



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Menurut Irma (2016:1), “Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi dibawah kontrol program yang tersimpan dimemorinya.”.

Sedangkan Menurut Kadir (2017:2), “Komputer merupakan peralatan elektronik yang bermanfaat, melaksanakan berbagai pekerjaan yang dilakukan oleh manusia” Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari komputer adalah sebuah alat elektronik yang dapat memproses data digital dan informasi yang bermanfaat bagi *user* sehingga dapat membantu terhadap pekerjaan manusia dengan mudah dan cepat.

2.1.2. Pengertian Perangkat Lunak

Menurut Sukanto & Shalahuddin (2018:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Sedangkan menurut Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”. Berdasarkan pendapat diatas, dapat disimpulkan bahwa perangkat lunak merupakan cara menggunakan yang ditujukan kepada komputer sesuai kebutuhan pemakai.

2.1.3. Pengertian Data

Menurut Kristanto (2018:8) menyatakan bahwa, “Data merupakan bentuk yang belum dapat memberikan manfaat yang besar bagi penerimanya, sehingga



perlu suatu model yang nantinya akan dikelompokkan dan diproses untuk menghasilkan informasi”.

Selain itu menurut Rusdiana dan Irfan (2016:68), “Data adalah fakta-fakta mentah yang harus dikelola untuk menghasilkan informasi yang memiliki arti bagi suatu organisasi atau perusahaan. Berdasarkan pendapat di atas, dapat disimpulkan bahwa definisi dari data adalah suatu fakta-fakta yang mewakili suatu objek dan bermanfaat bagi suatu organisasi.

2.1.4. Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2018:125), “RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:

1. *Inception* (permulaan)

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

2. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

3. *Construction* (kontruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada



implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

4. *Transition* (transisi)

Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal. Aktivitas pada tahap ini termasuk pada pelatihan user, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan user.

2.2. Teori Khusus

2.2.1. *Unified Modeling Language* (UML)

Sukanto dan Shalahuddin (2018:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”

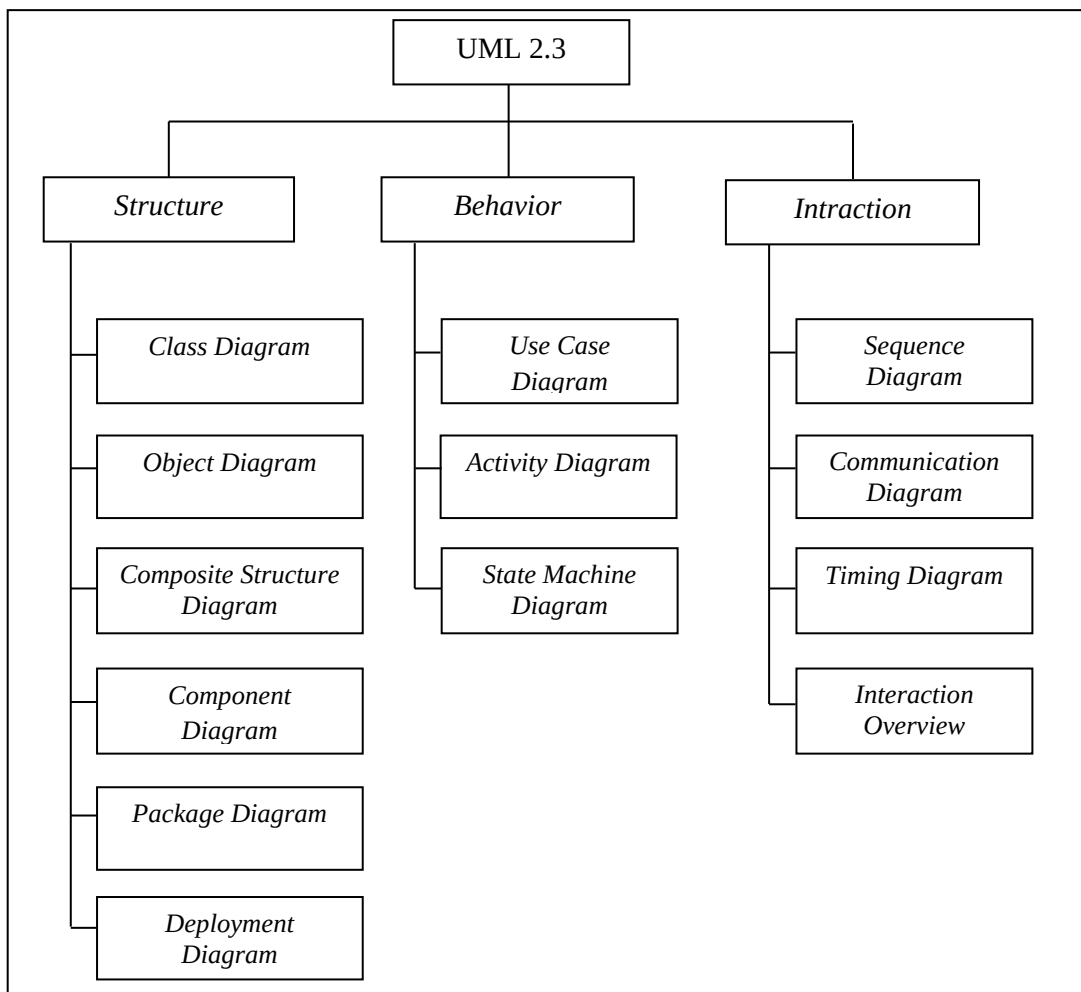
UML menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.



Gambar 2.1. Logo UML



Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



Gambar 2.2 Kategori dan Macam-macam Diagram UML

2.2.2. Jenis-jenis Diagram UML

2.2.2.1. Diagram Use Case

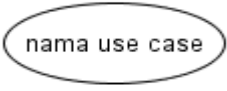
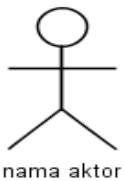
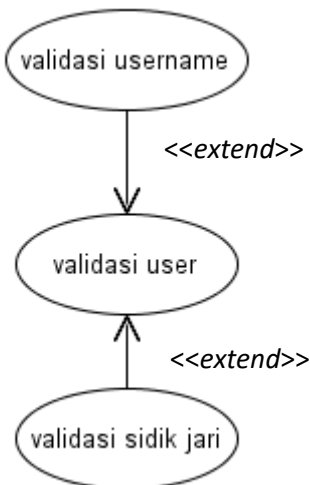
Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case diagram* sebagai berikut :

“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara

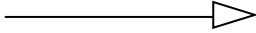
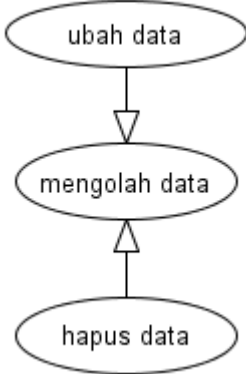
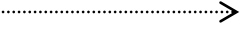
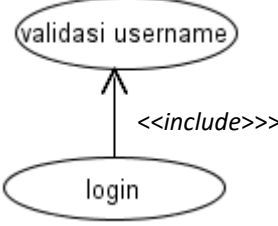


satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1. Simbol-simbol Diagram *Use Case*

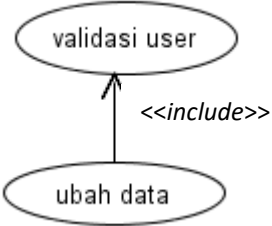
No	Simbol	Deskripsi
1.	<i>Use case</i> 	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i> .
2.	aktor / <i>actor</i> 	orang, proses, atau sistem lain yang berinteraksi 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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	asosiasi / <i>association</i> 	komunikasi antar 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 <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya 

Lanjutan Tabel 2.1. Simbol-simbol Diagram *Use Case*

No	Simbol	Deskripsi
		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: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	menggunakan / include / uses  <<include>>  <<uses>>	relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>: • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut: 



Lanjutan Tabel 2.1. Simbol-simbol Diagram Use Case

No	Simbol	Deskripsi
		• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre> graph BT A([ubah data]) -- "<<include>>" --> B([validasi user]) </pre> 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.

2.2.2.2. Diagram Class

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

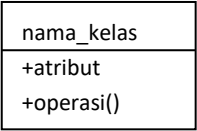
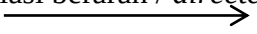
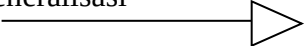
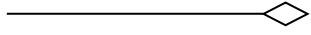
“*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat



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 2.2. Simbol-simbol Diagram *Class*

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <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>
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: Sukanto dan Shalahuddin (2018:146)

2.2.2.3. Diagram Activity

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

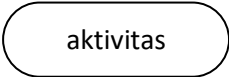
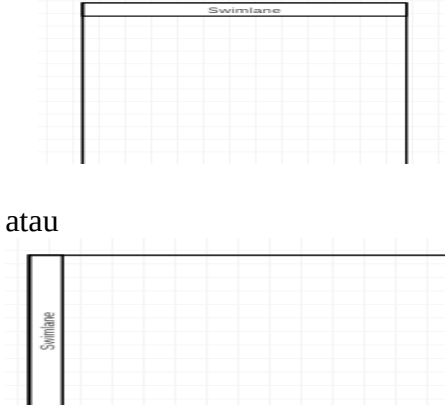
“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat



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 Diagram Activity

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 / <i>decision</i> 	Asosiasi percabangan di mana 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: Sukanto dan Shalahuddin (2018:162)



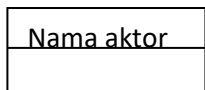
2.2.2.4. Diagram Sequence

“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). “*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu” (Safaat, 2015:33-34).

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.

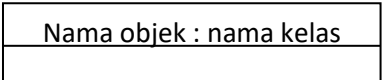
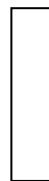
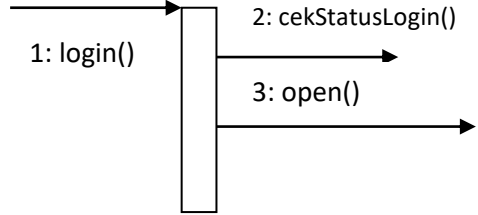
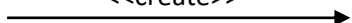
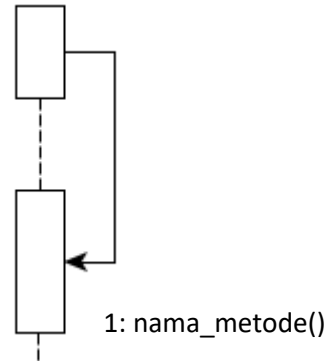
Berikut simbol-simbol pada *sequence diagram* :

Tabel 2.4. Simbol-simbol Diagram Sequence

No	Simbol	Deskripsi
1.	Actor  nama aktor 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 <i>actor</i>

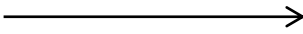
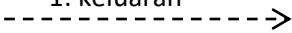
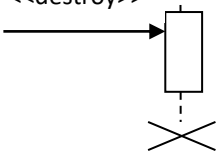


Lanjutan Tabel 2.4. Simbol-simbol Diagram Sequence

No	Simbol	Deskripsi
2.	Garis hidup / <i>lifeline</i> 	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
6.	Pesan tipe call 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, 



Lanjutan Tabel 2.4. Simbol-simbol Diagram Sequence

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

Sumber: Sukanto dan Shalahuddin (2018:165-167)

2.3. Teori Judul

2.3.1. Pengertian Sistem Informasi

Indrajani (2014:3), “Sistem Informasi merupakan kombinasi teratur apa pun dari orang-orang, hardware, software, jaringan komunikasi, dan sumber daya data, yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi”.

2.3.2. Pengertian Pendaftaran

Dalam Kamus Besar Bahasa Indonesia V, pendaftaran adalah proses, cara, perbuatan mendaftar (mendaftarkan); pencatatan nama, alamat, dan sebagainya dalam daftar.



2.3.3. Pengertian Mahasiswa

Dalam Kamus Besar Bahasa Indonesia V, mahasiswa adalah orang yang belajar di perguruan tinggi.

2.3.4. Pengertian Album

Dalam Kamus Besar Bahasa Indonesia V, album adalah buku tempat menyimpan kumpulan foto (potret), perangko, dan sebagainya.

2.3.5. Pengertian Judul Secara Keseluruhan

Sistem Informasi Pendaftaran Mahasiswa Baru di Politeknik Negeri Sriwijaya Studi Kasus Pembuatan Album Ujian dan Cetak Kartu Ujian adalah sistem informasi yang digunakan untuk mempermudah dan mempercepat proses pembuatan album ujian dan percetakan kartu ujian pada pendaftaran mahasiswa baru Politeknik Negeri Sriwijaya.

2.4. Teori Program

2.4.1. Pengertian PHP (*Hypertext Preprocessor*)

Menurut Madcoms (2016:2), “PHP (*Hypertext Preprocessor*) adalah bahasa script yang dapat ditanamkan atau disisipkan kedalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis. PHP sering juga digunakan untuk membangun sebuah CMS. PHP adalah bahasa pemrograman script *server-side* yang yang di desain untuk pengembangan web. Selain itu menurut Wardana (2014:65), “ PHP *Hypertext Preprocessor* atau sering disebut PHP merupakan bahasa pemrograman berbasis *server side* yang dapat melakukan *parsing script php* menjadi script web sehingga dari sisi client menghasilkan suatu tampilan yang menarik”. Berdasarkan pendapat tersebut, dapat disimpulkan bahwa PHP adalah bahasa *script* untuk pengembangan web agar menghasilkan tampilan yang menarik.



2.4.2. Pengertian Basis Data (*Database*)

Sukamto dan Shalahuddin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah ada yang diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.”. Selain itu menurut Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.”. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. terdapat *record-record* yang sejenis, sama besar, sama bentuk, Dalam satu file yang merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari field-field yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu *record*. Definisi Konsep Basis Data, pada basis data ini akan akan dibahas tentang definisi yang terdiri dari *Database*, *File*, *Entity*, dan *Record*. *Entity* adalah orang, tempat, kejadian atau konsep yang informasinya direkam pada suatu basis data misalnya informasi lalu lintas, *entity* antara lain kemacetan, kecelakaan dan lain sebagainya. Database adalah kumpulan field-field yang mempunyai kaitan antara satu file dengan field yang lain sehingga membentuk bangunan data untuk menginformasikan kondisi lalu lintas dalam bahasa tertentu. File adalah kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda-beda datanya.

Jadi dapat disimpulkan bahwa basis data merupakan kumpulan data atau informasi dalam komputer untuk memenuhi berbagai kebutuhan. Dalam membangun Sistem Informasi Pendaftaran Mahasiswa Baru di Politeknik Negeri Sriwijaya Studi Kasus Pembuatan Album Ujian dan Cetak Kartu Ujian, penulis menggunakan sistem *database* MySQL sebagai tempat untuk menyimpan dan mengolah data.

2.4.3. Pengertian HTML (*HyperText Markup Language*)

Fauziah (2014:3) mengemukakan bahwa, “HTML singkatan (*Hyper Text Markup Language*) merupakan jenis bahasa yang digunakan untuk membuat halaman website yaitu dengan menggunakan tag-tag yang telah di deklarasikan



pada halaman *notepad* dan dapat saling berhubungan dengan dokumen HTML yang lainnya yang sering kita kenal dengan istilah *link*".

2.4.4. Pengertian MySQL

Menurut Fathansyah (2015:489), "MySQL merupakan salah satu *database* relasional yang mendukung pemakaian SQL dan dirancang untuk penggunaan aplikasi dengan arsitektur client/server". Terdapat 4 komponen sistem basis data, yaitu Perangkat Keras (*Hardware*), Sistem Operasi (*Operating System*), Basis Data (*Database*), Sistem Pengelola Basis Data (*Database Management System/DBMS*). MySQL digunakan oleh sebagian besar *Web Server* yang ada di jagat internet. Disamping karena dianggap *simple*, juga dapat di-*porting* pada berbagai sistem operasi kelas *server*, seperti Windows, Linux, Solaris, Mac OS, BSD, Unix, IBM-AIX. Sedangkan menurut Madcoms (2016:152), "MySQL adalah salah satu jenis *database server* yang sangat terkenal". Dari beberapa pendapat di atas dapat disimpulkan bahwa MySQL adalah program untuk membuat *database* yang siapa saja boleh menggunakannya.



Gambar 2.3 Logo Mysql

2.4.5. Tipe Data MySQL

Hidayatullah dan Kawistara (2014:181), MySQL menggunakan tipe data numeric standar ANSI. Berikut ini adalah tipe data dalam MySQL yaitu :

2.4.5.1. Tipe Data Numerik

Hidayatullah dan Kawistara (2014:181) "MySQL menggunakan seluruh tipe data numeric standar ANSI".

Berikut ini adalah tipe data numeric yang biasanya digunakan beserta penjelasannya.

**Table 2.5** Tipe Data Numerik

Tipe Data	Deskripsi
INT	Nilai Integer yang bias bertanda atau tidak. Jika Bertanda, maka rentang yang diperbolehkan adalah -2147483648 sampai 2147483647, sedangkan jika tidak bertanda maka rentangnya dari 0 sampai 4294967295
TINYINT	Nilai integer yang sangat kecil. Rentangnya 128-127 untuk yang bertanda dan 0-255 untuk yang tidak bertanda.
SMALLINT	Nilai integer yang sangat kecil dengan rentang 31768 sampai 32767 untuk yang bertanda sedangkan untuk yang tidak bertanda dari 0-65535.
MEDIUMINT	Integer dengan ukuran sedang dengan rentang -8388608 sampai 8388607a atau 0 sampai 16777215.
BIGINT	Integer dengan ukuran besar dengan rentang 9223372036854775808 sampai 92233722036854775807 atau 0 sampai 18446744073709551615.
FLOAT(M,D)	Bilangan pecahan dengan panjang (termasuk jumlah decimal) M dan jumlah decimal D. Presisi desimalnya bisa sampai 24 digit. Defaultnya float (10,2). Bilangan float selalu bisa bertanda
DOUBLE(M,D)	Adalah bilangan pecahan dengan presisi dua kali lipat. Panjang (termasuk jumlah decimal M dan jumlah decimal D. Presisi desimalnya bisa sampai 53 digit. Defaultnya Double(16,4). Bilangan float selalu bisa bertanda, Sinonim dari DOUBLE and REAL
DECIMAL(M,D)	Adalah bilangan pecahan dan harus didefinisikan M dan D-nya. Setiap decimal membutuhkan tempat 1 byte. Sinonim dari DECIMAL dan NUMERIC.

(Sumber: Hidayatullah dan Kawistara, 2014:181)



2.4.5.2. Tipe Data Tanggal dan Waktu

Berikut ini adalah tipe data tanggal dan waktu di dalam MySQL yakni :

Table 2.6 Tipe Data Tanggal dan Waktu

Tipe Data	Deskripsi
DATE	Adalah tipe data tanggal dengan format YYYY-MM-DD antara 1000-01-01 and 9999-12-31 Contoh: 17 Agustus 1945 akan di simpan sebagai 1945-08-17
DATETIME	Adalah kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:SS dan rentang data antara 1000-01-01 00-00-00 sampai dengan 9999-12-31 23-59-59. Contoh: Pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 1945-08-17 10:00:00
TIMESTAMP	Sebuah penanda waktu antara 1 Januari 1970 tengah malam sampai dengan tahun 2037. Formatnya mirip dengan DATETIME tetapi tanpa pembatas di antara angkanya. Contoh: pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 19450817000000
TIME	Menyimpan waktu dalam format HH:MM:SS contoh pukul 10:00 akan disimpan menjadi 10:00:00
YEAR(M)	Menyimpan data tahun dalam format 2 dan 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 1970 – 2069 sedangkan jika M diisi dengan nilai 4 maka YEAR bisa bernilai 1901 sampai dengan 2155. Default nilai M adalah 4.

(Sumber: Hidayatullah dan Kawistara, 2014:182)



2.4.5.3. Tipe Data String

Berikut ini adalah tipe data String yang ada di dalam MySQL yaitu :

Table 2.7 Tipe Data String

Tipe Data	Deskripsi
CHAR(M)	String dengan ukuran tetap. Ukurannya antara 1 sampai 255 karakter. Ukuran ditentukan dengan nilai M. Contoh: CHAR (6)
VARCHAR(M)	String dengan ukuran bervariasi antara 1 sampai dengan 255 karakter. Contoh: VARCHAR(25)
TEXT	String dengan ukuran maksimum 65535 karakter. String yang tersimpan di dalam TEXT dianggap tidak case sensitive. Untuk kapasitas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karakter. Sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT (maksimal 16777215 karakter) dan LONGTEXT (maksimal 4294967295 karakter).
BLOB	Binary Large Object(BLOB) adalah tipe data untuk menyimpan data binary dalam jumlah besar. Biasanya digunakan untuk menyimpan citra. Untuk menyimpan data yang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) sedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMBLOB (maksimal 16777215 karakter) dan LONGBLOB (maksimal 4294967295 karakter).
ENUM	Enumerasi atau sebuah list (daftar). Jadi misalnya anda ingin bahwa sebuah nilai terbatas hanya boleh dengan nilai tertentu saja maka anda bisa membuat sebuah daftar. Misalnya saja nilai hanya bisa menjadi ENUM ('A','B','C','D','E').

(Sumber: Hidayatullah dan Kawistara, 2014:183)



2.4.6. Pengertian Web Server

Madcoms (2016:4), “Web server adalah suatu program computer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari computer klien, yang dikenal dengan nama web browser dan melayani mereka dengan menyediakan respon HTTP berupa konten data, biasanya berupa halaman web yang terdiri dari dokumen HTML dan objek terkait seperti gambar dan lain-lain. Untuk bisa mengakses aplikasi web yang akan dibuat oleh penulis, maka diperlukan yang namanya *webserver*. Penulis menggunakan *Xampp* sebagai *webserver* dalam membangun Sistem Informasi Pendaftaran Mahasiswa Baru di Politeknik Negeri Sriwijaya Studi Kasus Pembuatan Album Ujian dan Cetak Kartu Ujian.

2.4.6.1. Pengertian XAMPP

Pratama (2014:440), “XAMPP adalah aplikasi web server bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi linux maupun sistem operasi windows”. Selain itu menurut Santosa & Nurmalina (2016:86-87), “Dengan menggunakan XAMPP tidak dibingungkan dengan penginstalan program-program lain, karena semua kebutuhan telah tersedia oleh XAMPP. Yang terdapat pada XAMPP diantaranya: Apache, MySQL, PHP, FilZilla FTP Server, PhpmyAdmin dll”. Berdasarkan pendapat tersebut, dapat disimpulkan bahwa XAMPP yaitu aplikasi web server dengan kebutuhan akan program-program telah terinstal.



Gambar 2.4 Logo XAMPP



2.4.7. Pengertian Sublime Text

Menurut Supono & Putratama (2018:14), “Sublime Text merupakan perangkat lunak *text editor* yang digunakan untuk membuat atau meng-edit suatu aplikasi. Sublime Text memiliki fitur *plugin* yang memudahkan programmer. Selain itu, Sublime Text juga memiliki desain yang simple dan keren menjadikan Sublime Text terkesan elegan untuk sebuah *syntax editor*. Selain ringan, IDE ini memiliki kecepatan proses simpan dan buka file. Tidak heran kalau IDE ini paling banyak digunakan terutama dikalangan programmer berbasis web”.