BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Fathul Hafidh & M.Dedy. (2018) dalam jurnalnya "Aplikasi Penjadwalan Program Praktikum Fakultas Teknik Informasi Universitas Islam Kalimantan (UNISKA) Muhammad Arsyad Al Banjari Banjarmasin" membuat sebuah aplikasi sistem perangkat lunak untuk memudahkan pelaksanaan program praktikum yang dilaksanakan, dengan tujuan meningkatkan kemampuan mahasiswa dalam keilmuan pemrograman. Dengan menggunakan bahasa PHP pada pembuatan aplikasinya dan juga MySQL sebagai database. Dengan tujuan membuat aplikasi yang membantu dalam pendaftaran mahasiswa yang akan mengikuti kelas praktikum dan juga membantu dosen dalam memanajemen jadwal praktikum. Dengan hasil akhir yang didapatkan merupakan penjadwalan yang membantu mahasiswa dalam mendapatkan informasi dan kemudahan dalam memilih jadwal sesuai dengan kebutuhan mahasiswa. Aplikasi yang dibuat juga memiliki sifat *multiplatform* sehingga dapat dijalankan menggunakan sistem operasi linux, windows, android dan lainnya

Wijaya.S (2012) dalam jurnalnya "Penerapan Web Service Pada Aplikasi Sistem Akademik Pada Platform Sistem Operasi Mobile Android" Sistem informasi perkuliahan menjadi kebutuhan mahasiswa, akan tetapi semua sumber informasi pengirimannya hanya melalui papan informasi. Permasalahan seperti ini bisa diselesaikan dengan memanfaatkan web service yang bisa menangani berbagai aplikasi baik itu Android maupun iOS dan lainnya. Dengan pengiriman data yang digunakan dalam format JSON, dimana pada aplikasi ini client hanya perlu melakukan parsing data dan menyajikan data sesuai dengan yang diinginkan oleh pengguna. Dengan menggunakan teknik normalisasi pada aplikasi dapat mengurangi redundansi data pada database dalam skala besar. Sehingga akan mampu menghasilkan desain pada relasi database yang terintegrasi dan kinerja yang optimal. Dalam penggunaan aplikasi juga terkadang memiliki banyak kendala seperti pemanggilan data ketika aplikasi digunakan untuk mengakses data melalui

web service, sehingga terlihat beberapa kali aplikasi mengalami force close ketika pemakaian internet yang sedang tinggi.

Irsan.M (2015) dalam jurnalnya yang berjudul "Rancang Bangun Aplikasi Mobile Notifikasi Berbasis Android Untuk Mendukung Kinerja di Instansi Pemerintahan" menganalisis terhadap pengujian aplikasi yang dibuat sehingga didapatkan bahwa aplikasi akan membantu dan memudahkan *user* dalam melakukan berbagai proses yang dibutuhkan penggunanya. Dengan menggunakan aplikasi tersebut juga para *user* juga akan mendapati berbagai informasi-informasi mengenai pekerjaannya.

Dalam perkembangan teknologi terdapat tuntutan teciptanya inovasi-inovasi baru yang mendukung terciptanya sistem yang dapat menyelesaikan berbagai permasalahan mengenai prakikum. Penelitian ini akan menyelesaikan permasalahan; 1). Pendaftaran dan pendataan peserta praktikum yang masih manual, 2). Kurangnya pengetahuan mahasiswa terhadap jadwal pendaftaran hingga pelaksanaan praktikum. 3). Absensi peserta praktikum yang memerlukan pekerjaan cukup lama karena masih menggunakan cara manual dengan tanda tangan dikertas yang disediakan selama proses praktikum. Dengan berbagai permasalahan tersebut, membuat aplikasi pendaftaran praktikum teknik elektro UMY yang dapat mendaftar secara online dan tersusun rapi dalam satu aplikasi android. Aplikasi pendaftaran praktikum yang akan dibangun dapat membantu pendaftaran dan pendataan mahasiswa yang akan mengikuti kelas praktikum dan juga presensi praktikum yang berjalan dan membantu mahasiswa dapat mememilih jadwal sesuai dengan yang mereka inginkan.

Berdasarkan ketiga penelitian di atas maka penulis berinisiatif membuat UPracte sebagai aplikasi pendaftaran, penjadwalan dan pendataan praktikum. Tager dari aplikasi ini adalah mahasiswa di Laboratorium Teknik Elektro Universitas Muhammadiyah Yogyakarta. Mahasiswa dapat menggunakan aplikasi ini sebagai sumber informasi praktikum, mulai dari jadwal pendaftaran, pendaftaran, jadwal praktikum, hingga pendataan pada nilai praktikum dengan cara menginstall aplikasi dan terhubung dengan internet.

2.2 Landasan Teori

2.2.1 Pengertian Aplikasi

Aplikasi adalah suatu sub kelas perangkat lunak computer yang memanfaatkan kemampuan computer langsung untuk melakukan suatu tugas seperti yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan computer, tapi tidak secara langsung menerapkan kemampuan tersebut dalam mengerjakan suatu tugas yang menguntungkan pengguna. Adapun pengertian aplikasi antara lain:

1. Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas.
2. Aplikasi adalah sistem lengkap yang mengerjakan tugas spesifik.
3. Aplikasi basis data terdiri atas sekumpulan menu, formulir, laporan dan program yang memenuhi kebutuhan suatu fungsional unit bisnis / organisasi / instansi.

Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi para penggunanya. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *application suite*. Aplikasi dalam suatu paket biasanya memiliki *user interface* yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelejar dan menggunakan aplikasi. *Software Application* adalah *software* program yang memliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. *Sofware Application* terdiri dari bahasa pemrograman (*programming language)*, program aplikasi (*application program*), program paket (*package program*), program utilitas (*utility program*), *games, enternainment*, dan lain-lain. Untuk mendukung operasi *software application* maka tugas pengguna dibagi menjadi beberapa bagian yaitu sebagai Analis Sistem, *Progammer, Operator, Administrator Database, Administrator* Jaringan.

2.2.2 Mobile Application

Mobile Application adalah perangkat lunak yang dikembangkan untuk perangkat *mobile* atau perangkat nirkabel seperti *smartphone* dan *tablet*. *Mobile Application* juga memungkinkan penggunanya mengakses suatu layanan dengan mudah dimanapun dan kapanpun. Davis Pogue yang merupakan seorang pakar teknologi juga mengatakan bahwa *smartphone* yang lebih baru dapat dikatakan sebagai "telepon app" untuk membedakannya dari perangkat yang kurang canggih sebelumnya. Yang artinya, *mobile application* adalah perangkat lunak yang dijalankan di dalam telepon genggam atau *mobile phone*.

2.2.3 Unified Modeling Language

Unified Modeling Language atau yang biasa disingkat UML merupakan sebuah standar bentuk permodelan dalam suatu sistem, baik itu perangkat lunak maupun konsep pembuatan sistem lainnya. Dikutip dari buku *The Unified Modeling Language Reference Manual Second Edition* yang dikarang oleh Rumbaugh dkk, UML merupakan bahasa umum *modeling visual* yang digunakan untuk menentukan, membangun, dan mendokumentasikan prototipe dari sistem perangkat lunak yang akan dibuat. UML sendiri menguraikan tentang desain, konfigurasi, *maintain* dan informasi pengendalian sistem. UML juga dapat digunakan untuk segala metode pengembangan tahapan *lifecycle, application domains,* dan media.

Dalam standar UML akan merekam informasi tentang struktur *behavior* atau tangkah laku sistem secara statis dan dinamis. Adapun sistem tersebut dimodelkan sebagai kumpulan dari beberapa *class* yang berinteraksi satu sama lain agar program dapat berjalan. Sistem akan dipetakan menjadi dia klasifikasi besar berdasarkan *behavior,* yaitu *dynamic view* dan *static view.*

1. ***Use Case View***

   Use Case View memodelkan fungsi dari sebuah objek yang dirasakan langsung oleh *user* yang disebut sebagai *actor*. *Actor* akan berinteraksi dengan subjek dari sudut pandang tertentu. *Use case* menjelaskan tentang sebuah transaksi antara *user* dan subjek. Tujuan dari *use case view* adalah untuk menguraikan seberapa banyak *actor* yang ikut terlibat

dalam sebuah sistem. Adapun *use case view* digambarkan oleh diagram yang disebut *use case diagram.*

2. ***Activity View***

   Activity View merupakan model yang akan menampilkan diagram alir komputasional pada sebuah sistem. Pada model ini juga akan digambarkan *step by step* dari setiap proses di *background*. *Activity view* sendiri digambarkan oleh *activity diagram*.

2.2.4 Java Code

Java merupakan bahasa pemrograman yang dapat dijalankan diberbagai *device*, mulai dari computer hingga *handphone*. Java diciptakan oleh James Gostling dari perusahaan Sun Microsystem pada tahun 1991. Versi pertama dari Java yaitu Java 1.0 yang dirilis pada 1995. Dari waktu ke waktu pemrograman Java terus dikembangkan sehingga menambahkan banyak *libraries*. Sampai sekarang versi terbaru dari Java adalah very 8 yang baru dirilis pada tanggal 17 Oktober 2017. Java sendiri banyak mengadopsi *syntax* dari bahasa pemrograman C dan C++, lalu kemudian dibuat lebih sederhana.

Gambar 2.1 Logo Java

Pada saat ini pemrograman Java telah digunakan oleh banyak *platform.* Java sendiri menjadi pondasi bagi bahasa pemrograman lainnya. Sepeti Kotlin, Scala, Clojure, Groovt, Jruby, Jython dan lainnya. Semua pemrograman tersebut

memanfaatkan Java *Virtual Machine* sebagai rumahnya. Selain itu, Java digunakan untuk membuat aplikasi *native* pada android.

Penggunaan bahasa java sebagai bahasa pemrograman pada android memiliki alasan. Dikutip dari laman web bahasa java, terdapat beberapa alas an Google menjadikan bahasas java sebagai bahasa pemrograman android. Berikut adalah beberapa penjelasannya:

1. Java merupakan bahasa pemrograman yang popular. Setiap pengembang tidak perlu lagi mempelajari bahasa pemrograman yang baru, dikarenakan java sudah familiar dan merupakan bahasa paling terkenal di dunia.
2. Dikarenakan java terkenal, maka para pengembang dari pemrograman java sudah banyak. Dengan banyaknya pengembang java, maka *tools* yang terdedia hingga *community* pun banyak. Dengan banyaknya *libraries* yang tersedia maka para pengembang lebih dimudahkan karena tidak perlu lagi membuat suatu metode baru.
3. Java beroperasi dalam *virtual machine*, sehingga tidak perlu melakukan proses *recompile*. Proses pemisahan operasi dari aplikasi satu sama lain lebih mudah karena antara satu aplikasi dengan aplikasi lainnya tidak akan saling menggangu.

2.2.5 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middle ware* dan aplikasi. Android juga salah satu platform perangkat lunak yang merupakan sistem operasi untuk *mobile device*. Android memiliki basis kernel Linux yang dikembangkan oleh Google dan *Open Handset Alliance*. Android merupakan sistem operasi *open source*, dimana *developer* atau pengembang dapat membuat, mengedit dan memodifikasi sistem operasi tersebut berbasis bahasa java. Pada tahun 2005, Google membeli Android. Dikutip dari situs Venture Beat, Google membeli Android seharga USD $50 miliar. Keputusan itu terbukti tepat karena saat ini android sudah menjadi sistem operasi dengan jumlah *user* terbanyak mengalahkan IOS. Sundar Pichai selaku CEO google

juga menerangkan bahwa hingga bulan Mei 2017 sudah terdapat dua miliar pengguna aktif android. Selain itu, dikutip dari laman Info Kompuer, aplikasi Android pada play store sudah menembus angka 2,93 juta aplikasi.

Android selalu dikembangkan oleh pihak google setiap waktu. Sampai saat ini, android telah merilis versi terbarunya, yaitu android versi 8 dengan *codename* Oreo. Berikut adalah versi android dari waktu ke waktu.

2.2.6 Android Studio

Untuk mengembangkan sebuah aplikasi maka diperlukan sebuah *tools*. Aplikasi tidak bisa terbentuk hanya dengan satu *tool*, maka dari itu terdapat suatu istilah yang bisa menggabungkan semua *tools* pengembangan aplikasi menjadi satu, yaitu IDE (*Integrated Development Environment*). IDE setidaknya harus memiliki 4 fasilitas utama, yaitu:

1. *Editor*, yang berfungsi dalam menuliskan *source code* dari sebuah perangkat lunak.
2. *Compiler*, adalah fasilitas untuk melakukan pengecekan terhadap *syntax* dan *source code*, lalu kemudian menerjemahkannya menjadi bahasa *binary*.
3. *Linker*, yaitu berfungsi dalam menyatukan data-data *binary* menjadi suatu program yang siap dieksekusi
4. *Debugger*, berfungsi dalam melakukan pengetesan jalannya program. *Debugger* juga memiliki funsi sebagai pencari *bugs* atau banyaknya kesalahan yang terdapat pada program.

Gambar 2.2 Android Studio

Saat ini terdapat beberapa IDE untuk pengembangan aplikasi android. Salah satunya adalah android studio. Android studio merupakan IDE yang resmi dirilis tepatnya pada tanggal 16 Mei 2013 di Google I/O *conference.* Android Studio merupakan perangkat lunak gratis yang lisensinya dimiliki oleh Apace Lisence 2.0. Pertama, Google merilis android studio dalam versi *preview* versi 0.1. Setelah versi *preview*, Google mengeluarkan android Studio versi beta 0.8 pada bulan Juni 2014. Versi *stable* dari Android Studio baru dirilis pada bulan Desember 2014, dimulai pada versi 1.0. Adapun versi terbaru android studio sekarang adalah versi 3.0.0.

Selain Android Studio, terdapat beberapa IDE *tools* lainnya, yaitu Eclipse, Xamarin, App inventor, React Native, Iconic dan yang lainnya. Namun penulis menggunakan Android Studio karena beberapa kelebihan. Adapun kelebihan Android Studio dari IDE *tools* lainnya adalah sebagai berikut:

A. *Intelligent Code Editor*

Intelligent Code Editor merupakan suatu fitur khusus yang menjadi kelebihan paling utama dari Android Studio. Dengan fitur ini, android memiliki kemampuan *auto completion,* yaitu menampilkan *suggestion* untuk sebuah *syntax* yang mengalami *error.* Android Studio mampu menganalisis sebuah *code* dengan mumpuni serta dapat melakukan *refactoring,* sehingga waktu pembuatan program menjadi lebih cepat.

B. *Instant Run*

Fitur *instant run* merupakan fitur yang berfungsi dalam mempercepat *compile* dan *run* saat dilakukan untuk kedua kalinya. *Compile* dan *run* untuk pertama kalinya memakan waktu yang cukup lama. Namun, setelah pemrosesan kedua, saat program di *compile* dan *run* lagi, maka waktunya menjadi lebih cepat. Hal ini dikarenakan Android Studio tidak membuat ulang APK, melainkan hanya mengaplikasikan perubahan dari hasil file apk sebelumnya.

C. Sistem *Build* yang Handal dan Fleksibel

Fitur ini akan memudahkan pengembang dalam melakukan *compile*. Saat perintah *compile* dieksekusi, maka secara otomatis file apk akan terbentu. Sehingga pengembang tidak perlu lagi *build* aplikasi kembali.

Dalam android studio juga user diberikan kemudahan dalam menentukan *activity* yang akan digunakan pada aplikasi dengan catatan *activity* yang dimaksudkan implementasinya berupa *file class* (file java) yang akan dibuat. Berikut adalah pilihan – pilihan dari activity tersebut :

1. ***Add No Acitivy***

   Untuk pilihan ini, *user* akan membuat *project* aplikasi android dimana *project*-nya tidak memiliki *activity* pada saat dibuat. *Project* ini juga tidak memiliki *layout* yang langsung terbuat. Untuk pilihan ini, biasanya membuat *class* dan *layout* dilakukan secara manual setelah *project* terbuat. Setelah membuat *project* pada pilihan ini, *project* belum dapat dijalankan (*running).*

2. ***Basic Activity***

   Pilihan *project* ini akan otomatis membuat *activity* dan *layout* kosong dengan FloatingActionButton pada sudut kanan layar. *Activity* yang terbentuk sudah memiliki *coding* untuk *handle* apabila tombol FloatingActionButtonnya dipilih. Setelah membuat *project* pada pilihan ini, *project* akan dapat langsung dijalankan (*running).*

3. ***Empty Activity***

   Pilihan ini akan membuat *project* dimana sudah memliki *activity* dan tampilan, tetapi *activity* dan *layout* belum banyak memiliki *coding*. Isi *coding file* pada javanya hanya berupa setContentView untuk menampilkan *layout* tampilan. Setelah membuat *project* pada pilihan ini *project* akan dapat langsung dijalankan (*running*).

4. ***Fullscreen Activity***

   Pilihan ini akan membuat *project* dimana sudah memliki *activity* dan *layout* sudah berisi *coding* untuk membuat tampilan *project* menjadi

fullscreen ketika dijalankan. *Fullscreen* yang dimaksudkan adalah dengan menghilangkan *navigation view* paling atas dari tampilan *smartphone* di-*replace* dengan *fullscreen* aplikasi. Umumnya tampilan seperti ini digunakan untuk iklan atau *view* sesuatu yang mengharuskan *user* untuk melihatnya. *Project* pada pilihan ini dapat langsung dijalankan (*running*).

5. **Google Admob *Ads Activity***

   Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk menampilkan iklan dari google admob. Untuk menampilkan iklan dalam bentuk langsungnya, kita harus memasukkan id unit google admob pada string.xml. *Project* pada pilihan ini dapat langsung dijalankan (*running*).

6. **Google Maps *Activity***

   Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk menampilkan google maps. Untuk menjalankan *project* ini , maka *user* harus memiliki google maps api.

7. ***Login Activity***

   Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk *handle* login yang terdiri dari isian *username, password* dan *button login*.

8. ***Master/Detail Flow***

   Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk menampilkan *list data*, apabila *list data* dipilih akan menampilkan *detail view* dari data tersebut. *Project* pilihan ini bisa langsung dijalankan (*running*).

9. ***Navigation Drawer Activity***

   Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* yang apabila dijalankan seperti *sliding* menu sepeerti umumnya menu utama kebanyakan aplikasi pada masa sekarang.

10. *Scrolling Activity*

Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk tampilan *scrolling*. Tampilan *scrolling* yang dimaksudkan adalah tampilan layar yang memiliki *heading/title* yang dapat mengecil apabila *user scroll* layar ke bawah.

11. *Setting Activity*

Pilihan ini akan membuat *project* dimana *activity* dan *layout* sudah berisi *coding* untuk menampilkan berbagai bentuk model *setting* aplikasi yang banyak digunakan pada masa sekarang. *Activity* sudah memiliki *coding* yang sangat kompleks. *Project* ini bisa langsung dijalankan (*running*).

12. *Tabbed Activity*

Pilihan ini akan membuat *project* yang dalam satu *activity* menampilkan tiga tampilan layar dalam bentuk *tabbed* yang dapat digeser ke kanan/kiri. *Project* pada pilihan ini dapat langsung dijalankan (*running*).

Adapun pada android studio memiliki struktur hierarki project android untuk membuat aplikasi yaitu dengan memiliki folder sebagai berikut;

1. Manifest, folder ini berisi *file* AndroidManifest.xml yang berfungsi untuk *setting* dan konfigurasi keperluan sistem. Didalam *file* ini lah semua *activity* yang ada harus didaftarkan dan begitu juga keperluan konfigurasi seperti *permission* dan lainnya.
2. Java, folder ini berisi *file* - *file* java activity yang *user* gunakan.
3. Res**.** folder ini berisi *file* - *file layout* yang berupa xml, folder ini juga bisa berisi keperluan aplikasi *user* seperti gambar dan lainnya.
4. Gradle Scripts, folder ini berisi *file* - *file* konfigurasi sistem seperti target sdk, minimum platform sdk, *setting* kompilasi dan lain – lain yang berhubungan dengan sistem *gradle*.

2.2.7 User Interface (UI)

Secara umum *user interface* (UI) pada aplikasi android berasal dari tampilan *file* yang ber-ekstension xml. *File* xml inilah yang meng*handle* tampilan dari

aplikasi android yang dapat dilihat oleh pengguna aplikasi. Tampilan / *layout file* xml yang digunakan untuk tampilan aplikasi android secara umum dibagi menjadi tiga *layout* yaitu LinearLayout, RelativeLayout, TableLayout. Setiap *file* xml memiliki root tag salah satu *layout* yang menjadi *layout* induk yang akan digunakan.

A. LinearLayout

Tampilan dari LinearLayout akan membuat tampilan dengan konsep linear, dimana element – element tampilan akan menempati posisi linear dengan element lainnya baik itu secara vertikal maupun horizontal. Secara *default* LinearLayout menggunakan tampilan dengan horizontal.

B. RelativeLayout

RelativeLayout adalah posisi element didalamnya tergantung pada element lainnya. Tampilan LinearLayout memungkinkan kita membuat element yang satu *relative* dengan posisi element yang lain. Ada enam dasar *Relative* yaitu;

1. Di bawah (below)
2. Di atas (above)
3. Samping kiri (toLeftOf)
4. Samping kanan (toRightOf)
5. alignTop / alignBottom
6. alignParentTop / alignParentBottom

2.2.8 Adobe XD

Adobe XD merupakan kepanjangan dari *Xperiance Design*, adobe xd adalah alat berbasis vector yang dikembangkan dan diterbitkan oleh Adobe Inc. untuk merancang dan membuat prototipe pengguna untuk aplikasi web maupun aplikasi *smartphone*. Perangkat lunak ini tersedia diberbagai sistem operasi seperti Windows, macOS, iOS dan Android. Adobe XD juga mendukung desain vektor dan *wireframing* situs web. Aplikasi ini juga memadukan desain dengan mudah,

interactive prototype membuat desain *user interface* saling terhubung dari satu halaman ke halaman berikutnya dan kembali lagi, *live preview* untuk melihat hasil akhir dari rancangan desain, dan terakhir *esay sharing* komponen pada adobe XD dapat di-*export* ke berbagai *platform* .

Gambar 2.3 Logo Adobe XD

Adobe XD juga menyediakan *user interface* untuk aplikasi website hingga seluler. Dengan berbagai fitur yang digunakan untuk mendesain. Adapun beberapa kelebihan Adobe XD antara lain:

1. *Grid*

   Grid tools memungkinkan pengguna untuk menduplikasi item seperti galeri foto. *Grid* ini juga memungkina item yang berbeda untuk diduplikasi sebanyak apapun yang kita inginkan dengan pengembangan dan pembaruan yang mencakup penyimpanan otomatis.
2. *Prototype*

   Aplikasi ini juga memungkinkan pengguna untuk membuat prototipe animasi sebelum membuat sebuah coding, melalui tautan dengan *artboards*. Prototipe ini juga dapat di animasikan.
3. *Animation*

   Artboards pada aplikasi ini dapat dianimasikan dan dapat diintegrasikan dengan menggunakan After Effect untuk opsi lebih lanjut.

2.2.9 Database

2.2.9.1 Firebase Realtime Database

Firebase Realtime Database adalah database yang di-host di *cloud*. Data disimpan sebagai JSON dan disinkronkan secara *realtime* ke setiap *client* yang terhubung. Ketika membuat aplikasi lintas *platform* dengan SDK

Android, iOS dan JavaScript, semua *client* akan berbagi sebuah *instance* realtime database dan menerima update data terbaru secara otomatis. Adapun kemampuan utama dari firebase realtime database adalah sebagai berikut;

A. Realtime

   Sebagai ganti permintaan HTTP biasa, Firebase Realtime Database menggunakan sinkronisasi data setiap kali data berubah, semua perangkat yang terhubung akan menerima *update* dalam waktu milidetik. Sehingga memberikan pengalaman kolaboratif dan imersif tanpa perlu memikirkan kode jaringan.

B. Dapat diakses dari perangkat *client*

   Firebase Realtime Database dapat diakses secara langsung dari perangkat seluler atau brower web; sertver aplikasi tidak diperlukan. Keamanan dan validasi data dapat diakses melalui aturan kemanan firebase realtime database yang merupakan kumpulan aturan berbasis ekspresi dan dijalankan ketika data dibaca atau ditulis.

Cara kerja dari firebase realtime database adalah dengan membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi *client*. Data disimpan di *drive local*. Sehingga saat offline sekalipun, *realtime* akan terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang reponsif. Sehingga ketika koneksi dari perangkat pulih kembali, realtime database akan menyinkronkan perubahan data *local* dengan update jarak jauh yang terjadi selama *client* offline, sehingga setiap perbedaan akan otomatis digabungkan.

2.2.9.2 Firebase Cloud Firestore

Firebase cloud firestore adalah *database* yang fleksibel dan skalabel untuk pengembangan seluler, web dan server di Firebase dan Google Cloud Platform. Seperti firebase realtime database, cloud firestore membuat data

akan tetap terhubung di aplikasi *client* melalui *listener realtime* dan juga memiliki dukungan secara offline untuk seluler dan web. Dengan begitu, pengguna dapat membuat aplikasi yang responsive dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi internet. Cloud firestore juga memiliki integrase yang lancer dengan produk Firebase dan Google Cloud Platform lainnya, termasuk cloud functions. Adapun kemampuan utama dari cloud firestore adalah sebagai berikut;

A. Fleksibilitas

   Model data cloud firestore mendukung struktur data yang hirearkis dan fleksibel. Menyimpan data kedalam dokumen yang tersusun dalam koleksi, sub koleksi dan dokumen yang dapat berisi objek bertingkat yang kompleks.

B. Pembuatan Query yang Ekspresif

   Pada cloud firestore, pengguna dapat menggunakan kueri untuk mengambil masing – masing dokumen tertentu atau semua dokumen dalam koleksi yang sesuai dengan parameter kueri pengguna. Kueri daoat meliputi beberapa filter berantai dan menggambungkan penyaringan dan pengurutan. Kueri juga diindeks secara *default*, sehingga performa kueri sebanding dengan ukuran kumpulan hasil, bukan kumpulan data.

C. Update Realtime

   Seperti realtime database, cloud firestore menggunakan sinkronisasi data untuk mengupdate data pada perangkat yang terhubung. Namun, cloud firestore juga dirancang untuk membuat kueri pengambilan 1 kali yang sederhana secara efisien.

D. Dukungan Offline

   Cloud firestore menyimpan data yang digunakan secara aktif oleh aplikasi. Sehingga aplikasi dapat menulis, membaca, mendeteksi, dan melakukan kueri data meskipun perangkat sedang offline. Saat perangkat kembali online, cloud firestore akan menyinkronkan semua perubahan local kembali ke cloud firestore.