

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Dalam penyusunan penelitian ini, penulis mengambil referensi dari beberapa penelitian sebelumnya untuk menggali informasi tentang teori yang berkaitan dengan judul yang digunakan penulis sebagai landasan teori. Adapun penelitian sebelumnya yang menjadi tinjauan pustaka dalam penulisan ini adalah sebagai berikut :

(Ependi, Panjaitan, & Syakti, 2020), melakukan penelitian dengan judul “Pengembangan Aplikasi *Mobile Travel Guide* Pada Provinsi Sumatra Selatan”. Penelitian ini bertujuan untuk mengembangkan aplikasi *mobile* untuk menyediakan informasi pariwisata yang ada di provinsi Sumatra Selatan. Proses pengembangan aplikasi ini menggunakan *Mobile D*, dengan tahapan *explore*, *initialize*, *productionrize*, *stabilize*, *system test* dan *fix*. Hasil dari penelitian ini dapat ditunjukkan dengan instrument *heuristic evaluation* yang mendapatkan nilai rata-rata 0.2 dan *system usability scale* mendapatkan nilai rata-rata 84.75, nilai tersebut menunjukkan bahwa aplikasi *mobile* yang dihasilkan tidak memiliki masalah *usability* dan dapat diterima dengan mudah oleh pengguna.

(Djunaedi, 2011), melakukan penelitian dengan judul “Konsep Rancang Bangun Aplikasi *e-Travelling* Berbasis *Customer Relationship Management* Menggunakan *Rich Internet Application* (Studi Kasus Bayu Buana *Travel Services*)”. Tujuan dari penelitian ini adalah untuk membuat contoh rancangan

sebuah situs *e-travel* yang mengimplementasikan *Customer Relationship Management* dalam menjaga hubungan baik dengan *customer*. Hasil dari penelitian ini merupakan sebuah sistem informasi yang dapat mencatat dan mengikuti semua aspek dalam penjualan, menangani keluhan atau komplain pelanggan, dan menyediakan dukungan pelanggan yang memadai. Kesimpulan dari penelitian ini menerangkan bahwa *Rich Internet Application* dapat mempermudah dalam pencarian data dengan sumber yang lebih luas yaitu internet tanpa harus memasukkan data tersebut ke dalam memori *web e-travel*.

(Wijaya, Handojo, & Andjawirawan, 2015), melakukan penelitian dengan judul “Aplikasi Sistem Perancangan *Travel* Berbasis Android Pada PT. Kirana Anugerah Terindah”. Pada penelitian ini bertujuan untuk membantu driver dalam mencari alamat dan menentukan rute terbaik, serta membantu perusahaan dalam mengembangkan sisi teknologi yang berupa pemesanan menggunakan android. Hasil dari penelitian tersebut berupa aplikasi yang berjalan pada *mobile device* bersistem operasi android yang dibagi menjadi aplikasi *admin*, *customer*, dan armada. Kesimpulan yang didapat pada penelitian ini adalah sistem pemesanan pada Kirana *Travel* sudah berjalannya dengan baik karena dengan adanya aplikasi yang dibuat, serta *upload file* berjalan dengan baik hanya saja untuk lebih memaksimalkan kinerja dari *upload file* disarankan berada di tempat yang memiliki koneksi internet yang baik.

2.2 Teori Terkait

2.2.1 Dart

Bahasa pemrograman Dart merupakan bahasa pemrograman *general-purpose* yang dirancang oleh Lars Bak dan Kasper Lund. Bahasa pemrograman ini dikembangkan sebagai bahasa pemrograman aplikasi yang dapat dengan mudah untuk dipelajari dan disebar. Bahasa pemrograman besutan Google ini dapat digunakan untuk mengembangkan berbagai macam *platform* termasuk di dalamnya adalah *web*, aplikasi *mobile*, *server*, dan perangkat yang mengukung teknologi *Internet of Things*. Bahasa pemrograman tersebut dapat digunakan untuk mengembangkan aplikasi untuk dijalankan pada berbagai macam peramban *modern*. Dart juga dapat digunakan untuk mengembangkan aplikasi dari *codebase* tunggal menjadi aplikasi Android dan iOS. Bahasa pemrograman Dart dapat digunakan secara bebas oleh para *developer*, karena bahasa ini dirilis secara *open-source* oleh Google dibawah lisensi BSD. Bahasa pemrograman Dart merupakan bahasa pemrograman berbasis class dan berorientasi terhadap objek dengan menggunakan sintaks bahasa pemrograman C.

Bahasa ini dikenalkan oleh Google sebagai pengganti bahasa pemrograman JavaScript, akan tetapi secara opsional bahasa ini dapat dikompilasi ke dalam JavaScript dengan menggunakan *dart2js* compiler. Sedikit berbeda dengan bahasa pemrograman JavaScript yang bertipe statis, bahasa pemrograman Dart merupakan bahasa pemrograman bertipe dinamis.

Adapun kelebihan bahasa pemrograman Dart antara lain adalah sebagai berikut :

1. Fleksibel

Salah satu kelebihan bahasa pemrograman Dart adalah bahasa pemrograman tersebut termasuk ke dalam bahasa bertipe dinamis. Bahasa pemrograman ini dapat dikompilasi ke dalam bahasa pemrograman JavaScript dengan compiler yang sudah disertakan di dalamnya. Bahasa pemrograman ini dikembangkan untuk mudah digunakan dalam pengembangan, sesuai dengan pengembangan aplikasi modern, dan memiliki implementasi berkinerja tinggi. Bahkan, bahasa pemrograman ini dapat digunakan juga sebelum dikompilasi.

Dart VM menawarkan kemampuan untuk menjalankan secara langsung kode sumber tanpa perlu dikompilasi terlebih dulu. Bahasa pemrograman ini juga dapat langsung digunakan pada peramban Chrome tanpa perlu dikompilasi. Bahasa pemrograman Dart mendukung banyak arsitektur, termasuk di dalamnya IA-32, X64, MIPS, ARMv5TE, ARMv6, ARMv7, dan arsitektur ARM64. Bahasa pemrograman ini mendukung secara *native* pengembangan aplikasi mobile untuk ke dua *platform* Android dan iOS.

2. Berdiri Sendiri

Kelebihan bahasa pemrograman Dart lainnya adalah ketersediaan SDK yang dilengkapi dengan berbagai macam tools pengembangan. Salah satu *tool*-nya adalah Dart VM, dimana tool tersebut akan membantu para *developer* untuk menjalankan kode dalam lingkungan tampilan *command line*.

3. Concurrency

Bahasa pemrograman Dart memiliki kelebihan dengan adanya konstruksi nyata dari concurrency dan paralelisme. Kelebihan bahasa pemrograman Dart ini ditawarkan dengan bentuk *Dart Isolates*. Dengan adanya *Dart Isolates*, program-program akan terisolasi untuk bekerja secara independen tanpa adanya pembagian memori, akan tetapi tetap terdapat komunikasi diantaranya. Setiap program Dart menggunakan setidaknya satu buah isolasi.

2.2.2 Flutter



Gambar 2.1 Logo Flutter

Flutter adalah SDK untuk pengembangan aplikasi *mobile* yang dikembangkan oleh Google. Sama seperti *react native*, *framework* ini dapat digunakan untuk membuat atau mengembangkan aplikasi *mobile* yang dapat berjalan pada *device* iOS dan Android. Dibuat menggunakan bahasa C, C++, Dart dan Skia. Pada *framework* ini semua kodenya di compile dalam kode *native* (Android NDK, LLVM, AOT-compiled) tanpa ada *intrepreter* pada prosesnya sehingga proses compile menjadi lebih cepat. Dari segi penulisan kodenya, Flutter sangat berbeda dari *react native* dan lebih cenderung mendekati Java Android.

Versi pertama Flutter dikenal sebagai “Sky” dan berjalan pada sistem operasi Android. Diresmikan pada perhelatan Dart *Developer Summit* tahun 2015,

dengan tujuan untuk mampu merender grafis secara konsisten pada 120 bingkai per detik. Komponen utama Flutter termasuk :

1. *Flutter Engine*

Flutter Engine, ditulis terutama dengan bahasa pemrograman C++, memberikan dukungan rendering tingkat rendah menggunakan *library* grafik Skia milik Google. Selain itu, *Flutter Engine* juga berinteraksi dengan perangkat pengembangan perangkat lunak (SDK) spesifik-serambi (*platform-specific*) seperti yang disediakan oleh Android dan iOS.

2. *Foundation Library*

Foundation Library, ditulis dengan bahasa pemrograman Dart, menyediakan fungsi dan *class-class* dasar yang digunakan untuk membangun aplikasi menggunakan Flutter, seperti API untuk berkomunikasi dengan *engine*.

3. *Widget* Spesifik Desain

Framework Flutter berisi dua set *widget* yang disesuaikan dengan bahasa desain tertentu. *Widget material design* menerapkan bahasa desain Google dengan nama yang sama, sedangkan *widget* ‘Cupertino’ meniru desain iOS milik Apple.

2.2.3 PHP

PHP sering digunakan para *programmer* untuk membuat situs *web* yang bersifat dinamis karena gratis dan berguna dalam merancang aplikasi *web*. Bahwa PHP adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan

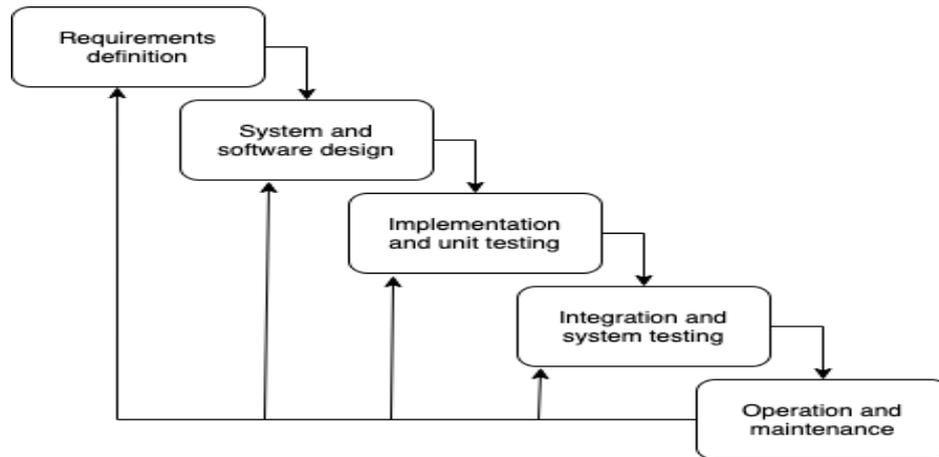
baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis *server-side* yang dapat ditambahkan ke dalam HTML.

Sedangkan menurut beberapa peneliti lainnya PHP merupakan salah satu bahasa pemrograman berbasis *web* yang ditulis oleh dan untuk pengembangan *web*. PHP merupakan bahasa (*script*) pemrograman yang sering digunakan pada sisi *server* sebuah *web*. Bahasan beberapa hal tersebut menerangkan bahwa PHP merupakan bahasa pemrograman untuk membuat atau mengembangkan aplikasi berbasis *website* dan bersifat *open-source* dan ditanamkan dalam *script* HTML.

2.2.4 Waterfall

Menurut Rosa dan M. Shalahuddin, model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*) (Rosa & Shalahuddin, 2013). Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut.

Pengembangan model *waterfall* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing / verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan (Pascapraharastyan, Supriyanto, & Sudarmaningtyas, 2014).



Gambar 2.2 Alur Metode *Waterfall*

Gambar diatas merupakan tahapan umum dari proses ini. Akan tetapi Pressman memecah model ini menjadi 6 tahapan meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya. Berikut ini adalah penjelasan tahap yang dilakukan di dalam model ini menurut Pressman (Pressman, 2008) :

- 1) *Requirements definition*. Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*, misalnya fungsi yang dibutuhkan, *user interface*.
- 2) *System and Software Design*. Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “*blueprint*” *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan sebelumnya.
- 3) *Implementation and Unit Testing*. Untuk dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi

bentuk yang dapat di mengerti oleh mesin, yaitu kedalam bahasa pemrograman melalui proses *coding*.

- 4) *Integration and System Testing*. Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan *software*. Semua fungsi-fungsi *software* harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah di definisikan sebelumnya.
- 5) *Operation and Maintenance*. Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu.

2.2.5 Basis Data (*Database*)

Basis data (*database*) berasal dari dua kata yaitu basis dan data. Basis dapat diartikan dengan markas atau gudang sedangkan data adalah representasi fakta yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, keadan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Sehingga basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis. Dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut yaitu perangkat lunak manajemen basis data (*database management system / DBMS*).

Menurut Andri Kristanto, 2008 Hal 79, Basis data merupakan *file*, tabel yang berkumpul menjadi satu dan saling terhubung yang kemudian disimpan dalam penyimpanan elektronik.

Menurut Kusri M.Kom, 2007 Hal 2, Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter atau *symbol*). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

- Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat di manfaatkan dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik

2.2.6 MySQL

Menurut Anhar (2010:21), MySQL (*My Structure Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS dari sekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang *multithread*, *multi-user* yang bersifat gratis di bawah lisensi GNU *General Public Licence* (GPL).

Menurut Arief (2011) *My Structure Query Language* atau yang biasa disingkat MySQL merupakan salah satu dari berbagai macam *database* server yang cukup digemari dalam membangun aplikasi berbasis *web* yang mengharuskan *database* sebagai sumber dan pengelolaan datanya. MySQL adalah *Structured*

Query Language yang memiliki sifat *open source*. MySQL juga dapat dijalankan dalam berbagai sistem operasi seperti *windows* dan *linux*.

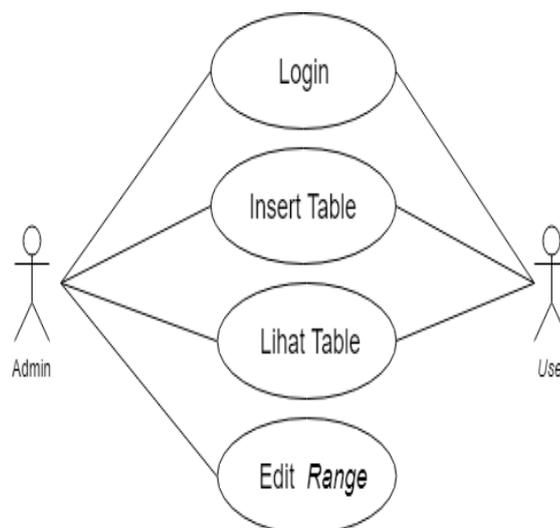
2.2.7 *Unified Modal Language (UML)*

Unified Modelling Language atau yang biasa disebut sebagai UML merupakan standarisasi dalam merancang sebuah aplikasi. UML dipakai untuk pemodelan dan perancangan dari sebuah aplikasi (Booch,2005).

Menurut Nugroho (2010: 6), UML merupakan standar rancangan sebuah *software* yang bersifat *Object Oriented*. Rancangan ini sebenarnya dipakai untuk menyederhanakan masalah-masalah yang rumit agar dapat lebih mudah untuk dimengerti dan dipelajari.

2.2.8 *Use case diagram*

Use case diagram merupakan sebuah bagan yang memebrikan gambaran hubungan antara sistem dengan pengguna. *Use case diagram* memiliki arti gambaran tentang siapa saja pemakai sistem dan upaya seperti apa yang diharapkan pemakai agar dapat terhubung dengan sistem (Whitten dkk, 2004 : 257).



Gambar 2. 3 Use case diagram

Tabel 2.1 Simbol Use case diagram

Simbol	Nama Simbol	Keterangan
	Use case	Abstraksi dan interaksi antara sistem dan aktor
 Actor	Aktor/ Pengguna	Mewakili peran orang, sistem lain, atau alat ketika berkomunikasi dengan use case.
	Relationship/ Hubungan	Abstraksi antara dari penghubung antara aktor dengan use case.
	Include	Menspesifikasikan bahwa use case sumber secara eksplisit.

2.2.9 Activity Diagram

Tabel 2.2 Activity Diagram

No	Simbol	Keterangan
1.		Kondisi Awal Bagaimana objek dibentuk atau diawali.
2.		Kondisi Akhir Bagaimana objek dibentuk dan dihancurkan

3.		<p>Kondisi Transisi</p> <p>Menunjukkan kondisi transisi antar aktivitas</p>
4.		<p>Swimlane</p> <p>Menunjukkan aktor dari diagram aktivitas yang dibuat</p>
5.		<p>Aktivitas</p> <p>Menunjukkan aktivitas-aktivitas/langkah yang terdapat pada diagram aktivitas</p>
6.		<p>Pengecekan Kondisi</p> <p>Menunjukkan pengecekan terhadap suatu kondisi</p>