



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”. Adapun pengertian Komputer Menurut Sanders (Dikutip Susanto, 2009:2) “Komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan supaya secara otomatis menerima dan menyimpan data input, memprosesnya dan menghasilkan *output* dibawah pengawasan suatu langkah-langkah, instruksi-instruksi program yang tersimpan di memori (*stored program*)”.

Jadi dapat disimpulkan bahwa komputer adalah sekelompok alat elektronik yang terdiri atas perintah *input*, alat yang mengolah *input*, dan peralatan *output* yang memberikan informasi serta bekerja secara otomatis.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Sutabri (2014:62), “Perangkat Keras komputer tidak berfungsi tanpa perangkat lunak. Teknologi yang canggih dari perangkat keras akan berfungsi bila intruksi-intruksi tertentu telah diberikan kepadanya. Intruksi-intruksi tersebut disebut dengan perangkat lunak. Perangkat Lunak dapat diklasifikasikan kedalam dua bagian besar, yaitu:

1. Perangkat Lunak Sistem (*system software*), yaitu perangkat lunak yang mengoperasikan sistem komputer. Perangkat lunak sistem dapat dikelompokkan menjadi 3 bagian, yaitu:
 - a. Perangkat lunak sistem operasi (*operating system*), yaitu program yang ditulis untuk mengendalikan dan mengoordinasikan operasi dari sistem komputer.
 - b. Perangkat lunak sistem bantuan (*utility*), yaitu program yang ditulis untuk bantuan yang berhubungan dengan sistem komputer, misalnya memformat disc, menyalin disc, mencegah dan membersihkan virus.



- c. Perangkat lunak bahasa (*language software*), yaitu program yang digunakan untuk menerjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman kedalam bahasa mesin supaya dapat dimengerti oleh komputer.
2. Perangkat lunak aplikasi (*aplication software*), yaitu Program yang ditulis dan diterjemahkan oleh perangkat lunak bahasa untuk menyelesaikan suatu aplikasi tertentu”. Begitupula menurut Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa perangkat lunak dapat diklasifikasi menjadi 2 bagian yaitu perangkat lunak sistem yaitu perangkat lunak yang mengoperasikan sistem komputer dan perangkat lunak aplikasi yaitu Program yang ditulis dan diterjemahkan oleh perangkat lunak bahasa untuk menyelesaikan suatu aplikasi tertentu, dan juga perangkat lunak digunakan untuk melaksanakan tugas sesuai kehendak pemakai.

2.1.3 Pengertian Perangkat Keras (*Hardware*)

Sutabri (2014:32), “Perangkat keras sebagai subsistem dari sistem komputer juga mempunyai komponen, yaitu

1. Komponen peranti masukan (*input device*) yaitu alat yang digunakan untuk menerima masukan data atau masukan program misalnya *keyboard*, *scanner*, sensor, dan pengenalan suara.
2. Komponen peranti pemroses (*processing device*), yaitu alat di mana instruksi-instruksi program dieksekusi untuk memproses data yang dimasukkan lewat alat masukan yang hasilnya nanti akan ditampilkan dialat output misalnya *CPU (central processing unit)*.
3. Komponen peranti keluaran (*output device*), yaitu alat untuk menampilkan pengolahan data seperti tulisan (huruf, kata, angka, karakter, dan simbol-simbol), *image* (bentuk grafik atau gambar) dan suara (bentuk musik) misalnya *hardcopy device (printer)* dan *softcopy device (monitor)*.
4. Komponen peranti penyimpanan (*storage*) eksternal adalah penyimpanan luar, karena terletak diluar alat prosesnya atau disebut *mass storage* atau

penyimpanan masal karena penyimpanan umumnya lebih besar dari *main memory*”. Lain halnya menurut Kadir (2017:2), “Perangkat keras adalah peranti-peranti yang terkait dengan komputer dan terlihat secara fisik. Monitor, *hard disk*, dan *mouse* adalah contoh perangkat keras”.

Dari beberapa definisi perangkat keras penulis menyimpulkan bahwa perangkat keras adalah beberapa komponen yang bekerja saling mendukung sesuai dengan instruksi *software*.

2.1.4 Metode Pengembangan Sistem

Sukamto dan Shalahuddin (dalam Fitria dan Widiowati, 2017:28), RUP (*Rational Unified Process*) merupakan pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*welldefined*) dan penstrukturan yang baik (*wellstructured*)”.

Proses pengulangan/iteratif pada RUP secara global dapat dilihat pada gambar berikut:



(Sumber: Sukamto dan Shalahuddin (dalam Fitria dan Widiowati, 2017:28))

Gambar 2.1 Proses iteratif RUP (*Rational Unified Process*)



2.1.4.1 Fase *Rational Unified Process*(RUP)

Sukamto dan Shalahuddin (dalam Fitria dan Widiowati, 2017:28), “RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. Metode RUP mempunyai empat fase”.

a. *Fase Inception*

Tahap dimana kita memodelkan proses bisnis yang dibutuhkan (*businessmodeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Hasil dari fase ini berupa Analisa pemodelan proses bisnis *as-is* dan *to-be*, analisis persyaratan, dan pemodelan *usecase*.

b. *Fase Elaboration*

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini lebih pada analisis dan desain sistem.

c. *Fase Construction*

Tahap dimana kita mengembangkan komponen dan fitur-fitur sistem. Implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Hasil dari fase ini yaitu implementasi sistem informasi monitoring berbasis website dan hasil pengujian sistem yaitu dengan blackbox validation testing dan compatibility testing.

d. *Fase Transition*

Tahap dimana kita deployment atau Instalasi sistem agar dapat dimengerti oleh user. Aktivitas pada tahap ini termasuk pada pelatihan user dan pemeliharaan.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap, keempat fase pada RUP dijalankan secara urut dan berulang, dengan setiap iterasi digunakan untuk memperbaiki iterasi berikutnya.

2.2 Teori Khusus

2.2.1 Pengertian UML (*Unifed Modeling Language*)

Windu dan Grace (dalam Suendri, 2018:2) “*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”.



2.2.2 Klasifikasi Diagram *Unified Modeling Language (UML)*

Sukanto dan Salahuddin (2016:140), “*UML (Unified Modelling Language)* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. *Behavior Diagram*

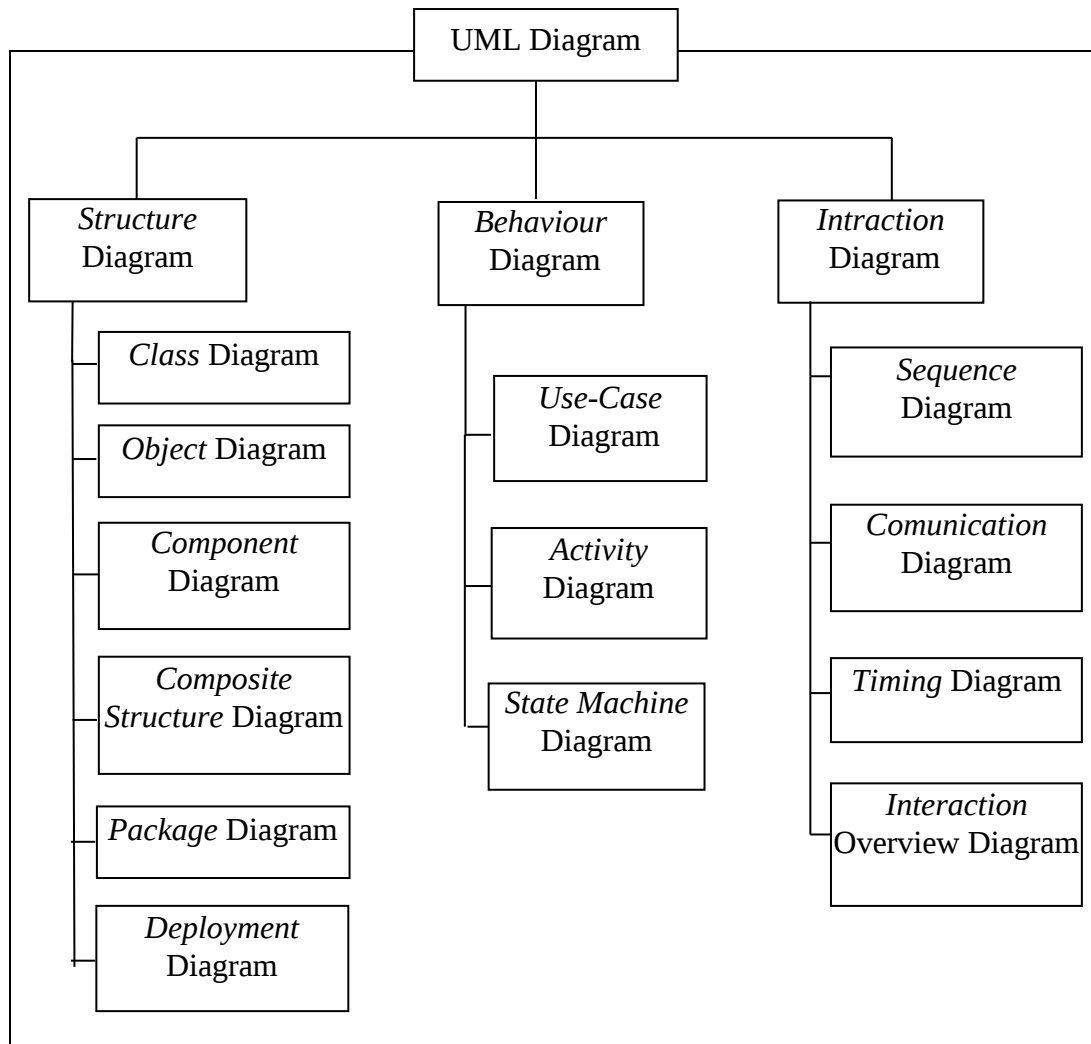
Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. *Interaction Diagram*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar yang terklasifikasi seperti dibawah ini:

Struktur Klasifikasi Diagram UML (Unified Modelling Language)



(Sumber: Sukamto dan Shalahuddin (2016:140))

Gambar 2.1. Klasifikasi Diagram UML (*Unified Modelling Language*)

2.2.3 Pengertian *Use Case Diagram*

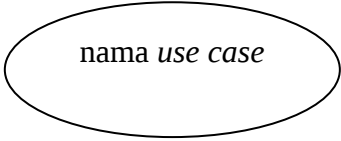
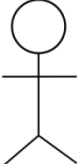
Sukamto dan Shalahuddin (2016:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



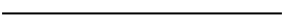
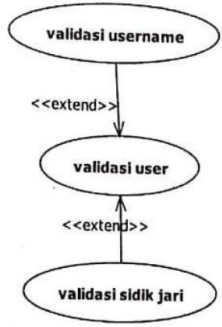
Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

- a. Aktor merupakan orang, proses, atau sistem 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.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

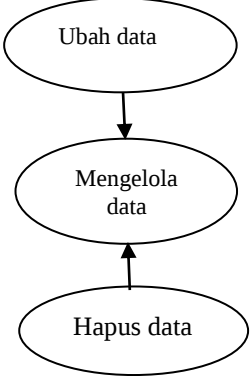
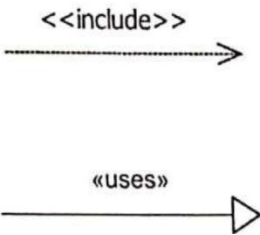
Tabel 2.1 Simbol-simbol dalam *Use case*

No.	Simbol	Deskripsi
1.		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal <i>frase</i> nama <i>use case</i> .
2.		Orang, proses, atau sistem lain yang berinteraksi 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 <i>frase</i> nama actor
3.		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Lanjutan Tabel 2.1 Simbol-simbol dalam *Use case*

No.	Simbol	Deskripsi
3.	Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.	Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah usecase dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; ditambahkan, misal ;  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.
5.	Generalisasi/ <i>generalization</i> 	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:

Lanjutan Lanjutan Tabel 2.1 Simbol-simbol dalam *Use case*

No.	Simbol	Deskripsi
		 Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
6.	Menggunakan/ <i>include/uses</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana use-case yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i>

(Sumber: Sukanto dan Shalahudin, 2016:156-158)

2.2.4 Pengertian *Class Diagram*

Sukanto dan Shalahuddin (2016:141), “Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.



Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Sukanto dan Shalahuddin (2013:146) :

Tabel 2.2 Simbol-simbol dalam *Class Diagram*

No.	Simbol	Deskripsi
1.	Kelas Nama_kelas +atribut +operasi()	Kelas pada struktur system
2.	Antarmuka/<i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/<i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah (<i>directed Association</i>)	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan Tabel 2.2 Simbol-simbol dalam *Class Diagram*

No.	Simbol	Deskripsi
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Kebergantungan/ <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas
7.	Agregasi/ <i>aggregation</i> 	Relasi antarkelass dengan makna semua-bagian (<i>whole-part</i>)

(Sumber: Sukamto dan Shalahudin, 2016:146-147)

2.2.5 Pengertian *Activity Diagram*

Sukamto dan Shalahuddin (2016:161), *Activity Diagram* atau Diagram Aktivitas 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.

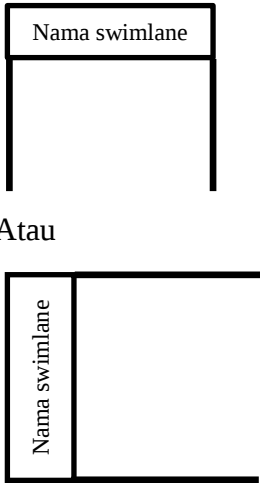
Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan
3. Rancangan pengujian dimana seriap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak

Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

Lanjutan Tabel 2.3 Simbol-simbol dalam Activity Diagram

No.	Simbol	Deskripsi
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan 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.

(Sumber: Sukamto dan Shalahudin, 2016:162-163)

2.2.6 Pengertian *Sequence Diagram*

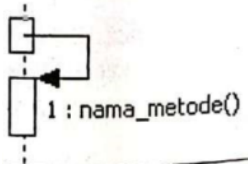
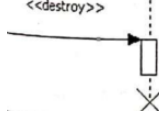
Menurut Sukamto dan Shalahuddin (2016:165), Diagram sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karna itu, untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat sekenario yang ada pada *use case*.

Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

No.	Simbol	Deskripsi
1.	Aktor  Nama aktor Atau Nama aktor tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal <i>frase</i> nama <i>actor</i> .
2.	Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
3.	Objek <u>Nama objek</u> :nama kelas	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.
5.	Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada arah objek yang dibuat

Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

No.	Simbol	Deskripsi
6.	Pesan tipe <i>call</i> <pre>1 :nama_metode{</pre> 	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 <i>send</i> <pre>1 :masukan</pre> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> <pre>1 :keluaran</pre> 	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 pada objek yang menerima kembalian.
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

(Sumber: Sukanto dan Shalahudin, 2016:165-167)



2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Juansyah (2015:2), dalam kamus Komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Koyuko dkk (dalam Oki 2016), Aplikasi adalah suatu *software* yang dapat melakukan beberapa tugas tertentu seperti membuat dokumen, penambahan data dan beberapa tugas lainnya. *suite* aplikasi (*application suite*) adalah penggabungan dari berbagai macam aplikasi menjadi satu. salah satu contohnya *Microsoft Office* dan *Open Office* yaitu menggabungkan aplikasi *Microsoft word*, *Microsoft excel*, dan beberapa aplikasi lainnya. Pada umumnya aplikasi-aplikasi yang berada dalam suatu paket mempunyai *interface* yang sama sehingga memudahkan *user* untuk mempelajari dan memakai aplikasi tersebut dan juga aplikasi- aplikasi tersebut mempunyai bisa berinteraksi antar aplikasi sehingga memudahkan *user*. Contohnya, suatu lembar kerja dapat disimpan dalam suatu dokumen pengolah kata meskipun dibuat pada aplikasi lembar kerja yang terpisah.

Dari definisi diatas penulis menyimpulkan bahwa aplikasi adalah perangkat lunak atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu.

2.3.2 Pengertian Penjadwalan

Hantoro (dalam Sagala 2018:88), Mendefiniskan penjadwalan sebagai pengurutan proses produkai yang sistematis, sehingga urutan proses dapat berjalan dengan lancar dengan memanfaatkan semua fasilitas yang ada dalam perusahaan.

2.3.3 Pengertian Mahasiswa

Mahasiswa adalah seseorang yang sedang dalam proses menimba ilmu ataupun belajar dan terdaftar sedang menjalani pendidikan pada salah satu bentuk perguruan tinggi yang terdiri dari akademik, politeknik, sekolah tinggi, institut dan universitas (Hartaji Dalam Pujiono 2015:22).



2.3.4 Pengertian Magang

Sumardiono (2014:116), “Magang adalah proses belajar dari seorang ahli melalui kegiatan dunia nyata. Selain itu, magang adalah proses mempraktikkan pengetahuan dan keterampilan untuk menyelesaikan problem nyata di sekitar”.

Darmawan (2017:153) “Magang merupakan suatu proses pembelajaran yang mengandung unsur “belajar sambil bekerja” (learning by doing), dimana warga belajar (pemagang) akan membiasakan diri untuk mengikuti proses pekerjaan yang sudah biasa dilakukan oleh sumber belajar, fasilitator (permagang)”.

Dari definisi di atas dapat disimpulkan bahwa magang adalah pelatihan atau praktek untuk menguasai keahlian tertentu dibawah bimbingan dan pengawasan instruktur yang berpengalaman.

2.3.5 Pengertian Laporan Akhir

Hamid dan Rachman (dalam Rusmawan 2019:1), Skripsi adalah istilah yang digunakan di Indonesia untuk mengilustrasikan suatu karya tulis ilmiah berupa paparan tulisan hasil penelitian sarjana (S1) yang membahas suatu permasalahan atau fenomena dalam bidang ilmu tertentu dengan menggunakan kaidah-kaidah yang berlaku. Sedangkan Laporan Akhir adalah Istilah seperti di penjabaran di atas yang ditujukan untuk mahasiswa D3.

2.3.6 Pengertian Aplikasi Penjadwalan Mahasiswa Magang dan Laporan Akhir pada Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya

Aplikasi Penjadwalan Mahasiswa Magang dan Laporan Akhir pada Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya adalah Suatu Aplikasi yang memiliki serangkaian proses dan komponen atau elemen berisi informasi tentang siapa saja dosen pembimbing dan mahasiswa bimbingan pada saat pelaksanaan Magang dan Laporan Akhir beserta informasi mengenai penjadwalan sidangnya. Aplikasi ini dibuat menggunakan bahasa pemrograman *PHP* dan *Database MySQL*.



2.4 Teori Program

2.4.1 Pengertian Basis Data

Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”. Adapun pengertian Basis Data Menurut Connolly & Begg (dikutip Rusmawan, 2019:38), “Basis data adalah sebuah kumpulan data yang saling berelasi secara logika dan dirancang untuk memenuhi informasi yang dibutuhkan oleh suatu organisasi”.

Dari beberapa pernyataan di atas, dapat disimpulkan bahwa basis data adalah kumpulan data yang berelasi untuk memenuhi informasi yang tersimpan di komputer secara sistematis supaya bisa di pantau dengan program komputer.

2.4.2 Pengertian MySQL

Nugroho (2014:31), “MySQL atau program aplikasi database, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi text dan juga angka”. Adapun pengertian MySQL menurut Rusmawan (2019:97), “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6.000.000 instalasi di seluruh dunia”.

Dari beberapa pernyataan di atas, dapat disimpulkan bahwa MySQL adalah aplikasi database yang digunakan untuk menyimpan data atau server database yang mendukung bahasa database pencarian SQL .

2.4.3 Pengertian CodeIgniter

Andrianto dan Nursikuwagus (2017:48), “CodeIgniter adalah sebuah *framework* PHP yang dapat mempercepat pengembangan atau pembuatan sebuah *website*. Dengan dilengkapi library yang banyak dan helper yang berguna didalamnya sehingga dapat mempermudah proses development. CodeIgniter bersifat *open source* dan menggunakan metode MVC (*Model, View, Controller*)”.



2.4.4 Pengertian Bahasa Pemrograman

Kadir (2018:2), “Bahasa pemrograman adalah bahasa yang digunakan untuk menyusun program. Seperti halnya tulisan orang, tulisan dapat dituangkan dengan menggunakan bahasa manusia seperti bahasa Indonesia, bahasa Inggris, atau bahasa Jepang. Nah, program pun dapat dituangkan dengan menggunakan berbagai bahasa pemrograman”.

Secara alamiah, bahasa yang dimengerti oleh komputer adalah bahasa komputer. Bahasa ini menggunakan kode biner, yang merupakan perpaduan angka 0 dan 1 saja.

2.4.4.1 Pengertian HTML

Abdulloh (2018:127), “HTML merupakan singkatan dari *Hypertext Markup Language* yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa tag-tag yang menyusun setiap elemen dari website”. Lain hal nya pengertian HTML menurut Fathansyah (2018:17), “HTML adalah bahasa pengkodean yang digunakan untuk membuat halaman *web* agar bisa ditampilkan melalui *web browser*”.

Dari beberapa pernyataan di atas, dapat disimpulkan bahwa HTML adalah bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman *web* yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah *web* internet (*browser*).

2.4.4.2 Pengertian PHP

Raharjo (2016:38), “PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web”. Sedangkan menurut Abdulloh (2018:127), “PHP merupakan kependekan dari *PHP Hypertext Preprocessor* yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server”.

Dari beberapa pernyataan di atas, dapat disimpulkan bahwa PHP adalah *skrip server-side* yang didesain untuk pengembangan *web*.



2.4.4.3 Pengertian Web Server

Abdulloh (2018:4), “Web Server merupakan perangkat lunak yang berfungsi untuk menerima permintaan (*request*) melalui protocol HTTP atau HTTPS dari client kemudian mengirimkan kembali dalam bentuk halaman-halaman web”. Adapula pengertian Web Server menurut Fathansyah (dalam Prayitno dan Safitri, 2015:2), Menerangkan bahwa pengertian web server adalah “*Server Web* (Web Server) merujuk pada perangkat keras (*server*) dan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP ataupun variannya (seperti FTP dan HTTPS) atas berkas-berkas yang terdapat pada suatu URL ke pemakai”.

2.4.4.4 Pengertian Xampp

Suntoro (2019:11), “XAMPP adalah perangkat lunak yang bersifat *open source*, aplikasi *Apache* (web server) yang mudah di instal dan berisi MariaDB, PHP, dan Perl”. Menurut Wahana (dalam Prayitno dan Safitri (2015:2), “XAMPP adalah salah satu paket instalasi apache, PHP, dan MySQL secara instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut”.

2.4.4.5 Pengertian PhpMyAdmin

Nugroho (2014:10), phpMyAdmin adalah aplikasi manajemen *database server* MySQL berbasis *web*. Dengan aplikasi phpMyAdmin kita bias mengelola *database* sebagai **root** (pemilik server) juga disebut *Administrator database*. Lewat fasilitas phpMyAdmin, Anda bias membuat *database* baru, *table* dan struktur data mengelola data di dalam *database*.