BAB II

PEMBAHASAN TEORI

2.1. Pengertian Sistem

Pengertian dan definisi sistem pada berbagai bidang berbeda-beda, tetapi meskipun istilah *sistem* yang digunakan bervariasi, semua sistem pada bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

Berdasarkan persyaratan ini, sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Kumpulan elemen terdiri dari manusia, mesin, prosedur, dokumen, data atau elemen lain yang terorganisir dari elemen-elemen tersebut. Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya.

Terdapat beberapa pengertian sistem menurut para ahli, diantaranya:

1. Gordon B. Davis (1984) berpendapat bahwa sebuah sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud.

2. Raymond Mcleod (2001) berpendapat bahwa sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu.

2.1.1. Elemen Sistem

Sistem informasi terdiri dari elemen-elemen yang terdiri dari orang, prosedur, perangkat keras, perangkat lunak, basis data, jaringan komputer dan komunikasi data. Semua elemen ini merupakan komponen fisik.

1. Orang

   Orang atau personil yang di maksudkan yaitu operator komputer, analis sistem, programmer, personil data entry, dan manajer sistem informasi.

2. Prosedur

   Prosedur merupakan elemen fisik. Hal ini di sebabkan karena prosedur disediakan dalam bentuk fisik seperti buku panduan dan instruksi. Ada 3 jenis prosedur yang dibutuhkan, yaitu instruksi untuk pemakai, instruksi untuk penyiapan masukan, instruksi pengoperasian untuk karyawan pusat komputer.

3. Perangkat Keras

   Perangkat keras bagi suatu sistem informasi terdiri atas komputer (pusat pengolah, unit masukan/keluaran), peralatan penyiapan data, dan terminal masukan/keluaran.

4. Perangkat Lunak

   Perangkat lunak dapat dibagi dalam 3 jenis utama :

   a. Sistem perangkat lunak umum, seperti sistem pengoperasian dan sistem manajemen data yang memungkinkan pengoperasian sistem komputer.
   b. Aplikasi perangkat lunak umum, seperti model analisis dan keputusan.
   c. Aplikasi perangkat lunak yang terdiri atas program yang secara spesifik dibuat untuk setiap aplikasi.

5. Basis Data

   File yang berisi program dan data dibuktikan dengan adanya media penyimpanan secara fisik seperti *diskette, hard disk, magnetic tape,* dan sebagainya. *File* juga meliputi keluaran tercetak dan catatan lain diatas kertas, mikro film, dan lain sebagainya.

6. Jaringan Komputer

   Jaringan komputer adalah sebuah kumpulan komputer, *printer* dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data.

7. Komunikasi Data

   Komunikasi data adalah merupakan bagian dari telekomunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer-komputer dan pirant-piranti yang lain dalam bentuk digital yang dikirimkan melalui media komunikasi data. Data berarti informasi yang disajikan oleh isyarat digital. Komunikasi data merupakan bagian vital dari suatu sistem informasi karena sistem ini menyediakan infrastruktur yang memungkinkan komputer-komputer dapat berkomunikasi satu sama lain.

2.1.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. Komponen-komponen

   Komponen sistem atau elemen sistem dapat berupa :

   a. Elemen-elemen yang lebih kecil yang disebut *sub-system,* misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak, dan manusia.
   b. Elemen-elemen yang lebih besar yang disebut *supra system*. Misalkan bila perangkat keras adalah sistem yang memiliki *sub-system* CPU, perangkat I/O dan memori, maka *supra system* perangkat keras adalah sistem komputer.

2. Batas sistem

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem

   Lingkungan luar dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. lingkungan luar yang mengutungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung

   Penghubung merupakan media perantara antar *sub-system*. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu *sub-system* ke *sub-system* lainnya. *Output* dari satu *sub-system* akan menjadi *input* untuk *sub-system* yang lainnya dengan melalui penghubung. Dengan penghubung satu *sub-system* dapat berinteraksi dengan *sub-system* yang lainnya membentuk satu kesatuan.

5. Masukan

   Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa *maintenance input* dan *signal input*. *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk *sub-system* yang lain atau kepada *supra system*.

7. Pengolah

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran atau tujuan

   Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

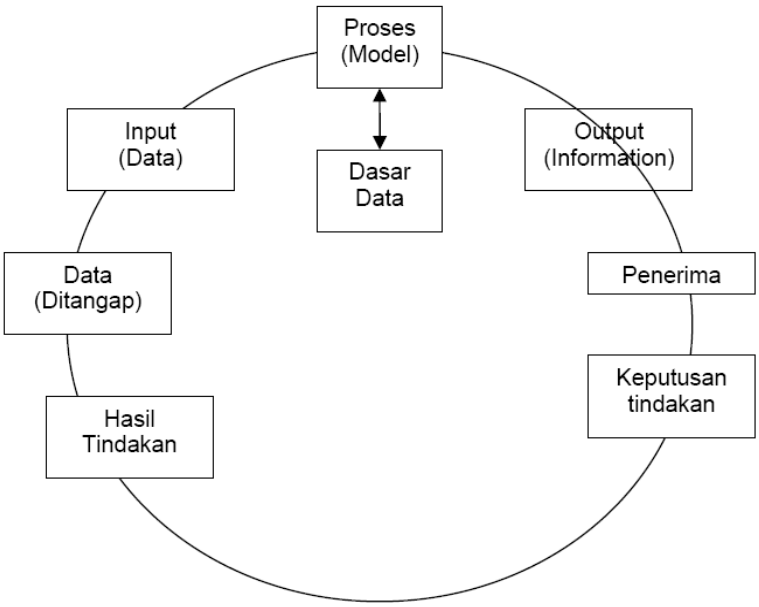
2.2. Pengertian Informasi

Di dalam suatu organisasi atau perusahaan, informasi merupakan sesuatu yang memiliki arti yang sangat penting didalam mendukung proses pengambilan keputusan oleh pihak manajemen. Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya.

Menurut Raymond Mcleod, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian adalah sesuatu yang terjadi pada saat tertentu. Di dalam dunia bisnis, kejadian-kejadian yang sering terjadi adalah transaksi perubahan dari suatu nilai yang disebut transaksi. Kesatuan nyata adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul-betul ada dan terjadi.

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dsb. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sabagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus informasi ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Siklus Informasi
Sumber: Dokumentasi laporan kerja praktek

Adapun fungsi-fungsi informasi adalah sebagai berikut :

1. Untuk meningkatkan pengetahuan bagi pemakai

2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai
3. Menggambarkan keadaan sebenarnya dari suatu hal

Informasi yang berkualitas harus akurat, tepat pada waktunya, dan relevan.

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat Waktu

   Berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal bagi organisasi. Saat ini mahalnya nilai informasi disebabkan harus cepatnya informasi itu didapat sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3. Relevan

   Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang berbeda-beda.

2.3. Pengertian Sistem Informasi

Pada saat ini dunia industri dan bisnis memerlukan informasi yang tepat, cepat dan relevan. Untuk mendapatkan informasi yang diinginkan tentunya harus menggunakan sistem informasi. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

Menurut Raymond Mcleod, Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi.

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan. Informasi dalam suatu lingkungan sistem informasi harus mempunyai persyaratan umum sebagai berikut :

1. harus diketahui oleh penerima sebagai referensi yang tepat

2. harus sesuai dengan kebutuhan yang ada dalam proses pembuatan / pengambilan keputusan
3. harus mempunyai nilai *surprise*, yaitu hal yang sudah diketahui hendaknya jangan diberikan
4. harus dapat menuntun pemakai untuk membuat keputusan. Suatu keputusan tidak selalu menuntut adanya tindakan.

Sistem informasi harus mempunyai beberapa sifat seperti:

1. Pemrosesan informasi yang efektif. Hal ini berhubungan dengan pengujian terhadap data yang masuk, pemakaian perangkat keras dan perangkat lunak yang sesuai
2. Manajemen informasi yang efektif. Dengan kata lain, operasi manajemen, keamanan dan keutuhan data yang ada harus diperhatikan
3. Keluwesan. Sistem informasi hendaknya cukup luwes untuk menangani suatu macam operasi
4. Kepuasan pemakai. Hal yang paling penting adalah pemakai mengetahui dan puas terhadap sistem informasi.

2.3.1. Komponen Sistem Informasi

Sistem informasi mempunyai enam buah komponen atau disebut juga dengan blok bangunan (building block), yaitu:

1. Komponen input
2. Komponen model
3. Komponen output
4. Komponen teknologi

5. Komponen basisdata
6. Komponen control

Keenam komponen ini harus ada bersama-sama dan membentuk satu kesatuan. Jika satu atau lebih komponen tersebut tidak ada, maka sistem informasi tidak dapat melakukan fungsinya, yaitu pengolahan data dan tidak dapat mencapai tujuannya, yaitu menghasilkan informasi yang relevan, tepat waktu, dan akurat. Komponen-komponen dari sistem informasi ini dapat digambarkan sebagai berikut :

1. Blok Masukan (*Input Block*)

   Input merupakan data yang masuk ke dalam sistem informasi.

2. Blok Model (*Model Block*)

   Kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

   Keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

   Teknologi merupakan kota alat (*toolbox*) dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara menyeluruh.

5. Blok Basis Data (*Database Block*)

   Merupakan sekumpulan data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*Control Block*)

   Beberapa pengendalian yang dirancang secara khusus untuk menanggulangi gangguan-gangguan terhadap sistem.

2.4. Metode Pengembangan Sistem Informasi

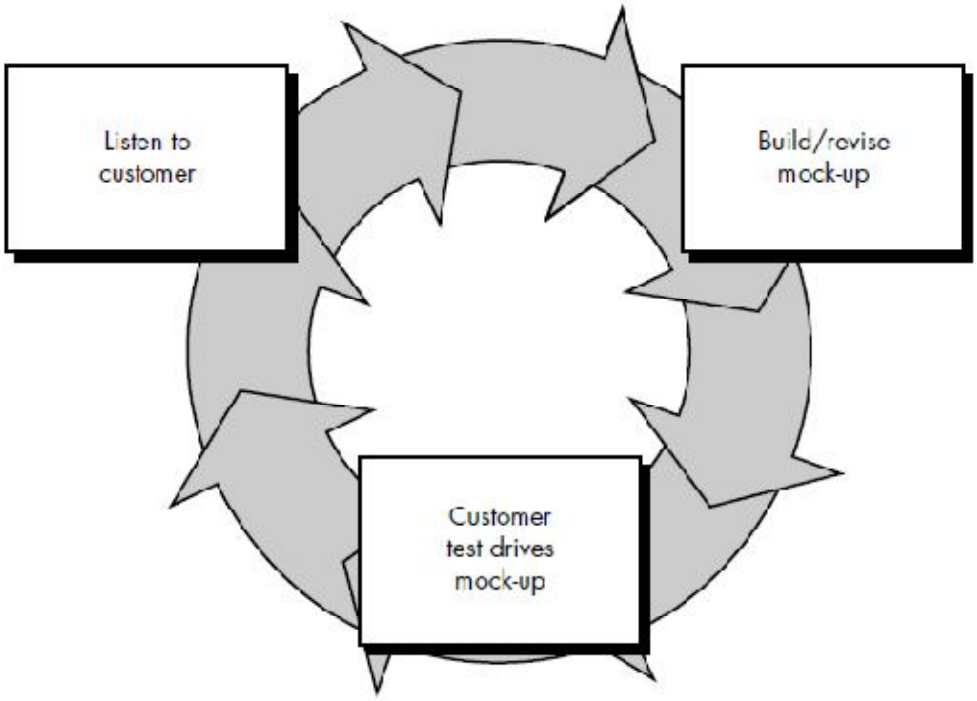
Dalam pembangunan apapun khususnya pembangunan sistem informasi dalam pelaksanaannya pada awalnya perlu mengidentifikasi masalah yang ada. Selanjutnya ditentukan metode yang sesuai untuk menyelesaikan masalah yang timbul. Pada pembangunan sistem informasi, salah satu metode yang dipakai untuk penyelesaian masalah disebut metode pengembangan sistem. Metode pengembangan sistem ini sendiri terdiri dari beberapa metode. Namun, yang akan penulis jabarkan hanyalah metode-

metode pengembangan sistem yang bisa dijadikan referensi bagi pembangunan sistem informasi yang sedang penulis kerjakan, yaitu *Tracking of Software Licensed System* dan *Helpdesk System*. Diantaranya adalah Model Prototipe (*Prototyping Model*), RAD Model (*Rapid Application Development Model*).

2.4.1. Model Prototipe (*Prototyping Model*)

Seringkali, pengguna sistem mendefinisikan secara umum tujuan dari *software* yang diinginkan, tetapi tidak mengidentifikasi secara detail *input*, proses, atau *output* yang dibutuhkan. Di lain kasus, seorang pengembang sistem mungkin tidak merasa yakin dengan efisiensi dari algoritma, kemampuan beradaptasi sebuah sistem operasi, atau bentuk interaksi yang seharusnya terjadi antara manusia dan mesin. Dalam hal ini, paradigma prototipe bisa menawarkan pendekatan yang terbaik.

Model Prototipe memulai dengan pengumpulan kebutuhan sistem. Pengembang dan pengguna sistem bertemu untuk mendefinisikan secara kesuluruhan tujuan *software* yang diinginkan, mengidentifikasi kebutuhan apa saja yang diketahui, dan secara garis besar dimana definisi lebih lanjut adalah wajib. Sebuah “rancangan cepat” kemudian dikerjakan.



Gambar 2.2 Prototyping Model
Sumber: Software Engineering a Practitioner's Approach (Roger S.Pressman) Hlm.31

Rancangan cepat ini berfokus pada representasi dari keseluruhan aspek *software* yang akan diperlihatkan kepada pengguna sistem. Rancangan cepat menuntun kea rah pembangunan sebuah prototipe. Protipe tersebut selanjutnya dievaluasi oleh pengguna sistem dan digunakan untuk memperjelas kebutuhan *software* yang akan dikembangkan.

Idealnya, prototipe menyediakan sebuah mekanisme untuk mengidentifikasi kebutuhan *software*. Jika prototipe yang sedang berjalan dibangun, pengembang sistem berusaha menggunakan program fragmen yang ada, atau menerapkan *tools* (misal, *report generator*, *window*

managers) yang mengaktifkan program yang berjalan yang dapat di-*generate* dengan cepat.

Adapun kelebihan dari Model Prototipe ini diantaranya sebagai berikut:

1. “Mengalamatkan ketidak-mampuan banyak pengguna sistem untuk merinci kebutuhan informasi mereka, dan kesulitan *system analysts* untuk memahami lingkungan pengguna sistem, dengan menyediakan pengguna sistem dengan sistem sementara untuk tujuan percobaan di awal waktu yang memungkinkan.” (Janson and Smith, 1985)
2. “Dapat digunakan untuk model realistis aspek penting dari sebuah sistem selama setiap fase dari siklus kehidupan sistem tradisional.” (Huffaker, 1986)
3. Meningkatkan partisipasi *user* di dalam pengembangan sistem dan komunikasi diantara *project stakeholder*.
4. Khususnya berguna untuk penyelesaian tujuan yang tidak jelas; pengembangan dan validasi kebutuhan *user*; percobaan dengan membandingkan berbagai macam solusi perancangan; atau melakukan investigasi terhadap kinerja dan *interface computer*.
5. Menyediakan implementasi cepat yang kurang lengkap, tapi fungsional, dan aplikatif.

Sedangkan kelemahan dari Model Prototipe diantaranya sebagai berikut:

1. Proses persetujuan dan pengendalian tidak ketat

2. Analisis masalah lengkap atau tidak memadai dapat terjadi dimana hanya kebutuhan yang paling jelas dan dangkal akan dibahas, sehingga dalam praktek yang tidak efisien saat ini yang mudah dibangun ke dalam sistem baru.
3. Kebutuhan sering berubah secara signifikan.
4. Identifikasi elemen-elemen non-fungsional sulit untuk didokumentasikan
5. *Designer* mungkin membuat prototipe terlalu cepat, tanpa analisis kebutuhan *user* yang cukup mendalam, menghasilkan rancangan yang tidak fleksibel dengan focus sempit yang membatasi potensi sistem di masa depan.
6. *Designer* mungkin mengabaikan dokumentasi, menghasilkan justifikasi yang tidak cukup untuk produk akhir dan catatan yang tidak memadai di masa depan.

2.4.2. RAD Model (*Rapid Application Development Model*)

Rapid Application Development merupakan model proses pengembangan *software incremental* yang menekankan pada siklus pengembangan yang sangat singkat. RAD model adalah adaptasi "kecepatan tinggi" dari *linear sequential model* dimana pembangunan yang cepat dicapai dengan menggunakan konstruksi berbasis komponen. Jika kebutuhan sistem dipahami dengan baik dan ruang lingkup projek dibatasi, proses RAD memungkinkan tim pengembangan membuat

“sistem yang berfungsi secara penuh” dalam waktu yang sangat singkat. Terutama digunakan untuk aplikasi-aplikasi sistem informasi, pendekatan RAD meliputi pase-pase berikut ini:

Business modeling. Arus informasi diantara fungsi bisnis dimodelkan dengan cara menjawab pertanyaan-pertanyaan berikut ini: Informasi apa yang mengendalikan proses bisnis? Informasi apa yang dihasilkan? Siapa yang menghasilkan informasi itu? Kemana informasi itu mengalir? Siapa yang memproses informasi itu?

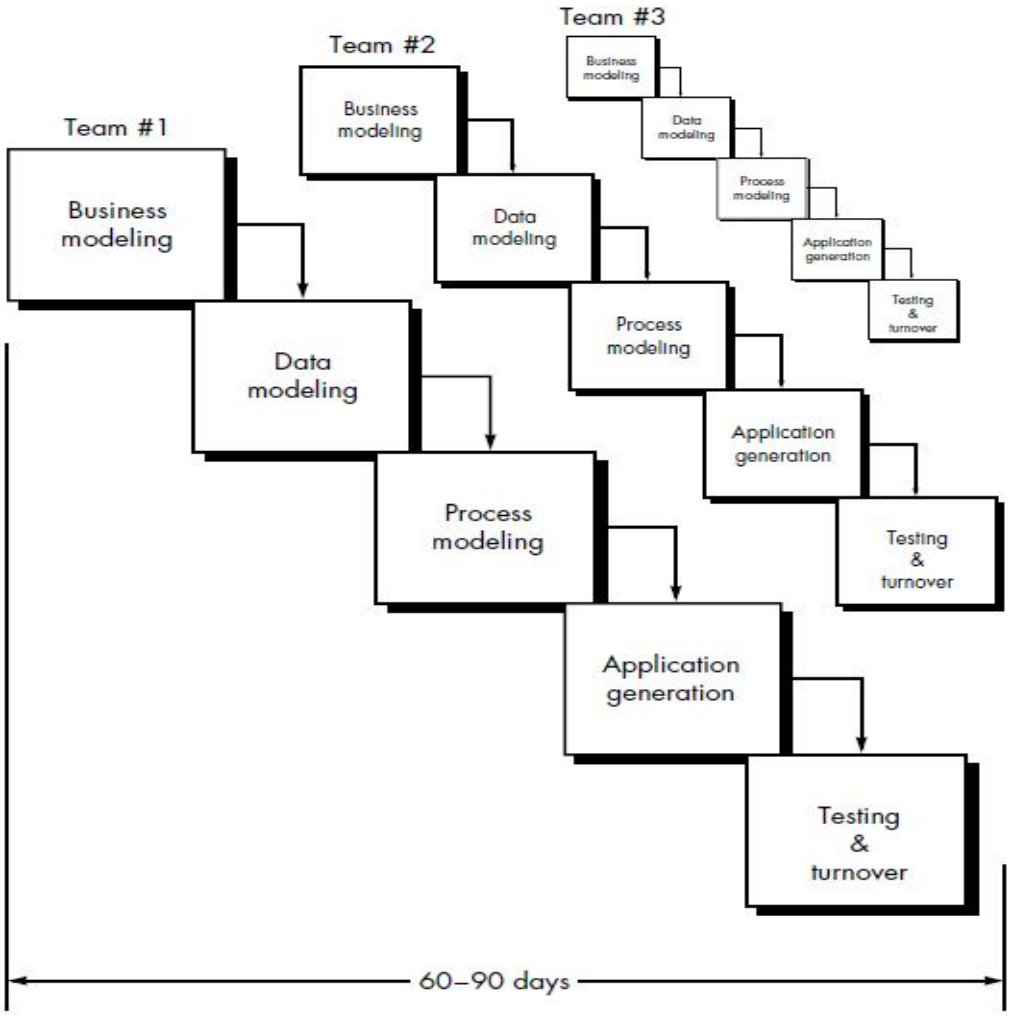
Data modeling. Arus informasi yang didefinisikan sebagai bagian dari fase model bisnis yang diperhalus ke dalam sekumpulan objek data yang diperlukan untuk mendukung bisnis. Karakteristik (disebut juga atribut) dari setiap objek diidentifikasikan dan didefinisikan hubungan antar objeknya.

Process modeling. Objek-objek data didefinisikan di dalam fase model data yang ditransformasi untuk memperoleh arus informasi yang dibutuhkan untuk mengimplementasikan sebuah fungsi bisnis. Deskripsi proses dibuat untuk menambah, memodifikasi, menghapus, atau mengembalikan nilai suatu objek data.

Application generation. RAD mengasumsikan pemakaian teknik generasi keempat. Daripada membuat *software* dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, proses RAD bekerja dengan menggunakan ulang komponen program yang ada (ketika memungkinkan) atau membuat komponen-komponen yang dapat

digunakan kembali (ketika dibutuhkan). Di seluruh kasus, alat terotomatisasi digunakan untuk memfasilitasi pembangunan *software*.

Testing and turnover. Sejak proses RAD menekankan penggunaan ulang, banyak komponen program yang siap diuji. Ini mengurangi waktu pengujian secara keseluruhan. Bagaimanapun, komponen-komponen baru dan semua *interface* pasti diuji dan dilakukan sepenuhnya.



Gambar 2.3 RAD Model
Sumber: Software Engineering a Practitioner's Approach (Roger S.Pressman) Hlm.33

Model proses RAD dapat dilihat pada Gambar 2.3. Jelas, keterbatasan waktu dikenakan pada sebuah projek RAD pada permintaan "ruang lingkup terukur". Jika aplikasi bisnis bisa dimodularisasi dengan cara mengaktifkan fungsi utama, maka akan selesai dalam waktu kurang dari tiga bulan. Fungsi utama ini akan menjadi kandidat bagi RAD. Tiap-tiap fungsi utama bisa dialamatkan oleh sebuah tim RAD terpisah dan kemudian mengintegrasikannya.

Kelebihan dari Model RAD ini diantaranya sebagai berikut:

1. Versi operasional dari aplikasi bisa didapatkan lebih cepat daripada dengan *Waterfall, Incremental,* atau *Spiral framework.*
2. Karena RAD menghasilkan sistem yang langsung mengarah ke fokus bisnis, pendekatan ini cenderung menghasilkan sistem dengan biaya lebih rendah.
3. Menimbulkan komitmen pada tingkat yg lebih besar dari *stakeholder,* baik dalam bisnis maupun teknis, daripada *Waterfall, Incremental,* atau *Spiral framework.* Pengguna dilihat sebagai keuntungan lebih dari rasa kepemilikan dari suatu sistem, sedangkan pengembang sistem dipandang mendapatkan keuntungan lebih banyak dari menghasilkan sistem yang sukses dengan cepat.
4. Berkonsentrasi pada elemen sistem yang penting dari sudut pandang pengguna.
5. Menyediakan kemampuan dengan cepat mengubah rancangan sistem sesuai yang diminta oleh pengguna.

6. Menghasilkan hubungan yang erat antara kebutuhan pengguna dengan spesifikasi sistem.
7. Umumnya secara dramatis menghasilkan sistem yang menghemat waktu, uang, dan tenaga manusia.

Sedangkan kekurangan dari Model RAD diantaranya sebagai berikut:

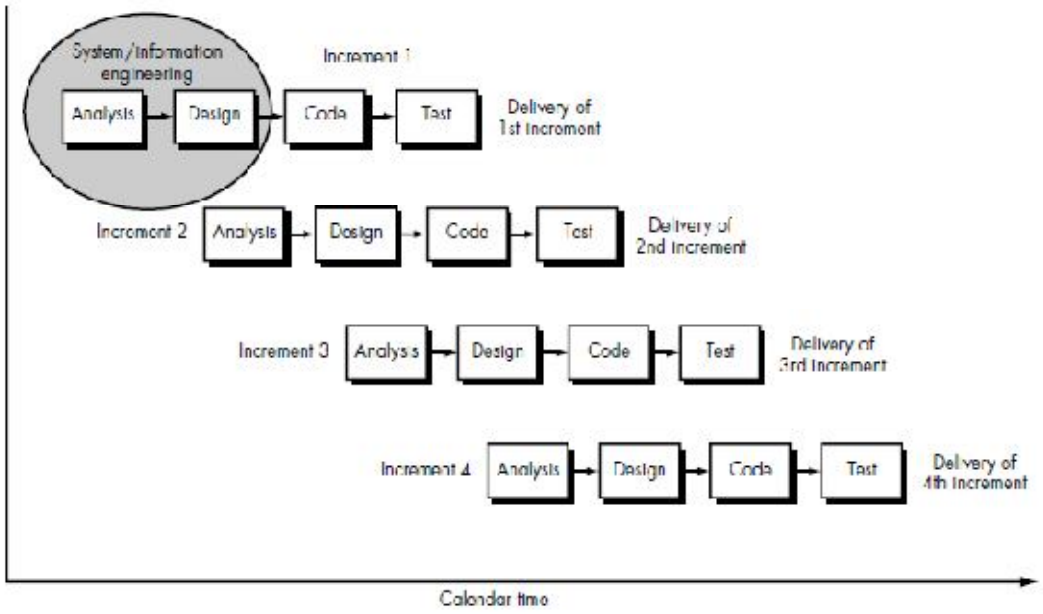
1. Kecepatan waktu dan biaya yang lebih rendah mungkin mengarah pada kualitas sistem yang rendah pula secara keseluruhan.
2. Bahaya dari tidak searahnya sistem yang dikembangkan dengan bisnis karena hilangnya informasi.
3. Projek mungkin berakhir dengan kebutuhan sistem lebih daripada yang dibutuhkan.
4. Berpotensi untuk terobosan fitur dimana semakin banyak fitur yang ditambahkan ke sistem selama masa pengembangan.
5. Berpotensi terhadap rancangan yang tidak konsisten di dalam dan diseluruh sistem.
6. Berpotensi terhadap pelanggaran standar pemrograman yang berhubungan dengan ketentuan penamaan yang tidak konsisten dan dokumentasi yang tidak konsisten.
7. Kesulitan dengan penggunaan ulang modul untuk sistem yang akan datang.

2.4.3. *Incremental Model*

Incremental model mengombinasikan elemen-elemen pada *linear sequential model* (diaplikasikan secara berulang-ulang) dengan filosofi iterasi pada prototipe. Mengacu pada Gambar 2.4, *incremental model* mengaplikasikan *linear sequence* dalam bentuk yang mengejutkan sebagai perkembangan berdasarkan waktu kalendar. Tiap-tiap *linear sequence* menghasilkan sebuah *software* "tambahan". Sebagai contoh, *software* pengolah kata (*word-processing*) dikembangkan menggunakan paradigma *incremental* yang mampu menyajikan manajemen dasar *file*, proses *edit*, dan fungsi produksi dokumen ada di *increment* pertama; kemampuan produksi dokumen dan proses *edit* yang lebih canggih ada di *increment* kedua; kemampuan tingkat tinggi *page layout* ada di *increment* keempat. Ini seharusnya dijadikan catatan bahwa alur proses pada setiap *increment* bisa digabungkan dengan pradigma prototipe.

Ketika *incremental model* digunakan, *increment* pertama sering disebut produk inti. Artinya, kebutuhan-kebutuhan mendasar telah dialamatkan, tetapi masih banyak fitur tambahan (sebagian diketahui, dan yang lainnya tidak diketahui) yang belum terkirim. Produk inti digunakan oleh konsumen (atau mengalami tinjauan rinci). Sebagai hasil penggunaan dan/atau evaluasi, sebuah rencana dikembangkan *increment* berikutnya. Rencana tersebut mengalamatkan modifikasi dari produk inti supaya lebih baik dengan mempertemukan kebutuhan *user* dan penyampaian (*delivery*)

fungsionalitas dan fitur tambahan. Proses ini terus diulang mengikuti penyampaian tiap-tiap *increment*, sampai produk lengkap diproduksi.



Gambar 2.4 ***Incremental Model***
Sumber: Software Engineering a Practitioner's Approach (Roger S.Pressman) Hlm.35

Incremental model, seperti prototipe dan pendekatan evolusioner lainnya, pada dasarnya adalah iterasi/perulangan. *Increment* awal adalah versi pelucutan dari produk akhir. Tetapi mereka memiliki kemampuan menyediakan layanan kepada *user* dan juga menyediakan sebuah *platform* untuk dievaluasi oleh *user*.

Pengembangan *incremental* khususnya sangat berguna ketika staf tidak tersedia untuk melengkapi implementasi melalui batas waktu bisnis yang telah dibangun selama projek. *Increment* awal bisa diimplementasikan hanya dengan sedikit orang. Jika produk inti telah diterima dengan baik, kemudian staf tambahan (jika diperlukan) bisa diikutsertakan untuk implementasi *increment* berikutnya. Sebagai

tambahan, *increment* bisa direncanakan untuk me-*manage* risiko teknis. Sebagai contoh, sistem utama mungkin membutuhkan ketersediaan *hardware* baru yang sedang dikembangkan dan yang tidak pasti tanggal pengirimannya. Ini memungkinkan untuk merencanakan *increment* awal dengan cara mencegah penggunaan *hardware* ini, sehingga memungkinkan fungsi parsial yang akan disampaikan kepada pengguna akhir tanpa penundaan yang berlebihan.

Kelebihan dari *Incremental Model* diantaranya sebagai berikut:

1. Potensi yang ada untuk mengekploitasi pengetahuan yang didapat di *increment* awal sebagai bahan pengembangan *increment* berikutnya.
2. Kontrol moderat dipertahankan selama hidup projek melalui penggunaan dokumentasi tertulis dan tinjauan formal serta persetujuan oleh pengguna dan manajemen teknologi informasi pada tonggak utama yang ditunjuk.
3. *Stakeholder* bisa diberikan bukti nyata dari status projek selama siklus hidup projek itu.
4. Membantu mengurangi risiko integrasi dan arsitektur sebelumnya dalam projek.
5. Memungkinkan pengiriman dari serangkaian implementasi yang lebih lengkap secara bertahap dan bisa masuk ke produksi yang lebih cepat sebagai rilis *incremental*.

6. Implementasi secara bertahap menyediakan kemampuan untuk memonitor efek dari perubahan bertahap, mengisolasi masalah dan melakukan penyesuaian sebelum organisasi mengalami dampak negatif.

Sedangkan kekurangan dari *Incremental Model* ini diantaranya sebagai berikut:

1. Ketika memanfaatkan seri dari mini *Waterfall* untuk bagian kecil dari sistem sebelum memindahkannya ke *incement* berikutnya, biasanya ada kurangnya pertimbangan keseluruhan dari masalah bisnis dan persyaratan teknis untuk sistem secara keseluruhan.
2. Karena beberapa modul akan diselesaikan jauh lebih awal daripada yang lain, diperlukan *interface* yang terdefinisi dengan baik.
3. Masalah yang sulit cenderung ditekankan di masa yang akan datang untuk menunjukan keberhasilan awal kepada pihak manajemen.

2.5. Metode Pendekatan Sistem Informasi

Pada pembangunan sebuah sistem informasi, ketentuan penggunaan metode pendekatan bergantung pada kasus yang terjadi. Pada umumnya ada dua metode pendekatan sistem informasi, yaitu *object-oriented approach* dan *structured analysis approach.* Apakah benar jauh berbeda analisis berbasis pendekatan objek dengan analisis terstruktur? Dalam hal ini, **Fichman** dan **Kemerer** berpendapat:

"Kami menyimpulkan bahwa pendekatan analisis berorientasi objek merupakan perubahan radikal atas metodologi yang berorientasi proses seperti analisis terstruktur, tetapi hanya perubahan *incremental* atas metodologi berorientasi data seperti teknik informasi. Metodologi

berorientasi proses fokus memperhatikan sifat yang melekat pada objek selama pemodelan proses dan mengarah pada model masalah domain yang ortogonal terhadap tiga prinsip penting dari orientasi objek: enkapsulasi, klasifikasi objek, dan turunan (*inheritance*)."

Secara sederhana, analisis terstruktur mengambil pandangan *input-process-output* yang berbeda kebutuhan. Data dianggap terpisah dari proses yang mengubah data tersebut. Tingkah laku sistem, walaupun penting, cenderung memainkan peran sekunder dalam analisis terstruktur. Pendekatan analisis terstruktur membuat berat penggunaan pada dekomposisi fungsional.

2.5.1. *Unified Modeling Language (UML)*

UML mungkin bisa disebut pengganti dari *Object Oriented Analysis and Design (OOAD)* yang tersebar selama perang metode antara tahun 80-an sampai awal 90-an. UML hanya bahasa komponen dari sebuah metode dan dilengkapi secara terpisah oleh *Rational Unified Process (RUP).* UML merupakan gabungan dari tiga metode utama bahasa pemodelan di dalam industri: *Booch*, *Rumbaugh*, dan *Jacobson*. UML melalui proses standarisasi dengan *Object Management Group* (OMG), dan sekarang menjadi standar OMG.

Aturan-aturan pada UML bisa diklasifikasikan ke dalam:

1. *Syntactic :* rincian aspek dan kombinasi aturan;
2. *Semantic :* rincian arti dari simbol, secara individual dan dalam konteks;
3. *Pragmatic :* panduan bagaimana menggunakan bahasanya (maksud dari simbol).

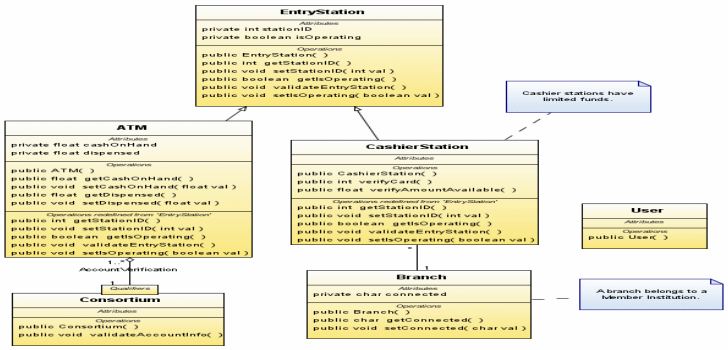
Konsep yang paling penting dalam pemahaman UML adalah: arsitektur UML, notasi (diagram), mekanisme ekstensi dan pembatas.

2.5.1.1. Diagram UML

Diagram UML berisi Sembilan jenis diagram, yang tujuh diantaranya digunakan dalam pemodelan bisnis. Contoh-contoh diagram yang disediakan maksudnya untuk mengungkapkan pemodelan bisnis atau sistem informasi pendukung bisnis.

1. *Class Diagram*

   Class diagram terdiri dari kelas dan hubungan antar kelasnya. *Class diagram* menggambarkan struktur dari suatu sistem. Kelas-kelas dapat mewakili informasi, produk, dokumen, atau organisasi. Kelas dalam *class diagram* dihubungkan melalui asosiasi, bisa dengan agregasi, komposisi, generalisasi dan dependensi.

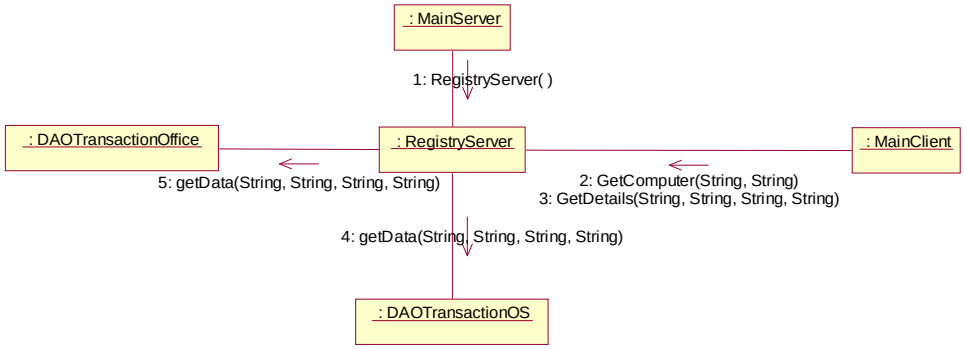


Gambar 2.5 Contoh *Class Diagram*
Sumber: Dokumentasi Pribadi

Package digunakan untuk mengatur dan menangani kompleksitas model untuk ukuran besar. Sebuah *package* berisi kumpulan kelas (*a group of classes*).

2. *Object Diagram*

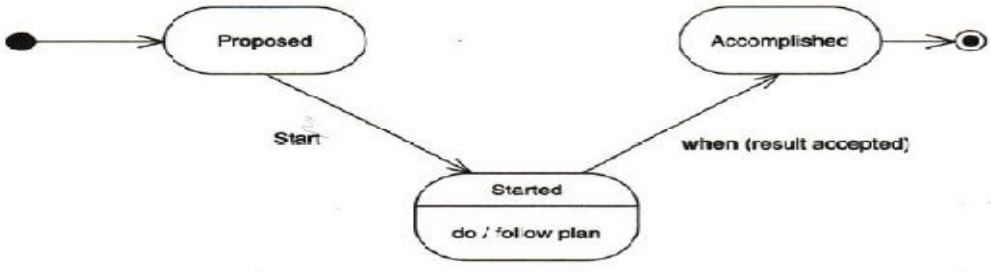
Object diagram merupakan instansiasi dari sebuah *class diagram* untuk situasi yang spesifik.



Gambar 2.6 Contoh *Object Diagram*
Sumber: Dokumentasi Pribadi

3. *Statechart Diagram*

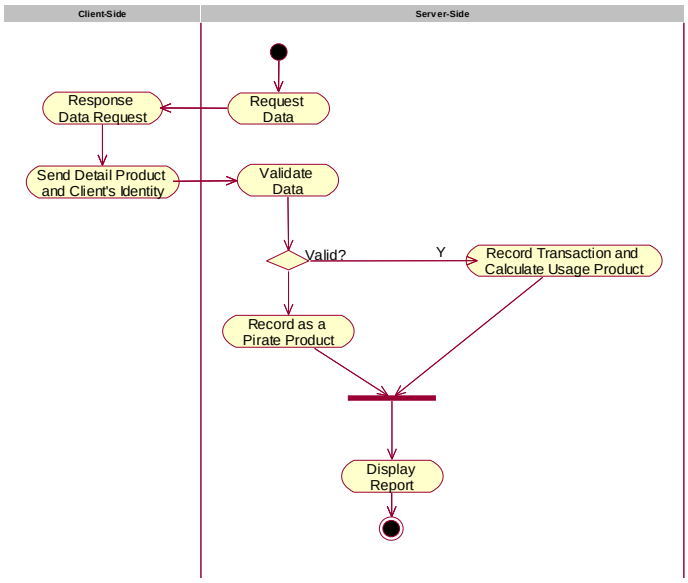
Statechart diagram menunjukan kemungkinan pernyataan yang ada pada sebuah sistem. *Statechart diagram* menangkap siklus hidup objek dan sistem, menunjukan pernyataannya bisa masuk ke dalam *event* dan bagaimana *event* yang berbeda bisa mempengaruhi pernyataan itu dari waktu ke waktu.



Gambar 2.7 Contoh *Statechart Diagram*
Sumber: Business Modeling UML vs. IDEF.Ovidiu S Noran.Griffith University(Hlm.10)

4. *Activity Diagram*

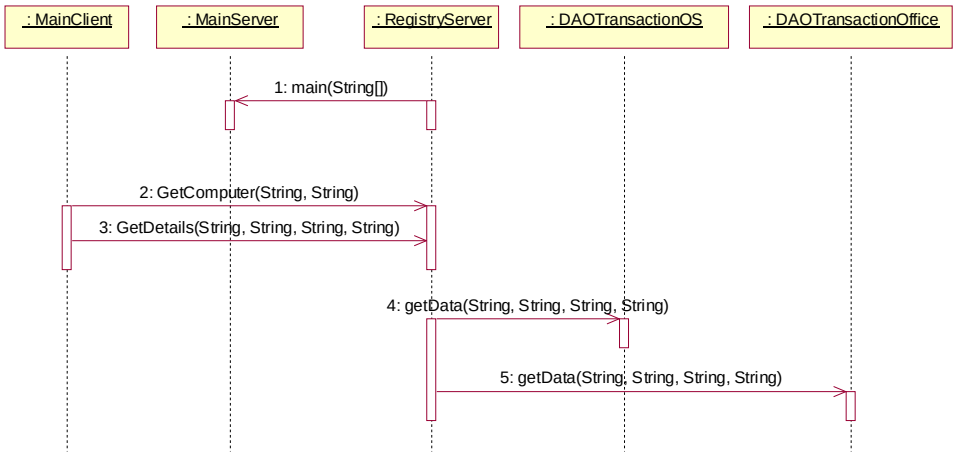
Activity diagram mendeskripsikan aktivitas-aktivitas dan aksi yang terjadi selama dalam sistem. Dalam pemodelan bisnis, *activity diagram* biasanya digunakan untuk memodelkan proses bisnis.



Gambar 2.8 Contoh *Activity Diagram*
Sumber: Dokumentasi Pribadi

5. *Sequence Diagram*

Sequence diagram mendeskripsikan urutan pesan yang muncul diantara sekumpulan objek. Urutan dan waktu objek dengan jelas digambarkan. Pesan dalam *sequence diagram* mungkin memiliki parameter. Pesan bisa saja sinkronis (wajib diisi sebelum yang lainnya diselesaikan) atau bisa juga tidak sinkronis.



Gambar 2.9 Contoh *Sequence Diagram*
Sumber: Dokumentasi Pribadi

6. *Collaboration Diagram*

   Collaboration diagram mirip dengan *sequence diagram*. Bagaimanapun, *collaboration diagram* memiliki kemampuan untuk mengungkapkan hubungan dan interaksi yang lebih kompleks diantara objek yang berkolaborasi.

7. *Use Case Diagram*

   Mendeskripsikan bagian-bagian dari fungsionalitas sistem. Sebuah *use case* adalah salah satu penggunaan sistem oleh seorang aktor (aturan yang dimiliki *user* di dalam sistem). *Use case diagram* menggambarkan hubungan antara *use case*.

8. *Component Diagram*

   Component diagram terutama digunakan untuk struktur komponen di dalam sistem perangkat lunak. *Component diagram* tidak digunakan di dalam pemodelan bisnis.

9. *Deployment Diagram*

   Adalah *class diagram* yang menggambarkan *hardware* di dalam sistem perangkat keras.