BAB II

TINJAUAN PUSTAKA

II.1 Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa kontribusi penelitian yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu :

Pada penelitian terkait yang dilakukan oleh Marleni Anike, Francisco C, J. De Melo; (2019) Tingkat kecepatan dalam melakukan *reload* pada *websiteserver* membutuhkan waktu 2 detik, dimana setiap 2 detik apabila laporan baru masuk admin tidak perlu lagi melakukan aksi *loadwebsite* secara manual. Dengan adanya aplikasi AEPb ini yang dapat di akses melalui teknologi *mobile* menggunakan *platform android*, dimana aplikasi ini berguna untuk memudahkan pengguna bila mengalami musibah atau keadaan *emergency* yang membutuhkan tindakan pertolongan secepatnya oleh para medis. Selain itu aplikasi ini memudahkan pihak rumah sakit untuk dapat bergerak cepat dalam melakukan tindakan pertolongan kepada pengguna yang berada di luar wilayah rumah sakit dikarenakan aplikasi ini telah dilengkapi dengan fungsi GPS (*maps*) dimana pihak rumah sakit dapat langsung menuju ke arah tempat tujuan dengan cepat tanpa harus berputar-putar jalannya.

Pada penelitian terkait yang dilakukan oleh Abd. Wahab Syahroni, Ubaidi; (2019). Berdasarkan hasil uji coba aplikasi menggunakan metode *blackbox*, semua fungsionalitas aplikasi dapat berjalan dengan baik dan aplikasi ini layak untuk digunakan di desa berdasarkan indeks kepuasan menggunakan skala likert dengan nilai 70,8% yang berarti responden sangat setuju.

Pada penelitian terkait yang dilakukan oleh Indri Neforawati, dkk (2019). Sistem notifikasi ke aplikasi Telegram berjalan dengan baik, dimana notifikasi diterima ketika terdapat kondisi yang diterima dari data base untuk dikirimkan ke notifikasi Telegram, apabila terdapat kondisi yaitu, air di akuarium menunjukan nilai. Waktu pengiriman notifikasi dari data base ke aplikasi telegram juga menunjukkan jeda waktu yang tidak terlalu lama. Pengiriman notifikasi dilakukan selama 1 menit sekali dengan kondisi yang sudah ditentukan. Sistem Notifikasi pada otomasi akuaponik ini berbasis Android dan baru di implementasikan pada aplikasi Telegram. Akan lebih baik, jika sistem ini juga dikembangkan di aplikasi Android lainnya, berbasis IOS maupun web.

Pada penelitian terkait yang dilakukan oleh David Witanis Kholidin, dkk (2018). Perancangan database riwayat perawatan dapat menampilkan suatu database riwayat perawatan motor diesel secara realtime menggunakan android untuk mengetahui hasil perawatan sebelumnya dan sebagai tolak ukur pelaksana selanjutnya. Data dapat diintegrasikan antara perangkat dengan website sebagai media penyimpanan data yang dapat

diolah dan dijadikan draft hasil perawatan. Sinkronisasi aplikasi dapat dilakukan pada lebih dari dua perangkat.

Pada penelitian terkait yang dilakukan oleh Mochammad Junus (2017). Apliksi perawatan panel pendukung berbasis android dapat dapat meminimalisir waktu serta menghemat biaya perawatan, hal ini dapat dilihat dari berjalannya aplikasi ketika teknisi selesai melakukan perawatan, data hasil perawatan yang di unggah oleh user, dapat langsung masuk ke web server yang di tandai dari pesan yang diterima oleh admin, kemudian di konfirmasi oleh admin.

Berdasarkan hasil penelitian yang terdahulu, maka dibuatlah kesimpulan untuk merancang dan mengembangkan sebuah aplikasi untuk mengimplementasikan sebuah aplikasi pengembangan yang belum pernah dibuat atau dibangun oleh orang lain berbasis *Android*. Sehingga pada penulisan skripsi ini dibuatlah sebuah judul *"***Rancang Bangun Aplikasi Jasa Penyediaan Pelayanan Perawatan Perbaikan Menggunakan Metode Agile Berbasis Android pada PT. Otani***"*. Adapun yang membedakan pada penelitian ini adalah aplikasi ini dilengkapi suatu button panic yang berfungsi sebagai tombol penyedia jasa layanan perawatan dan perbaikan yang dapat terhubung langsung antara devisi produksi dan devisi perawatan.

II.2 Landasan Teori

II.2.1 Sistem

Sistem dapat di definisikan dengan pendekatan prosedur dan dengan pendekatan komponen :

1. Dengan pendekatan prosedur, sistem dapat di definisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.
2. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainya membentuk satu kesatuan untuk mencapai tujuan tertentu. Dari pengertian diatas dapat di simpulkan bahwa sistem adalah suatu kumpulan dari suatu proses yang saling memiliki ketergantungan dan memiliki suatu tujuan tertentu.

Karakteristik sistem merupakan suatu sistem yang mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah dan sasaran atau tujuan. Adapun penjelasan dari masing-masing karakteristik sistem adalah sebagai berikut :

1. Komponen sistem

   terdiri dari sejumlah komponenyang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen bagian sistem.

2. Batasan Sistem

   Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainya atau dengan lingkungan luarnya. Batas

sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan dan menunjukan ruang lingkup dari sistem tersebut.

3. Lingkungan Luar Sistem

   Lingkungan luar dari suatu sistem adalah apapun diluar batas sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan.

4. Penghubung Sistem

   Penghubung merupakan media yang menghubungkan antara satu subsistem dengan subsistem lainya, melalui penghubung ini kemungkinan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainya.

5. Masukan Sistem

   Masukan sistem adalah tenaga yang dimasukkan ke dalam sistem, masukan dapat berupa perawatan dan masukkan sinyal maintenance input adalah energy yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal input adalah energy yang diproses untuk mendapatkan keluaran dari sistem.

6. Keluaran Sistem

   Keluaran sistem adalah energy yang diolah dan di klasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukkan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem mempunyai tujuan atau aturan, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan (Riskiono & Reginal, 2018).

1. II.2.2. Sistem Informasi

Sistem informasi diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain (Geovanne Farell, dkk: 2018).

II.2.3 Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk membantu melayani kebutuhan seseorang untuk melakukan aktivitas. Dengan adanya aplikasi maka kebutuhan akan pelayanan sebuah aktivitas menjadi lebih baik. Dimana setiap pekerjaan dapat dilakukan dengan mudah kalau menggunakan sebuah aplikasi (Agusdi Syafrizal: 2018).

II.2.4 Perawatan

Tindakan pemeliharaan barang agar dapat berfungsi dengan baik selama umur teknis. Tindakan ini dapat mencakup tindakan pemeriksaan, penyetelan ulang sistem, pembersihan, atau penggantian bagian, komponen, dan/atau asesoris yang cepat aus (rusak) (Badan Standarisasi Nasional ;2017).

II.2.5 Perbaikan

Tindakan memperbaiki suatu barang yang tidak berfungsi agar dapat kembali berfungsi dengan baik. Tindakan ini mencakup pemeriksaan terhadap kerusakan, penyetelan ulang, pembersihan atau penggantian bagian, komponen dan/atau asesoris yang tidak memenuhi fungsinya (Badan Standarisasi Nasional ;20017).

II.2.6 *Agile Development Methods*

Agile Method adalah sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan interaktif, dimana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir. Istilah ini diciptakan pada tahun 2001 ketika Agile Manifesto dirumuskan. Metode *Agile* umumnya mempromosikan displin proses dan adaptasi, filosofi kepemimpinan yang mendorong kerja sama dalam bentuk tim, pengorganisasian dan akuntabilitas, pratktek rekayasa yang memungkinkan pengiriman perangkat lunak berkualitas tinggi dengan cepat, dan

pendekatan bisnis yang sejalan dengan pengembangan kebutuhan pelanggan dan tujuan perusahaan (Paisal; 2020).

II.2.7 Android

Android adalah aplikasi sistem operasi untuk telepon seluler yang berbasis Linux.Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak (Deny Adhar; 2019).

II.2.8 Versi Android

1. Android 1.5 Cupcake

Cupcake dirilis 30 April 2009. Cupcake menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 Cupcake adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. Android 1.6 Donut

Android V1.6, codename Donut, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan reboot, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan toping diatasnya.

3. Android 2.0/2.1 Eclair

Android 2.0/2.1 Eclair Dirilis 26 Oktober 2009. Eclair adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. Android 2.2 Froyo

Dirilis 20 Mei 2010. Menggunakan codename Froyo, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari Yoghurt. Froyo singkatan dari Frozen Yoghurt, Froyo adalah yoghurt yang telah mengalami proses pendinginan,sehingga secara terlihat sama seperti es krim.

5. Android 2.3 Gingerbread

Android versi 2.3 Gingerbread dirilis resmi tanggal 6 Desember 2010. Gingerbread merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. Android 3.0 Honeycomb

Dirilis tanggal 22 February 2011. H adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, Honeycomb/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. Android 4.0 Ice Cream Sandwich

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2001. Dinamai Ice Cream Sandwich. Ice Cream

Sandwich es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. Android 4.1 Jelly bean

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis Linux Kernel dari Android 4.1 API Level 16, Android 4.2 API Level 17 , Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. Android 4.4 KitKat

Android 4.4 Kitkat API level 19.Google mengumumkan Android KitKat (dinamai dengan izin Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. KitKat merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai ‘ Pie’ (Hendra Nugraha Lengkong; 2016).

2. II.2.9. *Android Studio*

Android Studio merupakan salah satu penyedia *software* yang saat ini banyak digunakan para *developer* untuk membuat atau mengembangkan sebuah program seperti *android project*.

Software ini dibuat oleh pihak *Google* yang berbasis *IntelliJ IDEA*. *Android* Studio dari tahun ke mengalami peningkatan keunggulan seperti

dalam segi tampilan yang dapat multi *screen* sehingga *developer* lebih mudah dapat menulis kode. Selain itu, *Android Studio* juga mempunyai fitur *fleksible gradle* yang mana berfungsi untuk membuat banyak *APK* sekaligus dengan fitur aneka macam meskipun menggunakan *code* yang sama dalam pembuatannya [1].

Gambar II.1. *Template Android Studio*

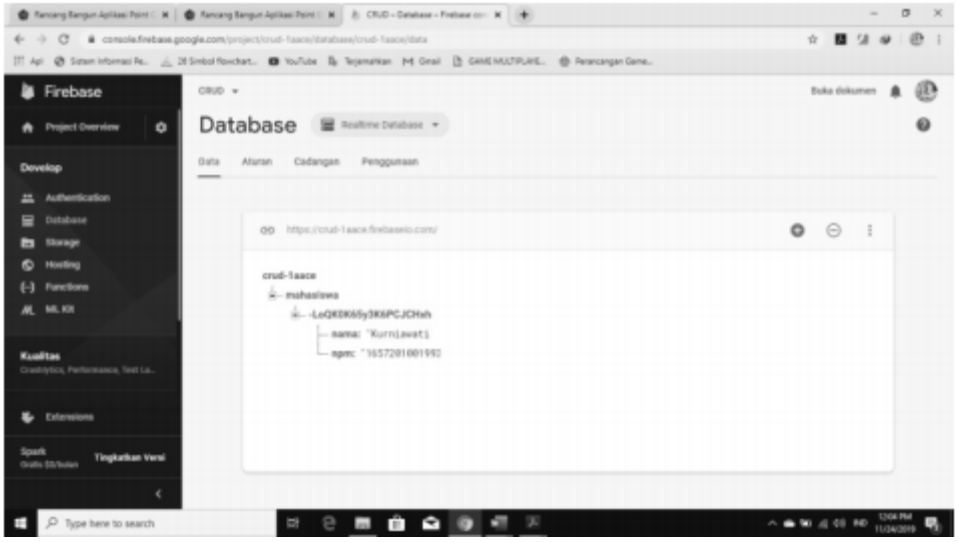
Sumber : (R. B. D. Putra, 2021)

3. II.2.10. *Firebase Realtime Database*

Pada penelitian ini menggunakan database yang disediakan oleh *firebase* yaitu *real time database* yang memiliki struktur *JSON Tree*. *Firebase* adalah teknologi yang memungkinkan untuk membuat sebuah aplikasi tanpa memikirkan *server-side programming*. Aplikasi yang menggunakan *Firebase* dapat melakukan control penggunaan data tanpa harus memikirkan bagaimana data itu disimpan, dan disinkronisasi ke seluruh pengguna aplikasi secara *real time* [1].

Firebase Realtime Database sendiri merupakan basis data berbasis *cloud* NoSQL yang menyinkronkan data di semua klien secara *realtime*, dan menyediakan fungsionalitas *offline*. Data yang diinputkan disimpan ke dalam *database Realtime* sebagai JSON. Semua klien yang terhubung dan

saling berbagi dalam satu waktu, secara otomatis akan menerima pembaruan dengan data terbaru



Gambar II.2. Contoh *Firebase Realtime Database*

Sumber : [1].

II.2.11 UML *(Unified Modeling language)*

Unified Modeling Language (*UML*), adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi *(sharing)* dan mengkomunikasikan rancangan mereka dengan yang lain.

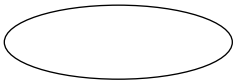
UML merupakan keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat

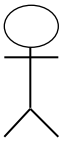
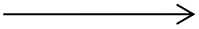
lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Pemodelan *visual* adalah salah satu cara berpikir tentang persoalan menggunakan model-model yang diorganisasikan seputar dunia nyata yang berguna untuk memahami persoalan, mengkomunikasikan dengan orang-orang yang terlibat dalam proyek (*costumer*, ahli dibidangnya, analisis, desainer dan lain-lain). Serta didefenisikan sebagai proses pemodelan sistem informasi menggunakan pengaturan standar elemen grafik dan objek-objek dalam sistem dan antar sistem itu sendiri. (Munawar; 2018: 49).

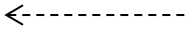
1. *Use Case* Diagram

Use Case diagram merupakan pemodelan untuk kelakuan (*behavior)* sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use Case* diagram yaitu:

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	*Use Case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor,

	biasanya dinyatakan dengan menggunakan kata kerja di awal nama *Use Case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *Use Case*, tetapi tidak memiliki *control* terhadap *Use Case*.
	Asosiasi antara aktor dan *Use Case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan *Use Case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

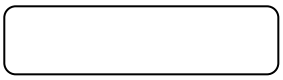
	*Include*, merupakan di dalam *Use Case* lain (*required*) atau pemanggilan *Use Case* oleh *Use Case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *Use Case* lain jika kondisi atau syarat terpenuhi.

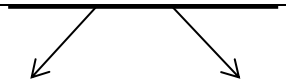
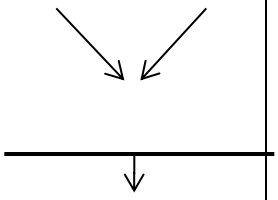
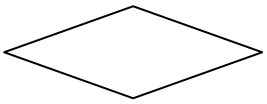
(Sumber : Munawar; 2018: 89)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan *Workflow* (Aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *Activity diagram*, yaitu :

Tabel II.2. Simbol Diagram Aktivitas

Gambar	**Keterangan**
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan

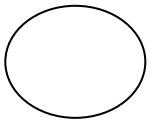
	secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points,* menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

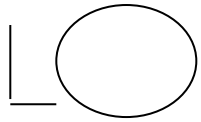
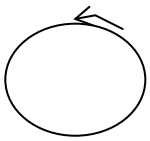
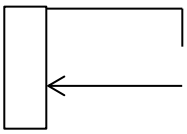
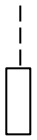
(Sumber : Munawar ; 2018 : 137)

3. Diagram Urutan *(Sequence Diagram)*

Diagram urutan menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam diagram urutan, yaitu

Tabel II.3. Simbol Diagram Urutan

Gambar	Keterangan
	*Entity Class,* merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.

	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan *formentry* dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Munawar ; 2018 : 186)