

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Menurut Jogiyanto [2, p683] "Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu". Sebuah sistem dapat terdiri dari sistem-sistem bagian (*subsistem*) masing-masing dapat terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen yang saling berinteraksi dan saling berhubungan membentuk suatu kesatuan sehingga tujuan atau sasaran sistem dapat tercapai. Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu:

2.1.1. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Sedangkan suatu sistem yang mempunyai sistem yang lebih besar disebut supra sistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. [2, p684]

2.1.2. Batasan sistem

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

2.1.3. Lingkungan luar sistem

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

2.1.4. Penghubung sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung. Dengan penghubung satu sub sistem dapat berinteraksi dengan subsistem yang lainnya membentuk satu kesatuan. [2, p684]

2.1.5. Masukan sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

2.1.6. Keluaran sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat berupa masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna dan merupakan sisa hasil pembuangan, sedang informasi adalah keluaran yang dibutuhkan

2.1.7. Pengolahan sistem

Suatu sistem dapat mempunyai suatu bagian 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. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

2.1.8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem akan sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.[2, p688]

2.1.9. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu : tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem :

2.1.9.1. Tujuan

Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.

2.1.9.2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang

berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

2.1.9.3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lebih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

2.1.9.4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

2.1.9.5. Batas

Yang disebut batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi

sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbatasan dana.

2.1.9.6. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

2.1.9.7. Lingkungan

Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, adalah sebagai berikut :

Sistem abstrak (*abstract system*) dan sistem (sistem fisik). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik adalah sistem yang secara fisik.

Sistem alamiah (*natural system*) dan sistem buatan manusia (sistem yang dibuat manusia). Sistem sistem adalah alam yang terjadi melalui [3, p.4] Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

2.2.1. Abstrak (*abstract system*) dan Fisik (*physical system*)

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teknologi, yaitu system yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya system komputer, sistem akuntansi, sistem produksi dan sebagainya. Alamaiah (*natural system*) dan Buatan manusia (*human made system*) Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi merupakan contoh *man-machine system*, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

2.2.2. Tertentu (*deterministic system*) dan Tidak tentu (*probabilistic system*)

Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. [3, p4]

2.2.3. Tertutup (*closed system*) dan Terbuka (*open system*)

Sistem yang tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem ini ada, tetapi kenyataannya tidak ada system yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relative tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja. [3, p4]

2.3. Pengertian Informasi

Informasi adalah Sekumpulan data/ fakta yang diorganisasi atau diolah dengan cara tertentu sehingga mempunyai arti bagi penerima data yang telah diolah menjadi sesuatu yang berguna bagi si penerima maksudnya yaitu dapat memberikan keterangan atau pengetahuan. Dengan demikian yang menjadi sumber informasi adalah data. Informasi dapat juga di katakan sebuah pengetahuan yang diperoleh dari pembelajaran, pengalaman, atau instruksi.[2, p129]

2.4. Pengertian Sistem Informasi

Istilah sistem informasi menyiratkan suatu pengumpulan data yang terorganisasi beserta tatacara penggunaannya yang mencakup lebih jauh dari sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaannya. Keberhasilan suatu sistem informasi yang diukur berdasarkan maksud pembuatannya tentu bergantung pada tiga pokok utama, yaitu :

1. keserasian dan mutu data
2. pengorganisasian data
3. tatacara penggunaannya

Menurut Cook [4]. Setiap sistem informasi menyajikan tiga gatra pokok : (1) pengumpulan dan pemasukan data, (2) penyimpanan dan pengambilan kembali

(retrieval) data, dan (3) penerapan data, yang dalam hal sistem informasi terkomputer penayangan (display)

Menurut Knapp [5]. Suatu sistem informasi terkomputer pada dasarnya terdiri atas lima komponen yang menjadi subsistemnya, yaitu :

1. pelambangan (encoding) data dan pemrosesan masukan.
2. pengolahan data.
3. pengambilan kembali data.
4. pengolahan dan analisis data
5. penayangan data.

Suatu sistem informasi dibuat untuk suatu keperluan tertentu atau untuk memenuhi permintaan penggunaan tertentu, maka struktur dan cara kerja sistem informasi berbedabeda bergantung pada macam keperluan atau macam permintaan yang harus dipenuhi. Oleh karena kepentingan yang dilayani sangat beraneka, maka macam sistem informasi pun sangat beraneka. Namun demikian, sistem informasi mempunyai banyak tampilan (*features*) umum dan menghadapi banyak persoalan yang mirip. Jadi, disamping perbedaan yang jelas terdapat banyak persamaan antar berbagai sistem informasi. Suatu persamaan yang menonjol ialah semua sistem informasi menggabungkan berbagai ragam data yang dikumpulkan dari berbagai sumber. Untuk dapat menggabungkan data yang berasal dari berbagai

sumber diperlukan suatu sistem alih rupa (transformation) data, sehingga menjadi tergabungkan (*compatible*). Berapa pun ukurannya dan apa pun ruang lingkungannya, suatu sistem informasi perlu memiliki ketergabungan (*compatibility*) data yang disimpannya. Ketergantungan dakhil ialah yang merupakan tampakan umum suatu pokok suatu system informasi.

2.5. Metode Pengembangan dan Perancangan Sistem

Metode Pengembangan adalah sebuah cara yang tersistem atau teratur yang bertujuan untuk melakukan analisa pengembangan suatu sistem agar sistem tersebut dapat memenuhi kebutuhan. [1, p34]

Jadi dalam melakukan pembangunan atau perbaikan suatu sistem yang terkomputerisasi harus melakukan langkah-langkah dalam mengimplementasikannya. Metode Pengembangan Rekayasa Perangkat Lunak terdapat 4 metode yaitu : *waterfall*, *Prototyping*, dan RAD. Masing-masing mempunyai kelebihan dan kelemahan sendiri.

Metodologi ini melakukan beberapa penyesuaian terhadap SDLC pada beberapa bagian sehingga lebih cepat untuk sampai ke tangan pengguna. metodologi ini biasanya mensyaratkan beberapa teknik dan alat2 khusus agar proses bisa cepat
METODE RAPID APPLICATION DEVELOPMENT (RAD)

2.5.1. Pengertian *Rapid Application Development* :

Rapid application development (RAD) atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* (bertingkat). RAD menekankan pada siklus pembangunan pendek, singkat, dan cepat. Waktu yang singkat adalah batasan yang penting untuk model ini. *Rapid application development* menggunakan metode *iteratif* (berulang) dalam mengembangkan sistem dimana *working model* (modeln kerja) sistem dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan (*requirement*) pengguna. Model kerja digunakan hanya sesekali saja sebagai basis desain dan implementasi sistem akhir. *Rapid Application Development* (RAD) adalah metodologi pengembangan perangkat lunak yang berfokus pada membangun aplikasi dalam waktu yang sangat singkat. Istilah ini nmenjadi kata kunci pemasaran yang umum menjelaskan aplikasi yang dapat dirancang dan dikembangkan dalam waktu 6090 hari, tapi itu awalnya ditujukan untuk menggambarkan suatu proses pembangunan yang melibatkan *application prototyping* dan *iterativen development*. [1, p34]

Unsur-Unsur *Rapid Application Development* RAD memiliki banyak unsur-unsur yang membuat sebuah metodologi yang unik termasuk *prototyping*, *iterative development*, *time boxing*, *team members*, *management approach*, dan *RAD tools*. Sebuah aspek kunci dari RAD adalah pembangunan *prototype* untuk tujuan membangkitkan kembali desain untuk kebutuhan pengguna. Tujuannya adalah untuk membangun sebuah fitur ringan yang hasil akhirnya dalam jumlah pendek dengan

waktu yang memungkinkan. Prototipe awal berfungsi sebagai bukti konsep untuk klien, tetapi lebih penting berfungsi sebagai titik berbicara dan alat untuk kebutuhan pemurnian. Mengembangkan prototipe cepat dicapai dengan *Computer Aided Engineering CASE tools Software* yang berfokus pada menangkap persyaratan, mengkonversi mereka ke model data, mengubah model data ke database, dan menghasilkan kode semua dalam satu alat. *CASE tools* populer di 80an dan awal 90an, tetapi sebagai teknologi telah berubah dan (COBOL telah menjadi usang) beberapa alat mengambil keuntungan penuh dari potensi penuh dari teknologi KASUS alat. Perusahaan rasional adalah yang paling terkenal meskipun prototipe potensi pembangkitnya terbatas. Pada Otomatis Arsitektur produk cetak biru kami berfokus pada peningkatan tingkat aplikasi *enterprise web* yang berfungsi sebagai prototipe karena kecepatan yang mereka dapat diciptakan dalam hitungan menit.

2.5.2. Model Sekuensial Linier atau Waterfall Development Model

Model *Sekuensial Linier* atau sering disebut Model Pengembangan Air Terjun, merupakan *paradigma* model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan.[1, p28]

2.5.3. Model Prototype

Metode Prototype merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau *waterfall development model*. Dalam *Model Prototype*, *prototype* dari perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.[1, p31]

2.5.4. Model Evolutionary Development / Evolutionary Software Process Models

Model Evolutionary Development bersifat iteratif (mengandung perulangan). Hasil prosesnya berupa produk yang makin lama makin lengkap sampai versi terlengkap dihasilkan sebagai produk akhir dari proses. *Model Evolutionary Development / Evolutionary Software Process* terbagi menjadi 2, yaitu :

2.5.5. Model Incremental

Model Incremental merupakan hasil kombinasi elemen-elemen dari model *waterfall* yang diaplikasikan secara berulang, atau bisa disebut gabungan dari Model linear sekuensial (*waterfall*) dengan *Model Prototype*. Elemen-elemen tersebut

dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.

2.5.6. Model Spiral / Model Boehm

Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model *prototyping* dengan pengulangannya dan model *waterfall* dengan pengendalian dan sistematikanya. Model ini dikenal dengan sebutan Spiral Boehm. Pengembang dalam model ini memadupadankan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.[1, p39]

2.6. Perangkat Lunak Pendukung

2.6.1. HTML

Hyper Text Markup Language (HTML) HTML (*HyperText Markup Language*) dikenal sebagai bahasa kode berbasis teks untuk membuat sebuah halaman web, keberadaannya dikenal dengan adanya ekstensi *.htm atau *.html [9]. HTML merupakan suatu bahasa dari website (www) yang dipergunakan untuk menyusun dan membentuk dokumen agar dapat ditampilkan pada program browser. Ketika user mengakses web, maka ia mengakses dokumen seseorang yang ditulis dengan gunakan format HTML. Dapat disimpulkan bahwa HTML merupakan

protokol yang digunakan untuk transfer data atau dokumen dari web server ke browser. [8]

2.6.2. PHP

PHP merupakan singkatan dari "PHP: *Hypertext Preprocessor*" adalah skrip yang dijalankan di *server*. Hasilnyalah yang dikirimkan ke klien, tempat pemakai menggunakan *browser*. Keuntungan PHP, kode yang menyusun program tidak perlu diedarkan ke pemakai sehingga kerahasiaan kode dapat dilindungi.[9, P2]

2.6.3. MySQL

MySQL merupakan *software* yang berbasis *structure query language* (SQL) tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *MySQL* adalah sebuah implementasi dari sistem manajemen *database* relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan *MySQL*, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial [10].

2.6.4. CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.

2.6.5. Dreamweaver

Adobe Dreamweaver Adobe Dreamweaver adalah program yang digunakan untuk membuat atau menyunting halaman web. Software Dreamweaver dikeluarkan oleh Adobe System.[11]

2.6.6. Use Case

Use case diagram ialah model fungsional sebuah system yang menggunakan actor dan use case. Use case adalah layanan (services) atau fungsi-fungsi yang disediakan oleh system untuk penggunaanya (Henderi et al, 2008).

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah system. Yang tekankan adalah “apa” yang dibuat system, dan bukan “bagaimana” sebuah use case menerangkan sebuah interaksi antar actor dengan system. Use case merupakan sebuah pekerjaan tertentu, misal :login ke system, meng-create sebuah daftar belanja, dsb. Seorang sebuah actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu.

Use Case Diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah system, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada system. Sebuah 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 fungsional yang common.

Sebuah use case juga dapat meng-extended use case lain dengan behavior-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.[1, p176]

2.6.7. Activity Diagram

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.[1, p233]

2.6.8. Sequence Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.[1, p208]

2.6.9. Class Diagram

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.[1, p205]

Class juga memiliki 3 area pokok (utama) yaitu : nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. Dalam mendefinisikan metode yang ada di dalam class harus diperhatikan yang namanya Cohesion dan Coupling, Cohesion adalah ukuran keterkaitan sebuah instruksi di sebuah metode, Coupling adalah ukuran keterkaitan antar metode. Di dalam class diagram terdapat hubungan antar class secara konseptual, yang disebut Relasi antar Class, di UML disediakan macam-macam relasi antar Class, diantaranya: Asosiasi (Hubungan statis antar kelas), Agregasi (hubungan dari keseluruhan objek), Generalisasi (relasi beberapa subkelas ke super kelas), Dependency (keterhubungan tiap kelas).