# BAB II
# TINJAUAN PUSTAKA

## II.1 Profil Perusahaan

### II.1.1 Perusahaan Daerah Kebersihan Kota Bandung

Perusahaan Daerah Kebersihan (PD Kebersihan) Kota Bandung adalah lembaga yang dibentuk oleh Pemerintah Kota Bandung dengan fungsi dan tugas pokok sebagai unit usaha dalam menyelenggarakan pelayanan pengelolaan sampah Kota Bandung. Perusahaan ini didirikan pada tahun 1985 dan berbentuk Badan Usaha Milik Daerah (BUMD) merupakan hasil alih status dari Dinas Kebersihan Kota Bandung sehingga seluruh permodalan dari asset yang dipisahkan dari asset Pemerintah Kota Bandung.

Perubahan status dari Dinas Kebersihan menjadi Perusahaan Daerah Kebersihan dilatarbelakangi oleh sebuah tuntutan peningkatan pelayanan dan penanggulangan beban permasalahan sampah kota yang cukup berat terutama dari segi kebutuhan jumlah prasarana dan sarana dan jumlah tenaga kerjanya.

PD Kebersihan Kota Bandung didirikan pada tahun 1985 sesuai dengan Peraturan Daerah Kotamadya Daerah Tingkat II Bandung nomor 02/-PD/1985, tentang Pembentukan Perusahaan Kebersihan Kotamadya Daerah Tingkat II Bandung.

Guna kelancaran dan ketertiban dalam melaksanakan kegiatannya PD. Kebersihan Kota Bandung berpedoman pada beberapa peraturan yang dapat mendukung legalitas serta mengatur kelembagaan, diantaranya adalah [1]:

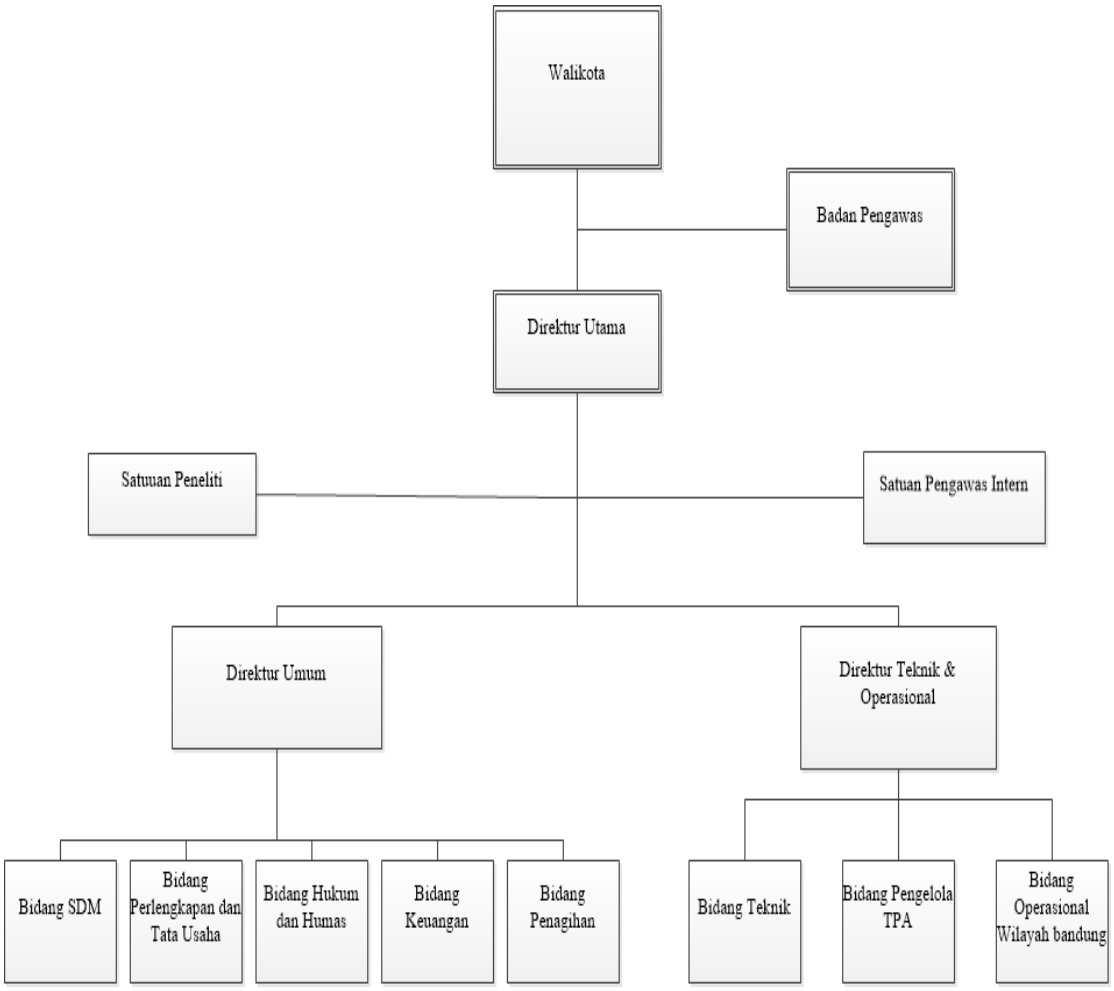
1. Peraturan Daerah Kotamadya Daerah Tingkat II Bandung nomor 02/-PD/1985 tentang Pembentukan Perusahaan Kebersihan Kotamadya Daerah Tingkat II Bandung.
2. Peraturan Daerah Kota Bandung Nomor 27 tahun 2001 tentang Pengelolaan Kebersihan Kota Bandung.
3. Keputusan Walikota Bandung Nomor 644 tahun 2002 tentang Tarif Jasa Kebersihan di Kota Bandung.

4. Peraturan Walikota no. 101 tahun 2006 tentang Susunan Organisasi dan Tata Kerja Perusahaan Daerah Kebersihan Kota Bandung.

Susunan organisasi dan manajemen PD. Kebersihan Kota Bandung mengacu pada Surat Keputusan Walikotamadya Daerah Tingkat II Bandung Nomor 101 tahun 2006 tentang Susunan Organisasi dan Tata Kerja Perusahaan Daerah Kebersihan Kotamadya Daerah Tingkat II Bandung. Perusahaan Daerah Kebersihan Kota Bandung dipimpin oleh tiga orang direksi yaitu:

1. 1 orang Direktur Utama
2. 1 orang Direktur Umum
3. 1 orang Direktur Teknik dan Operasi

Berikut ini adalah gambar struktur organisasi PD Kebersihan Kota Bandung dapat dilihat pada gambar II.1 berikut:




**Gambar II.1 Struktur Organisasi**

### II.1.2 Fungsi Perusahaan Daerah Kebersihan Kota Bandung

Perusahaan Kebersihan Kota Bandung memiliki beberapa fungsi yaitu:

1. Perumusan kebijakan dan strategi pengelolaan kebersihan dan usaha jasa kebersihan di bidang persampahan sejalan dengan visi dan misi Kota Bandung.
2. Penyelenggaraan pengelolaan kebersihan di bidang persampahan kota meliputi penyapuan, pengumpulan, pengangkutan, pembuangan dan pengolahan akhir.
3. Penyelenggaraan usaha jasa pelayanan kebersihan di bidang persampahan.

### II.1.3 Visi dan Misi Perusahaan Daerah Kebersihan Kota Bandung

Terwujudnya Kota Bandung Bersih dari Sampah Melalui Pengembangan Sistem Pengelolaan Sampah Ramah Lingkungan dan Berkelanjutan adalah visi Perusahaan Daerah Kebersihan Kota Bandung, Adapun misi dari Perusahaan Daerah Kebersihan Kota Bandung adalah :

1. Mengembangkan kemampuan sumber daya manusia dengan basis kompetensi pengelolaan sampah kota.
2. Mengembangkan sistem pengelolaan sampah dengan basis teknologi tepat guna dan mengarah kepada teknologi tinggi sesuai dengan tuntutan perkembangan.
3. Mengembangkan sistem pembiayaan pengelolaan sampah yang mampu mendukung penyelenggaraan pelayanan secara optimal.
4. Mengembangkan sistem pengelolaan sampah dengan pola kemitraan dengan masyarakat dan swasta.

## II.2 Landasan Teori

Subbab ini berisikan teori-teori yang digunakan untuk analisis dan implementasi pada pembangunan perangkat lunak *trash organizer.*

### II.2.1 Sampah

Pada prinsipnya sampah dibagi menjadi sampah padat, sampah cair dan sampah dalam bentuk gas. Sampah padat dapat dibagi menjadi beberapa jenis yaitu :

1. Berdasarkan zat kimia yang terkandung didalamnya:

   a. Sampah anorganik misalnya : logam-logam, pecahan gelas, dan plastik.
   b. Sampah organik misalnya : sisa makanan, sisa pembungkus dan sebagainya.
2. Berdasarkan dapat tidaknya dibakar:
   a. Mudah terbakar misalnya : kertas, plastik, kain, kayu.
   b. Tidak mudah terbakar misalnya : kaleng, besi, gelas.
3. Berdasarkan dapat tidaknya membusuk:
   a. Mudah membusuk misalnya : sisa makanan, potongan daging.
   b. Sukar membusuk misalnya : plastik, kaleng, kaca.

Sampah memiliki beberapa karakterisistik yaitu:

1. *Garbage* yaitu jenis sampah yang terdiri dari sisa-sisa potongan hewan atau sayuran dari hasil pengolahan yang sebagian besar terdiri dari zat-zat yang mudah membusuk, lembab, dan mengandung sejumlah air bebas.
2. *Rubbish* terdiri dari sampah yang dapat terbakar atau yang tidak dapat terbakar yang berasal dari rumah-rumah, pusat-pusat perdagangan, kantor-kantor, tapi yang tidak termasuk garbage.
3. *Ashes* (Abu) yaitu sisa-sisa pembakaran dari zat-zat yang mudah terbakar baik dirumah, dikantor, industri.
4. *Street Sweeping* (Sampah Jalanan) berasal dari pembersihan jalan dan trotoar baik dengan tenaga manusia maupun dengan tenaga mesin yang terdiri dari kertas-kertas, daun-daunan.
5. *Dead Animal* (Bangkai Binatang) yaitu bangkai-bangkai yang mati karena alam, penyakit atau kecelakaan.
6. *Houshold Refuse* yaitu sampah yang terdiri dari *rubbish, garbage, ashes,* yang berasal dari perumahan.
7. *Abandonded Vehicles* (Bangkai Kendaraan) yaitu bangkai- bangkai mobil, truk, kereta api.
8. Sampah Industri terdiri dari sampah padat yang berasal dari industri-industri, pengolahan hasil bumi.
9. *Demolition Wastes* yaitu sampah yang berasal dari pembongkaran gedung.

10. *Construction Wastes* yaitu sampah yang berasal dari sisa pembangunan, perbaikan dan pembaharuan gedung-gedung.

Ada beberapa tahapan di dalam pengelolaan sampah padat yang baik, diantaranya:

1. Tahap pengumpulan dan penyimpanan di tempat sumber

Sampah yang ada dilokasi sumber (kantor, rumah tangga, hotel dan sebagainya) ditempatkan dalam tempat penyimpanan sementara, dalam hal ini tempat sampah. Sampah basah dan sampah kering sebaiknya dikumpulkan dalam tempat yang terpisah untuk memudahkan pemusnahannya. Adapun tempat penyimpanan sementara (tempat sampah) yang digunakan harus memenuhi persyaratan berikut berikut ini :

a. Konstruksi harus kuat dan tidak mudah bocor.
b. Memiliki tutup dan mudah dibuka tanpa mengotori tangan.
c. Ukuran sesuai sehingga mudah diangkut oleh satu orang.

Dari tempat penyimpanan ini, sampah dikumpulkan kemudian dimasukkan ke dalam dipo (rumah sampah). Dipo ini berbentuk bak besar yang digunakan untuk menampung sampah rumah tangga. Pengelolaanya dapat diserahkan pada pihak pemerintah. Untuk membangun suatu dipo, ada bebarapa persyaratan yang harus dipenuhi, diantaranya :

a. Dibangun di atas permukaan tanah dengan ketinggian bangunan setinggi kendaraan pengangkut sampah.
b. Memiliki dua pintu, pintu masuk dan pintu untuk mengambil sampah.
c. Memiliki lubang ventilasi yang tertutup kawat halus untuk mencegah lalat dan binatang lain masuk ke dalam dipo.
d. Ada kran air untuk membersihkan.
e. Tidak menjadi tempat tinggal atau sarang lalat atau tikus.
f. Mudah dijangkau masyarakat.

Pengumpulan sampah dapat dilakukan dengan 2 metode :

a. Sistem duet : tempat sampah kering dan tempat sampah basah.
b. Sistem trio : tempat sampah basah, sampah kering dan tidak mudah terbakar.

2. Tahap pengangkutan

Dari dipo sampah diangkut ke tempat pembuangan akhir atau pemusnahan sampah dengan mempergunakan truk pengangkut sampah yang disediakan oleh Dinas Kebersihan Kota.

### II.2.2 Sistem Informasi

Menurut James B Bower sistem informasi adalah suatu cara tertentu untuk menyediakan informasi yang dibutuhkan oleh organisasi agar dapat beroperasi secara benar dan menguntungkan. Pada dasarnya sistem informasi yang dibuat oleh manusia untuk mencapai suatu tujuan yaitu menyajikan informasi. Sistem informasi menerima masukan data dan instruksi, mengolah data sesuai instruksi, dan mengeluarkan hasilnya. Suatu sistem informasi pada dasarnya memiliki komponen yang saling berinteraksi membentuk suatu keatuan untuk mencapai sasaran sistem, yaitu [2]:

1. Komponen Tujuan

   Setiap sistem yang dibangun pasti mempunyai tujuan atau *goal* tertentu. Tujuan ini merupakan pedoman sistem untuk melaksanakan tugas serta merupakan pemacu untuk mencapai hasil tertentu. Sesuai dengan keberagaman sistem, setiap sistem tiak mempunyai tujuan yangidentik sama persis. Meskipun berbeda-beda sistem, namun secara umum tujuan dari sebuah sistem menurut Hall adalah untuk mendukung organisasi dari sistem tersebut, unutk menentukan pengambilan keputusan dari sistem dan untuk menentukan arah kegiatan dari operasi perusahaan.

2. Komponen Masukan

   Komponen masukan didalam sistem informasi adalah meliputi media dan metode yang digunakan untuk mendapatkan data yang akan dimasukkan. Data yang dimaksud adalah seperti dokumen dasar sebelum diolah menjadi informasi.

3. Komponen Proses

   Semua bahan yang dimasukan ke dalam sistem akan diolah atau diproses menjadi output, yaitu inormasi yang berguna bagi pemakainya. Kegiatan yang

ada dalam proses meliputi, mencatat, mengklasifikasikan, menghitung, menganalisis, membuat hipotesa dan perkiraan-perkiraan, menarik kesimpulan, serta membuat keputusan. Hasil proses ini akan diberikan pada bagian berikutnya yaitu output.

4. Komponen Keluaran

   Seperti tujuan dari sistem informasi yaitu sebagai penghasil informasi, maka komponen keluaran yang dimaksud berupa informasi berkualitas yang dikemas dalam bentuk-bentuk sesuai keinginan pemakai informasi tersebut.

5. Komponen *Feedback*

   *Feedback* merupakan bahan masukan dari pengguna yang membuat sebuah aplikasi akan menjadi lebih baik lagi dan lebih diinginkan dengan memperbarui apliksi dengan sesuai dan dibutuhkan pengguna.

6. Komponen Teknologi

   Komponen teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran serta membantu pengendalian dari sistems secara keseluruhan.

7. Komponen Basis Data

   Merupakan kumpulan data yang berhubungan satu dengan yang lainnya, tersimpan diperangkat keras computer dan perangkat lunak untuk memanipulasinya.

8. Komponen Batas

   Batas merupakan pemisah antara sistem dengan daerah di luar sistem. Sistem yang berada di luar sistem disebut dengan lingkungan [3].

### II.2.3 Konsep Citizen Journalism

*Citizen Journalism* (CJ) adalah sebuah konsep bahwa publik memiliki peranan yang aktif dalam proses mengumpulkan, melaporkan, menganalisa, dan menyebarkan informasi. *Citizen journalism* bertujuan untuk memberikan informasi yang independen, dapat dipercaya, akurat, memiliki pandangan yang luas, dan relevan yang dibutuhkan demokrasi [4].

Menurut Mark Deuze, ada 4 kategori *citizen journalism* yaitu:

1. *Mainstream News Sites* : web yang dimiliki oleh media konvensional. Contohnya, suaramerdeka.com, kompas.com, metrotvnews.com, dan lain-lain.
2. *Index & Category Sites :* web semacam ini adalah web yang menghubungkan antara pembaca dengan tulisan yang dibutuhkan yang ada di internet. Contohnya, *yahoo!, Google, dan lain-lain.*
3. *Meta & Comment Sites :* web semacam ini adalah web yang berisi informasi dan hasil penelitian yang berhubungan dengan journalisme dan media. Contohnya adalah poynter.org, dailysocial.net, sudutpandang.com, dan lain-lain.
4. *Share & Discussion Sites :* kategori ini berisi web yang fokus pada kepentingan publik. Web merupakan web yang memiliki system editorial dan moderasi yang minim. Contohnya, Slashdot, Kuro5shin, Twitter, dan lain-lain.

### II.2.4 Metode Peramalan

Peramalan (*forecasting*) merupakan prakiraan atau prediksi terhadap kejadian yang mungkin akan terjadi pada waktu yang akan datang. Dalam melakukan peramalan dilakukan pendekatan secara matematik atau pendekatan diskriptif yang dapat diterima sebagai hasil disiplin ilmu. Tujuan peramalan untuk penelitian ini mengetahui dalam hitungan waktu bahwa tempat pembuangan sampah sementara akan penuh setiap jam ataupun hari.

Peramalan mempunyai tipe-tipe yang mendasarinya yaitu [5]:

1. *Time Series Model*

   Metode *time series* adalah metode peramalan secara kuantitatif dengan menggunakan waktu sebagai dasar peramalan.
2. *Causal Model*

   Metode peramalan yang menggunakan hubungan sebab-akibat sebagai asumsi, yaitu bahwa apa yang terjadi di masa lalu akan terulang pada saat ini.
3. *Judgemental Model*

   Bila time series dan causal model bertumpu pada kuantitatif, pada judgemental mencakup untuk memasukkan faktor-faktor kuantitatif / subjektif ke dalam metode peramalan. Secara khusus berguna bilamana faktor-faktor subjektif

yang diharapkan menjadi sangat penting bilamana data kuantitatif yang akurat sudah diperoleh.

Peramalan memiliki banyak model metode, dengan masing-masing memiliki perhitungan dan pengujiannya sendiri, dikarenakan banyak metode yang ada maka perlunya mengetahui bagaimana cara memilihi metode peramalan yang tepat untuk kasus yang kita hadapi. Berikut adalah langkah-langkah peramalan:

1. Definisikan tujuan peramalan.
2. Plot Data masa lalu.
3. Memilih metode yang paling memenuhi tujuan peramalan dan sesuai dengan plot data.
4. Hitung parameter fungsi peramalan untuk masing-masing metode.
5. Hitung *fitting error* untuk semua metode yang dicoba.
6. Pilih metode yang baik, yaitu metode yang memberikan angka *error* paling kecil.
7. Ramalkan permintaan untuk periode mendatang.
8. Lakukan verivikasi peramalan

Tipe peramalan yang digunakan pada penelitian ini berdasarkan *causal model* yang dimana berdasarkan hubungan sebab-akibat sebagai asumsi, yaitu bahwa apa yang terjadi di masa lalu akan terulang pada saat ini. Metode yang digunakan untuk penelitian ini yaitu metode regresi, metode ini sering kali digunakan untuk periode peramalan jangka pendek. Hal- hal yang perlu diketahui sebelum melakukan peramalan dengan metode regresi adalah mengetahui terlebih dahulu kondisi- kondisi seperti :

1. Adanya informasi masa lalu.
2. Informasi yang ada dapat dibuatkan dalam bentuk data (dikuantifikasikan).
3. Diasumsikan bahwa pola data yang ada dari data masa lalu akan berkelanjutan dimasa yang akan datang.

Adapun data-data yang ada dilapangan yaitu [5]:

1. Musiman(*seasonal)*

   Terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulanan, atau hari-hari pada minggu tertentu). Penjualan

dari produk seperti minuman ringan, es krim, dan bahan bakar pemanas ruang semuanya menunjukkan jenis pola ini.

2. Horizontal(*stationary*)

   Terjadi bilamana data berfluktuasi disekitar nilai rata-rata yg konstan. Suatu produk yg penjualannya tdk meningkat atau menurun selama waktu tertentu termasuk jenis ini.

3. Siklus(*cylikal*)

   Terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Contoh: Penjualan produk seperti mobil, baja, dan peralatan utama lainnya.

4. Trend

   Terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Contoh: Penjualan banyak perusahaan, GNP dan berbagai indikator bisnis atau ekonomi lainnya.

Berikut pemaparan akan metode yang digunakan pada perangkat lunak *trash organizer*:

1. Tujuan dari peramalan dalam kasus ini adalah untuk mengetahui berapa jumlah sampah perharinya di setiap tempat pembuangan sampah di kota bandung.
2. Data yang dimiliki bersifat horizontal atau stabil yang dimana perharinya angka yang didapatkan tetap dan tidak terlalu jauh.
3. Metode peramalan yang digunakan untuk peramalan jumlah sampah perharinya di setiap tempat pembuangan sampah sementara adalah moving average (Metode rata-rata bergerak). Metode yang didasarkan mempunyai sifat serial data tetap atau horizontal dan sangat tepat untuk digunakan dalam jangka waktu pendek. Ada pun secara matematisnya mempunyai rrumusan seperti berikut:

$$F_{t+1} = \frac{\sum_{i=1}^{t-N=1} X_i}{N} = \frac{X + X_{t-1} + \cdots + X_{t-N+1}}{N}$$

Dimana:

$X_i$ : Data pengamatan periode t.

$N$ : Jumlah deret waktu yang digunakan.

$F_{t+1}$ : Nilai peramalan periode t+1

4. Perhitungan kesalahan peramalan yang digunakan adalah metode *mean absolut percentage error* (MAPE). Perhitungan yang berguna mengukur kesalahan ramalan dalam unit yang sama sebagai deret asli dengan menampilkan persentase. Semakin kecil persentase kesalahan semakin baik. MAPE dapat dirumuskan sebagai berikut:

$$MAPE = \frac{\sum_{i=1}^{n} \frac{e}{x_i} 100}{n}$$

Dimana :

$e = X_t - F_t$ : Kesalahan peramalan

n : Data yang diamati

### II.2.5 Object Oriented Analisys and Design

OOP (*Object Oriented Programming*) atau yang dikenal dengan Pemrograman Berorientasi Objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus ke dalam kelas-kelas atau objek-objek. Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu

masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri. Pemrograman orientasi-objek menekankan konsep-konsep yaitu [4]:

1. *Class*

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. *Object*

Membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer. Objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.

3. *Abstract*

Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat 23 melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. *Enkapsulation*

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek

tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. *Polymorphism*

Melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim.

6. *Inheritance*

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas).

### II.2.6 UML (Unified Modeling Language)

UML (*Unified Modeling Language*) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemgrograman berorientasi objek. Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang uml berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang bangun perangkat lunak efektif.

UML merupakan dasar bagi perangkat (tools) desain berorientasi objek dari IBM. UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. 25 UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan

standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan system. Fungsi UML pada perangkat lunak ini untuk memodelkan sistem atau alur pada perangkat lunak yang akan dibangun ini. UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu [5]:

1. *Use Case* Diagram untuk memodelkan proses bisnis.
2. *Conceptual* Diagram untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence* Diagram untuk memodelkan pengiriman pesan (message) antar objek.
4. *Collaboration* Diagram untuk memodelkan interaksi antar objects.
5. *State* Diagram untuk memodelkan perilaku objects di dalam sistem.
6. *Activity* Diagram untuk memodelkan perilaku Use Cases dan objects di dalam sistem.
7. *Class* Diagram untuk memodelkan struktur kelas.
8. *Object* Diagram untuk memodelkan struktur object.
9. *Component* Diagram untuk memodelkan komponen object.
10. *Deployment* Diagram untuk memodelkan distribusi aplikasi.

### II.2.7 Android

Pada tahun 2000, jauh sebelum produk ponsel pintar beredar dipasaran, Google mengakuisisi perusahaan bernama Android. Perusahaan tersebut merupakan pendatang baru berfokus untuk pengembangan system operasi embedded dan software untuk perangkat mobile. Kemudian setelah Android.inc menjadi group Google, para pendiri android diantaranya Andy Rubin, Rich Miner, Nick Sears dan Chris White berpindah ke Google. Pada saat itu OS Android dikembangkan hanya untuk kebutuhan internal oleh Google dan belum berlinsensi *open source*.

Android drilis pertama dengan standar *open source* pada tanggal 5 November 2007 bersama OHA, Alasan Google merilis android sebagai *open source* adalah untuk penetrasi pasar produk berbasis android dan juga mempermudah user

dalam mengakses layanan Google dan juga android digunakan sebagai media baru penayangan iklan yang dapat menambah *revenue* (pendapatan) Google, yang sebagian pendapatannya itu untuk pengembangan OS dan aplikasi android. *Platform* ini memudahkan untuk membangun perangkat lunak dikarenakan dapat bekerja diberbagai format seperti JSON [6].

### II.2.8 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform independent*).

Berikut ini adalah sifat dari Eclipse:

1. Multi-*platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan plug-in baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java. Eclipse memiliki konsep yaitu [6]:

1. Terbuka(*open*).
2. Mudah diperluas(*extensible*) untuk apa aja.
3. Tidak untuk sesuatu yang spesifik.

### II.2.9 JSON (Javascript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur [9]:

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

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

### II.2.10 API (Application Programming Interface)

Antarmuka pemrograman aplikasi (Application Programming Interface/API) adalah sekumpulan perintah, fungsi, dan protocol yang dapat digunakan untuk membangun perangkat lunak untuk system operasi tertentu. API menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.API dapat menjelaskan cara sebuah tugas (*task*) tertentu dilakukan. Dalam *pemrograman procedural* seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakannya [10].

### II.2.11 LBS (Location Based Service)

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

*Location based Service* memiliki dua unsur utama yaitu [11]:

1. *Location Manager* (*API Maps*)

   Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programming Interface* (API) Maps menyediakan fasilitas untuk menampilkan atau memanipulasi peta. Paket ini berada pada *com.google.android.maps*.

2. *Location Providers* (*API Location*)

   Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API Location* berhubungan dengan data GPS (*Global Positioning 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.