



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Berikut ini ada beberapa definisi tentang komputer yang disajikan oleh beberapa para ahli :

Sujatmiko (2012:156), “Komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program”.

Sutanta (2011:01), “Komputer berasal dari bahasa latin, yaitu *computare* yang berarti menghitung (*to compute / to reckon*)”.

Jogiyanto (2006:2), “Komputer adalah alat elektronik yang menerima input data, mengolah data, dan memberikan informasi dengan menggunakan suatu program yang tersimpan di memori komputer (*stored program*) dan menyimpan program dan hasil pengolahan yang bekerja secara otomatis.

Dari definisi diatas, dapat disimpulkan bahwa komputer adalah alat elektronik yang dapat menghitung atau mengolah data dengan mengikuti serangkaian perintah.

2.1.2. Pengertian Aplikasi

Berikut ini ada beberapa definisi tentang aplikasi yang disajikan oleh beberapa para ahli :

Sutabri (2012:147), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”.

Sujatmiko (2012:23), “Aplikasi merupakan program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Ms. Word, Ms. Excel*”.

Dari definisi diatas, dapat disimpulkan bahwa aplikasi adalah penerapan program yang siap untuk digunakan dalam mengerjakan tugas-tugas tertentu.



2.1.3. Pengertian Rapor

Berikut ini ada beberapa definisi tentang rapor yang disajikan oleh beberapa para ahli :

Gultom dan Hatta (2014:05), “Rapor adalah kumpulan nilai prestasi belajar murid di sekolah yang berupa laporan guru kepada orang tua atau wali murid”.

Surapranata (2014:27), “Rapor merupakan dokumen yang menjadi penghubung komunikasi baik antara sekolah dengan orang tua peserta didik maupun dengan pihak-pihak lain yang ingin mengetahui tentang hasil belajar peserta didik pada kurun waktu tertentu”.

Dari definisi diatas, dapat disimpulkan bahwa rapor adalah hasil dari kumpulan nilai murid selama belajar di sekolah sehingga menjadi laporan nilai.

2.1.4. Pengertian Kurikulum

Berikut ini ada beberapa definisi tentang kurikulum yang disajikan oleh beberapa para ahli :

Gultom dan Nuh (2014:04), “Kurikulum merupakan salah satu unsur yang memberikan kontribusi untuk mewujudkan proses berkembangnya kualitas potensi peserta didik tersebut”.

Nursidik (2013:01), “Kurikulum adalah seperangkat rencana dan pengaturan mengenai tujuan, isi dan bahan pelajaran serta cara yang digunakan sebagai pedoman penyelenggaraan kegiatan pembelajaran untuk mencapai tujuan pendidikan tertentu”.

Dari definisi diatas, dapat disimpulkan bahwa kurikulum adalah seperangkat rencana mengenai penyelenggaraan kegiatan pembelajaran untuk mencapai tujuan pendidikan tertentu.

2.1.5. Pengertian Kurikulum 2013

Berikut ini ada beberapa definisi tentang Kurikulum 2013 yang disajikan oleh beberapa para ahli :

Gultom dan Nuh (2014:04), “Kurikulum 2013 dikembangkan berbasis pada kompetensi sangat diperlukan sebagai instrument untuk mengarahkan peserta



didik menjadi : (1) manusia yang berkualitas yang mampu dan proaktif menjawab tantangan zaman yang selalu berubah; (2) manusia terdidik yang beriman dan bertakwa kepada Tuhan Yang Maha Esa, berakhlak mulia, sehat, berilmu, cakap, kreatif, mandiri; dan (3) warga negara yang demokratis, bertanggung jawab”.

Surapranata (2014:17), “Kurikulum 2013 merupakan kurikulum yang disiapkan untuk menjawab tantangan masa kini dan masa depan”.

Dari definisi diatas, dapat disimpulkan bahwa kurikulum 2013 merupakan pengembangan sikap, pengetahuan, kreativitas dan keterampilan serta menerapkannya dalam berbagai situasi di sekolah dan masyarakat.

2.1.6. Pengertian Aplikasi Rapor Kurikulum 2013 Pada SMPN 2 Air Kumbang Kabupaten Banyuasin

Aplikasi Rapor Kurikulum 2013 Pada SMPN 2 Air Kumbang merupakan suatu program kumpulan nilai belajar murid di sekolah yang berdasarkan kurikulum operasional yang disusun dan dilaksanakan seluruh warga sekolah.

2.2. Teori Khusus

2.2.1. UML (*Unified Modelling Language*)

Sukamto dan Shalahuddin (2013:133), “UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Bahasa pemograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemograman Smalltalk pada awal 1980-an. Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *Unified Modeling Language* (UML) diakomodasi oleh *Object Management Group* (OMG) sehingga sampai saat ini *Unified Modeling Language* (UML) telah



memberikan kontribusinya yang cukup besar dalam metodologi berorientasi objek.

2.2.2. Use Case Diagram

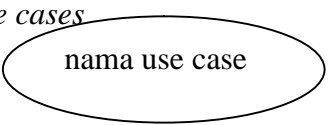

Sukanto dan Shalahuddin (2013:155), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

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*.


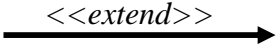

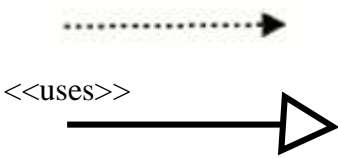
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case* menurut Rosa dan Shalahuddin (2013:156):

Tabel 2.1 Simbol-simbol diagram *use case*

No.	Simbol	Keterangan
1.	<p><i>Use cases</i></p>  <p>nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2.	<p>Aktor / <i>Actor</i></p>  <p>nama aktor</p>	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 <i>actor</i>

Lanjutan Tabel 2.1 Simbol-simbol diagram *use case*


No.	Simbol	Keterangan
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 actor
4.	Ekstensi / <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; mirip dengan prinsip <i>inheritance</i> 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
6.	Menggunakan / <i>include</i> / <i>uses</i>  <i><<include>></i> <i><<uses>></i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> .

2.2.3. Diagram Aktivitas (*Activity Diagram*)

Sukanto dan Shalahuddin (2013:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.


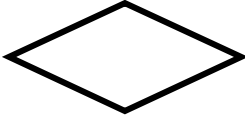

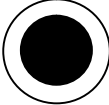
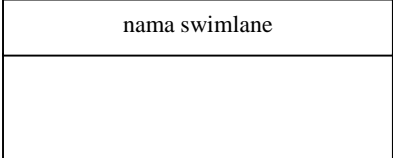
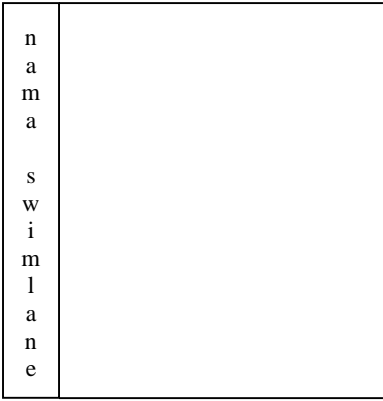
Berikut adalah simbol-simbol yang ada pada diagram aktivitas menurut Rosa dan Shalahuddin (2013:162):

Tabel 2.2 Simbol-simbol diagram aktivitas

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



Lanjutan Tabel 2.2 Simbol-simbol diagram aktivitas

No.	Simbol	Keterangan
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  atau 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi



2.2.4. Diagram Kelas (*Class Diagram*)

Sukanto dan Shalahuddin (2013:141), “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 Rosa dan Shalahuddin (2013:146):

Tabel 2.3 Simbol-simbol diagram kelas

No.	Simbol	Keterangan
1.	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">Kelas</div> <div style="border: 1px solid black; padding: 5px;"> <div style="border-bottom: 1px solid black; padding: 2px 5px;">nama_kelas</div> <div style="border-bottom: 1px solid black; padding: 2px 5px;">+ atribut</div> <div style="padding: 2px 5px;">+ operasi()</div> </div> </div>	Kelas pada struktur system
2.	<div style="display: flex; align-items: center; justify-content: center;"> <div style="margin-right: 10px;">Antarmuka / <i>interface</i></div> <div style="text-align: center;">  <p>nama_interface</p> </div> </div>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek



Lanjutan Tabel 2.3 Simbol-simbol diagram kelas

No.	Simbol	Keterangan
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>
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 antarkelas dengan makna semua-bagian (<i>whole-part</i>)

2.2.5. Diagram Sekuen (*Sequence Diagram*)


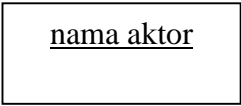

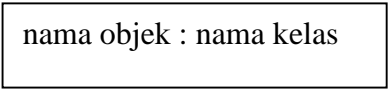

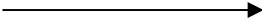

Sukanto dan Shalahuddin (2013:165), “*Sequence diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirim dan diterima antar objek”.

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 interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.



Berikut adalah simbol-simbol yang ada pada diagram sekuen menurut Sukamto dan Shalahuddin (2013:165):

Tabel 2.4 Simbol-simbol diagram sekuen

No.	Simbol	Keterangan
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	<p>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 actor</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p>  <p>nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
5.	<p>Pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>



Lanjutan Tabel 2.4 Simbol-simbol diagram sekuen

No.	Simbol	Keterangan
7.	Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	Pesan tipe return 	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 	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>

2.2.5. Kamus Data

Sukanto dan Shalahuddin (2013:73), Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Kamus data biasanya berisi:

- Nama-nama dari data
- Digunakan pada – merupakan proses-proses yang terkait data
- Deskripsi – merupakan deskripsi data
- Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan
- komponen yang membentuk data



Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan. Simbol-simbol yang ada dalam kamus data menurut Rosa dan Shalahuddin (2013:74) adalah sebagai berikut:

Tabel 2.5 Simbol-simbol dalam kamus data

No.	Simbol	Arti
1	=	Disusun atau terdiri atas
2	+	Dan
3	[]	Baik ...atau...
4	{ } _n	n kali diulang/ bernilai banyak
5	()	Data opsional
6	*...*	Batas komentar

2.3. Model Pengembangan Perangkat Lunak

2.3.1. SDLC (*Software Development Life Cycle*)

Sukanto dan Shalahuddin (2013:13), “SDLC (*Software Development Life Cycle*) adalah proses mengembangkan atau mengubah suatu system perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan system-sistem perangkat lunak”.

Sukanto dan Shalahuddin (2013:28), “model SDLC air terjun (*water fall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*)”.

Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari :

1. Analisis kebutuhan perangkat lunak.
2. Desain perangkat lunak.
3. Pembuatan kode program.
4. Pengujian.
5. Pendukung (*support*) atau pemeliharaan (*maintenance*).



2.4. Teori Program

2.4.1. Basis Data (*Database*)

Berikut ini ada beberapa definisi tentang Basis Data (*Database*) yang disajikan oleh beberapa ahli :

Al Fatta (2007:10), “*Database* yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.

Sutanta (2011:35), “*Database* (basis data) merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar *record* dan rincian data terhadap objek tertentu”.

Dari definisi diatas, dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan dari satu atau lebih organisasi.

2.4.2. *Visual Basic. Net 2010*

Darmayuda (2014:14), “*Visual Basic.NET 2010* tidak jauh berbeda dengan *Visual Basic.NET* sebelumnya atau *Visual Basic 6.0 IDE (Interface Development Environment)*”.

Adapun antarmuka *Microsoft Visual Basic.NET 2010* yaitu :

- a) *Title Bar*, berfungsi untuk menampilkan nama *project* yang aktif atau sedang dikembangkan.
- b) *Menu Bar*, berfungsi untuk pengelolaan fasilitas yang dimiliki oleh *Visual Basic.NET 2010*, sedangkan *Tool Bar*, berfungsi untuk melakukan perintah khusus secara cepat.
- c) *Form*, adalah objek utama berfungsi untuk melakukan objek-objek yang terdapat pada *Toolbox* yang digunakan dalam melakukan perancangan sebuah tampilan program aplikasi.
- d) *ToolBox*, berfungsi untuk menyediakan objek-objek atau komponen yang digunakan dalam merancang sebuah form pada program aplikasi.
- e) *Solution Explorer*, berfungsi untuk menampilkan nama *project*, *file* konfigurasi beserta folder, *file-file* pendukung yang terdapat pada sebuah program aplikasi.



- f) *Properties Window*, berfungsi untuk mengatur *properties-properties* pada objek (*setting object*) yang diletakan pada sebuah *form*.

2.4.3. MySQL

Berikut ini ada beberapa definisi tentang MySQL yang disajikan oleh beberapa ahli :

Sibero (2014:97), “*MySQL* atau dibaca “*My Sekuel*” dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”.

Hidayatullah dan Kawistara (2014:231), “*MySQL* adalah salah satu aplikasi *Database Management System (DBMS)* yang sudah sangat banyak digunakan oleh para pemogram aplikasi web”.

Nugroho (2014:31), “*MySQL* adalah *software* atau program aplikasi *database*, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka”.

Dari definisi diatas, dapat disimpulkan bahwa *MySQL* adalah *database* yang biasa digunakan dalam menyimpan data dari sebuah program seperti *visual basic*, NET dan PHP.