

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Adapun penelitian terkait ini, peneliti akan membandingkan hasil penelitiannya dengan tiga jurnal yaitu :

1. Muhammad Amir Baihaqi, Didik Aribowo, Mustofa Abu Hamid (2020) dengan judul penelitian “Pengembangan Aplikasi Sistem Informasi Prakerin Berbasis Android Di Jurusan Elektronika Industry Smkn 1 Cikande”. dalam penelitian ini terdapat Sistem informasi prakerin berbasis Android dapat membantu proses praktek kerja industri di Jurusan Elektronika Industri SMK Negeri 1 Cikande. Proses untuk mendapatkan informasi jauh lebih mudah dan cepat dilakukan dengan cara menggunakan Smartphone. Hasil kelayakan dari ahli media diperoleh persentase sebesar 93,75% sehingga dinyatakan “sangat layak”, penilaian dari guru sebagai pengguna sebesar 78,7% dinyatakan ‘layak’ dan penilaian dari siswa sebesar 79,86% dinyatakan ‘layak’.

2. Ayu Lestari, Fanani Bastian Ahmad, & Mega Novita (2019) melakukan penelitian: “Sistem Informasi Magang Berbasis Website Pada Dinas Kesehatan Provinsi Jawa Tengah”. Aplikasi ini merupakan suatu system yang digunakan untuk pengelolaan data peserta magang untuk melakukan pendaftaran, mendapatkan informasi bagian yang tersedia dan pengumuman hasil peserta magang yang diterima oleh instansi/perusahaan. Melalui aplikasi ini *user* dapat melihat jadwal magang, bagian dan kuota pendaftar yang disediakan oleh

instansi/perusahaan sehingga membuat proses pendaftaran magang lebih efektif dan efisien tanpa menanyakan langsung ke kantor instansi/perusahaan.

3. Hendro Gunawan, Ardi Kurniawan Hadi Saputro (2017) melakukan penelitian : “Pemanfaatan Aplikasi Mobile Untuk Mempercepat Pencarian Tempat Indekos Berbasis Android”. dengan aplikasi ini Banyak mahasiswa / pegawai yang berasal dari luar daerah / provinsi yang tinggal di Purwokerto untuk melangsungkan kehidupannya. Mahasiswa / pegawai yang mencari indekos mendapatkan informasi dari teman atau mencari langsung, hal ini menjadi kurang efektif dan tidak efisien. Selain itu, budget, fasilitas, dan juga letak indekos menjadi pertimbangan. Disisi lain, pemilik indekos kesulitan dalam mempublikasikan kamar indekos yang kosong, sehingga aplikasi ini dapat membantu pemilik indekos dalam mempublikasi kamar indekosnya secara maksimal.

4. Budi Sudrajat (2021) melakukan penelitian: “Penggunaan Teknologi Flutter dalam Aplikasi Mobile Untuk Pengembangan Kedai Kopi”. Dengan adanya aplikasi ini membuat para pemilik kedai kopi mendapatkan terobosan untuk menaikkan pendapatan mereka, salah satunya melalui penjualan secara online khususnya dengan menggunakan perangkat mobile. Aplikasi ini akan menjembatani antara para pemilik kedai kopi dengan para penikmat kopi dalam transaksi di masa pandemi saat ini.

5. Muhammad Arifin, Evangs Mailoa(2021) melakukan penelitian dengan judul: “Appllication Of Haversine Formula In Nearest Salatiga Repair Shop Search Geographic Information System Using Flutter Framework”. penelitian ini

menghasilkan suatu sistem informasi geografis pencarian jarak terdekat dari suatu bengkel motor dan mobil yang berada di kota Salatiga. Penelitian ini menghasilkan sebuah aplikasi mobile dengan perhitungan jarak yang dapat dikatakan mendekati akurat dan menampilkan rute dari titik pengguna ke bengkel yang dipilih dengan pembangun peta digital dari Google Maps.

II.2. Uraian Teoritis

II.2.1. Aplikasi

Menurut Irsan, M. (2015), Aplikasi Mobile adalah perangkat lunak yang berjalan pada perangkat mobile seperti smartphone atau tablet PC. Aplikasi Mobile juga dikenal sebagai aplikasi yang dapat diunduh dan memiliki fungsi tertentu sehingga menambah fungsionalitas dari perangkat mobile itu sendiri. Untuk mendapatkan mobile application yang diinginkan, user dapat mengunduhnya melalui situs tertentu sesuai dengan sistem operasi yang dimiliki. Google Play dan iTunes merupakan beberapa contoh dari situs yang menyediakan beragam aplikasi bagi pengguna Android dan iOS untuk mengunduh aplikasi yang diinginkan.

II.2.2. Magang

Program Magang bertujuan untuk mempraktekkan hasil teori-teori serta praktek yang diberikan di jenjang Pendidikan dalam dunia industri sehingga program magang dibuat untuk meningkatkan skill kompetensi bagi mereka, (Rugaiyah, 2011).

II.2.3. Android

Android adalah kumpulan perangkat lunak untuk perangkat seluler. Android menyediakan platform terbuka bagi pengembang untuk membuat aplikasi mereka sendiri. Bahasa pemrograman pada Android adalah bahasa pemrograman Java (Christina, S. dkk, 2019).

Android merupakan salah satu sistem operasi mobile yang sedang berkembang diantara sistem operasi lainnya yang saat ini sedang dalam pengembangan. Android adalah *platform open source* komprehensif yang dirancang untuk perangkat seluler. Android adalah sistem operasi berbasis Linux untuk perangkat seluler. Android adalah sistem operasi yang paling populer di masyarakat karena menawarkan manfaat bersifat *open source* yang memberikan kebebasan kepada pengembang untuk menulis aplikasi (Irnin Agustina Dwi Astuti, 2017).

Namun seiring perkembangannya, android berubah menjadi *platform* yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya yaitu Google. Google-lah yang mengakuisisi android, kemudian membuatkan sebuah *platform*.

Platform android terdiri dari sistem operasi berbasis linux, sebuah GUI (*Graphic User Interface*), sebuah web browser dan aplikasi *end-user* yang dapat di *download* dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

II.2.4. Flutter

Menurut Krisnada, F. E., & Tanone, R. (2019) *Flutter* adalah sebuah *SDK* atau *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS. *Flutter* menggunakan bahasa pemrograman *Dart* dalam pengkodean. Perbedaan *framework Flutter* dengan yang lainnya yaitu dalam *build* aplikasi, pada *framework* ini semua kodenya di *compile* dalam kode *native*-nya (Android NDK, LLVM, AOT compiled) tanpa ada interpreter pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat.

Menurut Andrean, K (2020). *Flutter* adalah *Mobile App SDK* milik Google dimana *developer* dapat membuat aplikasi untuk IOS dan Android dengan menggunakan Bahasa dan *source code* yang sama [2]. Bahasa yang digunakan pada *flutter* adalah Bahasa pemrograman *Dart*. Pada *flutter*, *developer* dapat membuat aplikasi secara *native* dengan menggunakan Bahasa pemrograman *Dart*. *Flutter* juga mempunyai *widget* nya sendiri. Dengan didukung oleh Google, *flutter* dapat berkembang sangat cepat dan dapat bersaing dengan *Xamarin*, *React Native*, dan *framework mobile* lainnya. Berikut adalah gambar dari struktur dari *flutter framework*.

II.2.5. Software Development Kit (SDK)

Menurut Sigit mintoro, Kurniawati, & Muhammad Rudini (2019) *SDK* adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa

pemrograman *Dart*. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android *SDK (Software Development Kit)* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada platform Android.

II.2.6. *Dart Programming Language*

Dart dirancang dan dikembangkan oleh Lars Bak dan Kasper Lund di Google yang secara resmi diperkenalkan pada Oktober 2011 di Konferensi GOTO, Denmark dan versi pertama Dart yaitu versi 1.0 resmi dirilis pada Nov 2013. *Dart* merupakan *programming language* lintas *platform* atau *platform* independen yang artinya dapat dijalankan pada sistem operasi yang berbeda seperti Windows, Linux, Unix dan Mac OS, dll yang awalnya dikembangkan oleh Google dan kemudian disetujui sebagai standar oleh Ecma, yang saat ini digunakan untuk membangun aplikasi web, server, desktop, dan seluler.

II.2.7. *Unified Modeling Language (UML)*

Menurut Ariani R. Sukanto dalam Taufik (2017) “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram teks-teks pendukung”.

Pendapat lainnya menurut Fowler, M. dalam (B. O. Lubis, 2016) UML (*Unified Modeling Language*) adalah “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). Definisi ini merupakan definisi yang sederhana”.

Jadi UML (*Unified Modeling Language*) dapat diartikan sebagai bahasa visual untuk menggambarkan definisi-definisi tentang *requirement*, membuat analisis dan desain serta menggambar arsitektur dalam pemrograman berorientasikan objek dengan menggunakan teks-teks pendukung.

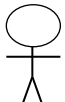
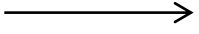
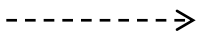
1. *Use case Diagram*

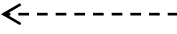
Menurut Mamed Rofendy Manalu, (2015) “*Use Case Diagram* adalah sesuatu atau proses merepresentasikan hal-hal yang dapat dilakukan oleh aktor dalam menyelesaikan sebuah pekerjaan.”

Menurut Shalahuddin dalam jurnal (Umar Al Faruq, 2015) mengungkapkan : “*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. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.” Berikut adalah simbol-simbol yang ada pada diagram *use case* pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	<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> .
	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 control terhadap <i>use case</i> .
	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.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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

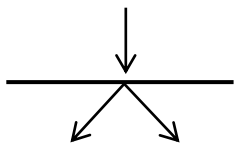
	program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

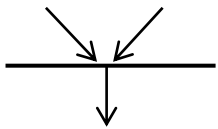
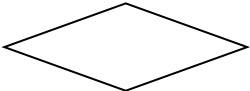
Sumber : Jurnal (Umar Al Faruq, 2015)

2. Activity Diagram

Menurut Rosa dalam jurnal (Sari dan David) mengungkapkan : “*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas dapat dilakukan oleh sistem.” Simbol-simbol yang digunakan dalam *activity diagram* sebagai berikut :

Tabel II.2. Simbol Activity Diagram

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

(Sumber : Ade Hendini ; 2016)

3. Class Diagram

Menurut Rosa dalam jurnal (Sari dan David) mengungkapkan : “*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. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.”

Tabel II.3. Multiplicity Class Diagram

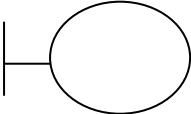
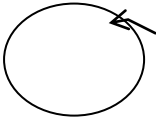
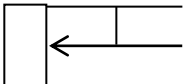
Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

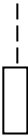
(Sumber : Ade Hendini ; 2016)

3. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence* diagram dapat dilihat pada tabel II.6 dibawah ini :

Tabel II.4. Simbol *Sequence* Diagram

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>, <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 : Ade Hendini ; 2016)