

BAB 2

LANDASAN TEORI

2.1 Profil Dinas Pangan Dan Pertanian Kota Bandung

Dinas Pertanian dan Ketahanan Pangan Kota Bandung di bentuk pada tanggal 15 April Tahun 2001, berdasarkan Perda Kota Bandung No. 26 tahun 2001 yang merupakan gabungan dari 3 dinas yaitu, Dinas Pertanian Tanaman Pangan, Dinas Peternakan dan Dinas Perikanan, oleh karena nya kewenangan Dinas Pertanian dan Ketahanan Pangan Kota Bandung meliputi kewenangan subsektor pertanian tanaman pangan, peternakan, perikanan, dan bidang perkebunan. Selanjutnya pada tahun 2010 berdasarkan Perda No. 13 tahun 2009 kembali bergabung Bagian Ketahanan Pangan menjadi salah satu Bidang di Dinas Pertanian sehingga nama Dinas berubah menjadi Dinas Pertanian dan Ketahanan Pangan Kota Bandung.

2.1.1 Logo Dinas Pangan Dan Pertanian Kota Bandung

Logo dari Dinas Pangan Dan Pertanian Kota Bandung dapat dilihat pada Gambar 2.1.



Gambar 2.1 Logo Dinas Pangan Dan Pertanian Kota Bandung

2.1.2 Visi Dan Misi

Dinas Pangan Dan Pertanian Kota Bandung mempunyai Visi dan Misi sebagai berikut :

2.1.2.1 Visi Dinas Pangan Dan Pertanian Kota Bandung

Dinas Pangan dan Pertanian Kota Bandung adalah “Terwujudnya Penyelenggaraan Pangan dan Pertanian yang Tangguh dan Unggul”.

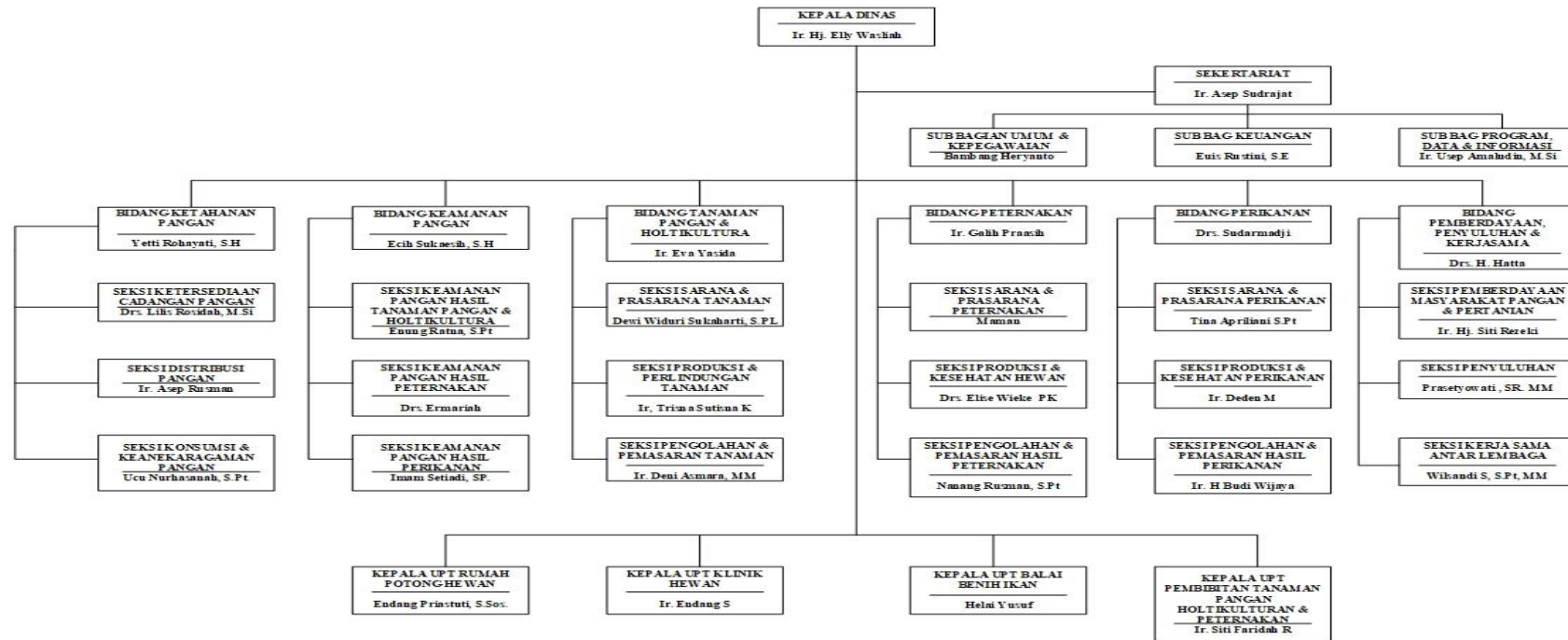
2.1.2.2 Misi Dinas Pangan Dan Pertanian Kota Bandung

Dinas Pangan dan Pertanian Kota Bandung adalah sebagai berikut:

1. Meningkatkan Penyelenggaraan Pangan.
2. Meningkatkan Pengawasan Keamanan Pangan Segar dan Penyakit Zoonosa.
3. Mengembangkan Potensi Sumberdaya Pertanian dan Perikanan Secara Berkelanjutan dan Berwawasan Lingkungan.
4. Meningkatkan Pelayanan di Bidang Pertanian dan Perikanan.

2.1.3 Struktur Organisasi

STRUKTUR ORGANISASI DINAS PANGAN DAN PERTANIAN KOTA BANDUNG



Gambar 2.2 Struktur Organisasi

2.2 Landasan Teori

Landasan teori merupakan penjelasan dari beberapa teori yang berkaitan dengan aplikasi mobile android yang akan dibangun. Landasan teori yang digunakan dalam pembangunan aplikasi monitoring kualitas buah berbasis *Mobile* Android dengan memanfaatkan teknologi Sensor Suhu dan Kamera, meliputi Profil Dinas Pangan dan Pertanian, Pengertian Aplikasi, Pengertian Gudang, Pengertian Monitoring, Android, Penjelasan Sensor, Suhu, Pengolahan Citra, Android Studio, UML, web service, MySQL, PHP.

2.2.1 Aplikasi

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari penggunanya. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan instruction set yang akan dijalankan oleh pemroses, yaitu berupa software. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. [3]. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada instructionset. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi mahasiswa atau aplikasi yang lain. Contoh- contoh aplikasi ialah program pemroses kata dan Web Browser. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app.

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (enterprise).
2. Perangkat lunak infrastruktur perusahaan.
3. Perangkat lunak informasi kerja.
4. Perangkat lunak media dan hiburan.
5. Perangkat lunak pendidikan.
6. Perangkat lunak pengembangan media.

7. Perangkat lunak rekayasa produk.

Aplikasi menjadi sebuah media yang dibutuhkan dalam melakukan penelitian ini. Yang akan dibuat dengan tujuan membantu petugas perpustakaan dan mahasiswa dalam kegiatan perpustakaan. Dengan manfaat dapat mengurangi antrian pendaftaran buku, pemeriksaan buku, pencarian buku, dan laporan petugas perpustakaan. Sehingga membuat pengelolaan perpustakaan menjadi lebih sistematis dan efisien.

2.2.2 Android



Gambar 2.3 Logo Android

Android merupakan sebuah sistem operasi telepon seluler dan komputer tablet layer sentuh (*touch screen*) yang berbasis linux. Namun seiring perkembangannya, Android berubah menjadi platform yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakangnya, yaitu Google. Google-lah yang mengakuisisi Android, kemudian membuatkan sebuah platform. [4]

Platform Android terdiri dari Sistem Operasi berbasis Linux, sebuah GUI (*Graphic User Interface*), sebuah Web Browser dan Aplikasi End-User yang dapat di download dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan oleh berbagai macam perangkat.

2.2.2.1 Sejarah Android

Pada Awalnya, Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan IT & Communication ; Andy Rubin, Rich Miner, Nick Sears dan Chris White. Menurut Rubin, Android Inc didirikan untuk mewujudkan mobile device yang lebih peka terhadap lokasi dan preferensi pemilik. Dengan kata lain, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya.

Saat ini, sistem operasi android menjadi pilihan yang menguntungkan bagi banyak vendor smartphone, karena memiliki biaya lisensi yang lebih murah dan sifatnya yang semi-open source. Selain itu, Android tentunya juga akan support dengan berbagai layanan dari Google. Dengan dukungan berbagai vendor smartphone, saat ini Android telah menumbangkan Apple sebagai sistem operasi paling banyak digunakan. Android kini telah menguasai 59% penjualan smartphone dan tablet di dunia.

2.2.3 Android SDK (Software Development Kit)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java.

2.2.4 ADT (Android Development Tools)

Android Development Tools adalah plugin yang di desain untuk IDE Eclipse yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan

memudahkan dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya.

2.2.5 Android Life Cycle

Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan.

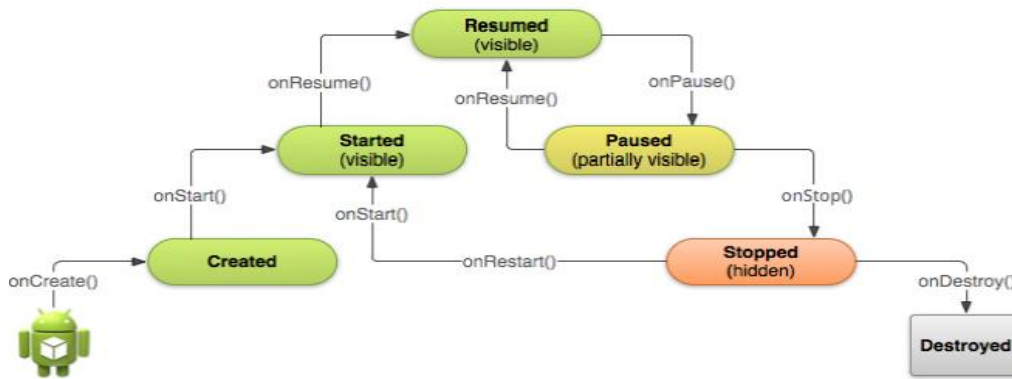
Tabel 2.1 Komponen Aplikasi Android 1

<i>Functionality</i>	<i>Java Base Class</i>	<i>Examples</i>
<i>Focused thing a</i>	<i>Activity</i>	<i>Edit a note, play a game</i>
<i>Background process</i>	<i>Service</i>	<i>Play music, update weather icon</i>
<i>Receive messages</i>	<i>BroadcastReceiver</i>	<i>Trigger alarm upon event</i>
<i>Store and retrieve data</i>	<i>ContentProvider</i>	<i>Open a phone contact</i>

Setiap aplikasi pasti menggunakan minimal satu dari komponen tersebut, akan tetapi terdapat beberapa komponen yang mengharuskan mencantumkan *specified permission* sebelum digunakan seperti komponen *Service*, *BroadcastReceiver*. *ContentProvider* .

Android memiliki paradigma pemrograman lain tidak seperti paradigma pemrograman biasa di mana aplikasi yang dijalankan pada fungsi *main()*, sistem android menjalankan kode dalam method *Activity* dengan menerapkan metode *callback* tertentu yang sesuai dengan tahap tertentu dari siklus hidup. Setiap aplikasi yang berjalan dalam sistem operasi android memiliki siklus hidup yang berbeda dengan aplikasi desktop atau web. Hal ini dikarenakan aplikasi mobile memiliki tingkat interupsi proses yang lumayan tinggi seperti ketika *handling panggilan* masuk aplikasi diharuskan menghentikan proses sementara. Penerapan siklus hidup

juga berguna untuk memastikan aplikasi tidak menghabiskan sumber daya baterai pengguna.



Gambar 2.4 Siklus Hidup Android

Terdapat beberapa state dalam siklus hidup android yang terjadi seperti diilustrasikan pada Gambar 2.4 Siklus Hidup Android ,akan tetapi hanya beberapa dari state tersebut yang menjadi statis diantaranya :

1. Resumed

Resumed terjadi ketika aplikasi berjalan setelah state paused . State ini akan menjalankan perintah program yang ditulis pada method `onResume()` (Google Inc,2014).

2. Paused

Dalam keadaan ini aktivitas yang terjadi dihentikan secara sementara tetapi masih terlihat oleh pengguna karena terdapat proses yang memiliki prioritas lebih tinggi seperti panggilan telepon. Aplikasi tidak dapat menjalankan perintah apapun ataupun menampilkan apapun dalam state ini (Google Inc, 2014).

3. Stopped

Dalam keadaan ini, aplikasi benar-benar tidak ditampilkan dan tidak terlihatoleh pengguna tetapi masih meninggalkan service dibackground (Google Inc, 2014).

State lain seperti Created dan Started bersifat sementara dan sistem dengan cepat menjalankan state berikutnya dengan memanggil metode life cycle callback berikutnya. Artinya, setelah sistem onCreate() dipanggil, dengan cepat sistem akan memanggil method onStart(), kemudian diikuti oleh onResume() (Schwarz, Dutson, Steele, & To, 2013).

2.2.6 Arsitektur Android

Secara garis besar Arsitektur Android dapat di jelaskan dan di gambarkan sebagai berikut :[5]

1. *Application dan Widgets*

Application dan Widgets adalah *layer* dimana *user* berhubungan dengan aplikasi saja, dimana biasanya user men-*download* aplikasi, melakukan instalasi dan menjalankan aplikasi.

2. *Application Frameworks*

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau member kemampuan untuk membangun aplikasi yang inovatif.

3. *Libraries*

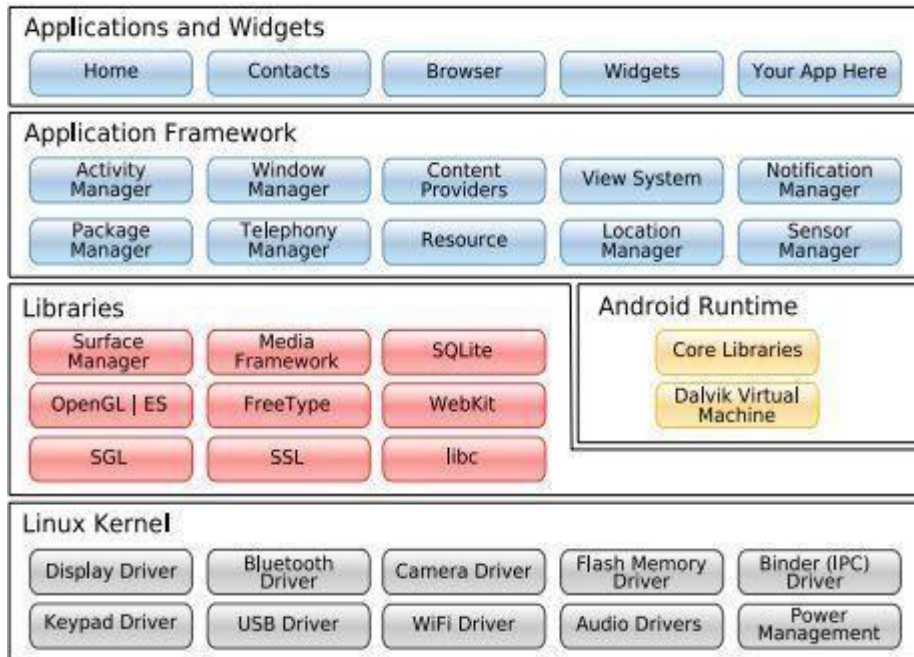
Libraries adalah *layer* dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya.

4. *Android Runtime*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux.

5. *Linux Kernel*

Linux Kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada.



Gambar 2.5 Arsitektur Android

2.2.7 Gudang

Gudang merupakan tempat penyimpanan material yang diperlukan untuk proses produksi, material tersebut akan terus disimpan hingga siap diproses sesuai dengan jadwal produksi atau order konsumen. [6] Adapun tujuan dari kegiatan penyimpanan material adalah :

1. Untuk menyeimbangkan antara kemampuan produksi dengan demand konsumen.
2. Untuk memberikan suatu customer service yang spesifik
3. Untuk menambah nilai pada produk

Aktivitas yang umumnya dilakukan berkaitan dengan penyimpanan material dalam gudang, adalah Receiving, Prepacking, Put-away, Storage, Order picking, Packaging, Sortation and accumulation, Packing and Shipping (Tompkins & Tanchoco, 1996). Dalam perancangan tata letak gudang terdapat beberapa prinsip yang umum dijadikan sebagai acuan, yaitu: Popularity, Similarity, Size, Characteristics dan Space utilization. Beberapa karakteristik komponen yang penting yaitu: Perishable materials (komponen yang mudah rusak), Oddly shaped

and crushable items (komponen bentuk khusus dan mudah rusak), Hazardous materials (komponen berbahaya), Security items (komponen dengan pengamanan khusus) dan Compatibility (kecocokan/kesesuaian).

2.2.8 Suhu(Temperatur)

Temperatur adalah suatu ukuran untuk tingkat panas suatu benda. Suhu suatu benda ialah keadaan yang menentukan kemampuan benda tersebut untuk mentransfer panas atau menerima panas, dari benda satu ke benda yang lain. Distribusi suhu di dalam atmosfer sangat bergantung terutama pada keadaan radiasi matahari, oleh sebab itu suhu udara selalu mengalami perubahan.

Karakteristik iklim pada permukaan bumi berbeda dari satu tempat ke tempat lain hal ini dipengaruhi oleh posisi relatif terhadap garis edar matahari (posisi lintang), keberadaan lautan, pola angin, bentuk permukaan daratan bumi, kerapatan vegetasi. Peredaran(revolution) bumi mengelilingi matahari dan perputaran (rotation) bumi pada sumbunya menyebabkan seluruh permukaan bumi secara bergantian dapat menerima radiasi matahari. Radiasi matahari mempengaruhi suhu rata-rata di masing-masing wilayah, semakin besar jumlah energi radiasi yang diterima suatu wilayah menyebabkan semakin tinggi suhu permukaan pada wilayah tersebut. Suhu udara akan berfluktuasi dengan nyata pada setiap periode 24 jam. Suhu udara maksimum tercapai beberapa saat setelah intensitas cahaya maksimum tercapai pada saat berkas cahaya jatuh tegak lurus, yakni tengah hari (Lakitan, 2002). [7]

2.2.9 Sensor

Sensor adalah alat yang dapat digunakan untuk mengukur, menganalisa, memantau suatu kondisi dan kemudian merespon terhadap perubahan di sekitarnya. Alat ini dapat ditemukan pada perangkat modern seperti smartphone dengan sistem operasi android, sebagai ponsel pintar yang memiliki sepaket teknologi canggih.

Sebagian besar smartphone kelas atas telah dilengkapi banyak sensor yang terintegrasi, mampu menganalisa data dengan presisi dan akurasi tinggi,

merespon berdasarkan gerakan, orientasi, dan berbagai kondisi lingkungan didekatnya. [8].

2.2.9.1 Kategori Tipe Sensor

Beberapa macam kategori tipe sensor yang umum:

1. Sensor Gerak atau Motion sensors.

Motion sensors adalah alat yang berfungsi untuk mengukur kekuatan akselerasi dan rotasi sepanjang tiga sumbu (X,Y,Z). Kategori ini meliputi kecepatan atau akselerasi, sensor gravitasi, gyroscope, dan rotasi vektor.

2. Sensor Lingkungan atau Environmental sensors.

Environmental sensors adalah alat yang berfungsi untuk mengukur berbagai parameter keadaan sekitar, seperti suhu udara dan tekanan, pencahayaan, dan kelembaban. Kategori ini termasuk barometers, photometers, dan thermometers.

3. Sensor Posisi atau Position sensors.

Position sensors adalah alat yang berfungsi untuk mengukur posisi fisik perangkat. Kategori ini meliputi sensor orientasi dan magnetometer.

2.2.9.2 Jenis Sensor

Berikut adalah jenis sensor pada smartphone android yang mampu menganalisa data dengan presisi dan akurasi tinggi:

1. Accelerometer adalah alat sensor yang digunakan oleh sistem

untuk mendeteksi orientasi suatu perangkat berdasarkan gerakan ke segala arah atau dengan menggoyangkan yang memungkinkan fitur untuk bertindak.

Fungsi: mengukur percepatan bahwa perangkat mengalami perubahan yang relatif sesuai dengan tiga sumbu XYZ atau kanan, kiri, atas, bawah, dan datar.

Level Kegunaan: penting.

2. Gyroscope adalah sensor yang digunakan untuk mendeteksi rotasi atau perputaran suatu perangkat berdasarkan gerakan. Alat ini bekerja sama dengan accelerometer untuk fitur seperti memiringkan atau memutar ponsel.

Fungsi: memberikan informasi orientasi tapi dengan lebih presisi sampai perputaran 360 derajat.

Level Kegunaan: cukup penting

3. Proximity adalah sensor yang digunakan pada saat percakapan atau panggilan berdasarkan jarak kedekatan perangkat dan pengguna. Jika perangkat didekatkan ke telinga, secara otomatis mematikan layar atau mengunci layar dan mencegah perintah dari sentuhan yang tidak diinginkan.

Fungsi: mengukur dan menganalisa jarak kedekatan.

Level Kegunaan: cukup penting.

4. Ambient Light Sensor adalah alat yang digunakan untuk mengatur kecerahan layar secara otomatis berdasarkan cahaya sekitar.

Sensor ini dapat mengukur seberapa terang cahaya sekitar sehingga dapat melestarikan ketahanan baterai. Sistem menggunakan data dari alat ini untuk menyesuaikan kecerahan layar secara otomatis.

Fungsi: mengatur dan menyesuaikan kecerahan cahaya pada layar.

Level Kegunaan: cukup penting.

5. Digital Compass atau Magnetometer adalah sensor yang dapat digunakan untuk mengukur magnetisasi dari bahan magnetik atau untuk mengukur kekuatan arah medan magnet pada titik dalam ruang atau sebagai kompas.

Fungsi: mendeteksi medan magnet seperti kutub utara, kompas, mendeteksi terkait penggunaan logam.

Level Kegunaan: cukup penting.

6. Barometer adalah alat atau instrumen ilmiah yang digunakan dalam meteorologi untuk mengukur tekanan atmosfer. Sensor ini memiliki kecenderungan agar dapat memperkirakan perubahan cuaca dalam jangka pendek. Banyak pengukuran tekanan udara yang digunakan dalam analisis cuaca pada permukaan untuk membantu menemukan palung permukaan, sistem tekanan tinggi dan batas-batas frontal.

Fungsi: mengukur tekanan udara atau atmosfer, sehingga bisa mendapatkan perkiraan secara umum tentang apa yang akan terjadi dengan apakah tekanan udara naik atau turun, meningkatkan hasil elevasi GPS, karena perangkat dapat terpengaruh oleh tekanan atmosfer.

Level Kegunaan: cukup penting

7. Termometer adalah alat yang digunakan untuk mengukur suhu atau gradien suhu lingkungan. Beberapa alat sensor mengkonversi perubahan fisik ini menjadi nilai numerik, misalnya skala atau nilai yang ditandai pada termometer raksa pada kaca.

Fungsi: mengukur suhu udara, mengontrol dan mengatur proses dalam studi cuaca, dalam kedokteran, dan penelitian ilmiah.

Level Kegunaan: cukup penting.

8. Pedometer adalah sensor gerak yang digunakan untuk menghitung jumlah langkah yang telah dilakukan pengguna. Data tersebut biasanya diperoleh dari analisa sensor accelerometer, tetapi pedometer jauh lebih akurat dan hemat daya.

Fungsi: mendeteksi gerakan, mengukur setiap gerak langkah dari jarak yang ditempuh dalam satuan panjang seperti kilometer atau mil.

Level Kegunaan: cukup penting.

9. Fingerprint Sensor adalah Alat pemindai sidik jari. Fitur ini menawarkan pengguna membuka cepat tampilan layar, sementara

itu sambil memberikan dinding keamanan yang stabil untuk setiap pengintai yang mungkin ingin menggali setiap data pada perangkat.

Fungsi: Pemindaian sidik jari, otentikasi sidik jari.

Level Kegunaan: cukup penting.

2.2.10 UML

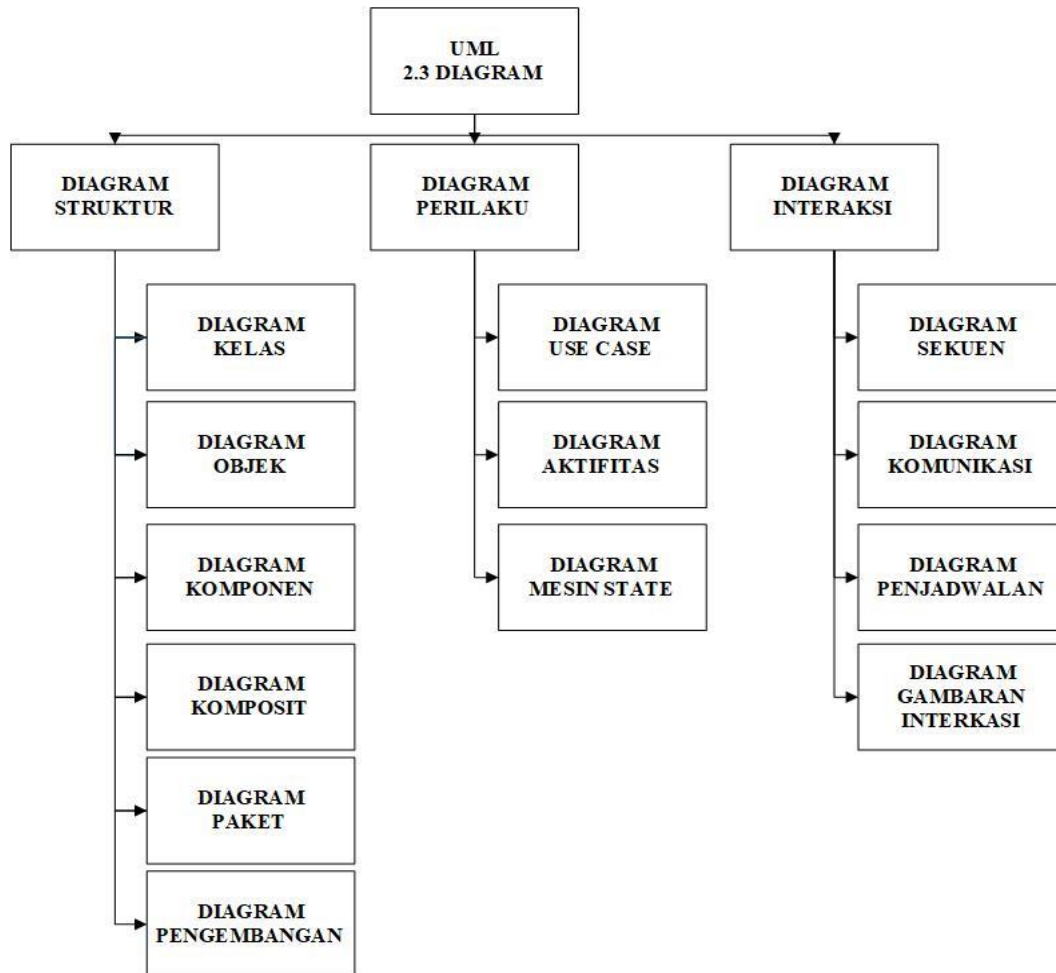
UML adalah alat perancangan sistem yang berorientasi pada objek. UML diagram memiliki tujuan untuk membantu tim pembangunan proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Diagram UML diklasifikasikan menjadi tiga kategori; struktur, perilaku, interaksi. Diagram struktur untuk menggambarkan suatu struktur statis dari sistem yang di modelkan. Diagram struktur yang akan dibahas dalam penulisan ini adalah class diagram, sedangkan diagram perilaku untuk memvisualisasikan, menspesifikasikan, membangun dan mendokumentasikan aspek dinamis dari sistem. Diagram perilaku yang akan dibahas pada penulisan ini adalah use case diagram, dan *activity diagram*. Lalu yang terakhir yaitu diagram interaksi untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram interaksi yang akan dibahas pada penulisan ini adalah sequence diagram.

UML menjadi metode yang digunakan untuk membangun sistem adalah OOAD(Object Oriented Analysis dan design). [8]

2.2.10.1 UML Diagram

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.6



Gambar 2.6 UML Diagram

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

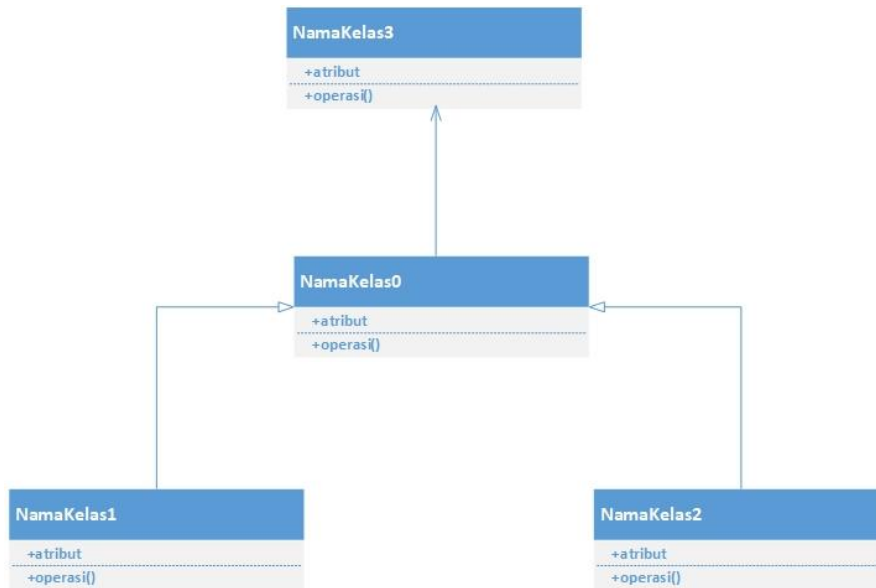
1. Diagram Struktur yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek statis dari sistem yang dimodelkan.
2. Diagram Perilaku yaitu diagram untuk memvisualisasikan, mendokumentasi, menspesifikasikan, dan membangun aspek dinamis dari sistem yang dimodelkan.
3. Diagram Interaksi yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.10.2 Class Diagram

Diagram Kelas menunjukkan sekumpulan kelas, interface dan kolaborasi dan keterhubungan. Diagram kelas ditujukan untuk pandangan statik terhadap sistem. Kelas memiliki apa yang disebut nama, atribut, metode atau operasi.

1. Nama kelas harus unik. Nama akan menjadi identifier di program, seharusnya sedini mungkin dipilih nama yang memenuhi aturan (semua) bahasa pemrograman.
2. Atribut adalah properti bernama di kelas yang mendeskripsikan *range* nilai yang dipunyai instan kelas. Kelas dapat mempunyai sejumlah atribut atau tidak sama sekali.
3. Operasi atau metode adalah implementasi layanan yang dapat diminta pada sembarang objek kelas untuk mempengaruhi perilaku sistem. Kelas dapat mempunyai sejumlah operasi atau tidak sama sekali.

Berikut Contoh *Class Diagram* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Contoh Class Diagram

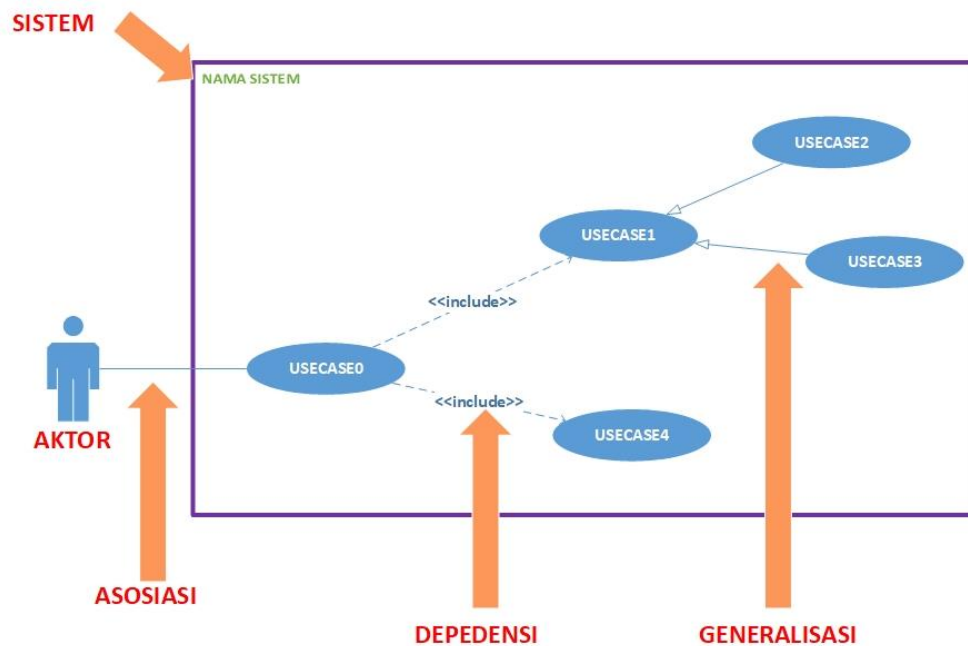
2.2.10.3 Use Case Diagram

Use case Diagram merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram *use case* menunjukkan sekumpulan *use case*, aktor, dan hubungannya. Diagram *use case* adalah penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram-diagram *use case* merupakan pusat pemodelan perilaku sistem, subsistem, dan kelas. Diagram *use case* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Ada empat hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case* [8].

1. Sistem yaitu sesuatu yang hendak kita bangun.
2. Asosiasi adalah komunikasi antar aktor dan usecase yang berpartisipasi pada usecase atau usecase memiliki interaksi dengan aktor.
3. Depedensi yaitu mengidentifikasi hubungan antara dua usecase dimana yang satu memanggil yang lain.
4. Generalisasi hubungan antara dua buah usecase dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

5. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
6. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Contoh *Use Case Diagram*

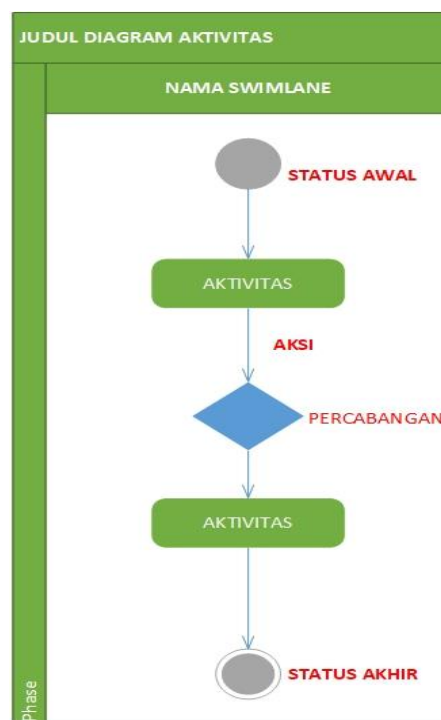
2.2.10.4 *Activity Diagram*

Diagram aktivitas adalah diagram flowchart yang diperluas yang menunjukkan aliran kendali satu aktivitas ke aktivitas lain di sistem. Diagram aktivitas ini digunakan untuk memodelkan aspek dinamis sistem. Diagram aktivitas mendeskripsikan aksi-aksi dan hasilnya. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokkan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut Contoh *Activity Diagram* dapat dilihat pada Gambar 2.9.



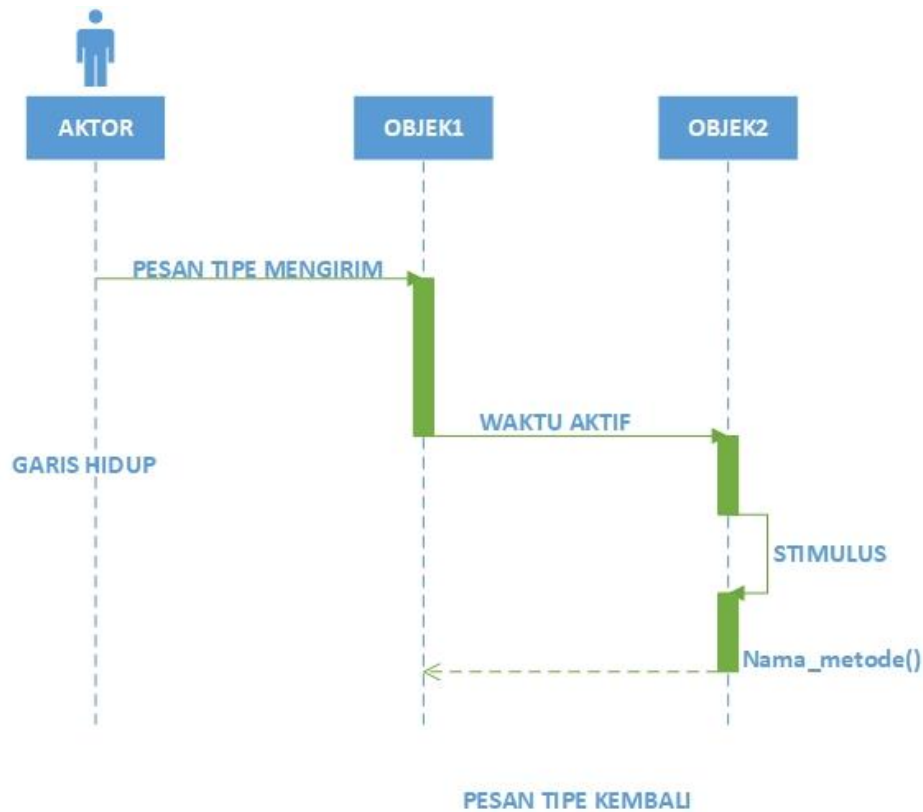
Gambar 2.9 Contoh *Activity Diagram*

2.2.10.5 *Sequence Diagram*

Diagram sekuen menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem. Diagram ini menekankan pada sisi basis keberurutan waktu dari pesan-pesan yang terjadi. Diagram sekuen menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya

menunjukkan barisan kejadian, bukan perwaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan terjadi.

Berikut Contoh *sequence diagram* dapat dilihat pada gambar 2.10 :



Gambar 2.10 Contoh *Sequence Diagram*

2.2.11 OOAD (Object Oriented Analysis System)

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang berlakunya terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak di bangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek. [8]

Konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai Teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

Terdapat beberapa konsep dalam OOAD, yaitu:

1. Objek (*object*)

- A. Objek adalah benda secara fisik dan konseptual yang ada di sekitar kita. Sebuah objek memiliki keadaan sesaat yang disebut state.
- B. State dari sebuah objek adalah kondisi dari objek atau himpunan keadaan yang menggambarkan objek tersebut. State dinyatakan dengan nilai dari atribut objeknya.
- C. Atribut adalah nilai internal suatu objek yang mencerminkan karakteristik objek, kondisi sesaat, koneksi dengan objek lain dan identitas.
- D. *Behaviour* (perilaku objek) mendefinisikan bagaimana sebuah objek bertindak dan memberi reaksi. *Behaviour* ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan oleh objek tersebut, yang dicerminkan oleh interface, service, dan method dari objek tersebut.
- E. Interface adalah pintu untuk mengakses service dari objek
- F. Service adalah fungsi yang dapat dikerjakan oleh sebuah objek.
- G. Method adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut.

2. Kelas (*class*)

Class adalah himpunan objek yang sejenis yaitu mempunyai sifat atau atribut, perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. Class adalah abstraksi dari objek

dalam dunia nyata. Class menetapkan spesifikasi perilaku dan atribut dari objek tersebut.

3. Pengujian Kotak Hitam (black box)

Sebuah objek adalah Kotak hitam. Konsep ini menjadi dasar implementasi objek. Dalam operasi OO hanya developer yang dapat memahami detail proses yang ada didalam Kotak tersebut, sedangkan mahasiswa tidak perlu mengetahui apa yang dilakukan yang penting mereka dapat menggunakan objek untuk memproses kebutuhan mereka. Kotak hitam berisi kode dan data.

- A. Encapsulation, yaitu proses menyembunyikan detail implementasi sebuah objek. Untuk mengakses data objek tersebut adalah melalui interface. Untuk berkomunikasi dengan objek digunakan message.
- B. Message adalah permintaan agar objek menerima untuk membawa metode yang ditunjukkan oleh perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender)

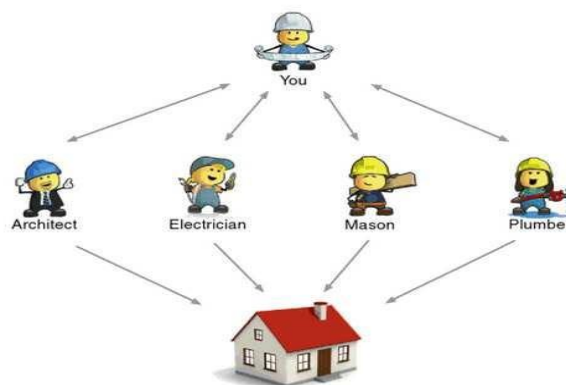
4. Asosiasi dan Agregasi

Asosiasi adalah hubungan yang mempunyai makna antara sejumlah objek. Asosiasi digambarkan dengan sebuah garis penghubung diantara objeknya. Contohnya : Asosiasi karyawan dengan unit kerja. Setiap karyawan bekerja di satu unit kerja, sedangkan unit kerja dapat memiliki beberapa karyawan.

Agregasi adalah bentuk khusus sebuah asosiasi yang menggambarkan seluruh bagian pada suatu objek merupakan bagian dari objek yang lain. Contohnya : Kopling dan piston adalah bagian dari mesin, sedangkan mesin, roda, body merupakan bagian dari sebuah mobil.

2.2.12 API (*Application Programming Interface*)

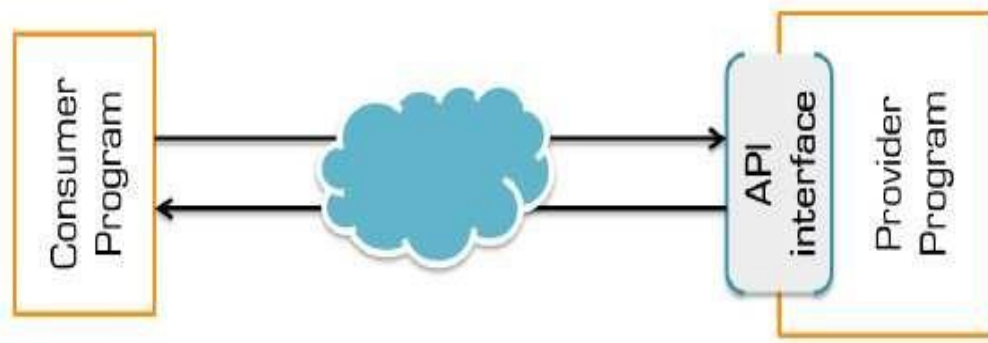
Menurut Sugiarto (2013:3) API atau Application Programming Interface merupakan suatu dokumentasi yang terdiri dari interface, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat. Dengan adanya API ini, maka memudahkan programmer untuk membongkar suatu software untuk kemudian dapat dikembangkan atau diintegrasikan dengan perangkat lunak lain. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan programmer menggunakan *system function*. [9]



Gambar 2.11 Analogi API pada Pembangunan Rumah

(Sumber: *API Design for C*, Reddy, 2011)

Interface pada *software* merupakan suatu *entry points* yang digunakan untuk mengakses seluruh *resources* yang terdapat di dalam *software* tersebut. Dengan adanya API, maka terdapat aturan bagaimana *software* dapat berinteraksi dengan *software* lain untuk mengakses *resources* melalui *interface* yang telah tersedia.



Gambar 2.12 Skema Konektivitas API Antar *Software*

(Sumber: *What is an API?*, 3Scale Networks, 2011)

Secara struktural, API merupakan spesifikasi dari suatu data *structure, objects, functions*, beserta parameter-parameter yang diperlukan untuk mengakses *resource* dari aplikasi tersebut. Seluruh spesifikasi tersebut membentuk suatu interface yang dimiliki oleh aplikasi untuk berkomunikasi dengan aplikasi lain, dan API dapat digunakan dengan berbagai bahasa programming, ataupun hanya dengan menggunakan URL (*Uniform Resource Locator*) yang telah disediakan oleh suatu website.

API dapat diklasifikasikan menjadi beberapa kategori, hal ini dilihat dari abstraksi apa yang dideskripsikan di dalam sistem. Kategori-kategori ini diantaranya:

Tabel 2.2 Kategori API 1

Kategori API	Deskripsi	Contoh
<i>Operating System</i>	API yang digunakan untuk fungsi dasar yang dapat dilakukan oleh komputer. Seperti proses I/O, eksekusi program.	API for MS Windows
<i>Programming Languages</i>	API yang digunakan untuk memperluas kapabilitas dalam melakukan eksekusi terhadap suatu bahasa pemrograman.	Java API
<i>Application Services</i>	API yang digunakan untuk mengakses data dan layanan yang disediakan dari suatu aplikasi.	API for mySAP (BAPI/ <i>Business Application</i>)
<i>Infrastructure Services</i>	Digunakan untuk mengakses infrastruktur dari suatu komputer. Infrastruktur disini adalah komputer beserta <i>peripheral</i> seperti <i>storage</i> , aplikasi, dan lain-lain.	Amazon EC2 (<i>Elastic Compute Cloud</i>) untuk akses untuk <i>virtual computing</i> dan Amazon S3 (<i>Simple Storage</i>)
<i>Web Services</i>	API yang digunakan untuk mengakses <i>content</i> dan layanan yang disediakan oleh suatu <i>web application</i> .	<i>Facebook Graph API</i> yang digunakan untuk mengakses informasi yang dapat dibagikan.

2.2.13 Pengolahan Citra

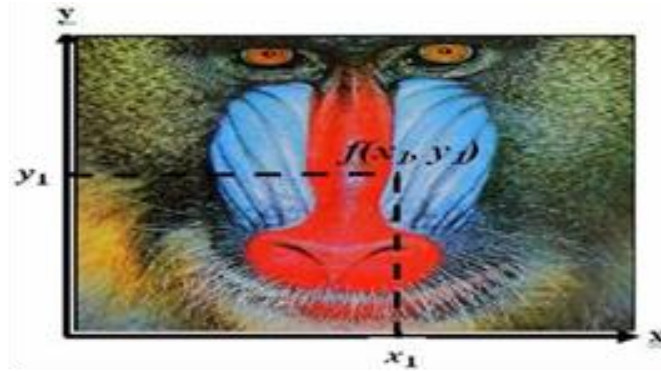
Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan. [10]

1. Citra Analog

Citra Analog adalah citra yang bersifat continue, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak dikertas foto, lukisan, pemandangan alam, hasil CT scan, gambar-gambar yang terekam dalam pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak dapat diproses dikomputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses dikomputer, konversi analog ke digital harus dilakukan terlebih dahulu. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, webcam, CT scan, sensor rontgren untuk foto thorax, sensor gelombang pendek pada sistem radar, sensor ultrasound pada sistem USG, dan lain-lain.

2. Citra Digital

Pengolahan citra digital dimulai sekitar awal tahun 1920-an dari dunia pemberitaan media cetak, di mana sebuah citra dikirim melalui kabel bawah laut dari London menuju ke New York. Proses ini menghemat waktu pengiriman dari seminggu menjadi kurang dari tiga jam. Sebelum dikirim, citra terlebih dahulu dikodekan dan setelah diterima citra direkonstruksi ulang. Citra digital adalah citra yang dapat diolah oleh komputer. Yang disimpan dalam memori komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel. Karena berbentuk data numerik, maka citra digital dapat diolah dengan komputer. Citra digital dapat didefinisikan sebagai fungsi dua variabel $f(x,y)$, dimana x dan y adalah koordinat



Gambar 2.13 Contoh Citra Digital

2.2.14 Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut: [12]

1. Menggunakan Gradle-based build Sistem yang fleksibel.
2. Bisa mem-build multiple APK.
3. Template support untuk Google Services dan berbagai macam tipe perangkat.
4. Layout editor yang lebih bagus.
5. Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
6. Import library langsung dari Maven repository.

Adapun beberapa persyaratan sistem pada windows untuk menggunakan Android Studio adalah sebagai berikut:

1. Microsoft Windows 8/7/Vista/2003 (32 atau 64 bit).
2. RAM minimum 2 GB, Rekomendasi 4 GB RAM.
3. Disk kosong 500 MB
4. Sedikitnya 16 GB untuk Android SDK, Emulator Sistem images, and caches.
5. Java Development Kit (JDK) 7 atau yang lebih tinggi.
6. Resolusi minimum 1280 x 800

2.2.15 Web Service

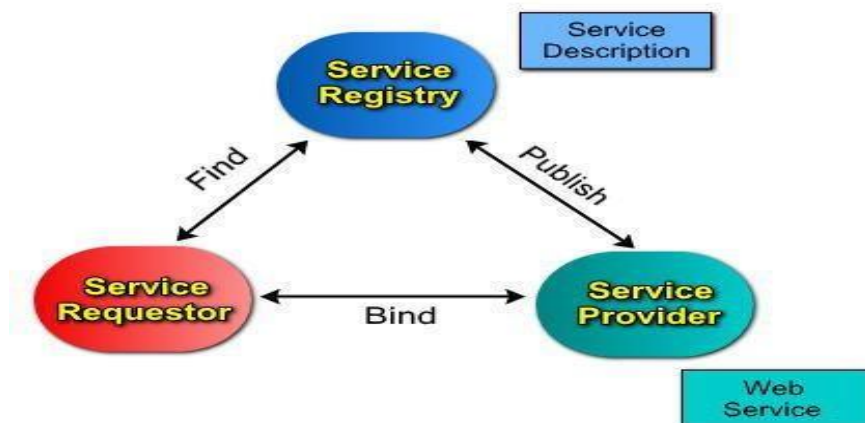
Web service adalah suatu 41rotoc perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar 41rotoc pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada 41rotoc lain, sehingga 41rotoc lain dapat berinteraksi dengan 41rotoc tersebut melalui layanan-layanan (service) yang disediakan oleh suatu 41rotoc yang menyediakan web service. Web service menyimpan data informasi dalam format XML, sehingga data ini dapat diakses oleh 41rotoc lain walaupun berbeda platform, 41rotoc operasi, maupun 41rotoc compiler.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu 42rotoc operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan 42rotocol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

2.2.15.1 Arsitektur *Web service*



Gambar 2.14 Arsitektur *Web Service*

Pada gambar diatas, ada tiga komponen yang membuat *web service* berjalan. Ketiga komponen itu adalah:

1. ***Service provider***, merupakan pemilik *Web service* yang berfungsi menyediakan kumpulan operasi dari *Web service*.
2. ***Service requestor***, merupakan aplikasi yang bertindak sebagai klien dari *Web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.
3. ***Service registry***, merupakan tempat dimana *Service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, *Service registry* bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.

2.2.15.2 Jenis-jenis *Web Service*

Jenis *web service* dapat dibagi menjadi dua, yaitu REST dan SOAP.

1. *Representational State Transfer* (REST)

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi *Client*. [13]

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi

pada *resourcenya* itu, sedangkan aplikasi lowongan pekerjaan membutuhkan pemanggilan metode yang bisa dikerjakan terhadap kumpulan *resource* data lowongan. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API (*Application Program Interface*), YouTube API, Amazon API.

2. *Simple Object Access Protokol* (SOAP)

SOAP merupakan protokol standar yang ringan dan ditujukan untuk pertukaran informasi dalam desentralisasi sistem. Protokol ini melakukan pemanggilan metode dalam bahasa XML (*Extensible Markup Language*). Namun, selain itu juga kita dapat menggunakan format pertukaran data lainnya yaitu JSON (*JavaScript Object Notation*). Dengan demikian, SOAP adalah suatu mekanisme yang sederhana untuk melakukan pertukaran struktur dan tipe informasi dalam lingkungan yang tersebar dan terdistribusi menggunakan XML/JSON. Dokumen SOAP yang digunakan untuk melakukan *request* disebut SOAP *request* sedangkan dokumen SOAP yang diperoleh dari web service disebut dengan SOAP *response*.

2.2.16 PHP

Menurut Anhar, ST (2010), “PHP singkatan dari Personal Hypertext Preprocessor yaitu bahasa pemrograman web server-side yang bersifat open source. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server. PHP adalah script yang digunakan untuk membuat halaman website yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. [14]

PHP pertama kali dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, namun sekarang di ambil oleh oleh The PHP Group. Pada awalnya PHP adalah singkatan dari Personal Home Page, namun dalam perkembangannya, di ubah menjadi PHP: Hypertext Preprocessor, sebuah kepanjangan rekursif.

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa ServerSideScripting. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya web server.PHP ini bersifat opensource sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi Windows maupun Linux.PHP juga dibangun sebagai modul pada web apache dan sebagai binary yang dapat berjalan sebagai CGI. Pada pembangunan aplikasi pelaporan bencana kebakaran Kota Bandung khususnya halaman backend menggunakan script PHP untuk membuat halaman pelaporan. Berikut contoh skrip dasar PHP bisa dilihat pada gambar 2.19:

```
<html>
<head>
<title>Contoh Kode PHP </title>
</head>
<body>
<?php
echo ("Ini adalah Kode PHP yang diletakan di antara kode HTML");
?>
</body>
</html>
```

Gambar 2.15 Contoh Skrip PHP

Menurut Vikram Vaswani (2004), PHP memiliki beberapa keunggulan, antara lain[10] :

1. Kinerja

Script yang ditulis dalam PHP mengeksekusi lebih cepat dibandingkan yang ditulis dalam bahasa script lain.

2. Portabilitas

PHP tersedia untuk UNIX, Microsoft Windows, Mac OS, dan OS/2.PHP merupakan program portable antar platform. Kemampuan untuk melakukan cross-platform merupakan salah satu keunggulan bagi lingkungan perusahaan yang multiplatform.

3. Kemudahan dalam penggunaan

PHP adalah bahasa pemrograman yang sangat canggih dan dilengkapi dengan 5000 fungsi. Hal ini merupakan salah satu alasan PHP disukai sebagai alat prototyping untuk aplikasi berbasis web.

4. Open Source

PHP merupakan bahasa opensource dan tersedia secara bebas di web serta dapat digunakan tanpa biaya lisensi.

2.2.17 MySQL

MySQL merupakan *software database* yang paling populer di kalangan para pengembang aplikasi *database*. MySQL juga adalah suatu perangkat lunak yang menganut atau mengimplementasikan model basis data relasional maka MySQL disebut sebagai *Relational Database Management System (RDBMS)*. MySQL merupakan *software RDBMS* (atau *server database*) yang dapat mengelola *database* dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user (multi-user)*, dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*) [15].

Saat ini, MySQL banyak digunakan di berbagai kalangan untuk melakukan penyimpanan dan pengolahan data, mulai dari kalangan akademis sampai ke industri, baik industri kecil, menengah, maupun besar. Lisensi MySQL terbagi menjadi dua. Anda dapat menggunakan MySQL sebagai produk *open source* dibawah GNU *General Public License* (gratis) atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis masih dapat digunakan dengan baik. Beberapa contoh aplikasi yang menggunakan MySQL adalah Joomla, Wordpress, MyBB, phpBB, dan masih banyak lagi

2.2.18 Metode Pengujian Sistem

Sebuah perangkat lunak perlu di jaga kualitasnya bahwa kualitas bergantung kepada kepuasan pelanggan (*customer*). Kualitas perangkat lunak perlu dijaga untuk keperluan sebagai berikut : [16]

1. Agar dapat bertahan hidup di dunia bisnis perangkat lunak.
2. Dapat bersaing dengan perangkat lunak yang lain.
3. Penting untuk pemasaran global (*global marketing*)

4. Mengefektifkan biaya agar tidak banyak membuang perangkat lunak karena kegagalan pemasaran atau kegagalan produksi.
5. Mempertahankan pelanggan dan meningkatkan keuntungan

Metode pengujian sistem terdiri dari Pengujian *black box* dan *beta* yang dilakukan untuk mengetahui efektifitas dari perangkat lunak (*software*) yang digunakan.

2.2.18.1 Pengujian *Black Box*

Pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian ini dianalogikan seperti melihat suatu kotak hitam yang hanya bisa dilihat penampilan luarnya saja, tanpa tahu ada apa dibalik bungkus hitamnya. *Black box* testing melakukan evaluasi untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi tidak benar atau hilang.
2. Kesalahan *interface* atau antarmuka.
3. Kesalahan dalam struktur data atau akses *database eksternal*.
4. Kesalahan kinerja atau perilaku dan kesalahan inisialisasi dan terminasi.

2.2.18.2 Skala *Likert*

Skala likert adalah suatu skala psikometrik yang digunakan dalam kuesioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Rensis Likert telah mengembangkan sebuah skala untuk mengukur sikap masyarakat di tahun 1932 yang sekarang terkenal dengan nama skala Likert. Skala Likert ini merupakan skala yang dapat dipergunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang mengenai suatu gejala atau fenomena .

Dalam skala Likert terdapat dua bentuk pernyataan yaitu pernyataan positif yang berfungsi untuk mengukur sikap positif, dan pernyataan negative

yang berfungsi untuk mengukur sikap negative objek. Skor pernyataan positif dimulai dari :

1. Untuk sangat tidak setuju (TS).
2. Untuk tidak setuju (KS).
3. Untuk ragu-ragu (CS).
4. Untuk setuju (S).
5. Untuk sangat setuju (SS).

Skor pernyataan negative dimulai dari 1 untuk sangat setuju (SS), 2 untuk setuju (S), 3 untuk raguragu (CS), 4 untuk tidak setuju (KS), dan 5 untuk sangat tidak setuju (TS). Beberapa menghilangkan option “Ragu-ragu” dalam instrument untuk memudahkan dalam melihat angket yang responden isikan. Skala Likert digunakan untuk mengukur kesetujuan dan ketidaksetujuan seseorang terhadap sesuatu rencana program, pelaksanaan program ataupun tingkat keberhasilan suatu program.

Kekurangan Skala Likert :

1. Karena ukuran yang digunakan adalah ukuran ordinal, skala Likert hanya dapat mengurutkan individu dalam skala, tetapi tidak dapat membandingkan berapa kali satu individu lebih baik dari individu yang lain.
2. Kadangkala total skor dari individu tidak memberikan arti yang jelas, karena banyak pola respons terhadap beberapa item akan memberikan skor yang sama. Adanya kelemahan di atas sebenarnya dapat dipikirkan sebagai error dari respons yang terjadi.

Kelebihan Skala Likert :

1. Mudah dibuat dan di terapkan.
2. Skala Likert lebih mudah membuatnya dibanding lain seperti skala Thurstone.
3. Terdapat kebebasan dalam memasukan pertanyaan- pertanyaan, asalkan sesuai dengan konteks permasalahan yang diteliti.

4. Jawaban suatu item dapat berupa alternative, sehingga informasi mengenai item tersebut diperjelas.
5. Reliabilitas pengukuran bisa diperoleh dengan jumlah item tersebut diperjelas.

