

BAB II

LANDASAN TEORI

2.1 Bisnis Budidaya Sarang Walet

Budidaya sarang walet adalah bisnis mengembangbiakkan burung walet yang ikut serta mengambil keuntungan dengan menjual sarangnya. Salah satu syarat keberhasilan dalam pembudidayaan burung walet adalah mengetahui habitat makro dari burung walet (Saepudin, 2006). Habitat makro mempunyai peranan yang penting dalam pembudidayaan burung walet dan sangat mempengaruhi dalam pemilihan serta penentuan lokasi bangunan (Marzuki, 1994).

Pemilihan bangunan rumah burung walet sangat mendukung keberhasilan pengoperasian gedung burung walet. Dalam pemilihan lokasi, pemilik harus memperhatikan beberapa faktor lingkungan yang cocok untuk kehidupan burung walet, berdasarkan sifat dan secara alami disukai oleh burung walet (Alhaddad, 2003). Bangunan yang digunakan untuk budidaya sarang walet dibuat sedemikian rupa agar memiliki suasana mirip seperti yang ada di dalam goa yaitu gelap dan lembab. Setelah membuat bangunan maka dibutuhkan alat berupa tape recorder yang digunakan untuk memanggil burung walet datang bersarang di dalam bangunan.

Burung walet berkembang biak setiap tahun, dimana proses berkembang biak burung walet tidak jauh berbeda dengan burung lainnya. Sarang burung walet dibuat dalam waktu 40-80 hari, di dalamnya terdapat telur burung walet yang berjumlah 2 butir, telur-telur tersebut dierami selama 23-24 hari. Burung walet biasanya mengeram dalam koloni atau secara berpasangan. Tidak lama setelah telur menetas, anakan walet langsung dipelihara oleh kedua induknya hingga bisa terbang (Iswanto, 2002). Ketika sarang sudah ditinggalkan maka proses panen akan dilakukan.

Dalam melakukan panen ada cara khusus yang harus dilakukan untuk bisa mengambil sarang walet sehingga dibutuhkan tenaga yang lebih berpengalaman bertujuan untuk memastikan kelayakan sarang apakah sudah siap dipanen atau belum dan juga memastikan ketika panen sarang tersebut tidak mengalami kerusakan karna proses pemisahan sarang dari dinding gedung yang membutuhkan tenaga memungkinkan bahwa sarang tersebut bisa saja patah ataupun hancur, hal tersebut harus sangat diperhatikan karena hasil panen akan diseleksi berdasarkan bentuk sarang, hal ini yang disebut grade pada sarang walet sehingga hasil bentuk dari panen akan sangat berpengaruh terhadap nilai jual.

2.2 Sistem Monitoring

Monitoring adalah proses pengawasan atau pemantauan terhadap kinerja pada suatu sistem yang bertujuan untuk bisa mengukur dan mempertimbangkan apakah sebuah program telah berjalan sesuai rencana. Kegiatan monitoring lebih terfokus pada kegiatan yang akan dilaksanakan. Monitoring dilakukan dengan cara menggali untuk mendapatkan informasi secara reguler berdasarkan indikator tertentu, dengan maksud mengetahui apakah kegiatan yang sedang berlangsung sesuai dengan perencanaan dan prosedur yang telah disepakati. Indikator monitoring mencakup esensi aktivitas dan target yang ditetapkan pada perencanaan program. Apabila monitoring dilakukan dengan baik akan bermanfaat dalam memastikan pelaksanaan kegiatan tetap pada jalurnya. Juga memberikan informasi kepada pengelola program apabila terjadi hambatan dan penyimpangan, serta sebagai masukan dalam melakukan evaluasi.

Sistem informasi monitoring budidaya sarang walet merupakan suatu hal yang penting bagi pemilik usaha bisnis untuk memastikan seluruh proses bisnis yang berjalan sesuai dengan yang diharapkan, oleh karena itu monitoring usaha harus dilakukan secermat mungkin. Pemilik dapat menentukan dan memperhitungkan hasil produksi yang didapat mengalami peningkatan atau penurunan sehingga bisa memperhitungkan penyebab dari masalah yang terjadi. Begitu juga dengan penjualan, pemilik bisa mengetahui bahwa cabang usaha mana yang mengalami penurunan atau bahkan peningkatan hasil penjualan hanya dengan menggunakan aplikasi.

Jadi dapat disimpulkan bahwa sistem manajemen bisnis budidaya sarang walet menjadi suatu solusi yang bertujuan untuk menghasilkan informasi yang dapat memberikan dukungan terhadap monitoring usaha pada bisnis ini.

2.3 Android

Android merupakan sistem operasi untuk telpon seluler yang berbasis Linux. Seperti yang diketahui Google merupakan pengembang dibalik keberhasilan android dalam melakukan inovasi yang begitu cepat. Google merupakan perusahaan yang mengakuisisi Android dan dibuatkan sebuah *platform*. *Platform* pada android terdiri dari sistem operasi berbasis Linux, GUI (*Graphic User Interface*) serta aplikasi pada *web browser* dan aplikasi pada *mobile* yang dalam diunduh serta membebaskan pengembang untuk ikut serta dalam membuat sebuah aplikasi yang bisa digunakan oleh banyak orang.

Selain Smartphone Perkembangan android sendiri sudah mencakupi beberapa keluaran produk yang dikembangkan oleh Google yaitu Android TV untuk televisi pintar, Android auto

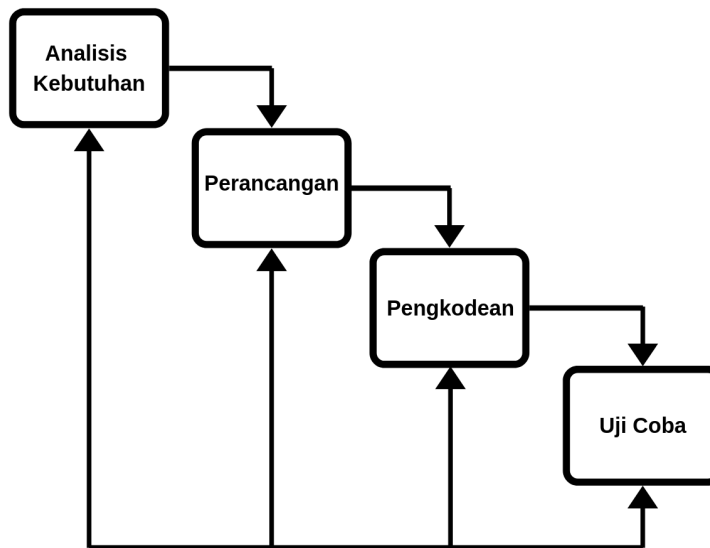
untuk mobil serta Android Wear untuk jam tangan dan produk lainnya dengan menampilkan antarmuka yang khusus untuk masing-masing produk.

Android merupakan sistem operasi yang *open source* yang berlisensi Apache yang bisa dikembangkan dan didistribusikan secara luas dan bebas. Dengan memanfaatkan *open source* membuat pengembang sistem bisa merancang dan membuat sistem Android menggunakan *software* yang telah didukung oleh Android.

2.4 Metode Waterfall

Waterfall atau bisa diartikan dengan air terjun adalah sebuah metode pengembangan sistem yang dimana memiliki beberapa fase dalam penyelesaiannya yang harus dilakukan secara berurutan. Dalam mengimplementasikan metode *waterfall* ini sebuah fase atau langkah harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan berikutnya.

Menurut (Sommerville, 2004), Metode *waterfall* memiliki beberapa tahapan utama dari *waterfall* model yang mencerminkan aktifitas pengembangan dasar, terdapat lima tahapan pada metode *waterfall*, yaitu analisis kebutuhan sistem, desain dan perancangan, implementasi, dan uji coba.



Gambar 2.1 Mode Pengembangan Metode *Waterfall*

2.4.1 Analisis Kebutuhan

Analisis Kebutuhan sistem adalah proses pengumpulan data yang sebelumnya didapatkan melalui wawancara terhadap klien yang telah menjelaskan dan mendefinisikan tujuan dari sistem yang dibangun. Komunikasi ini bertujuan agar pengembang bisa memahami sistem yang diharapkan oleh pengguna untuk difokuskan dalam pembuatan aplikasi.

2.4.2 Perancangan

Rancangan merupakan proses memusatkan kerja pada perancangan sistem yang diharapkan, pada tahapan ini akan menggunakan *tools* untuk merancang, *Use Case Diagram*, perancangan *database*, perancangan antarmuka.

1. *Use Case Diagram*

Use Case Diagram adalah sebuah diagram yang menggambarkan proses interaksi sistem dengan pengguna. Dengan menggunakan *Use Case Diagram* maka secara grafis dapat menggambarkan siapa yang menggunakan sistem dan bagaimana pengguna berinteraksi dengan sistem (Whitten et al., 2007). *Use Case Narrative* merupakan tentang kejadian bisnis dan bagaimana seorang pengguna akan berinteraksi dengan sistem untuk menyelesaikan suatu tugas. Adapun beberapa komponen yang digunakan sebagai berikut:

a. *Use-Case*

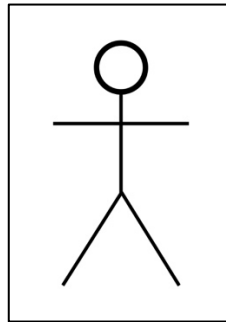
Use-Case digambarkan dalam bentuk *elips* yang merupakan gambaran suatu aktifitas sistem yang digunakan oleh aktor seperti Gambar 2.2.



Gambar 2.2 Simbol *Use Case*

b. Aktor (*actor*)

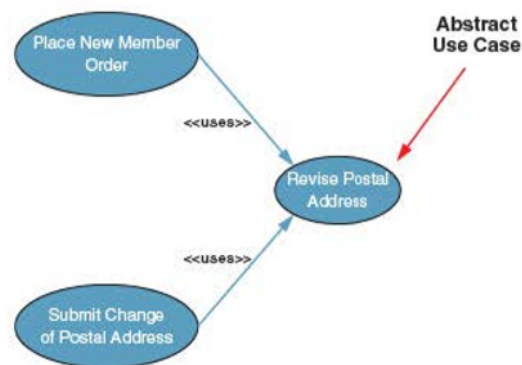
Aktor adalah segala sesuatu yang berinteraksi dengan sistem untuk pertukaran informasi. Pelaku disimbolkan seperti pada Gambar 2.3.



Gambar 2.3 Simbol Aktor

c. Hubungan (*relationship*)

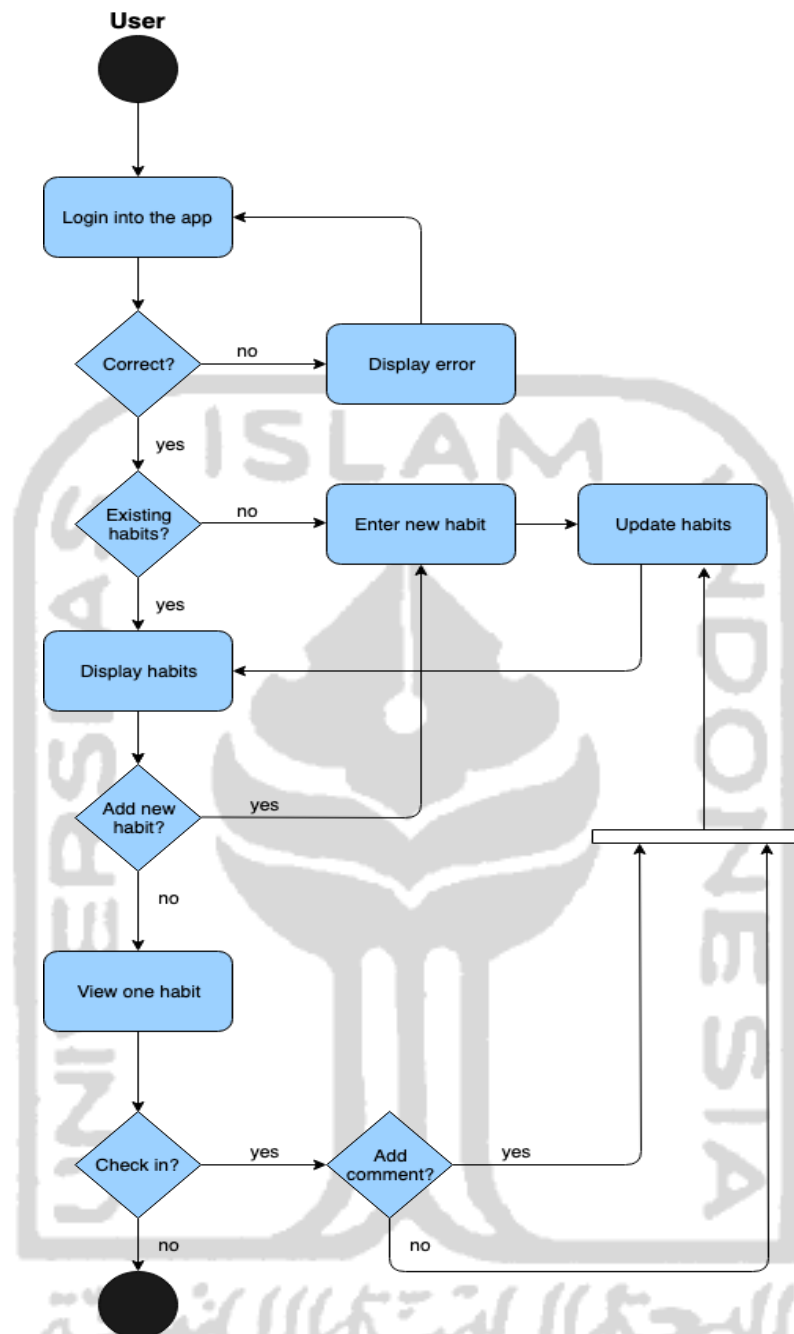
Hubungan digambarkan bermacam-macam garis yang memiliki anak panah yang menghubungkan antar dua buah simbol yaitu aktor dan use case. Hubungan disimbolkan seperti pada Gambar 2.3.



Gambar 2.4 Simbol Hubungan

2. Activity Diagram

Activity diagram merupakan rangkaian aliran dari suatu aktivitas yang berurutan dan dibentuk dalam suatu operasi sehingga dapat digunakan untuk aktivitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memiliki model yang mengalir dari satu status ke status lainnya. Namun mekanisme pada *activity diagram* dibuat secara paralel. Pembuatan *activity diagram* pada awal pemodelan membantu memahami keseluruhan proses.



Gambar 2.5 Activity Diagram

a. *Initial Node*

Berbentuk lingkaran, menggambarkan dimulainya sebuah proses.

b. *Flow*

Tanda panah pada diagram yang menunjukkan perkembangan dengan tindakan. Kebanyakan tidak disertai kata-kata untuk mengidentifikasinya.

c. *Action*

Berbentuk panjang dengan sisi melengkung, menggambarkan setiap tahapan. Urutan tindakanya dibuat sebanyak total aktivitas yang ditampilkan oleh diagram.

d. *Decision*

Berbentuk belah ketupat dengan 1 aliran masuk dan 2 atau lebih aliran keluar. Aliran yang dibuat untuk mengidentifikasi kondisi.

e. *Merge*

Berbentuk belah ketupat dengan 2 atau lebih aliran masuk dan 1 aliran keluar, digunakan untuk menyatukan yang dipisahkan oleh decision.

f. *Join*

Berbentuk balok hitam panjang yang memiliki 2 atau lebih aliran masuk dan 1 aliran keluar. Seluruh tindakan yang ada harus selesai sebelum melanjutkan proses selanjutnya.

g. *Fork*

Berbentuk balok hitam panjang yang memiliki 1 aliran masuk dan 2 aliran atau lebih keluar, berperan untuk memecah alur paralel dalam sebuah aktivitas.

h. *Activity Final*

Berbentuk lingkaran penuh di dalam sebuah lingkaran lain yang menandakan berakhirnya suatu *Activity Diagram*.

3. Perancangan Antarmuka (UI)

Perancangan antarmuka merupakan gambaran visual dari sebuah aplikasi untuk memastikan bagaimana *user* berinteraksi dengan sistem serta bagaimana informasi yang disampaikan di dalamnya. Tujuan dari perancangan antarmuka sendiri ialah untuk meningkatkan *Usability* dan *User Experience* dalam mengenal aplikasi. Perancangan antarmuka akan dirancang dengan menggunakan *tools* Adobe XD

2.4.3 Implementasi

Setelah melakukan perancangan sistem, maka hasil perancangan tersebut diimplementasikan dan diterjemahkan ke dalam bahasa pemrograman. Pada tahapan ini akan diimplementasikan menggunakan perangkat berupa laptop dan *gadget* yang disediakan peneliti.

2.4.4 Pengujian

Pengujian dilakukan untuk menguji apakah sistem yang dibuat sudah berfungsi dengan baik dan sudah sesuai dengan apa yang telah diharapkan sebelumnya. Pengujian meliputi pengujian fungsionalitas menggunakan *Black box testing* dan pengujian terhadap pengguna dengan memberikan kuisioner untuk mengetahui respon terhadap aplikasi yang digunakan, untuk menampilkan persentase keberhasilan akan digunakan metode pengujian Skala Likert.

2.5 Teknologi

Teknologi merupakan beberapa *tools* yang digunakan untuk dukungan dalam membangun sebuah aplikasi, pada proses pengembangan aplikasi berbasis android maka akan dibutuhkan beberapa teknologi untuk membangun aplikasi tersebut, pada tugas akhir ini pengembang akan menggunakan beberapa teknologi yang mendukung dalam pembuatan aplikasi ini.

2.5.1 Android Studio

Android studio adalah *Integrated Development Environment* (IDE) yang diluncurkan secara resmi oleh android untuk pengembangan aplikasi android, yang diresmikan oleh google pada tahun 2013.

Sebagai media yang menjadi pengembangan dari Eclipse, Android Studio tentunya sudah dilengkapi dengan gaya baru serta mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan *Gradle* sebagai *build environment*. Android Studio dipilih karena memiliki banyak fitur yang mudah untuk digunakan para pembuat program terutama programmer level dasar. Selain memiliki banyak fitur, Android Studio juga memiliki banyak *library* yang dapat digunakan. Walaupun Android Studio banyak menghabiskan memori, tetapi hal ini dapat ditutupi dengan kelebihan - kelebihan yang dimiliki oleh Android Studio itu sendiri.

Android Studio menggunakan bahasa Java yang menawarkan banyak keunggulan di dalamnya untuk meningkatkan produktifitas dalam membangun aplikasi android, seperti:

1. Emulator yang kaya dengan fitur

Android studio memiliki fitur berupa emulator yang di desain sama seperti halnya handphone android yang memiliki banyak fitur di dalamnya salah satunya memiliki sistem operasi yang lengkap, hal ini sangat menguntungkan karena pengembang tidak harus menggunakan perangkat keras lainya seperti handphone.

2. *Instan Run*

Instan run merupakan salah satu fitur dari android studio yang melakukan *compile* dan *run* lebih cepat. namun *compile* ini ketika pertamakali dilakukan akan lebih lambat, hal ini disebabkan ketika pertamakali digunakan android studio akan membuat file APK terlebih dahulu dan setelah itu android studio hanya menjalankan file APK yang sebelumnya sudah dibuat sehingga proses *compile* akan terasa cepat.

2.5.2 Microsoft Visual Studio

Visual studio adalah aplikasi konsol perangkat lunak yang digunakan untuk proses pengkodean dalam pengembangan sebuah aplikasi. Visual studio sudah mencakup *compiler* berupa C++, Visual Basic dan beberapa *compiler* lainnya serta sudah mencakup *Software Development Kit* (SDK) dan *Integrated Development Environment* (IDE). Dengan menggunakan aplikasi visual studio ini, akan bisa membangun aplikasi GUI, aplikasi *console*, aplikasi web dan aplikasi berbasis *mobile*.

2.5.3 React Native

React Native adalah sebuah *framework* dari *javascript* yang dikembangkan oleh facebook dan dirilis pada tahun 2015 untuk memudahkan *developer* dalam mengembangkan aplikasi berbasis Android dan IOS.

Dengan React Native, pengembang diberikan kesempatan untuk membangun aplikasi *mobile* yang hampir menyerupai aplikasi native mobile apps, tidak seperti *web mobile apps* atau *framework hybrid apps* lain yang performanya bisa dikatakan jauh dari harapan penggunaannya. Karena melalui React native, kode dalam bentuk *Javascript* tersebut akan dikompilasi ke bentuk kode native menggunakan masing-masing kompilernya seperti Java untuk Android atau Obj C untuk IOS menyesuaikan *platform* dimana aplikasi akan dijalankan.

2.5.4 Google Cloud Platform

Google Cloud *Platform* (GCP) adalah kumpulan layanan komputasi *cloud* yang dimiliki oleh Google. Produk GCP ini memiliki infrastruktur yang sama dengan apa yang digunakan oleh Google untuk berbagai macam produk yang dimiliki oleh Google, seperti Google Search, Youtube dan Gmail. Adapun layanan yang disediakan GCP sendiri yaitu komputasi, penyimpanan data, analisis dan *learning machine*. Agar bisa menggunakan layanan ini pengguna harus menyediakan kartu kredit atau rekening bank untuk melakukan sewa terhadap layanan GCP.

2.5.5 Navicat

Navicat adalah *software* untuk melakukan *remote database* secara GUI, dimana pengguna dapat mengkonfigurasi *database* secara *grafic* atau dengan secara instan tanpa harus

menggunakan *script*. Kita bisa *manage* beberapa *database server* di aplikasi ini, seperti Oracle, MySQL Server dll.

Software ini bekerja dengan MySQL *Database Server* dari versi 3.32 atau di atasnya, dan mendukung sebagian besar fitur terbaru MySQL termasuk Fungsi *Event*, *Stored* Prosedur, *Trigger*, *Manage User*, *View* dll.

2.5.6 Postman

Postman adalah sebuah *software* yang sering digunakan oleh developer untuk menguji data pada API, dengan menggunakan postman pengguna akan diberikan kemudahan ketika ingin melakukan *request* terhadap API.

Adapun kelebihan beberapa fitur yang dapat digunakan untuk dapat menyesuaikan data api adalah:

1. *Http method*

Dengan menggunakan fitur ini, semua request *http* telah disediakan sesuai kebutuhan pengguna dalam melakukan *request* data API.

2. Parameter dalam *header*

Jika api ingin melakukan penambahan parameter yang perlu dimasukkan ke dalam *header* fitur ini dapat digunakan.

3. *File* dalam *parameter body*

Fitur ini digunakan ketika ingin menguji data *body table* pada *database*, tidak hanya data berupa *teks*, data berupa foto yang digunakan pengguna untuk disimpan di *server*.

4. *History*

Setiap *request* yang telah dikirimkan akan dibaca oleh postman dan akan masuk ke dalam *history*. Tujuan utamanya adalah kita dapat memanggil kembali request tersebut tanpa harus mengetik kembali *http method* yang ingin digunakan.

5. *Share request* atau *collection*

Fitur ini merupakan kelebihan tersendiri bagi postman yaitu kita bisa membagikan *request* yang telah kita buat kepada orang lain atau ke dalam sebuah tim. Dan seluruh *request* bisa disimpan ke dalam sebuah folder yang disebut *collection* agar lebih rapi dan praktis ketika *request* ingin digunakan

2.5.7 Bitbucket

Bitbucket adalah sebuah layanan *hosting* yang berbasis web untuk kode sumber dan pembangunan proyek yang menggunakan *Mercurial* ataupun sistem kendali versi Git yang dimiliki oleh Atlassian. Bitbucket menawarkan paket akun komersial dan gratis. Akun gratis tersebut menawarkan sebuah layanan repositori dengan jumlah yang tidak terbatas (yang mana dapat mempunyai 5 pengguna per tim untuk akun gratis) sejak bulan September 2010. Bitbucket terintegrasi dengan perangkat lunak Atlassian lain seperti Jira, HipChat, Confluence dan Bamboo.

Hal ini mirip dengan GitHub, yang utamanya menggunakan Git. Bitbucket secara tradisional disesuaikan untuk mempermudah pengembang profesional dengan kode pribadi mereka, terutama sejak diakuisisi oleh Atlassian pada tahun 2010. Pada bulan September 2016, Bitbucket mengumumkan telah mencapai 5 juta pengembang dan 900.000 tim pada *platform* tersebut. Bitbucket mempunyai 3 model *deployment*: Cloud, Bitbucket Server dan Data Center.

2.5.8 Expo

Expo adalah *software* atau *tools* yang digunakan dalam pengembangan aplikasi, expo merupakan *software* yang bisa di *install* di *smartphone* sebagai emulator sehingga pengguna dapat melihat *preview project* yang dibuat menggunakan framework react native. Untuk melakukan koneksi expo menggunakan jaringan yang terhubung satu jaringan dengan *smartphone* sehingga tanpa menggunakan kabel USB untuk melakukan *preview*. Keuntungan lain yang didapatkan adalah expo *live refresh* yaitu ketika melakukan *save* pada *project* react native yang dibangun maka otomatis tampilan yang ada di expo ikut berubah tanpa melakukan *refresh* manual, selain itu expo juga memiliki fitur *publish* yaitu pengguna dapat melakukan pembuatan APK pada aplikasi sehingga tidak perlu repot lagi untuk menggunakan *software* lain untuk melakukan *publish* pada aplikasi.

2.6 Metode Pengujian

2.6.1 Pengujian *Black Box*

Black box testing adalah pengujian yang dilakukan dengan mengamati hasil pengujian melalui data uji dan memeriksa fungsionalitas dari perangkat lunak. Digambarkan seperti pengguna *input* ke dalam sistem kemudian sistem memproses dan sistem akan merespon

dengan *output* sesuai perintah *input* pengguna. Pada proses sistem itulah dianalogikan *black box* tersebut. Jadi pengujian menggunakan metode *black box* menguji fungsionalitasnya tanpa mengetahui apa sesungguhnya yang terjadi pada proses detilnya atau hanya mengetahui *input* dan *output*. Kelebihan dari metode *black box* ini adalah dapat memilih *subse test* secara efektif dan efisien

2.6.2 Skala Likert

Skala likert merupakan salah satu metode penghitungan terhadap penilaian, sikap, pendapat dan persepsi seseorang dalam sebuah variabel penelitian. Variabel yang diukur menjadi sebuah indikator yang kemudian dijadikan sebagai titik tolak untuk menyusun item instrument berupa pernyataan atau pertanyaan (Sugiyono, 2010).

Jawabannya mempunyai gradasi dari sangat positif sampai sangat negatif kemudian dapat diberi skor, misalnya Sangat Setuju dengan skor 5, Setuju dengan skor 4, Ragu-ragu dengan Skor 3, Tidak Setuju dengan skor 2 dan Sangat Tidak Setuju dengan skor 1 (Sugiyono, 2010) Tabel 2.1 merupakan contoh perhitungan kuesioner dengan skala likert dalam bentuk *checklist*.

Tabel 2.1 Skala Likert bentuk *checklist*

No	Pertanyaan	Jawaban				
		SS	S	R	TS	STS
1	Perusahaan ini akan menggunakan teknologi informasi dalam melakukan pengawasan terhadap karyawan		√			
2					

Kemudian dari kuesioner tersebut diberikan kepada 100 orang karyawan dengan hasil 25 orang menjawab SS, 40 orang menjawab S, 5 orang menjawab R, 20 orang menjawab TS dan 10 orang menjawab STS. Selanjutnya dianalisis dengan menghitung rata-rata pada setiap jawaban responden, dengan perhitungan sebagai berikut:

$$\text{Jumlah skor SS} = 25 \times 5 = 125$$

$$\text{Jumlah skor S} = 40 \times 4 = 160$$

$$\text{Jumlah skor R} = 5 \times 3 = 15$$

$$\text{Jumlah skor TS} = 20 \times 2 = 20$$

$$\text{Jumlah skor STS} = 10 \times 1 = 10$$

$$\text{Total} = 350 \quad (2.1)$$

Idealnya jika seluruh responden menjawab SS maka skor bernilai $5 \times 100 = 500$, sedangkan total skor yang diperoleh pada penelitian adalah 350. Berdasarkan data tersebut maka tingkat persetujuan responden terhadap pertanyaan dari kuesioner diatas adalah $(350 : 500) \times 100\% = 70\%$ dari yang diharapkan.

