

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian sebelumnya begitu banyak peneliti yang menggunakan *firebase* dikarenakan mudah dan banyak peneliti yang menggunakannya. Walaupun begitu hanya sedikit yang melakukan penelitian untuk menguji sinkronisasi setiap datanya. Alasan yang banyak menggunakan *Firestore*, dikarenakan dapat mengekspor data secara langsung menggunakan *REST API* saja dalam bentuk *JSON* atau *CSV*[9].

Tabel 2.1 Penelitian Terdahulu

No	Judul	Perbandingan	Perbedaan	Kekurangan	Pengumpulan	Ringkasan
1.	<i>Application For Attendees of a Talk Show Event</i> [10].	Pengembangan aplikasi untuk peserta acara <i>talk show</i> dengan menggunakan <i>realtime database</i>	Menggunakan <i>javascript</i> dalam merancang aplikasi dan menggunakan <i>IOS</i> .	Pada tahap pengujian hanya menggunakan <i>firebase</i> . Dan tidak dihitung pada waktu yang digunakan	Adapun cara yang digunakan peneliti dalam proses mengumpulkan data yaitu menggunakan metode Observasi, Kuesioner, Wawancara.	Peneliti membuat sebuah aplikasi yang memberikan informasi mengenai layanan <i>firebase</i> yang digunakan.
2.	Implementasi Teknologi <i>Internet of Things</i> Pada Sistem Pemantauan Kebocoran Gas LPG dan Kebakaran Menggunakan <i>Database</i> Pada <i>Google Firestore</i> [11].	Membangun teknologi IoT dan hasil data akan diunggah ke fitur <i>realtime database</i> pada <i>firebase</i> . Lalu data akan ditampilkan pada <i>smartphone</i> melalui aplikasi	Tidak membangun aplikasi android yang akan teliti. Peneliti hanya menggunakan yang sudah ada	Pada uji coba <i>delay</i> terdapat perangkat tersambung ke sensor. Oleh karena itu <i>delay</i> terdapat banyak waktu digunakan.	Pengumpulan data dilakukan ujicoba perangkat IoT dan observasi. Untuk pengiriman data dilakukan dengan aplikasi <i>wireshark</i> dihitung dari <i>delay</i> nya sebanyak 30 kali.	Peneliti berhasil membangun teknologi IoT dan meneliti waktu tunda pengiriman data. Didapat nilai tertinggi waktu <i>delay</i> adalah 64,61 detik dan nilai terendah nya adalah 61,562 detik
3.	<i>Android based instant messaging application using firebase</i>	analisa aplikasi <i>instant messaging</i> dengan berbentuk <i>chatroom</i> untuk menyimpan data	Bertujuan untuk menganalisa data secara real time dan tidak secara	Aplikasi yang di Analisa tidak menggunakan yang telah dibuat melainkan	Surveinya didasarkan pada fitur-fitur aplikasi yang ada seperti <i>WhatsApp</i> , <i>Snap Chat</i> ,	Aplikasi komunikasi real-time yang memudahkan pengguna berkomunikasi

No	Judul	Perbandingan	Perbedaan	Kekurangan	Pengumpulan	Ringkasan
	[12]	pada satu tempat secara langsung	langsung menganalisa data yang ada pada firebase. Dan aplikasi hanya menggunakan yang sudah ada	yang sudah ada seperti <i>whatsapp</i> , <i>snap chat</i> dan lain lain	<i>Hike</i> , <i>Telegram</i> , <i>Facebook Messenger</i> . Lalu mempertanyakan kepada 50 orang dari kelompok umur berbeda	i satu sama lain.
4.	Merancang dan membangun aplikasi <i>chatting</i> berbasis android menggunakan <i>database firebase</i> di universitas islam al-ihya kuningan[13]	Membangun aplikasi <i>chat</i> menggunakan <i>database firebase</i> disertai dengan algoritma enkripsi AES (Progressed Encryption Standard)	Pengujian memasukkan metode enkripsi AES	Peneliti mengumpulkan data dengan sedikit data dan tidak terlalu efektif.	Pengumpulan data dengan pengujian proses enkripsi teks pada saat penggunaan aplikasi	Aplikasi <i>chatting</i> ini dirancang dan dibuat memungkinkan pengguna untuk mengirim pesan ke orang secara real time dan terhubung ke Internet menggunakan teknologi enkripsi simetris. ditujukan untuk keamanan transmisi data..
5.	Analisis Kinerja <i>Internet Of Things</i> Berbasis <i>Firestore Real-Time Database</i> [14].	Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja <i>Firestore</i> sebagai <i>database</i> yang dapat mengupdate informasi secara <i>real time</i> melalui internet.	Pengujian dilakukan dengan menggunakan tiga buah jaringan yaitu 3G, 4G, dan LAN(Local Area Network)	Pada pengujian delay peneliti hanya observasi dengan menyalakan/mematikan lampu. Dan tidak dijelaskan menggunakan aplikasi yang membantu.	Pengumpulan data menggunakan observasi dengan menyalakan/mematikan lampu secara jarak jauh melalui internet menggunakan <i>smartphone</i> dan akan diganti setiap jaringan yang berbeda	Dari penelitian ini rata-rata delay yang dihasilkan oleh jaringan 3G, 4G, dan LAN masing-masing adalah 3,57 detik, 1,42 detik, dan 1,4 detik.

Pada penelitian yang dilakukan oleh Carlos Hernández Prieto, pada tahun 2021 yang merancang aplikasi untuk peserta acara *talk show*. Pada tahap perancangan menggunakan *Firestore* untuk menyimpan *data user*, *scoring*, *session* dan sebagainya. Penelitian pun memberikan penjelasan singkat alasan peneliti menggunakan *firebase* [10]. Namun hanya berfokus pada pengembangan aplikasi

dan penjelasan *model firebase*, tidak menjelaskan kecepatan untuk sinkronasi data pada *firebase*.

Pada penelitian yang dilakukan oleh Abi Sabila Mustaqim, Danny Kurnianto, dan Fikra Titan Syifa pada tahun 2020 yang merancang IoT pada sistem pemantau kebakaran. Pada tahap perancangan IoT data yang dipantau akan diunggah ke fitur *realtime database* pada *firebase*, dan nantinya ditampilkan pada *smartphone* melalui aplikasi *android*. Lalu menggunakan *wireshark* untuk Pengujian Waktu Tunda (*Delay*) dengan pengujian pengiriman data dari sisi tx ke sisi rx[11]. Akan tetapi pada penelitian hanya menggunakan satu database yaitu *firebase* dan tidak dibandingkan dengan database lain.

Pada penelitian Sai Spandhana Reddy Emmadi, Sirisha Potluri pada tahun 2019 yang merancang dan menganalisa aplikasi *chat*. Menggunakan *database firebase* pada aplikasi *chat*, penelitian ini dapat menggunakan ruang obrolan sehingga pengguna dapat bertukar pandangan dan informasi tentang berbagai topik. Identitas pengguna juga dapat disembunyikan di ruang obrolan publik ini. Pada aplikasi hanya berguna pada *chat* belum ada fitur seperti suara gambar, streaming dan lain lain[12].

Pada Penelitian Jaenal Gopur Asmanul Salam, Annurahim Fakhri pada tahun 2022 yang membuat aplikasi *chat* dengan enkripsi AES (Progressed Encryption Standard). Pada tahap pengujian tidak dijelaskan bahwa jaringan berpengaruh atau tidak. Dan pada data enkripsi yang dianalisa hanya sedikit data yang dikumpulkan, dan tidak terlalu efektif pada saat Analisa datanya[13]

Pada Penelitian I Nyoman Buda Hartawan , I Wayan Sudiarsa pada tahun 2019 yang merancang IoT dan mengevaluasi kinerja *Firestore* sebagai *Real-Time Database*. Pada tahap perancang peneliti mengembangkan aplikasi berbasis *mobile android* dengan database online menggunakan layanan *firebase* dan melakukan pengujian perbedaan konektivitas jaringan terhadap *delay* pada jaringan *3G*, *4G*, dan *LAN*[14]. Akan tetapi pada penelitian ini tahap pengujian konektivitas

dilakukan pada jaringan tidak pada *database firebase*, walaupun begitu peneliti mendapatkan hasil *delay* yang memuaskan pada jaringan yang berbeda beda.

Dari penelitian-penelitian sebelumnya terdapat beberapa penelitian yang menggunakan *Firebase*, maupun perbedaan penggunaan kedua jaringan pada *database* tersebut tidak membandingkan kelebihan dan kekurangan penggunaan tersebut antara keduanya. Pada penelitian ini nantinya akan menggunakan database dengan jaringan berbeda. Dengan aplikasi *chat* ini nantinya dapat dibandingkan kedua jaringan pada *firebase* dengan menghitung berapa lama delay data yang akan diunggah maupun di unduh.

2.2. Dasar Teori

Kajian atau penjelasan yang membahas dasar beberapa teori yang digunakan peneliti pada penelitian sebagai berikut:

2.2.1. Data dan informasi

Sebelum membahas istilah *database* terlebih dahulu kita mengetahui istilah data dan informasi. Secara umum, data adalah sesuatu yang harus diproses meskipun tidak memiliki arti bagi penerimanya. Data dapat berupa peristiwa, situasi, gambar, suara, huruf, angka, matematika, bahasa, atau simbol lain yang dapat digunakan sebagai alat untuk memahami dunia di sekitar kita, objek, peristiwa, atau gagasannya[15].

Informasi adalah hasil dari analisis dan sintesis data; dengan kata lain, informasi adalah data yang telah disusun untuk memenuhi kebutuhan tertentu[16].

Jadi dapat disimpulkan bahwa data terdiri dari semua fakta, kata, dan angka yang dapat digunakan untuk mengumpulkan informasi yang berguna untuk penelitian ini.

2.2.2. Database

Database atau basis data adalah sekelompok data yang disimpan secara sistematis di komputer sehingga dapat diperiksa oleh *program* untuk mengambil data dari basis data. Konsep dasar dari basis data adalah kumpulan catatan, atau potongan pengetahuan. Deskripsi terorganisir dari jenis fakta yang disimpan didalamnya: penjelasan ini disebut skema[17].

Database di *manage* menggunakan Sistem manajemen basis data (*DBMS*), yang menyimpan konten basis data, memungkinkan pembuatan dan pemeliharaan data serta pencarian dan akses lainnya. Ada beberapa database yang masih digunakan, termasuk *MySql*, *Sql Server*, *Ms. Access*, *Oracle*, dan *PostgreSQL*[18].

Adapun beberapa fungsi dari database adalah :

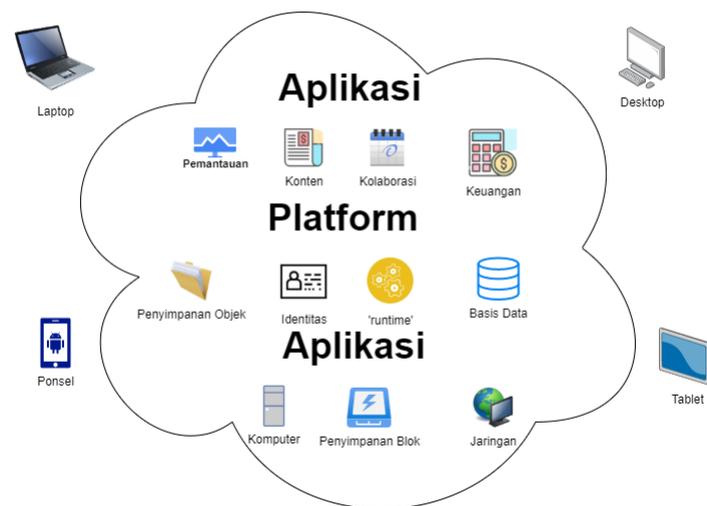
1. Menyimpan informasi secara digital
2. Pengelompokan data untuk memfasilitasi bantuan data, salah satu contohnya adalah pembuatan beberapa tabel atau kolom terpisah.
3. Meminimalisir adanya suatu data ganda.
4. Mempermudah penggunaan user, seperti penginputan data baru

2.2.3. Cloud

Cloud adalah kelompok objek *computer resource* yang dapat didefinisikan secara tepat, diakses dari lokasi mana pun, dan *resource*-nya dapat ditambahkan atau dihapus dengan cepat dan mudah. Ini menunjukkan bahwa *cloud* berisi teknologi yang dapat mengatasi masalah yang ditimbulkan oleh perangkat IoT[19].

Cloud Computing (Komputasi awan) adalah sebuah mekanisme yang memungkinkan pengguna untuk menggunakan sumber daya TIK yang terhubung dan hampir tidak terbatas sesuai permintaan di jaringan pribadi

dan publik. Infrastruktur dan aplikasi sepenuhnya dimiliki dan dikendalikan oleh pihak ketiga dalam model *cloud computing*. Dengan mekanisme ini pengguna layanan *cloud computing* tidak perlu menyiapkan komputer lokal untuk mengakses *file*, melainkan dapat diakses secara *real time* melalui internet[20].



Gambar 2.1 *Cloud Computing*[21]

Dari gambar bisa dilihat bahwa arsitektur *cloud computing* terdiri dari :

a. Infrastruktur

Infrastruktur ini terdiri dari semua perangkat keras dan perangkat lunak tambahan, termasuk semua perangkat infrastruktur jaringan, perangkat infrastruktur *server*, dan infrastruktur media penyimpanan.

b. Platform

Terdiri dari dari basis data yang digunakan, user yang digunakan, kecepatan, antrian data, dan objek penyimpanan.

c. Aplikasi

Terdiri dari aplikasi yang digunakan agar fungsi dari servicenya dapat digunakan.

Agar dapat berfungsi dengan baik dan sesuai kebutuhan, masing-masing dari ketiga komponen tersebut harus saling membantu[22].

Layanan pada *cloud computing* terdapat model layanan dengan perbandingan dari segi kemampuan yang disediakan salah satunya adalah sebagai berikut:

Software as a Service (SaaS)

Model ini memberikan aplikasi bisnis yang diakses melalui internet. Biasanya layanan ini pelanggan tidak perlu mengeluarkan biaya untuk memiliki perangkat lunak, akan tetapi akan dikenakan biaya tambahan jika menggunakan fitur-fitur yang lebih lengkap[23].

2.2.4. *Firebase*

Firebase adalah *API* yang disediakan google untuk menyimpan dan menampilkan data dalam aplikasi untuk *Android*, *iOS*, atau *web*. *Firebase* memiliki banyak fitur seperti *authentication*, *database*, *storage*, *hosting*, pemberitahuan dan lain-lain[24].



Gambar 2.2 *Firebase*[25]

Firebase Realtime Database adalah aspek menarik dari *Firebase*. *Realtime database* adalah salah satu *service* yang memungkinkan pengambilan data cepat dari dan penyimpanan dalam basis data[24], tetapi *Firebase* jauh lebih dari itu. Keunggulan yang ditawarkan *firebase* adalah menyimpan data secara lokal saat perangkat tidak terhubung ke internet[26].

2.2.5. *Delay*

Penundaan waktu suatu paket atau biasa yang disebut dengan delay adalah waktu yang diperlukan data untuk menempuh jarak dari asal ke tujuannya. Latensi dapat dipengaruhi oleh jarak, media fisik, kemacetan, atau waktu pemrosesan yang lama[27].

delay dapat dihitung menggunakan persamaan *delay transmisi* berikut ini :

$$Delay = waktu\ data\ terima - waktu\ data\ kirim \quad (1)$$

Dimana didapatkan delay dengan mengurangkan waktu yang diterima dengan yang dikirim. Sebagai contoh jika didapat waktu kirim yaitu “11:09:04.899” dengan waktu terima yaitu “11:09:05.157” maka didapatkan delay yaitu “00:00:00.258” atau 258 mili detik.

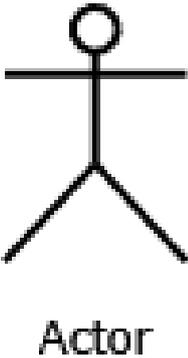
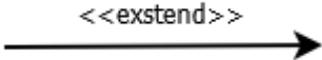
2.2.6. *UML*

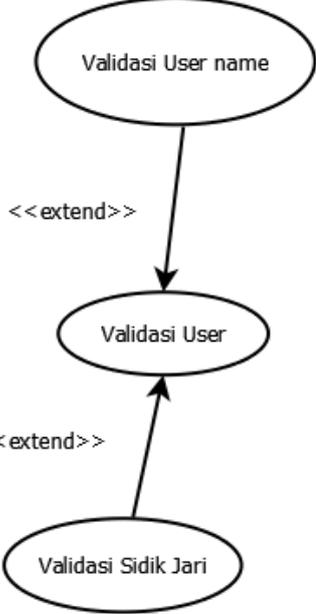
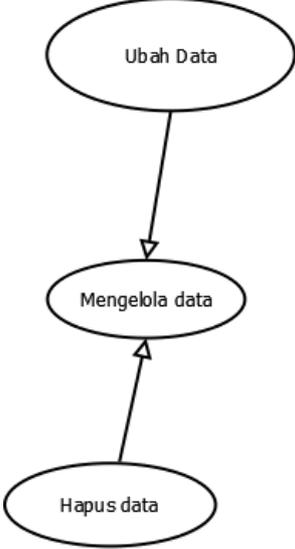
UML atau Uni fied Modelling Language adalah alat/model untuk merencanakan perbaikan program berorientasi objek. UML sendiri juga memberikan pedoman untuk menyusun kerangka kerangka, yang mencakup konsep persiapan bisnis, menyusun kelas-kelas dalam bahasa pemrograman tertentu, skema database, dan komponen-komponen yang diperlukan dalam sebuah *software*[28]. Berikut ini diagram-diagram yang ada dalam UML:

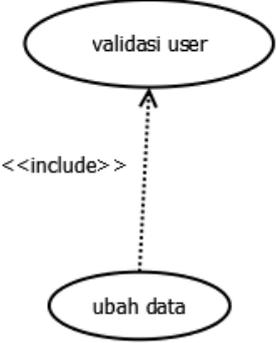
a. Use Case Diagram

Diagram Use Case adalah merupakan pemodelan perilaku (behavior) sistem informasi yang dibuat. Sebuah *use case* menggambarkan interaksi antara satu atau lebih aktor dan sistem informasi yang dibuat. Pada diagram ini biasanya menggambarkan yang dilakukan pada perancangan sistem informasi yang dibuat[29]. Diagram *use case* dijelaskan secara rinci, seperti:

Tabel 2.2 Simbol *Use Case Diagram*.

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>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>Aktor / <i>actor</i></p> 	<p>Meskipun lambang bagi aktor adalah gambaran seseorang, aktor belum tentu merupakan orang; ini biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. Sebaliknya, aktor merupakan gambaran suatu sistem yang berinteraksi dengan orang lain, proses, atau sistem lain yang berada di luar sistem informasi yang akan dibuat itu sendiri.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Interaksi antara aktor dan <i>use case</i> yang melibatkan aktor secara langsung atau melalui partisipasi dalam <i>use case</i>.</p>
<p><i>Ekstensi / extend</i></p> 	<p>Idenya sebanding dengan hubungan antara <i>use case</i> tambahan dan <i>use case</i> ketika <i>use case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>use case</i> lebih lanjut. Dalam pemrograman berorientasi objek, pewarisan terjadi ketika <i>use case</i> baru sering kali diperkenalkan dengan nama depan yang sama dengan <i>use case</i> aslinya, seperti:</p>

Simbol	Deskripsi
	 <pre> graph TD A([Validasi User name]) B([Validasi User]) C([Validasi Sidik Jari]) B -- "<<extend>>" --> A C -- "<<extend>>" --> B </pre> <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>
<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan antara dua <i>use case</i>, dimana satu fungsi lebih umum dari yang lain, dalam hal generalisasi dan spesialisasi (generik-spesifik);</p>  <pre> graph TD A([Ubah Data]) B([Mengelola data]) C([Hapus data]) B --> A C --> B </pre>

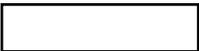
Simbol	Deskripsi
	 <p>Tergantung pada keadaan dan penafsiran yang diinginkan, satu atau semua penafsiran di atas dapat digunakan.</p>

a. *Sequence Diagram*

Sequence Diagram adalah menunjukkan bagaimana objek berperilaku dalam kasus penggunaan dengan merinci masa pakainya serta pesan yang dikirim dan diterima di antara objek tersebut. Oleh karena itu, untuk membuat *sequence diagram*, perlu mengetahui objek dan metode yang membentuk kelas objek dalam *use case* tertentu.[29]. *Diagram sequence* dijelaskan secara rinci, seperti:

Tabel 2.3 Simbol *Sequence Diagram*.

Simbol	Deskripsi
Aktor  Atau 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem.
<i>Lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek	Merupakan objek yang berinteraksi pesan.

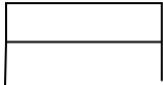
Simbol	Deskripsi
	
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan.
Pesan tipe <i>create</i> 	Menyatakan suatu objek memuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain.
Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya.
Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembali ke objek tertentu.
Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain.
<i>Entity Class</i> 	Menggambarkan hubungan yang akan dilakukan .
<i>Boundary Class</i> 	Menggambarkan gambaran dari foem.
<i>Control Class</i> 	Menggambarkan penghubung antara boundary dengan tabel.
<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message.

Simbol	Deskripsi
	

a. Activity Diagram

Activity Diagram digunakan untuk membuat aktivitas yang dibuat dalam satu operasi berguna untuk aktivitas selanjutnya, seperti kasus penggunaan atau interaksi. *Diagram activity* juga dapat menunjukkan proses yang berjalan secara bersamaan dan mungkin memerlukan banyak eksekusi.[29]. *Diagram activity* dijelaskan secara rinci, seperti:

Tabel 2.4 Simbol Activity Diagram.

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