



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Komputer

Menurut Puspitosari (2013:1), “Komputer adalah sebuah alat yang digunakan untuk mengolah data menurut perintah yang telah dirumuskan. Komputer dapat didefinisikan sebagai sekumpulan alat elektronik yang saling terkoordinasi satu sama lain sehingga dapat menerima data, kemudian mengolah data, dan pada akhirnya akan menghasilkan suatu keluaran yang berupa informasi (Input > Proses > Ouput)”. Sedangkan menurut Kadir (2013:2), “Komputer merupakan peralatan elektronik yang biasa dipakai orang untuk membantu pelaksanaan pekerjaan”.

Berdasarkan pengertian diatas dapat disimpulkan bahwa pengertian komputer adalah perangkat keras yang digunakan untuk mengolah data, informasi dan perangkat lunak, yang membantu dalam melakukan pekerjaan.

2.1.2 Data

John J. Longkutoy dalam bukunya *Pengenalan Komputer* (Sutabri, 2012:2). “Istilah data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau simbol-simbol yang menunjukkan suatu ide, objek, kondisi atau situasi dan lain-lain. Jelasnya, data itu berupa apa saja dan dapat ditemui di mana saja. Kegunaan data adalah sebagai bahan dasar yang objektif (relatif) di dalam proses kebijaksanaan dan keputusan oleh pimpinan organisasi”.

2.1.3 Website

Menurut Abdulloh (2018:1), “Website atau disingkat web dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet”. Sedangkan menurut



Sidik (2017:1), “Situs Web (*web site*) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep *hyperlink*, yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penelusuran informasi di Internet) untuk mendapatkan informasi, dengan cukup mengklik suatu link berupa teks atau gambar, maka informasi dari teks atau gambar akan ditampilkan secara lebih rinci (*detail*)”.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian website adalah sekumpulan halaman yang terdiri dari beberapa informasi berisi tentang video, gambar, teks, animasi yang diakses dengan menggunakan internet maupun tidak tergantung dengan laman web yang diakses.

2.2 Teori Judul

2.2.1 Sistem

Sistem dapat didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama (Pratama, 2014:7).

Sistem adalah kumpulan dari obyek-obyek seperti orang, *resources*, konsep dan prosedur yang ditujukan untuk melakukan fungsi tertentu atau memenuhi suatu tujuan. Kemudian sistem juga merupakan kumpulan dari komponen yang berinteraksi bersama-sama secara kolektif untuk melaksanakan tugas (Pratiwi, 2016:4).

Berdasarkan uraian tersebut maka dapat disimpulkan bahwa pengertian sistem adalah sekumpulan komponen atau elemen yang dapat diartikan sebagai suatu kumpulan atau obyek-obyek yang terhubung satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan.

2.2.2 Keputusan

Keputusan merupakan hasil pemikiran berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapi (Pratiwi, 2016:2).



Jenis-jenis keputusan dibedakan menjadi tiga macam yaitu keputusan terstruktur, keputusan tidak terstruktur, dan keputusan semi terstruktur (Pratiwi, 2016:5-6).

1. Keputusan terstruktur

Keputusan-keputusan yang berkaitan dengan persoalan yang telah diketahui sebelumnya. Proses pengambilan keputusan seperti ini biasanya didasarkan atas teknik-teknik tertentu dan sudah dibuat standarnya. Kategori keputusan ini juga dapat dikatakan suatu proses jawaban secara otomatis pada kebijakan yang sudah ditentukan sebelumnya. Secara alamiah hampir semua masalah rutin dan berulang memiliki parameter-parameter persoalan yang telah diketahui dan terdefinisi dengan baik, sehingga jawaban atau proses pengambilan keputusan pun bersifat rutin dan terjadwal.

2. Keputusan tak terstruktur

Keputusan-keputusan yang berkaitan dengan berbagai persoalan baru. Keputusan tidak terstruktur biasanya juga berkaitan dengan persoalan yang cukup pelik, karena banyak parameter yang tidak diketahui atau belum diketahui. Oleh karena itu, untuk mengambil keputusan ini biasanya intuisi serta pengalaman seorang pelaku organisasi akan sangat membantu.

Keputusan tak terstruktur, adalah “*fuzzy*”, permasalahan kompleks dimana tak ada solusi yang mengikutinya. Masalah yang tak terstruktur adalah tak adanya 3 fase proses yang terstruktur. Keputusan tidak terstruktur (unstructured decision) bukan merupakan keputusan yang berulang dan rutin. Contohnya adalah memilih sampul depan sebuah majalah, mengontrak manajemen tingkat senior, dan memilih proyek penelitian awal yang akan dilakukan. Tidak ada kerangka atau model yang dapat memecahkan masalah sejenis ini. Bahkan, dibutuhkan banyak sekali pertimbangan dan intuisi. Walaupun demikian, keputusan tidak terstruktur dapat didukung oleh bantuan dari keputusan yang diambil berdasar hasil komputer, yang berfungsi untuk memfasilitasi pengumpulan informasi dari berbagai sumber. Contohnya adalah keputusan untuk pengembangan teknologi baru, pengembangan jenis



usaha baru, keputusan untuk bergabung dengan perusahaan lain, perekrutan eksekutif.

3. Keputusan semi terstruktur

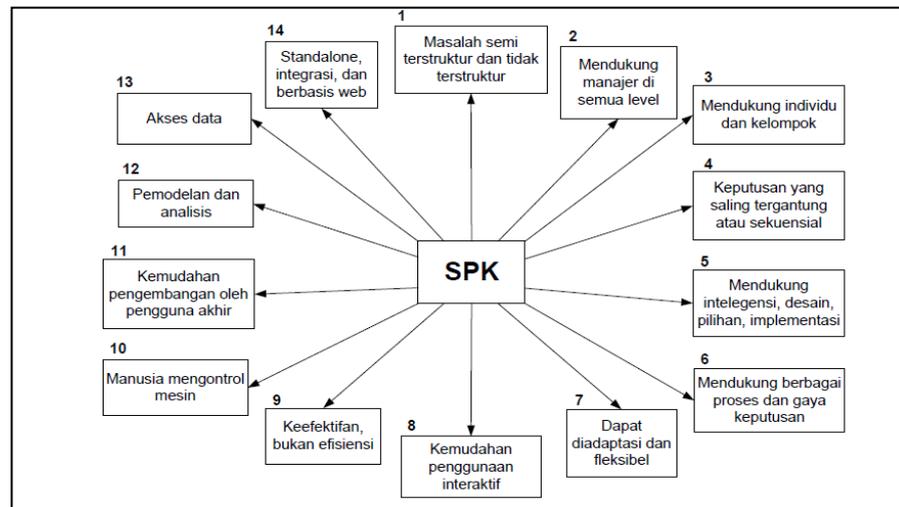
Terdapat beberapa keputusan terstruktur, tetapi tak semua dari fase-fase yang ada. Keputusan semi terstruktur (*semistructured decision*) ditandai dengan dengan peraturan-peraturan yang tidak lengkap untuk mengambil keputusan, dan adanya kebutuhan untuk membuat penilaian serta pertimbangan subjektif sebagai pelengkap analisis data yang formal. Menetapkan anggaran pemasaran untuk suatu produk baru adalah contoh dari keputusan semi terstruktur. Walaupun keputusan seperti ini biasanya tidak dapat secara penuh diotomatisasikan, namun sering didukung dari komputer (*computer-based decision*). Contoh keputusan jenis adalah investasi keuangan, pengevaluasian kredit, penjadwalan produksi, pemberian dana rehabilitasi sekolah, dan pengendalian persediaan.

2.2.3 Sistem Pendukung Keputusan

Pengertian sistem pendukung keputusan yang dikemukakan oleh McLeod (1998) yang menyatakan bahwa sistem pendukung keputusan merupakan sistem penghasil informasi yang ditujukan pada suatu masalah yang harus dibuat oleh manajer, sistem pendukung keputusan merupakan suatu sistem informasi yang ditujukan untuk membantu manajemen dalam memecahkan masalah yang dihadapinya.

Definisi selengkapnya adalah sistem penghasil informasi spesifik yang ditujukan untuk memecahkan suatu masalah tertentu yang harus dipecahkan oleh manajer pada berbagai tingkatan. Sistem pendukung keputusan juga merupakan suatu sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model (Pratiwi, 2016:4).

A. Karakteristik dan kapabilitas DSS



Sumber: Turban, E., dkk (2005:142)

Gambar 2.1 Karakteristik dan kapabilitas kunci dari DSS

1. Dukungan untuk pengambil keputusan, terutama pada situasi semistruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi. Masalah-masalah tersebut tidak dapat dipecahkan (atau tidak dapat dipecahkan dengan konvenien) oleh sistem komputer lain atau oleh metode atau alat kuantitatif standar.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok. Misalnya yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain. DSS mendukung tim virtual melalui alat-alat Web kolaboratif.
4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan di semua fase proses pengambilan keputusan: inteligensi, desain, pilihan dan implementasi.
6. Dukungan di berbagai proses dan gaya pengambilan keputusan.



7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat, dan dapat mengadaptasikan DSS untuk memenuhi perubahan tersebut. DSS bersifat fleksibel dan karena itu pengguna dapat menambahkan, menghapus, menggabungkan, mengubah, atau menyusun kembali elemen-elemen dasar. DSS juga fleksibel dalam hal dapat dimodifikasi untuk memecahkan masalah lain yang sejenis.
8. Pengguna merasa seperti rumah. Ramah-pengguna, kapabilitas grafis yang sangat kuat, dan antarmuka manusia mesin interaktif dengan satu bahasa alami dapat sangat meningkatkan keefektifan DSS. Kebanyakan aplikasi DSS yang baru menggunakan antarmuka berbasis-Web.
9. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, timeliness, kualitas) ketimbang pada efisiensinya (biaya pengambilan keputusan). Ketika DSS disebarkan, pengambilan keputusan sering membutuhkan waktu lebih lama, namun keputusan lebih baik.
10. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam langkah memecahkan suatu masalah. DSS secara khusus menekankan untuk mendukung pengambilan keputusan, bukannya menggantikan.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sendiri sistem sederhana. Sistem yang lebih besar dapat dibangun dengan bantuan ahli sistem informasi. Perangkat lunak OLAP dalam kaitannya dengan data warehouse membolehkan pengguna untuk membangun DSS yang cukup besar dan kompleks.
12. Biasanya model-model digunakan untuk menganalisis situasi pengambilan keputusan. Kapabilitas pemodelan memungkinkan eksperimen dengan berbagai strategi yang berbeda dibawah konfigurasi yang berbeda.
13. Akses disediakan untuk berbagai sumber data, format, dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi-objek.
14. Dapat dilakukan sebagai alat standalone yang digunakan oleh seorang pengambil keputusan pada satu lokasi dan di beberapa organisasi sepanjang rantai persediaan. Dapat diintegrasikan dengan DSS lain dan atau aplikasi



lain, dan dapat didistribusikan secara internal dan eksternal dengan menggunakan networking dan teknologi Web.

B. Tahapan Pengambilan Keputusan

Alur/ proses pemilihan alternatif tindakan/ keputusan biasanya terdiri dari langkah-langkah berikut (Pratiwi, 2016:10-11):

1. Tahap *Intelligence*

Pencarian kondisi-kondisi yang dapat menghasilkan keputusan.

Suatu tahap proses seseorang dalam rangka pengambil keputusan untuk permasalahan yang dihadapi, terdiri dari aktivitas penelusuran, pendeteksian serta proses pengenalan masalah. Data masukan diperoleh, diuji dalam rangka mengidentifikasi masalah.

2. Tahap *Design*

Menemukan, mengembangkan dan menganalisis materi-materi yang mungkin untuk dikerjakan.

Tahap proses pengambil keputusan adalah tahap *intelligence* meliputi proses untuk mengerti masalah, mengenali solusi dan menguji kelayakan solusi. Aktifitas yang biasanya dilakukan seperti menemukan, mengembangkan dan menganalisa alternatif tindakan yang dapat dilakukan.

3. Tahap *Choice*

Pemilihan dari alternatif pilihan yang tersedia, mana yang akan dikerjakan.

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

4. Tahap *Implementation*

Implementasi dari SPK yang telah dipilih. Tahap implementasi adalah tahap pelaksanaan dari keputusan yang telah diambil. Pada tahap ini perlu disusun serangkaian tindakan yang terencana, sehingga hasil keputusan dapat dipantau dan disesuaikan apabila diperlukan perbaikan.

C. Komponen-komponen DSS



Aplikasi DSS dapat terdiri dari subsistem sebagai berikut (Turban, E., dkk, 2005:143-144).

1. Subsistem manajemen data

Subsistem manajemen data. Subsistem manajemen data memasukkan satu database yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut **sistem manajemen database (DBMS)**. Subsistem manajemen data dapat diinterkoneksi dengan **data warehouse** perusahaan, suatu repositori untuk data perusahaan yang relevan untuk pengambilan keputusan. Biasanya data disimpan atau diakses via server web database.

2. Subsistem manajemen model

Subsistem manajemen model. Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak ini sering disebut **sistem manajemen basis model (MBMS)**. Komponen ini dapat dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model. Sistem manajemen dan metode solusi model diimplementasikan pada sistem pengembangan Web (seperti java) untuk berjalan pada server aplikasi.

3. Subsistem antarmuka pengguna

Subsistem antarmuka pengguna. Pengguna berkomunikasi dengan dan memerintahkan DSS melalui subsistem ini. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari DSS berasal dari interaksi yang intensif antara komputer dan pembuat keputusan. Browser Web memberikan struktur antarmuka pengguna grafis yang familier dan konsisten bagi kebanyakan DSS.

4. **Subsistem manajemen berbasis-pengetahuan**

Subsistem manajemen berbasis-pengetahuan. Subsistem ini dapat mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen. Ia memberikan intelegensi untuk memperbesar pengetahuan si



pengambil keputusan. Subsistem ini dapat diinterkoneksi dengan repositori pengetahuan perusahaan (bagian dari sistem manajemen pengetahuan) yang kadang-kadang disebut basis pengetahuan organisasional. Pengetahuan dapat disediakan via server web. Banyak metode kecerdasan tiruan diimplementasikan dalam sistem pengembangan web seperti Java, dan mudah untuk diintergrasikan dengan komponen DSS lainnya.

2.2.4 Perjalanan Dinas

Menurut Kamus Besar Bahasa Indonesia (dikutip Anggraini,2017:7), "Perjalanan diartikan sebagai kegiatan berpergian ke suatu tempat, dan Dinas diartikan sebagai melakukan tugas atau kewajiban".

Menurut Wursanto (2006:209) adalah "Perjalanan dinas yang dilakukan oleh pimpinan suatu lembaga atau perusahaan dalam rangka melaksanakan tugas kedinasan".

Dari definisi di atas, maka dapat disimpulkan perjalanan dinas merupakan perjalanan yang dilakukan oleh pejabat/pegawai suatu lembaga/perusahaan yang berkaitan dengan tugas suatu lembaga/perusahaan yang berkaitan dengan tugas pekerjaan kedinasan. Tugas pekerjaan kedinasan adalah tugas pekerjaan yang berkaitan dengan kepentingan lembaga/perusahaan yang bersangkutan.

2.2.5 Metode *Weighted Product*

Metode *weighted product* adalah salah satu metode yang sederhana dengan perkalian untuk menghubungkan rating atribut dimana setiap atribut harus dipangkatkan dengan bobot atribut yang bersangkutan (Latif,Jamil,dkk ,2018:26),"

Berikut adalah beberapa tahapan dan perumusan perhitungan dengan metode *weighted product* (Latif,Jamil,dkk ,2018:26) :

1. Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan sebagai tolok ukur penyelesaian masalah.

Pada tahap ini, mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan tolok ukur penyelesaian masalah. Adapun kriteria dari proses



penentuan pegawai yang akan melaksanakan perjalanan dinas yaitu ada 2 jenis penentuan pegawai perjalanan dinas didalam negeri dan diluar negeri. Berikut aspek penentuan pegawai yang melaksanakan perjalanan dinas didalam negeri yaitu jabatan, pendidikan, keahlian, loyalitas, terakhir perjalanan dinas, dan kesediaan. Dan aspek penentuan pegawai yang melaksanakan perjalanan dinas diluar negeri yaitu kemampuan bahasa asing, jabatan, pendidikan, keahlian, loyalitas, terakhir perjalanan dinas, dan kesediaan.

2. Menormalisasi setiap nilai *alternative* (nilai *vector*).

Berikut ini adalah rumus untuk melakukan normalisasi setiap *alternative* (nilai *vector*) yaitu sebagai berikut :

$$S_i = \prod_{j=1}^n X_{ij}^{w_j}$$

dengan $i = 1, 2, \dots, m$.

dimana :

S : Preferensi *alternative* dianalogikan sebagai vektor S

X : Nilai kriteria

W : Bobot kriteria / subkriteria

i : *Alternative*

j : Kriteria

n : Banyaknya kriteria

3. Menghitung nilai bobot preferensi pada setiap *alternative*.

Dimana $\sum W_j = 1$. Variabel W_j adalah pangkat bernilai positif untuk atribut keuntungan dan bernilai negatif untuk atribut biaya.

Preferensi relatif dari setiap alternatif, diberikan oleh Persamaan (2) :



$$V_i = \frac{\prod_{j=1}^n X^{ij} w_j}{\prod_{j=1}^n (X^{ij}) w_j}$$

V : Preferensi alternative dianalogikan sebagai *vector* V

X : Nilai Kriteria

W : Bobot kriteria / sub kriteria

i : *Alternative*

j : Kriteria

n : Banyaknya kriteria

4. Melakukan perankingan

Hasil akhir dari metode *weighted product* adalah ranking dari kandidat untuk pegawai yang akan melaksanakan perjalanan dinas baik didalam negeri maupun diluar negeri.

Dari definisi di atas, maka dapat disimpulkan metode *Weighted Product* merupakan salah satu metode yang menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan dulu dengan bobot atribut yang bersangkutan.

Adapun kelebihan dan kekurangan dari metode *Weighted Product* ini adalah sebagai berikut:

a. Kelebihan Metode WP

1. Mempercepat proses perhitungan nilai kriteria dan perankingan untuk setiap alternatif.
2. Mempermudah user untuk memberikan pembobotan terhadap kriteria yang memiliki nilai yang hampir sama.
3. Dapat digunakan untuk pengambilan keputusan *single* dan keputusan multidimensional.
4. Metode ini digunakan untuk menyelesaikan pengambilan keputusan secara praktis, karena konsepnya sederhana dan mudah dipahami.

b. Kelemahan Metode WP



1. Tidak banyak *user* yang menggunakan metode ini dalam pengambilan keputusan.
2. Metode ini hanya metode matematis tanpa ada pengujian secara statistik sehingga tidak ada batas kepercayaan dari kebenaran model yang terbentuk.

2.2.6 Penerapan Metode Weighted Product pada Sistem Pendukung Keputusan Penentuan Pegawai dalam Melaksanakan Perjalanan Dinas (Studi Kasus: PT. Permata Karya Jasa Perusahaan Gas Negara)

Sistem pendukung keputusan penentuan pegawai dalam melaksanakan perjalanan dinas menggunakan metode *weighted product* pada PT. Permata Karya Jasa Perusahaan Gas Negara adalah suatu sistem yang membantu untuk menentukan pegawai yang akan melaksanakan perjalanan dinas dengan menggunakan metode *weighted product* (WP) yang mana akan memilih sesuai kriteria yang telah ditentukan.

2.3 Teori Khusus

2.3.1 Unified Modeling Language (UML)

2.3.1.2. Pengenalan Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2018:133).



2.3.1.3 Sejarah *Unified Modeling Language* (UML)

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi- metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa A.S dan M. Shalahudin, 2014:138).

2.3.1.4. Diagram *Unified Modeling Language* (UML)

Rosa A.S dan M. Shalahudin (2018:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.



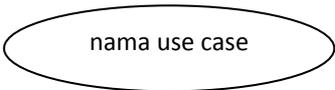
1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

2.3.1.5. Use Case Diagram

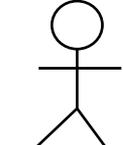
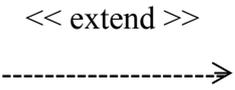
Menurut Rosa & Shalahuddin (2018:155) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Menurut Rosa & Shalahuddin (2018:156-158) simbol-simbol yang ada pada diagram *use case* yaitu:

Tabel 2.1 Simbol-simbol *Use Case* Diagram

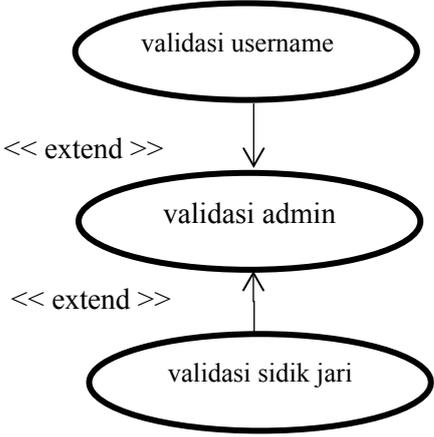
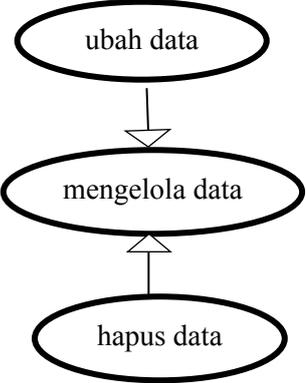
No.	Simbol	Deskripsi
1.	<i>Uses case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .



2.	<p>Actor / <i>actor</i></p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
3.	<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4.	<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use</i></p>



Lanjutan Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
		<p>case yang ditambahkan, misal</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	Generalisasi generalization 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>



Lanjutan Tabel 2.1 Simbol-simbol Use Case Diagram

No.	Simbol	Deskripsi
6.	Menggunakan / <i>include</i> / <i>uses</i> << include >> —————> << uses >> —————>▲	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none">• Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <pre>graph BT; login((login)) -- "<< include >>" --> validasi_username((validasi username));</pre> <ul style="list-style-type: none">• Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut: <pre>graph BT; ubah_data((ubah data)) -- "<< include >>" --> validasi_user((validasi user));</pre> <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung ada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Rosa &Salahudin, 2018.



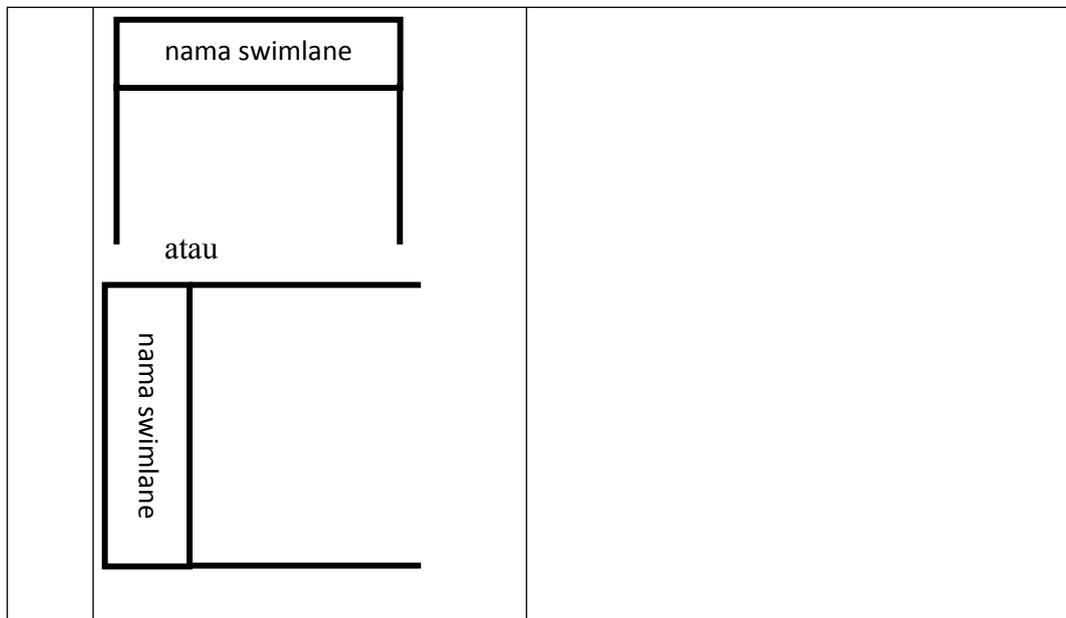
2.3.1.6 Activity Diagram

Rosa & Shalahuddin (2018:161) mengemukakan, “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 actor, jadi aktivitas yang dapat dilakukan oleh sistem. “Menurut Rosa & Shalahuddin (2018:162—163) simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Pecabangan / <i>decision</i> 	Asosiasi pecabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Lanjutan **Tabel 2.2** Simbol-simbol *Activity Diagram*



Sumber: Rosa & Salahudin, 2018.

2.3.1.7 Class Diagram

Menurut Rosa & Shalahuddin (2018:141) diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

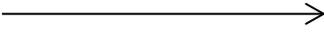
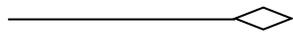
Menurut Rosa & Shalahuddin (2018:146-147) simbol-simbol yang ada pada diagram kelas:

Tabel 2.3 Simbol-simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem.
2.	<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Lanjutan **Tabel 2.3** Simbol-simbol *Class Diagram*



3.	Asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi / aggregation 	Relasi antar kelas dengan makna semua-bagian (whole-part).

Sumber: Rosa &Salahudin, 2018.

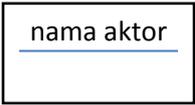
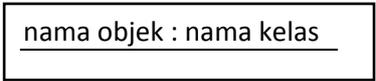
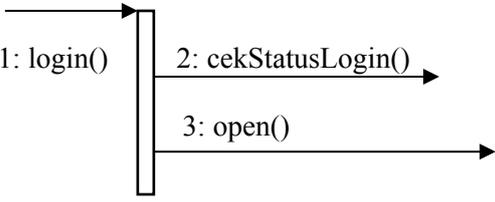
2.3.1.8 Sequence Diagram

Menurut Rosa & Shalahuddin (2018:165) Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

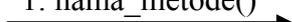
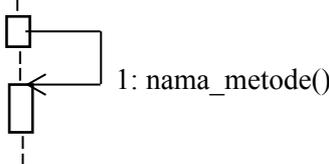
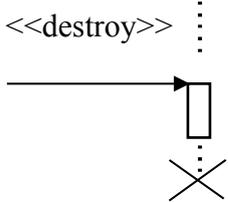
Menurut Rosa & Shalahuddin (2018:165-167) simbol-simbol yang ada pada *sequence diagram* yaitu:

Tabel 2.4 Simbol-simbol *Sequence Diagram*



No.	Simbol	Deskripsi
1.	Aktor  atau  tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang kan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.
3.	Objek 	Menyatakan objek yang berinteraksi pesan.
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya:  Maka cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.

Lanjutan Tabel 2.4 Simbol-simbol *Sequence* Diagram

No.	Simbol	Deskripsi
5.	Pesan tipe create <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe call 1: nama_metode() 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 1: masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 1: keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber: Rosa &Salahudin, 2018.

2.4 Teori Program



2.4.1 Bootstrap

Bootstrap merupakan salah satu framework CSS paling populer dari sekian banyak framework CSS yang ada. Bootstrap memungkinkan desain sebuah web menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik (Abdulloh,2018:261).

2.4.2 CSS (*Cascading Style Sheet*)

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan (Abdulloh,2018:45).

2.4.3 Database

Basis data (*database*) dapat didefinisikan sebagai himpunan kelompok data saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah (Hidayatullah dan Kawistara, 2017:147).

2.4.4 HTML (*Hypertext Markup Language*)

Hypertxt Markup Language (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman *web*. Yang bisa dilakukan HTML yaitu (Hidayatullah dan Kawistara, 2017:13).

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halamn *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, *java applet* dalam halaman *web*.
6. Menampilkan area gambar (*canvas*) di *browser*.

2.4.5 MySQL



MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Kelebihan dari *MySQL* adalah gratis, handal, selalu di-*update* dan banyak form yang memfasilitasi para pengguna jika memiliki kendala. *MySQL* juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah (Hidayatullah dan Kawistara, 2017: 180).

2.4.6 PHP (*Hypertext Preprocessor*)

PHP *Hypertext Preprocessor* atau disingkat dengan PHP ini adalah suatu bahasa *scripting* khususnya digunakan untuk *web development*. Karena sifatnya yang *server side scripting*, maka untuk menjalankan PHP harus menggunakan *web server* (Hidayatullah dan Kawistara, 2017: 231).

2.4.7 XAMPP

Sutanto (2014:72) XAMPP merupakan singkatan dari X (empat operasi apapun), Apache, MySQL, PHP, dan Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP server, PhpMyAdmin dan berbagai pustaka bantu lainnya.

Menurut Pratama, I Putu Agus Eka (2014 : 440) “XAMPP adalah aplikasi *web server* bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi Linux maupun di sistem operasi Windows.

2.4.8 Black Box Testing

Menurut Sukanto dan Shalahuddin (2018:275), “Menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.

2.5 Metode Pengembangan *Rational Unified Process* (RUP)

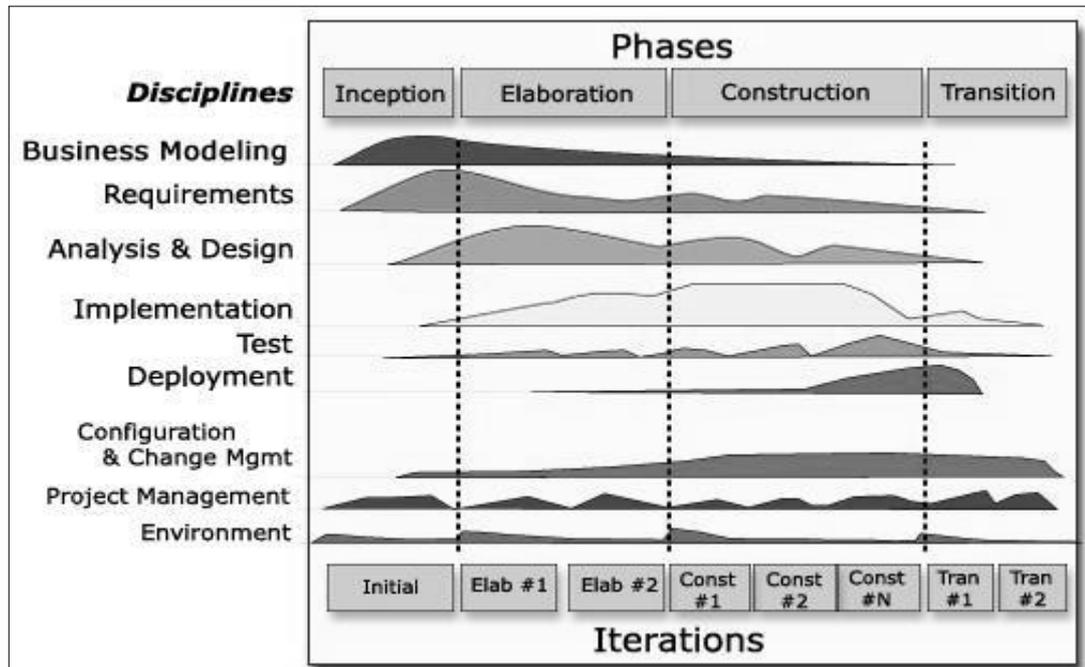


Menurut Suryana (2007) Rational Unified Process (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai best practises yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan use- case driven dan pendekatan iteratif untuk siklus pengembangan perangkat lunak.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

- **Dimensi pertama** digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction, dan Transition*.
- **Dimensi kedua** digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how dan when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration dan Change Manegement, Project Management, Environtment*.

Gambar dibawah menunjukkan secara keseluruhan arsitektur yang dimiliki RUP.



Gambar 2.2 Arsitektur *Rational Unified Process*

Sumber : Suryana, 2007

2.5.1 Fase Rational Unified Process (RUP)

RUP (Rational Unified Process) memiliki empat tahap atau fase yang dapat dilakukan secara iteratif. Berikut ini penjelasan untuk setiap fase pada *RUP (Rational Unified Process)*.

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

b. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*.



Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada *RUP (Rational Unified Process)* dijalankan secara berurutan dan iteratif dimana setiap iterasi dapat digunakan untuk memperbaiki iterasi berikutnya.

2.5.2 Aliran Kerja Utama *Rational Unified Process (RUP)*

Adapun aliran kerja utama pada Metodologi *Rational Unified Process (RUP)* adalah sebagai berikut:

a. *Pemodelan Bisnis (Business Modeling)*

Mendeskripsikan struktur dan proses-proses bisnis organisasi.

b. *Kebutuhan (Requirement)*

Mendefinisikan kebutuhan perangkat lunak dengan menggunakan metode *use case*.

c. *Analisis dan Perancangan (Analysis and Design)*

Mendeskripsikan berbagai arsitektur perangkat lunak dari berbagai sudut pandang.

d. *Implementasi (Implementation)*

Menuliskan kode-kode program, menguji, dan mengintegrasikan unit-unit programnya.

e. *Pengujian (Test)*

Mendeskripsikan kasus uji, prosedur, dan alat ukur pengujian.

f. *Deployment*

Menangani konfigurasi sistem yang akan diserahkan.

2.5.3 Aliran Kerja Pendukung *Rational Unified Process (RUP)*

Adapun aliran kerja pendukung *Rational Unified Process (RUP)* adalah sebagai berikut:

a. *Manajemen konfigurasi dan perubahan (configuration and change management)*



Mengendalikan perubahan dan memelihara artifak-artifak proyek.

b. Manajemen proyek (*Project Management*)

Mendeskripsikan berbagai strategi pekerjaan dengan proses yang berulang.

c. Lingkungan (*Environment*)

Menangani infrastruktur yang dibutuhkan untuk mengembangkan sistem.

2.6 Metode Pengujian *Black Box*

Metode pengujian digunakan untuk mengetahui fungsi yang telah ditentukan bahwa suatu sistem telah dirancang dapat menunjukkan bahwa masing-masing fungsi sepenuhnya beroperasi. Pengujian kotak hitam (*black box*), juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak.

Artinya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam (*black box*) bukan teknik alternatif untuk kotak putih (*white box*).

Pengujian kotak hitam (*black box*) berupaya untuk menemukan kesalahan dalam kategori berikut: (1) fungsi yang salah atau hilang, (2) kesalahan dalam struktur data atau akses basis eksternal, (4) kesalahan perilaku atau kinerja, dan (5) kesalahan inisialisasi dan penghentian (Pressman, 2012:597).

2.7 Kamus Data

Menurut Sukanto dan Shalahuddin (2018:73), “Kamus Data (*Data Dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus Data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa Kamus Data adalah alur dari suatu aplikasi yang terdiri dari masukan, proses dan keluaran.