

BAB II

KAJIAN LITERATUR

Kajian literatur merupakan teori-teori pendukung yang didapat dari kegiatan studi kepustakaan. Teori yang dijabarkan merupakan teori yang berkaitan dengan penelitian yang sedang dilakukan. Berikut ini merupakan penjabaran dari beberapa teori pendukung dalam penelitian ini.

2.1 Donasi Barang

Secara harfiah definisi donasi merupakan upaya memberi dengan tujuan amal dan membantu pihak lain yang membutuhkan. Segala bentuk donasi bersifat suka rela sehingga tidak ada jumlah yang ditentukan maupun harapan untuk mendapatkan imbalan (www.kompasiana.com). Donasi adalah sumbangan tetap (berupa uang) dari penderma kepada perkumpulan (KBBI). Donasi juga dapat berupa bantuan kemanusiaan saat terjadinya bencana, dalam hal medis dapat berupa donor darah/*transfuse* darah, sehingga bentuk pemberian donasi dapat berupa barang atau jasa (Anggraini, 2018). Pemberi donasi disebut donatur.

Jadi, donasi adalah gerakan kemanusiaan dengan memberi bantuan secara suka rela kepada orang-orang yang membutuhkan seperti para korban bencana alam, anak yatim piatu, dan sebagainya. Donasi dapat dikatakan sebagai pertolongan yang diberikan oleh para dermawan. Barang adalah segala sesuatu yang berwujud atau berjasad (KBBI). Barang memiliki nilai guna. Nilai guna dari barang ini yang dibutuhkan oleh para calon penerima donasi. Jadi, donasi barang adalah gerakan kemanusiaan dengan memberi bantuan secara suka rela dalam bentuk barang. Donasi tidak hanya berupa uang melainkan dapat berupa barang. Barang yang diberikan merupakan barang yang masih layak pakai.

2.2 Aplikasi Mobile

Perangkat lunak aplikasi atau *application software* terdiri dari beberapa program yang didesain untuk membuat penggunaanya lebih produktif dan/atau membantu dalam melakukan tugas (Irwansyah & Moniaga, 2014). Program aplikasi merupakan *software* yang banyak digunakan untuk membantu menyelesaikan tugas tertentu. Aplikasi *mobile* adalah aplikasi perangkat lunak yang dibuat khusus untuk

dijalankan di dalam tablet dan juga *smartphone*. Aplikasi *mobile* memiliki karakteristik yang berbeda dengan aplikasi lainnya. Aplikasi *mobile* yang baik mampu menyajikan informasi dan layanan secara efisien dan praktis (Tolle, Pinandito, Kharisma, & Dewi, 2017). Jadi, aplikasi *mobile* adalah perangkat lunak yang dibuat dengan program-program aplikasi untuk tujuan tertentu, yang dijalankan khusus di dalam tablet atau *smartphone*.

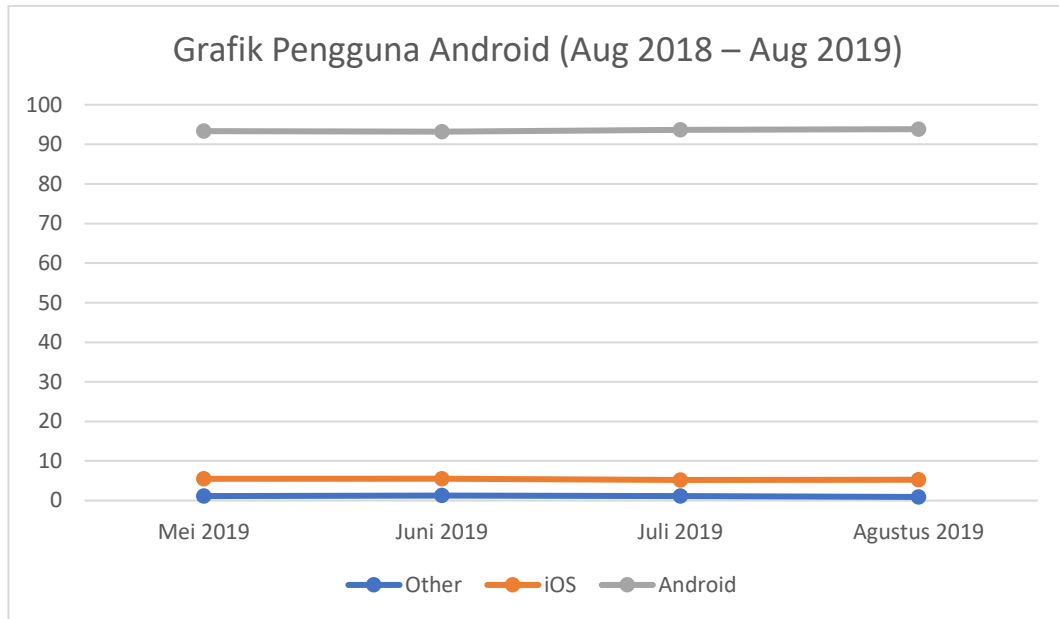
Perkembangan aplikasi *mobile* sangat bergantung terhadap perkembangan *smartphone*. Terdapat tiga pendekatan yang digunakan untuk mengembangkan aplikasi *mobile*, yaitu (Prasetya, 2013):

- a. Aplikasi *Native*, adalah aplikasi yang secara khusus ditujukan untuk *platform mobile* tertentu dan menggunakan bahasa pemrograman serta perangkat lunak pengembangan sesuai *platform*.
- b. Aplikasi Web, merupakan aplikasi *website* yang secara spesifik dioptimalkan untuk penggunaan di lingkungan *smartphone*. Aplikasi ini dibangun menggunakan standar teknologi-teknologi web, seperti HTML5, CSS3, dan JavaScript.
- c. Aplikasi *Hybrid*, intuisi dari aplikasi ini adalah menanamkan aplikasi *mobile* HTML5 ke dalam *container native*. Pendekatan ini akan mengonversi aplikasi web *mobile* HTML5 ke aplikasi *native smartphone* target.

2.2.1 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone*. Android adalah sistem operasi *open source*, dan Google merilis kodenya di bawah Lisensi Apache.

Android secara sederhana bisa diartikan sebagai sebuah *software* yang digunakan pada perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci yang dirilis oleh Google. Sehingga Android mencakup keseluruhan sebuah aplikasi dari mulai sistem operasi sampai pengembangan aplikasi tersebut. Pengembangan aplikasi pada *platform* android menggunakan dasar bahasa pemrograman Java. (Enterprise, Mengenal Dasar-Dasar Pemrograman Android, 2015)



Gambar 2.1 Grafik Pengguna Android (Aug 2018 – Aug 2019)

Sumber: StatCounter, 2019

Di Indonesia sendiri sistem operasi android paling banyak digunakan oleh para pengguna *smartphone*. Bisa dilihat pada gambar 2.1 di bulan Agustus 2019 sebanyak 93,85% konsumen memilih *platform* Android untuk sistem operasi *mobile* mereka (StatCounter, 2019).

2.2.2 Framework React Native

Framework adalah kerangka kerja. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer/programmer* dalam menangani berbagai masalah-masalah pemrograman, sehingga pekerja *developer* lebih fokus dan lebih cepat dalam membangun aplikasi (Yudhanto & Prasetyo, 2019). Jadi, *framework* adalah kumpulan perintah dasar untuk membantu *developer* dalam membangun aplikasi.

Banyak macam *framework*, salah satunya yaitu *react native*. *React native* adalah *framework* javascript untuk membuat aplikasi *mobile*, seperti iOS dan Android (AMC College, 2017). *React native* adalah *platform* yang gratis untuk mengembangkan aplikasi *mobile native*, sebagian besar dikembangkan oleh Facebook. Bagian yang keren bekerja menggunakan *react native* adalah aplikasi

yang dibangun seluruhnya *native* (Paul & Nalwaya, 2019). Fitur pada *react native*, yaitu (AMC College, 2017):

- React*, ini adalah *framework* untuk membangun aplikasi web dan *mobile* menggunakan JavaScript.
- Native*, dapat menggunakan *native components controlled* dari JavaScript.
- Platform*, mendukung *platform* iOS dan Android.

Keuntungan menggunakan *react native* yaitu (AMC College, 2017):

- JavaScript, dapat menggunakan pengetahuan JavaScript yang ada untuk membangun aplikasi seluler.
- Code sharing*, dapat membagikan sebagian besar kode di berbagai *platform*.
- Community*, banyak komunitas *react* dan *react native*, sehingga memudahkan ketika ingin bertanya mengenai kendala dan masalah yang dihadapi selama proses *develop* aplikasi menggunakan *react native*.

	 Xamarin	 NativeScript	 React Native	 Flutter
Year Introduced	2011	2014	2015	2018
Backed by	Microsoft	Telerik	Facebook	Google
Presentation language	XAML and/or xamarin.forms	Proprietary but looks like XML	Proprietary but looks like JSX	Dart
Procedural Language	C#	JavaScript	JavaScript	Dart
Still need to know some truly native development	Very high	High	High	Low
Current popularity	High • 34.6K StackOverflow questions • (Not on Github)	Very low • 3.8K StackOverflow questions • 15.8K Github stars	High • 44K StackOverflow questions • 72.4K Github stars	Low • 8.8K StackOverflow questions • 48.2K Github stars
Recent popularity trend	Slightly decreasing	Flat	Increasing	Increasing
Cost	Teams of > 5 must buy a license to Visual Studio	Free	Free	Free

Gambar 2.2 Perbandingan *Xamarin*, *NativeScript*, *React Native* dan *Flutter*

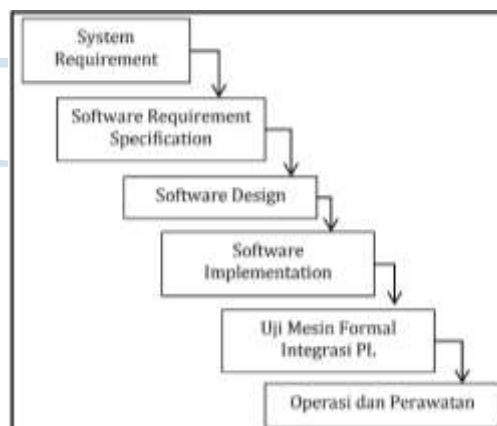
Sumber: Sophia, 2019

2.3 Waterfall

Metode pengembangan sistem merupakan serangkaian aktivitas selama proses pengembangan suatu sistem. Salah satu metode pengembangan sistem yaitu *waterfall model*. Model *waterfall* sebenarnya adalah “*Linear Sequential Model*”, yang sering juga disebut dengan “*classic life cycle*”. Metode ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing*, dan *maintenance* (Muharto & Ambarita, 2016). *Waterfall* merupakan salah satu metode dalam *Software Development Life Cycle* (SDLC) yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya (Yurindra, 2017). Jadi, *waterfall* adalah metode pengembangan sistem dengan pendekatan secara sistematis dan berurutan, tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Penggunaan *Waterfall* sebagai metode pengembangan sistem memiliki kekurangan dan kelebihan. Kekurangan dari metode *waterfall* yaitu sifatnya kaku, sehingga susah melakukan perubahan di tengah proses. Selain itu, metode *waterfall* membutuhkan daftar kebutuhan yang lengkap di awal. Kelebihan metode *waterfall*, yaitu (Yurindra, 2017):

- a. Proses menjadi lebih teratur, urutan proses pengerjaan tahap demi tahap.
- b. Setiap proses tidak dapat saling tumpang tindih.
- c. Jadwal menjadi lebih menentu, jadwal setiap proses dapat ditentukan secara pasti sehingga dapat dilihat jelas target penyelesaian pengembangan program.



Gambar 2.3 Tahapan Metode *Waterfall*

Sumber: Yurindra, 2017

SDLC identik dengan teknik pengembangan sistem *waterfall*, karena tahapannya menurun dari atas ke bawah (Mulyani, 2016). Tahapan dari metode *waterfall*, sebagai berikut (Yurindra, 2017):

- a. *Requirement analysis*, seluruh kebutuhan program aplikasi harus bisa didapatkan dalam fase ini termasuk batasan program aplikasi. Informasi biasanya diperoleh melalui wawancara, diskusi, atau survei.
- b. *System design*, tahap ini bertujuan untuk memberikan gambaran yang seharusnya dikerjakan, tampilan program aplikasi, serta mendefinisikan arsitektur sistem secara keseluruhan.
- c. *Implementation*, tahap ini dilakukan pemrograman atau *coding*.
- d. *Integration and testing*, tahap pengujian setelah pemrograman aplikasi selesai.
- e. *Operation and maintenance*, ini merupakan tahap terakhir dalam model *waterfall*. Pemeliharaan termasuk dalam memperbaiki kesalahan yang ditemukan pada langkah sebelumnya.

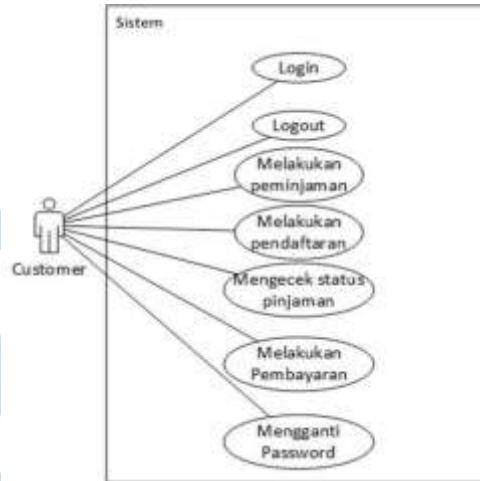
2.4 Unified Modelling Language (UML)

UML adalah alat bantu untuk pemodelan sistem. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Hanief & Pramana, 2018). UML (*Unified Modelling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggambarkan diagram dan teks-teks pendukung (Nurdiana, 2018). UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S & Shalahuddin, 2014). Jadi, UML adalah bahasa atau alat bantu untuk menggambarkan rancangan sebuah sistem dalam bentuk diagram. Terdapat beberapa jenis diagram pada UML. Berikut ini contoh beberapa jenis diagram UML:

- a. *Use Case Diagram*

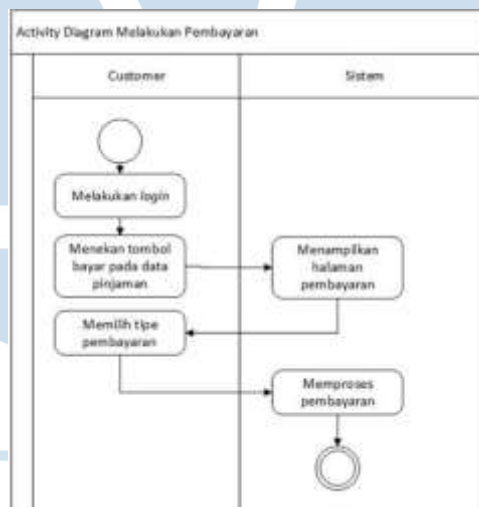
Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Sukamto & Shalahuddin, 2014). *Use case* merupakan suatu diagram yang berisi *use case*, *actor*, serta *relationship*

diantaranya. *Use case* merupakan titik awal yang baik dalam memahami dan menganalisis kebutuhan sistem pada saat perancangan (Indrajani, 2015). Jadi, *use case diagram* adalah gambaran dari interaksi sistem dan *user*.



Gambar 2.4 Contoh *Use Case Diagram*
(Sumber: Abiezer & Kandaga, 2019)

b. *Activity Diagram*

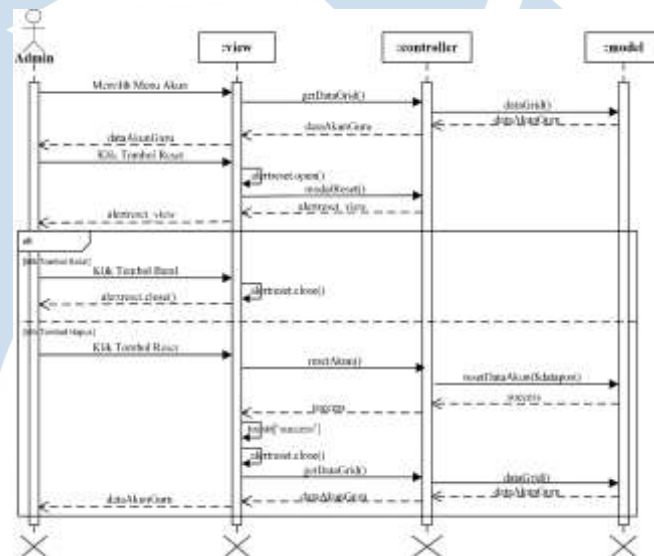


Gambar 2.5 Contoh *Activity Diagram*
(Sumber: Kurniati & Jaroji, 2018)

Activity diagram menggambarkan aliran kerja dari sebuah sistem (Sukamto & Shalahuddin, 2014). *Activity diagram* digunakan untuk menganalisis *behavior* dengan *use case* yang lebih kompleks dan menunjukkan interaksi-interaksi di antara satu sama lain (Indrajani, 2015). Jadi, *activity diagram* adalah aliran kerja sebuah sistem sesuai dengan *use case*.

c. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (Sukamto & Shalahuddin, 2014). *Sequence diagram* merupakan suatu diagram interaksi yang menggambarkan bagaimana objek-objek berpartisipasi dalam bagian interaksi (*particular interaction*) dan pesan yang ditukar dalam urutan waktu (Indrajani, 2015). *Sequence diagram* adalah diagram yang menggambarkan serangkaian interaksi internal sistem.

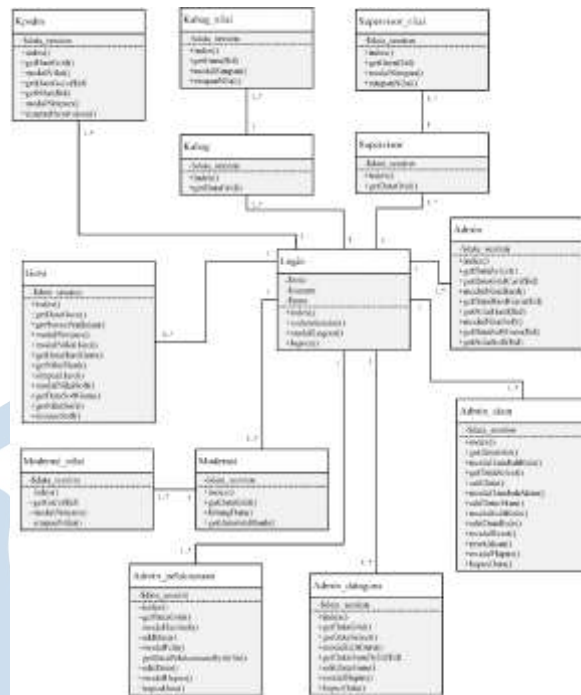


Gambar 2.6 Contoh *Sequence Diagram*

(Sumber: Hanifatulqolbi, 2018)

d. *Class Diagram*

Class diagram digunakan untuk menggambarkan perbedaan yang mendasar antar *class*, hubungan antar *class*, dan dimana sub sistem *class* tersebut (Indrajani, 2015). *Class diagram* menggambarkan suatu sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Sukamto & Shalahuddin, 2014). Jadi, *class diagram* adalah diagram yang menggambarkan struktur *class* dan *package* untuk membangun sistem.



Gambar 2.7 Contoh *Class Diagram*
(Sumber: Hanifatulqolbi, 2018)

2.5 Database

Database atau basis data adalah sebuah kumpulan data yang saling berhubungan secara logis dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah organisasi (Indrajani, 2015). Basis data atau (*database*) adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi (Kadir, 2014). *Database* adalah kumpulan file yang saling berkaitan. Jadi, *database* adalah kumpulan data yang saling berhubungan. Kegunaan *database* adalah untuk mengatasi masalah-masalah pada penyusunan data. Masalah-masalah pada penyusunan data antara lain (Kristanto, 2015):

- a. Redundansi dan inkonsisten data
- b. Kesulitan pengaksesan data
- c. Isolasi data untuk standarisasi
- d. *Multiple user* (banyak pemakai)
- e. Masalah keamanan
- f. Masalah integrasi
- g. Masalah data *independence*

Salah satu jenis *database server* yaitu MySQL. MySQL adalah sebuah *software database*. MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan (Winarno & Zaki, 2014). MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*Database Management System*). MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*) (Huda, 2015). Jadi, MySQL merupakan *software database* yang sering digunakan karena dapat diakses secara gratis.

Ada beberapa model data dalam sistem basis data. Salah satu jenis model data yaitu ER Model atau *Entity Relationship Diagram* (ERD). ERD adalah *tools* yang digunakan untuk memodelkan struktur data dengan menggambarkan entitas dan hubungan antar entitas (*relationship*) secara abstrak (konseptual) (Mulyani, 2016).

2.6 RESTful API

Web service adalah sistem perangkat lunak untuk menyediakan suatu layanan kepada sistem lain dari berbagai *framework* dan *platform*. Terdapat beberapa jenis *web service*, salah satunya yaitu RESTful *web service*. RESTful *web service* atau juga dikenal dengan nama RESTful Web API merupakan sebuah *web service* yang diimplementasikan menggunakan http dengan prinsip-prinsip REST. *Service* yang digunakan menggunakan *method* milik http antara lain GET, PUT, POST or DELETE (Kurniaji, 2016).

REST adalah pola desain untuk berinteraksi dengan sumber daya yang disimpan di *server*. Empat metode HTTP utama menentukan empat operasi yang umumnya dilaksanakan oleh RESTful *web service*. Metode HTTP POST digunakan untuk membuat sumber data, GET digunakan untuk query, PUT digunakan untuk mengubah data, dan DELETE digunakan untuk menghapus data.

2.7 Pengujian

Pengujian perangkat lunak dilakukan pada tahap akhir pengembangan, setelah seluruh produk perangkat lunak selesai (Galim, 2018). Pengujian dilakukan untuk menjamin kualitas aplikasi dan untuk memastikan aplikasi berjalan sesuai dengan *requirement* yang telah ditentukan.

2.7.1 Black Box Testing

Black box testing adalah pengujian yang mengabaikan pemeriksaan mekanisme internal sistem dan hanya berfokus pada *output* yang dihasilkan sebagai *respon* terhadap masukan dan kondisi eksekusi. *Tester* tidak harus berurusan dengan kode (Sharma & Sangwan, 2016).

Black box testing cenderung untuk menemukan hal seperti fungsi yang tidak benar atau tidak ada, kesalahan antarmuka, kesalahan pada struktur data dan akses basis data, kesalahan performansi, dan kesalahan inisialisasi dan terminasi (Rahayuda, 2017). Jadi, teknik *black box testing* adalah teknik pengetesan yang tidak berurusan dengan kode dan berfokus pada hasil dari respon suatu proses dalam sebuah sistem, pengetesan dilakukan dengan cara mengetes antarmuka, atau mengetes setiap proses yang ada di sebuah sistem.

2.7.2 User Acceptance Test

User Acceptance Test (UAT) merupakan uji penerimaan yang dilakukan oleh pemilik proses bisnis dan pengguna. Suatu sistem informasi dikatakan dapat diterima apabila telah lulus dari UAT (Ramdani, 2018). UAT juga disebut pengujian beta (*beta testing*) atau pengujian pengguna akhir (*end user testing*). UAT adalah tahapan pengembangan perangkat lunak ketika perangkat lunak diuji pada dunia nyata yang dimaksud oleh pengguna. UAT biasanya dilakukan dengan mendistribusikan perangkat lunak secara luas dengan melakukan pengujian versi yang tersedia. Pengalaman awal pengguna akan diteruskan kembali kepada para pengembang yang membuat perubahan sebelum akhirnya melepaskan perangkat lunak komersial.

Tujuan dari pengujian ini adalah untuk mengetahui apakah perangkat lunak sudah memenuhi harapan pengguna dan bekerja sesuai dengan yang diharapkan (Simarmata, 2010). Jadi, UAT adalah pengujian sistem yang dilakukan oleh pengguna untuk mengetahui apakah perangkat lunak sudah memenuhi harapan atau sesuai dengan harapan pengguna sebelum perangkat lunak diserahkan kepada pengguna.

2.7.3 Skala Likert

Skala *likert* adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, penerbit sebuah laporan yang menjelaskan penggunaannya (Maryuliana, 2016).

Sewaktu menanggapi pertanyaan dalam skala *likert*, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format pada tabel 2.1.

Tabel 2.1 Skala Jawaban

Skala Jawaban	Skor
Sangat Setuju (SS)	5
Setuju (S)	4
Cukup (C)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Kemudian, berdasarkan skala tersebut dilakukan perhitungan kriterium yang dinormalisasi menggunakan rumus berikut:

$$Skor\ kriterium = \frac{nilai\ skala\ jawaban}{nilai\ skala\ jawaban\ terbesar} \times 100 \quad (1)$$

Setelah diperoleh nilai kriterium masing-masing skala, selanjutnya ditentukan nilai *rating scale*. Untuk nilai *rating scale* ditentukan nilai batas bawah terendah yaitu 1 karena saat pengisian angket semua pertanyaan harus dijawab. Kemudian untuk nilai batas atas dan bawah masing-masing skala, yaitu sebagai berikut:

Batas atas = skor kriterium
Batas bawah STS = batas bawah terendah
Batas bawah TS = batas atas STS + 1
Batas bawah C = batas atas TS + 1
Batas bawah S = batas atas C + 1
Batas bawah SS = batas atas S + 1

Untuk mendapatkan hasil di setiap pertanyaan menggunakan rumus berikut:

$$Hasil = frekuensi\ kemunculan\ jawaban \times nilai\ skala \quad (2)$$

Kemudian, total hasil akhir dihitung dengan normalisasi menggunakan rumus berikut:

$$Skor\ Akhir = \frac{Total\ hasil\ tiap\ titik}{nilai\ skala\ jawaban\ terbesar \times Jumlah\ responden} \times 100 \quad (3)$$

Setelah didapatkan hasil akhir, maka dapat kita interelasikan skornya berdasarkan interval sebagai berikut

0% – 19,99%	= Sangat (tidak setuju/buruk/kurang sekali)
20% – 39,99%	= Tidak setuju / Kurang baik)
40% – 59,99%	= Cukup / Netral
60% – 79,99%	= (Setuju/Baik/suka)
80% – 100%	= Sangat (setuju/Baik/Suka)

Metode ini dipilih untuk penelitian ini karena sesuai dengan kebutuhan peneliti yang akan membuat kuisisioner dengan memberikan 5 opsi pilihan dengan penilaian tiap opsi yang berbeda-beda.

2.8 Penelitian Terkait

Dalam penelitian ini, penulis melakukan studi kepustakaan dengan penelitian yang sudah ada sebelumnya. Tujuan dilakukan hal ini yaitu untuk melakukan perbandingan dengan penelitian yang sudah ada sebelumnya. Selain itu, agar penulis tidak melakukan penelitian yang sama dengan penelitian sebelumnya. Daftar penelitian tersebut bisa dilihat pada Tabel 2.2.

Tabel 2.2 Penelitian Terkait

Nama	Judul Penelitian	Tahun	Kesimpulan
Andika Kurnia Adi Pradana	Pembuatan Aplikasi Berbasis <i>Crowdsourcing</i> dalam Upaya Penanggulangan Penduduk Miskin	2016	Aplikasi dibuat dalam bentuk <i>website</i> , menggunakan bahasa pemrograman PHP. <i>Framework</i> yang digunakan adalah <i>CodeIgniter</i> . Proses bisnis aplikasi yang dibuat adalah kegiatan donasi. Donasi yang diberikan berupa uang. Donasi ditujukan untuk membantu penduduk miskin.
Deni Fadjri, Ilhamsyah, Dian Prawira	Rancang Bangun Sistem Informasi Pengumpulan Dana Panti Asuhan menggunakan Metode <i>Crowdfunding</i> dengan Model Situs Donasi	2018	Aplikasi dibuat dalam bentuk <i>website</i> , menggunakan pemrograman PHP dengan <i>framework</i> Yii2. Aplikasi dibuat untuk penggalangan donasi bagi panti asuhan. Donasi yang diberikan berupa uang.
Yuris Socio Perestroika, Adam Hendra Brata, dan Yusi Tyroni Mursityo	Pengembangan Sistem Informasi Penggalangan Donasi pada Yayasan Gerakan Nurani Orang Tua Asuh (GN-OTA) Kabupaten Lumajang	2018	Aplikasi yang dibuat pada penelitian ini berbasis web, menggunakan <i>framework</i> <i>CodeIgniter</i> , dengan model MVC. Aplikasi ini dibuat untuk penggalangan donasi dan publikasi bagi yayasan GN-OTA. Donasi yang diberikan berupa uang.

Tabel posisi penelitian bertujuan untuk memperlihatkan perbedaan penelitian yang dilakukan dengan penelitian terkait. Pada table 2.3 dapat diperlihatkan posisi penelitian yang berjudul Rancang Bangun Aplikasi Donasi Barang Berbasis *Mobile* Menggunakan *Framework React Native* dengan penelitian yang terkait.

Tabel 2.3 Posisi Penelitian






Judul	Pembuatan Aplikasi Berbasis <i>Crowdsourcing</i> dalam Upaya Penanggulangan Penduduk Miskin	Rancang Bangun Sistem Informasi Pengumpulan Dana Panti Asuhan menggunakan Metode <i>Crowdfunding</i> dengan Model Situs Donasi	Pengembangan Sistem Informasi Penggalangan Donasi pada Yayasan Gerakan Nurani Orang Tua Asuh (GN-OTA) Kabupaten Lumajang	Rancang Bangun Aplikasi Donasi Barang Berbasis <i>Mobile</i> Menggunakan <i>Framework React Native</i>
Bentuk Donasi	Donasi Uang	Donasi Uang	Donasi Uang	Donasi Barang
Proses Donasi	Melalui transfer bank, lalu bukti transfer di- <i>upload</i>	Melalui transfer bank, lalu bukti transfer di- <i>upload</i>	Melalui transfer bank, lalu bukti transfer di- <i>upload</i>	Barang dijemput, dengan melalui proses verifikasi barang oleh <i>campaigner</i>
Riwayat donasi	Tidak ada riwayat donasi	Tidak ada riwayat donasi	Tidak ada riwayat donasi	Ada riwayat donasi

STT - NF

Campaigner	Setiap orang bisa membuat <i>campaign</i>	Setiap orang bisa membuat <i>campaign</i>	Setiap orang bisa membuat <i>campaign</i>	Hanya <i>user</i> yang didaftarkan oleh admin sebagai <i>campaigner</i>
Platform	Web	Web	Web	<i>Mobile</i>
Framework	<i>CodeIgniter</i>	Yii2	<i>CodeIgniter</i>	<i>React Native</i>

Selain menunjukan posisi penelitian untuk mengetahui perbedaan penelitian yang dilakukan dengan penelitian yang terkait terdapat juga table perbandingan aplikasi. Pada tabel 2.4 menunjukan perbandingan aplikasi kitapunya sebagai aplikasi pada penelitian ini dengan beberapa aplikasi donasi berbasis *online* yang ada di Indonesia.

Tabel 2.4 Perbandingan Dengan Aplikasi Lain

					
Platform	Web	Web, <i>Mobile</i>	Web, <i>Mobile</i>	Web	Web, <i>Mobile</i>
Donasi	Barang	Uang	Uang	Uang	Barang
Pembuat Campaign	Admin	Akun terverifikasi	Akun terverifikasi	Admin	Akun yang mempunyai hak akses sebagai <i>campaigner</i>
Pemberian Donasi	Dijemput	Transfer	Transfer	Transfer	Dijemput
Penerima Manfaat	Yayasan Cahaya Keluarga Fitrah	Masyarakat	Masyarakat	Masyarakat	Masyarakat