



BAB II

TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penulis menggunakan referensi dari jurnal-jurnal dan penelitian terkait untuk membantu penelitian yang dilakukan agar sesuai dengan yang diharapkan.

Penelitian yang dilakukan oleh Ardi Anggara mahasiswa Universitas Wijaya Kusuma Surabaya pada tahun 2018 yang berjudul “ Sistem Informasi Manajemen Kos-Kosan Di Kos Hadi ”. Penelitian tersebut telah menghasilkan sebuah sistem informasi tentang informasi kamar kos serta memudahkan pemilik kos dalam sistem pembayaran.

Penelitian yang dilakukan oleh Pratama Rezalendra pada tahun 2019 tentang “ Aplikasi Persebaran Rumah Kos Berbasis Android Studi Kasus Kota Malang ”. Penelitian yang dilakukan dapat memetakan persebaran rumah kos di Kota Malang, kemudian memudahkan masyarakat dapat mencari kos tersebut dengan mudah.

Penelitian yang dilakukan Renita Selviana dan Laili Dwi Utari pada tahun 2019, yang berjudul “ Sistem Informasi Pencarian Kos Kosan Di Sekitar Kecamatan Beji Berbasis Web ”. Hasil dari penelitian ini adalah menghasilkan sistem informasi yang dapat digunakan dan membantu pengguna rumah kos memudahkan calon penyewa untuk menemukan rumah kost sesuai dengan kriteria yang diharapkan.

Penelitian yang dilakukan Desi Purnama Sari pada tahun 2021, yang berjudul, “ Rancang Bangun Kosan Online Berbasis Android ”. Penelitian ini dibuat dengan melakukan survei dan wawancara terhadap masyarakat dan pemilik kos. Penelitian ini guna sebagai penyedia informasi yang lengkap mengenai keberadaan kosan di Bandar Lampung.

Penelitian yang dilakukan Yamonaha pada tahun 2021, yang berjudul, “ Perancangan Aplikasi E-Kost Berbasis Android Di Kota Batam”. Penelitian ini dibangun dengan sebagai media alternatif dalam mengetahui sistem informasi tentang pencarian kos. Dengan bantuan aplikasi ini, diharapkan dapat memfasilitasi para pencari kos dan pemilik kos dalam melakukan transaksi pemesanan kos.

II.2. Landasan Teori

Adapun landasan teori yang digunakan sebagai landasan dalam penulisan proposal skripsi ini adalah sebagai berikut :

II.2.1. Perancangan

Perancangan adalah suatu proses yang bertujuan untuk menganalisis, menilai, memperbaiki dan menyusun suatu sistem, baik sistem fisik maupun non fisik untuk waktu yang akan datang dengan memanfaatkan informasi yang ada. Dalam perancangan ada beberapa tahapan yang harus dilakukan sebelum hasil dari rancangan dijadikan sebuah produk berjenis aplikasi.

Menurut Rusdi Nur, dkk (2018), “ Perancangan adalah suatu proses untuk membuat dan mendesain sistem yang baru.

Berdasarkan definisi tersebut dapat disimpulkan bahwa perancangan adalah perencanaan dan pembuatan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya serta memungkinkan terjadinya kegagalan yang harus diselesaikan menyangkut pengaturan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem yang berjalan.

II.2.2. Kost

Perkembangan penyediaan kost-kostan semakin banyak di berbagai wilayah tanpa terkecuali. Sasaran penyedia kost-kostan biasanya membangun di tempat di wilayah tertentu misalnya sekolah, kampus, perkantoran, rumah sakit maupun tempat lainnya yang membutuhkannya tempat tinggal. Mencermati perkembangan penyediaan kost-kostan akhir-akhir ini, membuat suatu penanganan permasalahan dimana para pemilik kost harus bersaing secara strategis untuk mempermudah para pencari kost-kostan agar tertarik untuk menyewa kamar kost yang disediakan. Seperti yang diketahui bahwa pencari kost- kostan membutuhkan tempat kost yang nyaman, berfasilitas dan tentunya dengan harga yang terjangkau.

Kost merupakan tempat tinggal yang sifatnya sementara dan disewakan selama beberapa waktu tertentu, sesuai dengan prosedur antara pemilik kost

dengan penyewa kost. Kost disediakan untuk memfasilitasi wanita maupun pria dari kalangan pelajar, mahasiswa dan pekerja pada umumnya untuk tempat tinggal sesuai jangka waktu yang telah ditentukan.

II.2.3. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. Membeli Android Inc., Pendatang baru yang membuat peranti lunak untuk ponsel. Android merupakan *open source*, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. *Platform* ini akan terus berkembang untuk membangun aplikasi *mobile* yang inovatif.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *Open Source* pada perangkat *Mobile*. Di lain pihak, *Google* merilis kode-kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang dapat mendukung penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution (OHD)*.

Telepon yang pertama memakai sistem operasi Android adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2010 diperkirakan hampir semua perusahaan seluler di dunia menggunakan Android sebagai *operating system*. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut :

1. Android versi 1.0 & 1.1 (Astro-Alpha & Bender-Beta)

Platform pertama Android dirilis pada september 2008 disini lah awal urutan QS Android bermula lewat *smartphone* pertama yaitu *HTC Dream*.

2. Android versi 1.5 (*Cupcake*)

Dirilis pada tanggal 30 April 2009 dengan berbagai fitur di sebuah perangkat *smartphone* untuk menggantikan *feature phone* pada saat itu.

3. Android versi 1.6 (*Donut*)

Dirilis pada 15 September 2009 beberapa bulan setelah dirilisnya versi sebelumnya. Android ini pun memberikan dukungan layar yang lebih besar.

4. Android versi 2.0 / 2.1 (*Eclair*)

Dirilis pada 3 Desember 2009 kembali diluncurkan Android dengan dukungan fitur *Bluetooth* hingga fitur kamera yang mulai menjadi nilai jual *smartphone* kala itu. Android ini digunakan pada perangkat seperti *HTC Nexus One*.

5. Android versi 2.2 (*Froyo: Frozen Yoghurt*)

Dirilis pertama kali pada tanggal 20 Mei 2010, memberikan peningkatan pada kecepatan kerja, fitur *USB tethering*, *WiFi hotspot*, serta fitur keamanan.

6. Android versi 2.3 (*Gingerbread*)

Diluncurkan pada Desember 2010, memberikan kelebihan terutama pada tampilan tatap muka alias *user interface* yang digunakan pada tipe Android yang satu ini.

7. Android versi 3.0 (*Honeycomb*)

Diluncurkan pada 10 Mei 2010 sebagai revisi dari versi Android sebelumnya, untuk mendukung Samsung yang merilis perangkat *table Samsung Galaxy Tab Series*.

8. Android versi 4.0 (*Ice Cream Sandwich*)

Dirilis pada 19 Oktober 2011 silam dan masih tetap menggunakan nama *dessert*. Android ini mampu memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan.

9. Android versi 4.1 dan 4.3 (*Jelly Beans*)

Sistem operasi Android ini dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis dimana bisa memberikan peningkatan fungsi pada *user inter face* dan **teknologi Vsync** yang digunakannya.

10. Android versi 4.4 (*KitKat*)

Dirilis pada Oktober 2013 jenis Android terfavorit karena mendukung hampir seluruh perangkat *smartphone* di dunia memberikan

optimalisasi yang baik, termasuk pada perangkat *smartphone* yang memiliki spesifikasi kurang mumpuni alias cukup rendah saat itu.

11. Android versi 5.0 dan 5.1 (*Lollipop*)

Dirilis pada Juni 2014 versi Android ini sudah mendukung arsitektur 64-bit yang sudah memungkinkan penggunaan RAM di atas 3GB. Salah satunya **Asus Zenfone 2** yang sudah mengusung RAM 4GB saat itu.

12. Android versi 6.0 (*Marshmallow*)

Sistem operasi ini pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015 silam. Nama OS Android ini secara jelas memberikan peningkatan pada sistem keamanan dengan *finger print sensor* sebagai sistem keamanan biometrik yang digunakan.

13. Android versi 7.0 dan 7.1 (*Nougat*)

Pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot Android dengan batangan Nougat. Versi Android ini mengalami perubahan dari segi tampilan antarmuka. Selain itu, ada juga fitur splitscreen untuk membagi tampilan layar untuk dua aplikasi sekaligus.

14. Android versi 8.0 dan 8.1 (*Oreo*)

Dirilis mulai Agustus 2017 sistem operasi ini menawarkan *project treble* yang mampu mempercepat dalam proses pembaruan pada *smartphone*.

15. Android versi 9.0 (*Pie*)

Diperkenalkan pada Agustus 2018 memberikan navigasi berupa *gesture* yang menggantikan tombol fisik *Home*, *Back*, dan *Recent Apps*. Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan, hingga sistem *screenshot* terbaru yang lebih mudah.

16. Android versi 10

Dirilis pada 13 Maret 2019 salah satu fitur dalam versi Android terbaru 2019 ini adalah ***Dark Mode*** alias mode gelap yang diklaim mampu meningkatkan performa baterai.

17. Android versi 11

Dirilis pada tahun 2020 dimana terdapat tambahan fitur seperti notifikasi, *screen recorder* tanpa aplikasi tambahan, *picture – in - picture*, hingga mengambil *screenshot* panjang.

18. Android versi 12

Dirilis pada 18 Mei 2021 Android 12 ini menawarkan beragam fitur menarik yang belum ada pada pendahulunya, seperti perubahan desain *material you* yang memudahkan pengguna handphone, peningkatan fitur privasi, dan mode satu tangan.

II.2.4. *Visual Studio Code*

Visual Studio Code adalah editor source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk

debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source, meskipun unduhan resmi berada di bawah lisensi proprietary.

Teks editor *Visual Studio Code* juga bersifat *open source*, yang mana kode sumbernya dapat dilihat dan dapat berkontribusi untuk pengembangnya. *Visual Studio Code* telah dirancang untuk bekerja dengan alat-alat yang ada, dan *Microsoft* menyediakan dokumentasi untuk membantu pengembang bersama, dengan bantuan untuk bekerja dengan *ASP.NET 5*, *Node js*, dan *Microsoft* naskah, serta alat-alat yang digunakan untuk membantu membangun dan mengolah aplikasi *Node js*.

II.2.5. Flutter

Flutter merupakan sebuah SDK untuk pengembangan aplikasi mobile yang dikembangkan oleh *Google* untuk membangun aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke *platform Android* dan *iOS* dari codebase tunggal. *Flutter* dapat dengan mudah dipelajari karena menggunakan bahasa pemrograman *Dart* yang pastinya terasa familiar jika sudah terbiasa menggunakan bahasa pemrograman *Java* atau *Javascript*. Selain itu *flutter* juga menyertakan kerangka *reactive-functional*, mesin render 2D, *widget* siap pakai, dan *tools* untuk pengembangan. (CrossTechno, 2019).

II.2.6. Xampp

Xampp adalah sebuah paket perangkat lunak (*software*) komputer yang sistem penamaannya diambil dari akronim kata *Apache*, *MySQL (dulu) / MariaDB* (sekarang), *PHP*, dan *Perl*. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti *OS Linux*, *OS Windows*, *Mac OS*, dan juga *Solaris*.

Xampp bisa digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada didalam website kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui *Xampp control panel*, atau istilahnya *website offline*.

II.2.7. Laravel

Laravel adalah *framework* aplikasi *web* kontemporer, *open source* dan digunakan secara luas untuk perancangan aplikasi *web* yang cepat dan mudah. *Laravel* dibuat oleh Taylor Otwell pada tahun 2011. *Framework* ini dibuat untuk pengembangan aplikasi *website* dimana mengikuti arsitektur *MVC (model view controller)*.

MVC itu sendiri adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. *MVC* memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, *controller*, *user interface*.

II.2.8. Unified Modelling Language (UML)

Pemodelan perangkat lunak bekerja dengan cara yang cukup serupa layaknya seorang arsitek atau insinyur teknik sipil yang akan membuat sebuah bangunan, ia biasanya membuat denah – denah yang menggambarkan bentuk jadi dari bangunan.

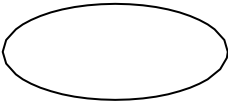
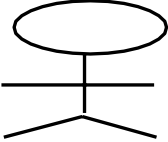
UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa dan Shalahuddin 2018:137).

II.2.8.1. Diagram UML

1. 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 .

Tabel II.1. Use Case Diagram

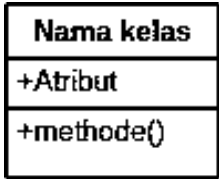
No	Simbol	Deskripsi
1.	<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.	Aktor / <i>actor</i> 	Adalah orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
3.	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> .
4.	<<Include>> 	Merupakan di dalam <i>use case</i> lain (required) atau pemanggilan <i>use case</i> lain, contohnya adalah pemanggilan sebuah program.
5.	<<Extends>> 	Merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat.

Sumber : Rosa dan Shalahuddin (2018).

2. Class Diagram

Adalah sebuah spesifikasi yang jika diinisiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Adapun simbol *class diagram* yang ditunjukkan pada gambar dibawah ini :

Tabel II.2. Multiplicity Class Diagram

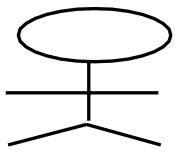
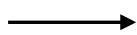
No	Simbol	Penjelasan
1.	<i>Class</i> 	Menggambarkan sebuah kelas pada sistem yang terbagi menjadi 3 bagian. Bagian atas adalah nama kelas. Bagian tengah adalah atribut kelas. Bagian bawah adalah <i>methode</i> dari kelas.
2.	<i>Association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
3.	<i>Agregation</i> 	Relasi antar kelas dengan makna semua-bagian (whole-part).
4.	<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.	<i>Generlisasi</i> 	Relasi antar kelas dengan makna generalisasi - generalisasi- spesialisasi (umum khusus).

Sumber : Rosa dan Shalahuddin (2018).

3. Sequence Diagram

Diagram sequence “menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* 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*. Adapun simbol *sequence* diagram yang ditunjukkan pada gambar di bawah ini :

Tabel II.3. Sequence Diagram

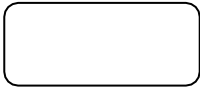
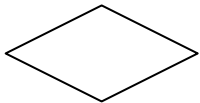
No	Simbol	Deskripsi
1.	<i>Actor</i> 	Adalah orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2.	<i>Lifeline</i> 	Menyatakan kehidupan suatu objek
3.	<i>Message</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah
4.	<i>Active Time</i> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya

Sumber : Rosa dan Shalahuddin (2018).

4. 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. Penekanan pada diagram aktivitas adalah menggambarkan aktivitas sistem atau aktivitas yang dapat dilakukan oleh sistem, bukan apa yang dilakukan aktor. Adapun simbol *activity diagram* yang ditunjukkan pada gambar di bawah ini :

Tabel II.4. Activity Diagram

No	Simbol	Deskripsi
	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas 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 	Asosiasi, penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Sumber : Rosa dan Shalahuddin (2018).