BAB 2
TINJAUAN PUSTAKA

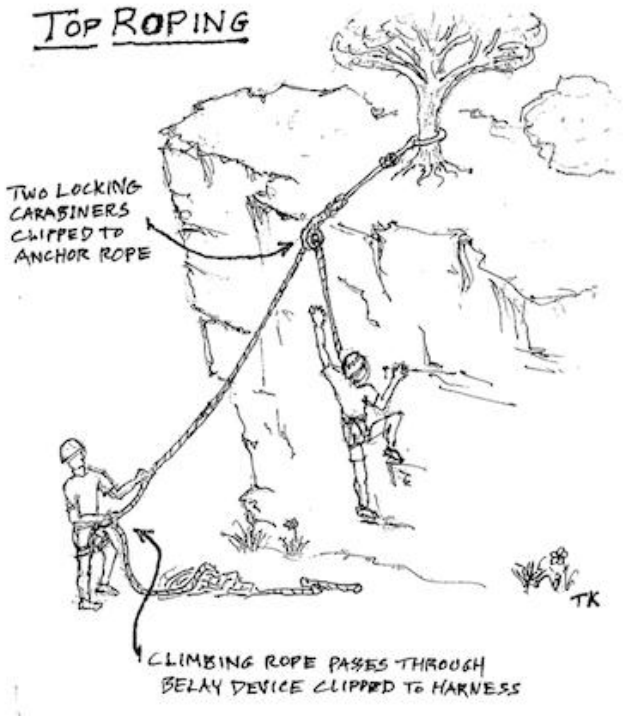
2.1 Tinjauan Pustaka

Tinjauan pustaka menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang akan dibangun. Tinjauan pustaka yang digunakan dalam penyusunan aplikasi rekomendasi olahraga panjat tebing di bandung, meliputi pengertian panjat tebing, android, TOPSIS, GPS, LBS, API, JSON, Google Maps, Openweather, UML, black box dan skala *likert*.

2.1.1 Panjat Tebing

Pada umumnya panjat tebing dilakukan pada daerah dengan kemiringan mencapai lebih dari 45º dan memiliki tingkat kesulitan tertentu [1]. Panjat tebing yang membutuhkan kemampuan fisik untuk dapat memanjat lebih tinggi, kemampuan teknik untuk menempatkan kaki dan tangan pada permukaan dinding, kemampuan untuk mengatur strategi dan menentukan jalur dan kemampuan berfikir untuk mengambil keputusan yang cepat, guna mencapai tempat yang lebih tinggi.

Pada olahraga panjat tebing terdapat beberapa jenis jenis pemanjatan yaitu *top-rope climbing*, *lead climbing* (rintisan) dan *bouldering*. *Top-rope climbing* merupakan pemanjatan yang dilakukan dimana tali sudah dikaitkan diakhir titik pemanjatan agar pemanjat sudah berada dalam posisi aman, jadi apabila pemanjat terjatuh atau terpeleset, tali yang telah dikaitkan ke puncak pemanjatan otomatis menjadi pengaman utamanya. Maka dapat dikatakan *top-rope* merupakan suatu cara kebanyakan orang pemanjat pemula untuk mempelajari atau mencoba kegiatan panjat tebing [10]. Gambaran pemanjatan *top-rope* dapat dilihat seperti pada gambar 2.1 berikut:



Sumber Gambar:
http://www.alpineinstitute.com/articles/expert-tips/rock-climbing-terms-styles-and-techniques/

Gambar 2. 1 Pemanjatan *Top-Rope*

2.1.1.1 Standar Keselamatan Dalam Pemanjatan *Top-Rope*

Standar keselamatan dalam dunia panjat tebing merupakan suatu pengetahuan umum bagi seorang pemanjat dalam melakukan olahraga ini, karena panjat tebing merupakan olahraga yang membutuhkan konsentrasi dan terukur dalam melaksanakannya.

Berikut ini merupakan hal-hal yang harus diperhatikan dalam memanjat *top-rope*, yakni sebagai berikut:

1. *Harness* yang dipakai sudah terpakai dengan benar, begitu juga ukuran *harness* yang digunakan sesuai dengan ukuran tubuh penggunanya.
2. Simpul delapan maupun simpul pengaman lainnya sudah terpasang dengan benar pada *harness* pemanjat.
3. Orang kedua *(belayer)* mampu, paham dan menguasai 5 langkah dalam mengamankan pemanjat.

4. Alat yang digunakan sudah teruji dan tersertifikasi UIAA (*Union Internationale des Associations d'Alpinishm)*.
5. Mampu menggunakan komunikasi pemanjatan secara umum.
6. Mengenal struktur tebing, pegangan maupun tumpuan.
7. Menggunakan pelindung kepala (Helm) dan sepatu khusus panjat tebing.
8. Lintasan tali sudah terpasang pada *anchor* (penambat) utama dengan benar dan aman, pastikan *anchor* (penambat) utama tidak hanya satu.
9. Orang kedua *(belayer)* berada dalam posisi aman.
10. Menjaga alat dan tidak teledor dalam penggunaannya.

2.1.2 Rekomendasi

Rekomendasi adalah saran yang sifatnya menganjurkan, membenarkan, atau menguatkan mengenai sesuatu atau seseorang. Rekomendasi sangat penting artinya untuk meyakinkan orang lain bahwa sesuatu atau seseorang tepat dan layak. Rekomendasi juga dapat dikatakan suatu bentuk komunikasi sekaligus promosi tidak langsung yang dilakukan oleh para konsumen yang sudah pernah membeli produk atau jasa yang kemudian menceritakan berbagai pengalamannya yang berkaitan dengan produk atau jasa tersebut kepada orang lain.

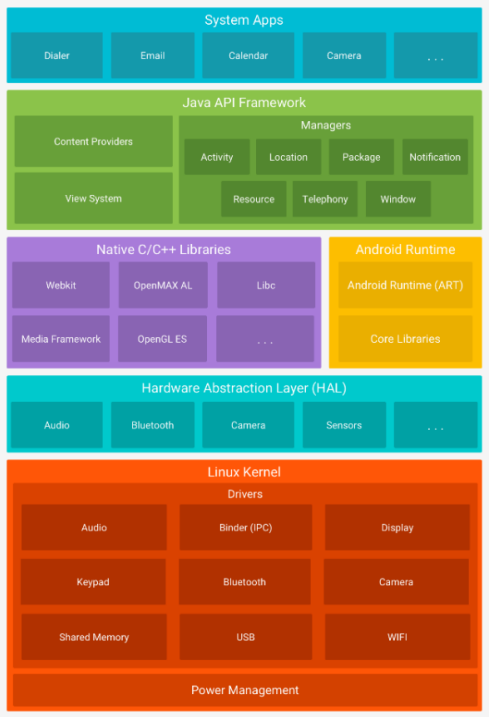
Misalkan ketika seseorang akan berencana mengunjungi suatu tempat wisata. Biasanya mereka akan mencari cerita perjalanan maupun testimoni dari orang-orang yang sudah pernah mengunjungi tempat tersebut sebelumnya, mencari tahu apakah objek yang menarik disana, bagaimana cara untuk kesana, dan berapa biaya yang sekiranya diperlukan untuk kesana. Jika banyak testimoni positif dan sesuai dengan keinginan yang dicari maka akan menambah keyakinan orang untuk pergi mengunjungi tempat tersebut.

2.1.3 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi [3]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Sistem operasi ini menjadi sangat populer di Indonesia. Dengan sistem

operasi android ini para pengembang dapat lebih mudah membuat aplikasi *mobile* yang baik dan terbarukan.

Sistem operasi android memiliki beberapa komponen utama yang disebut dengan arsitektur platform android. Berikut adalah Gambar 2.2 diagram komponen-komponen utama dari platform android :



Sumber Gambar :

https://developer.android.com/guide/platform/index.html?hl=id

Gambar 2. 2 Arsitektur Platform Android

Berikut adalah penjelasan dari setiap komponen utama arsitektur pada platform android pada Gambar 2.2 :

1. **Linux Kernel**

   Pondasi platform android adalah kernel Linux. Contohnya adalah *Android Runtime* (ART) yang bergantung pada kernel Linux untuk fungsionalitas dasar seperti *threading* dan manajemen memori tingkat rendah.

2. Hardware Abstraction Layer

Hardware Abstraction Layer (HAL) menyediakan antarmuka standar yang mengekspos kemampuan perangkat keras di perangkat ke kerangka kerja Java API yang lebih tinggi. HAL terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau *bluetooth*. Bila API kerangka kerja melakukan panggilan untuk mengakses perangkat keras, sistem android memuat modul pustaka untuk komponen perangkat keras tersebut.

3. Android Runtime

Perangkat android yang menjalankan Android versi 5.0 (API level 21) atau yang lebih tinggi, setiap aplikasi menjalankan proses masing-masing dengan tahap *Android Runtime* (ART). ART ditulis guna menjalankan beberapa mesin virtual pada perangkat bermemori rendah dengan mengeksekusi file *DEX*, format *bytecode* yang didesain khusus untuk android yang dioptimalkan untuk footprint memori minimal.

4. Native C/C++ Libraries

Native C/C++ Libraries adalah library yang mendukung komponen-komponen layanan sistem android yang dibuat dengan bahasa C dan C++. Library ini senantiasa mendukung kinerja layanan android.

5. Java API Framework

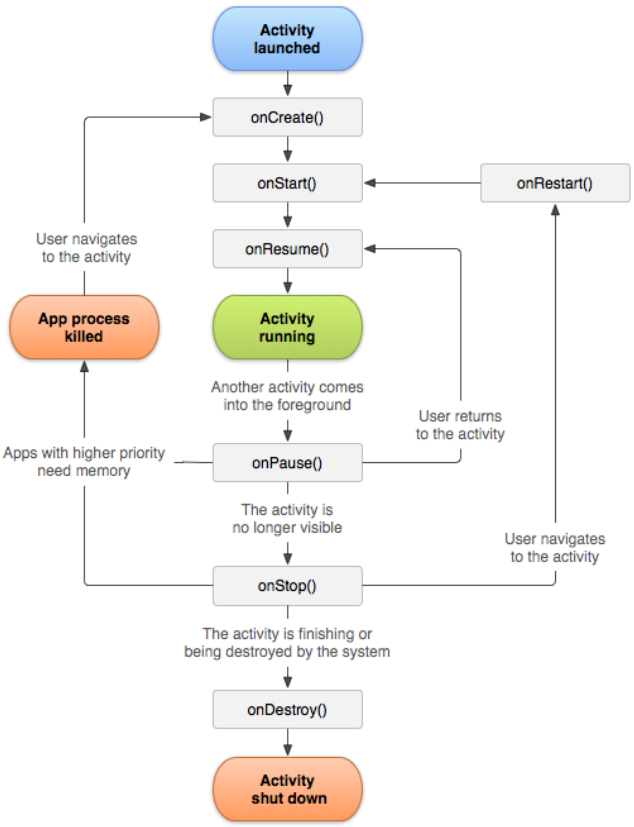
Keseluruhan rangkaian fitur pada android OS tersedia untuk melalui API yang ditulis dalam bahasa Java. API ini membentuk elemen dasar yang *developer* butuhkan untuk membuat aplikasi android dengan menyederhanakan penggunaan API .

6. System Apps

Android dilengkapi dengan serangkaian aplikasi inti untuk email, perpesanan SMS, kalender, menjelajahi internet, kontak, dll. Aplikasi yang disertakan bersama platform tidak memiliki status khusus pada aplikasi yang ingin dipasang pengguna.

Aplikasi sistem berfungsi sebagai aplikasi untuk pengguna dan memberikan kemampuan kunci yang dapat diakses oleh *developer* dari aplikasi mereka sendiri. Misalnya, jika aplikasi anda ingin mengirimkan pesan SMS, anda tidak perlu membangun fungsionalitas tersebut sendiri sebagai gantinya anda bisa menjalankan aplikasi SMS mana saja yang telah dipasang guna mengirimkan pesan kepada penerima yang anda tetapkan.

Di dalam sistem operasi android terdapat siklus hidup dimana pada siklus ini untuk menavigasi antara tahap *activity life-cycle*, android itu sendiri dengan menyediakan 6 *method* inti *callback* yaitu *onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy()*. Dalam siklus hidup *method callback*, kita dapat mendeklarasikan cara perilaku *activity* saat pengguna meninggalkan dan memasuki kembali ke sebuah *activity* itu. Disitulah peranan *method callback* dalam sebuah aplikasi pada *platform* android. Berikut adalah Gambar 2.3 *activity life-cycle* pada *platform* android:



Sumber Gambar : https://developer.android.com/guide/components/activities/activity-lifecycle.html

Gambar 2. 3 Siklus Hidup Android

Keenam *method callback* tersebut memiliki peranannya masing-masing dan dijelaskan secara rinci sebagai berikut :

1. **OnCreate()**

   Method ini adalah *method* utama dari setiap *activity*. *Method* ini akan dipanggil pertama kali ketika menjalankan sebuah sistem. Pengembang harus mengimplementasikan metode *onCreate()* untuk menjalankan logika memulai aplikasi dasar yang hanya boleh terjadi satu kali selama hidup aktivitas. Misalnya, implementasi *onCreate()* anda harus mendefinisikan antarmuka pengguna dan mungkin membuat instance beberapa variabel dalam cakupan kelas.

2. OnStart()

Method ini dipanggil ketika *method onCreate()* telah dipanggil. *onStart()* dipanggil ketika terlihat oleh *user*. *Method* ini selesai dengan cepat dan dilanjutkan dengan *method* setelahnya yaitu *onResume().*

3. OnResume()

Method ini dipanggil ketika *method onStart()* selesai dipanggil. *Method* ini adalah keadaan dimana pengguna berinteraksi dengan aplikasi. Aplikasi akan tetap dalam keadaan ini sampai terjadi suatu *statement* dari aplikasi semisal menerima panggilan telepon atau mematikan layar *smartphone.*

4. OnPause()

Method ini dipanggil ketika pengguna meninggalkan *activity* (meskipun tidak selalu berarti *activity* dihancurkan). *Method* ini berguna untuk menghentikan sementara operasi yang sedang berjalan semisal menjeda pemutaran musik dan lain-lain.

5. OnStop()

Method ini dipanggil ketika *activity* tidak terlihat lagi oleh pengguna, dengan kata lain *activity* berhenti dijalankan. Hal ini dapat terjadi semisal ada aktivitas baru dijalankan meliputi seluruh layar. Sistem juga dapat menghubungi *method* ini ketika *activity* selesai berjalan, dan akan segera dihentikan.

6. On Destroy()

Method ini adalah *method callback* ketika *activity* telah selesai dijalankan dan kemudian memanggil *method finish()* atau karena sistem untuk sementara menghancurkan proses yang berisi *activity* tersebut untuk menghemat ruang memori.

2.1.4 Metode TOPSIS

Technique For Others Reference by Similarity to Ideal Solution atau TOPSIS merupakan salah satu sistem pendukung keputusan multikriteria. Metode TOPSIS memiliki keuntungan yaitu merupakan salah satu metode yang *simple* dan konsep rasional yang mudah dipahami, dan mampu mengukur kinerja relatif dalam membentuk form matematika sederhana. Metode TOPSIS memiliki langkah-langkah sebagai berikut [11]:

1. Membuat matriks keputusan yang ternormalisasi.

   Keterangan:

   i = 1,2,...,m

   j = 1,2,...,n

   $$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}$$ *(Langkah 1)*

2. Membuat matriks keputusan yang ternormalisasi terbobot.

   Keterangan:

   i = 1,2,...,m

   j = 1,2,...,n

   $$y_{ij} = w_i r_{ij}$$ *(Langkah 2)*

3. Menentukan matriks solusi ideal posifit dan matriks solusi ideal negatif.

   Keterangan:

   $$\boxed{\begin{aligned} A^+ &= (y_1^+, y_2^+, \dots, y_n^+); \\ A^- &= (y_1^-, y_2^-, \dots, y_n^-); \end{aligned}}$$ *(Langkah 3)*

   i = 1,2,...,m

   j = 1,2,...,n

   $$y_j^+ = \begin{cases} max_i y_{ij} \\ max_i y_{ij} \end{cases} \qquad y_j^- = \begin{cases} min_i y_{ij} \\ min_i y_{ij} \end{cases}$$

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan negatif.

   a. Jarak antara alternatif Ai dengan solusi ideal positif dengan rumus sebagai berikut :

      $$D_i^+ = \sqrt{\sum_{j=1}^{n} (y_i^+ - yij)^2};$$ *(Langkah 4)*

      i = 1,2,...,m

   b. Jarak antara alternatif Ai dengan solusi ideal negatif dengan rumus

      $$D_i^- = \sqrt{\sum_{j=1}^{n} (yij - y_i^-)^2};$$

sebagai berikut :

i = 1,2,...,m

5. Menentukan nilai preferensi untuk setiap alternatif.

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \qquad \textit{(Langkah 5)}$$

i = 1,2,...,m

6. Nilai Vi yang lebih besar menunjukan bahwa alternatif Ai lebih dipilih.

2.1.5 GPS

GPS adalah sistem navigasi berbasis satelit terdiri dari jaringan 24 satelit ditempatkan ke orbit oleh Departemen Pertahanan Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978. Layanan GPS dahulu hanya dipergunakan untuk keperluan militer namun mulai terbuka untuk publik. 24 satelit GPS tersebut berada sekitar 12.000 mil di atas bumi bergerak mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, memiliki baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari dan memiliki roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya [3].

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 [3]:

a. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
b. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi:
    1. Latitude

2. Longitude
3. Elevasi

c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
d. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
e. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
f. 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.

2.1.6 LBS

Location Based Service (LBS) merupakan layanan informasi yang dapat diakses menggunakan *mobile devices*, yang dilengkapi kemampuan untuk mengetahui keberadaan lokasi dari si pengguna perangkat dan kemampuan memberikan informasi mengenai layanan yang tersedia berdasarkan lokasi mereka pada saat itu. Menurut Schiller J, *Location Based Service* dapat didefinisikan sebagai "Layanan yang mengintegrasikan lokasi perangkat *mobile* atau posisi dengan informasi lain sehingga dapat memberikan nilai tambah bagi pengguna"[4]. Dua unsur utama dari LBS adalah :

1. *Location Manager* (API Maps): Menyediakan perangkat bagi sumber atau source untuk LBS. API (*Application Programming Interface*) Maps menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada "https://google.android.maps.com".
2. *Location Providers* (API *Location*): Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) 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* [5].

Dalam layanan berbasis lokasi terdapat lima komponen penting seperti terlihat pada Gambar 2.4

Sumber Gambar: http://supeeerblog.blogspot.com/2013/05/location-based-services-lbs.html

Gambar 2. 4 Komponen Dasar LBS

Setiap komponen dalam Gambar 2.8 mempunyai fungsi:

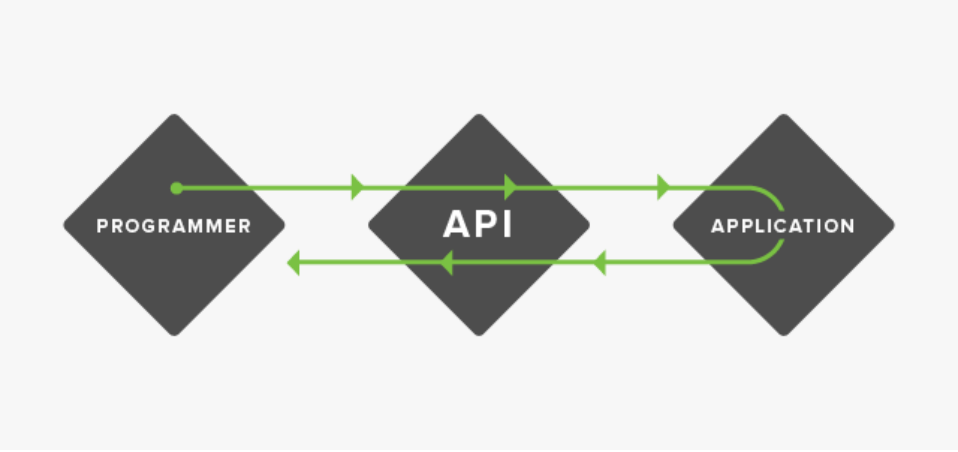
1. *Mobile devices*, merupakan suatu perangkat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar maupun teks.
2. *Communication network*, komponen ini mengirim data pengguna dan informasi yang diminta dari *mobile* terminal ke service provider kemudian mengirimkan kembali informasi yang diminta kepada pengguna. *Communication network* dapat berupa jaringan seluler (GSM, CDMA), *Wireless Local Area Network* (WLAN) atau *Wireless Wide Area Network* (WWAN).
3. *Positioning*, digunakan untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service and Content Provider*, penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.

2.1.7 API

Application Programming Interface (API) adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat *Hyper Text Transfer Protocol* (HTTP) dan mendapatkan respon berupa *Extensible Markup Language* (XML) atau *JavaScript Object Notation* (JSON). Pemanggilan fungsi ke suatu situs tertentu akan menghasilkan respon yang berbeda kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya [6].

Dengan API, panggilan-panggilan yang bolak-balik antar aplikasi diatur melalui *web service. Web service* adalah kumpulan standar teknis dan protokol, termasuk XML, bahasa umum yang digunakan oleh aplikasi-aplikasi tersebut selama berkomunikasi di internet. API dan *web service* sepenuhnya bekerja di belakang layar.

Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan *software*, agar dapat dimanfaatkan oleh pengembang pihak ketiga untuk mendesain aplikasi yang memanfaatkan layanan mereka dengan mudah.



Sumber Gambar : http://www.kursuswebsite.org

Gambar 2. 5 API

2.1.8 JSON

JavaScript Object Notation (JSON) adalah format penukaran data yang sederhana, bagi programer format ini mudah dibaca dan ditulis, sedangkan bagi mesin, format ini mudah untuk proses *parsing* dan *generate* [14]. JSON merupakan bagian dari JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON merupakan format teks bahasa pemrograman yang berdiri sendiri namun menggunakan konvensi standar yang biasa digunakan oleh para programer bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python, dan banyak lainnya. Hal ini menjadikan JSON sebagai bahasa penukaran data yang ideal [7]. JSON terbuat dari dua struktur yaitu:

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 terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

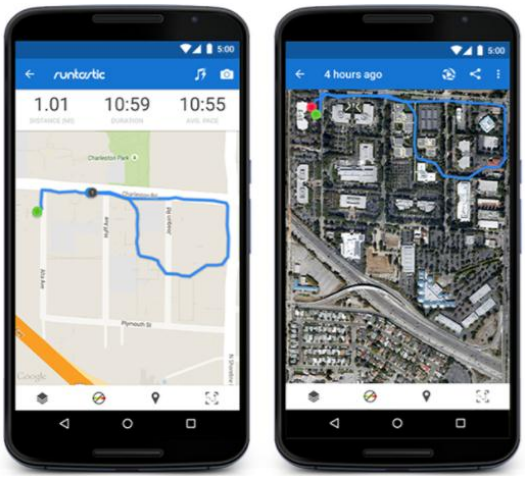
```
{
    "glossary": {
        "title": "example glossary",
                "GlossDiv": {
            "title": "S",
                        "GlossList": {
                "GlossEntry": {
                    "ID": "SGML",
                                        "SortAs": "SGML",
                                        "GlossTerm": "Standard Generalized Markup Language",
                                        "Acronym": "SGML",
                                        "Abbrev": "ISO 8879:1986",
                                        "GlossDef": {
                        "para": "A meta-markup language, used to create markup languages such as DocBook.",
                                                "GlossSeeAlso": ["GML", "XML"]
                    },
                                        "GlossSee": "markup"
                }
            }
        }
    }
}
```

Sumber Gambar : http://json.org/example.html

Gambar 2. 6 Contoh Data JSON

2.1.9 Google Maps API

Google maps API adalah layanan untuk menampilkan peta di aplikasi android. Pengembang dapat menambahkan peta ke aplikasi berdasarkan data di Google maps API secara otomatis menangani akses ke *server* Google maps, mengunduh data, menampilkan peta, dan merespon gerakan peta. Anda juga bisa menggunakan panggilan API untuk menambahkan *marker*, *poligon*, dan *overlay* ke peta dasar, serta mengubah tampilan area peta tertentu ke pengguna. Semua objek ini memberikan informasi tambahan tentang lokasi peta, dan memungkinkan interaksi pengguna dengan peta.



Sumber Gambar : https://developers.google.com/maps/?hl=id

Gambar 2. 7 GPS pada android

2.1.9.1 Cara Mendapatkan API key Google Maps

Berikut langkah-langkah mendapatkan API *key* Google maps :

1. Masuk ke halaman *https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&reusekey=true&hl=id&pli=1*
2. Buat atau pilih sebuah proyek.
3. Klik **Continue** untuk mengaktifkan Google maps API.
4. Pada laman **Credentials**, dapatkan **kunci API**.

Catatan: Jika anda sudah memiliki kunci API dengan pembatasan android, anda boleh menggunakan kunci itu.

5. Dari dialog yang menampilkan kunci API, pilih **Restrict key** untuk menyetel pembatasan android atas kunci API.

6. Di bagian **Restrictions**, pilih **Android apps**, kemudian masukkan sidik jari SHA-1 dan nama paket aplikasi anda. Misalnya:

```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1
:66:6E:44:5D:75
```

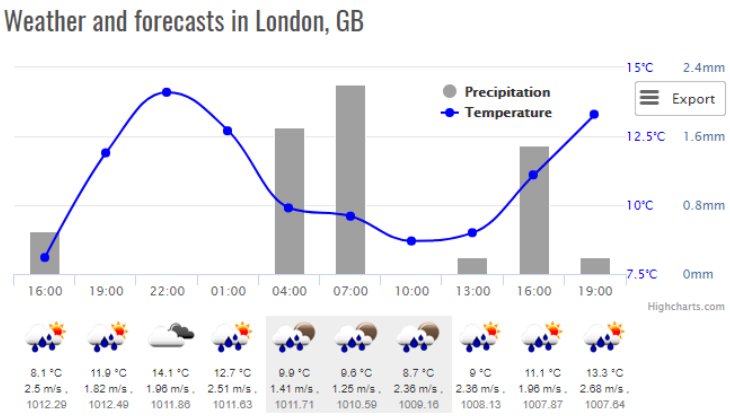
```
com.example.android.mapexample
```

7. Klik **Save**.

   Kunci API terbatas android yang baru akan muncul dalam daftar API *key* untuk proyek anda. Saat menggunakan API standar, bisa menggunakan API *key* yang sama untuk aplikasi Google maps android API dan aplikasi Google places API untuk android.

2.1.10 OpenWeather API

Openweather adalah layanan online yang menyediakan data cuaca terkini, termasuk data prakiraan dan data historis terkini untuk para pengembang layanan web dan aplikasi *mobile*. Untuk sumber data, Openweather menggunakan layanan siaran meteorologi, data mentah dari stasiun cuaca bandara, data mentah dari stasiun radar, dan data mentah dari stasiun cuaca resmi lainnya, seperti dapat dilihat pada Gambar 2.8:



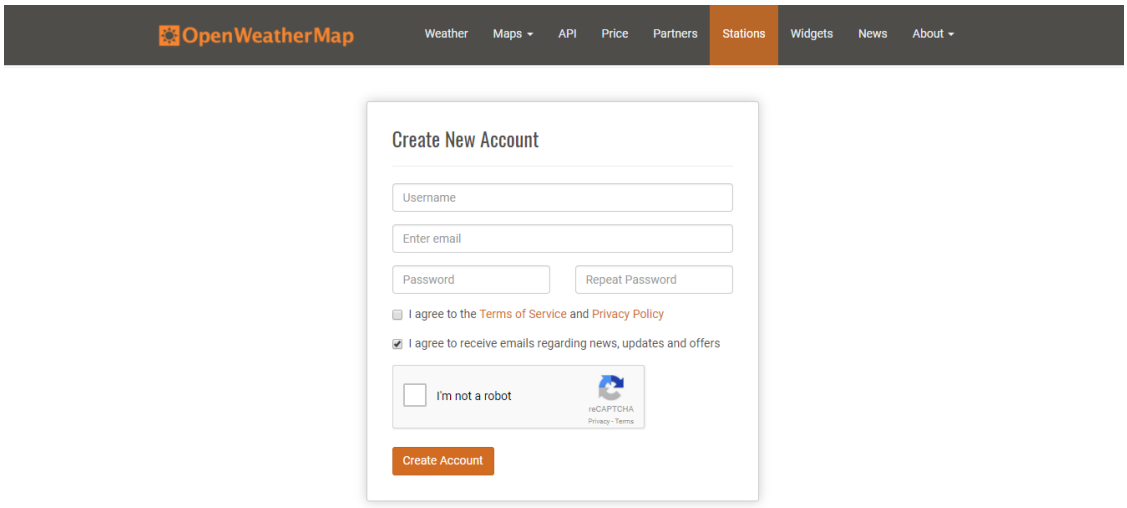
Sumber Gambar : https://openweathermap.org/

Gambar 2. 8 Contoh Data Cuaca Openweather

2.1.10.1 Cara menggunakan Openweather API

Berikut adalah langkah-langkah cara menggunakan Openweather API:

1. Mendapatkan API *key* dengan cara mendaftarkan akun pada alamat *website : https://home.openweathermap.org/users/sign_up*



Sumber Gambar : https://home.openweathermap.org/users/sign_up

Gambar 2. 9 Halaman Daftar

2. Aktivasi API *key* untuk versi gratis dan membutuhkan waktu 10 menit.

3. Setelah mendapatkan API *key* lakukan pemanggilan melalui *link : http://api.openweathermap.org/data/2.5/forecast?id=524901&APPID={APIKEY}*
4. APPID {APIKEY} adalah API key unik yang telah didapatkan.

2.1.10.2 Cara Mendapatkan Respon API yang akurat

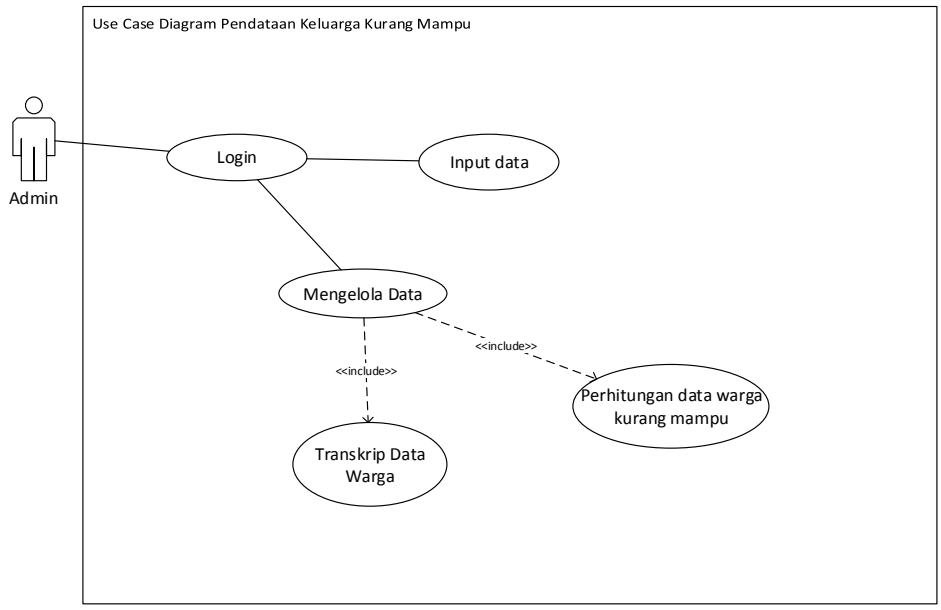
1. Jangan mengirim permintaan lebih dari 1 kali per 10 menit dari satu perangkat / satu API *key*. Biasanya cuaca tidak terlalu sering berubah.
2. Gunakan nama server **api.openweathermap.org**. Jangan pernah menggunakan alamat IP server.
3. Panggil API berdasarkan ID kota**,** bukan nama kota, koordinat kota atau kode pos.
4. Akun gratis dan Startup memiliki keterbatasan kapasitas dan ketersediaan data. Jika tidak mendapatkan *respond* dari server jangan segera mengulang permintaan, tetapi tunggu setelah 10 menit. Disarankan untuk menyimpan data permintaan sebelumnya.

2.1.11 UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [15]. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan kelas dan operasi dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET [8]. UML bisa juga berarti notasi grafis untuk menggambar diagram konsep perangkat lunak. Satu dapat menggunakannya untuk menggambar diagram dari domain masalah, desain perangkat lunak yang diusulkan, atau implementasi perangkat lunak yang sudah selesai.

2.1.11.1 Use Case Diagram

Use case diagram merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang menggambarkan bagaimana seseorang atau aktor dalam menggunakan dan memanfaatkan sistem. Use case terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi use case, dan skenario use case [9].



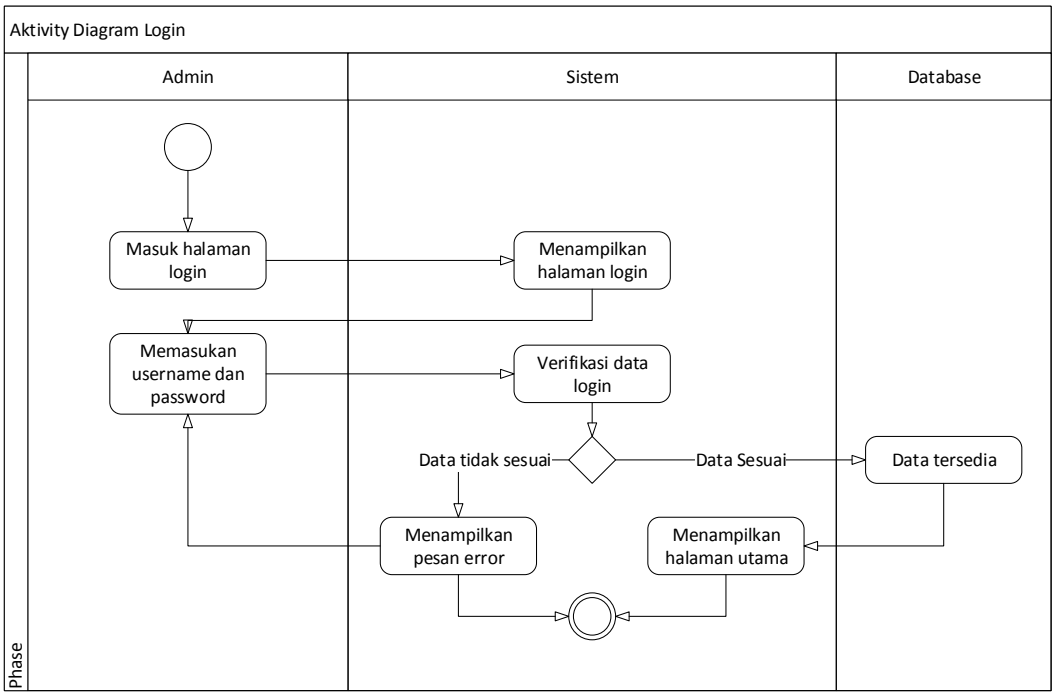
Gambar 2. 10 Use Case Diagram

2.1.11.2 Activity Diagram

Activity diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah flowchart karena dapat dimodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Seringkali bermanfaat bila dibuat sebuah aktivitas terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Activity diagram juga sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai use case saling berinteraksi. Dapat digunakan statechart diagram untuk memodelkan perilaku dinamis satu kelas atau objek. Statechart diagram memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan

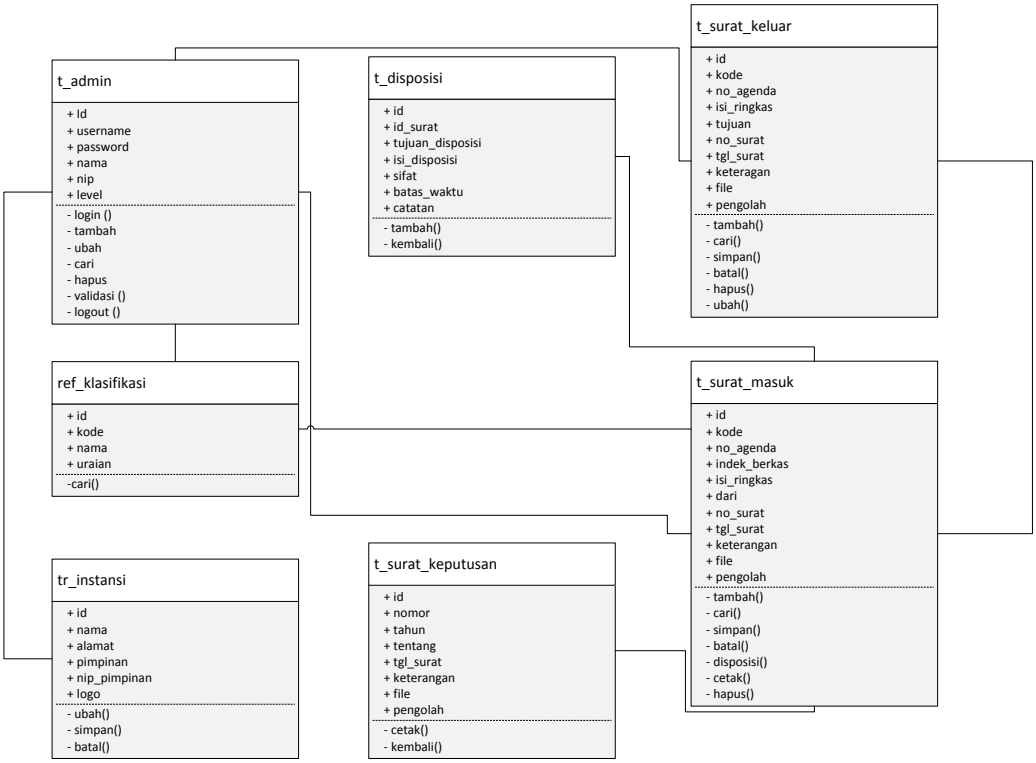
sebuah transisi dari satu *state* atau aktivitas ke *state* atau aktivitas lainnya. Diagram aktivitas paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses [8].



Gambar 2. 11 Activity Diagram

2.1.11.3 Class Diagram

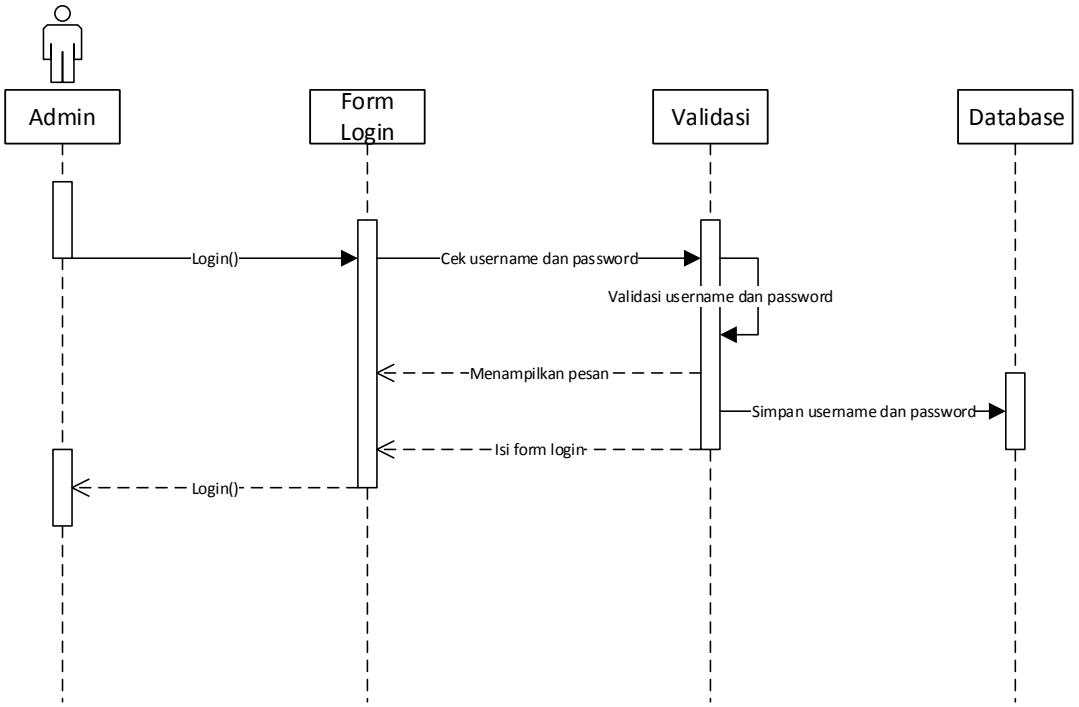
Class diagram membantu dalam visualisasi struktur kelas–kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap–tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan–aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama proses analisis, class diagram memperlihatkan aturan–aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat [8].



Gambar 2. 12 Class Diagram

2.1.11.4 Sequence Diagram

Diagram sequence menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara khusus berasosiasi dengan use case. Sequence diagram memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case. Diagram sequence sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti [8].



Gambar 2. 13 Sequence Diagram

2.1.12 Pengujian Black Box

Pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinsikan. Metode *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program.

Pengujian *black box* dapat menemukan kesalahan dalam kategori berikut [12]:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

2.1.13 Skala Likert

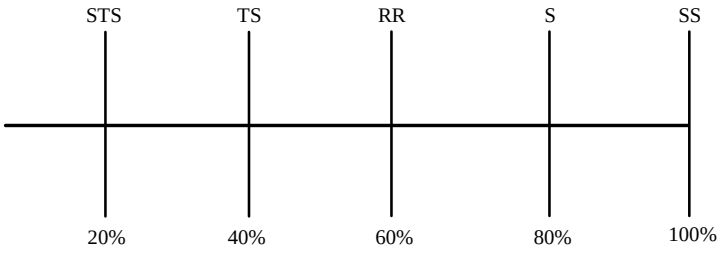
Skala *likert* adalah metode yang digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau kelompok orang tentang fenomena sosial. Dengan skala *likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Selanjutnya indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen pernyataan atau pertanyaan. Instrumen penelitian dapat dibuat dalam bentuk checklist ataupun pilihan ganda.

Jawaban setiap item instrumen yang menggunakan skala *likert* mempunyai gradasi dari sangat positif sampai sangat negatif dimana jawaban yang diberikan akan diberikan skor. Jawaban dapat berupa kata-kata sebagai berikut [13]:

a. Sangat setuju dengan skor 5
b. Setuju dengan skor 4
c. Ragu-ragu dengan skor 3
d. Tidak setuju dengan skor 2
e. Sangat tidak setuju dengan skor 1

Langkah-langkah skala *likert* adalah :

1. Menghitung jumlah skor ideal (kriterium) yaitu dengan rumus :
   Kriterium = nilai bobot maksimal x jumlah responden
2. Menghitung jumlah jawaban dari respoden dalam bentuk persentase, digunakan rumus :
   $$P = \frac{Total\ nilai}{kriterium} x100\%$$
   Keterangan :
   P : nilai persentasi yang dicari
   Total nilai : jumlah frekuensi x nilai bobot
3. Selanjutnya skor dimasukkan kedalam bentuk skala penilaian *likert*



Gambar 2. 14 Skala Penilaian Likert