

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh Anna Fitria Ummawati (2018) yang berjudul “*Aplikasi Pengenalan Pariwisata Pulau Bawean Berbasis Android*”. Penelitian ini menghasilkan suatu aplikasi Sistem Aplikasi Pengenalan Pariwisata Pulau Bawean Berbasis Android dengan menggunakan App Inventor 2 digunakan untuk membantu dan memberikan kemudahan dalam melakukan pengenalan adat dan budaya kepada pengguna aplikasi.

Penelitian lain yang dilakukan oleh Fani Andi (2017) dengan judul “*Aplikasi Pengenalan Pariwisata Kabupaten Bojonegoro Berbasis Android*”. Penelitian ini menghasilkan Aplikasi pengenalan pariwisata Kabupaten Bojonegoro menjadi media yang bisa memberi informasi bagi user. Aplikasi Pengenalan pariwisata Bojonegoro ini berjalan pada platform android. Aplikasi mobile dapat berfungsi pada android (4.0) Jelly Bean 70 % sedangkan android (4.4) kitkat dan android (5.0) Lollipop sukses 100%.

Penelitian lain yang dilakukan oleh Wira Shilviana Hanum, Dan Aries Saifudin (2019) dengan judul “*Rancang Bangun Aplikasi Panduan Pariwisata Di Kabupaten Banyuwangi Mobile Berbasis Android*”. Aplikasi Sistem ini aplikasi panduan Pariwisata di Kabupaten Banyuwangi memudahkan masyarakat dan wisatawan dalam mendapatkan pengetahuan tentang informasi Pariwisata di Banyuwangi yang lebih cepat dengan media smartphone android dan informasi pariwisata yang tidak hanya sebatas wisata populer di Banyuwangi.

Penelitian lain yang dilakukan oleh Hera Wulanratu Wulur, dkk (2015) dengan judul “*Aplikasi Virtual Tour Tempat Wisata Alam Di Sulawesi Utara*”. Penelitian ini bertujuan mengembangkan sebuah aplikasi interaktif yang dapat menampilkan informasi secara visual dari suatu tempat wisata alam di Sulawesi Utara. Tools yang digunakan dalam aplikasi ini adalah PT Gui, Eclipse dan Google Maps. Dalam aplikasi ini, pengguna bisa melihat keadaan 360 tempat wisata alam di 10 spot tempat wisata yang dibuat dengan teknik immersive photography. Dengan merepresentasikan informasi dalam bentuk gambar panorama 360 memudahkan pengguna untuk menampilkan informasi secara visual dari suatu tempat wisata alam di Sulawesi Utara.

Berdasarkan isi beberapa penelitian diatas adapun yang membedakan dari penelitian ini adalah membangun suatu aplikasi pengenalan objek wisata indah dengan memanfaatkan teknologi berbentuk gambar, teks dan *voice* berbasis android.

II.2. Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu melayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.3. Wisata

Wisata ialah merupakan suatu perjalanan yang dilakukan oleh seseorang/sekelompok lebih orang yang memiliki tujuan, antara lain: keinginan dalam mendapatkan kenikmatan & memenuhi rasa ingin tahu terhadap sesuatu, baik yang berkaitan dengan kegiatan olah raga, kesehatan, kegiatan konvensi, urusan keagamaan dan keperluan usaha lain nya. Wisatawan adalah merupakan satu orang atau beberapa orang (grup), yang melakukan suatu bentuk

perjalanan wisata. Jika mereka tinggal kurang dari 24 jam disebut pelancong atau pengunjung (Haris Febriyanto Ramadhan, dkk; 2019).

II.4. *Android*

Android adalah sistem operasi menggunakan Linux yang dirancang untuk perangkat seluler seperti telepon pintar (smartphone) dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc.. dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Android ialah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah lisensi apache. Kode dengan lisensi terbuka dalam lisensi perizinan Android memungkinkan perangkat lunak ini untuk dimodifikasi secara bebas dan didistribusikan oleh pembuat perangkat, operator nirkabel dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat (Winda Angela Hamka, Abdu Gani; 2016).

II.5. *Android Studio*

Android Studio adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Aplikasi *Android* yang dikembangkan menggunakan *Java* dan mudah menyesuaikan ke platform baru. *Android* terdiri dari satu tumpukan yang lengkap, mulai dari *boot loader*, *device driver*, dan fungsi-fungsi pustaka, hingga perangkat lunak API (*Application Programming Interface*), termasuk aplikasi SDK (*Software Development Kit*). *Android* bukanlah satu perangkat tertentu, melainkan sebuah *platform* yang dapat digunakan dan diadaptasikan untuk mendukung berbagai konfigurasi perangkat keras. Sistem operasi yang mendasari *Android* merupakan lisensi di bawah naungan GNU, *General Public License Versi 2(GPLv2)*, yang biasa dikenal dengan istilah *Copy left*. Istilah *copy left* ini merupakan lisensi yang setiap perbaikan oleh

pihak ketiga harus terus jatuh di bawah terms. Distribusi *Android* berada di bawah *lisensi Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua atau seterusnya. Pengembang aplikasi *Android* diperbolehkan untuk mendistribusikan aplikasi mereka di bawah skema lisensi apapun yang mereka inginkan. (Ganda Yoga Swara, dkk; 2017) .

II.6. Database

Basis data (database) diartikan sebagai markas atau gudang data, tempat bersarang atau berkumpul data. Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan dalam pengambilan data kembali. Adapun tujuan basis data diantaranya sebagai efisiensi yang meliputi speed, space&Accuracy, menangani data dalam jumlah besar, kebersamaan pemakaian, dan meniadakan duplikasi (Nofyat, dkk; 2018) .

II.7. Firebase Realtime Database

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh *Google*. Firebase sendiri merupakan solusi yang ditawarkan oleh *Google* untuk mempermudah pekerjaan *Mobile Apps Developer*. Dengan adanya *Fire-base*, *apps developer* tentu saja dapat fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend *Firebase Realtime Database* sendiri merupakan basis data berbasis cloud NoSQL yang menyinkronkan data di semua klien secara *realtime*, dan menyediakan fungsionalitas offline Data yang diinputkan disimpan ke dalam database *Realtime* sebagai JSON. Semua klien yang terhubung dan saling berbagi dalam satu waktu, secara otomatis akan menerima pembaruan dengan data terbaru (Kurniawati, dan Lukman Bachtiar, 2020).

II.8 . UML (*Unified Modeling language*)


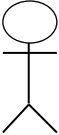

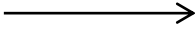
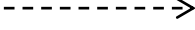
Unified Modeling Language (UML), adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

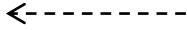
UML merupakan keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Pemodelan *visual* adalah salah satu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata yang berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (*costumer*, ahli dibidangnya, analisis, desainer dan lain-lain). Serta didefinisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik dan objek-objek dalam sistem dan antar sistem itu sendiri. (Munawar; 2018: 49).

1. *Use Case Diagram*

Use Case diagram 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. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* diagram yaitu:

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>Use Case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>Use Case</i>.</p>
	<p>Asosiasi antara aktor dan <i>Use Case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>Use Case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>Use Case</i> lain (<i>required</i>) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>




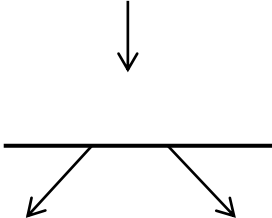
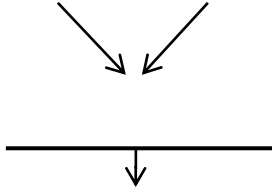
	<i>Extend</i> , merupakan perluasan dari <i>Use Case</i> lain jika kondisi atau syarat terpenuhi.
---	---

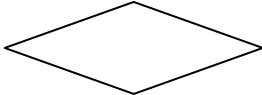
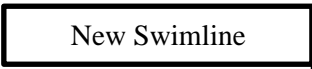
(Sumber : Munawar; 2018: 89)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan *Workflow* (Aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity diagram*, yaitu :

Tabel II.2. Simbol Diagram Aktivitas

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.

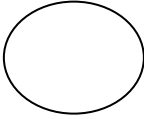
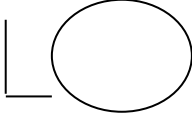
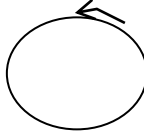
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
	<i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa.


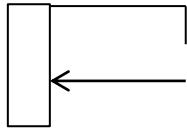
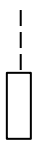
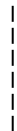
(Sumber : Munawar ; 2018 : 137)

3. Diagram Urutan (*Sequence Diagram*)

Diagram urutan menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam diagram urutan, yaitu

Tabel II.3. Simbol Diagram Urutan

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .

	
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Munawar ; 2018 : 186)

1. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram merupakan diagram statis dari suatu aplikasi. *Class Diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak (Munawar ; 2018 : 101).

Tabel II.4. Simbol *Class Diagram*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Munawar ; 2018 : 101)

