

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Teori sistem melahirkan konsep-konsep futuristic. Salah satu konsep yang terkenal adalah upaya sibernetika. Konsep bidang kajian ilmiah ini terutama berkaitan dengan upaya menerapkan berbagai disiplin ilmu, yaitu ilmu perilaku, fisika, biologi, dan teknik. Oleh karena itu, sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan oleh manusia sehingga melahirkan studi tentang robotika, kecerdasan buatan (*Artificial Intelligence*), dan lain adalah masukan, pengolahan, serta keluaran (Tata Sutabri, 2012)

Menurut Anggraeni, dkk (2017:1-2), Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan.

2.1.1. Definisi Sistem

Terdapat beberapa pengertian mengenai sistem menurut para ahli. Menurut Mulyadi (2016:1), sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan dengan lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Menurut Anastasia Diana & Lilis Setiawati (2011:3), sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu.

Pengertian sistem menurut Romney dan Steinbart (2015:3), sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar.

Menurut Tata Sutabri (2012:46), Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.

Sedangkan menurut Huttahean (2015:13), Sistem Informasi adalah suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Dengan demikian dapat kita disimpulkan bahwa sistem informasi adalah suatu kombinasi modul yang terorganisir yang berasal dari komponen yang terkait dengan *hardware*, *software*, Sumber Daya Manusia dan jaringan pendukung berdasarkan seperangkat komputer dan menghasilkan informasi untuk mencapai tujuan.

2.1.2. Karakteristik Sistem

Karakteristik sebuah sistem yang baik menurut Jerson Hutahaean (2014:3), harus memiliki beberapa hal diantaranya yaitu:

1. Komponen Sistem

Sebuah sistem memiliki beberapa komponen yang saling berhubungan, yaitu saling bekerja sama dalam membentuk satu kesatuan sistem. Komponen ini terdiri dari komponen sistem yang lebih kecil atau subsistem. Setiap subsistem mempunyai sifat dari sistem yang menjalankan suatu fungsi tertentu yang mempengaruhi keseluruhan sistem. Dan sistem juga bisa memiliki sistem yang lebih besar lagi atau disebut supersistem.

2. Batasan sistem (*Boundary*)

Maksud dari batasan sistem adalah batasan antara suatu sistem dengan sistem yang lainnya. Batasan sistem tersebut dapat memungkinkan sistem dipandang sebagai satu kesatuan dimana hal ini tidak dapat dipisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah kondisi yang ada di luar sebuah sistem, dalam hal ini tetap mempengaruhi jalannya sistem itu sendiri. Lingkungan sendiri bisa memiliki sifat menguntungkan atau merugikan bagi sistem tersebut. Dengan hal itu maka harus dipertahankan lingkungan yang menguntungkan dan dikendalikan lingkungan yang merugikan. Tujuannya agar sistem yang ada agar tetap dapat bekerja dengan baik.

4. Penghubung sistem (*Interface*)

Penghubung sistem adalah alat bantu untuk menghubungkan antara suatu sistem dengan subsistem yang lainnya. Hal ini membuat sumber daya mengalir dari satu subsistem ke subsistem lain dimana keluaran dari satu subsistem akan menjadi masukan bagi subsistem yang menerima sumber daya tersebut. Dengan begitu dapat membuat sistem yang terintegrasi.

5. Masukan Sistem (*Input*)

Masukan sistem adalah energy yang masuk ke dalam sistem, bisa berupa pemeliharaan dan sinyal. Pemeliharaan input adalah energy yang dimasukkan ke dalam sistem supaya bisa beroperasi, sedangkan sinyal input adalah energy yang diproses di dalam sistem agar menghasilkan keluaran yang diinginkan.

6. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energy yang telah diolah dan diklasifikasikan menjadi keluaran yang memiliki nilai. Keluaran sistem ini akan menjadi masukan sistem bagi subsistem lain yang menerimanya.

7. Pengolah Sistem (*Process*)

Pengolah sistem adalah bagian dari sebuah sistem yang akan mengubah masukan mejadi keluaran. Sistem akan mengolah dari data mentah yang diterima melalui masukan dan mengubah menjadi informasi dan akan dikeluarkan oleh keluaran sistem.

8. Sasaran Sistem (*Objective*)

Sasaran sistem adalah tujuan dari sebuah sistem dimana masukan dan keluaran dari sistem akan ditentukan pada bagian ini. Suatu sistem memiliki tujuan yang jelas, jika tidak maka sasaran operasi tidak ada gunanya. Sebuah

sistem akan dikatakan berhasil apabila mengenai tujuan yang telah ditentukan.

2.1.3. Sistem Informasi

Sistem informasi merupakan proses pengumpulan, penyimpanan, analisis sebuah informasi dengan tujuan tertentu. Sistem informasi terdiri dari data (*input*) dan menghasilkan laporan (*output*) sehingga diterima oleh sistem lainnya serta kegiatan strategi dalam suatu organisasi dalam melakukan tindakan atau keputusan.

2.1.3.1. Konsep Sistem Informasi

Sistem informasi dipahami sebagai kumpulan atau suatu himpunan dari kelompok orang-orang yang bekerja, prosedur-prosedur, dan sumber daya peralatan yang mengumpulkan data dan mengolahnya menjadi sebuah informasi, merawat, dan menyebarkan informasi tersebut dalam suatu organisasi atau perusahaan.

Marimin et al. (2006) membuat pemahaman terhadap sistem informasi menjadi lebih sederhana, yaitu sebagai komponen-komponen dalam organisasi atau perusahaan yang berhubungan dengan proses penciptaan dan pengaliran informasi yang nantinya digunakan baik oleh satu atau lebih pengguna. Pengguna tersebut biasanya tergabung dalam suatu kelompok atau organisasi formal, seperti departemen atau kelompok lain seperti direktorat, bidang, bagian sampai pada unit terkecil. Pada sistem informasi di dalamnya termuat banyak informasi penting mengenai segala hal seperti orang, tempat, dan segala sesuatu yang ada di dalam atau di lingkungan sekitar organisasi. Informasi menjelaskan mengenai organisasi dan mengenai apa yang telah terjadi pada masa lalu, saat ini, dan yang mungkin akan terjadi pada masa depan tentang organisasi tersebut.

2.1.3.2. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu:

a. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

b. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu. untuk menghasilkan keluaran yang sudah diinginkan.

c. Blok keluaran (*output block*)

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

d. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan, teknologi terdiri dari unsur utama:

1. Teknisi (*human ware* atau *brain ware*)
2. Perangkat lunak (*software*)
3. Perangkat keras (*hardware*)

e. Blok basis data (*data base block*)

Merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras. komputer dan digunakan perangkat lunak untuk memanipulasinya.

f. Blok kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya, Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat langsung diatasi.

2.1.4. Analisis Dan Perancangan Sistem

Rancang bangun sangat berkaitan dengan analisis dan perancangan sistem yang merupakan satu kesatuan untuk merancang dan membangun sebuah aplikasi.

Menurut ahli Mulyani (2017:80) pengertian Perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan pemakai sistem serta untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap. Dari penjelasan diatas dapat disimpulkan rancang bangun sistem merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang ada.

2.1.5. Konsep Dasar Web

Pada umumnya, *website* atau yang hanya disebut dengan web merupakan sekumpulan halaman yang terdiri dari beberapa laman serta berisi informasi dalam bentuk digital, baik tulisan (teks) maupun gambar animasi yang disediakan dalam internet. Dengan begitu dengan koneksi internet *website* bisa diakses oleh banyak pengguna di seluruh penjuru dunia.

Menurut Solichin (2016:3), *Web* merupakan bagian dari layanan yang dapat berjalan diatas teknologi internet. Halaman *web* dapat diakses dengan adanya teknologi *server* sebagai penyedia halaman *web*. *HTML* sebagai bahasa baku sebagai jalur pengiriman dokumen web.

2.1.6. HTML

Didalam bukunya, Solichin (2016:10) menjelaskan bahwa *HTML* merupakan singkatan dari *Hypertext Markup Language*. *HTML* dikembangkan pertama kali oleh Tim Berners-Lee bersamaan dengan protokol HTTP (*Hypertext Transfer Protocol*) pada tahun 1989. tujuan utama pengembangan *HTML* adalah untuk menghubungkan satu halaman *web* dengan halaman web lainnya.

HTML sekarang ini sudah berkembang tidak hanya berupa teks, tapi juga berupa media lainnya seperti gambar, video dan suara.

2.1.7. PHP

Selain *HTML* terdapat bahasa pemrograman berbasis web lainnya yang umumnya dipakai dalam pengembangan sistem aplikasi berbasis *web* yaitu *PHP*. Menurut Solichin (2016:11) dalam bukunya dikatakan bahwa *PHP* merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk

pengembang *web*. *PHP* pertama kali dikembangkan oleh Rasmus Lerdorf, seorang pengembang *software* dan anggota tim *Apache*, dan dirilis pada akhir tahun 1994.

PHP dikatakan sebagai bahasa sisi server (*server-side*) oleh sebab itu, kode *PHP* tidak akan terlihat pada saat user memilih perintah “*view source*” pada web browser yang mereka gunakan. *Syntax* Program *PHP* adalah bahasa yang dirancang untuk mudah diletakkan dalam kode *HTML*. Banyak dijumpai kode *PHP* yang menyatu dengan kode *HTML*. Kode *PHP* diawali dengan tag `<?php` dan diakhiri dengan tag `?>`. apabila kita melakukan konfigurasi terhadap file *PHP* ini untuk mengizinkan pengguna tag pendek (*short tag*) dengan mengubah nilai *short_open_tag* menjadi *on*, maka tag tersebut dapat digantikan dengan `<?` dan `?>`. dalam *PHP*, nilai *default* dari *short_open_tag* adalah *off*.

Berikut ini contoh kode *PHP*:

```
<?php
    echo "Hello World";
?>
```

Perintah *echo* di dalam *PHP* berguna untuk mencetak nilai, baik teks maupun numerik ke layar *web browser*.

2.1.8. XAMPP

Menurut Solichin (2016:15) salah satu software package dalam pengembangan web adalah XAMPP (<http://apachefriends.org>). XAMPP mengumpulkan beberapa perangkat lunak pengembangan web seperti *PHP*, *Apache*, *MySQL* dan *PHPMyAdmin* menjadi satu buah *instaler* saja. Proses konfigurasi juga dilakukan secara otomatis sehingga menjadi mudah dan praktis.

Dengan menginstall *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. *XAMPP* akan menginstal dan mengkonfigurasikannya secara otomatis untuk anda atau dikenal dengan sebutan *auto konfigurasi*.

2.1.9. CSS (*Cascading Style Sheet*)

Selain bahasa pemrograman web disebutkan diatas, terdapat pula bahasa pemrograman web yang fungsinya untuk mempercantik halaman web. Menurut Solichin (2016:10), CSS atau *Cascading Style Sheet* pertama kali diusulkan oleh Hakon Wium Lie pada tahun 1994 dan selanjutnya distandarisasi oleh W3C. CSS memberikan cara yang mudah dan efisien bagi pemogram untuk menentukan tata letak halaman web dan mempercantik halaman dengan elemen desain seperti warna, sudut bulat, gradien dan animasi.

2.1.10. Basis Data

Menurut Tata Sutabri (2012:47), basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu sama lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas.

2.1.11. MySQL (*My Structure Query Language*)

Menurut Anhar (2010:45), MySQL (*My Structure Query Language*) adalah salah satu *DataBase Management System* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, Postgre SQL dan lainnya. MySQL berfungsi untuk

mengolah *database* menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakan secara gratis. Pemrograman PHP juga sangat mendukung/ *support* dengan *database* Mysql.

2.1.12. Sistem *Monitoring*

Monitoring merupakan suatu siklus kegiatan yang meliputi: pengumpulan, peninjauan ulang, pelaporan, dan tindakan atas informasi suatu proses yang sedang diterapkan (Effendy & Noquba, 2016). Sedangkan menurut (Chairiansyah, 2018) mengemukakan bahwa Sistem informasi monitoring adalah sebuah sistem informasi yang mendukung kegiatan *monitoring* atau pemantauan.

Tujuan sistem *monitoring* menurut Amsler, dkk dalam (Effendy & Noquba, 2016) Meliputi:

1. Memastikan suatu proses dilakukan sesuai dengan kebijakan yang berlaku sehingga proses berjalan sesuai dengan jalur yang disediakan (*on the track*).
2. Menyediakan peluang yang tinggi akan keakuratan data bagi pelaku *monitoring*.
3. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat (tanpa menunggu proses berakhir).
4. Mengembangkan motivasi dan kebiasaan positif.

2.1.13. Proyek

Proyek adalah suatu usaha sementara yang dilaksanakan untuk menghasilkan suatu produk atau jasa yang unik (Yasin, 2012). Dalam hal ini sebuah proyek bersifat sementara yang berarti memiliki jangka waktu tertentu. Proyek

memiliki awal dan akhir yang pasti dalam setiap tahapan pengerjaannya, sehingga proyek akan dihentikan jika memang tujuan proyek sendiri telah berhasil tercapai ataupun dihentikan dalam keadaan tujuan proyek yang belum tercapai (*Project Management Institute*, 2008).

2.1.14. Model Pembelajaran *Project Based Learning*

Berdasarkan kriteria umum yang dipublikasikan oleh lembaga yang ditugaskan oleh Departemen Tenaga Kerja Amerika Serikat yaitu SCANS (*The Secretary's Commission on Achieving Necessary Skills*), perubahan lingkungan di abad 21 menuntut siswa untuk dapat bertanggung jawab terhadap diri sendiri dan masyarakat; mampu berpikir kritis, merencanakan, menalar, dan kreatif; memiliki kemampuan komunikasi yang baik; memiliki rasa toleransi terhadap perbedaan budaya; mampu membuat keputusan dan memvisualisasikan suatu masalah; mengetahui bagaimana dan kapan menggunakan teknologi (*Educational Technology Division Ministry of Education*, 2006). Hal ini sesuai dengan paradigma kurikulum 2013 yang diterapkan pada pendidikan di Indonesia. Paradigma pembelajaran dalam kurikulum 2013 yaitu orientasi pembelajaran berpusat pada murid (*student centered learning*). Orientasi pembelajaran tersebut mengharuskan siswa untuk berpartisipasi secara aktif, selalu berpikir kritis, mampu menganalisis dan dapat memecahkan masalah-masalahnya (Rahayu & Nuryata, 2010).

Tugas guru dalam rangka mewujudkan model *student centered learning* yaitu guru dituntut untuk melakukan banyak aktivitas yang terkait dengan persiapan pembelajaran. Guru harus mampu menjadi fasilitator untuk membantu siswa memenuhi kriteria umum kemampuan siswa abad 21 (*Educational Technology*

Division Ministry of Education, 2006). Salah satu cara guru dapat menjadi fasilitator yaitu dengan guru mampu menggunakan teknologi dalam pembelajaran. Hal ini bertujuan untuk membantu siswa menggunakan teknologi pada proses pembelajaran. Namun, Sebagian pendidik masih belum menggunakan teknologi dalam proses pembelajaran. Padahal salah satu ciri pendidik yang kompeten yaitu mampu menerima dan menggunakan teknologi dalam pembelajaran (Borthwick & Hansen, 2017). Menurut Mouza (2016), penggunaan teknologi dengan dengan bantuan pendekatan kolaboratif dalam proses belajar mengajar dapat meningkatkan pemahaman dan daya cipta siswa.

Penggunaan teknologi secara efektif dalam proses belajar mengajar dipengaruhi oleh kemampuan pendidik dalam mengembangkan pengetahuan dan kemampuan penguasaan materi dan pedagoginya sesuai dengan perkembangan media digital. (Herring, Koehler, & Mishra, 2016). Berdasarkan U.S. *Department of Education's Office of Educational Technology* (OET) dalam (Borthwick & Hansen, 2017), beberapa syarat pendidik inovatif yaitu aktif menggunakan teknologi dalam proses belajar mengajar, terus berkembang dan memperbaiki diri dalam menggunakan teknologi, memiliki pengalaman mengajar dengan menggunakan beberapa model pembelajaran, dan berusaha untuk melakukan penelitian dan mencoba kerangka kerja pembelajaran (*framework*). Menurut Isman dalam (Aguinaldo, 2016), kontribusi teknologi dalam pendidikan yaitu mempercepat proses distribusi informasi, dapat menyesuaikan dengan keadaan proses distribusi informasi, dapat menyesuaikan dengan keadaan belajar siswa, dapat menciptakan pembelajaran permanen, dapat menjadi dasar pembelajaran berbasis proyek, dan memberikan kesempatan belajar secara global. Koehler &

Mishra (2005) menyatakan bahwa penggunaan teknologi dan pendekatan pembelajaran sangat penting dalam meningkatkan pembelajaran pada materi tertentu dengan catatan pendidik mampu menggunakan model pembelajaran yang tepat terhadap teknologi yang ada. Model *Project Based Learning* lebih menekankan pembelajaran dengan otoritas siswa *student center learning*, pembelajaran kolaboratif, dan penilaian pembelajaran berdasarkan hasil kerja proyek siswa. Hal ini dapat meningkatkan perhatian siswa pada proses pembelajaran (Thomas, 2000).

Model *Project Based Learning* dikatakan sesuai dengan pembelajaran abad 21 karena model ini mendukung guru untuk memberikan penilaian asli terhadap siswa yang dilihat berdasarkan dokumentasi kemajuan dan perkembangan proyek siswa, demonstrasi proyek, dan kemampuan bekerjasama dengan teman sekelompoknya; mampu memperkenalkan pembelajaran seumur hidup karena memungkinkan siswa membangun pengetahuan baru dan mengontrol pembelajarannya sendiri di luar pendidikan sekolah; mampu menampung siswa dengan berbagai tipe belajar; dan penggunaan model yang dipadukan dengan teknologi dapat menjadikan siswa menjadi pekerja mandiri, mampu berpikir kritis, dapat menjadi pembelajar seumur hidup, dan mengajarkan cara bekerjasama dalam suatu proyek (*Educational Technology Division Ministry of Education*, 2006).

Model pembelajaran ini berguna untuk meningkatkan pembelajaran siswa dan mempersiapkan lulusannya untuk terjun ke lingkungan kerja *profesional* (Fernandes, 2014). Dalam pembelajaran berbasis proyek, guru hanya sebagai pengarah pembentukan pengetahuan dan siswa harus mengarahkan pembelajarannya sendiri. Kegiatan siswa harus terstruktur dan terfasilitasi agar

keberhasilan dan pembelajaran bermakna siswa dapat tercapai. Siswa harus dipantau dengan cermat melalui kemajuan tahapan proyek (Thomas, 2000). Menurut Yetkiner, Anderoglu, & Capraro dalam (Hope & Allen, 2009), proses pembelajaran berbasis proyek, pemantauan proyek siswa secara terus menerus dan adanya umpan balik dari guru sangat diperlukan. Akibatnya, guru dan siswa harus memberikan waktu lebih banyak daripada waktu yang dibutuhkan di ruang kelas tradisional. Pembuatan proyek dalam model pembelajaran *Project-based Learning* membutuhkan periode waktu yang panjang dalam pengerjaannya yaitu beberapa pertemuan hingga satu tahun dan membutuhkan kerjasama untuk mengerjakannya (Moursund, 1999), (Gultekin, 2007), dan (Simkins, 1999).

2.1.15. Metode *Critical Path*

Critical Path Method adalah metode analisis jalur kegiatan atau aktivitas dengan menunjukkan total waktu terlama dan waktu penyelesaian proyek tercepat dengan memprediksi durasi total waktu proyek (U. Nahdlatul, 2021). *Critical Path Method* pada dasarnya merupakan metode yang berorientasi pada waktu, dalam arti bahwa *Critical Path Method* akan berakhir pada penentuan waktu. Metode ini mengidentifikasi jalur kritis pada aktifitas yang ditentukan ketergantungan antar aktifitasnya. Aktifitas merupakan tugas spesifik yang mempunyai hasil yang dapat diukur dari durasi pengerjaannya diukur dari durasi pengerjaannya.

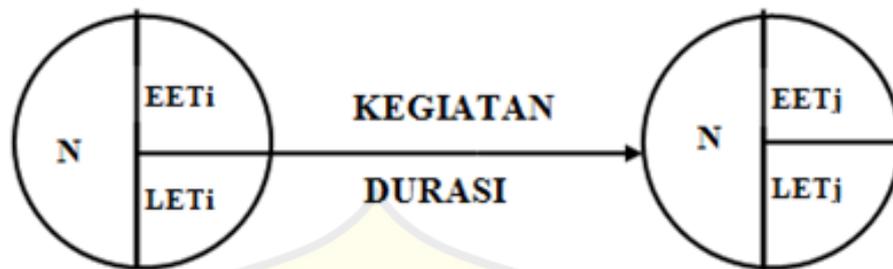
Critical path method (CPM) merupakan salah satu *tools* yang digunakan pada *project scheduling management* dimana *Schedule* atau penjadwalan merupakan salah satu faktor kesuksesan suatu proyek dengan nilai paling tinggi (M. Radujkovic and M. Sjekavica, 2017), (C.E.M. Serra and M. Kunc, 2015). *Critical path* adalah serangkaian aktivitas untuk menentukan *completion time* suatu proyek

yang ditentukan berdasarkan waktu minimum penyelesaian suatu proyek (M. F. Nagata, W. A. Manginelli, S. Lowe, and T. J. Trauner, *Construction delays*). Durasi aktivitas-aktivitas yang termasuk pada jalur kritis dapat mengontrol durasi keseluruhan proyek; penundaan salah satu atau lebih aktivitas pada jalur kritis akan menyebabkan terlambatnya tanggal selesai sebuah proyek. Total waktu yang dibutuhkan untuk menyelesaikan langkah ini adalah lamanya waktu yang dibutuhkan untuk menyelesaikan proyek.

CPM adalah metode penjadwalan proyek yang dapat digunakan untuk memperkirakan durasi minimum proyek dan menghitung *fleksibilitas* jadwal pengerjaan tugas proyek (*Project Management Institute*, 2013). CPM tidak mempertimbangkan keterbatasan SDM, CPM mengasumsikan SDM tak terbatas dengan menggunakan metode lanjutan hasil perhitungan CPM dapat diatur agar mempertimbangkan SDM (Addison-Wesley, 2011).

Metode jalur kritis (CPM) merupakan alat penjadwalan yang paling banyak digunakan dalam industri (E. Forcael, V. Gonzalez, F. Orozco, A. Opazo, A. Suazo, and P. Aranguiz, 2015). CPM disajikan dalam bentuk diagram jaringan yang diwakili oleh node yang menunjukkan hubungan dan ketergantungan dari kegiatan proyek (D. Satyanegara and F. Nurunnajmi, 2017). CPM pada dasarnya merupakan metode yang ber-orientasi pada waktu, dalam arti bahwa CPM akan berakhir pada penentuan waktu. Metode ini mengidentifikasi jalur kritis pada aktivitas yang ditentukan ketergantungan antar aktivitasnya (A. Nalhadi and N. Suntana, 2017). Dari CPM tersebut, dapat dilakukan analisa jaringan kerja, pengurangan, percepatan waktu, serta biaya yang optimal (D. A. Saputra M, E. Satria, and G. A. Pandey, 2017).

Penggambaran *Critical Path Method* menggunakan simbol yang dapat berbentuk segi empat atau lingkaran (R. Brando, 2017). Di bawah ini adalah gambar contoh penggambaran *Critical Path Method* untuk satu item pekerjaan.



Gambar 2. 1 Diagram CPM untuk satu item pekerjaan

Keterangan:

- Lingkaran disebut juga node menunjukkan berawalnya suatu pekerjaan ataupun berakhirnya suatu pekerjaan.
- Garis panah (*arrow*) menunjukkan pekerjaan, arah panah ke suatu node menunjukkan urutan antar pekerjaan.
- EETa : Saat paling awal pekerjaan dimulai
- EETb : Saat paling dini pekerjaan berakhir
- LETa : Saat paling lambat pekerjaan dimulai
- LETb : Saat paling lambat pekerjaan berakhir
- Durasi : Lama pekerjaan berlangsung
- N : Nomor pengidentifikasian node

2.1.15.1 Teknik Perhitungan Critical Path Method

1. Perhitungan Maju (*Forward Pass*)

Dimulai dari Start (*initial event*) menuju *Finish* (*terminal event*) untuk menghitung waktu penyelesaian tercepat suatu kegiatan (EF), waktu tercepat

terjadinya kegiatan (ES) dan saat paling cepat dimulainya suatu peristiwa (E) (A. Abdurrazyid, L. Luqman, A. Haris, and I. Indrianto, 2019).

Rumus untuk Hitungan Maju, yaitu: $EF(i-j) = ES(i-j) + D(i-j)$ (1)

Keterangan:

$EF(i-j)$: waktu penyelesaian tercepat suatu kegiatan

$ES(i-j)$: waktu tercepat terjadinya kegiatan

$D(i-j)$: durasi suatu kegiatan

2. Aturan dari rumus perhitungan maju (*Forward Pass*):

- a. Kecuali kegiatan awal, maka suatu kegiatan baru dapat dimulai bila kegiatan yang mendahuluinya (*predecessor*) telah selesai.
- b. Waktu selesai paling awal suatu kegiatan ($EF(i-j)$) samadengan waktu mulai paling awal ($ES(i-j)$) ditambah dengan kurun waktu ($D(i-j)$) kegiatan yang mendahuluinya.
- c. Apabila suatu kegiatan memiliki dua atau lebih kegiatan-kegiatan terdahulu yang menggabung, maka waktu mulai paling awal ($ES(i-j)$) kegiatan tersebut adalah sama dengan waktu selesai paling awal ($EF(i-j)$) yang terbesar dari kegiatan terdahulu.

3. Perhitungan Mundur (*Backward Pass*)

Dimulai dari *Finish* menuju *Start* untuk mengidentifikasi saat paling lambat terjadinya suatu kegiatan (LF), waktu paling lambat terjadinya suatu kegiatan (LS) dan saat paling lambat suatu peristiwa terjadi (L).

Rumus untuk Hitungan Mundur, yaitu : $LS(i-j) = LF(i-j) + D(i-j)$ (2)

Keterangan:

LS(i-j) : waktu paling lambat terjadinya suatu kegiatan

LF(i-j) : saat paling lambat terjadinya suatu kegiatan

D(i-j) : durasi suatu kegiatan

4. Aturan hitungan mundur (*Backward Pass*) seperti berikut:

Waktu mulai paling akhir suatu kegiatan sama dengan waktu selesai paling akhir dikurangi kurun waktu berlangsungnya kegiatan yang bersangkutan. Apabila suatu kegiatan terpecah menjadi 2 kegiatan atau lebih, maka waktu paling akhir (LF) kegiatan tersebut sama dengan waktu mulai paling akhir (LS) kegiatan berikutnya yang terkecil.

5. Total Float (TF)

Total Float menunjukkan jumlah waktu yang diperkenankan suatu kegiatan boleh ditunda, tanpa mempengaruhi jadwal penyelesaian proyek secara keseluruhan. Rumus perhitungan Total float, yaitu:

$$\mathbf{TF = LF(i-j) - EF(i-j) - D(i-j) \quad (3)}$$

Keterangan:

TF : Waktu luang yang diperkenankan suatu kegiatan ditunda

LF(i-j) : saat paling lambat terjadinya suatu kegiatan

EF(i-j) : waktu paling lambat suatu kegiatan

D(i-j) : Durasi suatu kegiatan

Untuk memanfaatkan float total, maka kegiatan terdahulu harus mulai seawal mungkin (ES), sebaliknya kegiatan berikutnya harus mulai selambat mungkin (LS).

2.1.15.2 Istilah – Istilah yang digunakan dalam *Critical Path Method* (CPM)

Menurut handoko (2000) dalam proses identifikasi jalur kritis ada beberapa istilah yang digunakan, yaitu :

a. *Earliest Start Time* (ES)

Waktu yang paling awal (tercepat) suatu kegiatan dapat dimulai dengan memperhatikan waktu kegiatan dan persyaratan urutan pengerjaan.

b. *Latest Start Time* (LS)

Waktu yang paling lambat untuk dapat memulai suatu kegiatan tanpa penundaan keseluruhan proyek.

c. *Earliest Finish Time* (EF)

Waktu yang paling awal kegiatan dapat diselesaikan atau dengan $ES +$ waktu kegiatan yang diharapkan.

d. *Latest Finish Time* (LF)

Waktu paling lambat untuk dapat menyelesaikan suatu kegiatan tanpa penundaan penyelesaian proyek secara keseluruhan atau sama dengan $LS +$ waktu kegiatan yang diharapkan.

e. *Float atau Slack*

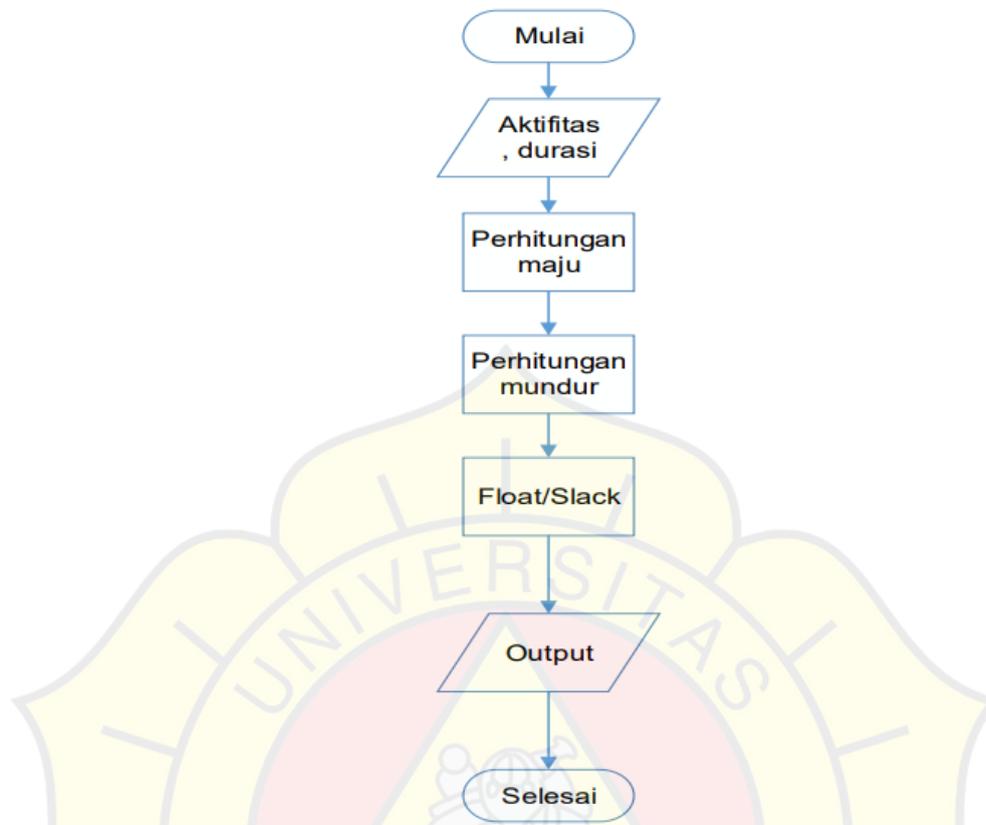
Float atau *Slack* adalah toleransi waktu telat sebuah aktifitas untuk dimulai dari *early start* (LS) tetapi tidak mengubah tenggang waktu keseluruhan proyek. Aktifitas – aktifitas pada jalur kritis memiliki *Float* atau *Slack*.

2.1.15.3 Kerangka Kerja *Critical Path Method*

Dalam penggunaan metode *Critical Path Method* terdapat kerangka kerja yang perlu dilakukan, Menurut Heizer dan Render (2014), *Critical Path Method* mengikuti enam langkah dasar sebagai berikut :

- 1) Menetapkan proyek dan menyiapkan struktur penguraian kerjanya.
- 2) Membangun hubungan antara aktivitas-aktivitasnya. Memutuskan aktivitas yang harus dilakukan lebih dahulu dan aktivitas yang harus mengikuti aktivitas lain.
- 3) Menggambarkan jaringan yang menghubungkan keseluruhan aktivitas.
- 4) Menetapkan perkiraan waktu dan/atau biaya untuk setiap aktivitas.
- 5) Menghitung jalur waktu terpanjang melalui jaringan. Hal ini disebut jalur kritis. Pada langkah kelima adalah penentuan jalur kritis, maka pada langkah inilah inti dari penjadwalan menggunakan metode CPM, Karena Aktivitas pada jalur kritis merepresentasikan tugas-tugas yang akan menunda keseluruhan proyek, kecuali mereka dapat diselesaikan secara tepat waktu.
- 6) Menggunakan jaringan untuk membantu perencanaan, penjadwalan, dan pengendalian proyek.

2.1.15.4 Tahapan Metode Jalur Kritis *Critical Path*



Gambar 2. 2 Diagram Alir Perhitungan dengan Metode CPM

Pada perhitungan maju, perhitungan bergerak dari *initial event* menuju terminal *event*. Maksudnya ialah menghitung saat yang paling cepat terjadinya *events* dan saat paling cepat dimulainya serta diselesaikannya aktifitas-aktifitas (TE, ES, dan EF).

Waktu kelaksanaan (*duration*) kegiatan A adalah 2 hari sehingga saat tercepat diselesaikannya aktifitas A adalah pada hari kedua atau $EF(0,1) = 2$. Karena aktifitas A ini adalah satu-satunya aktifitas yang memasuki *node* 1, maka saat tercepat terjadinya *event* nomor 1 juga pada hari kedua, atau

$TE(1) = 2$. Maka masukkan angka 2 ke dalam ruang kiri node 1.

$TE(2) = EF(1,2) = EF(0,1) + duration = TE(1) + 6 = 2 + 6 = 8$

$$TE(3) = EF(2,3) = EF(1,2) + duration = TE(2) + 5 = 8 + 5 = 13$$

$$TE(4) = EF(2,4) = EF(1,2) + duration = TE(2) + 4 = 8 + 4 = 12$$

Node 5 merupakan *merge event*, $EF(3,5) = 13 + 0 = 13$ dan $EF(4,5) = 12 + 0 = 12$. Maka $TE(5) = \max(13,12) = 13$, dan masukan angka 13 pada ruang kiri bawah dari node 4

$$TE(6) = EF(5,6) = EF(3,5) + duration = TE(5) + 9 = 13 + 9 = 22$$

$$TE(7) = EF(5,7) = EF(3,5) + duration = TE(5) + 7 = 13 + 7 = 20$$

$$TE(8) = EF(6,8) = EF(5,6) + duration = TE(6) + 6 = 22 + 6 = 28$$

$$TE(9) = EF(8,9) = EF(6,8) + duration = TE(8) + 9 = 28 + 9 = 37$$

Node 10 merupakan *merge event*, $EF(9,10) = 37 + 4 = 41$ dan $EF(7,10) = 20 + 5 = 25$. Maka $TE(10) = \max(41,25) = 41$, dan masukan angka 41 pada ruang kiri bawah dari node 10.

$$TE(11) = EF(10,11) = EF(9,10) + duration = TE(10) + 2 = 41 + 2 = 43$$

Pada perhitungan mundur, perhitungan bergerak dari dari terminal event menuju ke *initial event*. Tujuannya ialah untuk menghitung saat paling lambat terjadinya *events* dan saat paling lambat dimulainya dan diselesaikannya aktifitasaktifitas(TL, LS, dan LF).

Dari hasil perhitungan maju diperoleh $TE(11) = 43$, sehingga dengan sendirinya $TL(11) = 43$. Masukkan angka 43 pada ruang kanan bawah dari node 11. Aktifitas I dapat diselesaikan paling lambat pada hari ke-43 dengan *duration* 2 hari, maka aktifitas I dapat dimulai pelaksanaannya paling lambat setelah hari ke-43 - 2 = 41 sehingga $TL(10) = 41$

$$TL(9) = 41 - duration = 41 - 4 = 37$$

$$TL(8) = 37 - duration = 37 - 9 = 28$$

$$TL(7) = 41 - duration = 41 - 5 = 36$$

$$TL(6) = 28 - duration = 28 - 6 = 22$$

Node 5 merupakan *burst event* , $LS(5,6) = 22 - 9 = 13$ dan $EF(5,7) = 36 - 7 = 29$.

Maka $TL(5) = \min(13,29) = 13$, dan masukan angka 13 pada ruang kiri bawah dari node 5.

$$TL(4) = 13 - duration = 13 - 0 = 13$$

$$TL(3) = 13 - duration = 13 - 0 = 13$$

Node 2 merupakan *burst event* , $LS(2,3) = 13 - 5 = 8$ dan $EF(2,4) = 13 - 4 =$

9. Maka $TL(2) = \min(8,9) = 8$, dan masukan angka 8 pada ruang kiri bawah dari node 2

$$TL(1) = 8 - duration = 8 - 6 = 2$$

$$TL(0) = 2 - duration = 2 - 2 = 0$$

Setelah perhitungan maju dan perhitungan mundur selesai dilakukan, maka berikutnya dilakukan perhitungan kelonggaran waktu (float/slack) dari aktifitas (i,j), yang terdiri atas total float dan free float. Total float dihitung dengan cara mencari selisih antara saat paling lambat diselesaikannya aktifitas dengan saat paling cepat diselesaikannya aktifitas ($LF - ES$). Sedangkan free float aktifitas(i,j) dihitung dengan cara mencari selisih antara saat tercepat terjadinya event di ujung aktifitas dengan saat tercepat diselesaikannya aktifitas (i,j) tersebut.

Total float dapat dihitung dengan menggunakan rumus

$$S(i,j) = TL(j) - TE(i) - t(i,j)$$

Free float dapat dihitung dengan menggunakan rumus

$$SF(i,j) = TE(j) - TE(i) - t(i,j)$$

Suatu aktifitas yang tidak mempunyai kelonggaran (*float*) disebut aktifitas kritis. Dengan kata lain, aktifitas kritis mempunyai $S = SF = 0$. Pada kasus di atas, aktifitas kritisnya adalah aktifitas-aktifitas A, B, C, E, F, H, I, K dan L. aktifitas-aktifitas kritis ini akan membentuk lintasan kritis yang biasanya dimulai dari *initial event* sampai ke *terminal event*. Lintasan kritisnya adalah lintasan yang melalui node 0, 1, 2, 3, 5, 6, 8, 9, 10 dan 11.

2.1.16. ERD (*Entity Relationship Diagram*)

Menurut (Andriani dan Purnama, 2019) mengemukakan bahwa, “*Entity Relationship Diagram*” adalah sebuah penggambaran grafis untuk menggambarkan desain secara konseptual dari basis data yang akan dibangun”.

Model *Entity Relationship* ini digunakan untuk mentransformasikan data-data yang ada didunia nyata kedalam bentuk notasi-notasi sebagai perangkat konseptual menjadi diagram data yang dikenal dengan diagram *Entity-Relationship*. (Maniah & Hamidin, 2017).

2.1.17. Metode RAD (*Rapid Application Development*)

Menurut Habibi, dkk (2019:69), *Rapid Application Development* (RAD) yaitu suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak (Kendall dan Kendall, 2002).

Didalam bukunya juga Habibi, dkk (2019:69) menjelaskan bahwa *Rapid Application Development (RAD)* adalah model proses perkembangan *software* sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier serta perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen (Hariyanto, 2004). Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai 90 hari).

RAD digunakan pada aplikasi sistem konstruksi, maka menekankan *fase-fase*. Ada tiga *fase* dalam RAD yaitu (Kendall dan Kendall, 2008)

1. *Requirements Planning*

Dalam tahap ini diketahui apa saja yang menjadi kebutuhan sistem yaitu dengan mengidentifikasi kebutuhan informasi dan masalah yang dihadapi untuk menentukan tujuan, batasan-batasan sistem, kendala dan juga alternatif pemecahan masalah. Analisis digunakan untuk mengetahui perilaku sistem dan juga untuk mengetahui aktivitas apa saja yang ada dalam sistem tersebut.

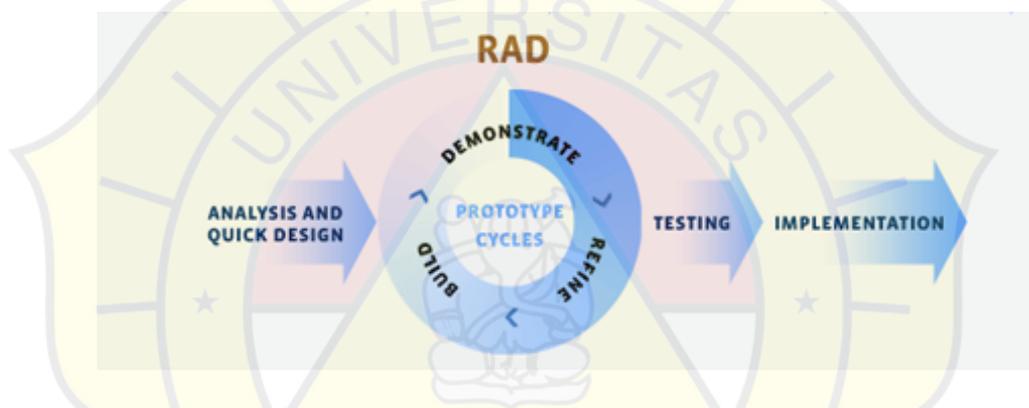
2. *Design Workshop*

Yaitu mengidentifikasi solusi alternatif dan memilih solusi terbaik. Kemudian membuat desain proses bisnis dan desain pemrograman untuk data-data yang telah didapat dan dimodelkan dalam arsitektur sistem

informasi. *Tools* yang digunakan dalam permodelan sistem biasanya menggunakan *Unified Modeling Language* (UML).

3. *Implementation*

Setelah selesai melalui tahap *Design Workshop*, selanjutnya sistem diimplementasikan (*coding*) ke dalam bentuk yang dimengerti oleh mesin yang diwujudkan dalam bentuk program atau unit program. Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan.



Gambar 2. 3 Langkah Penerapan Metode RAD

Metode pengembangan sistem RAD ini dirasa sangat cocok digunakan dalam rencana pelaksanaan rancang bangun sistem informasi *monitoring project* ini, karena proses proyek yang ada tetap berjalan dan bersamaan dengan itu dilakukan analisa dan pengembangan sistem informasi *monitoring project* mengenai kebutuhan, persyaratan pengguna yang menyesuaikan kebutuhan yang ada. Tahap implemtasi melalui metode RAD ini juga dirasa sesuai karena dapat mendeteksi kesalahan pengkodean atau *error* dapat dideteksi lebih awal karena dilakukan bersamaan dengan implementasinya.

2.2. Peralatan Pendukung Sistem (*Tools System*)

Menurut Nugroho (2013:15) Peralatan pendukung merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Adapun peralatan pendukung (*tools system*) yang dijelaskan sebagai model sistem yang akan dirancang meliputi *Use Case Diagram*, Skenario, *Activity Diagram* dan peralatan lainnya.

2.2.1 UML (*Unified Modelling Language*)

Menurut Nugroho (2010:6), UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek). Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

Menurut (Sukamto dan Shalahuddin, 2018:133) mengemukakan bahwa, “UML (*Unifield Modelling Language*) merupakan salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek”

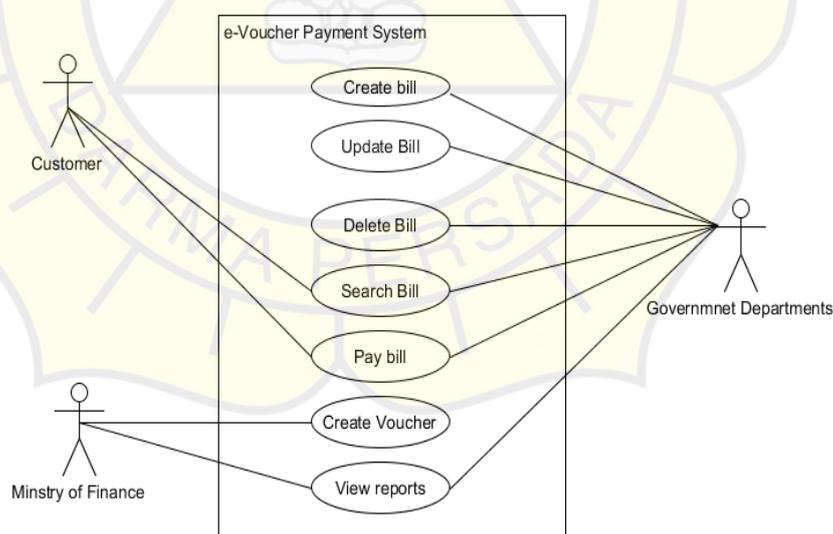
Berdasarkan pendapat tersebut dapat ditarik kesimpulan bahwa UML adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek (*Object Oriented programming*).

2.2.2 Model Diagram UML

A. Use Case Diagram

Menurut Nugroho (2010:35-36), kegunaan diagram *use case* sesungguhnya merupakan unit koheren dari fungsionalitas sistem/ perangkat lunak yang tampak dari luar dan diekspresikan sebagai urutan pesan-pesan yang dipertukarkan unit-unit sistem dengan satu atau lebih *actor* yang ada di luar sistem.

Kegunaan diagram *use case* juga adalah untuk mendefinisikan suatu bagian perilaku sistem yang bersifat koheren tanpa perlu menyingkap struktur internal sistem/ perangkat lunak yang digunakan. Definisi *use case* didalamnya mencakup semua perilaku yang ada dalam sistem yang sedang kita kembangkan: urutan utama, berbagai variasi dari perilaku normal (*exception*), serta tanggapan yang dikehendaki.



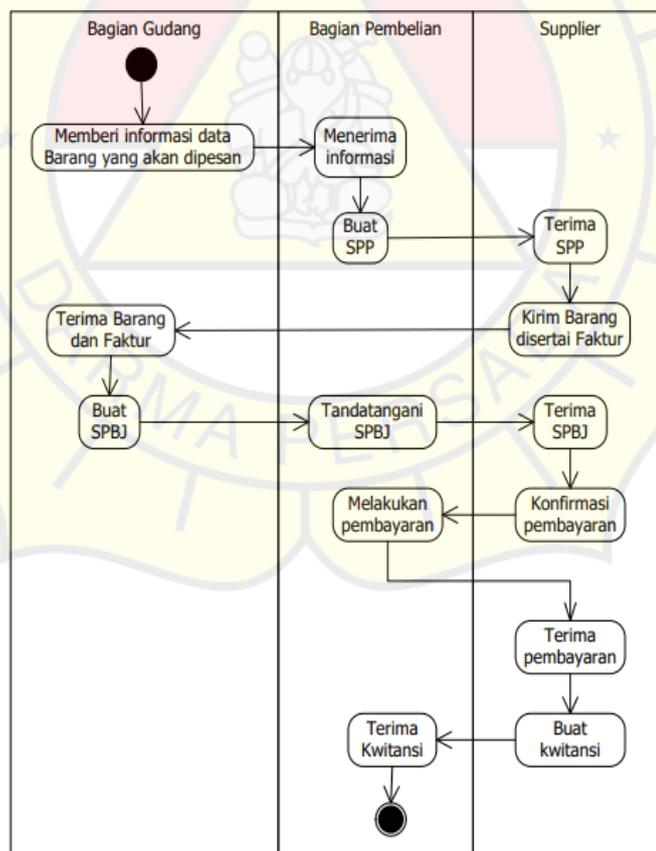
Gambar 2. 4 Contoh Use Case Diagram

B. Skenario

Menurut Yasin (2012:238), skenario adalah sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Form skenario merupakan penjelasan penulisan *use case* dari sudut pandang *actor*.

C. Activity Diagram

Menurut Nugroho (2010:62), Diagram Aktivitas (*Activity Diagram*) sesungguhnya merupakan bentuk khusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan.



Gambar 2. 5 Contoh *Activity Diagram*