

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Sistem Akademik Universitas Surakarta berbasis Delphi (Form Data Konversi Mahasiswa *transfer*)

Form data konversi pada Sistem Akademik Universitas Surakarta adalah form untuk menyimpan data konversi nilai mahasiswa *transfer* dari universitas lain untuk disesuaikan mata kuliah asal dengan mata kuliah di Universitas Surakarta sesuai dengan program studi. Form konversi Universitas Surakarta ditampilkan pada Gambar 2.1.

MK Asal	SKS	Nilai	MK Program S-1	SKS	MK Harus Ditempuh	SKS
			PENDIDIKAN AGAMA ISLAM	2	PENDIDIKAN AGAMA ISLAM	2
			PENDIDIKAN PANCA SILA	2	PENDIDIKAN PANCA SILA	2
			PRAK ALGORITMA &	1	PRAK ALGORITMA &	1
			FISIKA KOMPUTER	3	FISIKA KOMPUTER	3
			BAHASA INGGRIS I	2	BAHASA INGGRIS I	2
			PENDIDIKAN AGAMA KRISTEN	2	PENDIDIKAN AGAMA KRISTEN	2
			PENDIDIKAN AGAMA KATHOLIK	2	PENDIDIKAN AGAMA KATHOLIK	2
			PENDIDIKAN AGAMA HINDU	2	PENDIDIKAN AGAMA HINDU	2
			PENDIDIKAN AGAMA BUDHA	2	PENDIDIKAN AGAMA BUDHA	2
			KALKULUS	3	KALKULUS	3
			ALGORITMA &	2	ALGORITMA &	2
			PENOLAHAN DATA	2	PENOLAHAN DATA	2

SKS Program S-1 : 155
SKS yang Diikuti : 0
SKS yang Harus Ditempuh : 150

Gambar 2.1 Form konversi Universitas Surakarta

Form ini hanya dapat diakses oleh operator bagian akademik dan administrasi saja, dan hanya diperuntukkan mahasiswa *transfer* saja. Cara kerja Form konversi ini dengan menjumlahkan seluruh SKS dari program studi yang dipilih dan akan dikurangi mata kuliah asal, maka akan didapatkan hasil jumlah SKS yang harus ditempuh dan mata kuliah apa saja yang harus diambil.

Proses akhir dari form konversi ini adalah sebuah data berbentuk Excel (*.xls) dan Html (*.html) yang dapat digunakan untuk mencetak laporan hasil konversi nilai.

2.1.2. Sistem Informasi Komputerisasi Akademik STMIK Triguna Dharma berbasis Visual Basic (Form Konversi)

Sistem Informasi merupakan kebutuhan pokok dalam suatu institusi pendidikan saat ini. dengan adanya sistem informasi dapat membantu penyelenggaraan pendidikan yang dikelola secara manual dapat ditransformasikan kedalam sebuah sistem informasi yang sistematis, efisien, dan efektif. Untuk itulah Universitas STMIK Triguna Dharma merancang beberapa macam sistem untuk kebutuhan akademik, salah satunya sistem Konversi nilai *transfer* mahasiswa.

Form Konversi nilai digunakan untuk mengkonversi mata kuliah mahasiswa *transfer*. Konversi ini bertujuan untuk menyesuaikan mata kuliah dari perguruan tinggi lainnya dengan mata kuliah yang ada di STMIK Triguna Dharma. Form Konversi Nilai *Transfer* ditampilkan pada Gambar 2.2.

FORM KONVERSI STMIK TRIGUNA DHARMA

Dari TGD Luar TGD

Asal PTS : amik

Nama : saktia

Tempat, Tgl Lahir : 12 Wednesday, February

NIRM Asal :

Program Studi : 31 - Manajemen Informatika

NO	MK ASAL	SKS	NIL	KI	No	Kode MK	Nama MK	RPP	SKS	NIL	SEM
28	BISNIS	2	C		1	MI101	Pancasila	33	2	A	I
29	DISAIN WEB	3	C		2	MI102	Akuntansi Dasar		2		I
30	ELEKTRONIKA	3	B		3	MI103	Aplikasi Finansial Terapan		2		I
31	KALKULUS	3	A		4	MI104	Bahasa Inggris 1		2		I
32	KECERDASAN BUA...	3	B		5	MI105	Paket Program Niaga 1		2		I
33	KECERDASAN BUA...	3	B		6	MI106	Disain Grafis 1		2		I
34	PEMASARAN	2	A		7	MI107	Sistem Operasi		3		I
35	WARIGA NEGARA	2	A		8	MI108	Algoritma Dan Pemrograman 1		3		I
36	WIRAU SAHA	2	A		9	MI109	Struktur Data		3		I
37	BISNIS	2	C		10	MI201	Akuntansi Lanjutan		2		II
38	DISAIN WEB	3	C		11	MI202	Bahasa Inggris 2		2		II
39	ELEKTRONIKA	3	B		12	MI203	Paket Program Niaga 2		2		II
40	KALKULUS	3	A		13	MI204	Disain Grafis 2		2		II
41	KECERDASAN BUA...	3	B		14	MI205	Komputer Teknik		3		II
42	PANCASILA	2	A		15	MI206	Pemrograman Visual 1		2		II
43	PEMASARAN	2	A		16	MI207	Sistem Basis Data		3		II
44	WARIGA NEGARA	2	A		17	MI208	Algoritma Dan Pemrograman 2		3		II
45	WIRAU SAHA	2	A		18	MI209	Pengantar Instalasi Komputer		3		II
46	BISNIS	2	C		19	MI301	Kewarganegaraan		2		III
47	DISAIN WEB	3	C		20	MI302	Pendidikan Agama		2		III
48	ELEKTRONIKA	3	B		21	MI303	Ajabar Linier		2		III
49	KALKULUS	3	A		22	MI304	Pengantar Manajemen		2		III
50	KECERDASAN BUA...	3	B		23	MI305	Komputer Akuntansi Dasar		2		III
51	PANCASILA	2	A		24	MI306	Statistik Dasar		3		III

Total Seluruh SKS : -- Semester yang harus dijalani : --

Jumlah SKS yang sudah OK : --

Sisa SKS yang Harus diambil : --

Upload MK Aktifkan Simpan Pilih Simpan Cetak Batal Keluar

Tekan F5 untuk Mencari Mahasiswa Transfer 2:08 PM 22/06/2011

Gambar 2.2 Form Konversi Nilai *Transfer*

2.1.3. Perancangan Sistem Konversi Mahasiswa *Transfer* Jurusan Informatika Jenjang Sarjana Pada Perguruan Tinggi Raharja

Skripsi perancangan sistem konversi mahasiswa *transfer* jurusan Informatika jenjang sarjana pada perguruan tinggi raharja oleh yessi oktavyanti jurusan infomatika konsentrasi *software* engineering sekolah tinggi manajemen dan ilmu komputer STMIK Raharja Tangerang tahun 2015/2016

Perancangan ini dibuat untuk meningkatkan efektifitas dalam pengerjaan konversi atau penyetaraan mata kuliah. Perangkat lunak yang digunakan untuk membuat dan menjalankan aplikasi diantaranya, *Web browser, PHPMYADMIN, XAMPP Server*. Observasi dan wawancara mengenai kebutuhan pengguna disajikan pada Tabel 2.1.

Tabel 2.1. Kebutuhan Pengguna Sistem konversi Perguruan Tinggi Raharja

Functional	
Analisa Kebutuhan	
Saya ingin sistem dapat :	
1.	Menampilkan menu login dengan username dan password
2.	Menampilkan rekap laporan kurikulum
3.	Adanya menu kurikulum
4.	Adanya menu mahasiswa
5.	Adanya tombol tambah kurikulum pada menu kurikulum
6.	Adanya tombol edit kurikulum pada menu kurikulum
7.	Adanya tombol hapus kurikulum pada menu kurikulum
8.	Dapat menampilkan kurikulum per jurusan
9.	Adanya tombol edit mahasiswa pada menu mahasiswa
10.	Adanya tombol hapus mahasiswa pada menu mahasiswa
11.	Dapat mencari data mahasiswa melalui nim
12.	Dapat menampilkan NIM, nama, jurusan dan konsentrasi mahasiswa
13.	Dapat menampilkan menu tambah mata kuliah konversi

Analisa Kebutuhan	
Saya ingin sistem dapat :	
14.	Dapat mengedit mata kuliah konversi
15.	Dapat menghapus mata kuliah konversi
16.	Dapat menampilkan kode mata kuliah
17.	Dapat menampilkan nama mata kuliah secara otomatis
18.	Dapat menampilkan jumlah sks mata kuliah secara otomatis
19.	Dapat menampilkan laporan konversi per semester
20.	Dapat menampilkan laporan jumlah mahasiswa <i>transfer</i> per semester
21.	Terdapat menu cetak konversi
22.	Dapat menampilkan foto mahasiswa
Non Function	
ANALISA KEBUTUHAN	
Saya ingin sistem dapat :	
1.	Bersifat user friendly (mudah dipahami user)
2.	Sistem berjalan dengan baik

2.1.4. Perancangan Sistem Konversi Mahasiswa Universitas Sahid Surakarta

Berdasarkan tinjauan pustaka dari 3 sistem diatas, dapat disimpulkan bahwa terdapat kesamaan tujuan dari dari pembuatan sistem konversi yaitu membuat sebuah sistem yang dapat mengolah data mahasiswa yang melakukan *transfer* ataupun alih jenjang untuk menyesuaikan mata kuliah asal dengan mata kuliah di universitas sistem dibuat. Adapun kekurangan dari 3 sistem yang telah dibuat diatas yaitu sebagai berikut :

1. Sistem Akademik Universitas Surakarta berbasis Delphi adalah aplikasi desktop yang hanya dapat diakses secara offline dimana aplikasi tersebut diinstal dan tidak memiliki basis data yang disimpan dalam *database*.

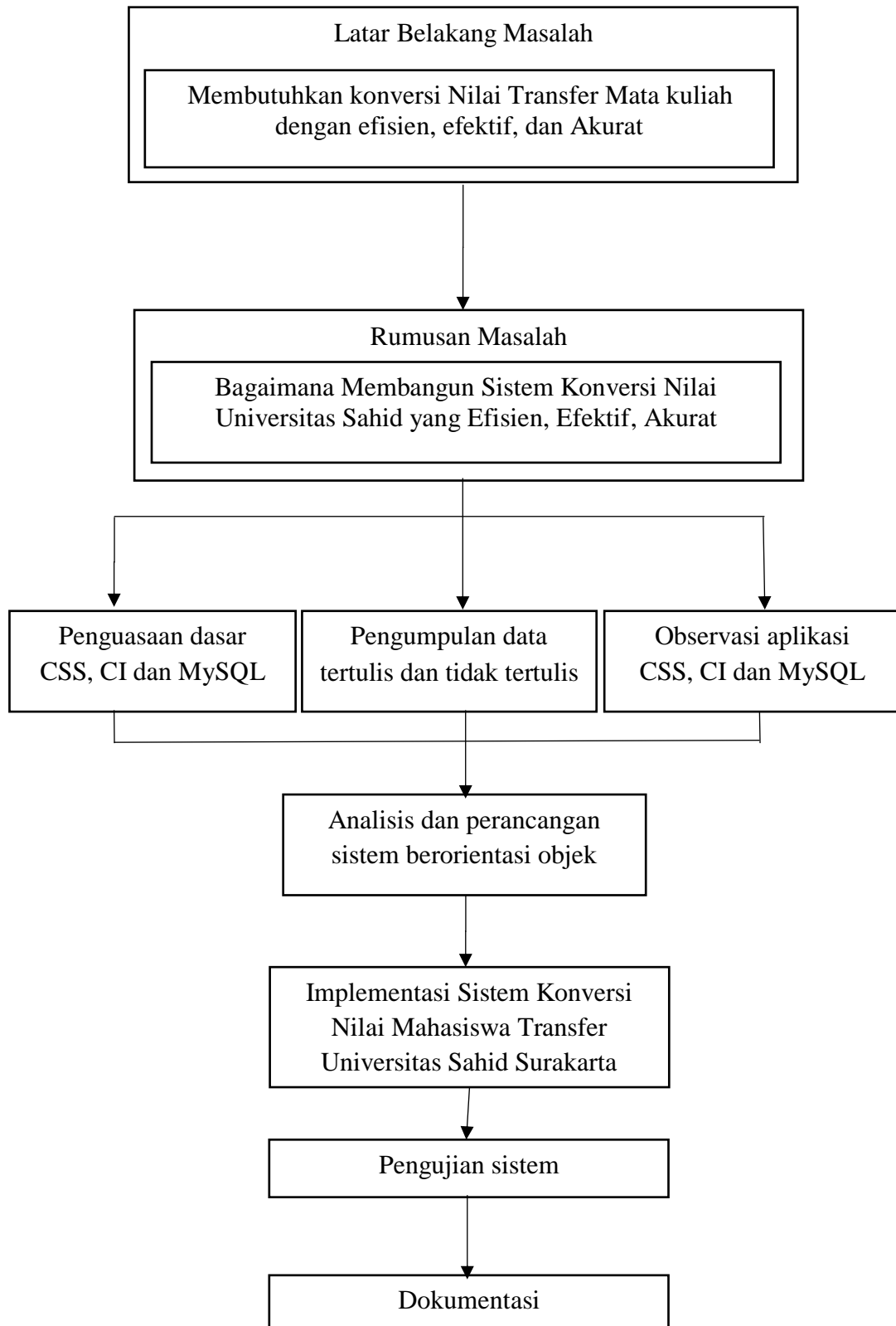
2. Sistem Informasi Komputerisasi Akademik STMIK Triguna Dharma berbasis Visual Basic ini dibuat untuk perorangan dan tidak bisa digunakan secara publik
3. Perancangan Sistem Konversi Mahasiswa *Transfer* Jurusan Informatika Jenjang Sarjana Pada Perguruan Tinggi Rahaarja belum memiliki menu untuk menyalin penginputan data dari mahasiswa yang berasal dari universitas yang sama, sehingga tidak dibutuhkan penginputan ulang agar proses pengolahan data transkrip nilai mahasiswa dapat dilakukan secara cepat dan efisien.

Dari kelemahan yang dimiliki 3 sistem yang dibuat diatas, maka dibuat analisa kebutuhan sistem yang disajikan pada Tabel 2.2.

Tabel 2.2. Tabel Analisa Kebutuhan Sistem yang akan dibuat

<i>Functional</i>	
Analisa Kebutuhan	
Saya ingin sistem dapat :	
1.	Dapat menginputkan data transkrip mahasiswa secara cepat
2.	Dapat menyalin data transkrip mahasiswa yang pernah diinputkan sebelumnya
3.	Melakukan penyetaraan mata kuliah secara otomatis
4.	Menghasilkan output hasil konversi
5.	Menghasilkan output transkrip nilai
<i>Non Functional</i>	
Analisa Kebutuhan	
Saya ingin sistem dapat :	
1.	Menampilkan hasil konversi yang dapat diakses mahasiswa secara online

2.2. Kerangka Berfikir



Gambar 2.3 Kerangka Berfikir

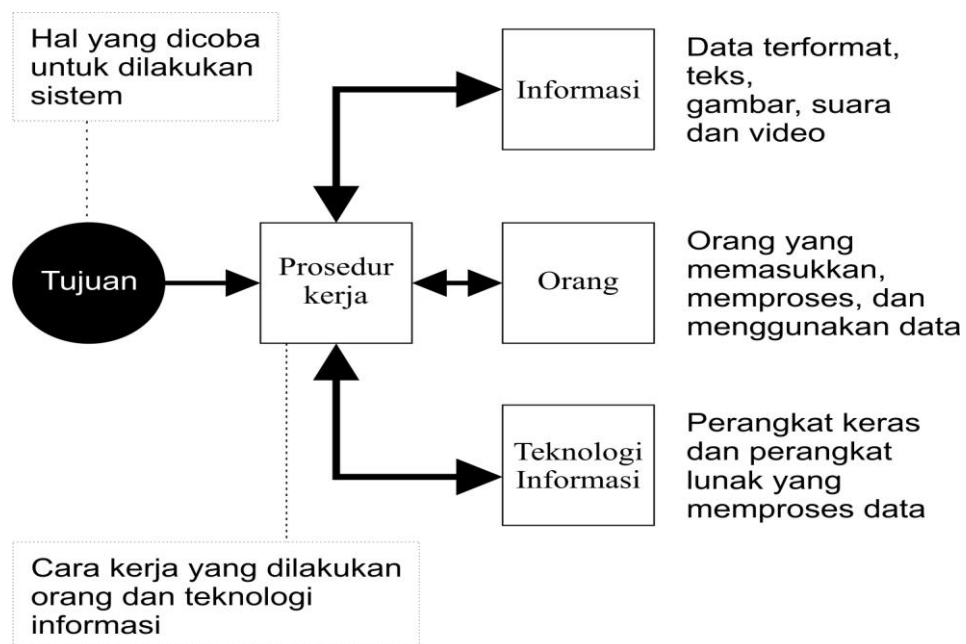
2.3. Teori Pendukung

2.3.1. Sistem

Sistem adalah suatu kesatuan, baik objek nyata atau abstrak yang terdiri dari berbagai komponen atau unsur berkaitan, saling tergantung, saling mendukung, dan secara keseluruhan bersatu dalam satu kesatuan untuk mencapai tujuan tertentu secara efektif dan efisien. Sistem juga diartikan sebagai tatanan (keterpaduan), sekelompok unsur atau elemen yang saling berhubungan satu dengan yang lain untuk mencapai satu tujuan. (Rahmawati, 2017)

2.3.1.1. Sistem Informasi

Menurut Alter (dalam Abdul Kadir, 2014) menjelaskan definisi Sistem Informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi. Definisi sistem informasi digambarkan pada Gambar 2.4.



Gambar 2.4 Definisi Sistem Informasi

2.3.1.2. Sistem Konversi Nilai

Konversi memiliki makna yang sama dengan kata penyetaraan, akan tetapi, konversi adalah istilah yang lebih sering digunakan dalam sebuah sistem. Kata

konversi tidak hanya digunakan dalam dunia pendidikan, namun lebih sering digunakan dalam nilai tukar uang suatu negara terhadap negara lain.

Sistem konversi nilai yaitu mengkonversikan nilai dan jumlah SKS mata kuliah yang telah ditempuh atau diperoleh dari kurikulum lama ke kurikulum baru. Pengkonversian nilai mata kuliah dilakukan dengan tujuan untuk memperoleh suatu hasil atau keputusan yang berhubungan dengan mata kuliah serta jumlah SKS dari mahasiswa dengan matakuliah kurikulum lama dan matakuliah kurikulum baru (Rini dkk., 2016).

2.3.2. Website

Website adalah sebuah kumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan serta dapat diakses secara luas menggunakan sebuah *browser*. *Website* dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam, bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang membentuk suatu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan. (Hastanti, 2015)

2.3.3. CodeIgniter

Menurut (Raharjo, 2015) *Codeigniter* adalah *framework web* untuk bahasa pemrograman PHP, yang dibuat oleh Rick Ellis pada tahun 2006, penemu dan pendiri EllisLab. *CodeIgniter* merupakan sebuah *toolkit* yang ditujukan untuk orang yang ingin membangun aplikasi *web* dalam bahasa pemrograman PHP. Beberapa keunggulan yang ditawarkan oleh *CodeIgniter* adalah sebagai berikut :

1. CodeIgniter adalah framework yang bersifat free dan opensource.
2. CodeIgniter memiliki ukuran yang kecil dibandingkan dengan framework lain. Setelah proses instalasi, framework CodeIgniter hanya berukuran kurang lebih 2 MB. Dokumentasi CodeIgniter memiliki ukuran sekitar 6 MB.
3. Aplikasi yang dibuat menggunakan CodeIgniter bisa berjalan cepat.
4. CodeIgniter menggunakan pola desain Model-View-Controller (MVC) sehingga satu file tidak terlalu berisi banyak kode. Hal ini menjadikan kode lebih mudah dibaca, dipahami, dan dipelihara dikemudian hari.

5. CodeIgniter dapat diperluas sesuai dengan kebutuhan.
6. CodeIgniter terdokumentasi dengan baik. Informasi tentang pustaka kelas dan fungsi yang disediakan oleh CodeIgniter dapat diperoleh melalui dokumentasi yang disertakan di dalam paket distribusinya.

2.3.4. Framework

Alasan seorang programmer professional menggunakan *Framework* dalam membuat sebuah sistem adalah kemudahan yang diberikan. Pada *Framework*, sudah tersedia struktur aplikasi yang baik seperti standard coding, best practice, design pattern dan common *function* sehingga programmer professional dapat langsung focus kepada proses pembangunan sistem tanpa harus melakukan mindset atau berfikir pada masalah struktur aplikasi, standard coding dan lain sebagainya.

Framework adalah suatu kumpulan kode berupa pustaka (*library*) dan alat (*tool*) yang dipadukan sedemikian rupa menjadi satu kerangka kerja (*framework*) guna memudahkan dan mempercepat proses pengembangan aplikasi *web*. Jadi, *Framework* adalah kumpulan-kumpulan potongan program yang dipadukan menjadi satu kerangka kerja yang digunakan untuk membantu dalam pembuatan sebuah aplikasi. (Raharjo, 2015)

2.3.5. Model View Controller (MVC)

Menurut (Hidayatullah dan Kawistara, 2017) MVC (*Model View Controller*) adalah sebuah metode untuk membuat sebuah aplikasi. Terdapat tiga komponen dalam MVC yaitu :

1. *Model*, Model berhubungan dengan *database* sehingga biasanya dalam *model* akan berisi *class* ataupun fungsi untuk *create*, *update*, *delete*, *search*, dan *select* pada *database*. Selain itu juga model akan berhubungan dengan perintah-perintah *query* sebagai tindak lanjut dari fungsi-fungsi.
2. *View*, bagian *User Interface* atau bagian tampilan untuk *end-user*. View bisa berupa halaman HTML, CSS, *Javascript* , *JQuery* dan *AJAX*. View tidak terdapat pemrosesan data ataupun pengaksesan *database*, sehingga *view* hanya menampilkan data-data hasil dari *Model* dan *Controller*.
3. *Controller*, penghubung antara *view* dan *model*, jadi *controller* digunakan sebagai jembatan keduanya. Sehingga tugas *controller* ialah sebagai

pemrosesan data atau Alur *Logic Program*, menyediakan *variable* yang akan ditampilkan di *view*, pemanggilan *model* sehingga *model* dapat mengakses *database*, *error handling*, validasi atau *check* terhadap suatu input data.

2.3.6. HTML

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan sebagian informasi di dalam sebuah penjelajah *web* internet dan permformatan *hypertext* sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. HTML adalah bahasa pemrograman *web* yang umum dan bisa digunakan serta mudah dipelajari. (Rahmawati, 2017)

2.3.7. PHP

PHP merupakan bahasa pemrograman yang bisa digunakan untuk membuat halaman HTML yang memungkinkan aplikasi *web* dinamis untuk mengelola data, pemrosesan data dari *user via form*, membuat toko *online*, dan lain sebagainya dengan mudah. PHP juga merupakan bahasa pemrograman berbasis *server side*, yang dapat dilakukan *parsing script* PHP menjadi *script web* sehingga dapat melakukan koneksi ke database, dan dari sisi *client* menghasilkan suatu tampilan yang menarik (Rahmawati, 2017)

2.3.8. Database

Database atau basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* ataupun *database management system* (Rosa, A., & Shalahuddin, 2016)

2.3.9. SQL

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengakses basis data yang tergolong relasional. Standar SQL mula-mula

didefinisikan oleh ISO (*Internet Standard Organization*) dan ANSI (*the American National Standards Institute*), yang dikenal dengan sebutan SQL86. Standar terakhir ketika buku ini disusun berupa SQL99. Pemahaman terhadap SQL sangat bermanfaat karena anda juga bisa memanfaatkannya untuk keperluan pemrograman.

Sesungguhnya SQL tidak terbatas hanya untuk mengambil data (*query*), tetapi juga dapat dipakai untuk menciptakan tabel, menghapus tabel, menambahkan data ke tabel, menghapus data di tabel, mengganti data di tabel, berbagai operasi yang lain. (Abdul Kadir, 2014)

2.3.10. Perancangan Berorientasi Objek


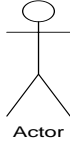

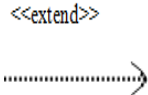

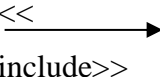
Perancangan berorientasi Objek adalah suatu teknik atau cara pendekatan baru dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Pengertian “berorientasi objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan pembangunan sebuah aplikasi berorientasi objek, yaitu UML (*Unified Modelling Language*). UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. (Munawar, 2018)

Ada 5 macam diagram dalam UML (*Unified modeling language*) yaitu:

2.3.10.1. Use Case

Use case adalah deskripsi fungsi dari sebuah sistem dari prespektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (Munawar, 2018). Simbo *use case* diagram dapat dilihat pada Tabel 2.3.

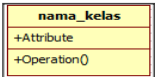
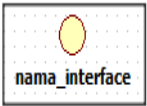




Tabel 2.3 Simbol *Use Case* Diagram

No	Simbol	Nama Komponen	Keterangan
1.		<i>Use case</i>	Fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan kata kerja di awal fase nama <i>use case</i>
2.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri
3.		<i>Asosiation</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.		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu
5.		<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
6.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini

2.3.10.2. Class Diagram

Class diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. *Class* diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi dari aplikasi perangkat lunak. *Class* diagram menggambarkan *attribut*, *operation* dan juga *constraint* yang terjadi pada sistem. *Class* diagram banyak digunakan dalam pemodelan sistem karena mereka adalah satu-satunya diagram UML, yang dapat dipetakan langsung struktur dengan bahasa berorientasi objek (Munawar, 2018). Simbol *Class* diagram dapat dilihat pada Tabel 2.4.







Tabel 2.4 Simbol *Class* Diagram

No	Simbol	Nama Komponen	Keterangan
1.		Kelas / <i>Class</i>	Kelas pada struktur sistem
2.		Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam objek pemrograman berorientasi objek
3.		Asosiasi / <i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya jugadisertai dengan <i>multiplicity</i>
4.		Generalisasi / <i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (Umum-khusus)
5.		Kebergantungan / <i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
6.		Agregasi / <i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>Whole-part</i>)

2.3.10.3. Activity Diagram

Activity diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika prosedur, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam *activity* diagram. *Activity* diagram mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity* diagram bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa. (Munawar,2018). Simbol *Activity* Diagram ditunjukkan pada Tabel 2.5.

Tabel 2.5 Simbol *Activity* Diagram

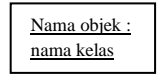
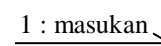
NO	Simbol	Nama Komponen	Keterangan
1		Status awal	Status awal aktifitas sistem
2		Aktifitas	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktifitas digabungkan menjadi satu
5		Status Akhir	Status akhir yang dilakukan sistem
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.3.10.4. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah Objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara Objek juga interaksi antara Objek. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequence* diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh Objek dan *message* (pesan) yang diletakkan diantara Objek- Objek

di dalam *use case*. Komponen utama *sequence* diagram terdiri atas Objek yang dituliskan dengan kotak segi empat bernama *message* diwakili oleh garis dengan tanda panah dan waktu ditunjukkan dengan *progress* vertikal (Munawar,2018). *Sequence* diagram dapat dilihat pada Tabel 2.6.

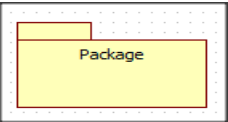
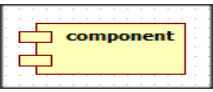



Tabel 2.6 Simbol *Sequence* Diagram

NO	Simbol	Nama	Keterangan
1.		Aktor	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 fase nama aktor
2.		<i>Lifeline</i>	Menyatakan kehidupan suatu objek
3.		Objek	Menyatakan objek yang berinteraksi pesan
4.		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi
5.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data /masukan /informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim

2.3.10.5. Component Diagram

Component diagram adalah bagian fisik dari sebuah sistem, karena menetap di komputer, bukan di benak para analis. Komponen bisa berupa tabel, file data, file exe, atau dokumen (Munawar, 2018). Simbol *component* diagram dapat dilihat pada Tabel 2.7


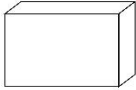


Tabel 2.7 Simbol *Component Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Package</i>	Package merupakan sebuah bungkusan dari satu atau lebih komponen.
2.		<i>Component</i>	Komponen system.
3.		<i>Dependency</i>	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4.		<i>Interface</i>	Sama dengan interface pada pemrograman berbasis objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
5.		<i>Link</i>	Relasi antar komponen.

2.3.10.6. Deployment Diagram

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*. (Munawar, 2018). Bagian utama hardware adalah node, yaitu nama umum untuk semua jenis sumber komputasi. Ada 2 tipe node yaitu *processor* dan *device*. *Deployment* diagram digunakan untuk menggambarkan topologi fisik komponen sistem, tempat dimana komponen perangkat lunak diimplemen. Simbol *Deployment* diagram ditunjukkan pada Tabe 2.8.

Tabel 2.8. Simbol *Deployment Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Component instance</i>	Suatu komponen dari infrastruktur sistem
2.		<i>Node</i>	Menggambarkan workstation, server atau hardware lainnya
3.		<i>Node Association</i>	Menggambarkan hubungan node ke node
4.		<i>Dependency</i>	Menggambarkan hubungan dari komponen ke komponen

2.3.11. Pengujian Sistem dengan *Black Box*

Pengujian menggunakan sekumpulan aktifitas validasi dengan pendekatan *black box* menurut Rosa, A., & Shalahudin (2016), Pengujian *Black box* adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksud untuk mengetahui apakah fungsi – fungsi masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian *Black Box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *black box* harus dibuat dengan kasus benar dan kasus salah.

2.3.12. Pengujian Sistem dengan Kuisisioner

Kuisisioner merupakan daftar pernyataan tertulis yang diberikan kepada subjek yang diteliti untuk mengumpulkan informasi yang dibutuhkan penulis Menggunakan kuisisioner bertujuan untuk mendapatkan penilaian saat proses pengujian. Kuisisioner pada umumnya bermodel tabel, kuisisioner yang terdiri dari baris dan kolom, pada kolom pertama berisi pernyataan yang sesuai dengan kebutuhan penelitian, kemudian kolom selanjutnya berisi tentang skala nilai untuk mengetahui nilai dari setiap pertanyaan yang disajikan. Penelitian ini membutuhkan dua tipe skala untuk pengujian aspek usability. Kuisisioner pertama dibentuk dalam skala dua poin dengan jawaban ya dan tidak, yang dianggap cocok untuk

menghitung rumus di setiap nama metrik. Koesioner kedua dibentuk dalam skala lima poin dengan model skala *likert* untuk memilih jawaban sebagai pengukuran tingkat persetujuan pengguna terhadap pernyataan. (Kusumah & Dedi, 2015).