



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB 2

LANDASAN TEORI

2.1 Definisi Pariwisata (*Travel*)

Menurut Richard Sihite dalam Marpaung dan Bahar (2000:46-47) menjelaskan definisi pariwisata sebagai berikut : pariwisata adalah suatu perjalanan yang dilakukan orang untuk sementara waktu, yang diselenggarakan dari suatu tempat ke tempat lain meninggalkan tempatnya semula, dengan suatu perencanaan dan dengan maksud bukan untuk berusaha atau mencari nafkah ditempat yang dikunjungi, tetapi semata-mata untuk menikmati kegiatan pertamsyaan dan rekreasi atau untuk memenuhi keinginan yang beraneka ragam.

Pariwisata merupakan salah satu kegiatan yang menjadi pilihan masyarakat luas pada saat ada waktu luang, dengan berkunjung ke area rekreasi suatu daerah tertentu didalam ataupun di luar negeri. Kegiatan ini setiap tahunnya terus berkembang dan menjadi besar, banyak wahana rekreasi yang muncul seiring dengan kebutuhan masyarakat untuk menyisihkan waktu luang untuk bersantai. Pariwisata menjadi salah satu industri yang memiliki dampak yang sangat luas termasuk untuk industri perhubungan, kerajinan dan kreatif. Pariwisata Indonesia sendiri memiliki pertumbuhan 7,2 persen pertahun, angka ini sangat tinggi jika di

bandingkan dengan rata-rata pertumbuhan pariwisata dunia yang hanya 4,7 persen pertahun (Badan Pusat Statistik, 2016).

2.2 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan *editor* kode *IntelliJ* dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas developer saat membuat aplikasi Android, misalnya:

1. Sistem pembuatan berbasis *Gradle* yang fleksibel.
2. *Emulator* yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. *Instant Run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.

Android studio menjadi pilihan banyak *developer* untuk membuat aplikasi dikarenakan android studio memiliki *layout xml* secara visual yang jauh lebih baik dibandingkan dengan IDE lain (Hafizh Herdi, 2014). Bagi para developer baru, mereka dapat belajar banyak hal mengenai pengembangan aplikasi android lewat *website* resmi android studio, maupun lewat forum komunitas yang ada di internet.

2.3 Google API

Google API dapat dikatakan bagian dari *Framework* Google. Google menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang web maupun aplikasi desktop untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti misalnya: *AdSense, Search Engine, Translation* maupun *YouTube*.

API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang kita buat dengan fungsi-fungsi yang dikerjakan. Misalnya dalam hal ini Google API berarti kode program (yang disederhanakan) yang dapat kita tambahkan pada aplikasi atau web kita untuk mengakses/menjalankan/memanfaatkan fungsi atau fitur yang disediakan Google. Misalnya saja kita bisa menambahkan fitur *Google Map* pada website yang developer buat menggunakan API dari Google Map yang sudah di siapkan.

Google API dapat dipelajari langsung melalui *Google Code*. Melalui *Google Code* kita dapat belajar tentang Google API dan dapat mengimplementasikan pada aplikasi web atau website yang kita kembangkan.

Ada banyak API yang disediakan oleh Google, beberapa diantaranya adalah:

1. Language API: untuk memanfaatkan fitur translation yang dimiliki Google.
2. Earth API: memanfaatkan fitur yang ada pada Google Earth.
3. Javascript API.

4. Maps API: memanfaatkan fitur yang ada pada Google Maps.
5. Search API: memanfaatkan fitur pencarian pada Google Search.
6. Visualization API: membuat grafik maupun chart dengan Google API.
7. YouTube API: memanfaatkan fitur yang ada pada YouTube misalnya untuk pencarian video.

2.4 Google Trips

Google Trips merupakan sebuah aplikasi perencanaan perjalanan wisata yang dikembangkan oleh Google Inc. yang telah terpasang lebih dari 1 juta smartphone diseluruh dunia. Google Trips menjadi salah satu aplikasi yang penting bagi turis modern yang membutuhkan *planner* dan *guide* untuk melakukan perjalanan ketempat-tempat yang baru. Google Trips masih terus mengembangkan wilayah jangkauan yang nantinya dapat di eksplorasi oleh wisatawan, dengan demikian wisatawan memiliki banyak pilihan untuk mengeksplorasi tempat-tempat yang baru.



Gambar 2.1 Google trip (Sumber : <https://www.theverge.com>)

Google Trips menjadi salah satu aplikasi planner yang sangat lengkap karena memiliki banyak sekali fungsi, diantaranya:

1. *Automatic Trip Organization*
2. *Bundled Reservations*
3. *Day Plans*
4. *Nearby Attraction*
5. *Things To Do*
6. *Offline Access*

2.5 Apiary.io (Oracle Database)

Apiary.io merupakan salah satu platform untuk mendesain, mengembangkan dan mendokumentasi API. Apiary merupakan perusahaan yang dibeli oleh oracle pada awal bulan januari 2017, perusahaan ini bergerak dibidang *cloud based aplication*. Apiary telah banyak membantu perusahaan dalam hal membuat, megembangkan, dan mendokumentasi API sehingga membantu perusahaan untuk dapat memperbaiki dari sistem yang digunakan.



Gambar 2. 2 Apiary.io (Sumber : <https://help.apiary.io>)

2.6 Software Development Kit

Software development kit (SDK) adalah satu set alat pengembangan perangkat lunak yang memungkinkan developer untuk pembuatan aplikasi untuk software tertentu, kerangka kerja perangkat lunak tertentu, platform perangkat keras, sistem komputer, video game console, sistem operasi, seperti halnya platform.

SDK merupakan sesuatu yang sederhana seperti sebuah Application Programming Interface (API) dalam bentuk beberapa *file* ke antarmuka untuk bahasa pemrograman tertentu atau mencakup perangkat keras yang canggih untuk berkomunikasi dengan sistem tertentu. Software ini termasuk alat bantu debugging dan utilitas lain sering disajikan dalam integrated development environment (IDE) didalamnya terdapat juga contoh code program pendukung lainnya yang berguna untuk menjelaskan poin dari kode program utama.

SDK sangat penting bagi developer karena sangat membantu dan berguna bagi kelangsungan pembuatan sebuah sistem maupun aplikasi. SDK didalam android studio memiliki berbagai macam fitur dan kegunaan, setiap developer pasti menggunakan SDK untuk mempermudah mereka untuk mendapatkan hasil fitur yang baik. SDK menjadi hal yang sangat penting didalam pengembangan sebuah sistem maupun aplikasi.

2.7 Flowchart

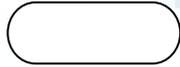
Menurut Sitorus (2015) flowchart menggambarkan urutan logika dari suatu proses pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah dalam simbol-simbol tertentu.

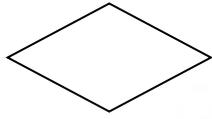
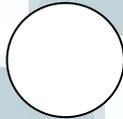
Menurut Sitorus terdapat beberapa aturan dalam perancangan flowchart antara lain:

1. Flowchart digambarkan dengan orientasi dari atas ke bawah dan dari kiri ke kanan.
2. Setiap kegiatan/proses dalam flowchart harus dinyatakan secara eksplisit.
3. Setiap flowchart harus di mulai dari satu *start* dan berakhir pada satu atau lebih terminal akhir/terminator/halt state.
4. Gunakan *connector* dan *off-page connector state* dengan label yang sama untuk menunjukkan keterhubungan antar *path* logika algoritma yang terputus/terpotong, misalnya sebagai akibat pindah/ganti halaman.

Terdapat beberapa simbol flowchart antara lain:

Table 2. 1 *Flowchart*

No.	Simbol	Nama	Fungsi
1.		<i>Terminal</i>	Menyatakan permulaan atau akhir suatu program.
2.		<i>Input</i> / <i>Output</i>	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis pelatannya.
3.		<i>Process</i>	Menyatakan suatu tindakan yang dilakukan oleh komputer.

4.		<i>Decision</i>	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya / tidak.
5.		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lain dalam halaman yang sama.
6.		<i>Offline connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8.		<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis di kartu.

2.8 Entity Relationship Diagram (ERD)

Menurut (Lubis, 2016) *entity relationship diagram* merupakan suatu permodelan berbasis pada persepsi dunia nyata yang mana terdiri dari kumpulan objek dasar yang disebut entitas dan hubungan antar objek-objek tersebut dengan perangkat konseptual dalam bentuk diagram.

Menurut (Mulyani, 2016) *entity relationship diagram*, merupakan deskripsi hubungan antara entitas dari kategori yang berbeda atau sama. Terdapat beberapa *relationship* dan *cardinality* dalam ERD antara lain:

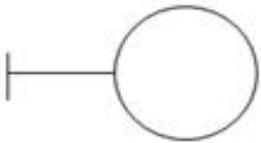
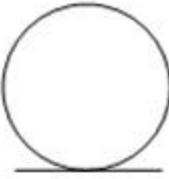
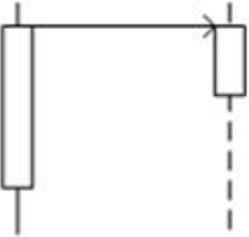
Table 2. 2 ERD

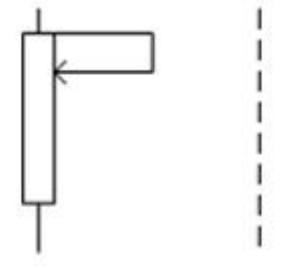
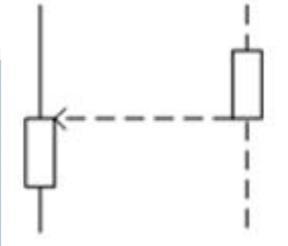
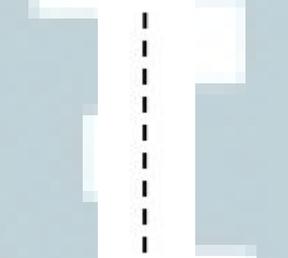
Cardinality	Minimum	Maksimum	Notasi
1	1	1	—+ (line with vertical bar and crossbar)
0 atau 1	0	1	—○+ (line with open circle, vertical bar, and crossbar)
1 atau >1 (many)	1	>1 (many)	— > (line with vertical bar and three diagonal lines)
0,1 atau >1 (many)	0	>1 (many)	—○ > (line with open circle, vertical bar, and three diagonal lines)
>1 (many)	>1 (many)	>1 (many)	—> (line with three diagonal lines)

2.10 Sequence Diagram

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display* yang berupa pesan/*message*. Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/*event* untuk menghasilkan *output* tertentu. Sequence Diagram diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Berikut ini merupakan notasi dasar didalam sequence diagram:

Table 2. 3 Sequence diagram

No.	Simbol	Nama	Fungsi
1.		<i>Actor</i>	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem
2.		<i>boundary</i>	Menggambarkan interaksi antara satu atau lebih actor dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.
3.		<i>Control</i>	Mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
4.		<i>Entity</i>	Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).
5.		<i>Object Message</i>	Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.

6.		<i>Offline connector</i>	Mengambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
7.		<i>Return Message</i>	Mengambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.
8.		<i>Lifeline</i>	Eksekusi obyek selama sequence (<i>message</i> dikirim atau diterima dan aktifasinya).

2.11 Penelitian Sebelumnya

Terdapat beberapa penelitian yang dilakukan oleh mahasiswa kampus lain dengan menggunakan metode penelitian yang sama dengan objek penelitian yang berbeda. Salah satu penelitian yang menjadi acuan penulis yaitu berjudul “Perancangan aplikasi penjualan berbasis web dengan metode prototyping pada CV Khatulistiwa”. Penelitian tersebut ditulis oleh 3 orang mahasiswa dari fakultas rekayasa industry, Telkom University yaitu Izharyan Iqbal, R. Wahyu Witjaksono, dan M. Teguh Kurniawan. Penulisan tersebut dilatar belakangi karena ingin mengimplementasikan kegiatan penjualan perusahaan menjadi online, sehingga dapat membantu menjangkau calon customer lain dengan lebih baik. Hasil

kesimpulan dari feedback prototype I yang didapatkan yaitu, desain interface yang telah dibuat cukup dipahami, fitur yang ada pada prototype I sudah cukup spesifik, dan tampilan dari aplikasi masih kurang menarik. Dari evaluasi tersebut dapat dikatakan user masih kurang puas tetapi secara fungsional sudah memenuhi kebutuhan dasar dan dapat dilanjutkan pengembangannya ketahap prototype II.

