BAB II
LANDASAN TEORI

2.1 Penelitian Terdahulu

Dalam peneletian ini penulis memaparkan penelitian terdahulu yang relevan dengan permasalahan yang akan diteliti tentang analisis "Sistem Keluhan Pelanggan pada PDAM Tirta Raharja Kabupaten Bandung".

Birda Ariyadi (2012) dalam tesisnya yang berjudul "Sistem Informasi Pemeriksaan *Online* Gangguan Sambungan Langganan Air PDAM Tirta Raharja Kabupaten Bandung" memaparkan tujuan dari perancangan sistem tersebut adalah untuk mempercepat waktu pemrosesan data sehingga data yang diterima menjadi lebih *up to date*, selain itu untuk memantau kinerja bagian teknik / koperasi ketika melakukan pemeriksaan dan pengiriman data dilapangan.[2]

Persamaan peneliti dengan penelitian yang dilakukan oleh Birda Ariyadi adalah peneliti melakukan analisis sistem survei yang dilakukan di PDAM Tirta Raharja Kabupaten Bandung, sedangkan untuk perbedaannya adalah peneliti mengambil sumber informasi keluhan berdasarkan keluhan yang diterima melalui sistem (*CRM) Customer Relationship Management* sedangkan Birda Ariyadi mengambil sumber informasi dari pembacaan yang dilakukan oleh koperasi dan perbedaan yang lainnya yaitu peneliti membuat sistem ini untuk mempercepat proses penanganan keluhan yang dilaporkan oleh pelanggan kepada Bagian Pelayanan Langganan sehingga tidak terjadi lagi menumpukan data / berkas serta untuk mendukung sistem (*CRM) Customer Relationship Management* yang sudah berjalan pada saat ini.

2.2 Survei

Istilah survei biasanya dirancukan dengan istilah observasi dalam pengertian sehari-hari. Pada hal kedua istilah tersebut mempunyai pengertian yang berbeda, walaupun keduanya merupakan kegiatan yang saling berhubungan. Menurut kamus Webster, pengertian survei adalah suatu kondisi tertentu yang menghendaki kepastian informasi, terutama bagi orang – orang yang bertanggung jawab atau yang tertarik [3].

Menurut Singarimbun survei yaitu penelitian yang mengambil sampel dari satu populasi dan menggunakan kuesioner sebagai alat pengumpul data yang pokok.[3] Dari berbagai definisi tentang survei tersebut di atas dapat di simpulkan bahwa survei merupakan suatu aktivitas atau kegiatan penelitian yang dilakukan untuk mendapatkan suatu kepastian informasi (seperti : jumlah orang, persepsi atau pesan-pesan tertentu), dengan cara mengambil sampel dari satu populasi dan menggunakan kuesioner sebagai alat pengumpul data yang pokok. Jenis-jenis survei terbagi menjadi dua bagian yaitu survei secara umum dan survei secara ilmu penelitian.

1. Jenis survei secara umum terbagi menjadi 2 yaitu :
    a. Survei yang lengkap, yaitu yang mencakup seluruh populasi atau elemen-elemen yang menjadi objek penelitian. Survei tipe ini disebut sensus.
    b. Survei yang hanya menggunakan sebagian kecil dari populasi, atau hanya menggunakan sampel dari populasi. Jenis ini sering disebut sebagai sample survei *method*.
2. Jenis survei secara ilmu penelitian terbagi menjadi 6 yairu :
    a. Penelitian Exploratif, yaitu penelitian yang terbuka, mencari-cari pengetahuan peneliti tentang masalah yang diteliti masih terbatas. Pertanyaan dalam studi penjajagan ini misalnya : Apakah yang paling mencemaskan anda dalam hal infrastruktur di daerah Kota Bandung dalam lima tahun

terakhir ini? Menurut anda, bagaimana cara perawatan infrastruktur jalan dan jembatan yang baik ?

b. Penelitian Deskriptif, yaitu penelitian yang mempelajari masalah dalam masyarakat, tata cara yang berlaku dalam masyarakat serta situasi-situasi, sikap, pandangan, proses yang sedang berlangsung, pengaruh dari suatu fenomena; pengukuran yang cermat tentang fenomena dalam masyarakat. Peneliti menegmbangkan konsep, menghimpun fakta, tapi tidak menguji hipotesis;
c. Penelitian Evaluasi, yaitu penelitian yang mencari jawaban tentang pencapaian tujuan yang digariskan sebelumnya. Evaluasi disini mencakup formatif (melihat dan meneliti pelaksanaan program), Sumatif (dilaksanakan pada akhir program untuk mengukur pencapaian tujuan);
d. Penelitian Eksplanasi, yaitu penelitian yag menggunakan data yang sama, menjelaskan hubungan kausal antara variabel melalui pengujian hipotesis
e. Penelitian Prediksi, yaitu penelitian yang meramalkan fenomena atau keadaan tertentu
f. Penelitian Pengembangan Sosial, yaitu penelitian yang dikembangkan berdasarkan survei yang dilakukan secara berkala. Misal, jumlah dan persentase penduduk miskin

2.3 Pelayanan

Pelayanan menurut Kamus Besar Bahasa Indonesia (KBBI) adalah sebagai suatu usaha untuk membantu menyiapkan atau mengurus apa yang diperlukan orang lain. Sedangkan menurut Moenir pelayanan adalah kegiatan yang dilakukan oleh seseorang atau sekelompok orang dengan landasan faktor materi melalui sistem, prosedur dan metode tertentu dalam rangka usaha memenuhi kepentingan orang lain sesuai dengan haknya. Pelayanan hakikatnya adalah serangkaian kegiatan, karena itu pelayanan merupakan sebuah proses. Sebagai proses, pelayanan berlangsung secara rutin dan berkesinambungan, meliputi seluruh kehidupan orang dalam masyarakat.[4]

Sedangkan menurut Groonros dalam Ratminto dan Atik pelayanan adalah suatu aktivitas atau serangkaian aktivitas yang bersifat tidak kasat mata (tidak dapat diraba) yang terjadi sebagai akibat adanya interaksi antara konsumen dengan karyawan atau hal-hal lain yang di sediakan oleh perusahaan pemberi pelayanan yang dimaksudkan untuk memecahkan permasalahan konsumen atau pelanggan.[4]

Berbeda dengan Supranto mengatakan bahwa pelayanan atau jasa merupakan suatu kinerja penampilan, tidak terwujud dan cepat hilang, lebih dapat dirasakan dari pada dimiliki, serta pelanggan lebih dapat berpartisipasi aktif dalam proses mengonsumsi jasa tersebut.[4]

Menurut Philip Kotler dalam Supranto karakteristik jasa dapat diuraikan sebagai berikut : [4]

1. *Intangible* (tidak terwujud)
   Suatu jasa memiliki sifat tidak berwujud, tidak dapat dirasakan dan dinikmati sebelum dibeli oleh konsumen.
2. *Inseparibility* (tidak dapat dipisahkan)
   Pada umumnya jasa yang diproduksi (dihasilkan) dan dirasakan pada waktu bersamaan dan apabila dikehendaki oleh seseorang untuk diserakan kepada pihak lainnya, maka dia akan tetap merupakan bagian dari jasa tersebut.
3. *Variability* (bervariasi)
   Jasa senantiasa mengalami perubahan, tergantung dari siapa penyedia jasa, penerima jasa dan kondisi dimana jasa tersebut diberikan.
4. *Perishability* (tidak tahan lama)
   Daya tahan suatu jasa tergantung suatu situasi yang diciptakan oleh berbagai faktor.

Sedangkan menurut Sampara dalam Sinambela pelayanan adalah suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antar seseorang dengan orang lain atau mesin secara fisik, dan menyediakan kepuasan pelanggan.[4]

Berdasarkan berbagai pendapat di atas maka peneliti mengambil kesimpulan bahwa pelayanan adalah kegiatan yang dilakukan oleh seorang maupun sekelompok orang untuk memenuhi kebutuhan orang lain sesuai dengan prosedur dan sistem yang telah ditetapkan sebelumnya.

2.3.1 Pelayanan Publik

Pelayanan publik sangat erat kaitannya dengan pemerintah, karena salah satu tanggung jawab pemerintah ialah memberikan pelayanan kepada masyarakat. Kualitas pelayanan publik yang diterima masyarakat secara langsung dapat dijadikan tolak ukur dalam menilai kualitas pemerintah. Pelayanan publik dalam perkembangannya timbul dari adanya kewajiban sebagai suatu proses penyelenggaraan kegiatan pemerintahan baik yang bersifat individual maupun kelompok. Pelayanan publik memiliki peranan penting dalam kehidupan masyarakat saat ini dikarenakan tidak semua jasa atau pelayanan disediakan oleh pihak swasta, oleh karena itu pemerintah memiliki kuwajiban untuk memenuhi kebutuhan pelayanan masyarakat yang tidak disediakan swasta tersebut.[5]

Menurut Sinambela pelayanan publik diartikan sebagai pemberian layanan (melayani) keperluan orang atau masyarakat yang mempunyai kepentingan pada organisasi iti sesuai dengan aturan pokok dan tatacara yang telah ditetapkan. [5]

Sedangkan di dalam Undang-undang No 25 tahun 2009 tentang pelayanan publik, mendefinisakan bahwa pelayanan publik adalah kegiatan atau serangkaian kegiatan dalam rangka pemenuhan kebutuhan pelayanan sesuai dengan peraturan perundang-undangan bagi setiap warga negara dan penduduk atas barang, jasa, dan atau pelayanan administratif yang disediakan oleh penyelenggara pelayanan publik. Adapun asas-asas pelayanan publik yang termuat dalam undang-undang tersebut meliputi:

1. Kepentingan umum

   Yaitu pemberian pelayanan tidak boleh mengutamakan kepentingan pribadi dan atau golongan.
2. Kepastian hukum

   Yaitu jaminan terwujudnya hak dan kewajiban dalam penyelengaraan pelayanan.
3. Kesamaan hak

   Yaitu pemberian pelayanan tidak membedakan suku, ras, agama, golongan, gender, dan status ekonomi.
4. Keseimbangan hak dan kewajiban

   Yaitu pemenuhan hak harus sebanding dengan kewajiban yang harus dilaksanakan, baik oleh pemberi maupun penerima pelayanan.
5. Keprofesionalan

   Yaitu pelaksana pelayanan harus memiliki kompetensi yang sesuai dengan bidang tugas.
6. Partisipatif

   Yaitu peningkatan peran serta masyarakat dalam penyelenggaraan pelayanan dengan memperhatikan aspirasi, kebutuhan, dan harapan masyarakat
7. Persamaan perlakuan/tidak diskriminatif

   Yaitu setiap warga negara berhak memperoleh pelayanan yang adil
8. Keterbukaan

   Yaitu setiap penerima pelayanan dapat dengan mudah mengakses dan memperoleh informasi mengenai pelayanan yang diinginkan.
9. Akuntabilitas

   Yaitu proses penyelenggaraan pelayanan harus dapat dipertanggungjawabkan sesuai dengan ketentuan peraturan perundang-undangan.
10. Fasilitas dan perlakuan khusus bagi kelompok rentan, yaitu pemberian kemudahan terhadap kelompok rentan sehingga tercipta keadilan dalam pelayanan.

11. Ketepatan waktu

    Yaitu penyelesaian setiap jenis pelayanan dilakukan tepat waktu sesuai dengan standar pelayanan.

12. Kecepatan, kemudahan dan keterjangkauan

    Yaitu setiap jenis pelayanan dilakukan secara cepat, mudah dan terjangkau.

Berdasarkan definisi di atas maka peneliti menyimpulkan bahwa pelayanan publik adalah serangkaian proses atau usaha yang dilakukan perorangan maupun instansi publik untuk mencapai tujuan tertentu atau melaksanakan ketentuan perundang-undangan.

2.4 Sistem Informasi

Beberapa penjelasan terkait dengan sistem informasi antara lain pengertian sistem, pengertian informasi, dan pengertian sistem informasi itu sendiri [6]

2.4.1 Sistem

Sistem dapat diartikan sebagai kumpulan dari komponen-komponen yang saling berkaitan untuk mencapai target atau tujuan tertentu.

2.4.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu diantaranya :

1. Komponen
2. Batas sistem
3. Lingkungan luar sistem
4. Penghubung sistem
5. Masukan sistem
6. Pengolahan sistem
7. Keluaran sistem
8. Sasaran sistem

2.4.1.2 Klasifikasi sistem

Suatu sistem dapat diklasifikasikan menjadi seperti berikut :

1. Sistem abstrak dan sistem fisik
2. Sistem alamiah dan sistem buatan manusia
3. Sistem tertentu (determinasi) dan sistem tak tentu (*probabilistic*)
4. Sistem tertutup dan sistem terbuka

2.4.2 Informasi

Informasi merupakan hasil dari pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan suatu keputusan.

2.4.2.1 Kualitas Informasi

Informasi yang berkualitas memiliki 3 kriteria, yaitu :

1. Akurat

   Informasi harus bebas dari kesalahan tidak bias ataupun menyesatkan. Akurat juga berarti bahwa informasi itu harus dengan jelas mencerminkan maksudnya.
2. Tepat waktu

   Informasi yang datang pada penerima tidak boleh terlambat.
3. Relevan

   Informasi harus mempunyai manfaat bagi pemakainya.

2.4.3 Sistem Informasi

Sistem informasi dapat disimpulkan sebagai serangkaian komponen berupa manusia, prosedur, data dan teknologi yang digunakan untuk melakukan sebuah proses yang menghasilkan informasi yang bernilai untuk pengambilan suatu keputusan. Sistem informasi melekat dan merupakan infrastruktur penunjang keberhasilan bagi setiap organisasi atau perusahaan untuk mencapai tujuannya.[6]

2.4.3.1 Komponen Sistem Informasi

Komponen informati terdiri dari 6 bagian antaralain :

1. Perangkat keras (*hardware*), mencakup berbagai piranti fisik seperti komputer dan printer
2. Perangkat lunak (*software*), yaitu sekumpulan instruksi atau prosedur yang memungkinkan perangkat keras memproses data.
3. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki
4. Manusia, yaitu semua pihak yang bertanggung jawab dalam pengembangan sistem informasi , pemrosesan dan penggunaan keluaran sistem informasi.
5. Basis data (*database*), yaitu kumpulan data, table yang saling berkaitan.
6. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2.4.3.2 Klasifikasi Sistem Informasi

Sistem informasi berdasarkan level organisasi di kelompokan menjadi 3 yaitu

1. Level Operasional

   Sistem informasi pada level operasional mendukung manajer operasional dengan menyimpan berbagai aktivitas elementer dan transaksi dari organisasi. Selain itu pula sistem informasi di level operasional membantu manajer operasional dalam memantau kegiatan atau transaksi rutin harian dalam organisasi.

2. Level Fungsional

   Kegunaan sistem informasi pada level ini adalah membantu mengintegrasikan pengetahuan baru ke dalam bisnis dan untuk membantu organisasi untuk mengendalikan pekerjaan administrasi. Sistem informasi pada level fungsional dirancang untuk melayani pengamatan, pengendalian, pembuat keputusan, dan aktivitas administrasi manejer level menengah. Contoh sistem informasi pada

level ini adalah, sistem penunjang keputusan (SPK) atau *Decision Support System* (DDS). Sistem Penunjang Keputusan adalah suatu sistem informasi untuk membantu level menengah untuk proses pengambilan keputusan.

3. Level Manajerial
   Sistem informasi pada level manajerial membantu perencanaan jangka panjang oleh para manajer senior. Perhatian utamanya terletak pada mengantisipasi perubahan pada lingkungan luar ke dalam perusahaan. Sistem informasi di level ini digunakan untuk perencanaan strategik dan pemecahan masalah. Sistem informasi yang digunakan di level ini adalah sistem informasi eksekutif atau *executife information system* (EIS) adalah suatu bagian yang menyediakan informasi bagi eksekutif mengenai kinerja keseluruhan perusahaan. Misalnya permasalahan tentang arah bisnis yang akan dilakukan di masa depan, posisi kompetitor dan bagaimana mengatasinya, dan lain – lain.

2.5 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Quacomm, T-Mobile, dan Nvidia. [7]

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Sekitar september 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada Januari 2008. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communication, diproduksi oelh Asustek Computer Inc, Garmin Ltd, Softbank, Sonny Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru. [8]

2.5.1 Versi versi dari android

Seiring dengan perkembangan teknologi dan perangkat *smartphone* maka android mengalami beberapa perkembangan diantaranya : [8]

1. Android v1.0 Astro (Alpha)

   Sebenarnya sebelum mereka memberikan nama-nama kudapan sebagai nama untuk versi OS nya, Android sempat memiliki 2 versi awal dengan nama Android Alpha dan Beta. Nama untuk versi pertama ini sendiri sebenarnya adalah Android Astro, namun karena alasan hak cipta (*trademark*), nama ini tidak jadi digunakan. Di versi awal ini belum ada perangkat dengan sistem operasi Android yang dijual secara komersil.
2. Android v1.1 Bender (Beta)

   Versi ini dirilis pada tanggal 5 November 2007 yang merupakan versi lanjutan dari Android Astro (Alpha). Sama seperti versi awalnya, nama Bender juga juga tak jadi digunakan karena alasan hak cipta (*trademark*). Kemudian lahirlah telepon seluler pertama dengan sistem operasi Android yang dijual secara komersil yakni HTC Dream.

3. Android v1.5 Cupcake

   Ini merupakan versi pertama yang menggunakan nama makanan manis sebagai kode nama untuk tiap versi Android yang kemudian tradisi untuk menamai versi Android dengan nama makanan manis masih diteruskan hingga saat ini. Android Cupcake dirilis pada tanggal 30 April 2009.

4. Android v1.6 Donut

   Dirilis tidak sampai setahun setelah perilisan Android Cupcake, yakni pada tanggal 15 September 2009. Versi ini dihadirkan untuk menutupi bug pada versi sebelumnya, sekaligus untuk penambahan beberapa fitur seperti misalnya dukungan untuk perangkat dengan ukuran layar yang lebih besar.

5. Android v2.0 – 2.1 Éclair

   Sistem operasi ini juga dirilis tidak sampai setahun setelah perilisan dua versi sebelumnya yakni pada tanggal 26 Oktober 2009. Mereka masih berfokus untuk menutupi bug yang ada dan juga menambahkan beberapa fitur seperti Bluetooth, flash pada kamera, fitur digital zoom pada kamera, *multi-touch*, live wallpaper, dan lainnya. Hadirnya perangkat seri Nexus dari Google yang pertama kali muncul yakni HTC Nexus One juga menggunakan versi OS Android Eclair.

6. Android v2.2 Frozen Yoghurt (Froyo)

   Dirilis pada tanggal 20 Mei 2010. Perangkat dengan OS Android semakin banyak dan kehadirannya mulai dilirik oleh pasar meski masih jauh dibawah kepopuleran OS lain seperti Symbian dan Windows Mobile.

7. Android v2.3 Gingerbread

   Dirilis pada tanggal 6 Desember 2010 bersamaan dengan dihadirkannya Nexus S yang merupakan perangkat *smartphone* seri Nexus yang diproduksi oleh Samsung. Versi OS ini juga mengawali kesuksesan Android di jagad *smartphone* meski masih kalah populer dengan BlackBerry OS. Beberapa vendor mulai serius untuk menggarap perangkat dengan OS Android.

   Pada saat itu, Samsung dengan Galaxy series nya berperan besar dalam kesuksesan Android. Promosi yang luar biasa gencarnya membuat orang awam

mulai mengenal sistem operasi Android. Bahkan saat itu sebagian besar orang beranggapan bahwa OS Android adalah milik Samsung karena kuatnya branding yang dilakukan oleh Samsung. Ini juga menjadi awal mula kedigdayaan Samsung di jagad smartphone.

8. Android v3.0 – 3.2 Honeycomb

   Versi ini dirilis pada tanggal 10 Mei 2011 dan dirancang khusus untuk perangkat tablet, yang kala itu mulai populer di pasaran salah satunya berkat promosi Samsung dan juga kepopuleran Apple iPad.

9. Android v4.0 Ice Cream Sandwich

   Dirilis pada 16 Desember 2011. Bisa dibilang merupakan Android Honeycomb yang disempurnakan, dan dioptimalkan untuk penggunaan baik smartphone maupun tablet. Perubahan yang paling terlihat dari versi ini dibanding dengan versi sebelumnya adalah dari segi User interface yang nampak lebih bersih dan elegan. Versi ini juga lebih dioptimalkan untuk urusan multitasking. Bersamaan dengan diperkenalkannya Android ICS, Google juga memperkenalkan perangkat Galaxy Nexus yang merupakan seri smartphone Nexus yang diproduksi oleh Samsung. Setelah versi ini, Google kemudian secara rutin memperkenalkan perangkat seri Nexus pada tiap kali mereka memperkenalkan versi Android terbaru.

10. Android v4.1 – 4.3 Jelly Bean

    Dirilis pada 9 Juli 2012. Bersamaan dengan diperkenalkannya versi OS 4.1 pada 27 Juni 2012, Google juga memperkenalkan Nexus 7 yang diproduksi oleh ASUS. Nexus 7 (generasi 1) merupakan seri Nexus pertama yang merupakan perangkat tablet. Jelly Bean mengalami 3x update versi yakni 4.1, 4.2 hingga 4.3. Selanjutnya mereka memperkenalkan Android v4.2 bersamaan dengan dihadirkannya Nexus 4, smartphone yang diproduksi oleh LG plus Nexus 10, perangkat tablet yang diproduksi oleh Samsung.

    Pada saat versi 4.3 dirilis, Google juga merilis Nexus 7 generasi 2 yang masih diproduksi oleh ASUS yang mana ia memiliki beberapa peningkatan seperti

misalnya penambahan kamera belakang serta dukungan untuk konektivitas internet.

11. Android v4.4 Kitkat

    Nama Kitkat diambil dari sebuah produk cemilan wafer berlapis coklat yang dimiliki oleh Nestle. Sebelumnya Android versi "K" ini disebut-sebut sebagai Key Lime Pie, namun atas beberapa pertimbangan akhirnya Google lebih memilih untuk memberi nama Kitkat. Ceritanya, Kitkat adalah salah satu cemilan yang tersedia di dapur kantor yang biasanya juga menemani para programer Google. Hingga seseorang berkata "Hey, kenapa kita tidak menamainya sebagai Kitkat?". Sesaat setelah ide itu muncul, Google segera menghubungi pihak Nestle sebagai pemilik merk dagang Kitkat dan mereka menyetujui pemberian nama Kitkat untuk versi Android K. Karyawan Google sendiri tidak mengetahui bahwa Android 4.4 akan diberi nama Kitkat karena yang mereka tau versi Android K adalah Key Lime Pie. Mereka baru mengetahuinya setelah patung maskot Android Kitkat diletakkan di kantor pusat Google.

    Versi ini diklaim lebih ramah terhadap perangkat dengan spesifikasi seadanya. Bahkan perangkat dengan RAM 512 MB masih bisa menjalankan OS versi ini dengan mulus. Berbeda dengan Jelly Bean yang minimal harus memiliki RAM diatas 756 MB agar dapat berjalan dengan mulus. Bersamaan dengan dirilisnya Android Kitkat pada tanggal 31 Oktober 2013, Google juga merilis Smartphone Nexus 5 yang diproduksi oleh LG.

12. Android v5.0 – 5.1 Lollipop

    Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih flat dengan konsep material design. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk penggunaan RAM diatas 3 GB pada hardware perangkat. Penggunaan prosesor 64-bit pun makin banyak diadopsi oleh para vendor, mulai dari penerapan pada perangkat flagship hingga perangkat kelas menengah kebawah.

13. Android v6.0 Marshmallow

    Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya Google juga memperkenalkan 2 perangkat *smartphone* Nexus sekaligus yang diproduksi oleh 2 vendor yang berbeda.

    Nexus 5X adalah versi smartphone Nexus kelas menengah dengan ukuran layar 5.2 inch yang diproduksi oleh LG. Sedangkan yang satunya lagi memiliki bentang layar yang lebih lebar yakni 5.7 inch yang diberi nama Nexus 6P yang merupakan *smartphone flagship* hasil kerjasama Google dengan Huawei.

14. Android v7.0 Nougat

    Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi "N" ini adalah Nutella. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan Nougat (yang sepintas lebih mirip dengan tempe itu). Sebelumnya, Google telah mengundang para penggunanya untuk memberikan ide penamaan pada versi ini. Beberapa nama termasuk Nutella dan Nastar pun muncul, hingga akhirnya Google lebih memilih nama Nutella.

2.5.2 Feature

Features yang tersedia pada *platform* Android saat ini antara lain: [8]

1. *Framework* Aplikasi yang mendukung penggantian komponen dan reusable.
2. Mesin Virtual Dalvik berjalan diatas Linux kernel dan dioptimalkan untuk perangkat mobile.
3. *Integrated browser* berdasarkan *open source engine WebKit*.
4. Grafis yang dioptimalkan dan didukung oleh *library* grafis 2D yang terkostumisasi, grafis 3D berdasarkan spesifikasi openGL ES 1,0 (Opsional akselarasi hardware).
5. SQLite untuk penyimpanan data.

6. *Media support* yang mendukung audio, video, dan gambar (MPGE, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
7. GSM Telephony (tergantung *hardware*).
8. Bluetooth, EDGE, 3G, GPS, dan WIFI (tergantung hardware).
9. Lingkungan *development* yang lengkap dan kaya termasuk perangkat emulator, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk Eclips IDE.
10. *Multi-touch*, kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.
11. Market, Seperti kebanyakan Handphone yang memiliki tempat penjualan aplikasi, market pada android merupakan katalog aplikasi yang dapat di download dan di install pada Handphone melalui internet.

2.6 *Database*

Dasabase adalah kumpulan file-file yang saling berelasi sehingga membentuk satu bangunan data, relasi tersebut biasa ditunjukan dengan kunci dari tiap file yang ada. Satu *database* menunjukkan satu kumpulan data yang dipakai dalam lingkup perusahaan, instansi. Bila terdapat file yang tidak dapat dipadukan atau dihubungkan dengan file lainnya berarti file tersebut bukanlah kelompok dari satu *database*, ia akan membentuk satu *database* sendiri.[9]

Komponen dalam *database*:

1. *Entity*: orang, tempat, kejadian atau konsep yang informasinya direkam. Pada bidang Administrasi Siswa misal entitynya adalah siswa, buku, pembayaran, dan nilai test.
2. *Attribute*: Setiap entity mempunyai attribute atau sebutan untuk mewakili suatu entity. Seorang siswa dapat dilihat dari attributenya, misalnya nama, nomor siswa, alamat. Attribute juga disebut juga sebagai data elemen, data field, atau data item.

3. *Data Value* (nilai atau isi data): data aktual atau informasi yang disimpan pada tiap data elemen atau attribute.
4. *Record/tuple*: kumpulan elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap.
5. *File*: Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, namun berbeda beda data valuenya.

2.6.1 Entity Relationship Diagram

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. [10]

Menurut salah satu para ahli, Brady dan Loonam, *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analys* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.[10]

Komponen-komponen *ERD* antara lain :

1. Entitas adalah objek dalam dunia nyata yang dapat dibedakan dengan objek lain, sebagai contoh mahasiswa,dosen,departemen. Entitias terdiri atas beberapa atribut sebagai contoh atribut dari entitas mahasiswa adalah nim,nama,alamat,email, dll. Atribut nim merupakan unik untuk mengidentifikasikan / membedakan mahasiswa yg satu dengan yg lainnya. Pada setiap entitas harus memiliki 1 atribut unik atau yang disebut dengan *primary key*.

2. Atribut adalah Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips. Ada dua jenis Atribut :
   a. *Identifier* (*key*) digunakan untuk menentukan suatu *entity* secara unik (*primary key*).
   b. *Descriptor* (*nonkey attribute*) digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.
3. Relasi adalah hubungan antara beberapa entitas. sebagai contoh relasi antar mahasiswa dengan mata kuliah dimana setiap mahasiswa bisa mengambil beberapa mata kuliah dan setiap mata kuliah bisa diambil oleh lebih dari 1 mahasiswa. relasi tersebut memiliki hubungan banyak ke banyak. Berikut adalah contoh *ERD*. Jenis relasi terbagi menjadi 3 jenis yaitu :
   a. *One-to-one* : sebuah entitas pada A berhubungan dengan entitas B paling banyak 1contoh diatas relasi pegawai dan departemen dimana setiap pegawai hanya bekerja pada 1 departemen
   b. *One-to-many* : sebuah entitas pada A berhubungan dengan entitas B lebih dari satu contoh diatas adalah 1 depertemen memiliki banyak pegawai
   c. *Many-to-many* : sebuah entitas pada A berhubungan dengan entitas B lebih dari satu dan B berhubungan dengan A lebih dari satu jugan contoh diatas adalah relasi mahasiswa dengan mata kuliah.

   Berikut adalah metode/tahap untuk membuat *ERD* :

   1. Menentukan Entitas
   2. Menentukan Relasi
   3. Menggambar *ERD* sementara
   4. Mengisi Kardinalitas
   5. Menentukan Kunci Utama
   6. Menggambar *ERD* berdasar *Key*

7. Menentukan Atribut
8. Memetakan Atribut
9. Menggambar *ERD* dengan Atribut

2.7 *Unified Modelling Language*

UML merupakan singkatan dari "*Unified Modelling Language*" yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi *UML* yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem *software*. Saat ini *UML* sudah menjadi bahasa standar dalam penulisan *blue print software*. [11]

2.7.1 Tujuan atau fungsi dari penggunaan *UML*

Tujuan atau fungsi dari penggunaan UML antara lain :

1. Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemerograman maupun proses rekayasa.
2. Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.
3. Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
4. Dapat berguna sebagai *blue print*, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
5. Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
6. Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

2.7.2 Jenis-Jenis Diagram *UML*

Dibawah ini dijelaskan jenis-jenis diagram uml seperti *use case diagram, activity diagram, sequence diagram, class diagram, statemachine diagram, communication diagram, deployment diagram, component diagram, object diagram, composite structure diagram, interaction overview diagram, package diagram, diagram Timing* berikut penjelasan dari masing-masing diagram.[11]

1. *Use case diagram*

   Use Case Diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan interaksi antara sistem dan aktor, *use case diagram* juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya.
2. *Activity Diagram*

   Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada *UML* yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.
3. *Sequence diagram*

   Sequence diagram yaitu salah satu jenis diagram pada *UML* yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence* diagram juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada *use case diagram.*
4. *Class diagram*

   Class diagram yaitu salah satu jenis diagram pada *UML* yang digunakan untuk menampilkan kelas-kelas maupun paket-paket yang ada pada suatu sistem yang nantinya akan digunakan. Jadi diagram ini dapat memberikan sebuah gambaran mengenai sistem maupun relasi-relasi yang terdapat pada sistem tersebut.
5. *Statemachine diagram*

   Statemachine diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.

6. *Communication diagram*

   Communication diagram yaitu salah satu jenis diagram pada *UML* yang dapat menggamabarkan tahapan terjadinya suatu aktivitas dan diagram ini juga menggambarkan interaksi antara objek yang ada pada sistem. Hampir sama seperti *sequence diagram* akan tetapi *communication diagram* lebih menekankan kepada peranan masing-masing objek pada sistem.

7. *Deployment diagram*

   Deployment diagram yaitu salah satu diagram pada *UML* yang menunjukan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. Jadi *Deployment diagram* intinya untuk menunjukan letak *software* pada *hardweare* yang digunakan sistem.

8. *Component diagram*

   Component diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan *software* pada suatu sistem. *Component diagram* merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa file data atau .exe, *source code*, table, dokumen dsb.

9. *Object diagram*

   Object diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan objek-objek pada suatu sistem dan hubungan antarnya.

10. *Composite structure diagram*

    Composite structure diagram yaitu salah satu jenis diagram pada *UML* yang menggambarkan struktur internal dari penklasifikasian (*class, component* atau *use case*) dan termasuk titik-titik interaksi penklasifikasian kebagian lainnya dari suatu sistem. Ini hampir mirip seperti *class diagram* akan tetapi *composite structure diagram* menggambarkan bagian-bagian dari individu kelas saja bukan semua kelas.

11. *Interaction Overview Diagram*

    Interaction Overview diagram yaitu salah satu jenis diagram pada *UML* yang berguna untuk men-visualisasikan kerjasama dan hubungan antara *activity diagram* dengan *sequence diagram*.

12. *Package diagram*

    Package diagram yaitu salah satu jenis diagram pada *UML* digunakan untuk mengelompokan kelas dan juga menunjukan bagaimana elemen model akan disusun serta mengambarkan ketergantungan antara paket-paket.

13. *Diagram Timing*

    Diagram timing yaitu salah satu jenis diagram pada *UML* yang disebut sebagai bentuk lain dari interaksi diagram, dimana fokus yang paling utamanya kepada waktu. *Diagram timing* berguna untuk menunjukan faktor-faktor yang membatasi waktu antara perubahan state terhadap objek yang berbeda.

2.8 OOP

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain. [12]

2.8.1 Konsep OOP (*Object Oriented Programming*)

2.8.1.1 Konsep Objek

Objek dalam pengertian sehari-hari adalah benda, baik yang secara fisik dapat diketahui keberadaannya maupun yang bersifat khayal (virtual). Contohnya, kamera, kalkulator, ponsel, ide, gagasan, dan sebagainya.

Objek dalam pemrograman (*soft object*) adalah suatu komponen dari suatu sistem yang dibangun, misalnya objek matematis, objek jendela (windows), objek tombol, objek kotak gambar, objek kotak teks, dsb.[12]

2.8.1.2 Konsep Kelas

Setiap objek memiliki suatu kelompok. Kelompok ini dalam konsep OOP disebut "kelas" atau "class". Misalnya Toyota, Daihatsu, dan Honda adalah termasuk kelompok 'mobil'.

Dalam suatu kelas terdapat dua hal: sifat-sifat (*attribute/field/property*) dan fungsi (*method*) atau kelakuan (*behavior*). Misalnya kelas mobil memiliki atribut warna body, jumlah tempat duduk, ukuran ban, dsb, kemudian terdapat fungsi seperti pengapian, pengisian aki, percepatan (*accelerator*), rem, dsb. Pada tingkat pemrograman suatu atribut bisa berupa definisi suatu *varabel/field* atau definisi fungsi, sedangkan kelakuan bisa berupa implementasi dari suatu fungsi.

Sifat-sifat dari suatu kelas bisa diwariskan (inheritance) ke sub-kelasnya, misal sub-kelas mobil sedan mewarisi atribut dari kelas mobil. Fungsi-fungsi kelas juga dapat diwariskan bilamana perlu.[12]

2.8.1.3 Kelas Konstruktor

Konstruktor adalah suatu fungsi khusus yang diberi nama sama dengan nama kelasnya. Pemberian nilai pada atribut kelas dapat dilakukan pada saat objek didefinisikan melalui kelas konstruktor (*class constructor*).

2.8.1.4 Enkapsulasi

Enkapsulasi (*encapsulation*) adalah suatu kelas sebenarnya yang terpisah dari kelas yang lain, dimana atribut dan fungsi terbungkus dalam kapsul sehingga akses dari luar terhadap isi kelas pada kelas pada dasarnya terbatas.[12]

2.8.1.5 Pewarisan

Peristiwa pemakaian kelas objek dengan parameter-parameter lokal yang sesuai dengan definisi antarmuka objek dapat dipandang sebagai pewarisan (*inheritance*) sifat-sifat objek ke program klien. Sebagai contoh disediakan objek bingkai atau *frame* dengan antarmuka seperti *height* (tinggi) dan *width* (lebar). Dalam hal ini program klien dapat memanggil obejk *frame* dan men-set tinggi dan lebarnya sesuai kebutuhan. *Inheritance* berarti pewarisan pada sub-kelas dari suatu kelas, yang bisa dilaksanakan melalui kata kunci *extends*. [12]

2.8.1.6 Polimorfisme

Polimorfisme adalah sifat dari kelas objek yang memungkinkan beberapa fungsi dapat memiliki nama yang sama walaupun isi fungsi tersebut berbeda. Sebagai contoh, fungsi konstruktor suatu kelas *circle* tetapi dengan parameter berbeda, *circle* dengan parameter x, *circle* dengan parmeter x,y, dan *circle* dengan parameter x,y,z. Jadi setiap kelas mempunyai parameter yang berbeda dan dapat digunakan sesuai kebutuhannya masing-masing.[12]

2.9 DFD (*Data Flow Diagram)*

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. [13]

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Kelebihan utama pendekatan aliran data, yaitu :

1. Kebebasan dari menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain dalam sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan

DFD terdiri dari context diagram dan diagram rinci (*DFD Levelled*).

a. Context Diagram

Berfungsi memetakan model lingkungan (menggambarkan hubungan antara entitas luar, masukan dan keluaran sistem), yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

b. DFD *Levelled*

Menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi. Dalam *DFD levelled* akan terjadi penurunan level dimana dalam penurunan level yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas, jadi dalam *DFD levelled* bisa dimulai dari DFD level 0 kemudian turun ke DFD level 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di level x harus berhubungan dengan aliran

data yang masuk dan keluar pada level x+1 yang mendefinisikan proses pada level x tersebut.

2.10 *Java*

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, desktop, web dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Java adalah bahasa pemrograman yang berorientasi objek (*OOP*) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source.[14]

2.10.1 Karakteristik *Java*

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut

1. Sederhana

   Bahasa pemrograman *Java* menggunakan sintaks mirip dengan C++ namun sintaks pada *Java* telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan multiple inheritance. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.
2. Berorientasi objek (*Object Oriented*)

   Java mengunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.
3. Dapat didistribusi dengan mudah

   Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya libraries networking yang terintegrasi pada *Java*.

4. *Interpreter*

   Program *Java* dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan source code *Java* yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada *platform* yang berbeda-beda.

5. *Robust*

   Java mempuyai reliabilitas yang tinggi. *Compiler* pada Java mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. *Java* mempunyai *runtime-Exception* handling untuk membantu mengatasi error pada pemrograman.

6. Aman

   Sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, *Java* memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

7. *Architecture Neutral*

   Program *Java* merupakan platform independent. Program cukup mempunyai satu buah versi yang dapat dijalankan pada platform yang berbeda dengan *Java Virtual Machine.*

8. Portabel

   Source code maupun program *Java* dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.

9. *Performance*

   Performance pada *Java* sering dikatakan kurang tinggi. Namun *performance Java* dapat ditingkatkan menggunakan kompilasi *Java* lain seperti buatan Inprise, Microsoft ataupun Symantec yang menggunakan Just In Time Compilers (JIT).

10. *Multithreaded*

    Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.