini adalah sebuah deskripsi elemen-elemen *user interface* dan elemen-elemen sistem *interface* yang lengkap, dimana kelengkapan

menunjukkan pemenuhan kebutuhan *user*. Hasil ini dilengkapi dengan sebuah *navigation diagram* yang menyediakan sebuah ringkasan dari elemen-elemen *user interface* dan perubahan antara elemen-elemen tersebut.

2.5.6. *Architectural Design*

Menurut Mathiassen et al. (2000, p.173), tujuan dari *architectural design* adalah untuk menstrukturkan sebuah sistem yang terkomputerisasi. Aktivitas yang dilakukan dalam *architectural design* seperti yang diilustrasikan pada **Gambar 2.4.** dibawah ini.



Gambar 2.4. Aktifitas pada *architectural design*

2.5.6.1. *Criteria*

Mathiassen et al. (2000, p.178) merekomendasikan penggunaan *list of classical criteria for software quality* seperti terlihat pada **Tabel 2.1.** berikut ini untuk menetapkan prioritas pada *architectural design*.

Criterion	Measure of
<i>Usable</i>	Kemampuan sistem untuk menyesuaikan diri dengan konteks, organisasi yang berhubungan dengan pekerjaan dan teknis
<i>Secure</i>	Ukuran keamanan sistem dalam menghadapi akses yang tidak terotorisasi terhadap data dan fasilitas
<i>Efficient</i>	Eksplorasi ekonomis terhadap fasilitas platform teknis
<i>Correct</i>	Pemenuhan dari kebutuhan
<i>Reliable</i>	Pemenuhan ketepatan yang dibutuhkan untuk melaksanakan fungsi
<i>Maintainable</i>	Biaya untuk menemukan dan memperbaiki kerusakan
<i>Testable</i>	Biaya untuk memastikan bahwa sistem yang dibentuk dapat melaksanakan fungsi yang dibentuk
<i>Flexible</i>	Biaya untuk mengubah sistem yang dibentuk
<i>Comprehensible</i>	Usaha yang diperlukan untuk mendapatkan pemahaman terhadap sistem
<i>Reusable</i>	Kemungkinan untuk menggunakan bagian sistem pada sistem lain yang berhubungan
<i>Portable</i>	Biaya untuk memindahkan sistem ke platform teknis yang berbeda
<i>Interoperable</i>	Biaya untuk menggabungkan sistem ke sistem yang lain

Tabel 2.1. *Classical criteria for software quality*

Tidak ada ukuran dan cara-cara yang pasti untuk menghasilkan suatu rancangan yang baik. Menurut Mathiassen et al. (2000, p.177), prinsip-prinsip berikut merupakan spesifikasi perancangan *criteria* dalam sebuah proyek pengembangan *software*:

1) Tidak memiliki kelemahan

Syarat ini menyebabkan adanya pendekatan pada evaluasi dari kualitas berdasarkan *review* atau eksperimen dan membantu dalam menentukan prioritas dari *criteria* yang akan mengatur dalam kegiatan desain.

2) Menyeimbangkan beberapa kriteria

Konflik sering terjadi antar *criteria*, oleh sebab itu untuk menentukan *criteria* mana yang akan diutamakan dan bagaimana cara untuk menyeimbangkannya dengan kriteria-kriteria yang lain bergantung pada situasi sistem tertentu.

3) *Usable, flexible, dan comprehensible*

Kriteria-kriteria ini bersifat universal dan digunakan pada hampir setiap proyek pengembangan sistem.

2.5.6.2. *Component Architecture*

Mathiassen et al. (2000, p.190), mengutarakan pendapatnya bahwa arsitektur komponen adalah suatu struktur sistem yang berhubungan dengan komponen.

Beberapa pola umum dalam desain komponen arsitektur:

1. Arsitektur *layered*

Merupakan bentuk yang paling umum dalam software. Contoh dari pola ini adalah model OSI yang sudah menjadi ISO untuk model jaringan.

2. Arsitektur *generic*

Pola ini digunakan untuk merinci sistem dasar yang terdiri dari antar muka, *function*, dan komponen-komponen model.

3. Arsitektur *client-server*

Pola ini awalnya dikembangkan untuk mengatasi masalah distribusi sistem di antara beberapa prosesor yang tersebar secara geografis. Komponen pada arsitektur ini adalah sebuah *server* dan beberapa *client*. Tanggung jawab daripada *server* adalah untuk menyediakan *database* dan *resource* yang dapat disebarkan kepada *client* melalui jaringan. Sementara *client* memiliki tanggung jawab untuk menyediakan antarmuka lokal untuk setiap penggunaanya. Berikut adalah **Tabel 2.2.** yang berisi beberapa jenis distribusi dalam arsitektur *client-server* dimana U (*user*), F (*function*), dan M (*model*).

<i>Client</i>	<i>Server</i>	<i>Architecture</i>
U	U+F+M	Distributed presentation
U	F+M	Local presentation
U+F	F+M	Distributed functionality
U+F	M	Centralized data
U+F+M	M	Distributed data

Tabel 2.2. Jenis Arsitektur *client-server*

2.5.6.3. Process Architecture

Definisi *process architecture* menurut Mathiassen et al. (2000, p.209), adalah struktur eksekusi sistem yang terdiri dari proses yang saling bergantung. Apabila ditemukan potensi *bottlenecks streaming* dari penggunaan sumber daya komputer yang digunakan bersama-sama (*shared use of computer resources*) seperti *processor*, *program component*, dan *external device*, maka rancangan dapat diubah atau dapat mengganti *hardware configuration* dengan menggunakan lebih banyak *processors* atau dengan *processors* yang lebih kuat.

Terdapat tiga jenis *distribution pattern* yang menggambarkan situasi dimana sistem didistribusikan pada sejumlah *processor* yang terhubung dengan suatu jaringan:

1. The centralized pattern

Menyimpan semua data pada *server* pusat dan *clients* menangani hanya *user interface*.

2. The distributed pattern

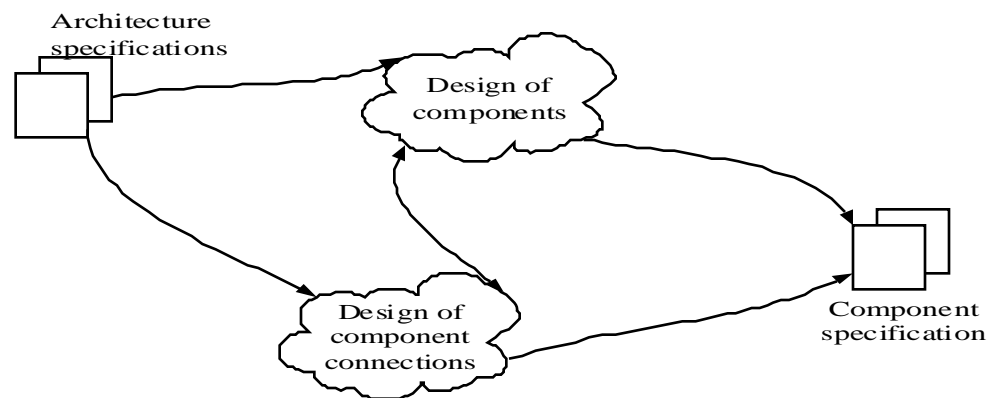
Semuanya didistribusikan pada *clients* dan *server* diperlukan hanya untuk menyiarkan *model* yang di-*update* diantara *clients*. Sebuah *copy* dari *model* yang lengkap berada pada setiap *client*.

3. The decentralized pattern

Clients memiliki data mereka masing-masing, hanya data umum yang berada di *server*. *Server* memegang *common model* dan *functions*, dimana *clients* memegang data miliknya yang merupakan bagian dari *application domain*.

2.5.7. Component Design

Tujuan dari *component design* adalah untuk menentukan kebutuhan di dalam kerangka arsitektur. *Component design* diilustrasikan pada **Gambar 2.5.** dibawah ini.



Gambar 2.5. *Component Design*

2.5.7.1. Model Component

Mathiassen et al. (2000, p.236) menuliskan *model component* adalah suatu bagian dari sistem yang mengimplementasikan *problem domain*.

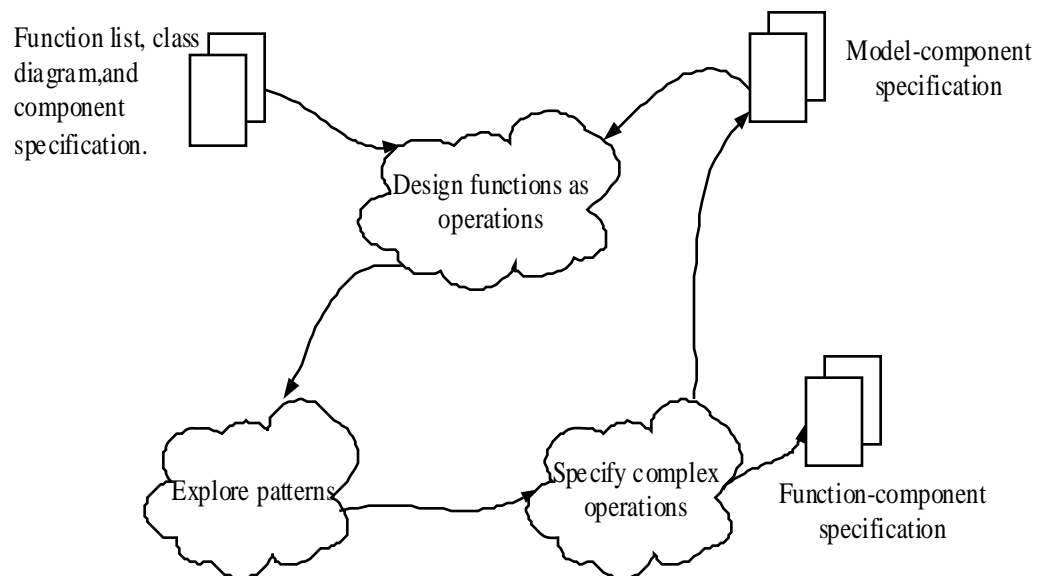
Tujuan dari komponen model adalah untuk mengirimkan data sekarang dan historik ke *function*, *interface* dan pengguna dan sistem yang lain. Konsep utama dalam desain komponen model adalah struktur.

Hasil dari kegiatan komponen model adalah *revised class diagram* dari kegiatan analisis. Kegiatan revisi biasanya terdiri dari kegiatan menambahkan kelas, atribut dan struktur baru yang mewakili *event*.

2.5.7.2. Function Component

Menurut Mathiassen et al. (2000, p.252), komponen *function* adalah bagian dari sistem yang mengimplementasikan kebutuhan fungsional. Tujuan dari komponen *function* adalah untuk memberikan akses bagi *user interface* dan komponen sistem lainnya ke model, oleh karena itu komponen *function* adalah penghubung antara model dan *usage*.

Hasil dari kegiatan ini adalah *class diagram* untuk komponen *function* dan perpanjangan dari *class diagram* komponen model. Berikut adalah sub kegiatan dalam perancangan komponen *function* dapat dilihat pada **Gambar 2.6.** dibawah ini:



Gambar 2.6. Sub activities in function-component design

Sub kegiatan ini menghasilkan kumpulan operasi yang dapat mengimplementasikan fungsi sistem seperti yang ditentukan dalam *analysis problem domain* dan *function list*.

- a. Merancang *function* sebagai *operation*.
- b. Menelusuri pola yang dapat membantu dalam mengimplementasi *function* sebagai *operation*.
- c. Spesifikasikan operasi yang kompleks