

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Rancang Bangun

Rancang bangun adalah merupakan kegiatan menerjemahkan hasil Analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada (Hasyim et al., 2014).

Rancangan sistem merupakan penentu pada saat proses data yang digunakan oleh sistem baru. Dengan sistem yang berbasis computer, maka rancangan dapat memberikan spesifikasi hardware computer yang akan digunakan. Pengembangan dan pembuatan alur sketsa dapat didefinisikan sebagai perancangan suatu sistem.

Tujuan dari rancangan sistem adalah :

- a. Untuk mencangkup kebutuhan yang diinginkan oleh pemakai sistem.
- b. Agar dapat memberikan pemodelan gambar secara detail dan rancang bangun yang sesuai gambaran kepada penyusun program.

Dengan demikian pengertian rancang bangun dapat diartikan dengan kegiatan dalam menjalankan hasil analisa ke dalam bentuk perangkat lunak dan akan tercipta suatu sistem baru atau menambah fitur di sistem sebelumnya (Purwanto, 2018).

2. Aplikasi

Istilah aplikasi berasal dari bahasa Inggris “application” yang berarti penerapan, lamaran ataupun penggunaan. Sedangkan secara istilah, pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus komputer eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Purwanto, 2018).

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu Teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Lestari et al., 2017).

3. Tabungan

Tabungan merupakan jenis simpanan yang sangat dikenal oleh masyarakat, baik masyarakat dari kalangan menengah ke bawah seperti masyarakat pedesaan maupun menengah ke atas seperti masyarakat kota.

Pengertian tabungan menurut Undang-Undang no. 10 tahun 1998 tentang perbankan atas Undang-Undang no. 7 tahun 1992 tentang perbankan pasal 1 ayat 9: “Merupakan simpanan yang penarikannya dapat dilakukan menurut syarat tertentu yang disepakati tetapi tidak dapat ditarik dengan cek atau alat yang dipersamakan itu”.

4. Qurban

Qurban berasal dari Bahasa Arab, “Qurban” yang berarti dekat. Di dalam ajaran Islam, qurban disebut juga dengan *al-udhhiyah* dan *adh-dhahiyah* yang berarti binatang sembelihan, seperti unta, sapi atau kerbau, dan kambing yang disembelih pada hari raya *Idul Adha* dan hari-hari *tasyriq* sebagai bentuk *taqqarubi* atau mendekatkan diri kepada Allah (Mulyana, 2016).

5. Android

Menurut (Yanto et al., 2017), *Android* adalah sebuah sekumpulan perangkat lunak untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi utama *mobile*. *Android* memiliki 4(empat) karakteristik sebagai berikut :

a. Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera, dan lain-lain. *Android* menggunakan sebuah mesin *virtual* yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang

terdapat di dalam perangkat. *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.

b. Semua Aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (*third-party application*). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

c. Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari *web* dengan data pada ponsel seseorang seperti kontak pengguna, kalender, atau lokasi geografis.

d. Pengembangan aplikasi yang cepat dan mudah

Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan *library* yang dipergunakan *tools* yang dapat digunakan untuk membangun aplikasi yang semakin baik. *Android* memiliki sekumpulan *tools* yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

Google Inc. Sepenuhnya membangun android dan menjadikan bersifat terbuka (*open source*) sehingga para pengembang dapat menggunakan android tanpa mengeluarkan biaya untuk lisensi dari *Google* dan dapat membangun android tanpa adanya batasan-batasan. *Android Software Development Kit (SDK)* menyediakan alat dan *Application Programming Interface (API)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*.

Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut :

a. Android Versi 1.1

Pada 9 Maret 2009, *Google* merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan email.

b. Android versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan Android dan *SDK (Software Development Kit)* dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengupload video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara

otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (*Donut*)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan, CDMA/EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel, pengadaan resolusi WVGA.

d. Android versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML₅, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, *Digital Zoom*, dan Bluetooth 2.1.

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

Pada bulan Mei 2010 Android versi 2.2 Rev 1 diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran, salah satunya adalah dipakai di Samsung FX tab yang sudah ada di pasaran.

Fitur yang tersedia di Android versi ini sudah kompleks di antaranya adalah:

- 1) Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia.
 - 2) *Dalvik Virtual Machine* dioptimalkan untuk perangkat mobile.
 - 3) Grafik: grafik di 2D dan grafis 3D berdasarkan *libraries* OpenGL.
 - 4) SQLite: untuk penyimpanan data.
 - 5) Mendukung media: audio, uideo, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
 - 6) GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware independent)
 - 7) Kamera, Global Positioning System (GPS), kompas, dan *accelerometer* (tergantung hardware).
- f. Android versi 2.3 (*Gingerbread*)

Android versi 2.3 diluncurkan pada Desember 2010, hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut:

- 1) *S/P-based VoIP*
- 2) *Near Field Communications* (NFC)
- 3) *Gyroscope* dan sensor
- 4) *Multiple cemerass support*
- 5) *Mixable audio effects*
- 6) *Dwonload manager*

g. *Android 3.0 (Honeycomb)*

Honeycomb diluncurkan pada bulan Mei 2011 pada dasarnya untuk memperluas *Android* untuk mendukung layar tablet. Versi *Android* ini paling diabaikan dari semua. Karena dirilis khusus untuk tablet dan tidak pernah sampai ke ponsel. Berikut adalah fitur yang menyertainya adalah beberapa perbaikan *UI* memanfaatkan layar besar, tombol perangkat keras diatuhkan untuk mendukung tombol layar, *browser web* mengenalkan *tabbed browsing*, *widget* yang lebih besar, aplikasi seperti *Gmail* dan *YouTube* dirancang ulang untuk menggunakan layar besar.

h. *Android 4.0 (Ice Cream Sandwich)*

Android Ice Cream Sandwich dirilis pada bulan Oktober 2011, versi *Android Ice Cream Sandwich* menghadirkan sejumlah fitur baru bagi pengguna. Ini menggabungkan banyak fitur versi *Honeycomb* tablet saja dengan *smartphone* yang berorientasi pada *Gingerbread*

Fitur yang menyertainya adalah desain ulang terbesar ke *android* dengan tema *holo*, pengalaman *browsing* lebih cepat, ruang penyimpanan multi untuk aplikasi, pengenalan wajah untuk membuka kunci telepon.

i. *Android 4.1 (Jelly Bean)*

Jellybean dimulai pada bulan Juni 2012 dengan merilis *android 4.1*. *Google* dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di

bawah label *Jelly Bean*, masing-masing pada bulan Oktober 2012 dan Juli 2013.

Berikut adalah fitur yang menyertainya *google now*, *tool* bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian, *project butter* untuk mendukung *frame rate* yang lebih tinggi saat menggesek menu dan layar rumah, mampu melihat foto dengan menggesek dari kamera untuk menuju ke *filmstrip*, *widget* menyetel ulang diri mereka untuk menambahkan yang baru, pemberitahuan fitur yang lebih banyak, fitur gerak dan aksesibilitas baru.

j. *Android 4.4 (KitKat)*

Android KitKat dirilis pada November 2013, *Android 4.4* adalah satu-satunya versi *OS* yang benar-benar menggunakan nama sepotong permen dan menjadi salah satu versi android yang paling disukai oleh pengguna *smartphone* di dunia.

KitKat memiliki fitur yang istimewa dari *OS* android sebelumnya. Fitur yang menyertainya adalah *immersive mode* untuk konsumsi konten yang lebih baik, *bar* navigasi yang lebih baik untuk masuk dan keluar dari *mode immersive*, dukungan *widget* layar kunci, dialler baru dengan fitur *Caller ID*, *wallpaper* layar penuh, emoji *keyboard* untuk *emoticon*, aplikasi *Hangouts* dan perpesanan terpadu, dukungan *cloud print* yang lebih baik, integrasi *Google Now* yang lebih cerdas dan *handsfree*.

k. *Android 5.0 (Lollipop)*

Android 5.0 Lollipop pertama kali diperkenalkan pada Mei 2014. *Android lollipop* merupakan perancangan ulang terbesar untuk *Android Smartphone Google Nexus 6*, bersama dengan tablet *Nexus 9*-nya, merupakan perangkat pertama yang memiliki *Lollipop* yang telah terpasang sebelumnya.

Peningkatan terbesar yang dilakukan oleh *Lollipop* adalah pengenalan *Material Design* yang dengan cepat menjadi bahasa desain terpadu yang diterapkan di seluruh produk *Google*. Fitur yang dimilikinya adalah dukungan pengaturan cepat yang lebih baik, masa pakai baterai yang disempurnakan dengan mode *battery, saver* yang baru, layar kunci baru, fitur *smart lock* melalui layanan *Google Play*, mode tamu untuk berbagi perangkat, pemasangan tombol.

l. *Android 6.0 (Marshmallow)*

Android 6.0 (Marshmallow) di rilis pada tahun 2015. Ini perangkat pertama yang dikirim bersama *Marshmallow* yang telah terpasang sebelumnya adalah *smartphone Google Nexus 6P* dan *Nexus 5X*, dengan tablet *Pixel C*-nya. Tujuan *marshmallow* memoles sudut kasar dan membuat versi *Lollipop* lebih baik lagi.

Fitur yang dimilikinya adalah dukungan sidik jari resmi untuk perangkat, dukungan untuk pembayaran seluler melalui *Android Pay*, model perizinan yang lebih baik untuk aplikasi, *Google Now* di Tap, deep menghubungkan *Apps*.

m. *Android 7.0 (Nougat)*

Android 7.0 (Nougat) Dirilis pada Tahun ,2016. Sebelum *Nougat* terungkap "*Android N*" dirujuk secara internal oleh *Google* sebagai "*New York Cheesecake*".

Fitur yang dimilikinya adalah *doze on the go* untuk waktu siaga yang lebih baik lagi, *multi window* untuk penggunaan dua aplikasi secara bersamaan, aplikasi setelan yang lebih baik, hapus semua di layar aplikasi baru-baru ini, balas langsung ke pemberitahuan, notifikasi dibundel, pengaturan cepat akan mengubah kustomisasi.

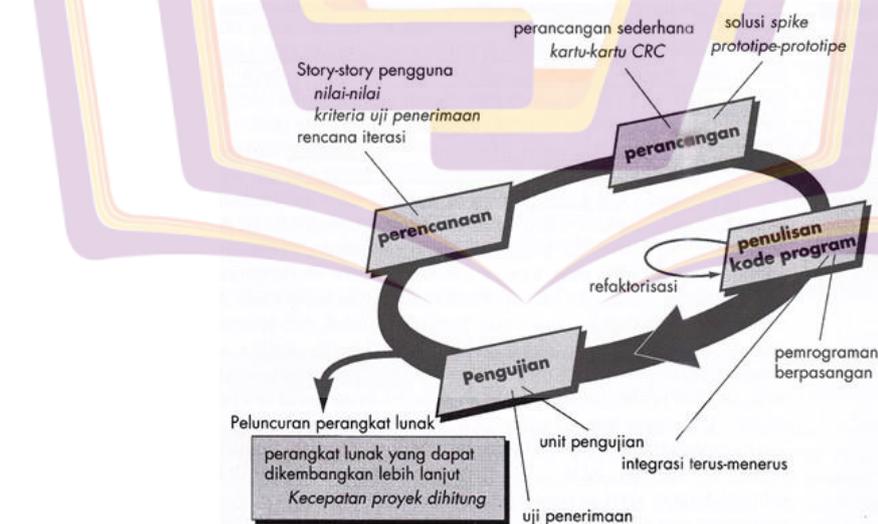
n. *Android 8.0 (Oreo)*

Pada bulan Maret 2017, *Google* rilis *Android 8.0 Oreo*, bulan Agustus, *Google* mengkonfirmasi *Oreo* akan menjadi nama publik untuk *Android 8.0*. Seperti yang kita ketahui ini adalah kedua kalinya *Google* memilih nama merek dagang untuk *android* (*Oreo* dimiliki oleh *Nabisco*).

Adapau sekarang versi ini adalah semua yang baru yang dimilikinya, diantaranya adalah pemberitahuan untuk prioritas dan kategorisasi yang lebih baik, pengelolaan warna lebih baik, *Android O* memiliki koleksi emoji baru yang telah didesain ulang, waktu *boot* lebih cepat, pada perangkat *Pixel*, sekarang bisa mengalami waktu *boot* dua kali lebih cepat dibandingkan dengan *Nougat*, mengisi otomatis dan mengingat kata sandi dalam aplikasi. Untuk informasi detail versi platform atau untuk mendapatkan informasi versi terbaru dari *android*, Anda dapat melihat di *website* dokumentasi resmi *developer android*.

6. Extreme Programming

Extreme Programming Menurut (Chandra, 2017) *Extreme Programming* merupakan salah satu model yang ada dalam pendekatan *Agile Process*. *Agile Process* dikenal dengan model yang interaktif dan incremental. Dalam pendekatan *Agile Process* suatu perangkat lunak dikembangkan dengan desain minimalis, pengujian bertahap, dan dokumentasi yang tidak berlebihan. Model *Extreme Programming* didefinisikan sebagai suatu metode ringan yang menekankan pada komunikasi yang intens, sehingga model pengerjaan yang interaktif dan *Incremental*. Di dalam pengembangan sistem yang menggunakan *Extreme Programming* terdapat empat tahapan yaitu *Planning*, *Design*, *Coding* dan *Testing*. Berikut adalah gambar tahapan *Extreme Programming*:



Gambar 2.1 Tahapan Extreme Programming
(Sumber: Roger S.Pressman, 2010)

Berikut ini tahapan-tahapan yang harus dilakukan dalam metode *Extreme Programming* dalam penelitian:

a. *Planning*

Pada tahapan ini dilakukan pembuatan *user story* (cerita) berdasarkan data yang diperoleh dari client. *User story* ini akan menjadi gambaran dasar dari sistem yang akan dikembangkan.

b. *Design*

Pada tahap desain dilakukan perancangan alur kerja sistem dan perancangan database berdasarkan *user story* yang telah dibuat sebelumnya.

c. *Coding*

Coding (pengkodean) merupakan tahap pembuatan sistem berdasarkan rancangan yang telah dibuat. Dalam pembuatan sistem ini dilakukan oleh dua orang yaitu seorang programmer dan seorang tester. Tahap ini dapat dilakukan secara berulang-ulang (*refactoring*) apabila terdapat koreksi.

d. *Testing*

Testing merupakan tahap pengujian sistem, setiap modul yang dikembangkan akan terlebih dahulu mengalami pengujian. Apabila masih belum sesuai permintaan, maka akan dilakukan perbaikan pada bagian yang dikoreksi. Jika sudah sesuai dengan permintaan maka sistem sudah dapat diimplementasikan.

7. Basis Data

Basis data merupakan sekumpulan data dalam sistem informasi dan tersusun table atau file. Mengingat bahwa sistem informasi menyajikan informasi berasal dari suatu maupun beberapa data yang diinputkan dan diolah, maka tentu diperlukan sebuah aplikasi untuk menyimpan, mengolah dan menyajikan data dan informasi secara terkomputerisasi (Rochman et al., 2019). Sedangkan menurut (Swara et al., 2016) Basis Data atau Database adalah kumpulan informasi yang disusun dan merupakan suatu kesatuan yang utuh yang disimpan di dalam perangkat keras (*computer*) secara sistematis sehingga dapat diolah menggunakan perangkat lunak. Dengan sistem tersebut data terhimpun dalam suatu database dapat menghasilkan informasi yang berguna.

Ada beberapa operasi basis data diantaranya :

- a. Pembuatan basis data baru (*create database*), yang identic dengan pembuatan lemari arsip yang baru.
- b. Penghapusan basis data (*drop database*), yang identic dengan perusak lemari arsip (sekaligus beserta isinya jika ada).
- c. Pembuatan file/table bar uke suatu basis data (*create table*), yang identik dengan penambahan map arsip bar uke sebuah lemari arsip yang telah ada.
- d. Penghapus file/table dari suatu baris data (*drop table*), yang dibentuk dengan perusakan map arsip lama yang ada di sebuah lemari arsip.

- e. Penambahan/pengisian data baru ke sebuah file/table di sebuah basis data (*insert*), yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
- f. Pengembalian data dari sebuah file/table (*retrieve/search*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
- g. Pengubah data dari sebuah file/table (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
- h. Penghapusan data dari sebuah file/table (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

B. Perangkat Lunak yang digunakan

1. *Firestore*

Firestore adalah sebuah layanan Baas (*Backend as a Service*) dari Google. Layanan ini menyediakan pengembang aplikasi API (*Application Programming Interface*) yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di cloud *Firestore* ini. *Firestore* menyediakan library untuk berbagai klien platform yang memungkinkan integrasi dengan *Android*, *iOS*, *JavaScript*, *Java*, *Objective-C* dan *Node.js* dan dapat juga disebut sebagai layanan DbaaS (*Database as a Service*) dengan konsep realtime. *Firestore* digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh developer (Abdurahman & Kurnia, 2019).

2. XML

XML adalah singkatan dari *eXtensible Markup Language*. Bahasa markup adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. XML terletak pada inti Web Service yang digunakan untuk mendeskripsikan data.

Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data dan aplikasi-aplikasi yang berbeda dengan partner luaran. Dengan XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lainnya (Wahyudi, 2019).

3. Android SDK

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Android merupakan subnet perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware*, dan aplikasi kunci yang di-release oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman java (Efendi, 2020).

4. Android Studio

Menurut (Karman & Mulyono, 2019). *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*.



Gambar 2.2 Logo Android Studio
Sumber: Karman & Mulyono, 2019)

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” diklaim dapat memonitor kecepatan, kegunaan, serta kompetabilitas aplikasi dengan cepat
- d. Mendukung *Proguard and App-Signing* untuk keamanan
- e. Memiliki GUI aplikasi Android lebih mudah
- f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan

5. JAVA

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari *Oracle* dan dirilis tahun 1995 (Pratama & Hermawan, 2016).

6. *Unified Modelling Language* (UML)

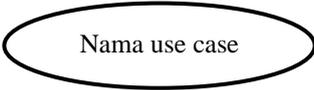
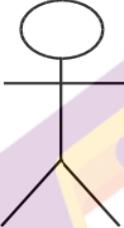
a. Pengertian UML

UML (*Unified Modelling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010).

b. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol pada diagram *use case* dapat dilihat pada tabel 2.1 (Rosa & Salahuddin, 2013).

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

Simbol	Keterangan
<p data-bbox="483 465 603 495"><i>Use case</i></p> 	<p data-bbox="885 465 1345 678">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
<p data-bbox="483 689 635 719">Aktor/actor</p> 	<p data-bbox="885 689 1345 969">Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan keta benda diawal frase nama actor.</p>
<p data-bbox="483 1070 762 1099">Assosiasi/association</p> 	<p data-bbox="885 1070 1345 1205">Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="483 1216 691 1245">Ekstensi/extend</p> <p data-bbox="571 1256 735 1285"><<extend>></p> 	<p data-bbox="885 1216 1345 1541">Relasi <i>use case</i> tambah ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>
<p data-bbox="483 1552 842 1581">Generalisasi/generalization</p> 	<p data-bbox="885 1552 1345 1731">Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.</p>

Tabel 2. 1 Simbol-Simbol *Use Case Diagram* (Lanjutan)

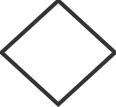
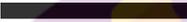
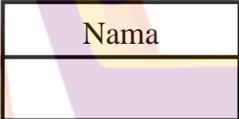
Simbol	Keterangan
<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> :</p> <ul style="list-style-type: none"> • Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan. • Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan.

(Sumber: Rosa & Salahuddin, 2013)

c. *Activity Diagram*

Menurut (Rosa & Salahuddin, 2013), *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.2.

Tabel 2.2 Simbol-Simbol Activity Diagram

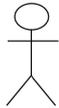
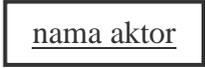
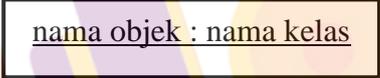
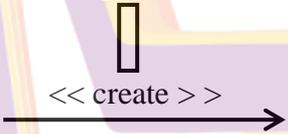
Simbol	Keterangan
Status awal 	Status awal aktifitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih 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 yang terjadi.

(Sumber: Rosa & Salahuddin, 2013)

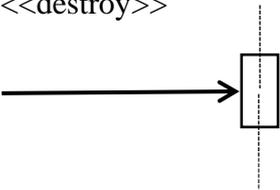
d. *Sequence diagram*

Menurut (Rosa & Salahuddin, 2013) *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram skuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.. Simbol-simbol *sequence diagram* dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol-Simbol *Sequence Diagram*

Simbol	Keterangan
<p>Aktor</p>  <p>Atau</p> <p>tanpa waktu aktif</p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
<p>Objek</p>	<p>Menyatakan objek yang berorientasi pesan.</p>
<p>Waktu aktif</p> 	<p>Menyatakan suatu objek dalam keadaan aktif dan berinteraksi pesan.</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe call</p> <p>1 : nama metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>

Tabel 2.4 Simbol-Simbol *Sequence Diagram* (Lanjutan)

Sumber	Keterangan
Pesan tipe destroy <<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

(Sumber: Rosa & Salahuddin, 2013)

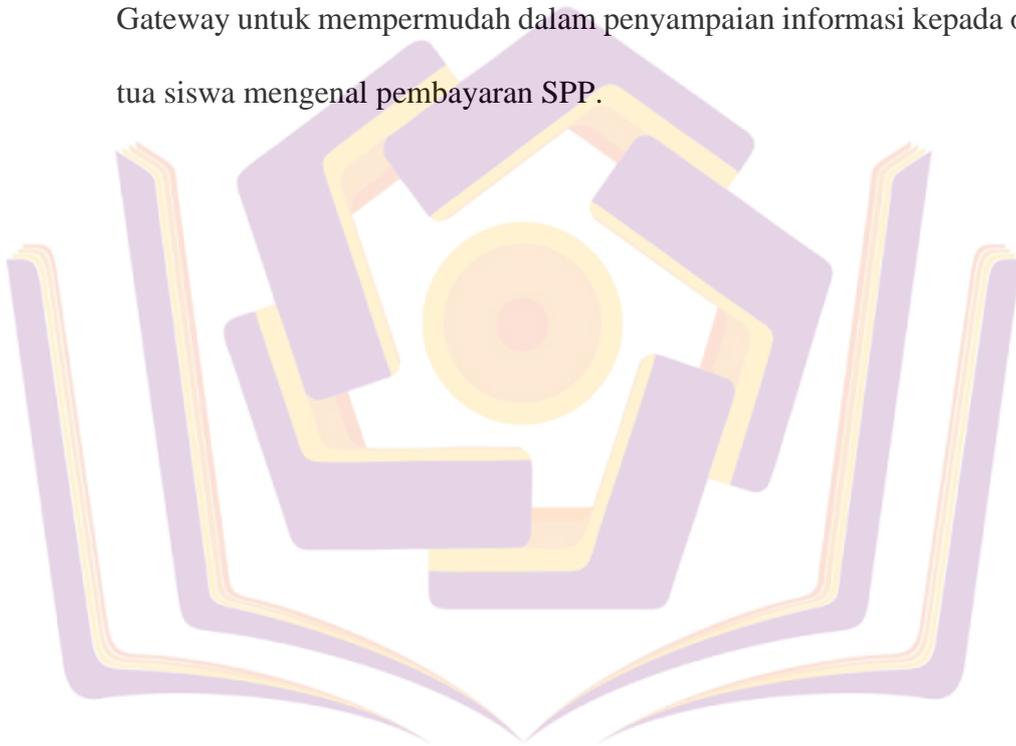
C. Penelitian Sebelumnya

1. Jurnal penelitian yang dilakukan (Sansprayada et al., n.d.) Universitas Bina Sarana Informatika, dengan judul “Implementasi Aplikasi Bank Sampah Berbasis *Android* Studi Kasus Perumahan Vila Dago Tangerang Selatan” dalam penelitian ini membahas tentang sistem informasi banks sampah modern berbasis *android* yang dapat menjadi solusi untuk permasalahan rentannya terjadi kehilangan data dan kurangnya efektifitas dalam menunjang kegiatan sehari-hari di perumahan vila dago Tangerang Selatan. Penelitian ini bertujuan untuk menerapkan Teknologi Informasi dan Komunikasi (TIK) sebagai komponen yang akan mempermudah pengelolaan dan peprtanggung jawaban setiap aktivitas yang terdapat di bank sampah kepada masyarakat. Metode penelitian yang digunakan adalah *Unified Modeling Languange* (UML). Hasil penelitian ini berupa sistem informasi bank sampah modern berbasis *android*.
2. Penelitian yang dilakukan (Bavany, 2017) AMIK BSI Pontianak, dengan judul “Pembuatan Aplikasi Tabungan Qurban Pada Baitulmall Pondok Moderen Munzalan Ashabul Yamin Kabupaten Kubu Raya berbasis *Web*”

dalam penelitian ini membahas tentang dalam pendataan tabungan nasabah qurban di baitul maal pondok modern munzalan ashabul yamin sering mengalami kesalahan, pengolahan data masih menggunakan excel, bagi nasabah untuk mengetahui saldo tabungan nasabah harus mengunjungi baitul maal secara langsung. Penelitian ini bertujuan membantu nasabah melakukan transaksi menabung dan mengetahui informasi saldo tanpa harus mengunjungi Baitul Maal secara langsung, serta membantu admin dalam melakukan pendataan nasabah. Metode yang digunakan dalam penelitian ini menggunakan metode *waterfall*. Hasil penelitian ini menghasilkan aplikasi tabungan qurban yang nantinya admin bisa mendata transaksi nasabah dalam satu sistem aplikasi serta dapat melakukan perhitungan secara otomatis.

3. Penelitian yang dilakukan oleh (Nisa et al., 2017) Universitas Telkom, dengan judul “Aplikasi Pembayaran SPP Dan Saldo Tabungan Berbasis Web Disertai SMS Gateway”. Dalam penelitian ini membahas tentang pembayaran SPP (Sumbangan Pembinaan Pendidikan) merupakan pembayaran iuran wajib perbulan bagi siswa yang bersekolah di sekolah swasta, begitu pula di MTS Mathla’ul Huda. Penelitian ini bertujuan membantu dalam pencatatan pembayaran, memfasilitasi dalam rekapsulasi data pembayaran SPP, mencegah adanya duplikasi data dan menyampaikan informasi kepada orangtua siswa mengenai tunggakan, pembayaran yang telah di lakukan siswa maupun konfirmasi apabila orangtua siswa telah menambahkan saldo tabungan siswa yang digunakan untuk pembayaran

SPP via SMS Gateway. Metode yang digunakan dalam penelitian ini adalah metode *System Development Life Cycle (SDLC)* model *waterfall*. Hasil penelitian ini menghasilkan aplikasi yang membantu pihak tata usaha dalam melakukan pendataan pembayaran SPP, menghitung rekapitulasi pembayaran SPP, laporan SPP lunas dan tunggakan dan penyimpanan saldo tabungan bagi siswa yang hendak menabung di sekolah serta fitur SMS Gateway untuk mempermudah dalam penyampaian informasi kepada orang tua siswa mengenal pembayaran SPP.



Tabel 2.5 Penelitian Sebelumnya

No.	Peneliti	Hasil Penelitian	Persamaan penelitian	Perbedaan Penelitian	
				Terdahulu	Sekarang
1.	Penelitian yang dilakukan (Sansprayada & Mariskhana, 2020)	sistem informasi bank sampah modern berbasis <i>android</i> .	Mengelola data nasabah dan rekapulasi transaksi	Tidak adanya fitur notifikasi pada aplikasi	Adanya fitur notifikasi pada aplikasi
2.	Penelitian yang dilakukan (Bavany, 2017)	menghasilkan aplikasi tabungan qurban yang nantinya admin bisa mendata transaksi nasabah dalam satu sistem aplikasi serta dapat melakukan perhitungan secara otomatis.	Mengelola data tabungan qurban	Tidak adanya kolom pencarian untuk lebih memudahkan admin dalam mencari data nasabah	Adanya fitur pencarian untuk memudahkan dalam pencarian data nasabah
3.	Penelitian yang dilakukan oleh (Nisa et al., 2017)	aplikasi yang membantu pihak tata usaha dalam melakukan pendataan pembayaran SPP, menghitung rekapitulasi pembayaran SPP, laporan SPP lunas dan tunggakan dan penyimpanan saldo tabungan bagi siswa yang hendak menabung di sekolah serta fitur SMS Gateway.	Mengelola data tabungan dan rekapulasi transaksi	<i>Database</i> menggunakan <i>MySQL</i>	<i>Database</i> menggunakan <i>firebase</i>

