BAB 2

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Pengertian Tinjauan Pustaka disebutkan bahwa "Tinjauan Pustaka" mempunyai arti : peninjauan kembali pustaka-pustaka yang terkait(*review of related literature*). Sesuai dengan arti tersebut, suatu tinjauan pustaka berfungsi sebagai peninjauan kembali (*review*) pustaka (laporan penelitian, dan sebagainya) tentang masalah yang berkaitan—tidak selalu harus tepat identic dengan bidang permasalahan yang dihadapi—tetapi termasuk pula yang seiring dan berkaitan(*collateral*). Fungsi peninjauan kembali pustaka yang berkaitan merupakan hal yang mendasar dalam penelitian, seperti yang dinyatakan oleh Leedy(1997) bahwa semakin banyak seorang peneliti mengetahui, mengenal dan memahami tentang penelitian-penelitian yang pernah dilakukan sebelumnya (yang berkaitan erat dengan topik penelitiannya), semakin dapat dipertanggung jawabkan caranya meneliti permasalahan yang dihadapi. Walaupun demikian, sebagian penulis (usulan penelitian atau karya tulis) menganggap tinjauan pustaka merupakan bagian yang tidak penting sehingga ditulis "asal ada" saja atau hanya untuk sekedar membuktikan bahwa penelitian (yang diusulkan) belum pernah dilakukan sebelumnya. Pembuktian keaslian penelitian tersebut sebenarnya hanyalah salah satu dari beberapa kegunaan tinjauan pustaka. Kelemahan lain yang sering pula dijumpai adalah dalam penyusunan, penstrukturan, atau pengorganisasian tinjauan pustaka.

Banyak penulisan tinjauan pustaka yang mirip resensi buku (dibahas buku per buku, tanpa ada kaitan yang bersistem) atau mirip daftar pustaka (hanya menyebutkan siapa penulisnya dan di pustaka mana ditulis, tanpa membahas apa yang ditulis). Berdasar kelemahan-kelemahan yang sering dijumpai di atas, tulisan ini berusaha untuk memberikan kesegaran pengetahuan tentang cara-cara penulisan tinjauan pustaka yang lazim dilakukan. Cakupan tulisan ini meliputi empat hal, yaitu: (a) kegunaan, (b) organisasi tinjauan pustaka, (c) kaitan tinjauan pustaka

dengan daftar pustaka, dan (d) cara pencarian bahan-bahan pustaka, terutama dengan memanfaatkan teknologi informasi.

Kegunaan Tinjauan Pustaka :

Leedy menerangkan bahwa suatu tinjauan pustaka mempunyai kegunaan untuk: (1) mengungkapkan penelitian-penelitian yang serupa dengan penelitian yang (akan) kita lakukan; dalam hal ini, diperlihatkan pula cara penelitian-penelitian tersebut menjawab permasalahan dan merancang metode penelitiannya; (2) membantu memberi gambaran tentang metoda dan teknik yang dipakai dalam penelitian yang mempunyai permasalahan serupa atau mirip penelitian yang kita hadapi; (3) mengungkapkan sumber-sumber data (atau judul-judul pustaka yang berkaitan) yang mungkin belum kita ketahui sebelumnya; (4) mengenal peneliti-peneliti yang karyanya penting dalam permasalahan yang kita hadapi (yang mungkin dapat dijadikan nara sumber atau dapat ditelusuri karya -karya tulisnya yang lain—yang mungkin terkait); (5) memperlihatkan kedudukan penelitian yang (akan) kita lakukan dalam sejarah perkembangan dan konteks ilmu pengetahuan atau teori tempat penelitian ini berada; (6) menungkapkan ide-ide dan pendekatan-pendekatan yang mungkin belum kita kenal sebelumya; (7) membuktikan keaslian penelitian (bahwa penelitian yang kita lakukan berbeda dengan penelitian-penelitian sebelumnya); dan (8) mampu menambah percaya diri kita pada topik yang kita pilih karena telah ada pihakpihak lain yang sebelumnya juga tertarik pada topik tersebut dan mereka telah mencurahkan tenaga, waktu dan biaya untuk meneliti topik tersebut [3].

2.1.1. Puri Tomat Hotel

Puri Tomat Hotel adalah salah satu badan usaha akomodasi yang menyediakan pelayanan jasa penginapan, *laundry*, *café* dan fasilitas jasa lainnya, dimana pelayanan yang diberikan diperuntukan kepada masyarakat umum, baik mereka yang datang untuk berlibur dan bermalam dihotel maupun mereka yang hanya menggunakan fasilitas yang ada dihotel tesebut. Puri Tomat Hotel berdiri pada tanggal 3 Oktober 2001, hotel ini berada dikawasan Bandung Kota yang terletak di Jalan Ir. H. Juanda No 240 Bandung. Puri Tomat Hotel menyediakan beberapa jenis tipe Kamar diantaranya: Flamboyan, Cemara, Tanjung, dan Pinus.

Kapasitas kamar yang dimiliki saat ini sebanyak 36 kamar. Dimana setiap jenis kamar memiliki fasilitas yang berbeda dan harga yang berbeda pula.

2.1.2. Visi dan Misi

Visi dan Misi adalah sasaran dan tujuan didirikannya suatu perusahaan atau instantsi. Setiap perusahaan atau instant pasti memiliki visi dan misi masing-masing untuk menjalankan aktivitas perusahaan tersebut. Adapun visi dan misi dari Puri Tomat Hotel.

2.1.2.1. Visi Hotel

Puri Tomat Hotel memiliki visi yang dijadikan sebagai acuan atas berdirinya hotel ini, yaitu

"menjadikan Puri Tomat Hotel bandung sebagai rumah kedua bagi para tamu".

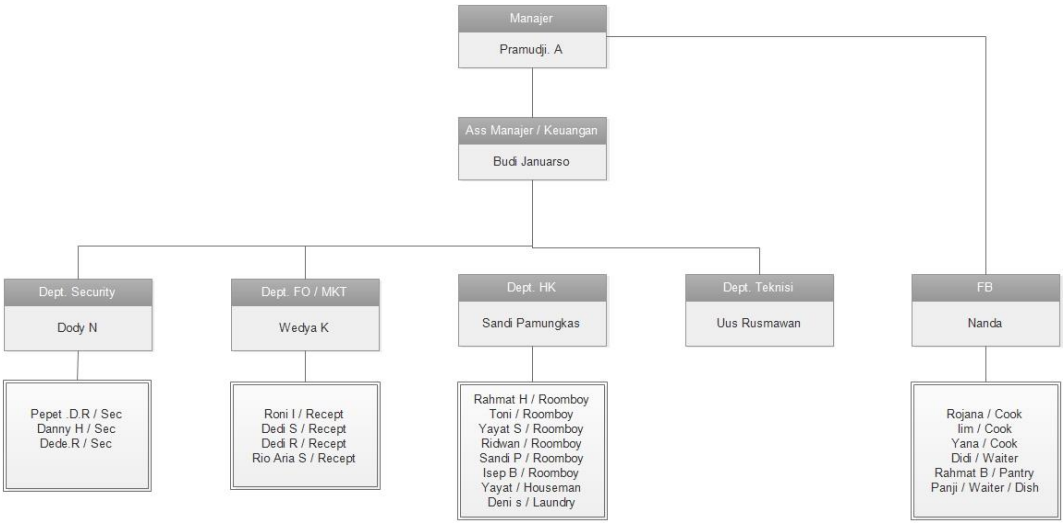
2.1.2.2. Misi Hotel

Misi Puri Tomat Hotel adalah senantiasa bertekad memberikan pelayanan yang bermutu tinggi guna memenuhi harapan para tamu melalui pengembangan sumber daya dan manajemen yang baik.

2.1.2.3. Logo Hotel

Gambar 2.1. Logo Puri Tomat Hotel

2.1.2.4. Struktur Organisasi



Gambar 2.2. Struktur Organisasi Puri Tomat Hotel

2.1.2.5. Deskripsi Tugas

Adapun deskripsi tugas dari struktur organisasi yang terlibat dalam pengelolaan hotel, yang menjadi bagian dari jabatannya. Jabatan itu sendiri mempunyai tugasnya masing-masing yang berperan sebagai acuan terhadap kinerja dalam bidangnya masing-masing.

Manajer Hotel mempunyai tugas sebagai berikut:

1. Mengatur dan meneliti pemesanan, penerimaan, pelayanan kamar, dan kegiatan pengurus/pelayan hotel
2. Mengawasi persiapan keamanan, kebun dan pemeliharaan barang-barang
3. Merencanakan dan mengawasi *café*/restaurant
4. Menilai dan memeriksa kepuasaan tamu
5. Memeriksa pembukuan dan kegiatan pembelian
6. Menetapkan pembuatan anggaran
7. Mengawasi pemilihan, pelatihan dan pengawasan terhadap staf
8. Menyediakan informasi wisata lokal dan mengatur transportasi untuk kunjungan/wisata kepada tamu

Asisten Manajer Hotel memiliki tugas sebagai berikut:

1. Memperkerjakan dan mengatur pelatihan staf baru

2. Mengawasi staf
3. Mengatur pelaksanaan departemen
4. Pemesanan persediaan hotel
5. Dapat mengisi posisi pekerja yang absen

Front Office memiliki tugas sebagai berikut:

1. Bertanggung jawab untuk menerima telpon masuk/keluar dengan mempergunakan standard greeting yang telah ditetapkan oleh hotel
2. Mempersiapkan telepon control sheet
3. Mengambil dan meneruskan pesan kepada tamu, maupun internal komunikasi
4. Melaksanakan wakeup call dan paging jika diperlukan
5. Menangani keluhan tamu
6. Mengetahui dan mampu menerangkan kepada tamu jam operasional , lokasi, promosi yang sedang berlangsung outlet food dan beverage, tipe kamar, fasilitas yang ada dihotel
7. Menjaga hubungan baik dengan tamu dengan menggunakan pendekatan professional, antara lain mempergunakan Bahasa inggris atau Bahasa asing lainnya
8. Menguasai dengan rinci prosedur dan kebijakan mengenai emergency dan melaksanakannya bila diperlukan
9. Memperhatikan kebersihan dan perawatan area kerja beserta peralatan

Teknisi Hotel memiliki tugas sebagai berikut:

1. Pemeliharaan dan perbaikan seluruh instalasi, alat mesin, bangunan dan fasilitas hotel lainnya
2. Penghematan energi dalam menggunakan segala keperluan
3. Menangani alat, mesin, dan instalasi lainnya yang menggunakan listrik , gas, dan air.

2.2. Pengertian Hotel

Secara harfiah, kata Hotel dulunya berasal dari kata *Hospitium* (Bahasa latin), artinya ruang tamu. Dalma jangka waktu lama kata hospitium mengalami proses perubahan pengertian dan untuk membedakan antara *Guest House* dengan *Mansion*

House (rumah besar) yang berkembang pada saat itu, maka rumah-rumah besar disebut dengan Hostel. Rumah-rumah besar atau hostel ini disewakan kepada masyarakat umum untuk menginap dan beristirahat sementara waktu, yang selama menginap para penginap dikoordinir oleh seorang *host*, dan semua tamu-tamu yang menginap harus tunduk kepada peraturan yang dibuat atau ditentukan oleh *host*. Sesuai dengan perkembangan dan tuntutan orang-orang yang ingin mendapatkan kepuasan, tidak suka dengan aturan atau peraturan yang terlalu banyak sebagaimana dalam hostel, dan kata hostel lambat laun mengalami perubahan. Huruf "s" pada kata hostel tersebut menghilang atau dihilangkan orang, sehingga kemudian kata hostel berubah menjadi Hotel seperti apa yang kita kenal sekarang.

Menurut Dirjen Pariwisata , Hotel adalah suatu jeniis akomodasi yang mempergunakan sebagian atau seluruh bangunan, untuk menyediakan jasa penginapan, makan dan minum, serta jasa lainnya bagi umum, yang dikelola secara komersial. Menurut surat Keputusan Menteri Perhubungan R.I No. PM 10/PW – 301/Phb. 77, tanggal 12 Desembet 1977: Hotel adalah suatu bentuk akomodasi yang dikelola secara komersial, disediakan bagi setiap orang untuk memperolah pelayanan penginapan, berikut makan dan minum. Menurut Webster, hotel adalah suatu bangunan atau suatu Lembaga yang menyediakan kamar untuk menginap, makan dan minum serta pelayanan lainnya untuk umum. [4]

2.2.1. Jenis – Jenis Hotel

Penentuan jenis hotel tidak terlepas dari kebutuhan pelanggan dan ciri atau sifat khas yang dimiliki wisatawan. Berdasarkan hal tersebut, dapat dilihat dari lokasi dimana hotel tersebut dibangun, sehingga dikelompokkan menjadi:

a. City Hotel

Hotel yang berlokasi diperkotaan, biasanya diperuntukkan bagi masyarakat yang bermaksuud untuk tinggal sementara(dalam jangka waktu pendek). *City Hotel* disebut juga sebagai transit hotel karena biasanya dihuni oleh para pelaku bisnis yang memanfaatkan fasilitas dan pelayanan bisnis yang disediakan oleh hotel tersebut.

b. Residential Hotel

Hotel yang berlokasi di daerah pinggiran kota besar yang jauh dari keramaian kota, tetapi mudah mencapai tempat-tempat kegiatan usaha. Hotel ini berlokasi di daerah-daerah tenang, terutama karena diperuntukkan bagi masyarakat yang ingin tinggal dalam jangka waktu yang lama.

c. Resort Hotel

Hotel yang berlokasi di daerah pegunungan (mountain hotel) atau di tepi pantai (beach hotel), di tepi danau atau di tepi aliran sungai. Hotel seperti ini terutama diperuntukkan bagi keluarga yang ingin beristirahat pada hari-hari libut atau bagi mereka yang ingin berekreasi.

d. Motel (Motor Hotel)

Hotel yang berlokasi di pinggiran atau disepanjang jalan raya yang menguhubungan satu kota dengan kota besar lainnya. Hotel ini diperuntukkan sebagai tempat istirahat sementara bagi mereka yang melakukan perjalanan dengan menggunakan kendaraan umum atau mobil sendiri. [5]

2.2.2. Pengertian Reservasi, *Check-in* dan *Check-Out*

Reservasi merupakan tamu Hotel yang memesan kamar jauh-jauh hari sebelum tamu tersebut menginap. *Check-In* merupakan tamu Hotel yang memesan kamar setelah mengurus administrasi kemudian menginap untuk beberapa hari. Sedangkan *Check-Out* adalah tamu Hotel yang telah menginap kemudian meninggalkan atau keluar Hotel setelah administrasi Hotel selesai. [4]

2.3. Landasan Teori

Landasan teori bertujuan untuk memberikan gambaran dari teori yang terkait dalam aplikasi yang akan dibangun. Landasan teori yang akan dibahas yaitu pengertian BlinkID SDK, API Midtrans, Java, MongoDB, Web service, Google Cloud Platform (GCP), RESTful API, Android, Metode yang digunakan dan Bahasa pemrograman yang digunakan dalam pembangungan aplikasi.

2.3.1. Sistem

Sistem berasal dari Bahasa Latin (*systēma*) dan Bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan Bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai

suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, dimana suatu model matematika seringkali bias dibuat.

Sistem juga merupakan satu kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesaatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara di mana yang berperan sebagai penggeraknya yaitu rakyat yang berada dinegara tersebut.

Kata sistem banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka. [6] [7]

2.3.2. Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem, penghubung, masukan, keluaran, pengolah dan sasaran atau tujuan.

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Jadi, dapat dibayangkan jika dalam suatu sistem ada subsistem yang tidak berjalan / berfungsi sebagaimana mestinya. Tentunya sistem tersebut tidak akan berjalan mulus atau mungkin juga sistem tersebut rusak sehingga dengan sendirinya tujuan sistem tersebut tidak tercapai.

2. Batas Sistem (*Boundary*)

Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.. Atau menuruta Azhar Susanto Batas Sistem merupakan garis abstraksi yang memisahkan antara sistem dan lingkungannya. Batas sistem ini bagi setiap orang sangat *relative* dan tergantung kepada tingkat pengetahuan dan situasi kondisi yang dirasakan oleh orang yang melihat sistem tersebut. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem nenunjukan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environments*)

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung Sistem

Penghubung sistem merupakan media penghubung anatara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke yang lainnya. Keluaran *output* dari satu subsistem akan menjadi masukan (*input*) untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan sistem adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan maintenance *input*) dan masukan sinyal (signal *input*). Maintanance input adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem *computer*, program adalah *maintenance*

input yang digunakan untuk mengoperasikan komputernya dan data adalah siganal *input* untuk diolah menjadi informasi.

6. Keluaran (*Output*) Sistem

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikan menjadi keluaran yang berguna dan sisi pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supersistem. Misalnya untuk sistem computer, panas yang dihaislkan adalah keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Pengolah (*Process*) Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-lpaoran lain yang dibutuhkan oleh manajemen.

8. Sasaran (*Objectives*) atau Tujuan (*Goal*)

Tujuan Sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. [8]

2.3.3. Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut.

1. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia melibatkan interaksi antara manusia dengan mesin disebut dengan human machine sistem atau ada yang menyebut dengan *man-machine system*. Sistem informasi

merupakan contoh *man-machine system*, karena menyangkut penggunaan computer yang berinteraksi dengan manusia.

2. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem *computer* adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
3. Sistem tertutup(*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataanya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed* (secara *relative* tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luanya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara *relative* tertutup karena sistem tertutup akan bekerja secara otomatis dna terbuka hanya untuk pengaruh yang baik saja. [6]

2.3.4. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah *Open Handset Alliance* (OHA), konsorsium

dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Sistem operasi android atau os android terdiri dari beberapa versi ,setiap versi android terbaru memiliki nama-nama unik tersendiri dan memiliki beberapa jenis kelebihan mulai dari tampilan hingga optomasi keamana,berikut daftar nama “os android” menurut versi .mulai dari nama os android pertama kali di keluarkan sampai dengan os android versi terbaru yang baru di keluarkan di tahun 2014.

1. Android versi 1.0

   Android versi 1.0, merupakan versi komersial pertama Android, dirilis pada tanggal 23 September 2008. Perangkat android pertama yang tersedia secara komersial adalah HTC Dream.
2. Android versi 1.1

   Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 dikenal dengan “petir four” meskipun nama ini tidak digunakan secara resmi. Versi ini memperbaiki beberapa bug, mengubah API android,dan menambahkan beberapa fitur baru
3. Android versi 1.5 Cupcake

   Pada 27 April 2009, android 1.5 dirilis, menggunakan kernel linux 2.6.27. versi ini adalah rilisan pertama yang secara resmi menggunakan nama kode berdasarkan nama-nama makanan pencuci mulut. Pembaruan versi ini termasuk beberapa fitur baru dan perubahan UI(user interface)
4. Android versi 1.6 Donut

   Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29.
5. Android versi 2.0 Eclair

   Dirilis pada tanggal 26 Oktober 2009, berbasis kernen Linux 2.6.29.
6. Android versi 2.2 Froyo

   Pada 20 mei 2010, SDK Android 2.2 (froyi, singakatan untuk frozen yogurt), yang berbasis kernel linux 2.6.32
7. Android versi 2.3 Gingerbread

   Pada tanggal 6 Desember 2010, berbasis kernel Linux 2.6.35

8. Android versi 3.0 Honeycomb

   Pada 22 Februari 2011, pembaruan pertama Android yang ditujukan hanya untuk computer tablet, berdasarkan kernel Linux 2.6.36. perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011.

9. Android versi 4.0 Ice Cream Sandwich

   SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1, dirilis pada 19 Oktober 2011.Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoretis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu.Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011.

10. Android versi 4.1 Jelly Bean

    Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus. Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

11. Android versi 4.4 Kitkat

    Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013.Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

12. Android versi 5.0 Lollipop

    Android 5.0, 5.0.2, 5.1 dan 5.1.1 "Lollipop" adalah versi stabil terbaru dari sistem operasi Android yang dikembangkan oleh Google, yang pada saat ini mencakup versi antara 5.0 dan 5.1. Diresmikan pada 25 Juni 2014 saat Google I / O, dan tersedia secara resmi melalui over-the-air (OTA) update pada tanggal 12 November 2014, untuk memilih perangkat yang menjalankan distribusi Android dilayani oleh Google (seperti perangkat Nexus dan Google Play edition). Kode sumbernya dibuat tersedia pada 3 November 2014. Salah satu perubahan yang paling menonjol dalam rilis Lollipop adalah user interface yang didesain ulang dan dibangun dengan yang dalam bahasa desain disebut sebagai "material design". Perubahan lain termasuk perbaikan pemberitahuan, yang dapat diakses dari lockscreen dan ditampilkan pada banner di bagian atas screen. Google juga membuat perubahan internal untuk platform, dengan Android Runtime (ART) secara resmi menggantikan Dalvik untuk meningkatkan kinerja aplikasi, dan dengan perubahan yang ditujukan untuk meningkatkan dan mengoptimalkan penggunaan baterai, yang dikenal secara internal sebagai Project Volta.

13. Android versi 6.0 Marshmallow

    Android 6.0 dan 6.0.1 "Marshmallow" merupakan pemutakhiran yang akan datang untuk sistem operasi telepon genggam Android, kemungkinan besar akan dirilis pada Q3 2015 ("sementara dijadwalkan untuk September"), dengan pratayang ketiga dan terakhir dirilis pada tanggal 17 Agustus 2015. Pertama diperkenalkan di Google I/O pada tanggal 28 Mei 2015, Marshmallow terutama akan berfokus pada perbaikan inkremental dan penambahan fitur lainnya. Pratayang pengembang Android "M" dirilis pada tanggal 28 Mei 2015, untuk telepon genggam Nexus 5 dan Nexus 6, tablet Nexus 9, dan set-top box Nexus Player, di bawah nomor bentukan MPZ44Q. Pratayang pengembang ketiga (MPA44G) dirilis pada tanggal 17 Agustus 2015 untuk Nexus 5, Nexus 6, Nexus 9 dan perangkat Nexus Player,dan diperbarui ke MPA44I yang membawa perbaikan yang berhubungan dengan profil Android for Work.

14. Android versi 7.0 Nougat

    Android "Nougat" (kode nama N dalam pengembangan) adalah rilis 7.0 besar dari sistem operasi Android. Ini pertama kali dirilis sebagai pratinjau pengembang pada tanggal 9 Maret 2016, dengan gambar pabrik untuk perangkat Nexus saat ini, serta dengan "Program Beta Beta" baru yang memungkinkan perangkat yang didukung ditingkatkan versinya ke versi Android Nougat melalui over-the -air update. Rilis terakhir adalah pada tanggal 22 Agustus 2016. Pratinjau akhir pembuatannya dirilis pada tanggal 18 Juli 2016, dengan nomor bangunan NPD90G.

15. Android versi 8.0 oreo

    Android Oreo adalah rilis utama ke 8 dari sistem operasi Android. Ini pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017, dengan gambar pabrik untuk perangkat Nexus dan Pixel saat ini. Pratinjau pengembang terakhir dirilis pada tanggal 24 Juli 2017, dengan rilis stabil yang diharapkan pada bulan Agustus atau September 2017.

Alasan kenapa menggunakan metode android dalam penelitian ini adalah kerena aplikasi yang digunakan sangat efisien dan mudah digunakan oleh user. Setiap orang sekarang hampir semua sudah memilki smartphone untuk digunakan, jadi ini adalah hal yang terbaik kalua membuatnya di android. [9] [10][18]

2.3.5. Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ* IDEA . Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android.

Alasan menggunakan android studio karena sangat membantu pengguna atau pembuat program untuk membuat aplikasi android. Karena fitur-fitur yang mudah

digunakan dan banyak referensi dari berbagai macam yang bias membantu pembuat program membuat aplikasi. [11]

2.3.6. MongoDB

MongoDB adalah salah satu produk database noSQL *Open Source* yang menggunakan struktur data JSON untuk menyimpan datanya. MongoDB adalah merupakan database noSQL yang paling populer di internet. MongoDB sering dipakai untuk aplikasi berbasis *Cloud*, *Grid Computing*, atau *Big Data*. Dalam konsep MongoDB tidak ada yang namanya tabel, kolom ataupun baris yang ada hanyalah *collection* (ibaratnya tabel), *document* (ibaratnya *record*). Data modelnya sendiri disebut BSON dengan struktur mirip dengan JSON.

Alasan menggunakan MongoDB karena performa yang ditawarkan oleh MongoDB sangat cepat yang sebabkan oleh memchaced dan format dokumennya berbentuk seperti JSON, ada fitur untuk membackup data secara *realtime*, proses CRUD terasa sangat ringan.

2.3.7. SDK (Software Development Kit)

SDK (Software Development Kit) adalah alat bantu dan API (*Application Programming Interface*) yang berguna untuk mengembangkan berbagai aplikasi platform android dengan bahasa programa Java dan juga merupakan pulgin dari *eclipse*. Aplikasi ini juga sangat support untuk mengubah handphone menjadi teraplikasi Android di dalamnya dengan menggunakan emulator yang ada. SDK Android terdiri dari: (1) *debugger*, (2)*libraries*, (3)*handset emulator*, (4)contoh kode, (5)dokumentasi, dan (6)*tutorial*

Android sudah mendukung arsitektur pada Linux (distribusi Linux apapun untuk *desktop modern*), Mac OS X 10.4.8 atau lebih, Windows XP atau Vista. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin *Android Development Tools* (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML.

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim *Android Developer Challenge* sengaja mengirimkan email ke semua pendatang di *Android Developer Challenge* untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman download pribadi. Pada tanggal 18 Agustus 2008, Android SDK 0.9 beta dirilis. Rilis ini menyediakan API yang diperbarui dan diperluas, perbaikan pada alat-alat pengembangan dan desain terbaru untuk layar awal. Petunjuk untuk meng-upgrade SDK sudah tersedia pada rilis sebelumnya. Pada tanggal 23 September 2008, Android 1.0 SDK telah dirilis. Pada tanggal 9 Maret 2009, Google merilis versi 1.1 untuk telepon seluler Android.Pada pertengahan Mei 2009, Google merilis versi 1.5 (Cupcake) pada sistem operasi Android dan SDK. Pembaruan ini termasuk banyak fitur baru seperti perekaman video, dukungan untuk bluetooth, sistem *keyboard* pada layar dan pengenalan suara. Rilis ini juga membuka *AppWidget framework* kepada para pengembang yang memungkinkan orang untuk membuat *widget* sendiri pada halaman home. Pada September 2009 versi 1.6 (Donut) dirilis yang menampilkan hasil pencarian yang lebih baik dan penggunaan indikator baterai. [11]

2.3.8. BlinkID SDK

BlinkID SDK untuk Android adalah SDK yang memungkinkan untuk melakukan pemindaian berbagai kartu *ID* di suatu aplikasi. Cukup mengitegrasikan SDK ke aplikasi dan dapat memanfaatkan fitur pemindaiain untuk berbagai dokumen. Dokumen yang didukung BlinkID mendukung semua dokumen yang mengandung MRZ(*Machine Readable Zone*), seperti *paspor* dan *ID*. Pemindaian semua dokumen lainnya diaktifkan dengan templete OCR khusus. BlinkID membutuhkan android versi 4.1 sebagai versi android minimum. Untuk kinerja dan kompatibilitas terbaik.

Alasan menggunakan BlinkID karena pemindaian kartu identitas untuk negara Indonesia didukung yang membantu pembuat aplikasi atau programmer untuk

membangun sebuah aplikasi, dan dapat mengekstrak gambar serta tanda tangan tersedia.

2.3.9. Google Cloud Platform (GCP)

Google Cloud Platform merupakan layanan *public cloud* computing dari *google* yang terdiri dari beragam layanan. Platform dari google ini menyediakan beragam layanan hosting mulai untuk komputasi, *storage* dan *application development* yang berjalan pada *hardware google*. Google cloud platform service dapat diakses oleh pengembang software, *administrator cloud,* dan *professional* IT lainnya menggunakan *internet public* atau melalui jaringan *dedicated*.

Layanan cloud computing yang diberikan oleh Google Cloud Platform antaralain:

1. *Google Compute Engine*: Layanan *infrastructur as a service* (Iaas) yang menyediakan penguna akan *virtual machine* yang bisa dibuat secara instan untuk *workload hosting*.
2. *Google App Engine*: Layanan *platform as a service* (PaaS) menawarkan pengembang *software* untuk mengakses *Google hosting* yang mudah terukur. Pengembang dapat juga menggunakan *software developer kit* (SDK) untuk mengembangkan produk *software* yang berjalan pada *Google App Engine*.
3. *Google Cloud Storage*: *Platform cloud storage* atau penyimpanan berbasi cloud yang didesain untuk menyimpan data yang besar, tidak terstruktur. Google juga menawarkan pilihan penyimpanan database termasuk *Cloud Datastore* untuk penyimpananan NoSQL non-relational, penyimpanan *Cloud* SQL untuk MySQL, dan database *Google Cloud Bigtable*.
4. *Google Container Engine*: Sistem manajemen dan pengukuran untuk *Docker container* yang berjalan didalam *Google public cloud*. *Google Container Engine* berbasis pada *engine Google Kubernetes*.

Google Cloud Platform juga menyediakan layanan *cloud* seperti pemrosesan dan analisis data, seperti Google *BigQuery* untuk SQL yang digunakan untuk

memproses set data multi terabyte. Sebagai tambahan, *Google Cloud Dataflow* merupakan layanan pemrosesan data yang khusus untuk analisis, *extrct transform and load* (ETL), dan *real-time computational projects*.

Alasan mengapa menggunakan Google Cloud Platform pada penelitian ini karena dengan menggunakan GCP pengguna dapat mengakses data yang telah disimpan di Google Cloud Platform dengan mudah selama ada akses internet [12]

2.3.10. RESTful API

REST (*Representational State Transfer*) merupakan standar arsitektur komunikasi yang menggunakan protocol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000.

Pada arsitektur REST, REST server menyediakan *resources* dan *REST client* mengakses dan menampilkan resources tersebut untuk penggunaan selanjutnya. Setiap *resources* diidentifikasi oleh URIs(*Universal Resource Identifiers*) atau global ID. *Resources* tersebut direpresentasikan dalam bentuk format teks, JSON, atau XML. Pada umumnya format menggunakan JSON dan XML. RESTful API memiliki 4 komponen penting di dalamnya diantaranya adalah: (1) URL *Design*, (2) HTTP *Verbs*, (3) HTTP *Responese Code*, dan (4) *Format Response*.

Alasan menggunakan RESTful dalam penelitian ini karena lebih sederhana untuk dikembangkan , menjadikan aplikasi yang memiliki performa yang baik, cepat dan mudah untuk dikembangkan terutama dalam pertukaran dan komunikasi data. [12]

2.3.11. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C

dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin *Virtual Java* (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

Alasan menggunakan aplikasi ini karena aplikasi ini sangat mudah digunakan bila kita memahami konsep – konsep yang ada dan fitur yang terdapat dalam aplikasi tersebut. Java sangat sederhana dan aman digunakan oleh usernya. Aplikasi ini berorientasi objek, karena itu sangat mudah digunakan oleh *user* karena terdapat berbagai fitur yang membantu user untuk membuat aplikasi java.

2.3.12. API (Application Programming Interface)

Sekumpulan perintah, fungsi, serta protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi.

Alasan menggunakan API karena API salah satu alat penghubung agar dapat berinteraksi dengan sistem operasi. Dalam API terdapat banyak fungsi atau perintah untuk menggantikan Bahasa yang digunakan *system calls* dengan Bahasa yang lebih terstruktur dan mudah dimengerti oleh *programmer*.

2.3.13. Midtrans API

Midtrans API adalah *channel* yang menerima pembayaran yang sangat bervariatif seperti: Kartu kredit maupun debit, BCA Klikpay, CIMB Clicks, XL Tunai, Mandiri Clickpay, T-Cash, transfer bank yang biasa, minimarket yang memang sudah terintegrasi lewat VT_Link, VT_Web, VT-Direc.

Alasan menggunakan Midtrans API karena mudah untuk diintegrasikan keberbagai platform yang akan dibuat oleh *programmer*, dapat memonitor transaksi secara *real time* dan mendapatkan laporan transaksi secara menyeluruh untuk penggunanya.

2.3.14. OOP (Object Oriented Programming)

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. 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. [13]

1. *Encapsulation* (Pengkapsulan)

Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat

mengaksesnya. Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

2. *Inheritance* (Pewarisan)

Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya. Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di anatara kelas yang mempunyai hubungan secara hirarki. Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya. Kelas Objek dapat didefinisikan atribut dan *service* dari kelas Objek lainnya. *Inheritance* menggambarkan generalisasi sebuah kelas.

3. *Polymorphism* (Polimorfisme)

Polimorfisme yaitu konsep yang menyatakan bahwa seuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama. Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

Alasan menggunakan OOP metode dalam penelitian ini adalah karena aplikasi yang dibuat dapat terstruktur dan lebih rapih. Dan dapat lebih mudah untuk dianalisa program yang akan dibangun. OOP juga membantu kita membuat dan membaca kode yang dibuat.

2.3.17. OOAD (*Object Oriented Analysis and Design*)

Suatu pendekatan rekayasa perangkat lunak dari sebuah sistem yang terdiri dari sekelompok objek yang saling berinteraksi, dan setiap objek itu mewakili

beberapa entitas. Yang ditandai dengan adanya sebuah kelas, elemen data dan perilaku dari objek tersebut.

1. Objek

Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah. Semua objek mempunyai identitas yang berbeda dengan lainnya. Istilah identitas berarti bahwa objek dibedakan oleh sifat yang melekat dan bukan dengan uraian sifat yang dimilikinya. Contohnya : kembar identik, walaupun mereka nampak seperti sama, tetapi merupakan dua orang yang berbeda.

2. Kelas

Suatu object class menggambarkan kumpulan dari objek yang mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum. Contoh : Orang, perusahaan , binatang, proses adalah objek.

Setiap orang mempunyai umur, IQ, dan mungkin pekerjaan. Setiap proses mempunyai pemilik, prioritas, list dari sumber daya yang dibutuhkan. Objek dan object class sering sama sebagai benda dalam deskripsi masalah.

3. Diagram Objek

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program.

Kenapa memilih metode OOAD metode dalam penelitian ini karena metode OOAD mempunyai relasi obyek dengan entitas umumnya, dapat di mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam sistem . Pendekatan visual, membuat metode ini mudah dimengerti oleh user maupun programmer. [14]

2.3.18. UML (*Unified Modelling Language*)

UML adalah bahasa untuk menspesifikasi,memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak)dari sistem perangkat lunak,seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object.UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. [15] [16]

1. *Use Case Diagram*

Menggambarkan sejumlah *external* actors dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam activity diagrams. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem.

2. *Class Diagram*

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan class yang lain), *specialed* (satu class merupakan spesialisasi dari class lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram*.

3. State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

4. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

5. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

6. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

7. Component Diagram

Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa *source code*, komponent biner, atau *executable component*. Sebuah komponent berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

8. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

2.3.19. Pengujian Alpha

Pengujian Alpha adalah salah satu strategi pengujian perangkat lunak yang paling umum digunakan dalam pengembangan perangkat lunak, hal ini khusus digunakan oleh organisasi pengembangan produk dengan tujuan agar *system* yang dikembangkan terhindar dari cacat atau kegagalan penggunaan.

Alpha testing tujuannya untuk identifikasi dan menghilangkan sebanyak mungkin masalah sebelum akhirnya sampai ke *user*, dilakukan setelah *software* jadi oleh orang-orang yang tidak terlibat dalam pengembangan dan memang ahli dibidangnya. Terdapat formulir resmi evaluasi.

1. Tes ini berlangsung di situs pengembang. Pengembang mengamati penggunaan aplikasi oleh pengguna selanjutnya pengguna mencatat temuan yang terjadi dari kecacatan aplikasi.
2. Pengujian Alpha adalah pengujian dari aplikasi saat pembangunan adalah tentang untuk menyelesaikan. perubahan desain kecil masih dapat dibuat sebagai hasil dari pengujian alpha.

Pengujian Alpha adalah pengujian akhir sebelum perangkat lunak ini diluncurkan untuk pengguna secara umum. Ini memiliki dua fase:

1. Pada tahap pertama dari pengujian alpha, perangkat lunak diuji oleh pengembang di lingkungan *internal developer*. Mereka menggunakan perangkat lunak *debugger*, atau *debugger hardware-assisted*. Tujuannya adalah untuk menangkap *bug* dengan cepat.
2. Pada tahap kedua pengujian alpha, software ini diserahkan kepada resepsionis hotel, untuk pengujian tambahan dalam lingkungan yang mirip dengan penggunaan yang dimaksudkan. Hal ini untuk mensimulasikan suasana atau lingkungan pengujian yang sebenarnya sehingga ketika *system* tersebut dipasang, sudah tidak terjadi kegagalan maupun cacat *system* secara real.

Kenapa dilakukan pengujian secara aplha karena pengujian ini adalah untuk mengetahui sampai sejauh mana aplikasi dapat berjalan, untuk mengetahui dimana titik kelemahan pada aplikasi tersebut. Apakah terjadi *bug* atau kecacatan pada sistem yang membuat aplikasi ini tidak dapat bekerja secara sempurna.

2.3.20. Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, Dimana pengujian dulakukan secara langsung terhadap pengguna, biasanya menggunakan kuisioner mengenai tanggapan pengguna atas perangkat lunak yang telah dibangun. Metode penilaian pengujuan yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna.