

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori menjelaskan beberapa teori-teori dan penjelasan yang berkaitan dengan aplikasi atau media yang akan dibangun. Landasan teori yang digunakan meliputi Pengertian Aplikasi, Android, GPS, LBS, Google Maps, Open Weather Map, Facebook, Pengertian Java, UML, Web Service, MySQL dan PHP.

2.1.1 Push Notification

Push Notification merupakan suatu komunikasi jaringan, yaitu server akan mengirimkan pesan pemberitahuan ke client jika ada perubahan data, sehingga client tidak perlu melakukan proses request data tiap periode untuk mengambil data pemberitahuan. Teknologi Push Notification untuk device mobile sangat efektif karena berjalan pada background proses sehingga memungkinkan aplikasi menerima pesan. Push Notification dapat diimplementasikan pada device berbasis Android dan device yang lain. Pada sistem operasi Android proses Push Notification dapat memanfaatkan layanan Google Cloud Messaging (GCM) untuk mengirim pesan atau notifikasi yang disediakan oleh Google [11].

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. Misalkan ketika seseorang akan menggunakan jasa *online shop*. Biasanya mereka akan melihat testimoni dari orang-orang yang sudah bertransaksi sebelumnya, apakah banyak yang merekomendasikan atau tidak. Jika banyak testimoni positif maka akan menambah keyakinan orang untuk bertransaksi.

2.1.3 Aplikasi

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju [12].

Pada dunia teknologi aplikasi merupakan sebuah perangkat lunak yang dibuat untuk membantu atau mempermudah pekerjaan manusia. Aplikasi juga merupakan suatu program perangkat lunak yang berjalan pada suatu sistem tertentu. Sebuah aplikasi dapat dibuat dan dikembangkan dengan tujuan untuk melakukan tugas yang bersifat umum atau juga dapat dikembangkan untuk melakukan tugas yang bersifat spesifik/khusus.

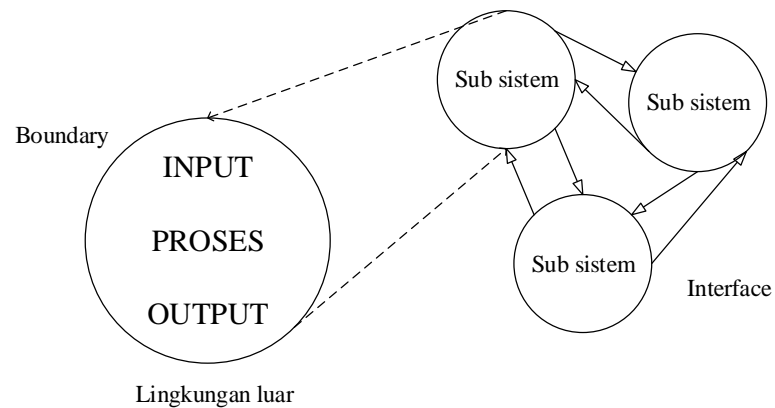
2.1.4 Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu, ini menjelaskan bahwa sistem bekerja dalam suatu jaringan kerja dari suatu prosedur yang saling berhubungan satu sama lain untuk menyelesaikan tujuan dan sasaran yang dimaksud. Definisi sistem juga dapat dijelaskan oleh Jogiyanto dalam bukunya *Analisa dan Desain sistem informasi*, menerangkan sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

Sistem juga diartikan sebagai sekumpulan elemen yang bekerja sama dalam suatu kesatuan untuk melaksanakan suatu fungsi yang berguna. Dalam bukunya Jogiyanto sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen. Definisi sistem menurut Abdul Kadir adalah sekelompok elemen-elemen yang saling terintegrasi dengan maksud dan tujuan yang sama untuk melaksanakan sasaran yang telah ditentukan [13].

2.1.4.1 Karakteristik Sistem

Suatu sistem memiliki karakteristik yang akan merepresentasikan ciri dari sebuah sistem itu sendiri. Berikut adalah karakteristik dari sebuah sistem :



Sumber Gambar : Materi Ilmu Komputer [14]

Gambar 2. 1 Karakteristik Sistem

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Hasil dari olahan itu akan menghasilkan sesuatu yang berbeda.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengolah semua masukan. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya [14].

2.1.4.2 Klasifikasi Sistem

Menurut Agus Mulyanto dalam bukunya yang berjudul *Sistem Informasi Konsep dan Aplikasi*. Sistem dapat di klasifikasikan berbagai sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem Abstrak dan Fisik (*Physical System*)

Sistem Abstrak yaitu sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara Fisik, misalnya sistem agama. Sementara sistem fisik adalah sistem yang keberadaannya dapat dilihat secara fisik, misalnya perusahaan, komputer.

2. Sistem Alamiah dan Buatan (*Human Made System*)

Sistem alamiah (*natural system*) yaitu sistem yang terbentuk melalui proses alami, misalnya sistem tatasurya, pencernaan. Sementara sistem buatan manusia adalah sistem yang dirancang dan dibangun oleh manusia yang melibatkan interaksi dengan mesin misalnya produksi di pabrik.

3. Sistem Tertentu dan Tidak Tertentu (*Probabilistic System*)

Sistem tertentu (*deterministic system*) yaitu sistem yang cara beroperasinya sudah dapat diprediksi, interaksi-interaksi didalamnya dapat dideteksi dengan pasti dan outputnya dapat diramalkan, misalnya pengolahan data. Sementara sistem tak tentu adalah sistem yang outputnya tidak dapat diprediksi dengan pasti karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Terbuka (*Open System*)

Sistem tertutup yaitu sistem yang tidak berhubungan dengan dunia luar dan tidak terpengaruh dengan lingkungan luarnya (bekerja secara otomatis). Sebenarnya sistem tertutup tidak ada yang ada adalah relatif tertutup. Sementara sistem terbuka adalah sistem yang mempunyai hubungan dengan dunia luar dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan output untuk subsistem yang lain.

2.1.5 Informasi

Informasi adalah data yang diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau yang akan datang. Informasi juga merupakan fakta-fakta atau data yang telah diproses sedemikian rupa atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi [14].

Informasi dibagi menjadi beberapa jenis berdasarkan fungsi, format penyajian, lokasi peristiwa dan bidang kehidupan sebagai berikut :

1. Informasi Berdasarkan Fungsi

Jenis informasi yang satu ini ialah informasi yang berdasarkan materi dan kegunaan informasi. Informasi jenis ini antara lain yaitu suatu

informasi yang menambah suatu pengetahuan dan informasi yang mengajari sih pembaca (Informasi edukatif).

2. Informasi Berdasarkan Format Penyajian

Jenis informasi yang satu ini ialah sebuah informasi yang berdasarkan bentuk penyajian informasi. Informasi jenis ini, antara lain yaitu berupa foto, karikatur, lukisan abstrak, dan tulisan teks.

3. Informasi Berdasarkan Lokasi Peristiwa

Jenis informasi yang satu ini ialah sebuah informasi yang berdasarkan lokasi peristiwa berlangsung, yaitu sebuah informasi dari dalam negeri dan informasi dari luar negeri.

4. Informasi Berdasarkan Bidang Kehidupan

Jenis informasi yang satu ini ialah sebuah informasi yang berdasarkan bidang-bidang kehidupan yang ada, misalnya pada pendidikan, olahraga, musik, sastra, budaya, dan iptek [14].

2.1.6 Kualitas Sistem

Pada penelitian yang dilakukan oleh istianingsih dan utami pada tahun 2009 hasil estimasinya menunjukkan bahwa kualitas sistem terbukti secara signifikan berpengaruh positif terhadap kepuasan pengguna, dan pada penelitian yang dilakukan Adrianty pada tahun 2012 juga menyatakan bahwa kualitas sistem terbukti secara signifikan berpengaruh positif terhadap kepuasan pengguna [15]. Untuk mengetahui suatu sistem itu berkualitas atau tidak, dinilai dari indikator atau dimensi-dimensi, diantaranya :

1. Kemudahan Untuk Digunakan (*Ease Of Use*)

Sistem informasi yang dapat dikatakan sebagai sistem yang berkualitas jika dirancang untuk memenuhi kemudahan dalam penggunaan sistem informasi tersebut.

2. Kecepatan Akses (*Response Time*)

Kecepatan akses merupakan salah satu indikator kualitas sistem informasi. Jika sistem informasi memiliki kecepatan akses yang optimal

maka layak untuk dikatakan bahwa sistem informasi yang diterapkan memiliki kualitas yang baik.

3. Keandalan Sistem (*Reliability*)

Keandalan sistem informasi adalah ketahanan sistem informasi dari kerusakan dan kesalahan. Keandalan sistem informasi ini juga dapat dilihat dari sistem informasi dalam melayani kebutuhan pengguna tanpa adanya masalah yang dapat mengganggu kenyamanan pengguna dalam menggunakan sistem tersebut.

4. Fleksibilitas Sistem (*Flexibility*)

Fleksibilitas yang dimaksud adalah kemampuan sistem informasi dalam melakukan perubahan-perubahan yang terkait dengan memenuhi kebutuhan pengguna. Pengguna akan merasa lebih puas menggunakan sistem informasi jika sistem tersebut fleksibel dalam memenuhi kebutuhan pengguna.

5. Keamanan Sistem (*Security*)

Keamanan sistem dapat dilihat melalui program yang tidak dapat diubah-ubah oleh pengguna yang tidak bertanggung jawab dan juga program tidak dapat terhapus jika terdapat kesalahan dari pengguna [15].

2.1.7 Kualitas Informasi

Kualitas informasi menunjukkan kualitas produk yang dihasilkan oleh aplikasi sistem informasinya dan informasinya akan mempunyai pengaruh pada pemakaiannya dan pada sistemnya. Kualitas informasi yang bermanfaat adalah relevan, akurat, tepat waktu, conciseness, jelas, dapat dikuantifikasi, dan konsisten. Semakin baik kualitas informasi, akan semakin tepat pula keputusan yang diambil [15]. Kualitas dari suatu informasi (quality of information) tergantung dari 3 hal, sebagai berikut :

1. Akurat

Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak biasa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber

informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

2. Tepat Pada Waktunya

Tepat berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

3. Relevan

Relevan berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda [15].

2.1.8 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. 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 terbaharukan.

Sistem Operasi Android memiliki beberapa komponen utama yang disebut dengan Arsitektur Platform Android. Berikut adalah diagram komponenkomponen 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

Fondasi 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 .

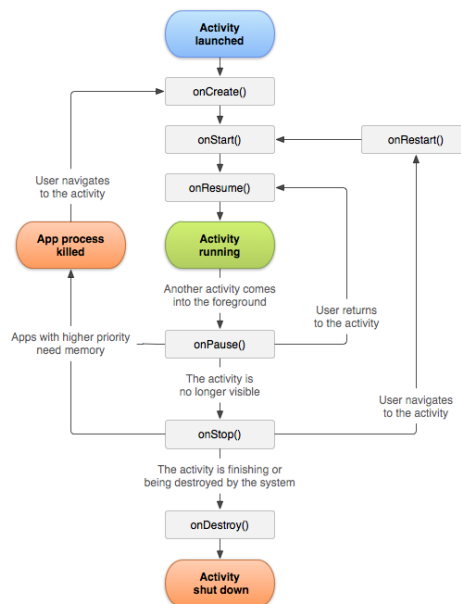
6. System Apps

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 mendefinisikan 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 *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 pernananya 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.

2.1.9 GPS

Global Positioning System atau yang biasa disingkat dengan GPS adalah alat navigasi elektronik yang menerima informasi dari 4 - 12 satelit sehingga GPS bisa memperhitungkan posisi dimana kita berada di Bumi. Satelit GPS tidak

mentransmisikan informasi posisi kita, yang ditransmisikan satelit adalah posisi satelit dan jarak penerima GPS kita dari satelit [16]. GPS terdiri dari tiga bagian yaitu sistem kontrol, satelit dan pengguna. Sistem kontrol adalah bagian yang mengontrol pergerakan satelit-satelit yang ada dan saling berinteraksi satu sama lain kemudian pengguna adalah alat navigasi yang digunakan seperti perangkat *mobile* yang kini sudah memiliki fitur GPS di dalamnya. GPS biasanya digunakan untuk menunjukkan suatu lokasi yang berada di permukaan bumi dengan tingkat akurasi yang cukup baik yaitu kurang dari 10 meter selama tidak ada benda padat yang dapat menghambat sinyal untuk mendapatkan lokasi pengguna.

1. GPS Control Segment

Control segment GPS terdiri dari lima stasiun yang berada di pangkalan *Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia* dan *Kwajalein*. Kelima stasiun ini adalah mata dan telinga bagi GPS. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data ephemeris, yang kemudian nantinya dikirimkan ke alat navigasi yang kita miliki.

2. GPS Space Segment

Space Segment adalah terdiri dari sebuah jaringan satelit yang terdiri dari beberapa satelit yang berada pada orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20.183 km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan. Terdapat 2 jenis gelombang yang hingga saat ini digunakan sebagai alat navigasi berbasis satelit. Masing-masingnya adalah gelombang L1 dan L2, dimana L1 berjalan pada frekuensi 1575.42 MHz yang bisa digunakan oleh masyarakat umum, dan L2 berjalan pada frekuensi 1227.6 Mhz dimana jenis ini hanya untuk kebutuhan militer saja.

3. GPS User Segment

User segment terdiri dari antenna dan prosesor *receiver* yang menyediakan *positioning*, kecepatan dan ketepatan waktu ke pengguna. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali (GPS *Control Segment*).

2.1.10 Location Based Service

Location Based Service 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" [17].

Dua unsur utama dari Location Based Services (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 "com.google.android.maps;".
2. *Location Providers (API Location)*: Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. PAI 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* [18].

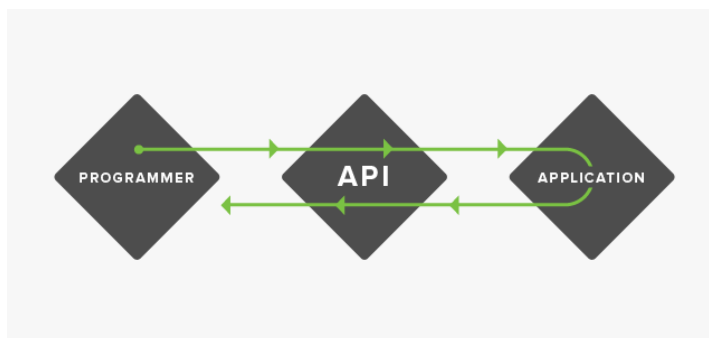
2.1.11 API (Application Programming Interface)

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 [19].

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

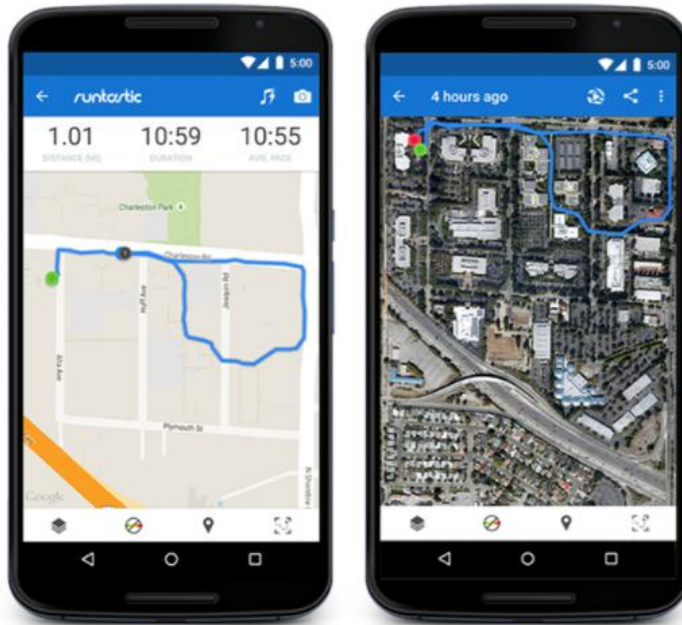
Dengan demikian, API adalah standar komunikasi yang dibuka oleh perusahaan *software*, agar dapat dimanfaatkan oleh pengembang pihak ketiga untuk mendesain aplikasi yang memanfaatkan layanan mereka dengan mudah.



Gambar 2. 4 API

2.1.12 Google Maps Android API

Google Maps Android 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 merespons 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. 5 GPS pada android

2.1.12.1 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 Android 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:

<pre>BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1 com.example.android.mapexample</pre>
--

7. Klik **Save**.

Kunci API terbatas-Android yang baru akan muncul dalam daftar *APIkey* untuk proyek Anda. Saat menggunakan API standar, bisa menggunakan API *key* yang sama untuk aplikasi Google Maps Android API dan aplikasi *Google Places API for Android*.

2.1.12.2 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 respons dari server jangan segera mengulang permintaan, tetapi tunggu setelah 10 menit. Disarankan untuk menyimpan data permintaan sebelumnya.

2.1.13 Java

Java adalah salah satu bahasa pemrograman yang paling populer. *Java* dapat digunakan untuk berbagai hal, termasuk pengembangan perangkat lunak, aplikasi *mobile*, dan pengembangan sistem yang besar. Inilah yang membuat bahasa pemrograman *Java* sangat terkenal di lingkungan pengembang perangkat lunak.

Seperti bahasa pemrograman lain, bahasa *Java* memiliki struktur sendiri, aturan sintaks, dan paradigma pemrograman. paradigma pemrograman bahasa *Java* didasarkan pada konsep OOP. Bahasa *Java* merupakan turunan bahasa *C* , sehingga aturan sintaks yang terlihat akan seperti bahasa *C*. Misalnya, blok kode yang modular dalam metode dan dibatasi oleh karakter ‘{‘ dan ‘}’, dan variabel dideklarasikan sebelum digunakan. Secara struktural, bahasa *Java* diatur dengan

package. Di Dalam *package* ada *class*, dan dalam *class* ada *method*, variabel, konstanta, dan banyak lagi.

Adapun sintak dalam bahasa pemrograman Java sebagai berikut :

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

2.1.14 PHP

PHP adalah singkatan dari PHP: *Hypertext PreProcessors*, PHP saat ini adalah bahasa pemrograman interpreter yang paling banyak digunakan saat ini dikarenakan bersifat *open source* dan juga paling banyak didukung oleh banyak *web server*. PHP (*Hypertext Preprocessor*), merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak *web server* (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang me-*request*-nya.

Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris - baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program di channel *output* standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk *server web* yang paling modern dan sebagai penerjemah mandiri pada sebagian besar sistem operasi dan komputer *platform* [20].

Adapun contoh sintak dalam bahasa pemrograman PHP (*Hypertext Preprocessor*) sebagai berikut :

```
<?php
    echo "Hello World...!!!";
    echo "<br />";
    print "Hello Again World...!!!";
    print "<br /><br />";
?>
```

2.1.15 JSON (Javascript Object Notation)

JSON adalah format penukaran data yang sederhana, bagi programmer format ini mudah dibaca dan ditulis, sedangkan bagi mesin, format ini mudah untuk proses parse dan generate. 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 programmer bahasa pemrograman C, C++, C#, Java, JavaScript, Perl, Python, and many others. Hal ini menjadikan JSON sebagai bahasa penukaran data yang ideal [21]. 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*.
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

2.1.16 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *Relational Database Management System* (RDBMS). MySQL merupakan *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License) dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*) [22].

Berikut contoh sintak pada MySQL :

1. Membuat database

```
CREATE DATABASE bioskop;
```

2. Melihat database

```
SHOW DATABASES;
```

3. Membuat tabel

```
CREATE TABLE operator(
  id VARCHAR (20) NOT NULL,
  nama VARCHAR (50) NOT NULL,
  password VARCHAR(100) NOT NULL,
  created_at DATETIME NOT NULL,
  updated_at TIMESTAMP,
  PRIMARY KEY (id)
);
```

4. Melihat tabel

```
SHOW TABLES;
```

5. Menghapus tabel

```
DROP TABLE operator;
```

6. Menghapus database

```
DROP DATABASE bioskop;
```

Dalam penggunaan *database* MySQL, setiap perintah yang diketikkan disebut *query*. Perintah MySQL dapat dikategorikan menjadi 3 sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*).

DDL (Data Definition Language)

Perintah dalam SQL yang pertama adalah perintah DDL. DDL sendiri merupakan kependekan dari apa yang dikenal dengan nama *Data Definition Language*. DDL dapat berarti sebuah perintah yang berhubungan dengan pendefinisian dari suatu struktur *database*. Terdapat beberapa perintah DDL pada MySQL sebagai berikut :

1. CREATE berfungsi untuk membuat *database* baru, tabel baru, *view* baru dan kolom.
2. ALTER berfungsi untuk mengubah struktur tabel. Seperti mengganti nama tabel, menambah kolom, mengubah kolom, menghapus kolom maupun memberikan atribut pada kolom.
3. DROP berfungsi untuk menghapus *database* dan tabel.
4. TRUNCATE berfungsi untuk menghapus semua catatan dari tabel.
5. COMMENT berfungsi untuk menambahkan komentar pada data.
6. RENAME berfungsi untuk mengubah nama obyek.

Adapun contoh sintaks dari perintah DDL pada MySQL sebagai berikut :

```
CREATE DATABASE NILAI;  
ALTER TABLE Mahasiswa ADD (NoTelp char(8));  
DROP INDEX nama_index ;  
TRUNCATE TABLE table_barang;  
RENAME table_barang to table_barang_gudang;
```

DML (Data Manipulation Language)

Data Manipulation Language (DML) ialah perintah yang digunakan untuk mengelola/memanipulasi data dalam *database*. Terdapat beberapa perintah DML pada MySQL sebagai berikut :

1. SELECT berfungsi untuk mengambil/menampilkan data dari *database*.
2. INSERT berfungsi untuk memasukkan data ke dalam tabel.

3. UPDATE berfungsi untuk memperbarui data dalam tabel.
4. DELETE berfungsi untuk menghapus data dari tabel.
5. CALL berfungsi untuk memanggil subprogram PL / SQL atau Java.
6. EXPLAIN PLAN berfungsi untuk menjelaskan jalur akses ke data.
7. LOCK TABLE berfungsi untuk mengunci tabel.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut :

```
INSERT INTO mahasiswa VALUES ("08052926", "Frenky", "70");
SELECT nama_mahasiswa FROM mahasiswa;
DELETE FROM mahasiswa;
UPDATE mahasiswa SET nama_mahasiswa='ujang'
WHERE nim='08052926';
```

DCL (Data Control Language)

Data Control Language (DCL) ialah perintah yang digunakan untuk melakukan pengontrolan data dan server databasenya. Terdapat beberapa perintah DCL pada MySQL sebagai berikut :

1. GRANT berfungsi untuk memberikan hak akses pengguna ke database.
2. REVOKE berfungsi untuk menghilangkan hak akses yang telah diberikan dengan perintah GRANT.

Adapun contoh sintaks dari perintah DML pada MySQL sebagai berikut :

```
GRANT all privileges on * to nm_user@localhost identified
by 'nm_passwd' with grand option;

REVOKE all on nm_db.nm_tbl from nm_user@localhost
identified by 'nm_passwd';
```

2.1.17 Web Server

Web Server adalah sebuah perangkat lunak yang bertugas menerima permintaan client melalui port HTTP maupun HTTPS dan merubahnya ke dalam format HTML. Terdapat beberapa format selain HTML yaitu PHP atau ASP, tetapi format – format tersebut hanyalah berfungsi untuk menghubungkan HTML dengan *database*. Fungsi utama dari web server yaitu untuk melakukan atau akan transfer berkas permintaan pengguna melalui komunikasi yang telah ditentukan sedemikian rupa. Halaman web yang diminta terdiri dari berkas teks, video, gambar, file dan

banyak lagi. Pemanfaatan web server berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang didalamnya berupa teks, video, gambar, file dan banyak lagi.

Web server yang umum digunakan adalah *Apache*. Tugas utama dari Web server adalah menerjemahkan permintaan ke dalam respon yang cocok untuk keadaan pada saat itu, ketika klien membuka komunikasi dengan *Apache*, *Apache* mengirimkan permintaan untuk sumber daya. *Apache* menyediakan sumber daya yang baik atau memberikan respon alternatif untuk menjelaskan mengapa permintaan tidak dapat terpenuhi. Setiap komunikasi HTTP dimulai dengan permintaan dan berakhir dengan jawaban. *Executable Apache* mengambil nama dari protokol, dan pada sistem Unix umumnya disebut `httpd`, kependekan daemon HTTP [23].

2.1.17.1 Contoh Web Server

Berikut adalah beberapa contoh web server antara lain :

2.1.17.1.1 KSWEB: Server + PHP + MySQL

KSWEB adalah satu set untuk pengembang web untuk platform Android. Paket ini terdiri dari: web server, server FTP, bahasa pemrograman PHP, MySQL DBMS dan scheduler. KSWEB tidak perlu hak root untuk memfungsikan, kecuali, tentu saja, jika ingin menggunakan server pada port 80. KSWEB adalah aplikasi shareware. Setelah peluncuran pertama, pengguna akan memiliki 5 hari percobaan untuk menguji aplikasi. Biaya KSWEB PRO adalah \$ 3,99. Biaya KSWEB Standar adalah \$ 2,99.

2.1.17.1.2 PAW Server

PAW Server adalah Server Web untuk perangkat Android. Webservice PAW dapat menggunakan fungsi telepon dari browser web, melayani halaman web sendiri atau mengembangkan aplikasi web yang diaktifkan ponsel. Dukungan Tasker dan Locale Plug-in. Plug-in PHP tersedia. Perangkat lunak ini masih dalam versi beta.

2.1.17.1.3 Bit Web Server

Aplikasi ini adalah aplikasi server web untuk android yang merupakan alat untuk pengembang web, termasuk `LIGHTTPD` sebagai Web Server, PHP sebagai

Server PHP, MYSQL sebagai Server MySQL, PhpMyAdmin sebagai Klien MySQL , dan MSMTTP sebagai SMTP Client. Itu semua dibungkus ke dalam satu aplikasi yang disebut Bit Web Server. Jadi mudah dijalankan di perangkat android, cukup sekali klik maka Anda dapat menjalankan aplikasi web dengan skrip PHP dan database MySQL. Aplikasi ini dirancang khusus untuk ponsel dan tablet, jadi aplikasi ini membutuhkan memori yang sangat kecil.

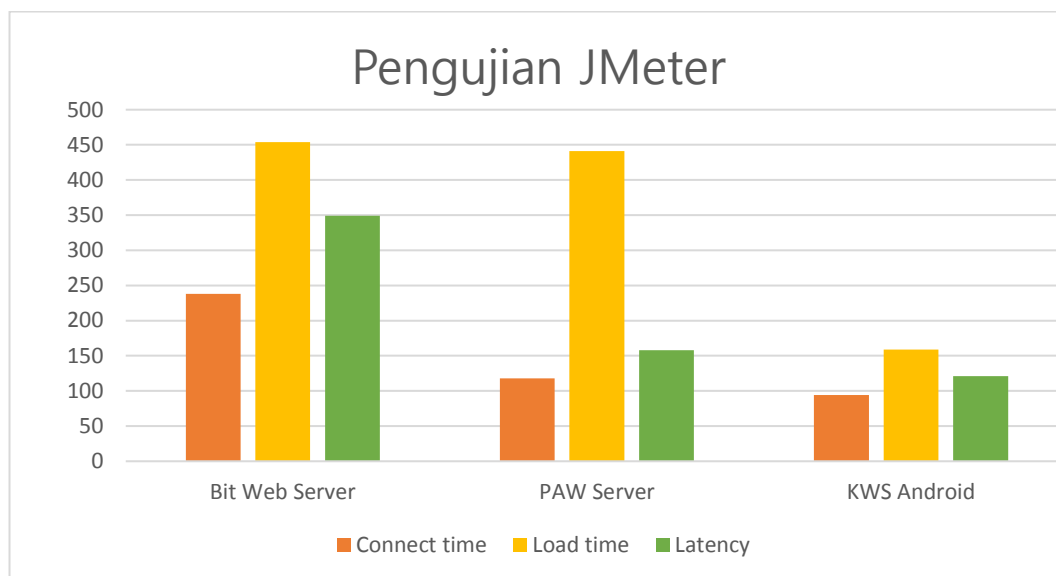
Aplikasi ini berjalan tanpa koneksi internet, membutuhkan internet hanya ketika pertama kali memeriksa lisensi, setelah aplikasi ini memiliki lisensi yang sah maka aplikasi ini dapat berjalan tanpa koneksi internet (OFFLINE). Bit Web Server dapat menjalankan berbagai CMS seperti Wordpress, Joomla, Drupal, Prestashop, dll. Selain itu CMS juga dapat menjalankan kerangka kerja PHP seperti Code Igniter, YII, CakePHP, dll. Aplikasi ini sama seperti WAMP atau LAMP atau XAMPP di komputer atau laptop, tetapi ini untuk perangkat android.

2.1.17.2 Perbandingan Web Server

2.1.17.1.4 JMeter

Pada tahap ini akan dipaparkan perbandingan antara aplikasi Web Server Android yang sudah digunakan setelah diuji menggunakan JMeter dan Apache Benchmark. Pertama adalah perbandingan aplikasi Web Server dengan JMeter. Dengan penjabaran tes sebagai berikut :

- a. Connect time adalah lama waktu terhubung request ke server
- b. Load time adalah lama waktu server memuat sebuah informasi kepada yang merequest.
- c. Sandar Deviasi adalah sebaran data dalam sampel, serta seberapa dekat titik data individu terhadap nilai mean atau rata-rata nilai sample uji.
- d. Latency adalah jeda waktu dalam pengantaran paket data dari pengirim ke penerima.



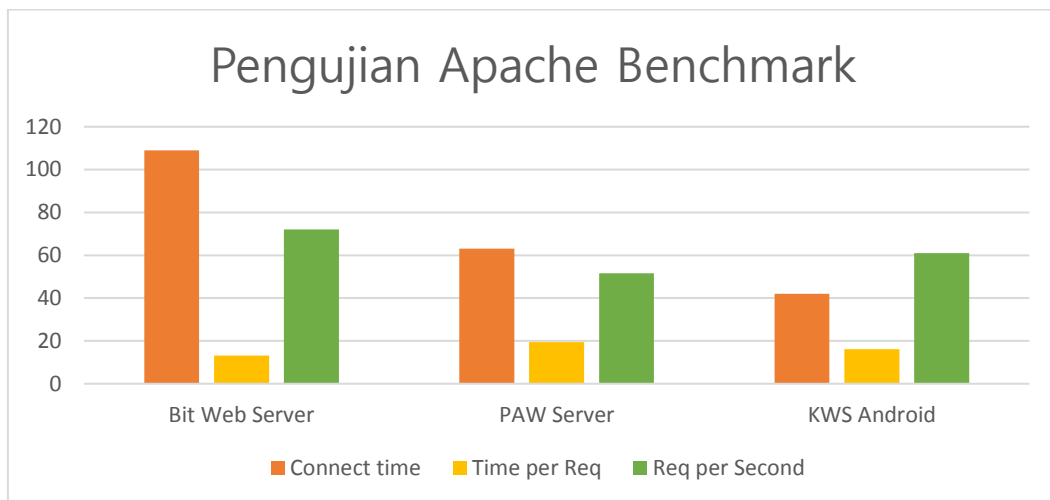
Gambar 2. 7 Penguujian JMeter

Dari Grafik diatas dapat disimpulkan bahwa secara umum hasil pengujian kWS Android lebih unggul dari kedua aplikasi web server lainnya. dalam kualitas kecepatan konektifitas dari client terhadap server, kWS Android lebih unggul dengan jumlah 94 ms, waktu untuk memuat konten 159 ms serta latency yang lebih cepat dari aplikasi lainnya.

2.1.17.1.5 Apache Benchmark

Perbandingan selanjutnya menggunakan Apache Bench terhadap tiga aplikasi web server android, untuk menentukan kualitas aplikasi yang lebih baik. Dengan penjabaran tes sebagai berikut :

- a. Transfer rate adalah jumlah yang dapat dibawa dari sebuah titik (node) ke titik (node) lain dalam jangka waktu tertentu.
- b. Time per Request adalah waktu yang dibutuhkan dalam memproses 1 buah request
- e. Sandar Deviasi adalah sebaran data dalam sampel, serta seberapa dekat titik data individu terhadap nilai mean atau rata-rata nilai sample uji.
- f. Request per Second adalah berapa request yang dapat diproses dalam satu waktu.



Gambar 2. 8 Penguujian Apache Benchmark

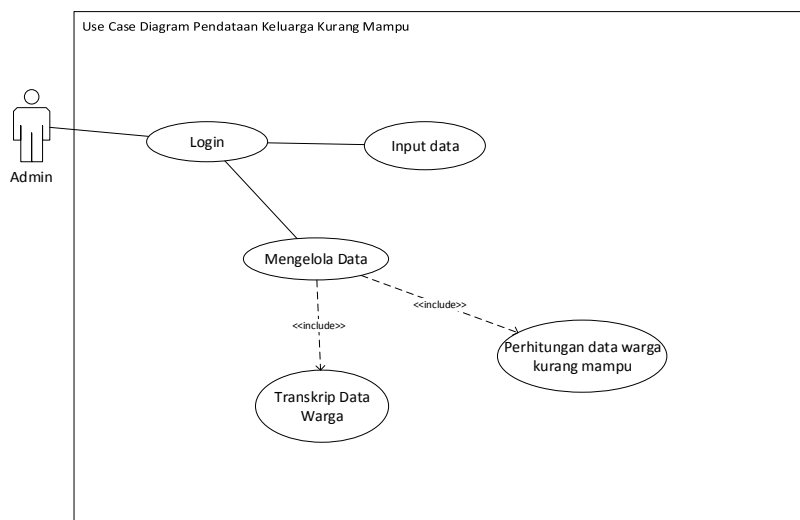
Dari hasil perbandingan diatas dapat disimpulkan dari pengujian dengan Apache Benchmark bahwa Aplikasi Bit Web Server lebih unggul dari kedua aplikasi lainnya terhadap 2 point yaitu Time per request yang hanya membutuhkan waktu 13.112 ms dan Request per second, Bit Web Server dapat melayani lebih banyak Client dalam satu waktu yakni 72 Request, sedangkan PAW Server 52 request dank WS 61 request.

2.1.18 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. 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 class dan operation dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET [24]. Unified Modeling Language (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.

2.1.18.1 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 [25].



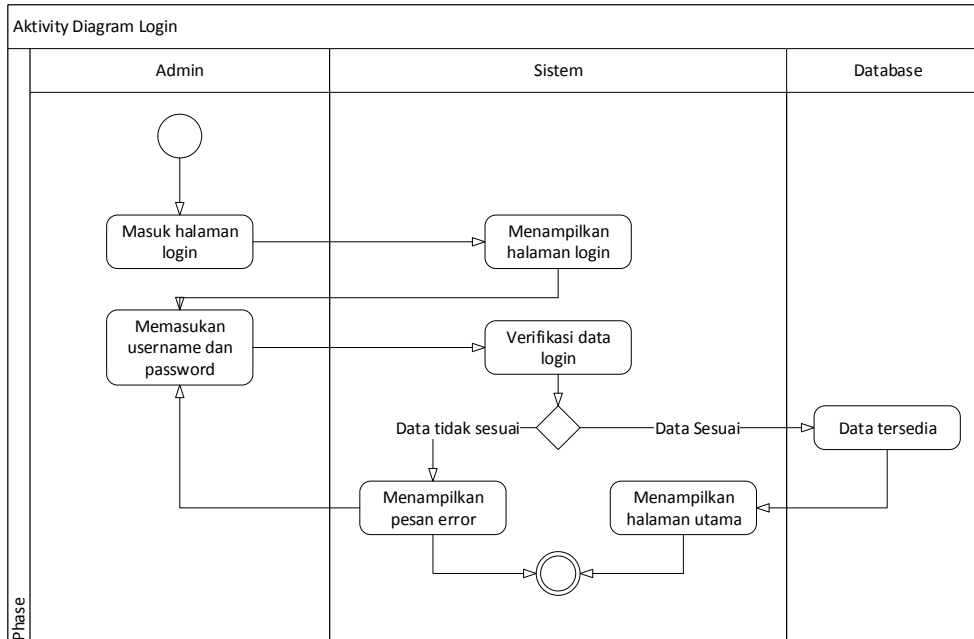
Gambar 2. 9 Use case diagram

2.1.18.2 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 activity 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 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, dan aksi yang menyebabkan perubahan satu state lainnya, dan aksi yang menyebabkan perubahan

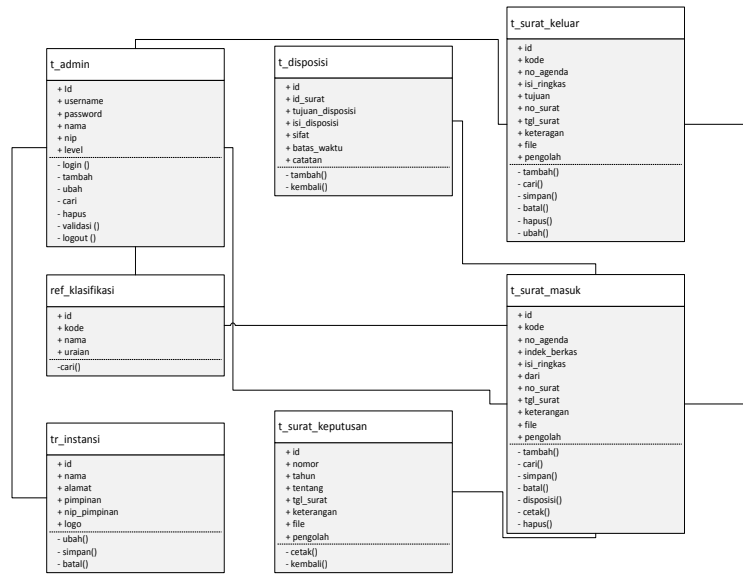
satu state atau aktivitas. Diagram aktivitas paling cocok digunakan untuk memodelkan urutan aktivitas dalam suatu proses [24].



Gambar 2. 10 Activity diagram

2.1.18.3 Class Diagram

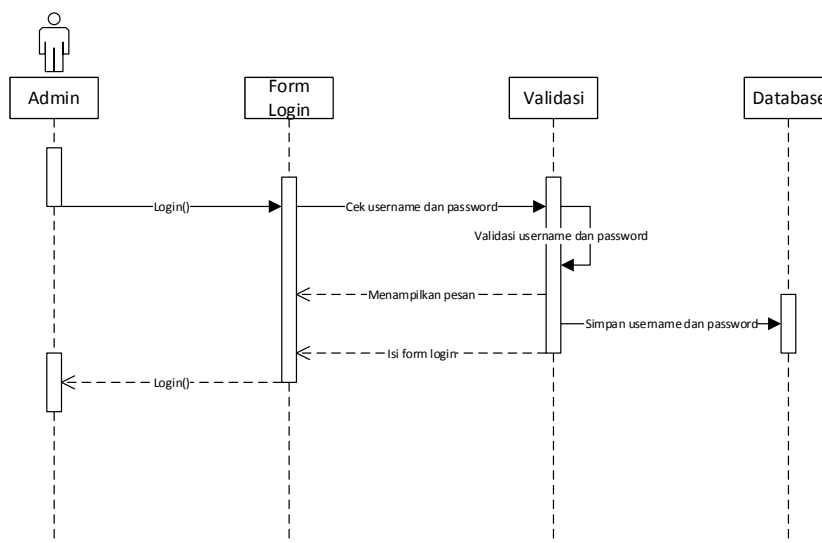
Class diagram membantu dalam visualisasi struktur kelas – kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap – tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, 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 decían, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat [24].



Gambar 2. 11 Class diagram

2.1.18.4 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 sequen sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk dimengerti [24].



Gambar 2. 12 Sequence diagram

