BAB II
TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2) mengemukakan bahwa, "Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan manusia. Meskipun komputer berasal dari kata "komputasi", komputasi yang memang di laksanakanya mungkin tidak terlihat secara eksplisit. Ketika orang menggunakan komputer untuk membuat dokumen, berbagai perhitungan yang di lakukakan tidak terlihat. Sebagai contoh, ketika pemakai memilih pengaturan "*justify*" atau rata di batas kiri dan batas kanan dokumen, perhitungan untuk menambahkan karakter-karakter spasi sebenarnya terjadi."

Wahyudin, S.T., M.T (2018:1), "Komputer adalah suatu alat elektronik yang mampu melakukan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan instruksi yang diberikan, menyimpan perintah-perintah dan hasil pengolahnya, serta menyediakan *input* dalam bentuk informasi".

Berdasarkan definisi diatas, dapat disimpulkan bahwa komputer merupakan alat elektronik yang dapat menerima, memproses, mencetak, serta menyimpan data yang diolah yang beroperasi dibawah perintah manusia.

2.1.2 Pengertian Perangkat Lunak

Menurut Kadir (2017: 2), "Perangkat lunak adalah intruksi-intruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi seperti *Microcsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak.

Rosa dan Shalahuddin (2018: 2), perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).

2.1.3 Pengertian Internet

Albert K dan Natasya Wiajaya (2018:156), "Internet merupakan bagian dari bidang teknologi informasi (TI) yang semakin canggih dari masa ke masa. Keuntungan dan kemudahan dalam menggunakan internet dapat membantu manusia dalam mengerjakan berbagai macam urusan".

Natalia S dan Cornelia Amedea (2018:4), "*Interconnection Networking* (Internet) merupakan sebuah jaringan komputer global yang saling menghubungkan satu sama lain di seluruh dunia dengan menggunakan *standart internet protocol suite*".

2.1.4 Metode *Simple Additive Weighting* (SAW)

Simple Additive Weighting (SAW) adalah metode penjumlahan bobot dari kinerja setiap objek-objek yang berbeda dan memiliki kesempatan yang sama pada semua kriteria yang dimiliki. Metode *Simple Additive Weighting* (SAW) memerlukan proses normalisasi matrik keputusan (X) ke suatu skala yang dapat dibandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{\max_i x_{ij}} & \text{jika j adalah atribut keuntungan benefit} \\ \dfrac{\min_i x_{ij}}{x_{ij}} & \text{jika j adalah atribut biaya (cost)} \end{cases}$$

Keterangan:

rij = nilai rating kinerja ternormalisasi

xij = nilai atribut yang dimiliki dari setiap kriteria

Maxi xij = nilai terbesar dari setiap kriteria i Mini xij = nilai tekecil dari setiap kriteria i Benefit = jika nilai terbesar adalah terbaik *Cost* = jika nilai terkecil adalah terbaik Dimana rij adalah rating kinerja ternormalisasi dari alternative

Ai dan atribut Cj; i = 1, 2, …, m dan j = 1, 2,…, n.

Nilai preferensi untuk setiap alternative (Vi) diberikan sebagai :

$$V_i = \sum_{j=1}^{n} W_j \; r_{ij}$$

Keterangan:

vi = ranking untuk setiap alternatif wj = nilai bobot dari setiap kriteri

rij = nilai rating kinerja ternormalisasi

nilai vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

Langkah penyelesaian Fuzzy MADM menggunakan metode SAW :

1. Menentukan kriteria-kriteria yang dijadikan acuan pengambilan keputusan, yaitu ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matrikskeputusan berdasarkan kriteria (ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh *matriks* ternormalisasi R.
4. Hasil akhir diperoleh dari proses perangkingan yaitu penjumlahan dari perkalian matrik ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

Kelebihan dari metode *Simple Additive Weighting* (SAW):

1. Menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perangkingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif.
2. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan.
3. Adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai benefit dan cost).

Kekurangan dari metode *Simple Additive Weighting* (SAW):

Digunakan pada pembobotan lokal. Perhitungan dilakukan dengan menggunakan bilangan *crisp* maupun fuzzy.

1. Untuk membantu para pelajar atau mahasiswa supaya dapat mencari ilmu yang sesuai dengan bidang yang hendak dikuasai, yang paling utama bagi yang memiliki masalah dalam pembiayaan.
2. Membuat pemerataan suatu ilmu pengetahuan atau pendidikan terhadap masing-masing orang yang memerlukan.
3. Membuat generasi baru yang lebih cerdas dan pintar. Karena dengan adanya bantuan beasiswa ini maka seseorang terutama kaum muda dapat memiliki kesempatan untuk memperoleh pendidikan pada jenjang yang lebih tinggi.

4. Meningkatkan kesejahteraan. Sesudah terciptanya sumber daya manusia baru yang pintar, diharapkan mereka ini dapat saling memberi dengan bantuan ide dan ilmu pengetahuan yang sudah didapatkan ketika menjalani masa pendidikan.

2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Juansyah (2015:2) "aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanankan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju".

Deslianti dan Imam Muttaqin (2016:27) "Aplikasi adalah suatu subkelas perangkat lunak *computer* yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media"

Berdasarkan pendapat diatas dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang digunakan untuk melakukan suatu tugas atau pekerjaan berupa penerapan, penggunaan, pengolahan, penambahan data dan pemutar media.

2.2.2 Pengertian Beasiswa

Menurut Dian, P.S. (2018:212) "Beasiswa adalah bentuk penghargaan yang diberikan kepada individu agar dapat melanjutkan pendidikan ke jenjang yang lebih tinggi. Penghargaan tersebut bisa berbentuk akses tertentu pada suatu instansi atau penghargaan berupa bantuan keuangan".

Menurut Sulistyo dan Sri Winiarti (2015:965) "Beasiswa merupakan pemberian berupa bantuan keuangan maupun pendidikan yang diberikan perorangan, mahasiswa atau pelajar yang digunakan demi keberlangsungan pendidikan yang ditempuh bagi yang memiliki prestasi dibidang akademik, non akademik dan kemampuan ekonominya lemah, serta telah memenuhi syarat-syarat yang ditentukan pihak pemberi beasiswa."

2.2.3 Pengertian CSR (*Corporate Social Responsibility)*

Resnawaty, dkk. (2018:252) *"Corporate Social Responsibility* (CSR) merupakan sebagai komitmen industri untuk mempertanggung-jawabkan dampak operasi dalam dimensi sosial, ekonomi, dan lingkungan serta menjaga agar dampak tersebut menyumbang manfaat kepada masyarakat dan lingkunganya."

Menurut Parengkuan(2017:566) *"Corporate Social Responsibility* (CSR) atau pertanggung-jawaban sosial perusahaan adalah mekanisme bagi suatu organisasi untuk secara sukarela mengintegrasikan perhatian terhadap lingkungan dan sosial ke dalam operasinya dan interaksinya dengan *stakeholders*, yang melebihi tanggung jawab organisasi di bidang hukum."

2.2.4 Pengertian *Website*

Menurut Abdulloh (2015:1) "Website dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet."

Sedangkan menurut Abdullah (2015:42) dalam jurnalnya menyebutkan :

"Terminologi *website* adalah kumpulan dari halaman- halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada didalam *World Wide Web* (WWW) di Internet. Sebuah *web page* adalah dokumen yang ditulis dalam format HTML *(Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*."

Dapat disimpulkan, *website* adalah kumpulan dari beberapa halaman yang berisi informasi terangkum dalam domain dan dapat di akses melalui koneksi internet.

2.2.5 Pengertian Aplikasi Pengolahan Data BEPASIBA pada CSR (*Corporate Social Responsibility*) pada PT Bukit Asam Tbk

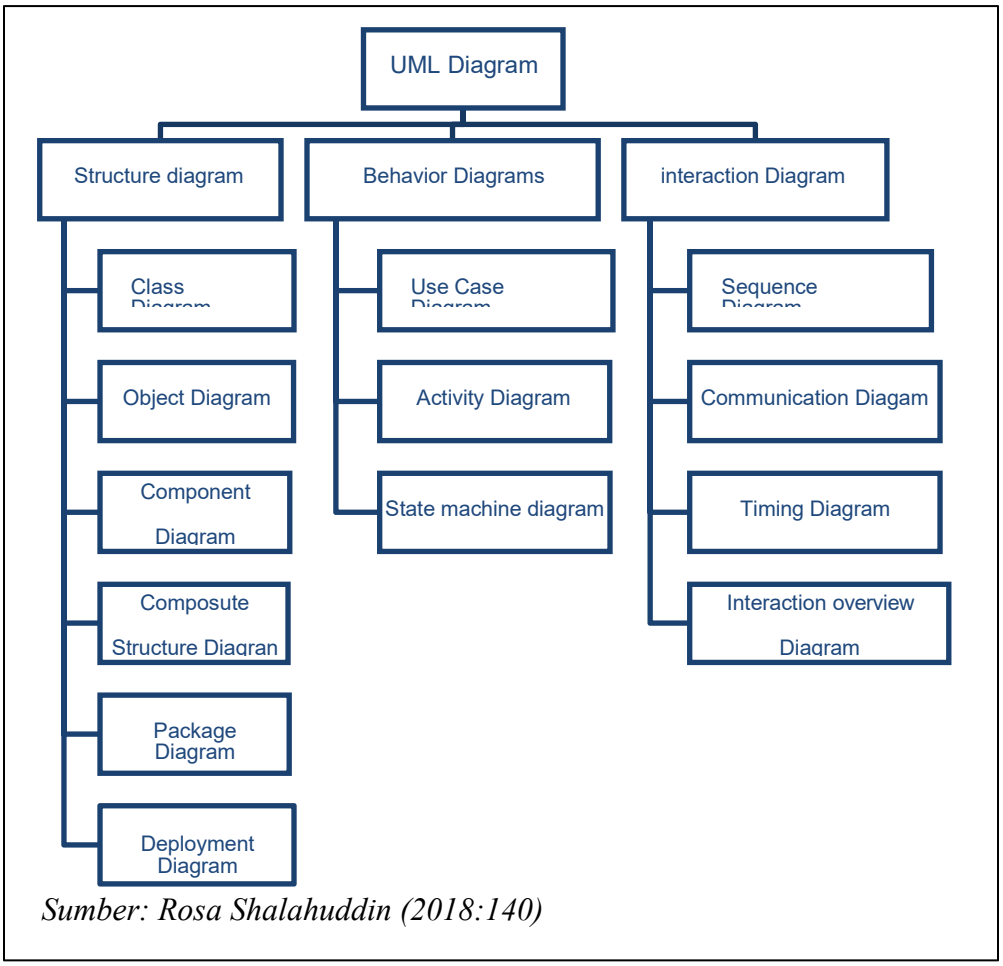
Berdasarkan pengertian yang telah dijelaskan diatas dapat disimpulkan bahwa aplikasi pengolahan data bepasiba pada CSR (*Corporate Social Responsibility*) merupakan suatu perangkat lunak yang dapat membantu dalam proses pendaftaran peserta beasiswa serta menghitung nilai seleksi peserta beasiswa yang memiliki prestasi akademik di suatu perguruan tinggi negeri / swasta dan menyandang status anak pegawai atau pensiunan bukit asam.

2.3 Teori Khusus

2.3.1 Pengertian *Unified Modeling Language* (UML)

Sukamto dan Shalahuddin (2018:137), menjelaskan *Unified Modeling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung."

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam – macam diagram tersebut yaitu :



Sumber: Rosa Shalahuddin (2018:140)

Gambar 2.1 Diagram UML

Berikut merupakan penjelasan singkat dari pembagian katergori tersebut.

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interactions diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun sistem interaksi antar subsistem pada suatu sistem.

2.3.2 Jenis-Jenis Diagram UML

2.3.2.1 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* , sebagai berikut :

"Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron."

Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

Tabel 2.1 Simbol-simbol *Class* Diagram

No	Simbol	Deskripsi
1.	Kelas **Nama_kelas** + atribut + operasi ()	Kelas pada struktur sistem
2	antarmuka / interface nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	asosiasi / association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai *multiplicity*
4.	asosiasi berarah / *directed assosiation*	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
5.	generalisasi	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / *dependency*	Relasi antarkelas dengan makna kebergantungan antar kelas

No	Simbol	Deskripsi
7.	agregasi / *aggregation*	Relasi antarkelas dengan makna semua-bagian (*whole-part*)

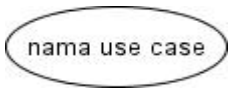
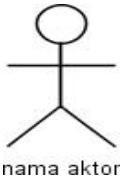
Sumber: Sukamto dan Shalahuddin (2018:146)

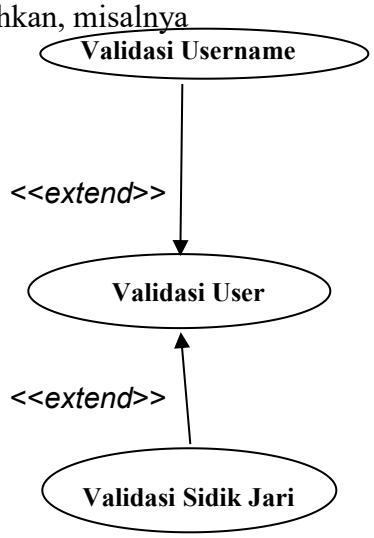
2.3.2.2 Pengertian *Use case* Diagram

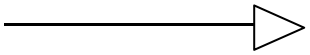
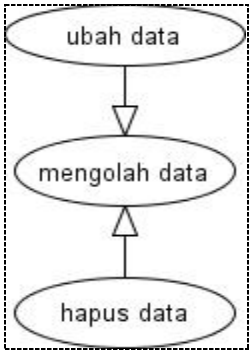
Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut :

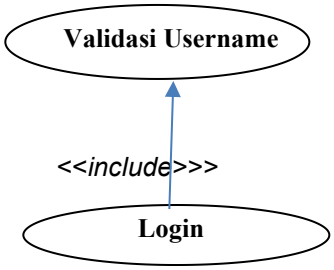
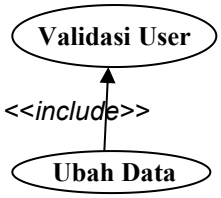
"Use case merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem." Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.2. Simbol-simbol *Use case* Diagram

No	Simbol	Deskripsi
1.	*Use case* 	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama *use case.*
2.	aktor / *actor* 	orang, proses, atau siatem lain yang berinteraksi dengan sistem informasi yang akan 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 actor.
3.	asosiasi / *association*	komunikasi antar aktor dan *use case* yang berpartisioasi pada *use case* atau *use case* memiliki ingeraksi dengan actor.

No	Simbol	Deskripsi
4.	ekstensi / *extend* <<extend>> ⟶	relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang di tambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misalnya  arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya

No	Simbol	Deskripsi
5.	Generalisasi / *Generalization* 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,  misalnya: arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).

No	Simbol	Deskripsi
6.	menggunakan / include / uses <<*include*>> <<*uses*>>	relasi tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini ada dua sudut pandang yang cukup besar mengenai include di *use case*: • *Include* berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan, misalnya pada kasus berikut:  • *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang di tambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:  kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber: Sukamto dan Shalahuddin (2018:156)

Ada dua hal utama pada *use case* yaitu:

1. Aktor merupakan 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 aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit- unit yang saling bertukar pesan antar unit atau aktor.

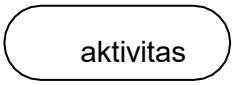
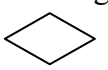
2.3.2.3 Pengertian *Activity Diagram*

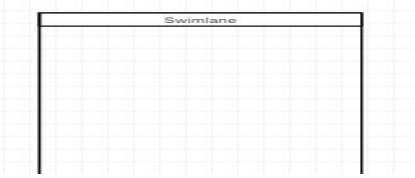
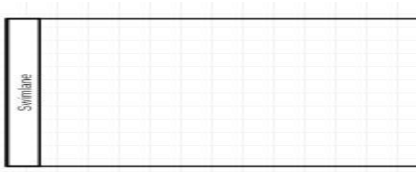
Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

"*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."

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.3 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.	Percabangan / *decision* 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / *join*	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

No	Simbol	Deskripsi
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Sukamto dan Shalahuddin (2018:162)

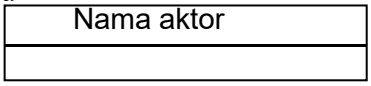
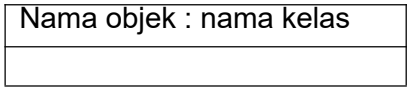
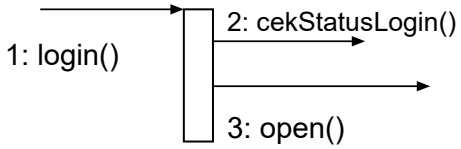
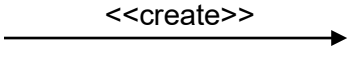
2.3.2.4 Pengertian *Sequence Diagram*

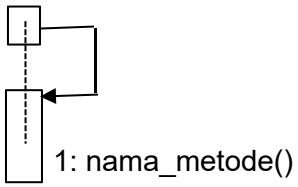
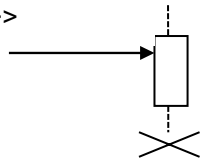
"Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki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

Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan- pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada Sequence Diagram :

Tabel 2.4 Simbol-simbol pada *Sequence* Diagram

No	Simbol	Deskripsi
1.	Actor atau  tanpa waktu aktif	orang, proses, atau sistem lain yang berinteraksi dengan 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 actor
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
5.	Pesan tipe create  	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

No	Simbol	Deskripsi
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 dikirimi
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:Sukamto dan Shalahuddin (2018:165-167)

2.4 Teori Program

2.4.1 Pengertian PHP

Gambar 2.2 Logo PHP

Sidik (2016:4), ”PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman *script –script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di *server web*, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor *teks* atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*”.

Abdulloh (2015:3) “PHP (*Hypertext Processor*) merupakan *server-hide programming*, yaitu bahasa pemrograman yang di proses di sisi server”. Pendapat lain oleh Abdullah (2015:42) “PHP adalah bahasa *scripting* yang menampilkan output HTML ataupun output lain sesuai dengan keinginan pemrogram (misalnya: PDF dan lain-lain) yang dijalankan pada *server side*. Artinya semua sintak yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirim pada browser hanya menghasilkan output saja.”

Dapat disimpulkan, PHP merupakan bahasa pemrograman yang dijalankan dengan *server hide programming* yang menghasilkan output pada browser.

2.4.2 Pengertian HTML

Menurut Abdulloh (2015:2) “HTML singkatan dari *Hyper Text Markup Language* yaitu script yang berupa tag – tag untuk membuat dan mengatur struktur website.” Menurut Harison dan Ahmad Syarif (2016:43) “*HyperText Markup Language* (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan formating hypertext

sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi."

Berdasarkan pendapat diatas, (*HyperText Markup Language*) HTML adalah bahasa markup berupa tag – tag yang digunakan untuk membuat halaman web dan menampilkan informasi didalam sebuah web internet.

2.4.3 Pengertian Basis Data .

Menurut Abdullah (2015:42) "Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol)."

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan secara cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redudancy)* yang tidak perlu, untuk memenuhi kebutuhan.
c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Database adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data. Sedangkan Menurut Rahmad dan Dedi Setiadi (2014:1333) "Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukan dengan kunci dari tiap file yang ada."

Dapat disimpulkan bahawa basis data atau database adalah sekumpulan data yang berisi deskripsi dari data yang dinyatakan dalam bentuk angka, karakter dan symbol yang dibutuhkan dalam sebuah organisasi.

2.4.4 Pengertian MySql

Gambar 2.3 Logo MySQL

Menurut Abdullah (2015:42) "MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user. MySQL sangat popular pengembang web (*web developers*) karena memiliki kecepatan dan ukuran yang kecil membuat MySQL lebih ideal untuk website ditambah lagi dengan fakta bahwa MySQL adalah *open source* yang berarti gratis." Dalam pengertian lain oleh Yuliansyah (2014:826) "MySQL adalah sebuah database manajemen system (DBMS) popular yang memiliki fungsi sebagai *relational database manajemen system* (RDBMS)."

Jadi, dapat disimpulkan, MySQL adalah sebuah perangkat lunak basis data yang memiliki fungsi sebagai *relational database manajemen system* RDBMS yang multithread dan multi-user.

2.4.5 Pengertian XAMPP

Gambar 2.4 Logo XAMPP

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL

secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi. (Lestanti dan Ardina, 2016:69)

Sedangkan menurut Poipessy dan Mirna Umasangadji (2018:40) "XAMPP merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP." Dapat disimpulkan XAMPP merupakan paket konfigurasi Apache, PHP dan MySQL yang digunakan untuk membantu pengembangan aplikasi.

2.4.6 Pengertian *Sublime Text*

Menurut Syifani dan Ardiansyah (2018:25) "*Sublime Text* adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform *operating system* dengan menggunakan teknologi *Phyton* API". Sedangkan menurut Putra (2016:181) "*Sublime text* adalah teks editor berbasis *Python*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudah dan simpel yang cukup terkenal dikalangan *developer* (pengembang) dan *desainer".*

Dapat disimpulkan bahwa, *sublime text* merupakan aplikasi editor kode dan teks yang kaya fitur, *cross platform*, mudah dan *simple* yang berbasis Phyton.