BAB 2
TINJAUAN PUSTAKA

2.1 Babakan Sangkuriang

Babakan Sangkuriang atau sering kali disebut dengan BBS merupakan salah satu kampung di desa Dayeuhkolot. Pada desa Dayeuhkolot kampung Babakan Sangkuriang termasuk wilayah RW 01 yang terdiri dari 3 RT, yaitu RT 01, RT 02, dan RT 03. Wilayahnya berdekatan dengan pinggir aliran sungai citarum dan berdekatan dengan industri atau perusahaan serta jalan Mohammad Toha. Jalan di kampung Babakan Sangkuriang juga dijadikan sebagai jalan alternatif bagi masyarakat yang ingin menuju jalan Mohammad Toha, Kabupaten Bandung ataupun ke jalan Katapang Andir, Kabupaten Bandung ketimbang harus melewati jalan Raya Dayeuhkolot. Namun saat musim hujan di kampung Babakan Sangkuriang ini sering kali mengalami banjir. Bahkan banjir yang terjadi sering kali mengakibatkan akses jalan terputus.

2.2 Data Tinggi Muka Air

Tinggi muka air merupakan ketinggian muka air pada suatu permukaan yang melebar. Data tinggi muka air sungai Citarum sudah disediakan oleh BBWSC yang telah bekerja sama dengan KOICA. Pada halaman website resminya http://103.184.53.146, akan menampilkan informasi hidrologi seperti ketinggian air dan curah hujan pada beberapa titik pemantau.

Untuk mengambil data tersebut gunakan metode *POST* pada *endpoint API* yang telah ditentukan. Untuk mengambil semua data dari pemantau tinggi muka air dan curah hujan dari semua titik pemantau gunakan *endpoint API* http://103.184.53.146/hydro/wl.json dengan metode POST. Data yang didapatkan cukup banyak seperti:

1. “wl” untuk data water level atau tinggi muka air,
2. “ymdhm” untuk data kejadian
3. “rf” untuk data rainfall atau curah hujan

```
{
    "wl": [
        {
            "relaycd": null,
            "agccd": null,
            "rfobscd": null,
            "etcaddr": null,
            "wlobscd": "206016005",
            "latMap": null,
            "lat": null,
            "lonMap": null,
            "obopndt": null,
            "obsnmsht": null,
            "addr": null,
            "lon": null,
            "obclsdt": null,
            "obsnm": "Majalaya",
            "bnkhgtlbnk": null,
            "alarmwl": null,
            "alertwl": null,
            "useyn": null,
            "mapSeq": null,
            "gsmnm": null,
            "criticalwl": null,
            "fbscd": null,
            "bnkhgtrbnk": null,
            "tm": null,
            "fstnyn": null,
            "gdt": null,
            "tmdhm": null,
            "ymd": null,
            "altitude": null,
            "mrf": null,
            "wl": "0.94",
            "hm": null,
            "rf": null
        },
     .....
     ],
 "ymdhm": "202306200100",

    "rf": [
        {
            "relaycd": null,
            "agccd": null,
            "rfobscd": "206014004",
            "etcaddr": null,
            "wlobscd": null,
            "latMap": null,
            "lat": null,
            "lonMap": null,
            "obopndt": null,
            "obsnmsht": null,
            "addr": null,
            "lon": null,
            "obclsdt": null,
            "obsnm": "Paseh-Cipaku",
            "bnkhgtlbnk": null,
            "alarmwl": null,
            "alertwl": null,
            "useyn": null,
            "mapSeq": null,
            "gsmnm": null,
```

```
            "criticalwl": null,
            "fbscd": null,
            "bnkhgtrbnk": null,
            "tm": null,
            "fstnyn": null,
            "gdt": null,
            "tmdhm": null,
            "ymd": null,
            "altitude": null,
            "mrf": "0.0",
            "wl": null,
            "hm": null,
            "rf": null
        },
       .....
       ],
}
```


Untuk pemanggilan data spesifik berdasarkan stasiun pemantau, hanya perlu id device pemantau saja. *Endpoint API* dengan spesifik stasiun dapat menggunakan http://103.184.53.146/hydro/wllist.json?timetype=1&stDt=&endDt=&wlobscd=206036003 dengan metode POST, *query* wlobscd merupakan id device sehingga ketika dimasukan id device 206036003 maka akan merujuk ke titik pemantau tinggi muka air Dayeuhkolot. Data yang didapatkan sangat banyak sekali karena data satu hari, berikut respon datanya.


```
{
"wl": [
        {
            "relaycd": null,
            "agccd": null,
            "rfobscd": null,
            "etcaddr": null,
            "wlobscd": "206036003",
            "latMap": null,
            "lat": null,
            "lonMap": null,
            "obopndt": null,
            "obsnmsht": null,
            "addr": null,
            "lon": null,
            "obclsdt": null,
            "obsnm": null,
            "bnkhgtlbnk": null,
            "alarmwl": null,
            "alertwl": null,
            "useyn": null,
            "mapSeq": null,
            "gsmnm": null,
            "criticalwl": null,
            "fbscd": null,
            "bnkhgtrbnk": null,
            "tm": "4.10",
            "fstnyn": null,
            "gdt": null,
```

```
            "tmdhm": null,
            "ymd": "2023-06-20",
            "altitude": null,
            "mrf": null,
            "wl": null,
            "hm": "00:00",
            "rf": null
        },
        .....
    ]
}
```


2.3 Banjir

Banjir adalah peristiwa tergenangnya daratan atau tanah yang biasanya kering, terjadi karena volume air yang meningkat dan melebihi kapasitas [5]. Banjir dapat disebabkan oleh luapan air yang berlebihan dari hujan lebat, jebolnya tanggul, naiknya permukaan laut atau mencairnya es. Banjir dapat menjadi bencana bila terjadi di daerah yang terpapar aktivitas manusia seperti menimbulkan kerusakan, kerugian atau korban jiwa [6]. Ada dua jenis kejadian banjir:

1. Banjir yang terjadi di daerah yang biasanya tidak terjadi banjir,
2. Banjir yang disebabkan oleh luapan banjir dari sungai yang disebabkan oleh debit air yang lebih besar dari kapasitas aliran sungai yang ada.

2.4 Peringatan

Peringatan adalah serangkaian kegiatan yang dimaksudkan untuk memberitahukan kepada masyarakat sesegera mungkin tentang kemungkinan terjadinya bencana di suatu lokasi oleh instansi yang berwenang. Masyarakat memiliki hak untuk berpartisipasi dalam pengambilan keputusan tentang kegiatan penanggulangan bencana, terutama yang berdampak pada dirinya dan komunitasnya, serta wajib memberikan informasi yang akurat kepada masyarakat tentang penanggulangan bencana [7]. Terdapat berbagai cara untuk memberi peringatan seperti menggunakan Lonceng Besar, Pengeras Suara, Sirene, radio, televisi dan sejenisnya. Adapun juga yang sudah menggunakan teknologi seperti notifikasi melalui smartphone.

2.5 Javascript

Javascript adalah bahasa script berdasarkan pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Dimana objek tersebut dapat berupa suatu *window*, *frame*, *URL*, dokumen, *form*, *button*, atau item yang lain. Semuanya itu memiliki properti yang saling berhubungan dengannya dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lainnya [8].

2.6 Cloud Messaging API

Cloud Messaging API adalah solusi pengiriman pesan lintas platform yang dapat mengirim pesan tanpa biaya. Cloud Messaging API merupakan salah satu teknologi dari Firebase. Dengan menggunakan Cloud Messaging dapat memberitahu aplikasi klien. Kemudian mengirim pesan notifikasi untuk mendorong interaksi kepada pengguna. Untuk penggunaan seperti *instant messaging*, pesan dapat mentransfer *payload* hingga 4.000byte ke aplikasi klien. Untuk dapat menggunakan Cloud Messaging API diperlukan token (Server Key) yang nantinya akan disisipkan pada kode aplikasi atau sistem yang dibangun. Kemampuan utama Cloud Messaging API adalah:

1. Mengirim pesan notifikasi atau pesan data

   Mengirim pesan notifikasi yang ditampilkan kepada pengguna. Atau mengirim pesan data dan menentukan sepenuhnya apa yang terjadi dalam kode aplikasi.
2. Penargetan pesan serbaguna

   Mendistribusikan pesan ke aplikasi klien dengan salah satu dari 3 cara, yaitu ke satu perangkat, ke grup perangkat, atau ke perangkat yang berlangganan topik.
3. Mengirim pesan dari aplikasi klien

   Mengirim konfirmasi, chat, dan pesan lain dari perangkat kembali ke server melalui saluran koneksi Cloud Messaging API yang andal dan hemat baterai.

Untuk mengirim pesan melalui Firebase Admin SDK atau protokol server Cloud Messaging API. Implementasi Cloud Messaging API mencakup dua komponen utama untuk mengirim dan menerima pesan:

1. Lingkungan tepercaya seperti Cloud Functions for Firebase atau server aplikasi yang akan digunakan untuk membuat, menargetkan, dan mengirim pesan.
2. Aplikasi klien Apple, Android, atau web (JavaScript) yang menerima pesan melalui layanan transportasi spesifik platform yang sesuai. [9]

2.7 Google Maps API

Google Maps API merupakan sebuah layanan yang diberikan oleh Google untuk memanfaatkan Google Maps dalam mengembangkan aplikasi. Google Maps merupakan sebuah layanan peta digital gratis yang disediakan oleh Google dan bersifat open-source. Google Maps juga dapat diintegrasikan ke dalam aplikasi-aplikasi baik berbasis mobile maupun berbasis website dengan menggunakan Google Maps API [10]. Untuk dapat menggunakan Google Maps API diperlukan token (API Keys) yang nantinya akan disisipkan pada kode aplikasi atau sistem yang dibangun.

2.8 Android

Android merupakan sistem operasi berbasis linux dengan sumber kode terbuka dibawah lisensi apache 2.0 yang dibuat untuk beragam perangkat yang berbeda. Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Tujuan awal pengembangan Android digunakan untuk mengembangkan sebuah sistem operasi canggih yang diperuntukan bagi kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi Symbian dan Windows Mobile [11].

Pada Tanggal 17 Agustus 2005, Google mengakuisisi Android Inc. dan menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google.

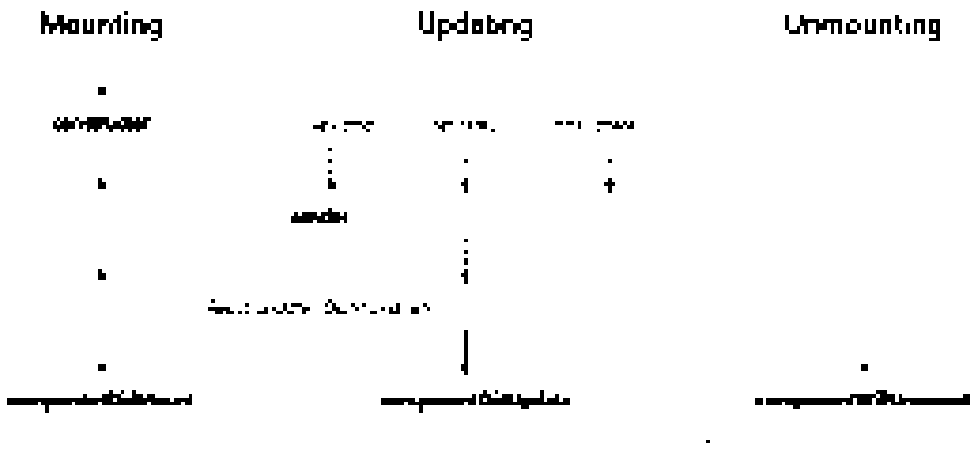
Pendiri Android Inc. seperti Rubin, Minder dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Pada Tahun 2023 Android sudah mencapai versi Android 13 dengan codename Tiramisu.

2.9 React Native

React Native adalah kerangka kerja berbasis javascript yang dibuat untuk membangun aplikasi *mobile* pada platform IOS dan Android yang dikembangkan oleh Facebook. Dengan begitu, react native memungkinkan untuk developer web untuk membuat aplikasi mobile. React Native membawa kerangka UI deklaratif dan multiplatform, sehingga dapat digunakan untuk iOS dan Android [12]. React Native sendiri memiliki keunggulan seperti:

1. *Declarative* atau Deklaratif. Bereaksi membuatnya mudah untuk membuat UI interaktif. Tampilan deklaratif membuat kode lebih dapat diprediksi dan lebih mudah di-debug.
2. *Component-Based* atau Berbasis Komponen. Membangun komponen terenkapsulasi, lalu menyusunnya untuk membuat UI yang kompleks.
3. *Developer Velocity* atau Kecepatan Pengembang. Perubahan lokal dalam hitungan detik. Perubahan pada kode JavaScript dapat dimuat ulang langsung tanpa membangun ulang aplikasi asli.
4. *Portability* atau Portabilitas. Dapat digunakan kembali kode di iOS, Android, dan platform lainnya.

Dalam react native terdapat method *Lifecycle* biasa digunakan, seperti pada gambar 2.2



Gambar 2.1 Lifecycle React Native

Sumber: https://legacy.reactjs.org/docs/react-component.html

Ada 5 lifecyle yang biasa digunakan oleh developer, terdiri dari:

1. constructor

   Metode ini berjalan sebelum komponen kelas dipasang. Biasanya, konstruktor hanya digunakan untuk dua tujuan di React. Ini memungkinkan Anda mendeklarasikan *state* lokal dan mengikat metode kelas ke instance kelas.

2. render

   Metode ini adalah satu-satunya metode yang diperlukan dalam class component.render(). Metode ini harus menentukan apa yang ingin ditampilkan pada layar.

3. componentDidMount

   Metode ini akan memanggilnya ketika komponen ditambahkan (dipasang) ke layar. Ini adalah tempat umum untuk memulai pengambilan data, mengatur langganan, atau memanipulasi data atau element.

4. componentDidUpdate

   Metode ini akan memanggilnya segera setelah komponen direnderr ulang dengan props atau state yang telah diperbarui. Metode ini tidak dipanggil untuk render awal.

5. componentDidWillUnmount

Metode ini akan dipanggil segera sebelum komponen dilepas dan dihancurkan. Biasa dilakukan untuk membatalkan pengambilan data dan melakukan pembersihan yang diperlukan [13].

2.10 Web Service

Web Service adalah mekanisme komunikasi yang didefinisikan antara sistem komputer yang berbeda. Tanpa web service, komunikasi peer-to-peer kustom menjadi rumit dan spesifik platform. Ini seperti ratusan jenis hal yang perlu dipahami dan ditafsirkan oleh web. Jika sistem computer sejajar dengan protokol yang dapat dipahami web dengan mudah, ini sangat membantu [14]. Jenis web service yang biasanya digunakan adalah *Simple Object Access Protocol* (SOAP) dan *Representational State Transfer* (REST). Jenis web service SOAP memanfaatkan komunikasi data yang berbasis XML. Sedangkan REST memanfaatkan *Hypertext Transfer Protocol* (HTTP) untuk komunikasi data antara klien dan server. Service yang digunakan menggunakan metode milik HTTP antara lain:

1. GET digunakan untuk mengambil atau membaca data,
2. PUT digunakan untuk mengubah data,
3. POST digunakan untuk mengirimkan data, dan
4. DELETE digunakan untuk menghapus data.[15]

Pesan yang diterima dari server berupa kode HTTP berhasil atau gagal. Berikut adalah kode-kode HTTP yang sering digunakan dalam penggunakan REST API:

a. 200 OK
   Perintah yang dikirim ke server benar dan berhasil dijalankan.
b. 400 *Bad Request*
   Perintah yang dikirim ke server berisi isian yang salah.
c. 401 *Unauthorized*
   Pengirim perintah mengirimkan kode kunci yang salah.
d. 403 *Forbidden*
   Pengirim perintah tidak memiliki hak akses ke dalam resource yang dituju.

e. 404 *Not Found*

Resource yang dituju tidak ditemukan dalam server.

f. 500 Internal Server Error

Server atau potongan program dalam resource mengalami kesalahan [16].

2.11 PHP

Bahasa pemrograman PHP (*PHP: Hypertext Preprocessor*) merupakan bahasa pemrograman untuk membuat website yang bersifat *server-side scripting* dan juga bersifat dinamis. PHP dapat dijalankan pada berbagai sistem operasi seperti Windows, Linux, dan Mac Os. Kemudian PHP juga mendukung web server seperti Apache, Microsoft ISS, Caudium, dan PWS. PHP dapat memenfaatkan database seperti MySQL, PostgreSQL, Oracle, dan lainnya untuk menghasilkan halaman web yang dinamis [17].

2.12 MySQL

MySQL merupakan suatu jenis database server yang sangat terkenal. MySQL termasuk jenis *Relational Database Manajement System* (RDBMS). MySQL mendukung Bahasa pemrograman PHP, bahasa permintaan yang terstruktur, karena pada penggunaannya SQL memiliki berberapa aturan yang telah distandarkan oleh asosiasi yang bernama ANSI. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya [18].

2.13 Global Positioning System

Global Positioning System (GPS) adalah sebuah alat atau sistem yang dapat memberikan sebuah informasi mengenai tata letak atau kebaradaan seseorang yang ada di permukaan bumi. Untuk menenttukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak (data *latitude* dan *longitude*), kecepatan, arah, dan waktu. Data angka *latitude* merupakan angka garis lintang dan yang satunya lagi adalah angka *longitude* merupakan angka garis

bujur. Jadi angka *latitude* dan *longitude* ini pada dasarnya adalah angka dalam sistem koordinat geografis yang digunakan untuk menentukan lokasi di suatu tempat pada permukaan bumi kita ini. Sistem yang serupa dengan GPS antara lain GLONASS Rusia, Galileo Uni Eropa, IRNSS India. Sistem GPS, yang nama aslinya adalah NAVSTAR GPS (*Navigation Satellite Timing and Ranging Global Positioning System*), mempunyai tiga segmen yaitu:

1. satelit,
2. pengontrol, dan
3. penerima / pengguna.

Satelit GPS yang mengorbit bumi, dengan orbit dan kedudukan yang tetap (koordinatnya pasti), seluruhnya berjumlah 24 buah dimana 21 buah aktip bekerja dan 3 buah sisanya adalah cadangan [19].

2.14 Google Directions API

Google Directions API adalah layanan yang menerima permintaan HTTP dan menampilkan arah berformat JSON atau XML antar-lokasi. Dengan Directions API, bisa mendapatkan rute untuk beberapa moda transportasi, seperti transportasi umum, mengemudi, berjalan kaki, atau bersepeda. Google Directions API menghitung rute antar-lokasi, termasuk detail berikut:

1. Rute untuk beberapa moda transportasi, termasuk transportasi umum, mengemudi, berjalan kaki atau bersepeda.
2. Rute multibagian menggunakan serangkaian titik jalan.
3. Tentukan asal, tujuan, dan titik jalan dengan beberapa cara, termasuk sebagai string teks (misalnya "Chicago, IL", atau "Darwin, NT, Australia"), ID tempat, atau koordinat lintang/bujur.

Directions API menampilkan rute yang paling efisien saat menghitung rute. Faktor produk dalam elemen perjalanan berikut ini saat menentukan rute yang paling efisien:

1. Waktu tempuh (utama)
2. Jarak

3. Jumlah belokan [20]

2.15 JavaScript Object Notation

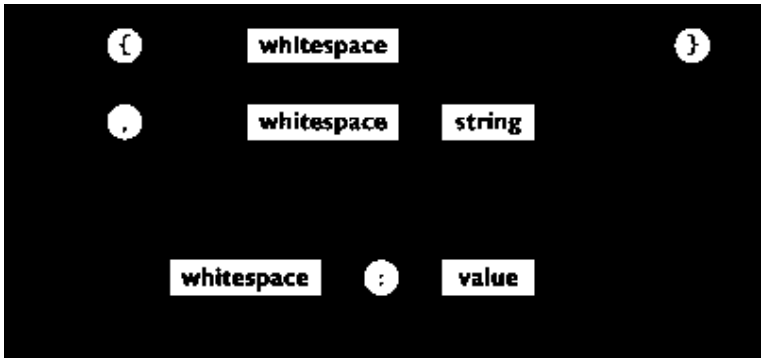
JavaScript Object Notation (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [21]. JSON dibangun di atas dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai yang diurutkan. Dalam kebanyakan bahasa, ini direalisasikan sebagai larik, vektor, daftar, atau urutan.

Ini adalah struktur data universal. Hampir semua bahasa pemrograman modern mendukungnya dalam satu atau lain bentuk. JSON menggunakan bentuk sebagai berikut:

1. Objek

   Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).

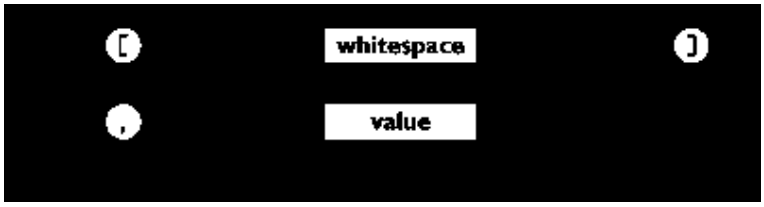


Gambar 2.2 JSON dalam bentuk objek

Sumber: https://www.json.org/json-id.html

2. Larik

   Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

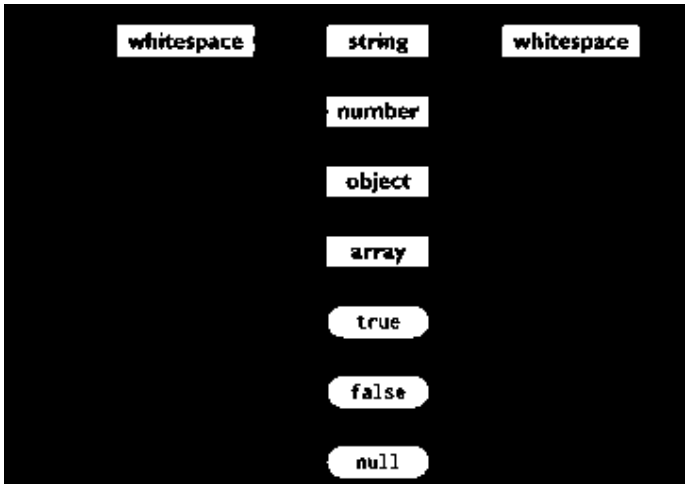


Gambar 2.3 JSON dalam bentuk larik

Sumber: https://www.json.org/json-id.html

3. Nilai

   Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.

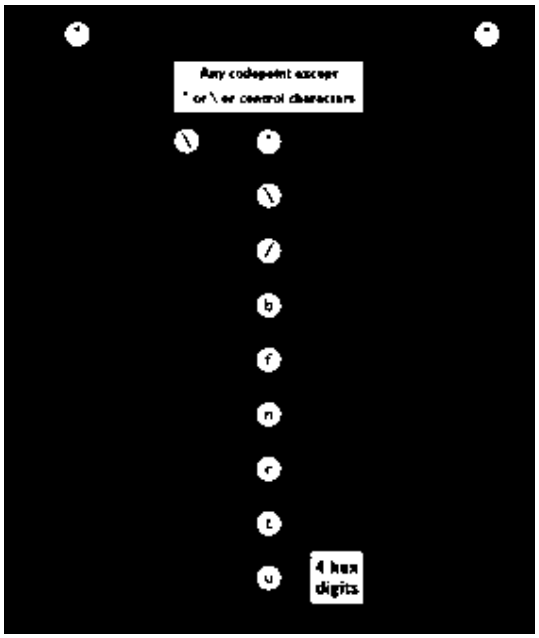


Gambar 2.4 JSON dalam bentuk nilai

Sumber: https://www.json.org/json-id.html

4. String

   String adalah kumpulan dari nol atau lebih karakter *Unicode*, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.

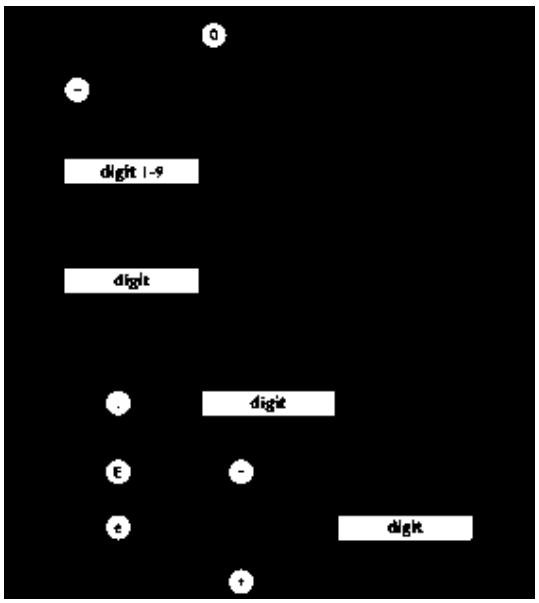


Gambar 2.5 JSON dalam bentuk string

Sumber: https://www.json.org/json-id.html

5. Number

   Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2.6 JSON dalam bentuk angka

Sumber: https://www.json.org/json-id.html

2.16 Application Programming Interface

Application Programming Interface (API) adalah konsep fungsi antarmuka pemrograman aplikasi, yang menjadi salah satu cara agar suatu aplikasi dapat diakses dan dimanfaatkan oleh pihak lain tanpa mengubah struktur kode utama maupun database pada sistem, serta memudahkan komunikasi antar sistem meskipun berbeda platform. Web Service berperan dalam memberikan akses pengguna dalam proses pengambilan data, sehingga data akan disajikan kepada para pengguna saat mengakses API [22].

2.17 Firebase Authentication

Firebase Authentication menyediakan layanan backend, SDK yang mudah digunakan, dan library UI siap pakai untuk mengautentikasi pengguna ke aplikasi. Firebase Authentication juga mendukung autentikasi menggunakan sandi, nomor telepon, serta penyedia identitas gabungan yang populer seperti Google, Facebook, Twitter, dan lain-lain. Firebase Authentication terintegrasi erat dengan layanan

Firebase lainnya dan memanfaatkan berbagai standar industri, seperti OAuth 2.0 dan OpenID Connect, sehingga dapat dengan mudah diintegrasikan dengan backend kustom.

Anda dapat memproses login pengguna ke aplikasi Firebase dengan menggunakan Firebase Authentication SDK untuk mengintegrasikan salah satu atau beberapa metode login ke aplikasi Anda secara manual.

1. Autentikasi berbasis email dan sandi

   Mengautentikasi pengguna dengan alamat email dan sandi. Firebase Authentication SDK menyediakan metode untuk membuat dan mengelola pengguna yang login menggunakan alamat email dan sandi. Firebase Authentication juga menangani pengiriman email reset sandi.
2. Integrasi penyedia identitas gabungan

   Mengautentikasi pengguna dengan mengintegrasikan penyedia identitas gabungan. Firebase Authentication SDK menyediakan metode yang memungkinkan pengguna untuk login dengan akun Google, Facebook, Twitter, dan GitHub.
3. Autentikasi nomor telepon

   Mengautentikasi pengguna dengan mengirim pesan SMS ke ponsel.
4. Integrasi sistem autentikasi khusus

   Menghubungkan sistem login aplikasi yang ada ke Firebase Authentication SDK, serta memperoleh akses ke Firebase Realtime Database dan layanan Firebase lainnya.
5. Autentikasi anonim

   Menggunakan fitur yang memerlukan autentikasi tanpa mewajibkan pengguna untuk login terlebih dahulu dengan membuat akun anonim sementara [23].

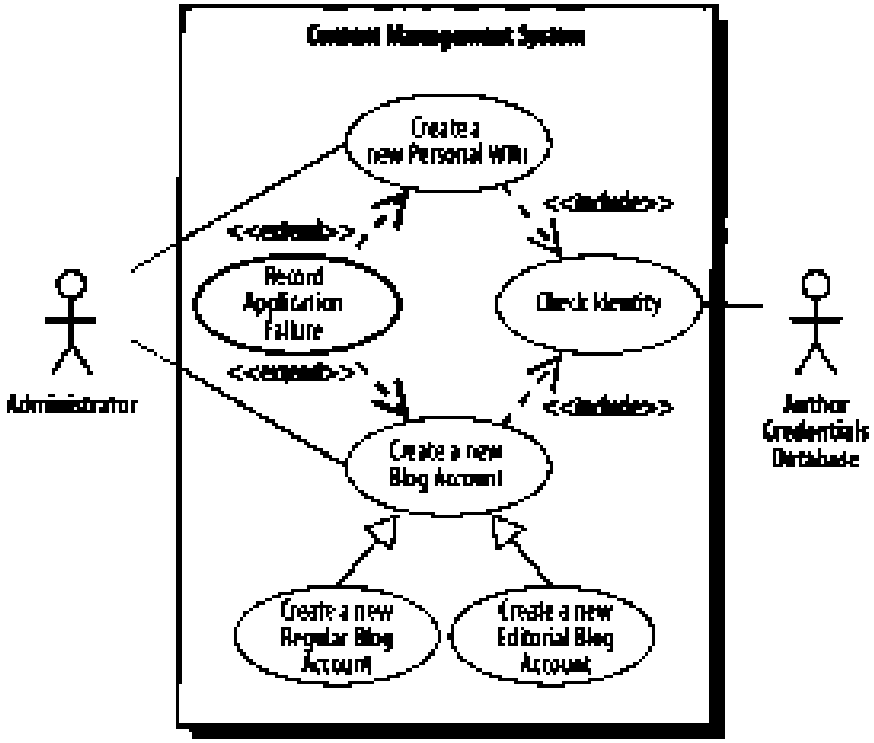
2.18 Unified Modeling Language

Unified Modeling Language atau lebih sering dikenal dengan sebutan UML, adalah cara standar untuk memodelkan sistem, khususnya sistem perangkat lunak

dengan desain *object-oriented* (OO) [24]. Dalam UML ada beberapa diagram seperti *use case diagram, activity diagram, class diagram* dan *sequence diagram*.

2.18.1 Use Case Diagram

Use case diagram adalah interaksi antara sistem dan pengguna atau sistem eksternal lainnya, juga sangat membantu dalam memetakan persyaratan untuk sistem. *Use case diagram* menangkap fungsionalitas yang disediakan sistem. *Use case diagram* adalah inti dari model, karena mempengaruhi dan memandu semua elemen lain dalam desain sistem [24].



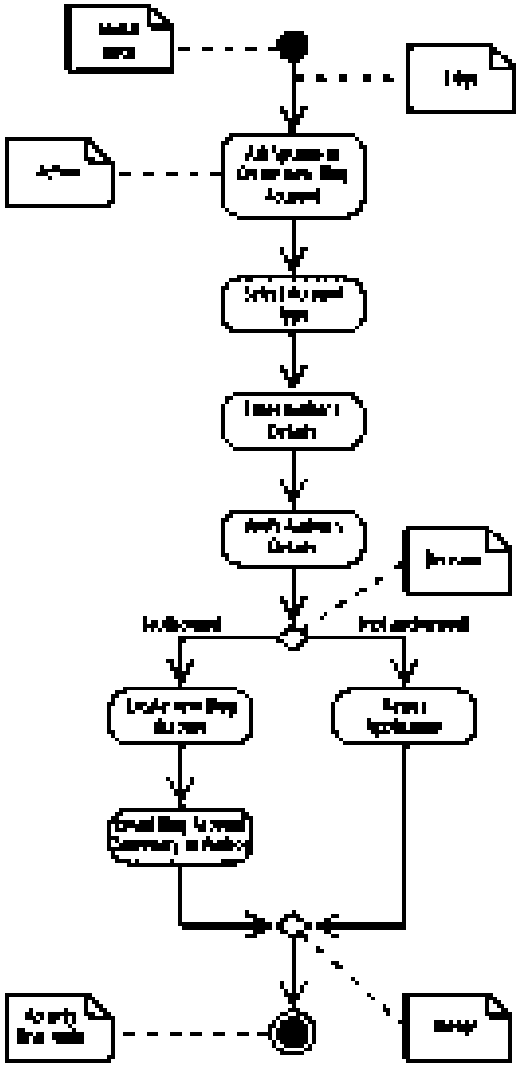
Gambar 2.7 Contoh *use case diagram*

Sumber: Buku Learning UML 2.0 - Russ Miles and Kim Hamilton

2.18.2 Activity Diagram

Activity diagram adalah salah satu diagram UML yang paling mudah diakses karena menggunakan simbol yang mirip dengan notasi diagram alur yang dikenal luas. Oleh karena itu, berguna untuk menggambarkan proses kepada khalayak luas. *Activity diagram* memungkinkan untuk menentukan bagaimana

sistem akan mencapai tujuannya. *Activity diagram* menunjukkan tindakan tingkat tinggi yang dirangkai bersama untuk mewakili proses yang terjadi di sistem [24].



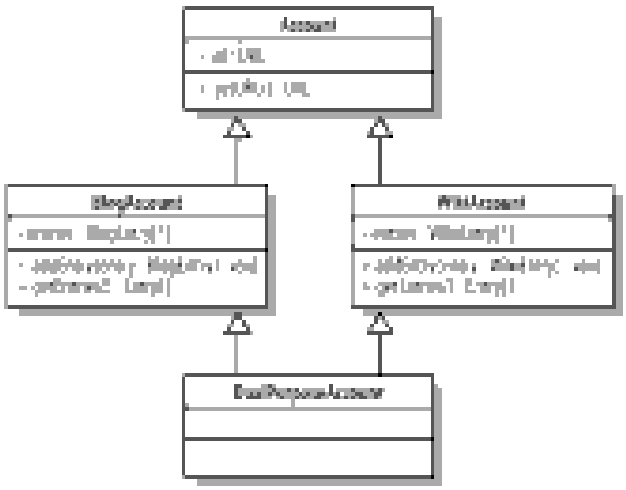
Gambar 2.8 Contoh *activity diagram*

Sumber: Buku Learning UML 2.0 - Russ Miles and Kim Hamilton

2.18.3 Class Diagram

Class diagram menunjukkan jenis-jenis objek dalam sistem. *Class* adalah inti dari sistem berorientasi objek apa pun. Oleh karena itu, diagram UML yang paling populer adalah *class diagram.* Struktur sistem terdiri dari kumpulan potongan yang sering disebut sebagai objek. *Class* menjelaskan berbagai jenis

objek yang dapat dimiliki sistem, dan *class diagram* menunjukkan kelas-kelas ini dan hubungannya [24].



Gambar 2.9 Contoh class diagram

Sumber: Buku Learning UML 2.0 - Russ Miles and Kim Hamilton

2.18.4 Sequence Diagram

Sequence diagram adalah tentang menangkap urutan interaksi antara bagian-bagian sistem. Dengan menggunakan *Sequence diagram*, dapat menjelaskan interaksi mana yang akan dipicu saat kasus penggunaan tertentu dijalankan dan dalam urutan apa interaksi tersebut akan terjadi. *Sequence diagram* menunjukkan banyak informasi lain tentang suatu interaksi, tetapi keunggulannya adalah cara yang sederhana dan efektif untuk mengomunikasikan urutan kejadian dalam suatu interaksi [24].

Gambar 2.10 Contoh sequence diagram

Sumber: Buku Learning UML 2.0 - Russ Miles and Kim Hamilton