

BAB II

LANDASAN TEORI

2.1 Rancang Bangun

Rancang bangun adalah suatu istilah umum untuk membuat atau mendesain suatu objek dari awal pembuatan sampai akhir pembuatan. Rancang bangun berawal dari kata desain yang artinya perancangan, rancang, desain, bangun. Sedangkan merancang artinya mengatur, mengerjakan atau melakukan sesuatu dan perancangan artinya proses, cara, perbuatan merancang (Ariansyah, A., Fajriyah, F., & Prasetyo, F. S. (2017).

rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada. (Putri, Kurniawan, & Sari, 2019)

2.2 E-Complaint

E-Complaint merupakan fasilitas komplain berbasis elektronik yang memiliki fungsi penting untuk menampung keluhan, kritik, dan saran dari pihak yang berhubungan dengan pihak developer perumahan. Keluhan, kritik serta saran merupakan suatu *instrument* dalam melakukan evaluasi dan deteksi dini terhadap kelemahan sistem ataupun penyimpangan pelaksanaan manual mutu. *Ecomplaint* merupakan sebuah sarana yang baik untuk mengetahui tingkat pelayanan yang telah diberikan oleh pihak developer perumahan kepada konsumen. *E-Complaint* dapat dijadikan sebuah wadah yang berfungsi menampung keluhan secara *online* terhadap pihak developer perumahan. Layanan *E-Complaint* merupakan fasilitas layanan yang berfungsi penting untuk mendapatkan *feedback* berupa keluhan yang berhubungan dengan kampus. *Ecomplaint* merupakan sarana yang baik untuk mengukur tingkat pelayanan yang diberikan pihak developer perumahan. Pengukuran tingkat pelayanan tersebut dipengaruhi oleh dokumen-dokumen keluhan yang ada dalam *E-Complaint*. Melalui dokumen *Ecomplaint* dapat

diketahui keluhan, saran atau kritik apa saja yang perlu ditindaklanjuti sesegera mungkin (Azima, M. F. 2018).

E-Complaint merupakan sarana yang baik untuk mengukur tingkat pelayanan yang diberikan oleh pihak developer. Pengukuran tingkat pelayanan tersebut dipengaruhi oleh dokumen-dokumen keluhan yang ada dalam *E-Complaint*. Melalui dokumen *E-Complaint* dapat diketahui keluhan, saran atau kritik apa saja yang perlu ditindaklanjuti sesegera mungkin. (Indriyati, 2018)

2.3 Android

Android adalah sebuah sistem operasi untuk smartphone dan Tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (device) dan penggunanya, sehingga pengguna bisa berinteraksi dengan device-nya dan menjalankan aplikasi-aplikasi yang tersedia pada device (Yuni Puspita Sari, 2016).

2.3.1 Elemen Android

1. Dalvik Virtual Machine (DVM)

Salah satu element kunci dari Android adalah Dalvik Virtual Machine (DVM). Android berjalan di Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan Java Virtual Machine (JVM) seperti Java ME (Java Mobile Edition), tetapi Android menggunakan virtual machine sendiri yang dirancang untuk memastikan beberapa fitur-fitur berjalan lebih efisien pada perangkat mobile.

2. Android SDK (Software Development Kit)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. 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 menggunakan bahasa pemrograman Java.

3. ADT (Android Development Tools)

Android development tools adalah plugin yang di desain untuk IDE Android Studio yang memberikan kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Android Studio. Dengan menggunakan ADT untuk Android Studio akan memudahkan dalam membuat aplikasi project Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

2.4 Perangkat Lunak Yang Digunakan

2.4.1 Android Studio

Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android (Yuni Puspita Sari, 2016).

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT plugin (Android Development Tools). Android studio memiliki fitur :

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah
- f. Didukung oleh Google Cloud Platfrom untuk setiap aplikasi yang dikembangkan.

2.4.2 Firebase

Firestore adalah penyedia layanan cloud dengan back-end sebagai servis yang berbasis di San Fransisco, California. *Firestore* membuat sejumlah produk untuk pengembangan aplikasi Mobile ataupun web. *Firestore* didirikan oleh Andrew Lee

dan James Tamplin pada tahun 2011 dan diluncurkan dengan cloud *database* secara realtime di tahun 2012. Produk utama dari *Firebase* yakni suatu database yang menyediakan API untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat multiple client. Perusahaan ini diakuisi oleh Google pada Oktober 2014.

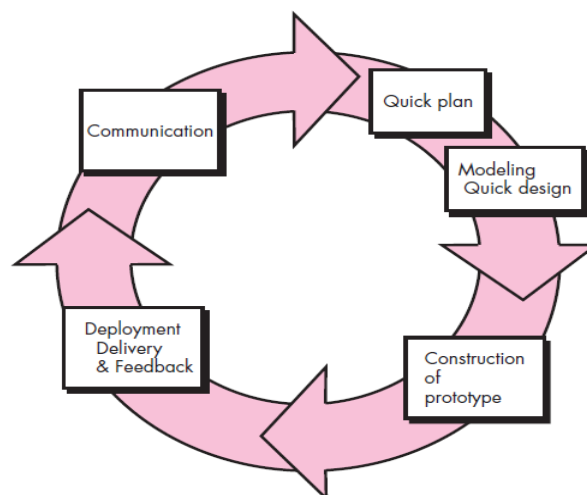
Firebase adalah penyedia layanan realtime database dan backend sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk client yang berbeda-beda dan disimpan pada cloud-nya 40 *Firebase*. *Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, Ios, Javacript, Java, Objective-C dan Node.JS . Database *Firebase* juga bersifat bisa diakses lewat REST API. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima push *notification* dari *server*. Pengembang menggunakan REST API untuk *post* data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data secara *realtime*. Pengembang juga dapat menggunakan *database* ini untuk mengamankan data menggunakan *server Firebase* dengan *rules* yang ada. Untuk *hosting* file *firebase* menyediakan hosting untuk static file dengan fasilitas CDN dan SNL. (Sonita & Fardianitama, 2018)

2.5 Prototype

Metode yang digunakan adalah metode *Prototype*, *Prototype* merupakan metode yang efektif dalam merancang perangkat lunak. *Prototype* dimulai dengan mengumpulkan kebutuhan yang akan di rancang. Pengembang mendefinisikan *object* keseluruhan dari perangkat lunak, mengidentifikasi segala aktifitas yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan format *output*), (Dwi, Nasution, & Sari, 2018).

Berikut adalah tahapan dalam metode *prototype* :

1. Komunikasi (*Communication*) dan pengumpulan data awal, yaitu komunikasi dengan klien dan *user* untuk menentukan kebutuhan.
2. Perencanaan cepat (*Quick Plan*), yaitu pembuatan perencanaan analisis terhadap kebutuhan pengguna.
3. Pemodelan perancangan cepat (*Modeling Quick Design*), yaitu membuat rancangan desain program.
4. Pembentukan *prototype* (*Construction of prototype*), yaitu pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
5. Penyerahan sistem dan umpan balik (*Development Delivery and Feedback*), yaitu memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.



Gambar 2.1 Diagram *Prototype*

2.6 Pengujian *Black-Box Testing*

Metode pengujian pada aplikasi ini menggunakan *Black Box Testing* yaitu yang berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang

tidak dicakup oleh *White Box Testing* (Pressman 2012). *Black Box Testing* cenderung untuk menemukan hal-hal berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.7 Sistem Pemodelan

2.7.1 *Unified Modelling Language* (UML)

Perancangan berorientasi obyek biasanya menggunakan model yang dikenal dengan *Unified Modeling Language* (UML) yang merupakan sebuah bahasa pemodelan objek standar sebagai ganti dari pendekatan atau metode berorientasi objek standar. *Unified Modeling Language* (UML) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. (Marini, M. 2019).

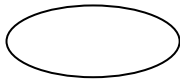
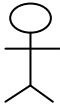

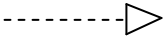
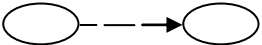
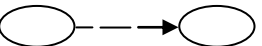
UML tersusun atas sejumlah elemen grafis membentuk diagram-diagram. Dalam penelitian ini melakukan desain dengan 3 diagram yaitu *Use Case Diagram*, *class diagram* dan *Activity Diagram*.

2.7.2 *Use Case Diagram*

Use Case diagram merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *Use Case* tidak hanya sangat penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk mencari kelas-kelas yang terlibat dalam aplikasi, serta untuk melakukan pengujian (Arfida, S., Amnah, A., & Wibowo, H. 2018).

Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.



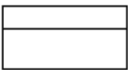

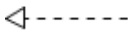
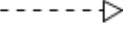

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Generalisasi 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.

2.7.3 Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desainberorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



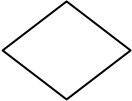


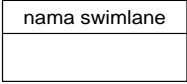
Tabel 2.2 Simbol Class Diagram

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

2.7.4 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh system.

Tabel 2.3 *Simbol Activity Diagram*

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.

2.8 Penelitian Terdahulu

Tabel 2.4 berikut ini adalah beberapa penelitian terdahulu yang berkaitan dengan media pembelajaran.

Tabel 2.4 Penelitian Terdahulu

Nama	Judul	Terbit	Uraian
Saffira Lindra Putri, Dyah Puteria Wati	ANALISA DAN PERANCANGAN SISTEM PENGATURAN KOMPLAIN BERBASIS WEB DENGAN MENGUNAKAN PENDEKATAN CUSTOMER RELATIONSHIP MANAGEMENT (CRM) DI PT. SERPONG MEGA SUKSES	STKIP Muhammadiyah Kuningan 2016	Penggunaan sistem berbasis <i>web</i> dengan konsep CRM diharapkan dapat mempermudah penyampaian informasi mengenai masa pemeliharaan dari pihak Residence One ke pemilik rumah serta memudahkan pemilik rumah dalam melakukan komplain. Dengan adanya sistem ini, kesalahan kontraktor dalam mengerjakan komplain dapat diminimalisir dengan tidak menggunakan rangkap <i>form</i> komplain yang tidak baik kualitasnya sehingga pemilik

			rumah dapat memperkirakan kapan komplain mereka akan selesai dikerjakan dengan adanya status pengerjaan komplain. Sehingga tidak adanya kesalahpahaman dalam waktu pengerjaan komplain.
Deddy Rakhmad Hidayat, Muhammad Riza Firdaus	ANALISIS PENGARUH KUALITAS LAYANAN, HARGA, KEPERCAYAAN, CITRA PERUSAHAAN, DAN KEPUASAN PELANGGAN TERHADAP LOYALITAS PELANGGAN : (Studi Pada Pelanggan Telkom Speedy di Palangka Raya)	Jurnal Wawasan Manajemen, <i>Vol. 2, Nomor 3</i> , 2014	Hasil dan implikasi penelitian ini secara teoritis adalah bahwa studi ini mendukung penelitian terdahulu yang menyatakan bahwa kualitas pelayanan berpengaruh positif dan signifikan terhadap citra perusahaan, citra perusahaan berpengaruh positif dan signifikan terhadap kepuasan pelanggan, kepuasan pelanggan berpengaruh positif dan signifikan terhadap kepercayaan,

			<p>harga berpengaruh negatif dan signifikan terhadap loyalitas pelanggan, kepuasan pelanggan berpengaruh positif dan signifikan terhadap loyalitas pelanggan.</p>
<p>Berliantina Rosita, Sigit Indrawijaya</p>	<p>PENGARUH PENANGANAN KOMPLAIN TERHADAP KEPUASAN NASABAH</p>	<p>Fakultas Ekonomi Dan Bisnis Universitas Jambi, 2015</p>	<p>Tujuan studi adalah menjelaskan penanganan komplain di BRI, dan menjelaskan pengaruh penanganan komplain terhadap kepuasan nasabah. Sampel adalah nasabah Bank BRI dengan ukuran sampel 100 responden. Penarikan sampel menggunakan teknik random sampling. Hasil penelitian menunjukkan hubungan yang signifikan baik hubungan simultan maupun parsial.</p>

			<p>Probabilita pengaruh ketiga variabel terhadap kepuasan nasabah adalah procedural justice (0,001), interactional justice (0.043), dan distributive justice (0,000). Dengan demikian, disarankan agar organisasi menginternalisasi penanganan keluhan sebagai kompetensi penting.</p>
<p>Fitria, M Fauzan Azima, Sulyono</p>	<p>TEKNOLOGI INFORMASI E-COMPLAINT PADA PERGURUAN TINGGI</p>	<p><i>Jurnal Informatika, Vol. 18, No.2 Desember 2018</i></p>	<p>Tujuan penelitian ini adalah merancang dan membangun sebuah teknologi informasi yang dapat menampung keluhan, kritik dan saran terhadap bagian operasional kampus dan dosen yang berada pada IBI Darmajaya berupa E-Complaint.</p>

<p>Sulyono, Fitria, Lia Indriyati</p>	<p>RANCANG BANGUN TEKNOLOGI INFORMASI E- COMPLAINT PADA PERGURUAN TINGGI</p>	<p>Seminar Nasional Teknologi dan Bisnis 2018 IIB DARMAJAYA Bandar Lampung, 14 Agustus 2018</p>	<p>Tujuan penelitian ini adalah merancang dan membangun sebuah teknologi informasi yang dapat menampung keluhan, kritik dan saran terhadap bagian operasional kampus dan dosen yang berada pada IBI Darmajaya berupa E-Complaint. Tahapan pengembangan sistem yang digunakan yaitu menggunakan model SDLC air terjun (waterfall) sering juga disebut model sekuensial linier. Aplikasi e-complaint ini memudahkan mahasiswa IBI Darmajaya untuk memberikan complaint (keluhan) maupun kritik dan saran kepada bagian operasional kampus maupun dosen. Aplikasi ini</p>
---	--	---	--

			<p>menggunakan framework bootstrap yang mampu mengelola tampilan aplikasi menjadi responsive. Dengan sifatnya yang responsive maka aplikasi dapat menyesuaikan bentuk pada berbagai ukuran device. Aplikasi ini berguna untuk membantu Quality Assurance Center (QAC) yang berfungsi sebagai lembaga penjamin mutu sebuah kampus .</p>
--	--	--	--