

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama-sama untuk mencapai tujuan (Tata Sutabri 2014).

2.2 Informasi

Informasi pada dasarnya adalah himpunan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas “Informasi yaitu data yang telah diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya (Hartono 2013).

2.3 Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi yang ada untuk menyediakan kepada pihak tertentu dengan laporan – laporan yang diperlukan (Tata Sutabri 2014).

2.4 Pengertian Pariwisata

Istilah pariwisata (tourism) baru muncul di masyarakat kira-kira pada abad ke-18, khususnya sesudah Revolusi Industri di Inggris. Istilah pariwisata berasal dari dilaksanakannya kegiatan wisata (tour), yaitu suatu aktivitas perubahan tempat tinggal sementara dari seseorang, di luar tempat tinggal sehari-hari dengan suatu alasan apa pun selain melakukan kegiatan yang bisa menghasilkan upah atau gaji.

Pariwisata adalah perjalanan dari satu tempat ketempat lain bersifat sementara, dilakukan perorangan atau kelompok, sebagai usaha mencari keseimbangan atau keserasian dan kebahagiaan dengan lingkungan dalam dimensi sosial budaya, dan alam.

Pariwisata adalah suatu aktivitas manusia yang dilakukan secara sadar yang mendapat pelayanan secara bergantian diantara orang-orang dalam suatu negara itu sendiri atau di luar negeri (meliputi pendiaman orang-orang dari daerah lain) untuk mencari kepuasan yang beraneka ragam dan berbeda dengan apa yang dialaminya dimana ia memperoleh pekerjaan tetap (Muljadi, 2012).

2.5 Sistem Informasi Geografis

2.5.1 Pengertian Sistem Informasi Geografis

Geographic Information Sistem (GIS) atau Sistem Informasi Geografis (SIG) diartikan sebagai sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisis dan menghasilkan data bereferensi geografis atau data geospasial, untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan penggunaan lahan, sumber daya alam, lingkungan transportasi, fasilitas kota, dan pelayanan umum lainnya (Indarto, 2013). Sistem Informasi Geografis adalah sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi – informasi geografi. SIG dirancang untuk mengumpulkan, menyimpan, dan menganalisis objek dan fenomena dimana daerah geografi merupakan karakteristik yang penting atau kritis untuk dianalisis (Hussein & Werdiningsih, 2012). Pada dasarnya di dalam istilah sistem informasi geografis yaitu merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografis. Dengan demikian pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami Sistem Informasi Geografis. Dengan melihat unsur-unsur pokoknya, maka jelas sistem informasi geografis merupakan suatu sistem yang menekankan pada unsur “informasi geografis”.

Istilah geografis merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau tertukar sehingga timbul istilah yang ketiga yaitu geospasial. Ketiga istilah ini mengandung pengertian yang sama di dalam konteks SIG. Penggunaan kata “Geografis” mengandung pengertian suatu persoalan mengenai bumi: permukaan dua atau tiga dimensi.

Istilah “Informasi Geografis” mengandung pengertian informasi mengenai tempattempat, pengetahuan mengenai posisi dimana suatu objek terleta, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui. (Triyono dan Wahyudi 2008).

2.5.2 Konsep Dasar Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai Geographic Information System (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan geografis. 40 tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi saja tetapi sudah merambah ke berbagai bidang seperti analisis penyakit epidemik (demam berdarah) dan analisis kejahatan (kerusuhan) termasuk analisis kepariwisataan. Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti query, menganalisisnya serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya (Prahasta, 2010).

2.5.3 Definisi Sistem Informasi Geografis

Istilah geography digunakan karena SIG dibangun berdasarkan pada geografi atau spasial. Objek ini mengarah pada spesifikasi lokasi dalam suatu space. Geographic Information System (GIS) merupakan sistem komputer yang berbasis pada sistem informasi yang digunakan untuk memberikan bentuk digital dan analisis terhadap permukaan geografi bumi. Geografi adalah informasi mengenal permukaan bumi dan semua obyek yang berada di atasnya, sedangkan sistem informasi geografis(SIG) atau dalam bahasa inggris disebut Geographic Information System (GIS) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan). Sistem informasi geografis adalah bentuk sistem informasi yang menyajikan informasi dalam bentuk grafis dengan menggunakan peta sebagai antarmuka. SIG tersusun atas konsep beberapa lapisan (layer) dan relasi (Prahasta, 2010).

2.5.4 Manfaat Sistem Informasi Geografis

Fungsi SIG adalah meningkatkan kemampuan menganalisis informasi spasial secara terpadu untuk perencanaan dan pengambilan keputusan. SIG mampu menyajikan informasi kepada *decision maker* untuk analisis dan penerapan database keruangan (Prahasta, 2010). SIG dapat memberikan kemudahan-kemudahan yang diinginkan. Dengan SIG pengguna dimudahkan untuk melihat fenomena kebumihan dengan perspektif yang berbeda dan lebih baik. SIG dapat mengakomodasi pemrosesan, penayangan, dan penyimpanan data spasial digital bahkan dengan integrasi data yang beragam, mulai dari citra satelit, foto udara, peta bahkan data statistik. SIG juga mengakomodasi dinamika data, pemutakhiran data yang menjadi lebih mudah.

2.5.5 Subsistem Sistem Informasi Geografis

Menurut (Prahasta, 2010), SIG dapat diuraikan menjadi beberapa subsistem sebagai berikut:

1. Data Input

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini bertanggung jawab untuk mengkonversi atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan oleh SIG.

2. Data Output

Subsistem ini menampilkan atau menghasilkan output seluruh atau sebagian basis data baik dalam bentuk softcopy maupun dalam bentuk hardcopy seperti : tabel, grafik, ataupun peta.

3. Data Manajemen

Subsistem ini mengatur baik data spasial maupun atribut kedalam sebuah basis data dengan sedemikian rupa sehingga mudah dipanggil, diupdate dan di-edit.

4. Analisis dan manipulasi data

Subsistem ini menentukan informasi apa saja yang dapat dihasilkan oleh SIG. Selain itu, subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

2.5.6 Cara Kerja Sistem Informasi Geografis

SIG dapat menampilkan dunia nyata pada monitor sebagaimana lembaran peta dapat merepresentasikan dunia nyata di atas kertas. Tetapi, SIG memiliki kekuatan fleksibilitas yang lebih dari pada lembaran pada kertas. Peta merupakan representasi grafis dari dunia nyata, obyek-obyek yang dipresentasikan di atas peta disebut unsur peta atau map features (contohnya adalah laut, taman, kebun, jalan, dan lain lain). Karena peta mengorganisasikan unsur-unsur geografis berdasarkan lokasi-lokasinya. SIG menyimpan semua informasi deskriptif unsur-unsurnya sebagai atribut-atribut didalam basis data. Kemudian, SIG membentuk dan menyimpannya di dalam tabel-tabel database. Dengan demikian, atribut-atribut ini dapat diakses melalui lokasi-lokasi unsur-unsur peta. Dan sebaliknya, unsur-unsur peta juga dapat diakses melalui atribut-atributnya (Prahasta, 2010).

2.5.7 Kemampuan Sistem Informasi Geografis

Sistem informasi geografis mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkannya, menganalisis dan akhirnya memetakan hasilnya: (Prahasta, 2010).

1. Memasukkan dan mengumpulkan data geografis (spasial dan atribut).
2. Mengintegrasikan data geografis.
3. Memeriksa, meng-update (meng-edit) data geografis.
4. Menyimpan atau memanggil kembali data geografis.
5. Mempresentasikan atau menampilkan data geografis.
6. Mengelola, memanipulasi dan menganalisis data geografis.
7. Menghasilkan output data geografis dalam bentuk peta tematik (view dan layout), tabel, grafik (chart) laporan, dan lainnya baik dalam bentuk hardcopy maupun softcopy.

2.6 Peta

Peta merupakan gambaran wilayah geografis, bagian permukaan bumi yang disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Peta dapat digambarkan dengan berbagai gaya, masing-masing menunjukkan permukaan yang berbeda untuk subjek yang sama untuk men-visualisasikan dunia dengan mudah, informatif dan fungsional. Peta berbasis komputer (digital) lebih serba guna dan dinamis karena bisa menunjukkan banyak view yang berbeda dengan subjek yang sama. Peta ini juga memungkinkan perubahan skala, animasi gabungan, gambar, suara, dan bisa terhubung ke sumber informasi tambahan melalui internet. Peta digital dapat diupdate ke peta tematik baru dan bisa menambahkan detail informasi geografi lainnya (Zainuddin, 2010).

2.7 Google Maps

Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dapat ditemukan di <http://maps.google.com>., yang menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia. Dan juga menawarkan perencanaan rute dan pencari letak bisnis di U.S., Kanada, Jepang, Hong Kong, Cina, UK, Irlandia (hanya pusat kota) dan beberapa bagian Eropa. Google Maps masih berada dalam tahap beta. Melalui fitur Google Maps, pengguna internet dapat browsing informasi grafis berikut:

1. Satellite Map

Pengguna dapat menikmati gambar satelit planet bumi. Pengguna juga dapat menikmati foto satelit lebih detail lengkap dengan cara zooming pada bagian peta yang diinginkan. Contoh satellite map disajikan pada Gambar 2.1



Gambar 2.1 Satellite Map.

2. Hasil Pencarian Integrasi

Mencari lokasi, bisnis, peta buatan pengguna dan real estate.

3. Draggable Maps

Peta digital mapping yang draggable (bisa digeser) dengan bantuan mouse.

4. Terrain Maps (Peta Topografi)

Terrain Maps menyediakan informasi fitur peta fisik atau peta topografi yang biasa disediakan buku peta Atlas, yang ditunjukkan pada Gambar 1.2



Gambar 2.2 Terrain Maps.

5. Earth Map

Earth Map menyediakan informasi peta bumi yang menampakkan bumi secara utuh dan bila di-zoom terlihat awan yang menyelimuti bumi beserta pulau dan lautan yang tampak nyata dari ketinggian. Tampilan earth map ditunjukkan pada Gambar 2.3



Gambar 2.3 Earth Map.

6. My Location

Dengan fitur ini pengguna dapat mengetahui letak dimana lokasi dari pengguna tersebut.

2.8 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment (IDE)* untuk *platform* Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android Studio bersifat *free* dibawah *Apache License 2.0*. Berbasiskan *JetBrainns' IntelliJ IDEA*, Android Studio didesain khusus untuk *Android Development* yang kini sudah bisa di *download* untuk *Windows*, *Mac OS X*, dan *Linux*. Android Studio menyediakan alat untuk pengujian, dan mempublikasikan tahap proses *development*, serta lingkungan *development* terpadu untuk membuat aplikasi bagi semua perangkat Android.

Lingkungan development menyertakan kode *template* dengan kode contoh untuk fitur aplikasi umum, alat pengujian dan kerangka kerja yang banyak, dan sistem pembangunan yang fleksibel (Android Developer Fundamental Course Concepts, 2016).

2.9 Google Maps API

Google Maps API adalah suatu library yang berbentuk JavaScript. Cara membuat Google Maps untuk ditampilkan pada suatu web atau blog membutuhkan pengetahuan mengenai HTML serta JavaScript, serta koneksi internet yang sangat stabil. Dengan menggunakan Google Maps API, dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga programmer dapat fokus hanya pada data-data yang ditampilkan. Dengan kata lain, programmer hanya membuat suatu data sedangkan peta yang ditampilkan adalah milik Google sehingga programmer menampilkan peta lokasi. Dalam pembuatan program Google Map API menggunakan urutan sebagai berikut:

1. Memasukkan Maps API JavaScript ke dalam HTML.
2. Membuat element div dengan nama map_canvas untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan properti-properti pada peta.
4. Menuliskan fungsi JavaScript untuk membuat objek peta.
5. Meng-inisiasi peta dalam tag body HTML dengan event onload

2.10 Mobile Android

Menurut Jubilee Enterprise (2015:1) Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Menurut Muhammad Sadeli (2014:2) Android merupakan sistem operasi berbasis Linux yang digunakan untuk perangkat *mobile* (Smartphone) atau pun perangkat tablet (PDA). Sedangkan Menurut Yosef Murya (2014:3) Android sistem operasi berbasis Linux yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar smartphone dan komputer tablet.

2.11 *Android Software Development Kit (SDK)*

Android SDK adalah *tool API (application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang release oleh Google.

Saat ini di sediakan *Android SDK (Software Development Kit)* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada platform *Android* menggunakan bahasa pemrograman java (Safaat H 2014: 5).

2.12 *Android Development Tools (ADT)*

Android Development Tools (ADT) adalah plugin *Eclipse*. Yang didesain untuk *IDE Eclipse* yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan *IDE Eclipse*. (Safaat H 2014: 6).

2.13 *Android Virtual Device (AVD)*

AVD yang merupakan emulator untuk menjalankan program aplikasi android yang kita buat, *AVD* ini nantinya yang kita jadikan sebagai tempat test dan menjalankan aplikasi android yang kita buat, *AVD* berjalan di *virtual Machine* (Safaat H 2014: 4).

2.14 *Unified Modeling Language (UML)*

2.14.1 *Pengenalan Unified Modeling Language (UML)*

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 merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak

terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek (Rosa A.S dan M. Shalahudin, 2014:133)

2.14.2 Sejarah *UML*

Bahasa pemrograman 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 pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi- metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *UML* diakomodasi oleh *OMG* sehingga sampai saat ini *UML* telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa A.S dan M. Shalahudin, 2014:138).

2.14.3 Diagram UML

Pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut. (Rosa A.S dan M. Shalahudin 2014:140)

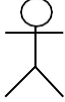



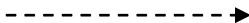
1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behavior diagram* terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

2.14.4 Use Case Diagram

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 actor 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 (Rosa dan M. Shalahudin 2014:155).

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol-simbol *use case* diagram.






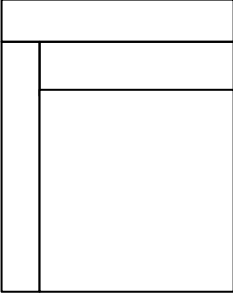
No	Simbol	Keterangan Fungsi
1.	Aktor 	Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.
2.	<i>Use Case</i> 	<i>Use Case</i> adalah deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
3.	Asosiasi 	Asosiasi adalah apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4.	Generalisasi 	Generalisasi adalah hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya atau sebaliknya dari bawah ke atas.
5.	<i>Defendancy</i> 	<i>Defendancy</i> (ketergantungan) adalah hubungan dimana perubahan yang terjadi pada suatu elemen defenden (mandiri) akan mempengaruhi elemen yang bergantung padanya (<i>independen</i>).

2.14.5 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan M. Shalahudin 2014:161)

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol-simbol *activity diagram*.

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <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 oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

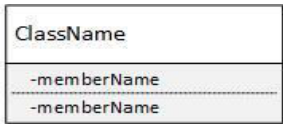


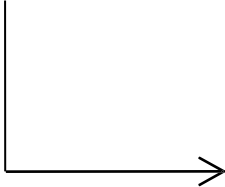

2.14.6 Class Diagram

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 *method* atau operasi (Rosa dan M. Shalahudin 2014:141). Berikut penjelasan atribut dan *method* :

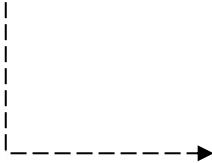
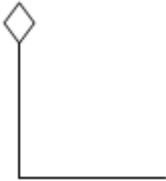
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3 Simbol-simbol *class diagram*.

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)

Tabel 2.3 (lanjutan)

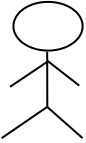
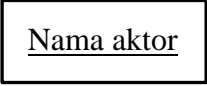

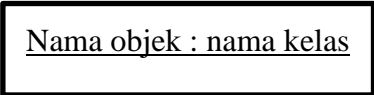
6	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7	Agrpasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

2.14.7 *Sequence Diagram*


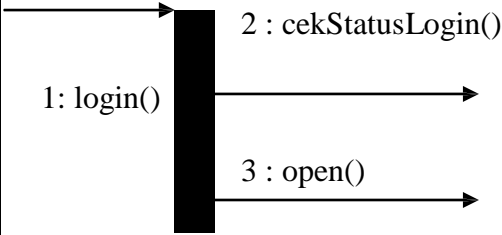
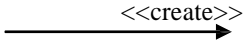
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario 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 interaksinya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak (Rosa dan M. Shalahudin 2014:165)

Berikut adalah simbol-simbol yang ada pada diagram sekuen :


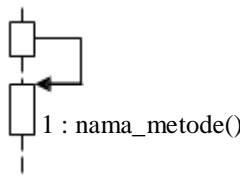

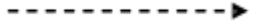
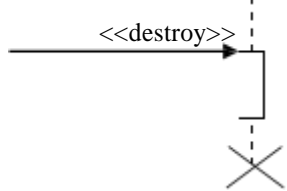
Tabel 2.4 Simbol-simbol *sequence diagram*.

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>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, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>

Tabel 2.4 (lanjutan)

4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif</p>
5	<p>Pesan tipe <i>create</i></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

Tabel 2.4 (lanjutan)

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>  <p>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</p>
7	<p>Pesan tipe <i>send</i></p> <p>1 : masukkan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
8	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>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</p>
9	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i></p>

2.15 PHP (*Hypertext Preprocessor*)

Abdul Kadir (2013:17), PHP adalah bahasa pemrograman yang ditunjukkan untuk kepentingan pembuatan aplikasi web. Sebagai bahasa pemrograman untuk web, PHP sebenarnya bukanlah satu-satunya, tetapi termasuk yang populer.

PHP memungkinkan pembuatan aplikasi web yang dinamis, dalam arti, dapat membuat halaman web yang dikendalikan oleh data. Dengan demikian, perubahan data akan membuat halaman web ikut berubah tanpa harus mengubah script atau kode yang menyusun halaman web.

2.16 JSON (*Javascript Object Notation*)

Adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

JSON terbuat dari dua struktur:

- a. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- b. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini.

2.17 Basis Data

Dalam pembuatan aplikasi, para pembuat aplikasi atau programmer menggunakan basis data yang digunakan untuk pengolahan data atau penataan file-file yang ada dan digunakan kembali sesuai dengan kebutuhan aplikasi tersebut. Basis data merupakan tempat pengolahan informasi yang sangat penting dalam upaya menciptakan suatu aplikasi yang terintegrasi.

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan (Rosa dan Shalahuddin 2015),.

Basis data (database) merupakan kumpulan data yang saling berhubungan atau punya relasi (Yakub dan Hisbanarto 2014).

Dapat ditarik kesimpulan bahwa basis data merupakan kumpulan data yang dapat diolah menjadi informasi yang berguna bagi penggunaannya yang tersimpan dan dapat digunakan di dalam komputer.

2.18 MySQL

MySQL adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan (Edy Winarno dan Ali Zaki, 2014:102).

Ichsan Risnandar, Hanafi S.S, Teguh Nugraha. (2013:92), *MySQL* merupakan basis data yang bersifat *open source* sehingga banyak digunakan di dunia. Ada dua cara untuk menggunakan *MySQL* yaitu:

- a. Menggunakan *console/command line* atau *PHP*. Kita harus hafal perintah-perintah dalam *MySQL*. *Interface* yang digunakan hanya *text-based*.
- b. Menggunakan *PHPMysqlAdmin*, dapat diakses dengan alamat <http://alamatwebsite/phpmyadmin>. apabila digunakan *localhost* maka alamat <http://localhost/phpmyadmin> atau di 127.0.0.1/phpmyadmin.

Penggunaan *PHPMysqlAdmin* lebih mudah digunakan karena menggunakan *interface* yang lebih mudah dipahami.

2.19 Metodologi Pengembangan Sistem

Proses pengembangan mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan dalam mengembangkan sistem-sistem perangkat lunak sebelumnya. (Rosa A.S M Shalahudin, 2014). Adapun tahapan-tahapan yang terdapat dalam metode *waterfall* yaitu :

- a. Perencanaan (*planning*), mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resource*) yang digunakan untuk memperoleh proses.
- b. Analisis (*analysis*), menganalisis kebutuhan pemakaian sistem perangkat lunak (*user*) dan mengembangkan kebutuhan user. Membuat dokumen kebutuhan fungsional.
- c. Perancangan (*design*), langkah yang fokus pada desain pembuatan program perangkat lunak. Representasi antarmuka, dan prosedur pengkodean.
- d. Pengkodean (*coding*), desain harus ditransmisikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.
- e. Implementasi (*implementation*), pengujian perangkat lunak dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sama dengan yang diinginkan