BAB 2

TINJAUAN PUSTAKA

2.1. Ruang Lingkup Objek Penelitian

Objek penelitian yang sedang diteliti adalah kantor Desa Dayeuhkolot yang berada di Kabupaten Bandung, yaitu bertempat di Jl.Mochamad Rahmat Djajadinata No.62 Bandug 40258.

2.1.1. Sejarah Desa Dayeuhkolot

Desa Dayeuhkolot merupakan salah satu pemerintahan tingkat desa yang berada di wilayah Kecamatan Dayeuhkolot Kabupaten Bandung. Dilihat dari letak Geografisnya Desa Dayeuhkolot, Sebelah utara berbatasan dengan Sungai Citarum / Kecamatan Baleendah. Sebelah timur berbatasan dengan Desa Citeureup. Sebelah selatan berbatasan dengan Sungai Citarum. Sebelah barat berbatasan dengan Desa Citeureup.

1. Keadaan Fisik Desa Dayeuhkolot sebagai berikut :
    1. Keadaan Tanah Tanah Daratan luasnya 4816 ha, termasuk tanah kondisi Subur. Dengan curah hujan rata-rata 2500 mm/pertahun.
    2. Keadaan Iklim Seperti halnya daerah desa-desa yang lain keadaan iklim di Desa Dayeuhkolot yaitu dengan curah hujan 2500 mm/tahun, jumlah bulan hujan 6 bulan dengan suhu rata-rata 28 º C, dengan tinggi tempat dari permukaaan laut 1250 mdl.
2. Keadaan Sosial Ekonomi Desa Dayeuhkolot sebagai berikut :
    1. Jumlah Penduduk seluruhnya 14.520 Jiwa Jumlah Laki-laki 6.716 Jiwa Jumlah Perempuan 7.804 Jiwa.
    2. Kewarganegaraan Semua penduduk Desa Dayeuhkolot Kecamatan Dayeuhkolot Kabupaten Bandung adalah WNI terdiri dari 7.804 Jiwa.

3. Kepadatan Penduduk adalah 1120 Jiwa/ km2.

2.1.2. Visi

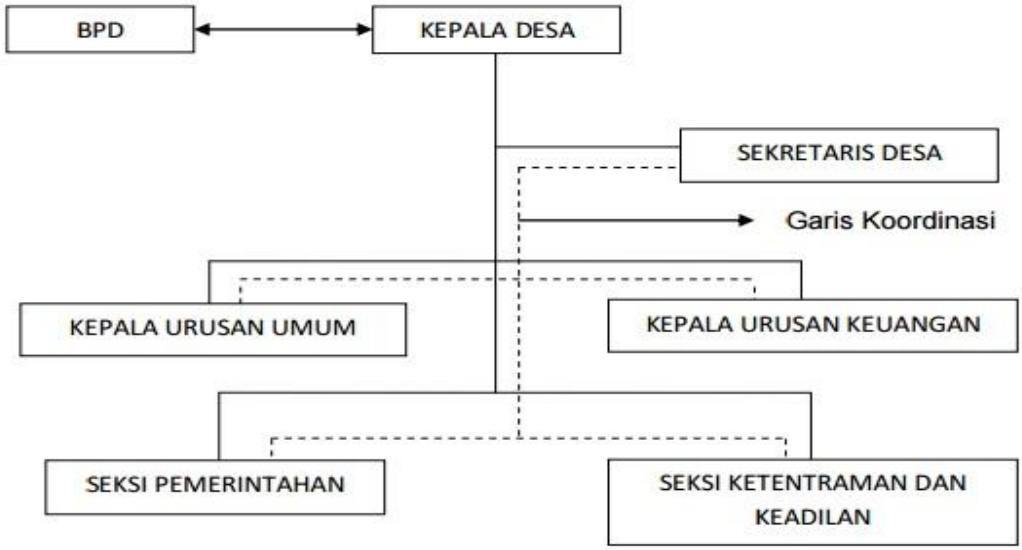
Terlaksananya Manajemen Pelayanan Prima kepada Masyarakat.

2.1.3. Misi

1. Melaksanakan penyelenggaraan pemerintahan Daerah di Wilayah Kecamatan yang transparan, bersih, berwibawa dengan melibatkan peran serta masyarakat.
2. Mewujudkan kondisi aman, tertib, damai, dan dinamis.
3. Mewujudkan pelayanan yang cepat, murah dan tepat.
4. Mewujudkan pengembangan otonomi daerah.
5. Melaksanakan Pembinaan terhadap pemerintah desa dan lembaga kemasyarakatan.

2.1.4. Struktur Organisasi Kantor Desa Dayeuhkolot

Berikut ini adalah susunan organisasi yang ada di Kantor Desa Dayeuhkolot.



Gambar 2.1 Struktur Organisasi Kantor Desa Dayeuhkolot [1]

1. Kepala Desa

Kepala Desa Kepala Desa mempunai tugas menyelenggarakan urusan pemerintahan, pembangunan, dan kemasyarakatan. Dalam melaksanakan kewajibannya Kepala Desa mempunyai wewenang sebagai berikut :

1. Memimpin penyelenggaraan pemerintahan desa berdasarkan kebijakan yang ditetapkan bersama BPD.
2. Mengajukan rancangan peraturan desa.
3. Menetapkan peraturan desa yang telah mendapat persetujuan bersama BPD.
4. Menyusun dan mengajukan rancangan peraturan desa mengenai APB Desa untuk dibahas dan ditetakan bersama BPD.
5. Membina kehidupan masyarakat Desa.
6. Mengkoordinasikan pembangunan desa secara partisipatif.
7. Memina perekonomian desa.

Adapun kewajiban Kepala Desa sebagai berikut :

1. Memegang teguh dan mengamalkan Pancasila, melaksanakan Undang-Undang Dasar Negara Republik Indonesia 1945 serta mempertahankan dan memelihara keutuhan Negara Kesatuan Republik Indonesia.
2. Meningkatkan kesejahteraan masyarakat.
3. Memelihara ketenteraman dan ketertiban masyarakat.
4. Melaksanakan kehidupan demokrasi.
5. Melaksanakan prinsip tata pemerintahan desa yang bersih dan bebas dari KKN.
6. Menjalin hubungan hubungan kerja dengan seluruh mitra kerja pemerintahan desa.
7. Menaati dan menegakan seluruh peraturan perundangundangan.

2. Sekertaris Desa

Sekretaris Desa dalam membantu Kepala Desa mempunyai tugas sebagai berikut :

1. Memberikan saran dan pendapat kepada Kepala Desa
2. Kegiatan yang dilaksanakan unsur teknis lapangan, dan unsur pembantu.
3. Memberikan informasi mengenai keadaan Sekretariat Desa dan keadaan desa.
4. Merumuskan program kegiatan Kepala Desa.
5. Membantu Kepala Desa dalam penyusunan atau perumusan rancangan Peraturan Desa.
6. Membantu Kepala Desa dalam menyusun laporan penyelenggaraan pemerintah desa kepada Bupati melalui Camat.
7. Membantu Kepala Desa dalam menyusun laoran keterangan pertanggungjawaban kepada BPD.
8. Mengadakan dan melaksanakan persiapan rapat dan mencatat hasil-hasil rapat.
9. Menyusun Rancangan Anggaran Penerimaan dan Belanja Desa.
10. Menyusun keuanangan Desa.
11. Mengadakan kegiatan inventarisasi (mencatat, mengawasi, memelihara) kekayaan Desa.
12. Melaksanakan administrasi kepegawaian Aparat Desa.

3. Kepala Urusan Umum

Kepala urusan Umum dalam membantu Sekretaris Desa mempunyai tugas sebagai berikut:

1. Menerima dan mengendalikan surat masuk dan keluar, serta melaksanakan tata kearsipan.
2. Melaksanakan pengetikan surat-surat hasil persidangan dan rapat-rapat atau naskah lainnya.
3. Melaksanakan penyediaan, penyimpanan dan pendistribusian alat-alat tulis kantor, serta pemeliharaan dan perbaikan peralatan kantor.

4. Menyusun jadwal serta mengikuti perkembangan peralatan piket.
5. Melaksanakan dan mengusahakan ketertiban dan kebersihan kantor serta bangunan lain milik desa.
6. Menyelenggarakan pengelolaan administrasi kepegawaian aparat desa.
7. Melaksanakan pengelolaan administrasi kepegawaian desa.
8. Melaksanakan pengelolaan buku administrasi umum.
9. Mencatat inventarisasi kekayaan desa.
10. Melaksanakan persiapan, penyelenggaraan rapat dan penerimaan tamu dinas, serta kegiatan kerumahtanggaan pada umumnya.
11. Melaksanakan tugas lain yang diberikan oleh Kepala Desa.

4. Kepala Urusan Keungan

Kepala Urusan Keuangan dalam membantu Sekretaris Desa mempunyai tugas sebagai berikut:

1. Mengelola keuangan desa dan sumber-sumber keuangan desa lainnya.
2. Melaksanakan pencatatan pengelolaan buku administrasi keuangan.
3. Melakukan pembuatan pertanggungjawaban keuangan.
4. Melaksanakan tugas lain yang diberikan oleh Kepala Desa.

5. Seksi Pemerintahan

Seksi Pemerintahan mempunyai tugas sebagai berikut :

1. Melaksanakan administrasi pemerintahan desa.
2. Melaksanakan asministrasi penduduk di desa.
3. Mengadakan kagiatan pencatatan mutasi tanah dan pencatatan administrasi pertanahan.
4. Melaksanakan dan memberikan pelayanan terhadap masyarakat dalam hal pembuatan KTP.
5. Melaksanakan kegiatan monografi/ profil desa.
6. Melaksanakan penyelenggaraan buku administrasi desa dan keputusan Kepala Desa.

7. Melakukan tugas lain yang diberikan oleh Kepala Desa.

6. Seksi Ketentraman dan Ketertiban

Seksi Ketentraman dan Ketertiban mempunyai tugas sebagai berikut :

1. Menjaga ketentraman, keamanan dan ketertiban masyarakat secara umum.
2. Mengkoordinasikan dan melaksanakan upaya-upaya dalam rangka menjaga keamanan dan ketertiban lingkungan.
3. Melaksanakan kebiatan kemasyarakatan termasuk kegiatan ketentraman dan ketertiban Perlindungan Masyarakat (LINMAS).
4. Menginventarisasi kegiatan dan personil keamanan lingkungan.
5. Melaksanakan tugas lain yang diberikan oleh Kepala Desa.

2.2. Landasan Teori

Landasan teori menjelaskan beberapa definisi dan teori yang berkaitan dengan pembangunan aplikasi Desa Siaga Bencana sebagai dasar pemahaman dalam sebuah sistem serta metode yang digunakan untuk kegiatan pembangunan aplikasi tersebut.

2.2.1. Desa Siaga

Desa dan Kelurahan Siaga Aktif adalah bentuk pengembangan dari Desa Siaga yang telah dimulai sejak tahun 2006. Desa atau Kelurahan Siaga Aktif adalah desa atau yang disebut dengan nama lain atau kelurahan,sebagai berikut :

1. Penduduknya dapat mengakses dengan mudah pelayanan kesehatan dasar yang memberikan pelayanan setiap hari melalui Pos Kesehatan Desa (Poskesdes) atau sarana kesehatan yang ada di wilayah tersebut seperti, Pusat Kesehatan Masyarakat Pembantu (Pustu), Pusat Kesehatan Masyarakat (Puskesmas) atau sarana kesehatan lainnya.

2. Penduduknya mengembangkan UKBM dan melaksanakan survailans berbasis masyarakat (meliputi pemantauan penyakit, kesehatan ibu dan anak, gizi, lingkungan dan perilaku), kedaruratan kesehatan dan penanggulangan bencana, serta penyehatan lingkungan sehingga masyarakatnya menerapkan Perilaku Hidup Bersih dan Sehat (PHBS).

Berdasarkan pengertian tersebut di atas maka Desa atau Kelurahan Siaga Aktif memiliki komponen sebagai berikut :

1. Pelayanan kesehatan dasar.
2. Pemberdayaan masyarakat melalui pengembangan UKBM dan mendorong upaya survailans berbasis masyarakat, kedaruratan kesehatan dan penanggulangan bencana serta penyehatan lingkungan.
3. Perilaku Hidup Bersih dan Sehat (PHBS).

Pengembangan Desa dan Kelurahan Siaga Aktif merupakan program lanjutan dan akselerasi dari program Pengembangan Desa Siaga yang sudah dimulai pada tahun 2006. Pengembangan Desa dan Kelurahan Siaga Aktif dilaksanakan melalui pemberdayaan masyarakat, yaitu upaya memfasilitasi proses belajar masyarakat desa dan kelurahan dalam memecahkan masalah-masalah kesehatannya.

Oleh karena merupakan upaya pembangunan desa dan kelurahan, maka program ini memerlukan peran aktif dari berbagai pihak mulai dari pusat, provinsi, kabupaten, kota, kecamatan, sampai ke desa dan kelurahan.

2.2.2. Bencana Alam

Menurut *Asian Disaster Reduction Center* “bencana alam adalah gangguan serius pada masyarakat yang bisa menyebabkan kerugian secara meluas dan dirasakan oleh semua lapisan masyarakat dan lingkungan, dan untuk mengatasinya dibutuhkan kemampuan yang melebihi kemampuan manusia”. Bencana alam menjadi salah satu

peristiwa yang tidak dapat diprediksi kapan terjadinya dan tentunya sangat tidak diharapkan oleh semua makhluk hidup.

Bencana alam banyak ragamnya, tanah longsor, banjir, tsunami, gempa bumi, dan gunung meletus merupakan beberapa contoh dari bencana alam. Setiap bencana alam juga memiliki faktor-faktor sendiri yang dapat memicu terjadinya sebuah bencana alam. Bencana alam dapat terjadi kapanpun dan dimanapun, terkadang bencana alam dapat terjadi tanpa kita duga sebelumnya.[3]

2.2.3. Banjir

Banjir merupakan keadaan dimana suatu wilayah berada pada posisi yang tergenang oleh air dalam volume yang begitu banyak dan begitu besar dan tidak dapat ditampung lagi oleh sungai, bendungan serta tidak dapat diserap lagi oleh tanah dan pohon yang berfungsi sebagai daerah resapan.

Pada dasarnya banjir tidak hanya terjadi didaerah yang rendah, banjir juga dapat terjadi ditempat yang tinggi. Karena banjir bisa terjadi dimana saja jika tempat itu tidak memiliki daerah resapan yang cukup. Terutama ketika musim penghujan tiba, hujan yang turun terus menerus menyebabkan air yang turun semakin banyak sehingga memenuhi sungai dan saluran-saluran air yang telah penuh. Dan jika tidak ada tempat lain lagi untuk menampung air hujan yang turun, maka disitu akan terjadi bencana banjir.

Banyak sekali faktor yang menyebabkan banjir, terutama curah hujan, kondisi topografi, jenis penutup tanah, jenis tanah, dan lain-lain. Penyebab tersebut merupakan parameter-parameter yang dapat menyebabkan banjir. Faktor kemiringan lereng, ketinggian, dan geologi sangat menentukan daerah rentan banjir. Selanjutnya kondisi fisik ditambah dengan perilaku manusia akan menentukan tingkat kerawanannya.[4]

2.2.4. Android

Android adalah sistem operasi untuk *handphone* yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli Android Inc. pendatang baru yang membuat peranti lunak untuk *handphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance, konsorsium* dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode–kode Android di bawah *lisensi Apache*, sebuah lisensi perangkat s*oftware* dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD).

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

Android merupakan sistem operasi yang berkembang dengan pesat, namun tidak menjadikannya sistem operasi yang sempurna ada beberapa kekurangan dari sistem operasi Android diantaranya Android terkesan rumit, karena mempunyai banyak sekali *widget* maupun aplikasi dengan banyak pengaturan sehingga pengguna harus banyak belajar mengenai Android, selain itu Android yang merupakan sistem

operasi terbuka sehingga pengguna dapat memasang aplikasi di luar toko aplikasi yang ditawarkan oleh perangkat Android tersebut sehingga sangat rentan terkena ancaman *malware* atau *virus*. Tidak semua perangkat Android dapat langsung memperbarui sistem operasi terbaru, karena produsen s*martphone* lebih mementingkan produk baru untuk diberi sistem operasi yang terbaru, dibanding dengan memberi pemberitahuan tentang *update* sistem operasi terbaru sehingga membutuhkan waktu lama untuk memperbarui sistem operasi bagi beberapa perangkat.

Dalam penelitian ini Android berguna sebagai sistem oprasi dalam membangun aplikasi Desa Siaga Bencana Di Desa Dayeuhkolot berbasis Android.

2.2.4.1.Versi Android

Berikut adalah versi-versi Android yang telah dibuat sampai saat ini yang sedang digunakan.

1. Versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail,* dan pemberitahuan *email.*

2. Versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Versi 2.0/2.1 (*Eclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*,

peningkatan *Google Maps* 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru.

4. Versi 2.2 (Froyo: *Frozen Yoghurt*)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript *engine* yang dipakai *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan *WiFi Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi Android *Market.*

5. Versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, Android versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan, peningkatan fungsi *copy paste*, layar antar muka *(User Interface)* didesain ulang, dukungan format video VP8 dan WebM, efek audio baru *(reverb, equalization, headphone virtualization, dan bass boost)*, dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

6. Versi 3.0/3.1 (*Honeycomb*)

Android *Honeycomb* dirancang khusus untuk *tablet.* Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung *multi prosesor* dan juga akselerasi perangkat keras *(hardware)* untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah *Motorola Xoom*.

7. Versi 4.0 (*Ice Cream Sandwich*)

Android *Ice Cream Sandwich* yang dirilis pada 19 oktober 2011 membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan data dan kontrol terpadu kontak

jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan NFC.

2.2.5. *JavaScipt Object Notation* (JSON)

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 [5]. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari 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* [5].
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*) [5].

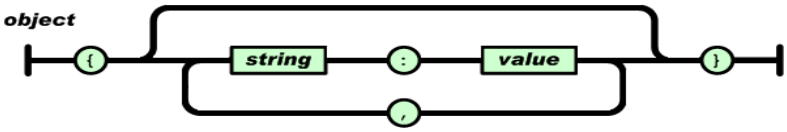
Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini[5].

2.2.5.1.Bentuk JSON

JSON menggunakan bentuk sebagai berikut:

1. Objek

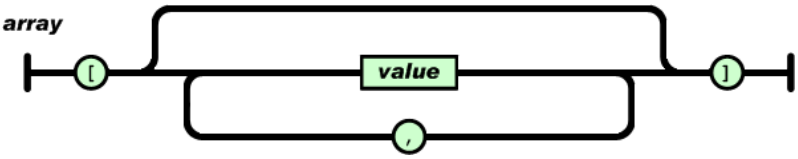
Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma) [5].



Gambar 2.2 Objek JSON [6]

2. Larik

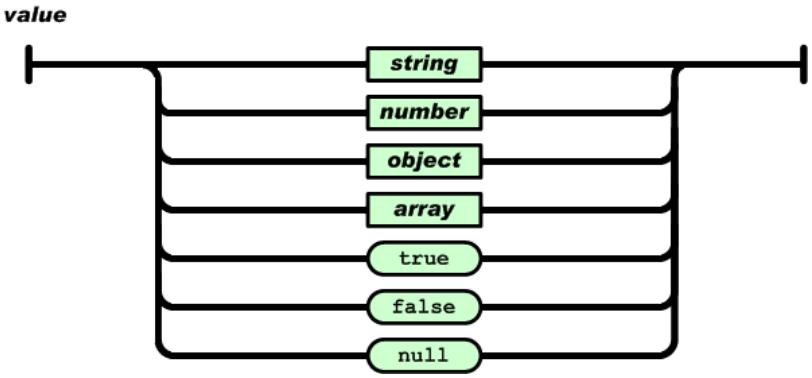
Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma) [5].



Gambar 2.3 Larik JSON [6]

3. Value

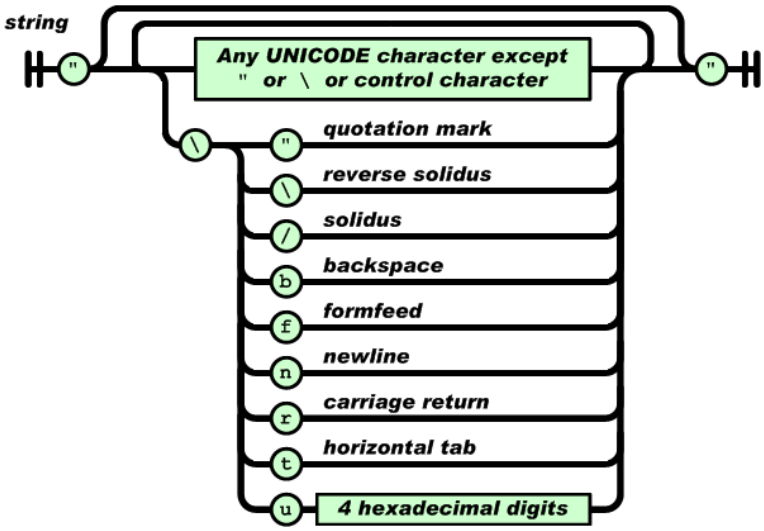
Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat [5].



Gambar 2.4 Value JSON [6]

4. String

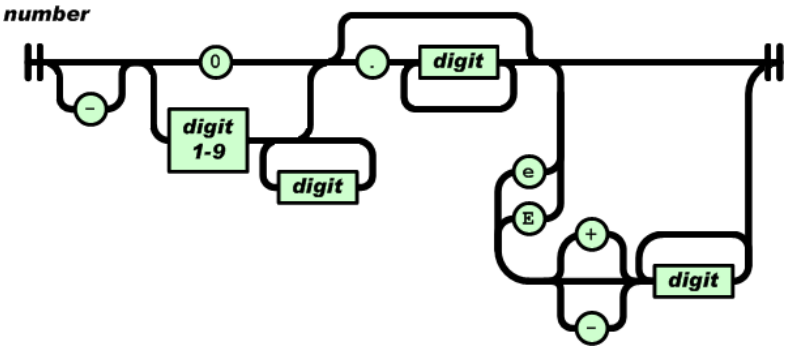
String adalah kumpulan dari nol atau lebih karakter *Unicode,* yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escapes* "\" untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada *string. String* sangat mirip dengan *string C atau Java* [5].



Gambar 2.5 *String* JSON [6]

5. Angka

Angka adalah sangat mirip dengan angka di C atau *Java*, kecuali format oktal dan heksadesimal tidak digunakan [5].



Gambar 2.6 Angka Dalam JSON [6]

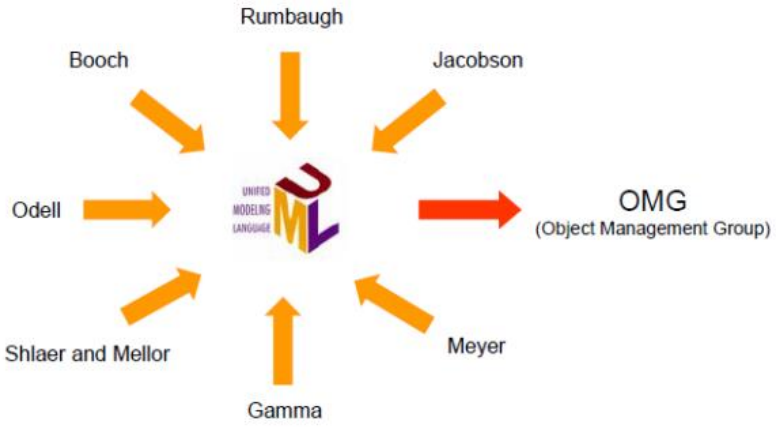
Dalam aplikasi yang di bagun pemanfaat fitur JSON sebagai format pertukaran data. Sebagai contoh, *twitter* menyediakan API untuk digunakan pada situs pihak ketiga atau aplikasi pihak ketiga misalnya *twitter client*. Data yang dikirimkan dari hasil request berupa data berformat JSON.

2.2.6. *Unified Modeling Language* (UML)

Pada bagian ini akan dijelaskan tentang pengertian UML ,*Use Case Diagram*, *Class Diagram, Sequence Diagram*.

2.2.6.1. Pengertian UML

Unified Modeling Language (UML) adalah termasuk kedalam rumpun jenis permodelan notasi grafis yang didukung oleh model – model tunggal. Permodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object – oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan – perusahaan konsorsium terbuka. UML merupakan suatu bahasa permodelan yang terdiri banyak model [7].



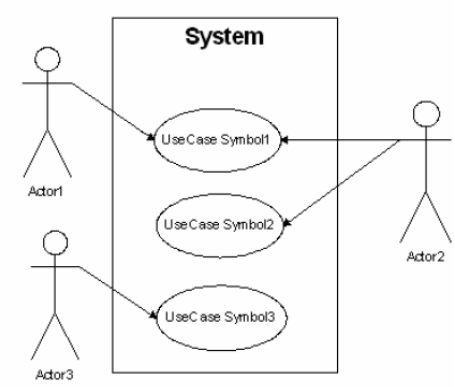
Gambar 2.7 *Unified Modeling Language*(UML)[7]

Berikut ini penjelasan singkat dari pembagian kategori tersebut .

1. *Structure* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan di dalam model UML.
2. *Behavior* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction* diagram yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub-sistem pada suatu sistem.

2.2.6.2. *Use Case Diagram*

Use case diagram merupakan suatu permodelan yang menitik beratkan pada fungsionalitas utama dalam suatu konteks sistem. *Use case diagram* juga menekankan tingkah laku fungsional utama dalam sistem berinteraksi dengan objek diluar sistem tersebut. Selain itu, *use case diagram* juga telah menitik beratkan jenis hubungan diantara fungsi utama. Adapun komponen komponen dalam *use case diagram* antaranya [8]:



Gambar 2.8 *Use Case diagram*

1. Aktor

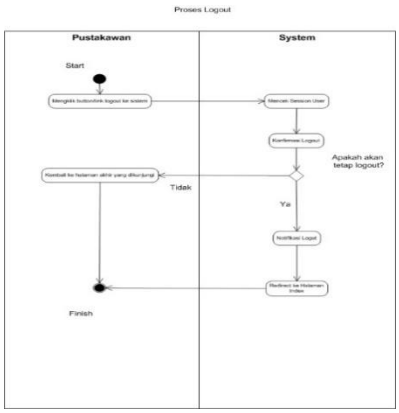
Aktor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. Aktor berbeda diluar sistem namun berkaitan erat dengan fungsionalitas didalamnya. *Acto*r dapat memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua fungsionalitas utama. Actor juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. *Actor* biasanya dapat berupa pengguna atau database yang secara pandang berada dalam suatu ruang lingkup sistem tersebut [8].

2. *Use Case*

Use case merupakan gambaran umum dari fungsi proses utama yang menggambarkan tentang salah satu prilaku sistem. Prilaku sustem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada *use case*, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. *Use case* menitik beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun diluar sistem [8].

Pemanfaat *Use case* diagram untuk memodelkan atau menjelaskan aplikasi yang akan di bangun, apa saja yang terlibat dan bagaimana alur dari aplikasi yang akan di bangun.

2.2.6.3. *Activity Diagram*



Gambar 2.9 *Activity Diagram*

Activity diagram atau diagram aktivitas adalah sebuah diagram yang dapat digunakan untuk menggambarkan secara grafis aliran proses bisnis, langkah langkah sebuah *use case* atau logika *behavior object*. Diagram ini serupa dengan *flowchart*, dimana secara grafis diagram ini menggambarkan aliran sekuential dengan *flowchart* dimana diagram ini menggambarkan mekanisme kegiatan paralel.

Oleh karena itu, diagram ini sangat berguna untuk memodelkan kegiatan yang akan dilakukan saat sebuah operasi dieksekusi dan untuk memodelkan hasil – hasil dari kegiatan ini, seperti memodelkan *event* yang menyebabkan *window* akan ditampilkan atau ditutup.

Dibawah ini merupakan simbol-simbol yang ada pada diagram aktivitas:

1. *Initial node*

   Lingkaran yang diisi penuh merupakan awal mulainya dari diagram ini.Initial ode tidak harus ada, tetapi dengan mengunakanya membuat diagramnya lebih mudah untuk dibaca.

2. *Activity*

   Segi empat bersudut tumpul menggambarkan kegiatan yang perlu dilakukan. Secara *activity* bisa secara fisik, seperti *inspect Forms* atau secara elektronik, seperti *Display Create Student Screen*.

3. *Control Flow*

   Panah yang ada di diagram, menggambarkan sasaran yang mengawa kegiatan.

4. *Fork*

   Bar hitam dengan satu flow yang ke dalamnya dan beberapa *flow* lainnya meninggalkannya, merupakan bar sinkronissi dimana kegiatan dapat dilakukan secara paralel.

5. *Join*

   Bar hitam dengan satu *flow* mengarah ke dalamnya dan ada satu *flow* yang meninggalkannya, merupakan bar sinkronisasi dimana beberapa aktivitas yang mempunyai tujuan yang sama dari keduannya digabung menjadi satu.

6. *Decicion*

   Gambar sebuah wajik, menggambarkan sebuah kegiatan keputusan.

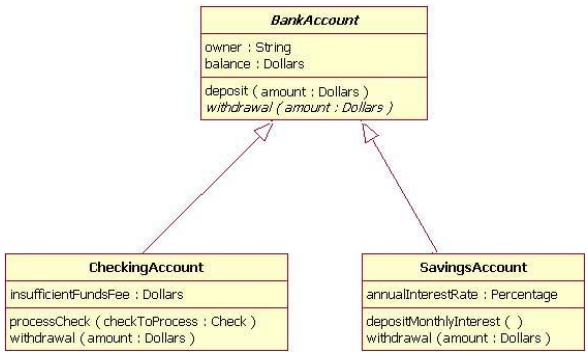
7. *Final Node*

   Gambar lingkaran yang diisi penuh dan ada batas lagi di luarnya, menggambarkan akhir dari sebuah proses.

2.2.6.4.*Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class* diagram menunjukan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi property dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis - jenis hubungan seperti *asosiatif, dependensi, agregasi, komposisi, spesifikasi* dan *generalisasi.* Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas. Masing – masing komponen penyusun kelas memiliki hak akses seperti *public, private dan protected*.

Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun Aplikasi Desa Siaga di Desa Dayeuhkolot Berbasis Android. *Class Diagram* Desa siaga Bencana di desa Dayeuhkolot berbasis Android menunjukkan struktur *class* satu yang berelasi dengan *class* lainnya.

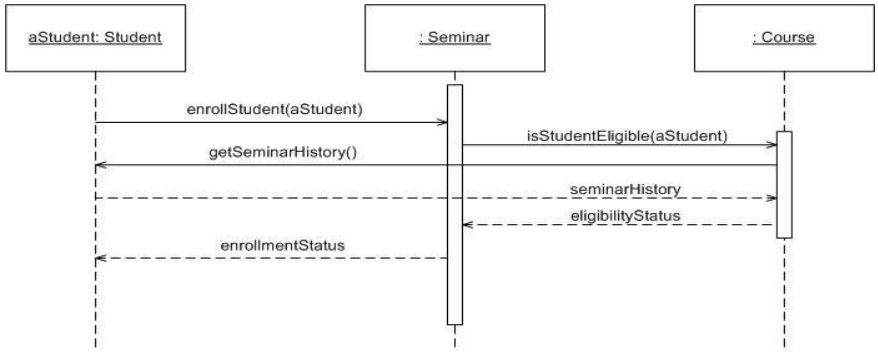


Gambar 2.10 ***Class diagram***

2.2.6.5. *Squence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case* : interaksi terjadi antar *class, operasi* apa saja yang terlibat, urutan antar *operasi* dan informasi yang diperlukan oleh masing - masing operasi [8].

Squence Diagram menggambarkan langkah-langkah proses yang terjadi pada aplikasi secara tersusun diantara lain: proses registrasi, proses *login*, proses lupa *password,* dan lain lain.



Gambar 2.11 *Sequence Diagram*

2.2.7. *Entity-Relationship Diagram* (ERD)

Entity-Relationship Diagram adalah diagram grafikal keseluruan struktur logikadari sebuah basis data. ERD tidak menggambarkan aliran data atau proses data, tetapi ERD lebih menggambarkan data pada data *store*.

ERD ini berfungsi untuk menggambarkan relasi dari dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi, yaitu relasi satu ke satu, satu kebanyak dan relasi banyak ke banyak.

Model ERD yang berisi komponen-komponen himpunan *entitas* dan himpunan *relasi* yang masing-masing dilengkapi dengan *atribut-atribut* yang mempresentasikan seluruh fakta yang ditinijau.

Merupakan model jaringan data yang menekankan pada struktur dan hubungan antardata di aplikasi Desa Siaga Bencana di Dayeuhkolot.

2.2.8. *Java*

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh *James Gosling* saat masih bergabung di *Sun Microsystems* saat ini merupakan bagian dari *Oracle* dan dirilis tahun

1995. Bahasa ini banyak mengadopsi *sintaksis* yang terdapat pada C dan C++ namun dengan *sintaksis* model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *p-code (bytecode*) dan dapat dijalankan pada berbagi *Java Vitrual Machine* (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non- spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Karena fungsionalitasnya yang memungkinkan aplikasi *Java* mampu berjalan di beberapa *platform* sistem operasi yang berbeda, *Java* dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini *Java* merupakan bahasa pemrograman yang paling popular digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web.*

Java merupakan bahasa berorientasi objek(OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program computer sebagai kelompok objek yang saling berinteraksi. Deskripsi OOP secara ringkas adalah mengorganisasikan program sebagai kumpulan komponen yang disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagain modul sekaligus tipe. Sebagai tipe maka pada saat jalan program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi objek yang merupakan instan kelas. Kelas dapat mewarisi kelas lain. Java tidak mengijinkan pewaris jamak namun menyelesaikan kebutuhan pewaris jamak.

2.2.9. Definisi *XAMPP*

XAMPP merupakan akronim dari *Windows, Apache, MySQL,* dan *PHP*. *WAMP* digunakan untuk membangun sebuah host lokal untuk membuat simulasi

sebuah *website* yang memiliki *server side script* sebelum dipublikasikan pada *server* yang sebenarnya. Penjelasan dari komponen dalam *XAMPP* adalah sebagai berikut :

1. **Apache**

Server HTTP Apache atau *Server Web* / *WWW Apache* adalah *server web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows dan Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas *web / www* ini menggunakan *HTTP*.

2. ***Personal Home Page* (PHP)**

PHP merupakan bahasa *scripting* yang bersifat *Server Side,* artinya kode – kode PHP akan dieksekusi oleh *web server* kemudian hasil dari eksekusi akan dikirimkan ke *browser client* berupa kode HTML. Cara kerja yang seperti ini mengakibatkan *client* tidak dapat melihat kode PHP dan hanya bisa melihat kode HTML yang telah diciptakan oleh mesin PHP yang berada pada *web server*. Dengan kata lain PHP memungkinkan untuk membuat halaman *web* dinamis yang aman.

PHP dapat berjalan dengan baik di banyak *Sistem Operasi* seperti *Windows, Linux, MacOS, dll.* PHP juga kompatibel untuk berbagai jenis database misalnya *MySQL, PostgreSQL, Oracle, Microsoft access, Interbase, dll.*

Bahasa pomrograman yang di gunakan dalam aplikasi Desa Siaga Bencana di Desa Dayeuhkolot Berbasis Android yang akan dibangun adalah *java.*

2.2.10. Android Studio

Android studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform* Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk *Manajer Google, Ellie Powers*. Android studio bersifat free dibawah *Apache License 2.0.* Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada

bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan JetBrainns' IntelliJ IDEA, Studio di desain khusus untuk *Android Development*. Ini sudah bisa di *download* untuk Windows, Mac OS X, dan Linux.

Berikut ini beberapa persyaratan system untuk menggunakan Android Studio :

1. Windows
   - Microsoft Windows 8/7/Vista/2003 (32 atau 64 bit)
   - RAM minimum 2 GB, Rekomendasi 4 GB RAM
   - Disk kosong 500 MB
   - Sedikitnya 16 GB untuk Android SDK, Emulator system images, and caches.
   - Java Development Kit (JDK) 7 atau yang lebih tinggi
   - Resolusi minimum 1280 x 800
2. OS X
   - Mac OS X 10.8.5 atau yang lebih tingggi, up to 10.10 to up 10.10.1 up 10.10.2 (Yosemite)
   - RAM minimum 2 GB, Rekomendasi RAM 4 GB
   - Disk kosong 500 MB
   - Sedikitnya 16 GB untuk Android SDK, emulator system images, dan caches
   - Java Development Kit (JDK) 7 atau yang lebih tinggi
   - Resolusi minimum 1280x 800

   Dalam aplikasi Desa Siaga Bencana di Desa Dayeuhkolot Berbasi Android yang akan dibagun menggunakan menggunakan Android Studio.

2.2.11. *Application Programming Interface* (API)

Application Programming Interface (API) merupakan fungsi-fungsi/perintah-perintah untuk mengganti bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstuktur dan mudah dimengerti oleh *programmer*. Fungsi yang dibuat dengan menggunakan *API* tersebut kemudian akan menggail *system calls* sesuai

dengan *system* oprasinya. Tidak tertutup kemungkinan nama dari *system calls* sama dengan nama di *API.*

Keuntungan membuat *program* dengan menggunakan *API* adalah :

1. **Portabilitas**. *Programmer* yang menggunakan *API* dapat menjalankan *programnya* dalam sistem oprasi mana saja asalkan sudah *ter-install API* tersebut. Sedangkan *system call* berbeda antar sistem oprasi, dengan catatandalam implementasinya mungkin saja berbeda.
2. **Lebih Mudah Dimengerti**. *API* menggunakan bahasa yang lebih terstuktur dan mudah dimengerti dari pada bahasa *system call*. Hal ini sangat dalam membantu programmer menggunakan *API.*

System calls interface berfungsi penghubung *API* dan *system call* yang di mengerti oleh sistem oprasi. *System call interface* akan menerjemahkan perintah dalam API dan kemudian akan memanggil *system call* yang diperlukan. Untuk membukan suatu *file* tersebut, *user* menggunakan program yang telah dibuat dengan menggunakan bantuan *API*, maka perintah dari *user* tersebut, diterjemahkan oleh program menjadi perintah *open().* Perintah *open()* merupakan perintah dari *API* dan bukan yang langsung dimengerti oleh kernel sistem oprasi. Oleh karena itu, agar keinginan *user* dapet dimengerti oleh sistem oprasi, makan perintah *open()* tadi diterjemahkan kedalam bentun *system call* oleh *system call interface*. Implementasi perintah *open*() bisa bermacam-macam, tergantung dari sistem yang di gunakan.

2.2.12. *Twitter API*

Pada Awalnya perusahan Summize yang menyediakan fasilitas mencari data di *Twitter.* Kemudian perusahaan Summize ini diakuisisi dan diganti menjadi merek menjadi *Twitter Search API* terpisah sebagai entitas sendiri.

API Twitter terdiri dari 3(tiga) bagaian yaitu :

1. ***Search API***. *Search API* dirancang untuk memudahkan *user* dalam mengelola *query search* dikonten *Twitter. User* dapat menggunakan untuk mencari *tweet* berdasarkan *keyword* khusus atau mencari *tweet* lebih spesifik berdasarkan username *Twitter*. *Search API* juga menyediakan akses pada data *Trending Topic*.
2. ***Representastional State Transfer (REST) API***. REST API memperbolehkan *developer* untuk mengakses inti dari *Twitter* seperti *timeline, status update* dan nformasi *user*. *REST API* digunakan dalam hal membangun sebuah aplikasi *Twitter* yag kompleks yang memerlukan inti dari *Twitter*.
3. ***Steaming API.*** *Streaming* API digunakan developer untuk kebutuhan yang lebih intensif seperti melakukan penelitian dan analistik data. *Streaming* API dapat menghasilkan aplikasi yang dapat mengetahui statistik status *update, follwer* dan lain sebagainya. Dalam penelitian ini, bagian *Twitter API* yang di gunakan adalah *REST API.*

Pemanfaatan fitur API Twitter pada aplikasi yang akan dibagun ini yaitu untuk mengambil tweet dari twitter yang akan di informasikan di aplikasi sebagai acuan pemberitahuan hujan dan banjir di berbagai daerah kawasan Bandung.

2.2.13. *Location Based Service* (LBS)

Location Based Service (LBS) atau layanan berbasis lokasi adalah sebuah layanan informasi yang dapat diakses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut. *Location Based Service* dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu, seperti menemukan lokasi tempat yang diinginkan terdekat atau lokasi lainnya.

Hal paling penting dari *Location Based Service* (LBS) dapat bekerja sesuai yang diinginkan oleh pengembang aplikasi Android. Android pun memungkinkan

pengembang menentukan metode pencarian lokasi yang dibutuhkan dan juga dapat mengatur kebutuhan daya, biaya dan akurasi berdasarkan *spesifik* yang akan dibuat untuk aplikasi tersebut.

2.2.13.1. Unsur-unsur *Location Based Service*

Berikut adalah unsur utama dari *Location Based Service*.

1. *Location Manager* (API Maps): Menyediakan perangkat bagi sumber atau *source* untuk *Location Based Service* (LBS), *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada "com.google.android.maps;".
2. *Location Providers* (*API Location*): Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API *Location* berhubungan dengan data GPS (*Global Positioning Sistem*) dan data lokasi *real-time*. *API Location* berada pada paket Android yaitu dalam paket *"android.location*". Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditentukan melalui *Location Manager*.

2.2.13.2. Komponen *Location Based Service*

Terdapat lima komponen pendukung utama dalam teknologi *Location Based Service*, antara lain:

1. *Piranti Mobile*, adalah salah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu (*tool)* bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Piranti mobile yang dapat digunakan bisa berupa *smartphone, laptop.* Selain itu, *piranti mobile* dapat juga berfungsi sebagai alat *navigasi* di kendaraan seperti halnya alat *navigas*i berbasis *Global Positioning Sistem* (GPS).
2. Jaringan Komunikasi, Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti *mobile-nya*

untuk kemudian dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Komponen *Positioning* (Penunjuk Posisi/Lokasi), Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi mobile atau juga menggunakan *Global Positioning Sistem* (GPS).
4. Penyedia layanan dan aplikasi, merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bias tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di *Yellow Pages* sesuai dengan permintaan, dan masih banyak lagi yang lainnya.
5. Penyedia data dan konten, Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh *basis* data *geografis* dan *lokasi* bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya. Secara lengkap kelima komponen pendukung LBS tersebut dapat dilihat pada gambar berikut.

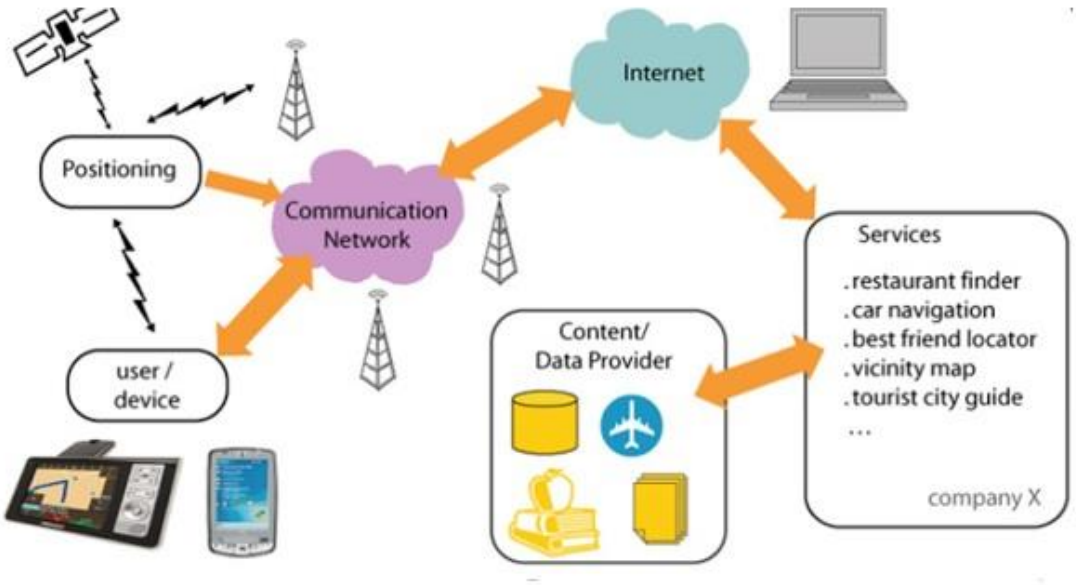
2.2.13.3. Cara Kerja *Location Based Service*

Untuk menggambarkan cara kerja *Location Based Service* adalah sebagai beriktu :

1. Fungsi pencarian telah *diaktifkan*, posisi pengguna sebenarnya dari perangkat mobile diperoleh dari *Global Positioning Service*. Hal ini dapat dilakukan baik oleh

perangkat menggunakan *Global Positioning Sistem* (GPS) sendiri atau layanan posisi jaringan yang berasal dari *provider (Cell Tower*). Setelah itu perangkat *mobile* pengguna mengirimkan permintaan informasi, yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke *gateway telekomunikasi.*

2. *Gateway* memiliki tugas untuk bertukar pesan di antara *jaringan komunikasi selular* dan *internet.* Oleh karena itu dia mengetahui alamat *web* dari beberapa aplikasi *server* dan *rute* permintaan ke *spesifik server* tertentu. *Gateway* akan menyimpan juga informasi tentang perangkat *mobile* yang telah meminta informasi
3. *Aplikasi server* membaca permintaan dan *mengaktifkan* layanan yang terkait.
4. Kemudian, *service* menganalisis lagi pesan dan memutuskan mana informasi dan posisi pengguna diperlukan untuk menjawab permintaan pengguna.
5. Selanjutnya *service* akan menemukan informasi lokasi yang diinginkan.
6. Setelah sekarang semua informasi *service* akan melakukan *buffer spasial* dan *query routing* untuk mendapatkan beberapa lokasi terdekat. Setelah itu hasil dikirim kembali ke pengguna melalui *internet*, *gateway* dan *jaringan mobile.*
7. Kemudian, informasi mengenai lokasi akan disampaikan kepada pengguna baik dalam bentuk peta *digital.*



Gambar 2.12 Cara Kerja *Location Based Service*[2]

2.2.14. *Global Positioning System* (GPS)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi *satelit* yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 *satelit* yang mengirimkan sinyal gelombang *mikro* ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, dimana GPS receiver ini akan mengumpulkan informasi dari satelit GPS, seperti :

1. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
2. Lokasi. GPS memberikan informasi dalam tiga dimensi :
    1. *Latitude*
    2. *Longitude*
    3. *Elevasi*
3. Kecepatan ketika berpindah tempat, GPS dapat menunjukan informasi kecepatan berpindah tersebut.

4. Arah perjalanan. GPS dapat menunjukan arah tujuan.
5. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver.*
6. Komulasi data. GPS *receiver* dapat menyimpan informasi *track,* seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan dan sebagainya.

Pemantafaan fitur GPS pada aplikasi ini untuk memberi informasi lokasi pelapor maupun petugas yang kan mendatangi pelapor, fitur GPS ini sangat penting dalam aplikasi ini karena memberikan informasi secara *real* yang meminimalkan terjadinya kesalahan pemberitahuan lokasi pelapor kepada petugas.

2.2.15. *Google Cloud Messaging* (GCM) Android

Google Cloud Messaging (GCM) adalah sebuah layanan pada Android yang berfungsi untuk menjalankan sebuah *Push Notification*. Di Android layanan GCM dijalankan oleh sebuah aplikasi yang berjalan di latar belakang dan bernama Layanan *Google Play*. Layanan ini akan membangunkan aplikasi yang bersangkutan saat menerima notifikasi baru dan akan menampilkan notifikasi tersebut pada panel notifikasi Android.

Aplikasi Android yang menggunakan GCM tidak perlu berjalan di belakang layar untuk menerima sebuah notifikasi. GCM (dalam hal ini Layanan *Google Play*) akan membangunakan aplikasi tersebut dengan mekanisme *Broadcoast Intent (Event* yang terjadi saat ada notifikasi baru) ketika ada pesan tiba selama aplikasi tersebut diatur untuk diizinkan menerima Pesan Siaran. Namun tidak sedikit pula aplikasi berjalan di latar belakang yang menggunakan *GCM Service.*

GCM ini “100 persen” menggunakan *server Google* dan Android akan memeriksa ke *server* tersebut setiap beberapa menit (tergantung pengembang *rom)*. Setiap pesan tersebut mempunyai batasan 1024 byte atau hanya 1 kb dan Google membatasi jumlah pesan secara keseluruhan dan jumlah pesan ke perangkat tertentu.

Fitur GCM ini berfungsi memberikan notifikasi pada *smartphone* pengguna aplikasi, yang bertujuan memberitahukan bahwa ada laporan yang masuk pada aplikasi tersebut.

2.2.16. MySQL

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *basisdata* yang telah ada sebelumnya; *Structured Query Language* (SQL). SQL adalah sebuah konsep *pengoperasian basisdata*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan *pengoperasian* data dikerjakan dengan mudah secara *otomatis*.

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja *pengoptimasi-nya* dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun *program-program aplikasi* yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung *operasi basisdata transaksional* maupun *operasi basisdata non-transaksional*. Pada modus *operasi non-transaksional*, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus *basisdata transaksional*, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus *non-transaksional*. Adapun beberapa keistimewaan dari MySQL sehingga membuat MySQL dipilih oleh kebanyakan orang :

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai *sistem operasi* seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga,* dan masih banyak lagi.

2. Perangkat lunak sumber terbuka (open source)

MySQL didistribusikan sebagai open source sehingga dapat digunakan secara gratis.

3. Multi-user

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data

MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp,* dan lain-lain.

6. Perintah dan Fungsi

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select d*an *Where* dalam perintah (*query*).

7. Keamanan

MySQL memiliki beberapa lapisan keamanan seperti *password* yang *terenkripsi.*

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah *record* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan *protokol TCP/IP, Unix soket (UNIX), atau named pipes (NT).*

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Antar Muka

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *Application Programming Interface* (API).

12. Klien dan Peralatan

MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk *administrasi basis data*, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur *table*

MySQL memiliki *struktur tabel* yang lebih *fleksibel* dalam menangani *ALTER TABLE,* dibandingkan *basis data* lainnya semacam *PostgreSQL* ataupun *Oracle*.

Database yang digunakan dalam membangun aplikasi Desa Siaga Bencana Di Desa Dayeuhkolot Berbasis Android yaitu menggunakan MySQL

2.2.17. Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)

BMKG mempunyai status sebuah Lembaga Pemerintah Non Departemen (LPND), dipimpin oleh seorang Kepala Badan.BMKG mempunyai tugas : melaksanakan tugas pemerintahan di bidang Meteorologi, Klimatologi, Kualitas Udara dan Geofisika sesuai dengan ketentuan perundang-undangan yang berlaku.

Dalam melaksanakan tugas sebagaimana dimaksud diatas, Badan Meteorologi Klimatologi dan Geofisika menyelenggarakan fungsi :

1. Perumusan kebijakan nasional dan kebijakan umum di bidang meteorologi, klimatologi, dan geofisika.
2. Perumusan kebijakan teknis di bidang meteorologi, klimatologi, dan geofisika.
3. Koordinasi kebijakan, perencanaan dan program di bidang meteorologi, klimatologi, dan geofisika.
4. Pelaksanaan, pembinaan dan pengendalian observasi, dan pengolahan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
5. Pelayanan data dan informasi di bidang meteorologi, klimatologi, dan geofisika.
6. Penyampaian informasi kepada instansi dan pihak terkait serta masyarakat berkenaan dengan perubahan iklim.
7. Penyampaian informasi dan peringatan dini kepada instansi dan pihak terkait serta masyarakat berkenaan dengan bencana karena factor meteorologi, klimatologi, dan geofisika.
8. Pelaksanaan kerja sama internasional di bidang meteorologi, klimatologi, dan geofisika.
9. Pelaksanaan penelitian, pengkajian, dan pengembangan di bidang meteorologi, klimatologi, dan geofisika.
10. Pelaksanaan, pembinaan, dan pengendalian instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
11. Koordinasi dan kerja sama instrumentasi, kalibrasi, dan jaringan komunikasi di bidang meteorologi, klimatologi, dan geofisika.
12. Pelaksanaan pendidikan dan pelatihan keahlian dan manajemen pemerintahan di bidang meteorologi, klimatologi, dan geofisika.
13. Pelaksanaan pendidikan profesional di bidang meteorologi, klimatologi, dan geofisika.
14. Pelaksanaan manajemen data di bidang meteorologi, klimatologi, dan geofisika.

15. Pembinaan dan koordinasi pelaksanaan tugas administrasi di lingkungan BMKG.
16. Pengelolaan barang milik/kekayaan negara yang menjadi tanggung jawab BMKG.
17. Pengawasan atas pelaksanaan tugas di lingkungan BMKG.
18. Penyampaian laporan, saran, dan pertimbangan di bidang meteorologi, klimatologi, dan geofisika.

Dalam melaksanakan tugas dan fungsinya BMKG dikoordinasikan oleh Menteri yang bertanggung jawab di bidang perhubungan[8].

Fitur BMKG dalam aplikasi ini berfungsi untuk mengambil data dari web BMKG yang memberitahukan mengenai informasi cuaca, kelembaban, secara real-time di dalam aplikasi Desa siaga bencana di Desa Dayeuhkolot.