

BAB II

LANDASAN TEORI

2.1 Pengertian Rancang Bangun

Mohamad Subhan (2012). mendefinisikan “Perancangan adalah proses pengembangan spesifikasi baru berdasarkan rekomendasi hasil analisis sistem”, sedangkan menurut pendapat Satzinger, Jazkson dan Burd (2010), ”Perancangan sistem adalah kumpulan aktivitas yang menggambarkan secara rinci bagaimana sistem akan berjalan”. Sementara itu, Sugianto (2013) berpendapat bahwa “Perancangan Sistem adalah suatu kegiatan membuat desain teknis berdasarkan kegiatan pada waktu proses analisis”.

Berdasarkan pengertian di atas peneliti dapat menyimpulkan bahwa perancangan sistem merupakan spesifikasi baru yang menggambarkan secara rinci untuk tahap lanjutan dari proses menganalisa sistem.

2.2 Pengertian Aplikasi

Berdasarkan Nazruddin Safaat H (2012). Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Sedangkan pendapat Sutabri, Tata (2012), Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Sementara itu Menurut Asropudin (2013), Aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms.World, Ms.Excel.

Dari hasil pengertian diatas bahwa apikasi merupakan sebuah perangkat lunak yang tersutruktur dan terintegrasi antara bagian antar muka dengan penggunaanya, yang dapat digunakan dengan mudah.

2.3 Pengertian Informasi

Gellinas and Dull (2012) menyatakan bahwa informasi merupakan data yang disajikan dalam suatu bentuk yang berguna terhadap aktifitas pengambilan keputusan. Sedangkan Rommey dan Steinbart (2011), berpendapat bahwa informasi adalah data yang telah dikelola dan di proses untuk memberikan arti dan memperbaiki proses pengambilan keputusan. Sementara itu Deni Darmawan dan Kunkun Nur Fauzi (2013), Informasi adalah hasil dari pengolahan data yang memiliki makna atau arti.

Berdasarkan pengertian para ahli di atas dapat disimpulkan bahwa informasi merupakan betuk atau hasil olahan data yang didalamnya memiliki arti atau makna dari hasil pengambilan keputusan seseorang.

2.4 Definisi Pendakian Gunung

Pendakian gunung adalah suatu olahraga keras, penuh petualangan dan membutuhkan keterampilan, kecerdasan, kekuatan serta daya juang yang tinggi. Dalam arti luas, pendakian gunung berarti suatu perjalanan, mulai dari *Hill walking* sampai dengan ekspedisi pendakian ke puncak-puncak yang tinggi dan sulit hingga memerlukan waktu yang lama, bahkan sampai berbulan-bulan. Definisi lainnya adalah sebuah kegiatan alam bebas yang menggunakan wahana gunung sebagai sarana kegiatannya. (Harry Wijaya 2011).

Pendakian secara umum merupakan sebuah kegiatan seseorang untuk menuju ke sebuah ketinggian suatu gunung. Yang didalamnya memiliki beberapa unsur kegiatan lain seperti olahraga dan sekaligus berekreasi yang berhubungan dengan alam bebas, atau disebut pecinta alam. Pendakian tidak hanya memanfaatkan tenaga dari tubuh seseorang saja, tetapi menggunakan akal fikiran serta hati, yang dimana untuk mengambil keputusan pada saat melalui jalur atau medan pendakian yang sulit dan dibutuhkan kerja sama serta saling semangat dari sebuah tim atau teman yang mendaki secara bersama-sama (Andriyana,dkk, 2013)

Pendaki adalah seseorang atau sekelompok orang yang melakukan suatu perjalanan menuju ke alam terutama gunung dengan tujuan untuk menikmati alam yang masih sangat natural dan untuk menyukuri ciptaan tuhan yang sangat indah. tapi bukan cuma ikut ikutan dan hanya mengotori alam itu. justru sebaliknya, jika ia mencintainya justru ia akan menjaganya bukan mengotorinya. (Taufik Susilo 2012)

2.4.1 Kategori Pendakian

Hill Walking / Fell Walking

Secara khusus kegiatan ini disebut mendaki gunung. Hill Walking adalah kegiatan yang paling banyak dilakukan di Indonesia. Kebanyakan gunung di Indonesia memang hanya memungkinkan berkembangnya tahap ini. Disini aspek yang lebih menonjol adalah daya tarik dari alam yang dijelajahi (*nature interested*) Perjalanan mendaki bukit-bukit yang relatif landai dan yang tidak atau belum membutuhkan peralatan-peralatan khusus yang bersifat teknis.

(Harry Wijaya 2011).

Hill Walking yang merupakan sebuah pendakian gunung yang biasa disebut dengan hiking. Namun jenis pendakian tersebut hanya mendaki di daerah perbukitan atau menjelajahi setiap area bukit tersebut tidak sampai menuju puncak gunung tersebut. Banyak para hiking yang hanya melintasi daerah hutan rimba, sungai, rawa-rawa yang dianggap masih asing bagi para hiking tersebut. Dalam pendakian gunung yang dikenal dengan hiking ini tidak membutuhkan banyak peralatan penting. Untuk pegangan hanya membawa tongkat atau yang biasa dikenal dengan stock atau tongkat pandu. Selain itu juga ada istilah dengan scrambling yang merupakan salah satu kegiatan mendaki gunung dengan wilayah yang memiliki ketinggian dan kemiringan 45 derajat. Dalam hiking alat utama ialah kaki, namun kalau untuk scrambling tersebut membutuhkan kekuatan kaki dan tangan, kaki untuk mendaki sedangkan tangan digunakan untuk mengatur keseimbangan dan membantu gerakan pendakian. Karena scrambling yang rutenya lebih jauh dan medanya dengan kemiringan yang cukup ekstrim, maka dari itu diperlukan gerakan tangan yang lincah untuk mencari pegangan, selain itu tali juga perlu untuk membantu mencari pegangan tangan. (Harry Wijaya 2011).

- Scrambling

Pendakian pada tebing-tebing batu yang tidak begitu terjal atau relatif landai, kadang-kadang menggunakan tangan untuk keseimbangan. Bagi pemula

biasanya dipasang tali untuk pengaman jalur di lintasan.
(Aidi Laili 2009)

- Climbing

Kegiatan pendakian yang membutuhkan penguasaan teknik khusus. Peralatan teknis diperlukan sebagai pengaman. Climbing umumnya tidak memakan waktu lebih dari satu hari. Climbing merupakan salah satu jenis mountaineering memanjat yang sering dilakukan pada daerah pegunungan atau jurang. Banyak orang yang mengenal climbing tersebut dengan panjat tebing, namun dalam hal climbing tidak hanya memanjat tebing, namun ada beberapa kegiatan lainnya yang termasuk dalam climbing.

Seperti halnya Repling yang memiliki arti turun tebing, ada juga Pursiking yang merupakan pemanjatan tebing dengan bantuan alat yang disebut dengan tali pursik. Panjat tebing merupakan olahraga yang sangat berbahaya dan membutuhkan tenaga yang lebih. (Aidi Laili 2009).

Bentuk kegiatan climbing ini terbagi menjadi dua bagian, yaitu :

A. Rock Climbing

Pendakian pada tebing-tebing batu yang membutuhkan teknik pemanjatan dengan menggunakan peralatan khusus. Olahraga ini juga sama dengan panjat tebing, namun untuk olahraga ini membutuhkan konsentrasi dan pengetahuan lebih. Pasalnya olahraga ini lebih berbahaya yang bisa saja mengancam nyawa pemanjat. Untuk itu dalam olahraga ini membutuhkan

bantuan alat yang lebih krusial dan alat yang lebih banyak. Selain itu juga pemanjat juga harus memahami betul teknik-teknik dalam pemanjatan tebing. (Aidi Laili 2009).

B. Snow & Ice climbing

Ice and Snow Climbing merupakan olahraga panjat tebing untuk mountaineering yang sebenarnya hampir sama dengan Rock Climbing. Namun untuk Ice and Snow Climbing ini dilakukan pada medan yang bersalju. Yang dibutuhkan dalam melakukan Ice and Snow Climbing ini juga hampir sama dengan menggunakan kekuatan tangan, kekuatan kaki, dan juga beberapa alat untuk mencari pegangan dalam pemanjatan tebing yang bersalju. Selain kekuatan tangan dan kaki, yang paling penting juga ketahanan tubuh dalam dinginnya salju terhadap tubuh kita. (Aidi Laili 2009)

C. Mountaineering

Merupakan gabungan dari semua bentuk pendakian di atas. Waktunya bisa sehari-hari, berminggu-minggu, bahkan berbulan-bulan. Disamping harus menguasai teknik pendakian dan pengetahuan tentang peralatan pendakian, juga harus menguasai manajemen perjalanan, pengaturan makanan, komunikasi, strategi pendakian, dll. (Aidi Laili 2009)

2.4.2 Persiapan Mendaki Gunung

- Pengenalan Medan

Esa Unggul

Untuk menguasai medan dan memperhitungkan bahaya obyek seorang pendaki harus menguasai pengetahuan medan, yaitu membaca peta, menggunakan kompas serta altimeter. Mengetahui perubahan cuaca atau iklim. Cara lain untuk mengetahui medan yang akan dihadapi adalah dengan bertanya dengan orang-orang yang pernah mendaki gunung tersebut. Tetapi cara yang terbaik adalah mengikut sertakan orang yang pernah mendaki gunung tersebut bersama kita. (Taufik Susilo 2012).

- **Persiapan Fisik**

Persiapan fisik bagi pendaki gunung terutama mencakup tenaga aerobik dan kelenturan otot. Kesegaran jasmani akan mempengaruhi transport oksigen melalui peredaran darah ke otot-otot badan, dan ini penting karena semakin tinggi suatu daerah semakin rendah kadar oksigennya. (Taufik Susilo 2012).

- **Persiapan Tim**

Menentukan anggota tim dan membagi tugas serta mengelompokkannya dan merencanakan semua yang berkaitan dengan pendakian. (Taufik Susilo 2012).

- **Perbekalan dan Peralatan**

Persiapan perlengkapan merupakan awal pendakian gunung itu sendiri. Perlengkapan mendaki gunung umumnya mahal, tetapi ini wajar karena ini merupakan pelindung keselamatan pendaki itu sendiri.

Esa Unggul

Gunung merupakan lingkungan yang asing bagi organ tubuh kita yang terbiasa hidup di daerah yang lebih rendah. Karena itu diperlukan perlengkapan yang memadai agar pendaki mampu menyesuaikan di ketinggian yang baru itu. Seperti sepatu, ransel, pakaian, tenda, perlengkapan tidur, perlengkapan masak, makanan, obat-obatan dan lain-lain. (Taufik Susilo 2012).

2.4.3 Bahaya Mendaki Gunung

Dalam olahraga mendaki gunung ada dua faktor yang mempengaruhi berhasil tidaknya suatu pendakian (Taufik Susilo 2012).

- Faktor Internal

Yaitu faktor yang datang dari si pendaki sendiri.

Apabila faktor ini tidak dipersiapkan dengan baik akan mendatangkan bahaya subyek yaitu karena persiapan yang kurang baik, baik persiapan fisik, perlengkapan, pengetahuan, ketrampilan dan mental.

- Faktor Eksternal

Yaitu faktor yang datang dari luar si pendaki. Bahaya ini datang dari obyek pendakiannya (gunung), sehingga secara teknik disebut bahaya obyek. Bahaya ini dapat

berupa badai, hujan, udara dingin, longsor hutan lebat dan lain-lain.

Kecelakaan yang terjadi di gunung-gunung Indonesia umumnya disebabkan faktor intern. Rasa keingintahuan dan rasa suka yang berlebihan dan dorongan hati untuk pegang peranan, penyakit, ingin dihormati oleh semua orang serta keterbatasan-keterbatasan pada diri kita sendiri.

2.4.4 Langkah – langkah dan Prosedur Pendakian

(Lasmana, Ujang Dede 2013). Umumnya langkah-langkah yang biasa dilakukan oleh kelompok-kelompok pencinta alam dalam suatu kegiatan pendakian gunung meliputi tiga langkah, yaitu :

- **Persiapan**

Yang dimaksud persiapan pendakian gunung adalah : Menentukan pengurus panitia pendakian, yang akan bekerja mengurus : Perijinan pendakian, perhitungan anggaran biaya, penentuan jadwal pendakian, persiapan perlengkapan / transportasi dan segala macam urusan lainnya yang berkaitan dengan pendakian.

Persiapan fisik dan mental anggota pendaki, ini biasanya dilakukan dengan berolahraga secara rutin untuk mengoptimalkan kondisi fisik serta memaksimalkan ketahanan nafas. Persiapan mental dapat dilakukan dengan mencari/mempelajari kemungkinan-kemungkinan yang tak terduga timbul dalam pendakian beserta cara-cara pencegahan/pemecahannya.

- Pelaksanaan

Bila ingin mendaki gunung yang belum pernah didaki sebelumnya disarankan membawa guide/penunjuk jalan atau paling tidak seseorang yang telah pernah mendaki gunung tersebut, atau bisa juga dilakukan dengan pengetahuan membaca jalur pendakian. Untuk memudahkan koordinasi, semua peserta pendakian dibagi menjadi tiga kelompok, yaitu :

- Kelompok pelopor
- Kelompok inti
- Kelompok penyapu

Masing-masing kelompok, ditunjuk penanggungjawabnya oleh komandan lapangan (penanggungjawab koordinasi). Daftarkan kelompok anda pada buku pendakian yang tersedia di setiap base camp pendakian, biasanya menghubungi anggota SAR atau juru kunci gunung tersebut. Didalam perjalanan posisi kelompok diusahakan tetap yaitu : Pelopor di depan (disertai guide), kelompok inti di tengah, dan team penyapu di belakang.

Jangan sesekali merasa segan untuk menegur peserta yang melanggar peraturan ini. Demikian juga saat penurunan, posisi semula diusahakan tetap. (Lasmana, Ujang Dede 2013).

Setelah tiba di puncak dan di base camp jangan lupa mengecek jumlah peserta, siapa tahu ada yang tertinggal.

- Evaluasi

Biasakanlah melakukan evaluasi dari setiap kegiatan yang anda lakukan, karena dengan evaluasi kita akan tahu kekurangan dan kelemahan yang kita lakukan. Ini menuju perbaikan dan kebaikan.

2.4.5 Pengetahuan Dasar Bagi Pendaki Gunung

2.4.5.1 Orientasi Medan

- Menentukan arah perjalanan dan posisi pada peta
Dengan dua titik di medan yang dapat diidentifikasi pada gambar di peta. Dengan menggunakan perhitungan teknik/azimuth, tariklah garis pada kedua titik diidentifikasi tersebut di dalam peta. Garis perpotongan satu titik yaitu posisi kita pada peta. (Wanadri 2013)

Bila diketahui satu titik identifikasi. Ada beberapa cara yang dapat dicapai :

- Kalau kita berada di jalan setapak atau sungai yang tertera pada peta, maka perpotongan garis yang ditarik dari titik identifikasi dengan jalan setapak atau sungai adalah kedudukan kita. (Wanadri 2013)
- Menggunakan altimeter. Perpotongan antara garis yang ditarik dari titik identifikasi dengan kontur pada titik ketinggian sesuai dengan angka pada altimeter adalah kedudukan kita. (Wanadri 2013)
- Dilakukan secara kira-kira saja. Apabila kita sedang mendaki gunung, kemudian titik yang berhasil yang diperoleh adalah puncaknya, maka tarik garis dari titik

identifikasi itu, lalu perkirakanlah berapa bagian dari gunung itu yang telah kita daki. (Wanadri 2013)

- **Menggunakan Kompas**

Untuk membaca peta sangat dibutuhkan banyak bermacam kompas yang dapat dipakai dalam satu perjalanan atau pendakian, yaitu tipe silva, prisma dan lensa. (Wanadri 2013)

- **Peka dalam Perjalanan.**

Dengan mempelajari peta, kita dapat membayangkan kira-kira medan yang akan dilalui atau dijelajahi. Penggunaan peta dan kompas memang ideal, tetapi sering dalam praktek sangat sukar dalam menerapkannya di gunung-gunung di Indonesia. Hutan yang sangat lebat atau kabut yang sangat tebal acap kali menyulitkan orientasi. Penanggulangan dari kemungkinan ini seharusnya dimulai dari awal perjalanan, yaitu dengan mengetahui dan mengenali secara teliti tempat pertama yang menjadi awal perjalanan. (Wanadri 2013)

2.4.5.2 Membaca Keadaan Alam

- **Keadaan udara**

Sinar merah pada waktu Matahari akan terbenam. Sinar merah pada langit yang tidak berawan mengakibatkan

Esa Unggul

esok harinya cuaca baik. Sinar merah pada waktu Matahari terbit sering mengakibatkan hari tetap bercuaca buruk. (Wanadri 2013)

Perbedaan yang besar antara temperature siang hari dan malam hari. Apabila tidak angin gunung atau angin lembab atau pagi-pagi berhembus angin panas, maka diramalkan adanya udara yang buruk. Hal ini berlaku sebaliknya.

Awan putih berbentuk seperti bulu kambing. Apabila awan ini hilang atau hanya lewat saja berarti cuaca baik. Sebaliknya apabila awan ini berkelompok seperti selimut putih maka datanglah cuaca buruk.

- Membaca sandi-sandi yang diterapkan di alam, menggunakan bahan-bahan dari alam, seperti :
 - Sandi dari batu yang dijejer atau ditumpuk
 - Sandi dari batang / ranting yang dipatahkan / dibengkokkan
 - Sandi dari rumput / semak yang diikat
 - Tujuan dari penggunaan sandi-sandi ini apabila kita kehilangan arah dan perlu kembali ke tempat semula atau pulang.
- (Wanadri 2013)

2.4.6 Klasifikasi Informasi Pendakian Gunung

Jenis – jenis informasi pendakian gunung menurut (Taufik Susilo 2012) .

- Informasi bersifat umum, prosedural atau administrative
- Informasi pada kategori ini dapat berbeda – beda tiap orang, organisasi, komunitas atau negara, tergantung pada faktor geografis, budaya dan lainnya. Pelaksanaan informasi pada kategori ini dapat menunjang kenyamanan, kelancaran kegiatan pendakian, sedangkan pengabaian terhadap informasi ini mengandung tingkat resiko yang kecil/ringan terhadap keselamatan.
- Informasi bersifat anjuran karena memiliki tingkat resiko yang sedang, saran yang menunjang efektifitas pendakian, informasi potensi bahaya, atau alternatif penanganan masalah

2.5 Pengertian Gunung

Pengertian Gunung Berikut ini adalah pengertian tentang gunung menurut Kamus Besar Bahasa Indonesia versi online :

- Gunung yaitu bukit yang sangat besar dan tinggi; (Biasanya tingginya
- lebih dari 600 Mdpl (meter diatas permukaan laut).
- Pegunungan yaitu tempat yg bergunung-gunung; (Terdiri atas
- Gunung- gunung). Ada banyak definisi tentang gunung salah satunya adalah pengertian seperti berikut: “gunung adalah permukaan tanah yang menaik yang terbentuk akibat dari tenaga endogen atau kegiatan vulkanik dari dalam tanah atau bumi.

Sebuah gunung biasanya lebih tinggi dan curam dari sebuah bukit
.. (KBBI 2017)

2.6 Pengertian Android

Purwantoro, Rahmawati dan Tharmizi (2013). mengatakan Android merupakan suatu perangkat lunak (*software*) yang digunakan pada perangkat berjalan (*mobile device*) yang meliputi sistem operasi, middleware dan aplikasi inti. Android menurut Satyaputra dan Aritonang (2014) adalah sebuah sistem operasi untuk smartphone dan tablet. Sistem operasi dapat diilustrasikan sebagai jembatan antara piranti (*device*) dan penggunaannya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*.

Arif Akbarul Huda (2013). berpendapat mengenai Android merupakan sistem operasi berbasis Linux yang khusus untuk perangkat bergerak seperti smartphone atau tablet. Sistem operasi Android ini bersifat *open source* sehingga banyak sekali programmer yang berbondong-bondong membuat aplikasi maupun memodifikasi sistem ini. Para programmer memiliki peluang yang sangat besar untuk terlibat mengembangkan aplikasi Android karena alasan *open source* tersebut. Sebagian besar aplikasi yang terdapat dalam Play Store bersifat gratis dan ada juga yang berbayar.

Google (2013). Menyimpulkan bahwa Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

2.6.1 Karakteristik Android

(Android SDK 2016) Android memiliki empat karakteristik sebagai berikut :

- Terbuka

Android dibangun untuk benar-benar terbuka sehingga sebuah aplikasi dapat memanggil salah satu fungsi inti ponsel seperti membuat panggilan, mengirim pesan teks, menggunakan kamera dan lain-lain. Android merupakan sebuah mesin virtual yang dirancang khusus untuk mengoptimalkan sumber daya memori dan perangkat keras yang terdapat di dalam perangkat. Android merupakan open source, dapat secara bebas diperluas untuk memasukkan teknologi baru yang lebih maju pada saat teknologi tersebut muncul. Platform ini akan terus berkembang untuk membangun aplikasi mobile yang inovatif.

- Semua aplikasi dibuat sama

Android tidak memberikan perbedaan terhadap aplikasi utama dari telepon dan aplikasi pihak ketiga (third-party application). Semua aplikasi dapat dibangun untuk memiliki akses yang sama terhadap kemampuan sebuah telepon dalam menyediakan layanan dan aplikasi yang luas terhadap para pengguna.

- Memecahkan hambatan pada aplikasi

Android memecah hambatan untuk membangun aplikasi yang baru dan inovatif. Misalnya, pengembang dapat menggabungkan informasi yang diperoleh dari web dengan

data pada ponsel seseorang seperti kontak pengguna, kalender atau lokasi geografis.

- Pengembangan aplikasi yang cepat dan mudah
Android menyediakan akses yang sangat luas kepada pengguna untuk menggunakan aplikasi yang semakin baik. Android memiliki sekumpulan tools yang dapat digunakan sehingga membantu para pengembang dalam meningkatkan produktivitas pada saat membangun aplikasi yang dibuat.

2.6.2 Kelebihan dan Kekurangan Andorid

Meski saat ini sudah banyak platform mobile di pasaran seperti :

Symbian, IOS, Windows Mobile, Blackberry, Java Mobile Edition, Linux Mobile (LiMo), dsb , namun Android masih bisa menyaingi platform mobile tersebut. Beberapa fitur – fitur memang sudah ada pada platform sebelumnya, namun Android adalah yang pertama menggabungkan fitur – fitur sebagai berikut:

- Merupakan pengembang platform bebas yang berbasis linux dan open source, bebas karena pengembang dapat menggunakan dan menyesuaikan platform tanpa harus membayar royalti.
- Sebuah komponen berbasis arsitektur yang terinspirasi oleh mashup internet. Bagian dari satu aplikasi dapat dikembangkan, digantikan, ditingkatkan

Esa Unggul

komponennya sesuai dengan versi anda sendiri. Sehingga dapat memunculkan berbagai kreativitas pada mobile application.

- Memiliki layanan berbasis lokasi menggunakan GPS atau triangulasi menara sel untuk menyesuaikan dimana anda berada.
- Memungkinkan anda untuk memanfaatkan kekuatan penyimpanan lokal untuk berhubungan dengan komputasi dan sinkronisasi Browser dan peta views dapat ditanamkan 36 Android Software Development Kit (SDK) bekerja pada Windows, Linux, dan Mac OS X. Aplikasi yang anda buat dapat digunakan pada setiap perangkat Android, sebelum anda memulai membuat program anda perlu menginstall JAVA, IDE, dan SDK Android.

Selain kelebihan – kelebihan tersebut ada beberapa kekurangan Tablet Android, diantaranya :

- Handset Android dibuat oleh banyak vendor dengan ukuran layar yang berbeda, sebagian menggunakan keyboard fisik, sebagian lagi hanya menggunakan touch screen Versi Android yang terpasang di Handset juga berbeda berbeda, hal ini menyebabkan pengalaman memakai android tidak seragam, hal ini juga sekaligus menyulitkan Programmer dalam membuat program. Karena apabila versi OS tidak sama, program itu tidak bisa diinstal.
- Android adalah produk open source dari Google dan Open Handset Aliance, itulah mengapa aplikasi yang ada di Android sangat menyatu dengan layanan

Google, sebut saja Email menggunakan GMail, kalender menggunakan Google Calendar, bahkan sampai Contact pun disimpan di Gmail Contact. Bagi penggemar Google ini merupakan keunggulan sendiri, namun bagi pengguna yang tidak familiar dengan produk Google (Misal orang yang baru pindah dari Windows 37 Mobile atau Symbian) hal ini cukup merepotkan karena mereka harus menyesuaikan kebiasaannya.

- Android cukup aktif dikembangkan, hampir tiap tahun mereka merilis sistem operasi baru, pada tahun 2010 Google merilis froyo (2.2), Gingerbread (2.3) dan Honeycomb (3.0), tahun 2011 ini sudah merilis Ice cream sandwich (4.0). Namun dari sisi Vendor atau operator, update ke user membutuhkan waktu berbulan bulan, sebagai contoh, froyo dirilis pada bulan mei 2010, namun pengguna android kebanyakan baru mengupdate setelah bulan september.

2.7 Database

2.7.1 Pengertian Database

Menurut Connolly dan Begg (2010), database adalah kumpulan dari data yang banyak dan terhubung secara logika yang menggambarkan perusahaan, dan deskripsi dari data itu, yang didesign untuk memenuhi kebutuhan informasi perusahaan.

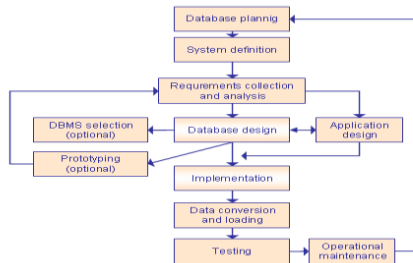
Dalam database, ada tiga hal yang menyusun sebuah database, yaitu entiti, atribut, dan relasi.

- Entiti atau entitas adalah sebuah objek yang keberadaannya dapat dibedakan terhadap objek lain.

- Atribut adalah ciri-ciri kualitatif yang dimiliki oleh suatu obyek, yang mencerminkan sifat-sifat dari obyek tersebut
- Relasi adalah hubungan antara tabel yang merepresentasikan hubungan antar obyek di dunia nyata.

2.7.2 Database Lifecycle

Menurut Connolly, dan Begg (2010), Database *Lifecycle* (Siklus hidup database) adalah pengembangan dasar, terkait dengan informasi sistem. Sistem database *lifecycle* tidak sepenuhnya berurutan, tetapi melibatkan beberapa jumlah pengulangan tahap sebelumnya melalui umpan balik.



Gamabar 2.1 Database Lifecycle

(<http://thesis.binus.ac.id/Asli/Pustaka/2011-1-00203->

IF%20Daftar%20Pustaka.pdf)

Penjelasan :

- *Database planning*
Merencanakan bagaimana tahapan dari siklus hidup dapat direalisasikan paling efisien dan efektif.
- *System definition*
Menentukan ruang lingkup dan batas-batas dari sistem database, termasuk pandangan pengguna utama, penggunaanya, dan area aplikasi.
- *Requirements collection and analysis*
Pengumpulan dan analisis persyaratan untuk sistem database baru
- *Database design*
Konseptual, logis, dan fisik desain database
- *DBMS selection(optional)*
Memilih DBMS yang sesuai untuk sistem database.
- *Application design*
Merancang user interface dan program aplikasi yang menggunakan dan memproses database.
- *Prototyping (opsional)*
Membangun model kerja dari sistem database, yang memungkinkan para desainer atau pengguna untuk memvisualisasikan dan mengevaluasi bagaimana sistem akhir akan terlihat dan berfungsi.
- *Implementation*
Menciptakan definisi database fisik dan program aplikasi.
- *Data conversion and loading*
Memuat data dari sistem lama ke sistem baru dan, jika mungkin, mengubah setiap aplikasi yang ada untuk dijalankan pada database baru.

- *Testing*
Sistem database yang diuji untuk kesalahan dan divalidasi terhadap persyaratan yang ditentukan oleh pengguna.
- *Operational maintenance*
Sistem database sepenuhnya dilaksanakan. Sistem ini terus dipantau dan dipelihara. Bila perlu, persyaratan baru yang dimasukkan ke dalam sistem database meskipun tahap terdahulu dalam lifecycle.

2.7.3 Pengertian NoSQL

Menurut Moniruzzaman dan Hossain (2013), NoSQL (*Not Only SQL*) merupakan group non relasi data manajemen sistem dimana database tidak dibangun dalam bentuk tabel dan pada umumnya tidak menggunakan SQL untuk manipulasi data. Database manajemen sistem NoSQL berguna ketika bekerja dengan jumlah data yang sangat besar dimana sifat data tidak membutuhkan model relasi. Sistem NoSQL terdistribusi dan dirancang untuk penyimpanan data berskala besar dan untuk pemrosesan data secara masif dan paralel di berbagai server. Mereka juga menggunakan bahasa non-SQL dan mekanisme untuk berinteraksi dengan data.

Sistem database NoSQL berkembang bersamaan dengan perusahaan besar di dunia internet seperti Google, Amazon, dan Facebook yang memiliki tantangan dalam menghadapi jumlah data yang besar ketika menggunakan solusi *Relational Database Management System (RDBMS)* konvensional. NoSQL dapat mendukung berbagai aktivitas seperti analisa data, transformasi data dengan ETL-style, dan non misi OLTP.

Pada awalnya, termotivasi pada aplikasi Web 2.0, sistem ini dirancang untuk menjangkau ribuan atau jutaan user yang mengupdate

sekaligus membaca, berbeda dengan DMBS tradisional dan data warehouses.

2.7.4 Sejarah NoSQL

Dari banyaknya model data yang ada, model relasi atau yang dikenal juga dengan RDBMS telah mendominasi semenjak tahun 80 an dengan implementasi seperti database Oracle, MySQL, dan Microsoft SQL Server.

Namun, belakangan ini terdapat peningkatan kasus dimana database relasional menimbulkan permasalahan karena defisit dan permasalahan dalam permodelan data dan juga batasan dari skalabilitas horizontal terhadap beberapa server dan besarnya jumlah data. Terdapat 2 trend yang membawa masalah ini ke permukaan komunitas software internasional :

- Pertumbuhan jumlah data yang dibuat oleh user, sistem dan sensor, mempercepat peningkatan jumlah data pada sistem terdistribusi seperti Amazon, Google, dan servis *cloud* lainnya.
- Meningkatnya kebergantungan dan kompleksitas dari data yang disebabkan oleh Internet, Web 2.0, jejaring sosial, dan akses terbuka ke sumber data dari berbagai macam sistem organisasi yang mengumpulkan sejumlah data yang tidak terstruktur berubah menjadi database non relasional, sekarang sering disebut dengan database NoSQL.
- Database NOSQL terfokus dalam menganalisa *dataset* berjumlah besar, meningkatkan skalabilitas dari penggunaan perangkat keras. Kebutuhan ruang penyimpanan dan komputasional dari aplikasi untuk *Big*

Data Analytics, Business Intelligence dan jejaring sosial yang berukuran *peta-byte* telah membuat database SQL kewalahan.

- Hal ini menyebabkan munculnya pengemabangan dari skalabilitas horizontal, penyimpanan data non relasi yang bernama NoSQL database seperti Google Bigtable, Facebook Cassandra, dan lainnya.

2.7.5 Klasifikasi dari Database NoSQL

Moniruzzaman dan Hossain (2013) mengklasifikasikan database NoSQL kedalam empat tipe yang sesuai dengan tugasnya masing masing :

- *Key-Value stores*

Pada umumnya, database ini menyimpan item dalam bentuk alpha-numeric identifier (keys) dan mengasosiasikan nilai dalam tabel yang simpel. Nilai yang ada dapat berbentuk string atau list dan sets yang lebih kompleks. Pencarian data hanya dapat dilakukan melalui keys, tidak melalui nilai dan hanya terbatas pada key yang sama persis.

Tabel 2.1 Contoh Penyimpanan Key-Value

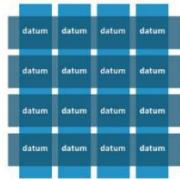
Esa Unggul

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year: 2005 Transmission: Auto

(NoSQL Database: New Era of Databases for Big data Analytics –
Classification Characteristics and Comparison-2013)

- Kesederhanaan dari *Key-Value Stores* membuat mereka ideal untuk pengambilan data yang dibutuhkan secara cepat untuk aplikasi seperti pengaturan profil pengguna atau session, atau mengambil nama produk. Contoh : Key-Value Stores- Dynamo (Amazon); Voldemort (LinkedIn); Redis; Berkeley DB; Riak.
- *Document Databases*
Terinspirasi dari Lotus Notes, database dokumen sesuai dengan namanya dirancang untuk mengatur dan menyimpan dokumen. Dokumen ini di-encode dengan format pertukaran data standar seperti XML, JSON (*Javascript Option Notation*) atau BSON (*Binary JSON*). Tidak seperti *Key-Value Stores*, kolom nilai di database dokumen mengandung data semi-terstruktur, khususnya nama atribut. Satu buah kolom dapat menyimpan ratusan atribut dan angka serta tipe atribut yang disimpan dapat bervariasi dari baris ke baris. Lalu, semua keys dan nilai dapat dicari di database dokumen.

Esa Unggul



Relational data model

Highly-structured table organization with rigidly-defined data formats and record structure.



Document data model

Collection of complex documents with arbitrary, nested data formats and varying "record" format.

Gambar 2.2 Database Dokumen

(<https://gigaom.com/wp-content/uploads/sites/1/2011/07/unql-1.jpg>)

- Database dokumen bekerja sangat baik dalam menyimpan dan mengatur dokumen berjumlah besar seperti dokumen teks, pesan email, dan dokumen XML. Contoh : CouchDB (JSON); MongoDB (BSON);
- *Wide-Column (atau Column-Family) Stores (BigTable-implementations)* Seperti database dokumen, penyimpanan WC/CF (*Wide-Column atau Column-Family*) menggunakan data struktur yang berorientasi kolom yang terdistribusi untuk mengakomodasikan beberapa atribut per key. Beberapa WC/CF memiliki kesamaan dengan *Key-Value*, kebanyakan mengikuti Bigtable Google.

Universitas

Esa Unggul

Wide Column Database	
Super Column Families : Customers	Super Column Families : Orders
RowID : 100001	RowID : 54311101
Super Column : Name	Super Column : Order
First Name : Sandip	OrderID : ORD10231001
Last Name : Shinde	Date : 01-01-2013
Super Column : Address	Super Column : Items
City : Pune	Item Code 1 : IS4002
Country : India	Item Code 2 : IS4101
PinCode : 411057	Super Column : Amounts
Super Column : Order Track	Discount : \$50.00
Last Order : ORD10231001	Amount : \$1500.00
Total Purchase : \$5400.00	RowID : 54311102
RowID : 100001	Super Column : Order
Super Column : Name	OrderID : ORD10231001
First Name : Manish	Date : 01-01-2013
Last Name : Kausik	Super Column : Items
Super Column : Address	Item Code 1 : IS4015
Address 1 : 31, M.G. Road	Super Column : Amounts
Address 2 : Near Bus Stop	Amount : \$700.00
City : Pune	
State : Maharashtra	
Country : India	
PinCode : 411001	
Super Column : Order Track	
Last Order : ORD50231201	
Total Purchase : \$15000.00	

Gambar 2.3 Contoh Wide Column Database

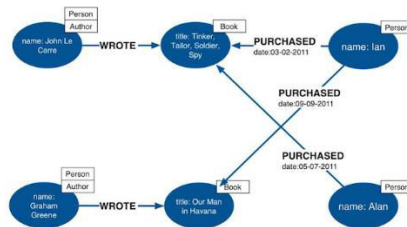
(<https://bi-bigdata.com/2013/01/13/what-is-wide-column-stores/>)

WC/CF berguna untuk

- Pendistribusian penyimpanan data khususnya data yang memiliki berbagai jenis versi karena WC/CF memiliki fungsi time-stamping.
- Pemrosesan data berorientasi batch dalam skala besar seperti *sorting*, *parsing*, *conversion*, *algorithmic crunching*, dan lainnya.
- Analisa yang dilakukan oleh programmer. Contoh : Bigtable (Google); Hypertable; Cassandra (Facebook); SimpleDB(Amazon); DynamoDB.
- *Graph Database*

Database grafik menggantikan tabel relasi dengan grafik relasi terstruktur dari pasangan key-value yang terhubung. Mereka mirip dengan database yang berorientasi objek. Grafik direpresentasikan dalam bentuk "*object oriented nodes*", "*node relationships*", dan "*properties*".

Labeled Property Graph Data Model



Gambar 2.4 Contoh Graph Database

(https://s3.amazonaws.com/dev.assets.neo4j.com/wp-content/uploads/property_graph_model.png)

Secara umum, database grafik berguna jika developer lebih tertarik pada hubungan antar data daripada data itu sendiri. Sebagai contoh, merepresentasikan jaringan sosial atau membuat rekomendasi.

Contoh : Neo\$; InfoGrid; Sones GraphDB; AllegroGraph; InfiniteGraph

2.7.6 Perbedaan dari Database Relasional dan Database NoSQL

Menurut Kumawat dan Pavate (2016) , perbedaan antara NoSQL dan database relasional sangatlah besar. NoSQL lebih cepat dan fleksibel. Selain itu karena sifatnya yang *open source*, developer dapat membuat *plugin* agar lebih mudah memakainya. Namun, database relasional lebih membantu dalam hal merancang database yang lebih kompleks dengan adanya relasi antar tabel dan struktur yang tetap. Database relasi dapat diandalkan dan meskipun kinerjanya lambat, database relasional merupakan pondasi bagi database terstruktur yang kompleks. Perbedaan utama antara kedua database tersebut adalah Database relasional memiliki relasi antar tabel.

Berikut merupakan perbedaannya dalam bentuk tabel :

Tabel 2.2 Perbedaan Database Relasional dan Database NoSQL

No	Relasional	NoSQL
1	Database dengan relasi antar tabel	Database tanpa adanya relasi antar tabel
2	Menggunakan SQL (Structured Query Language) untuk memanipulasi dan mendefinisikan data	Menggunakan Unstructured Query Language
3	Baik untuk menjalankan query yang kompleks	Tidak baik dalam menjalankan query yang kompleks
4	Cocok digunakan untuk aplikasi transaksional yang kompleks	Cocok digunakan untuk aplikasi dengan jumlah data yang sangat besar
5	Database berbasis tabel	Database berbasis dokumen, key-value, grafik, atau WC/CF
6	Memasukkan atau mengupdate data secara Synchronous	Memasukkan atau mengupdate data secara Asynchronous

(Correlation of NoSQL & SQL Database - 2016)

2.7.7 Keuntungan penggunaan NoSQL

Teknologi database tanpa relasional memiliki beberapa keuntungan yakni :

Esa Unggul

- Better scaling, scaling adalah sebuah istilah yang menunjukkan kemampuan sebuah teknologi dalam menyesuaikan dengan suatu kondisi. Misalnya , terjadi kondisi dimana user menjadi sangat banyak dalam waktu yang cepat(membeludak). Maka NoSQL memiliki kemampuan yang lebih baik dalam menyesuaikan kapasitasnya untuk menghadapi kondisi tersebut.
- Biaya maintenance murah, karena tidak membutuhkan database administrator dengan skill yang sangat tinggi.
- Schema-less , kita tidak perlu mendefinisikan fields dan tipe data pada tabel seperti pada mysql. (Degananda Ferdian 2013)

Dengan keuntungan diatas tidak menunjukkan bahwa NoSQL lebih hebat(powerful) dibandingkan dengan SQL. Karena hal itu tergantung dari berbagai aspek. Menurut saya aspek yang menjadi pertimbangan dalam memilih NoSQL dan SQL untuk dijadikan sebagai teknologi database adalah :

- Jenis aplikasi

Hal yang paling penting yakni apakah aplikasi tersebut membutuhkan sebuah query yang complex ? ataukah membutuhkan trigger untuk melakukan komputasi? jika hal tersebut dibutuhkan maka sql adalah pilihan yang terbaik. karena kedua case tersebut tidak dimiliki oleh nosql.

- Budget

Jika memiliki budget yang sedikit(startup) gunakan NoSQL. karena NoSQL jauh lebih murah dibandingkan

dengan Sql dan yang paling penting kita tidak perlu hire database administrator.

- **Konsistensi Data**

Apakah project kita membutuhkan konsistensi data yang sangat tinggi. Contohnya data stock membutuhkan konsistensi 100% karena berkaitan dengan jumlah barang(stock). Kesalahan stock akan berpengaruh pada banyak hal. Jika kondisinya seperti ini maka SQL adalah pilihan terbaik karena memiliki fitur seperti : trigger, stored procedure, view, relasi yang dapat membantu menjamin konsistensi data.

Sehingga, pada akhirnya, tergantung dari kebijakan dan kebutuhan dari project untuk menentukan penggunaan SQL atau NoSQL. Bahkan ada beberapa perusahaan besar yang menggabungkan keduanya. Contohnya adalah uber yang menyimpan json dalam schema di mysql.

2.7.7.1 Scaling NoSQL

Data yang terdapat pada RDBMS (Relational database management system) akan sangat menjadi masalah ketika data mencapai jumlah tertentu. Misalnya performa mysql akan menurun jika jumlah row mencapai satu juta. Untuk menjaga performa saat data mencapai jumlah tertentu RDBMS memiliki banyak sekali fitur untuk itu. Namun, hal ini tentunya membutuhkan database administrator dengan kemampuan yang sangat tinggi. Hal ini akan menambah beban (cost) untuk bisnis.

NoSQL melakukan scale-up(horizontal scaling) contohnya dengan menambahkan server baru(bukan mengupgrade kemampuan hardware pada satu server). Vertical scaling adalah

menambah kemampuan suatu hardware misalnya, meningkatkan RAM , clock dari processor yang kemudian disatukan dalam sebuah node.

(Degananda Ferdian, 2013).

2.7.7.2 Schema-less

Pada No-SQL kita tidak perlu mendefinisikan tipe data ataupun struktur lainnya. Meski begitu, istilah schema-less ini tidak 100% benar karena kita harus tetap mendefinisikan suatu “Struktur” pada level aplikasi. Contohnya jika kita menggunakan mongoose, kita tetap perlu mendefinisikan model dari data. (Degananda Ferdian, 2013).

2.7.7.3 Auto Sharding

Data dalam NoSQL database dapat diletakan pada suatu server yang berbeda-beda. Biaya menambah server jauh lebih murah daripada mengupgrade kapasitas suatu server. Data secara otomatis akan diseimbangkan dari berbagai server kepada server pool, jika terdapat satu server yang down maka data dapat secara langsung dimigrasikan pada server yang lain. Sehingga kemungkinan untuk down kecil dibandingkan RDMBS. (Degananda Ferdian, 2017).

2.7.8 SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. SQLite ini adalah mesin database SQL yang tertanam pada sistem yang kita gunakan. Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi overhead, latency times, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

(Sumber : <https://www.sqlite.org/about.html>)

Kelebihan SQLite :

- Karena database ini bersifat portabel, maka aplikasi ini bisa langsung disatukan dengan aplikasi yang sering disebut dengan embed
- Sangat cocok digunakan pada aplikasi yang tidak terhubung dengan Internet, baik aplikasi dekstop atau aplikasi mobile

- Tidak memerlukan pihak ketiga untuk mengakses
- Jika ingin copy database, cukup copy filenya saja

(Sumber : <https://www.sqlite.org/about.html>)

Kekurangan SQLite :

- Karena aplikasi ini langsung terhubung pada aplikasi tanpa menggunakan perantara, sehingga SQLite ini biasa digunakan untuk menyimpan data yang sedikit atau data temporary, misalnya game yang hanya menyimpan informasi skor pemenang.
- SQLite tidak menggunakan User Management, yang artinya jika mendapatkan file database, bisa saja langsung dibuka tanpa menggunakan username atau password
- Tidak semua perintah query bisa dilakukan pada SQLite ini
- Dimisalkan programmer ingin membuat aplikasi sebuah game yang sama tetapi pada beda platform, misalnya android, iphone dan blackberry. Maka programmer harus membuat database pada masing-masing platform tersebut (Sumber : <https://www.sqlite.org/about.html>)

2.7.9 JSON

Menurut Wijaya, Fenty ,dan Fiade JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di website. JSON dirancang untuk memudahkan pertukaran data pada situs dan

merupakan perluasan dari fungsi fungsi javascript. Diambil dari situs JSON.org, JSON terbangun atas 2 struktur :

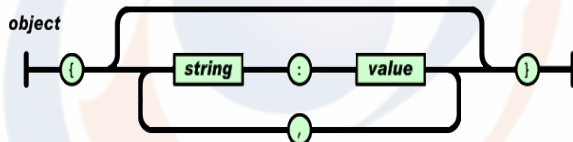
A. Object

Object merupakan kumpulan pasangan nama/nilai yang tidak berurut. Sebuah Object dimulai dengan “{“ dan berakhir dengan “}”. Setiap nama diikuti oleh “:” dan pasangan nama/nilai dipisahkan oleh “,”.

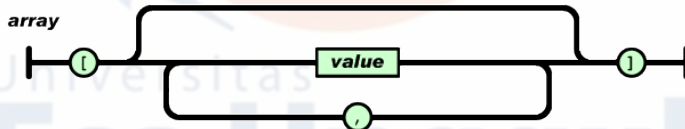
B. Array

Array merupakan kumpulan nilai yang berurutan. Sebuah array dimulai dengan “[“ dan diakhiri dengan “]”. Nilai dipisahkan dengan “,”.

Berikut merupakan gambaran dari JSON Object dan JSON Array



Gambar 2.5 JSON Object
(<http://www.json.org/>)



Gambar 2.6 JSON Array
(<http://www.json.org/>)

Menurut Peng, Cao, dan Xu (2011), penggunaan *JSON Parsing* dalam *HTTP Connection* ini bertujuan untuk memberi kemudahan bagi user ketika menggunakan aplikasi mobile. Dengan adanya *JSON Parsing* dalam *HTTP Connection*, informasi yang ada didalam website dapat ditampilkan di dalam sebuah aplikasi mobile. Hal ini menyebabkan aplikasi mobile tidak perlu menampilkan seluruh konten yang ada didalam website seperti halnya

pada mobile web browser. Aplikasi android akan memilih jenis konten yang ingin ditampilkan, seperti halnya gambar, deskripsi item, dan lain-lain. Dalam *JSON Parsing*, kita membutuhkan API (*Application Program Interface*) yang berfungsi untuk menghubungkan antara aplikasi mobile dan aplikasi website.

2.8 Pengertian Firebase Realtime Database

Google firebase (2011). Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.



Gambar 2.7 Logo Firebase
(Sumber: firebase.google.com)

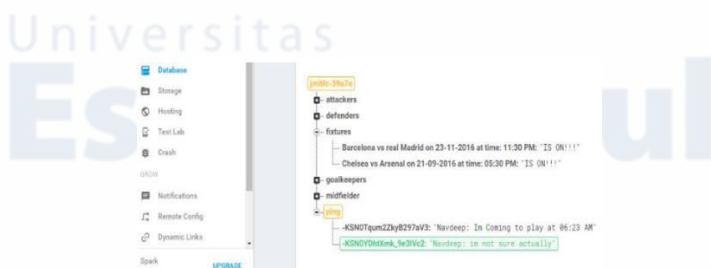
2.8.1 Cara Kerja Firebase Realtime Database

Firebase Realtime Database memungkinkan Anda untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat offline sekalipun, peristiwa realtime terus berlangsung,

sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis digabungkan.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firebase Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase Authentication, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

Realtime Database adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API Realtime Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.



Gambar 2.8 Contoh Database Firebase
(Study of Google Firebase API for Android - 2016)

2.8.2 Fitur Firebase

Menurut Google Firebase (2016), setelah API Firebase dimasukkan ke Dalam aplikasi Android atau iOS, pengembang bisa menggunakan fitur Firebase dengan coding yang simpel.

Beberapa fitur yang disediakan Firebase :

1. *Analytics*

Fitur ini membuat para pengembang aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. *SDK analytic* menangkap sendiri data yang dibutuhkan oleh para developer. *Dashboard* juga menyediakan detil seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. *Analytic* juga menyediakan data yang telah dirangkul.

2. *Authentication*

Fitur ini membuat para pengembang dapat mengizinkan pengguna untuk mengakses aplikasi. Firebase menyediakan fitur login melalui Gmail, Github, Twitter, Facebook dan juga autentikasi buatan sendiri.

3. *Messaging*

FCM (*Firebase Cloud Messaging*) membuat para pengguna dapat mengirimkan pesan ke berbagai *platform* tanpa biaya tambahan. *Messaging* juga dapat digunakan untuk kebutuhan notifikasi.

4. *Real-time Database*

Database di Firebase adalah database berbasis *cloud* dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa disebut juga NoSQL. Database ini sangatlah cepat dan dapat

Esa Unggul

diandalkan yang artinya data dapat dibaharui dan disinkronisasikan dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

5. *Storage*

Firebase juga menyediakan fasilitas penyimpanan. Firebase dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Meng-*upload* dan men-*download* juga dilakukan di *background*. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

6. *Hosting*

Firebase juga menyediakan fitur *hosting*. Firebase mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

7. *Crash reporting*

Fitur *crash reporting* di Firebase membuat pengembang dapat mengetahui kesalahan ketika terjadi *crash*.

Berikut merupakan contoh dari struktur data yang ada di Firebase

```
// An index to track Ada's memberships
{
  "users": {
    "alovelace": {
      "name": "Ada Lovelace",
      // Index Ada's groups in her profile
      "groups": {
        // the value here doesn't matter, just that the key exists
        "techpioneers": true,
        "womentechmakers": true
      }
    },
    ...
  },
  "groups": {
    "techpioneers": {
      "name": "Historical Tech Pioneers",
      "members": {
        "alovelace": true,
        "ghopper": true,
        "eclarke": true
      }
    },
    ...
  }
}
```

Gambar 2.9 Contoh Struktur Data Database Firebase
(<https://firebase.google.com/docs/database/web/structure-data>)

2.8.3 Menambahkan API Firebase ke Proyek

Para pengembang bisa menambahkan API Firebase ke semua proyek untuk Android 2.3 (*Gingerbread*) atau versi terbarunya dan juga harus memiliki servis Google Play 9.6.1 atau yang lebih baru. Tahapan yang harus dilakukan :

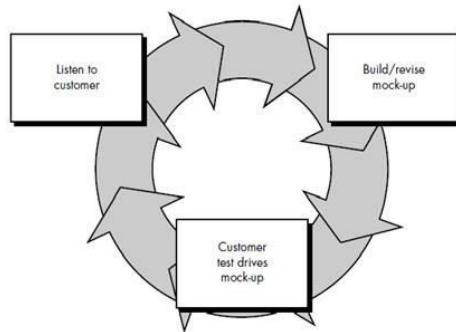
- Membuat proyek di konsol Firebase. Masukkan nama proyek dan lokasi pengembang.
- Klik tulisan "*Add Firebase to your Android app*".
- Masukkan package name dan juga sertifikat *SHA-1* yang bisa didapatkan melalui Android Studio.
- Download file *google-services.json* yang disediakan.
- Masukkan file json tersebut ke folder *app/* dari proyek

2.9 Metode Prototype

Prototype Model adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan Metode Prototyping ini pengembangan dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dibutuhkan, Pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang Kurang memperhatikan efisiensi Algoritma. Kemampuan sistem oprasi dan interface yang menghubungkan manusia dengan computer. (Roger, S. Pressman, Ph.D. 2012)

Pada Prototyping model kadang – kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detile input, proses atau detail output dilain waktu mungkin tim pembangun (developer) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika situasi seperti ini terjadi model prototyping sangat membantu proses pembangunan software. Proses pada prototyping bisa dijelaskan sebagai berikut.

- a) Pengumpulan Kebutuhan : developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.
- b) Perancangan : Perancangan dilakukan cepat dan rancangan me wakili aspek software yang diketahui. Dan rancangan ini menjadi dasar pembuatan prototype.
- c) Evaluasi Prototype : klien mengevaluasi prototype yang dibuat dan dipergunakan untuk memperjelas kebutuhan software



.Gambar 2.10 Model Prototyping menurut Pressman

2.9.1 Tahanan - tahapan Prototype

Tahap-tahap pengembangan Prototype model menurut Roger S.Pressman,Ph.D. adalah :

1. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari system dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu system yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana system yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. Merancang dan Membuat Prototype

Pada tahap ini, dilakukan perancangan dan pembuatan prototype system. Prototype yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

3. Uji coba

Pada tahap ini, Prototype dari system di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki Prototype yang ada.

2.9.1.1 Kelebihan metode Prototype :

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembangan dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Lebih menghemat waktu dalam pengembangan system
4. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

2.9.1.2 Kekurangan metode Prototype :

1. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
2. Interaksi pemakai penting. Sistem harus menyediakan dialog on-line antara pelanggan dan komputer.
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik.

2.10 Bahasa Pemograman Yang Digunakan

2.10.1 Java Script

Menurut Aloysius Sigit W, 2011 JavaScript merupakan bahasa Scripting yang bekerja disisi Client/Browser sehingga website bisa lebih interaktif. JavaScript adalah bahasa skrip yang populer di internet dan dapat bekerja di sebagian besar penjelajah web populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode JavaScript dapat disisipkan dalam halaman web menggunakan tag script.

2.10.2 XML

XML, atau **eXtensible Markup Language**, adalah sebuah cara untuk menyimpan data yang dapat dibaca baik oleh manusia maupun mesin. Format XML tersedia secara luas bagi banyak bahasa pemrograman. XML sangat mirip dengan JSON tapi membutuhkan lebih banyak teks sehingga lebih panjang isinya dan lebih lama untuk membaca dan menulisnya. XML harus dibaca dengan sebuah XML parser, namun JSON dapat dibaca menggunakan fungsi standar. Juga seperti JSON, XML tidak bisa menggunakan array. Kita akan melihat contoh data yang disimpan ke dalam format XML dan JSON. Affahruraji. (2017).

2.10.3 JSON

Menurut Wijaya, Fenty ,dan Fiade JSON (*Java Script Object Notation*) adalah format pertukaran data yang bersifat ringan, disusun oleh Douglas Crockford. Fokus JSON adalah pada representasi data di website. JSON dirancang untuk memudahkan pertukaran data pada situs dan merupakan perluasan dari fungsi fungsi javascript.

2.11 Basis Data Yang Digunakan (Database)

2.11.1 NoSQL

Menurut Moniruzzaman dan Hossain (2013), NoSQL (*Not Only SQL*) merupakan group non relasi data manajemen sistem dimana database tidak dibangun dalam bentuk tabel dan pada umumnya tidak menggunakan SQL untuk manipulasi data. Database manajemen sistem NoSQL berguna ketika bekerja dengan jumlah data yang sangat besar dimana sifat data tidak membutuhkan model relasi. Sistem NoSQL terdistribusi dan dirancang untuk penyimpanan data berskala besar dan untuk pemrosesan data secara masif dan paralel di berbagai server. Mereka juga menggunakan bahasa non-SQL dan mekanisme untuk berinteraksi dengan data.

2.11.2 Firebase Realtime Database

Google firebase (2011). Firebase Realtime Database adalah database yang di-host di cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung. Ketika Anda membuat aplikasi lintas-platform dengan SDK Android, iOS, dan JavaScript, semua klien akan berbagi sebuah instance Realtime Database dan menerima update data terbaru secara otomatis.

2.12 Aplikasi Pendukung

2.12.1 Android Studio

Dikutip dari situs resminya, Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android. Untuk dapat digunakan dalam pembuatan aplikasi android maka android studio membutuhkan plug-in yang disebut ADT, ADT adalah kepanjangan dari android development tools yang menjadi penghubung antara IDE android studio dengan Android Software Development Kit.

(<https://developer.android.com/studio/intro/index.html?hl=id>)

2.9.4 Java Development Kit (JDK)

Pemrograman java sebagai bahasa interpreter yang bersifat Multi Platform, tentu saja membutuhkan sebuah mesin virtual agar dapat berjalan pada sistem operasi yang disebut sebagai java runtime environtmen (JRE). Khusus untuk kategori pengembangan aplikasi, tidak hanya dibutuhkan sekedar sebuah virtual machine tapi juga antarmuka pemrograman aplikasi (API) yang dikemas dalam sebuah paket perangkat lunak pengembang yaitu Java Development Kit (JDK).

2.13 Metode Testing Pengujian

2.13.1 Black Box Testing

Pengujian menggunakan sekumpulan aktifitas validasi, dengan pendekatan black box testing. Menurut Shalahuddin dan Rosa (2011), black box testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai 25 dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah. Menurut Pressman (2012), black box testing juga disebut pengujian tingkah laku, memusat pada kebutuhan fungsional perangkat lunak. Teknik pengujian black box memungkinkan memperoleh serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Beberapa jenis kesalahan yang dapat diidentifikasi adalah fungsi tidak benar atau hilang, kesalahan antar muka, kesalahan pada struktur data (pengaksesan basis data), kesalahan performansi, kesalahan inisialisasi dan akhir program.