BAB II
LANDASAN TEORI

2.1 Badan Pusat Statistik

Badan Pusat Statistik adalah Lembaga Pemerintah Non Kementerian yang bertanggung jawab langsung kepada Presiden. Sebelumnya, BPS merupakan Biro Pusat Statistik, yang dibentuk berdasarkan UU Nomor 6 Tahun 1960 tentang Sensus dan UU Nomer 7 Tahun 1960 tentang Statistik. Sebagai pengganti kedua UU tersebut ditetapkan UU Nomor 16 Tahun 1997 tentang Statistik. Berdasarkan UU ini yang ditindaklanjuti dengan peraturan perundangan di bawahnya, secara formal nama Biro Pusat Statistik diganti menjadi Badan Pusat Statistik [9].

Berdasarkan Undang-Undang yang telah disebutkan di atas, peranan yang harus dijalankan oleh BPS adalah sebagai berikut [9]:

1. Menyediakan kebutuhan data bagi pemerintah dan masyarakat. Data ini didapatkan dari sensus atau survey yang dilakukan sendiri dan juga dari departemen atau lembaga pemerintahan lainnya sebagai data sekunder
2. Membantu kegiatan statistik di departemen, lembaga pemerintah atau institusi lainnya, dalam membangun sistem perstatistikan Nasional.
3. Mengembangkan dan mempromosikan standar teknik dan metodologi statistik, dan menyediakan pelayanan pada bidang pendidikan dan pelatihan statistik.
4. Membangun kerjasama dengan institusi internasional dan negara lain untuk kepentingan perkembangan statistik Indonesia.

BPS mempunyai kantor di setiap Kabupaten/Kota yang dikepalai oleh seorang kepala BPS Kabupaten/Kota dan dibantu oleh 1 orang Kepala Sub Bagian Tata Usaha, 5 orang kepala seksi yaitu Kepala Seksi Statistik Produksi, Kepala Seksi Statistik Sosial, Kepala Seksi Statistik Neraca dan Nerwilis, Kepala Seksi

Statistik Distribusi serta Kepala Seksi Statistik Integrasi Pengolahan dan Diseminasi Statistik (IPDS) beserta beberapa orang staff dan beberapa orang Koordinator Statistik Kecamatan (KSK).

BPS di setiap Kabupaten/Kota memiliki fungsi yang sama. Setiap BPS Kabupaten/Kota dibawahi oleh BPS Provinsi dan setiap BPS Provinsi dibawahi oleh BPS Pusat. Semua BPS melakukan survei ubinan yang dilaksanakan setiap periode setiap tahunnya termasuk BPS Kabupaten Sumedang. BPS Kabupaten Sumedang bertempat di Jalan Karapyak No. 61 Kelurahan Situ, Kecamatan Sumedang Utara, Kabupaten Sumedang.

2.1.1 Sejarah Badan Pusat Statistik

Masa Pemerintahan Hindia Belanda

Pada bulan Februari 1920, Kantor Statistik pertama kali didirikan oleh Direktur Pertanian, Kerajinan dan Perdagangan (*Directure Vand Landbow Nijeverheinden Handed*) dan bertempat di Bogor. Kantor ini diserahi tugas untuk mengelola dan mempublikasikan data statistik.

Pada bulan Maret 1923, dibentuk suatu komisi untuk statistik yang anggotanya merupakan wakil dari tiap-tiap departemen. Komisi tersebut diberi tugas untuk merencanakan tindakan-tindakan yang mengarah sejauh mungkin untuk mencapai kesatuan dalam kegiatan di bidang statistik di Indonesia.

Pada tanggal 24 September 1924, nama lembaga tersebut diganti dengan nama *Central Kantor Voor de Statistik* (CKS) atau kantor statistik dan dipindahkan ke Jakarta. Bersama dengan itu beralih pula pekerjaan mekanisme Statistik Perdagangan yang semula dilakukan oleh kantor *Invelor Vitvoer en Accijnsen* (IVA) yang sekarang disebut kantor Bea dan Cukai.

Masa Pemerintahan Jepang

Pada bulan Juni 1994, pemerintah Jepang baru mengaktifkan kembali kegiatan Statistik yang utamanya diarahkan untuk memenuhi kebutuhan perang atau militer. Pada masa ini Central Kantor *Voor de Statistik* (CKS) diganti namanya menjadi *Shomubu Chosasitsu Gunseikanbu*.

Masa Kemerdekaan Republik Indonesia

Setelah Proklamasi Kemerdekaan Indonesia RI tanggal 17 Agustus 1945, kegiatan Statistik ditangani oleh lembaga atau instansi baru sesuai dengan suasana kemerdekaan yaitu KAPPURI (Kantor Penyelidik Perangkaan Umum Republik Indonesia) dipindahkan ke Yogyakarta sebagai sekuen dari perjanjian Linggarjati. Sementara itu pemerintah Belanda (NICA) di Jakarta mengaktifkan kembali *Central Kantor Voor de Statistik* (CKS). Berdasarkan surat edaran kementerian kemakmuran tanggal 12 Juni 1950 Nomor 219/S.C, KAPPURI (Kantor Penyelidik Perangkaan Umum Republik Indonesia) dan *Central Voor de Statistik* (CKS) dilebur menjadi Kantor Pusat Statistik (KPS) dan berada dibawah dan bertanggung jawab menteri Kemakmuran.

Dengan surat Menteri Perekonomian tanggal 1 Maret 1952 Nomor p/44, Lembaga Kantor Pusat Statistik (KPS) berada dibawah dan bertanggung jawab menteri Perekonomian. Selanjutnya keputusan Menteri Perekonomian Tanggal 24 Desember 1953 Nomor: 18.099/M, KPS dibagi menjadi dua bagian yaitu bagian *Research* yang disebut Afdeling A dan bagian penyelenggaraan tata usaha yang disebut Afdeling B. Dengan keputusan Presiden RI Nomor 131 tahun 1957, kemerdekaan perekonomian dipecah menjadi kementerian Perdagangan dan Kementerian Perindustrian. Untuk selanjutnya keputusan Presiden RI Nomor 172, terhitung tanggal 1 Juni 1957 Kantor Pusat Statistik (KPS) diubah menjadi Biro Pusat Statistik yang semula menjadi tanggung jawab dan wewenang berada dibawah perdana menteri.

Masa Orde Baru Sampai Sekarang

Perencanaan dan evaluasi pembangunan maka untuk mendapatkan statistik yang handal, lengkap, tepat, akurat dan terpercaya mulai diadakan pembenahan

organisasi Biro Pusat Statistik. Dalam masa orde baru ini Badan Pusat Statistik telah mengalami empat kali perubahan struktur organisasi.

Tahun 1968, ditetapkan peraturan Pemerintah Nomor 16 Tahun 1968 yaitu tentang mengatur organisasi dan tata kerja di pusat dan di daerah. Tahun 1980 peraturan Pemerintah nomor 6 tahun 1980 tentang organisasi sebagai pengganti peraturan Pemerintah Nomor 16 tahun 1968. Berdasarkan peraturan pemerintah Nomor 6 tahun 1980 di tiap propinsi terdapat perwakilan BPS.

Pada tanggal 17 Juni 1998 dengan keputusan Presiden Republik Indonesia Nomor 86 tahun 1998 ditetapkan Badan Pusat Statistik sekaligus mengatur tata kerja dan struktur organisasi BPS yang baru. Saat ini Kantor BPS sudah terdapat di setiap Kabupaten/Kota.

2.1.2 Logo Badan Pusat Statistik

Gambar 3 berikut ini merupakan logo yang digunakan oleh Badan Pusat Statistik di seluruh Indonesia.



Gambar 2.1 Logo Badan Pusat Statistik

Logo BPS terdiri dari 3 warna yaitu biru, hijau dan oranye yang mewakili 3 sensus besar yang dilakukan oleh BPS setiap 10 tahun sekali. Warna biru mewakili Sensus Penduduk yang dilaksanakan setiap sepuluh tahun sekali di tahun yang berakhiran angka 0. Warna hijau mewakili Sensus Pertanian yang

dilaksanakan setiap sepuluh tahun sekali di tahun yang berakhiran 3. Warna Oranye mewakili Sensus Ekonomi yang dilaksanakan setiap sepuluh tahun sekali di tahun yang berakhiran angka 6.

2.1.3 Visi, Misi dan *Core Values* Badan Pusat statistik

Setiap organisasi tentu punya visi dan misi yang jelas sehingga organisasi yang akan dibentuk mempunyai tujuan yang jelas. Fungsi dari visi dan misi organisasi adalah untuk membuat jalannya organisasi tetap fokus dan terkendali. Pernyataan visi dan misi yang dibuat akan menjelaskan peran dari organisasi yang dibentuk. Berikut adalah Visi, Misi dan Core Values dari Badan Pusat Statistik:

a. Visi: Pelopor data statistik terpercaya untuk semua

b. Misi:

1. Menyediakan data statistik berkualitas melalui kegiatan statistik yang terintegrasi dan berstandar nasional maupun internasional.
2. Memperkuat Sistem Statistik Nasional yang berkesinambungan melalui pembinaan dan koordinasi di bidang statistik
3. Membangun insan statistik yang profesional, berintegritas dan amanah untuk kemajuan perstatistikan

c. *Core Values* (Nilai-Nilai Inti)

Core values (nilai–nilai inti) BPS merupakan pondasi yang kokoh untuk membangun jati diri dan penuntun perilaku setiap insan BPS dalam melaksanakan tugas. Nilai-nilai Inti BPS terdiri dari:

1. PROFESIONAL

a. Kompeten yaitu mempunyai keahlian dalam bidang tugas yang diemban
b. Efektif yaitu memberikan hasil maksimal
c. Efisien yaitu mengerjakan setiap tugas secara produktif, dengan sumber daya minimal

d. Inovatif yaitu selalu melakukan pembaruan dan/atau penyempurnaan melalui proses pembelajaran diri secara terus menerus
e. Sistemik yaitu meyakini bahwa setiap pekerjaan mempunyai tata urutan proses pekerjaan yang satu menjadi bagian tidak terpisahkan dari pekerjaan yang lain.

2. INTEGRITAS

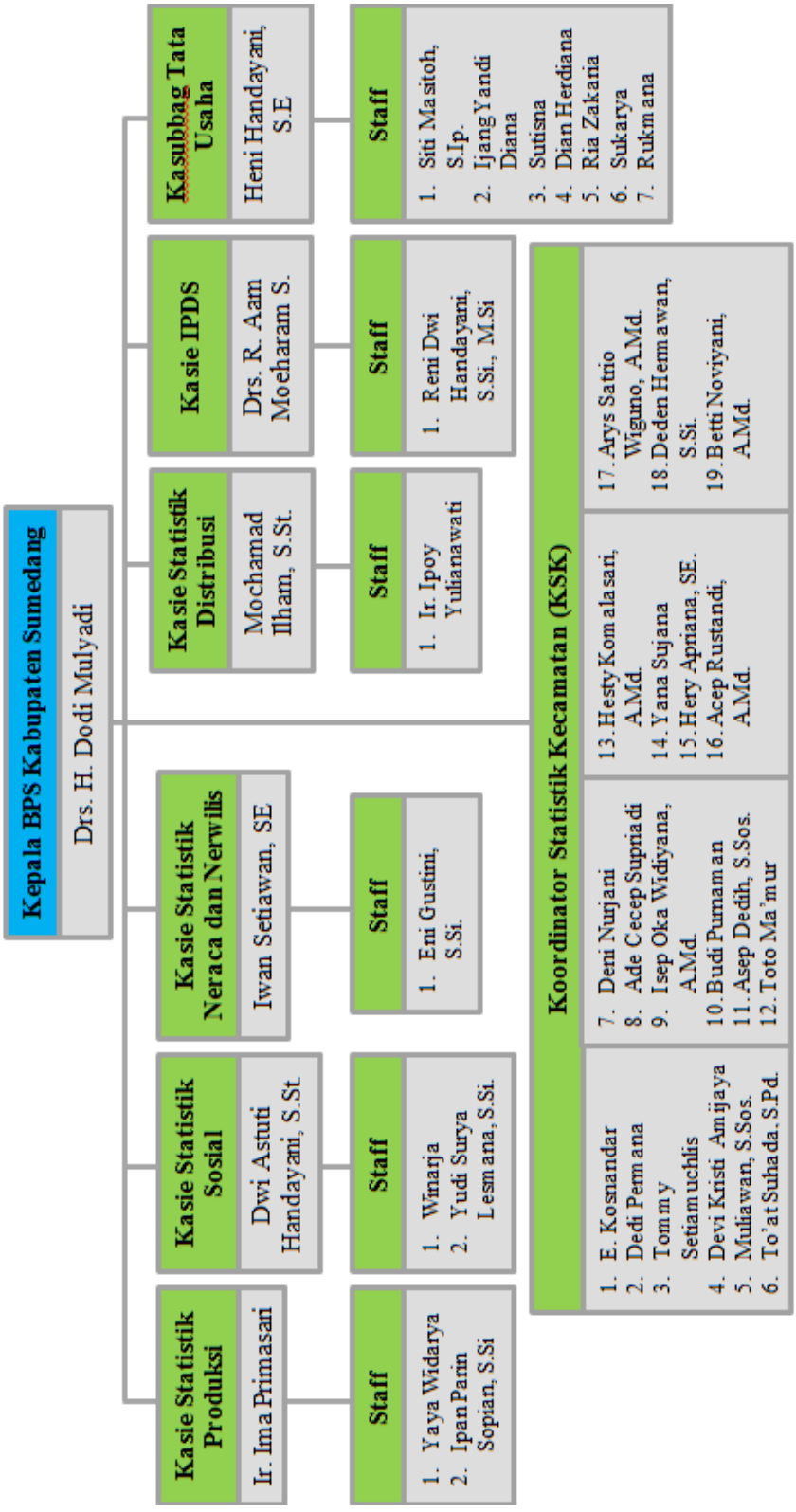
a. Dedikasi yaitu memiliki pengabdian yang tinggi terhadap profesi yang diemban dan institusi
b. Disiplin yaitu melaksanakan pekerjaan sesuai dengan ketentuan yang telah ditetapkan
c. Konsisten yaitu satunya kata dengan perbuatan
d. Terbuka yaitu menghargai ide, saran, pendapat, masukan, dan kritik dari berbagai pihak
e. Akuntabel yaitu bertanggung jawab dan setiap langkahnya terukur

3. AMANAH

a. Terpercaya yaitu melaksanakan pekerjaan sesuai dengan ketentuan, yang tidak hanya didasarkan pada logika tetapi juga sekaligus menyentuh dimensi mental spiritual
b. Jujur yaitu melaksanakan semua pekerjaan dengan tidak menyimpang dari prinsip moralitas
c. Tulus yaitu melaksanakan tugas tanpa pamrih, menghindari konflik kepentingan (pribadi, kelompok, dan golongan), serta mendedikasikan semua tugas untuk perlindungan kehidupan manusia, sebagai amal ibadah atau perbuatan untuk Tuhan Yang Maha Esa
d. Adil yaitu menempatkan sesuatu secara berkeadilan dan memberikan haknya

2.1.4 Struktur Organisasi Badan Pusat Statistik Kabupaten Sumedang

Berikut ini Struktur Organisasi di Badan Pusat Statistik Kabupaten Sumedang :



Gambar 2.2 Struktur Organisasi Badan Pusat statistik Kabupaten Sumedang

Setiap BPS Kabupaten/Kota dikepalai oleh seorang Kepala BPS Kabupaten/Kota. Seorang Kepala BPS Kabupaten/Kota Membawahi 5 Orang Kepala Seksi (Kasie), 1 Orang Kepala Sub Bagian (Kasubbag) Tata Usaha dan Koordinator Statistik Kecamatan yang bertanggung jawab langsung ke Kepala BPS Kabupaten/Kota. Kasie dan Kasubag membawahi beberapa staff yang membantu dalam menyelesaikan beberapa tugasnya. Seorang Koordinator Statistik Kecamatan (KSK) memegang satu kecamatan. Namun dikarenakan jumlah karyawan yang terbatas, tidak jarang KSK bekerja di lebih dari satu Kecamatan.

Salah satu survei yang diadakan di BPS salah satunya yaitu survei ubinan. KSK di setiap kecamatan dibantu oleh staff, melaksanakan survei ubinan di setiap kecamatan. Survei ubinan merupakan salah satu data andalan BPS karena hampir semua BPS di kabupaten dan Kota turut melaksanakan survei ubinan.

2.2 Survei Ubinan

Pengumpulan data produktivitas tanaman pangan dilakukan secara sampel melalui Survei Ubinan dengan pendekatan rumah tangga. Tanaman padi meliputi padi sawah dan padi ladang, sedangkan tanaman palawija meliputi jagung, kedelai, kacang tanah, kacang hijau, ubi kayu dan ubi jalar. Pengumpulan data produktivitas tanaman pangan menerapkan metode pengukuran langsung pada plot ubinan terpilih dan metode wawancara dengan petani sampel untuk mengumpulkan data beberapa karakteristik yang berkaitan dengan produktivitas seperti penggunaan pupuk, benih, pengairan, pestisida, cara penanaman, dan sebagainya [1].

Survei Ubinan dilakukan secara rutin setiap tahun dalam tiga periode, yaitu subround/SR I (periode Januari-April), SR II (periode Mei-Agustus), dan SR III (periode September-Desember). Pemutakhiran rumah tangga dan pendaftaran petak di setiap bidang lahan untuk suatu subround dilakukan pada bulan terakhir

subround sebelumnya. Unit pencacahan Survei Ubinan adalah rumah tangga usaha tanaman pangan yang melakukan panen pada subround tertentu [1].

Untuk mendapatkan unit pencacahan Survei Ubinan, ada beberapa proses yang harus dilakukan, berikut prosesnya:

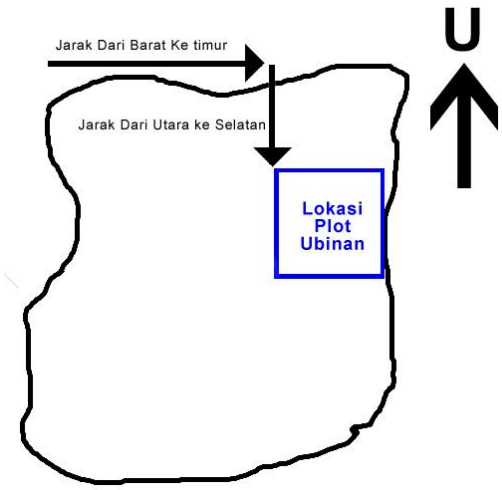
1. BPS Kabupaten/Kota menerima daftar blok sensus sample survei ubinan dari BPS Pusat melalui BPS Provinsi.
2. KSK melakukan pendaftaran rumah tangga pertanian di blok sensus yang termasuk dalam daftar sample berikut karakteristik komoditas tanaman pangan berikut perkiraan waktu panen pada batas subround yang ditentukan.
3. Berdasarkan daftaran sample dari KSK, staff BPS Kabupaten/Kota melakukan perhitungan unit pencacahan yang kemudian dilakukan observasi pengukuran hasil panen.

Setelah tersedia daftar unit pencacahan (Daftar SUB-DS), KSK melakukan pendataan Survei Ubinan secara langsung ke unit pencacahan. Berikut ini garis besar proses pendataan Survei Ubinan di lapangan:

1. Menghitung jumlah petak yang dikuasai oleh petani dan ditanami tanaman pangan terpilih. Kemudian memberikan nomor urut pada jumlah petak di seluruh bidang yang dikuasai dan ditanami tanaman pangan terpilih.
2. Jumlah petak dikalikan dengan angka random yang telah ditentukan oleh BPS Pusat, dan didapatkan nomor urut petak terpilih yang akan dilakukan pengukuran langsung.
3. Menggambarkan petak di seluruh bidang yang dikuasai dan ditanami tanaman pangan terpilih pada tempat yang sudah disediakan untuk mengetahui arah mata anginnya.
4. Menghitung panjang sisi petak dengan menggunakan langkah kaki biasa dari arah barat ke timur dan dari utara ke selatan.
5. Melakukan picingan mata pada tabel angka random untuk mendapatkan lima angka. Angka pertama untuk menentukan halaman tabel angka random yang akan digunakan (halaman 1 atau halaman 2), angka kedua dan ketiga untuk

menentukan baris pada tabel angka random (baris 1 sampai baris 35), angka keempat dan kelima untuk menentukan kolom pada tabel angka random (kolom 1 sampai kolom 25).

6. Dari halaman, baris dan kolom yang telah didapat, mulailah dicari empat angka untuk menentukan posisi panen yang akan diukur hasilnya. Angka pertama dan kedua untuk menentukan jarak dari barat ke timur nya. Angka ketiga dan keempat untuk menentukan jarak dari utara ke selatan nya. Ilustrasinya dapat dilihat pada Gambar 5 berikut ini:



Gambar 2.3 Menentukan Posisi Pengukuran Hasil Panen pada Survei Ubinan

7. Dari titik yang sudah didapat maka KSK melakukan pengukuran hasil panen pada titik tersebut dengan ukuran bidang yang dipanen yaitu panjang 2,5 meter dan lebar 2,5 meter.
8. Setelah plot selesai dipanen, lalu hasil gabah kering panen (GKP) yang didapat ditimbang dan dihitung jumlah rumpun nya. Tuliskan berat dan banyaknya rumpun pada kuesioner.
9. Lakukan wawancara kepada pengelola lahan untuk beberapa pertanyaan yang ada di kuesioner (luas bidang, teknik penanaman, penggunaan pupuk, bibit, alat perontokan padi dan perbandingan hasil panen dengan tahun sebelumnya).

Dikarenakan banyaknya sample survei ubinan di setiap daerah dan hanya dikerjakan oleh beberapa pelaksana survei ubinan saja, dan proses pendataan nya yang cukup panjang dan rumit, maka dirasa perlu dibuatkan alat bantu ubinan untuk mempermudah pekerjaan pelaksana survei ubinan.

2.3 Android

Android merupakan *platform mobile* yang memberikan pengembang untuk melakukan pengembangan sesuai dengan yang diharapkannya. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, *general Public Lisensi* Versi 2 (GPLv2), yang sering dikenal dengan istilah "*copyleft*" lisensi dimana setiap perbaikan pihak ketiga harus jatuh di bawah terms. Android didistribusikan di bawah Lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya [10].

Android dibuji sebagai "*platform mobile* pertama yang lengkap, terbuka dan bebas" [10].

1. Lengkap (*Complete Platform*)
   Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang pengembangan aplikasi.
2. Terbuka (*Open Source Platform*)
   Platform Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
3. Free (*Free Platform*)
   Android adalah *platform*/aplikasi yang bebas untuk *develop*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada

kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.3.1 Sejarah Android

Android adalah sebuah sistem informasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* 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, dibentuk lah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [10].

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 [10].

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service* (GMS) dan kedua adalah yang benar-benar bebas distribusi nya tanpa dukungan langsung Google atau di kenal sebagai *Open Handset Distribution* (OHD) [10].

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini di produksi oleh HTC corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang Communications, di produksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony 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 [10].

Pada masa saat ini sebagian besar vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi vendor smartphone di dunia yang memproduksi android. Hal ini, karena android itu adalah sistem operasi yang open source sehingga bebas didistribusikan dan dipakai oleh vendor manapun [10].

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena Android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, Aplikasi dan Tool pengembangan, Market aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia [10].

2.3.2 Fitur Android Berdasarkan Versi

Sejak kemunculannya, sistem operasi Android terus mengalami peningkatan baik dari segi kemampuan maupun performanya. Hal ini diawali dengan versi 1.0 selanjutnya meningkat, hingga versi 8.1 dengan beberapa fitur unggulannya. Berikut ini fitur-fitur dilihat dari perkembangan versi nya:

1. Android 1.6 Donut

   Donut membawa informasi dunia dengan kotak telusur kilat. Donut juga menjadi cikal bakal Android yang hadir dalam semua bentuk dan ukuran sementara Android Market kian beranjak dewasa. Kotak telusur kilat Android telah diperkenalkan sejak di sistem Android Donut. Donut dapat

menghasilkan penelusuran dari web dan dari konten lokal ponsel dari kotak telusur di layar utama.

Android dapat hadir dalam berbagai bentuk dan ukuran berkat kemampuan sistem operasi Android Donut, sehingga dapat berfungsi di berbagai resolusi layar dan rasio aspek. Ponsel juga dapat menggunakan resolusi layar lebih besar dari mode potret berukuran 320 x 480 piksel.

Sebelum Google Play muncul, ada Android Market. Diluncurkan pada 2008, Android Market didesain ulang di sistem Android donut untuk menampilkan aplikasi gratis dan aplikasi berbayar teratas ketika katalog aplikasi pihak ketiga Android mulai populer [11].

2. Android 2.1 Eclair

Pada sistem Android Eclair, layar dengan kepadatan piksel tinggi menampilkan wallpaper animasi yang memukau dan merespon sentuhan. Kita juga dapat mengemudi ke mana saja dengan navigasi belokan demi belokan dan informasi lalu lintas *real-time*, semuanya dari ponsel.

Google maps navigasi memberi makna baru pada definisi ponsel cerdas. Petunjuk arah belokan demi belokan yang menggunakan data Google maps menyertakan banyak fitur yang biasanya ada di sistem navigasi mobil seperti tampilan 3D hadap depan, panduan suara, dan informasi lalu lintas dan semuanya dapat digunakan secara gratis.

Di Android, kita dapat menyesuaikan layar utama sesuai selera. Kita juga dapat menyesuaikan nada dering, wallpaper, serta menyusun aplikasi dan widget di banyak layar atau dalam folder. Eclair memperkenalkan wallpaper animasi yang tampak lebih hidup di layar berukuran 854 x 480 piksel yang luar biasa.

Sebelum perintah suara “Ok Google” populer digunakan, kita dapat menge-tap ikon mikrofon untuk mendiktekan langsung ke ponsel. Sistem Android Eclair mengganti tombol koma di keyboards dengan ikon mikrofon. Cukup sekali tap, kata-kata yang diucapkan muncul langsung di layar [11].

3. Android 2.2 Froyo

   Froyo memperkenalkan ponsel secepat kilat yang dapat dikontrol oleh bunyi suara, dan dilengkapi dengan fitur hotspot agar kita selalu terhubung di mana saja.

   Froyo semakin memper canggih kemampuan suara Android dengan *Voice Action*, sehingga kita dapat melakukan fungsi tombol di ponsel seperti menelusuri, mendapatkan petunjuk arah, membuat catatan, menyetel alarm, dan lain-lain hanya dengan suara.

   Dengan tethering, Froyo memungkinkan kita mengubah ponsel menjadi hotspot Wi-Fi portable sehingga kita dapat tetap terhubung meski saat dalam perjalanan.

   Froyo memperkenalkan pengompilasi Dalvik JIT yang memberikan peningkatan performa hingga 5 kali lipat dalam kode yang terikat ke CPU. Froyo juga menghadirkan JavaScript ke browser Android, sehingga meningkatkan performa JavaScript sebesar 2-3 kali lipat [11].

4. Android 2.3 Gingerbread

   Gingerbread membuat pengalaman menggunakan Android lebih sederhana dan lebih cepat bagi pengguna dan pengembang. Bermain *game* menjadi lebih canggih, baterai lebih tahan lama, dan dukungan NFC memunculkan beragam aplikasi baru.

   Gingerbread mendorong *game* Android ke jenjang yang lebih tinggi. Pengembang aplikasi dapat membangun *game* 3D yang kaya grafis berkat tingkat akses baru yang lebih rendah untuk kontrol perangkat, grafis, dan penyimpanan.

   Beberapa tahun sebelum metode pembayaran seluler lazim digunakan di toko, Android meluncurkan dukungan *Near Field Communication* (NFC) agar pengguna dapat mentransmisikan informasi antar perangkat cukup dengan mendekatkannya. Melalui NFC, Gingerbread membuka lebih banyak peluang untuk memberikan layanan dengan sekali menge-tap perangkat.

Gingerbread membantu kita mengoptimalkan masa pakai baterai dengan mengetahui cara perangkat menggunakan tenaga baterai, mencari tahu banyaknya baterai yang dikonsumsi oleh tiap aspek perangkat, mulai dari kecerahan layar hingga aplikasi aktif [11].

5. Android 3.0 Honeycomb

   Honeycomb diluncurkan di era tablet dengan desain antarmuka fleksibel yang memamerkan gambar besar dan navigasi mulus di layar.

   Honeycomb mengoptimalkan ruang di layar tablet. Pola tata letaknya yang lebih besar menyempurnakan pengalaman membaca buku, menonton video, menjelajahi peta, dan lain-lain.

   Tidak ada lagi tombol fisik utama, kembali, dan menu. Dengan Honeycomb, bilah sistem baru memungkinkan kontrol navigasi dipasang di layar perangkat Android.

   Setelan cepat baru memudahkan kita mengakses informasi penting seperti melihat waktu, tanggal, masa pakai baterai, dan status sambungan perangkat, semua di satu tempat [11].

6. Android 4.0 Ice Cream Sandwich

   Ice Cream Sandwich memudahkan penyesuaian dan kontrol pengguna, sesuaikan layar utama, tentukan banyaknya data yang digunakan, dan langsung bagikan konten kapan pun kita mau.

   Ice Cream Sandwich memperkenalkan folder aplikasi dan baki favorit. Widget yang menyematkan konten aplikasi aktif di layar utama menjadi lebih fleksibel seperti luaskan widget untuk menampilkan lebih banyak konten atau ciut kan untuk menghemat tempat.

   Kelola penggunaan data jaringan untuk mengontrol biaya seluler. Lacak jumlah data yang kita gunakan, setel tingkat peringatan dan batas penggunaan, serta non aktifkan layanan data jika sudah mencapai batas.

   Fitur yang futuristik pada masanya, Android Beam memungkinkan dua ponsel berbagi konten melalui NFC cukup dengan menyentuhkan nya secara

bersamaan. Bagikan aplikasi, kontak, musik, dan video kepada orang lain dan semua tanpa perlu membuka menu, aplikasi, atau menyandingkan ponsel [11].

7. Android 4.1 Jelly Bean

   Kecerdasan meresap ke semua aspek Jelly Bean yang diluncurkan di masa bantuan personal bersama Google Now. Lebih mudah merespons notifikasi dan kita bisa memakai beberapa akun pengguna di satu perangkat.

   Dapatkan informasi yang dibutuhkan pada saat yang tepat dengan Google Now. Sebagai pelopor bantuan seluler jenis baru, Google Now memberi tahu cuaca hari ini saat kita sedang bersiap-siap dan waktu perjalanan sebelum kita keluar rumah.

   Pada sistem Android Jelly Bean, notifikasi menjadi lebih luas agar dapat menampilkan lebih banyak informasi. Kita juga dapat langsung merespons dari notifikasi.

   Jelly Bean memungkinkan fitur multi pengguna di satu perangkat. Tiap akun memiliki ruang sendiri yang dapat disesuaikan mulai dari layar utama hingga wallpaper, widget, dan aplikasi agar kita dapat menggunakan perangkat bersama-sama tanpa berbagi info. Fitur multi pengguna diluncurkan dengan tablet dan kemudian diluncurkan ke ponsel dengan Lollipop [11].

8. Android 4.4 KitKat

   Android KitKat membantu kita melakukan banyak hal cukup dengan suara. Ucapkan “Ok Google” untuk meluncurkan penelusuran suara, mengirim SMS, mendapatkan petunjuk arah, atau bahkan memutar lagu. Jika kita mahir menggunakan perangkat, desain baru yang imersif akan membuat kita semakin menikmati konten. Saat membaca buku, bermain *game* atau menonton film, desain imersif KitKat menyembunyikan item yang tidak ingin kita lihat.

   KitKat memprioritaskan kontak yang paling sering kita telepon dan kita dapat menelusuri tempat terdekat langsung dari ponsel. Saat mendapat panggilan

telepon dari nomor tak dikenal, ponsel akan mencari pencocokan dari direktori lokal di Google Maps [11].

9. Android 5.1 Lollipop

   Android telah hadir di layar besar dan kecil. Mulai dari ponsel, tablet, jam tangan, TV dan mobil. Lollipop dilengkapi gaya visual berani dan respons yang mulus dari Desain Material.

   Tampilan dan nuansa gres terbaru dari Android yang memudahkan navigasi perangkat. Didasarkan pada bayangan dan gerakan, Desain Material menggabungkan prinsip dasar desain yang keren dengan segala inovasi teknologi.

   Pada Android Lollipop, kita dapat berpindah dengan mudah dari ponsel ke tablet, jam tangan Android Wear, atau Android TV. Karena Lollipop bekerja di semua perangkat, kita dapat melanjutkan dari saat terakhir mengakses lagu, aplikasi, foto, dan bahkan penelusuran terbaru.

   Notifikasi akan berpindah ke layar kunci dan muncul dalam format kartu yang tersusun rapi. Lihat sekilas, atau bahkan lihat dan tanggapi pesan langsung dari layar kunci. Kontrol terperinci memungkinkan kita menyesuaikan konten yang muncul di layar kunci [11].

10. Android 6.0 Marshmallow

    Pada Android Marshmallow, kita mendapatkan fitur Now on Tap, masa pakai baterai yang lebih tahan lama, dan izin aplikasi baru untuk kontrol yang lebih leluasa. Dengan Now on Tap, kita mendapatkan bantuan tanpa harus keluar dari aplikasi ataupun website. Cukup tap dan tahan lama tombol utama. Pada Marshmallow, kita dapat menentukan fitur apa saja yang kita izinkan untuk sebuah aplikasi. Kita juga bisa mengubah perizinan kapan saja. Pada Android Marshmallow, kita juga dapat menikmati teknologi baterai yang lebih cerdas dan efisien. Marshmallow mengoptimalkan baterai untuk hal-hal paling penting dengan fitur istirahatkan dan aplikasi siaga [11].

11. Android 7.0 Nougat

Android 7.0 Nougat memperkenalkan beragam fitur baru dan kemampuan bagi pengguna dan developer, yaitu:

a. Nougat memiliki emoji-emoji baru dan kemampuan menggunakan dua atau lebih bahasa secara bersamaan.
b. Dapat beralih antar aplikasi dengan tap dua kali dan menjalankan dua aplikasi secara berdampingan. Sehingga, kita dapat terus menonton film sambil menulis pesan, atau membaca resep sementara timer terbuka.
c. Teknologi API Vulkan™ yang dapat menampilkan grafis 3D performa tinggi. Pada perangkat yang didukung, aplikasi tampak seolah hidup dengan grafis lebih tajam dan efek menawan.
d. Mode VR (*Virtual Reality*)
e. Fitur Istirahatkan dapat membantu menghemat daya baterai selagi dalam perjalanan. Sehingga, perangkat akan terus berada dalam mode daya rendah selagi dibawa di dalam saku atau dompet.
f. Tata ulang Setelan Cepat Khusus untuk akses lebih cepat ke aplikasi yang diinginkan.
g. Balas Langsung Notifikasi memungkinkan kita untuk membalas secara langsung tanpa membuka aplikasi apapun.
h. Gabungan Notifikasi
i. Penghemat Data dapat membatasi banyaknya data yang digunakan perangkat dengan Penghemat Data. Saat Penghemat Data diaktifkan, aplikasi di latar belakang tidak akan dapat mengakses data seluler.
j. Kontrol Notifikasi saat muncul notifikasi, cukup tekan dan tahan untuk mengubah setelan. Sehingga kita dapat menonaktifkan notifikasi mendatang langsung dari setelan ini.
k. Mengubah ukuran teks pada perangkat dan mengubah ukuran ikon [11].

12. Android 8.0 Oreo

Beragam pembaharuan fitur-fitur Android diberikan pada Android Oreo, yaitu:

a. Swift bergerak, di belakang layar lebih cepat. Memulai tugas favorit lebih cepat dengan kecepatan booting 2x saat menyalakannya (waktu booting yang diukur di Google Pixel). Android Oreo membantu meminimalkan aktivitas latar belakang pada aplikasi yang paling digunakan.
b. Dengan seizin pengguna, Autofill mengingat login untuk membawa kita ke aplikasi favorit dengan kecepatan supersonik.
c. *Picture in Picture* memungkinkan kita melihat dua aplikasi sekaligus.
d. Tekan titik pemberitahuan untuk segera melihat apa yang baru, dan mudah menghapusnya dengan menggesek pergi.
e. Teleport langsung ke aplikasi baru langsung dari browser Anda, tidak perlu instalasi.
f. Perlindungan Google Play bekerja agar perangkat dan data tetap aman dari aplikasi yang tidak sesuai dengan memindai lebih dari 50 miliar aplikasi per hari, bahkan yang belum terpasang.
g. Baterai lebih hemat.
h. Tambahan emoji baru [11].

Smartphone Android dirasakan sangat cocok untuk menjalankan sistem alat bantu ubinan karena memiliki fitur dan fasilitas yang banyak. Selain itu juga, pelaksana survei ubinan sebagian besar telah menggunakan smartphone Android sehingga pelaksana survei ubinan tidak akan kesulitan pada saat beradaptasi dengan sistem alat bantu ubinan yang akan dibangun.

2.3.2.1 A-GPS

Kemampuan sebuah *smartphone* dalam mengetahui posisi di muka bumi menjadi semakin penting. Bahkan saat ini sudah merambah ke penggunaannya yang sangat umum. Sedemikian umumnya, hingga semua *smartphone* sudah wajib memiliki kemampuan untuk mengetahui posisi penggunanya. Teknologi yang digunakan untuk mengetahui posisi suatu perangkat yaitu GPS (*Global Positioning System)*

GPS singkatan dari *Global Positioning System* (Sistem Pencari Posisi Global), adalah suatu jaringan satelit yang secara terus menerus memancarkan sinyal radio dengan frekuensi yang sangat rendah. Alat penerima GPS secara pasif menerima sinyal ini, dengan syarat bahwa pandangan ke langit tidak boleh terhalang, sehingga biasanya alat ini hanya bekerja di ruang terbuka. Satelit GPS bekerja pada referensi waktu yang sangat teliti dan memancarkan data yang menunjukkan lokasi dan waktu pada saat itu. Operasi dari seluruh satelit GPS yang ada disinkronisasi sehingga memancarkan sinyal yang sama. Alat penerima GPS akan bekerja jika ia menerima sinyal dari sedikitnya 4 buah satelit GPS, sehingga posisinya dalam tiga dimensi bisa dihitung. Pada saat ini sedikitnya ada 24 satelit GPS yang beroperasi setiap waktu dan dilengkapi dengan beberapa cadangan. Satelit tersebut dioperasikan oleh Departemen Pertahanan Amerika Serikat, mengorbit selama 12 jam (dua orbit per jari) pada ketinggian sekitar 11.500 mil dan bergerak dengan kecepatan 2000 mil per jam. Ada stasiun penerima di bumi yang menghitung lintasan orbit setiap satelit dengan teliti [12].

Rusia (GLONASS) dan China (BeiDou) juga memiliki solusi penentu lokasi yang serupa. GPS, GLONASS (Globalnaya Navigazionnaya Sputnikovaya Sistema/*Global Navigation Satellite System*), BeiDou, dan sebagainya itu adalah satelit-satelitnya. Secara umum, semua satelit navigasi itu dinamakan GNSS (*Global Navigation Satellite System*).

Smartphone digunakan sebagai penerima sinyal dari satelit-satelit GNSS (*receiver*). Secara umum, banyak yang disebut sebagai GPS *receiver*, karena GPS adalah rangkaian satelit navigasi yang paling umum dikenal dan yang pertama kali dipopulerkan. Sehingga yang berada di dalam *smartphone* adalah GNSS receiver atau penangkap sinyal GSS atau lebih mudahnya disebut sebagai sistem navigasi.

Dalam menggunakan sistem navigasi, kita tidak perlu menggunakan paket data atau internet. Untuk mengetahui lokasi dengan menggunakan satelit (GPS/GLONASS/BeiDou), *smartphone* tidak membutuhkan paket data atau

internet karena yang dilakukan oleh sistem navigasi di dalam *smartphone* kita adalah "mendengar" sinyal dari satelit, bukan melakukan transaksi data.

Cara kerja sistem navigasi global secara sederhana yaitu satelit (GPS/GLONASS/BeiDou) memancarkan sinyal. Kita bisa umpamakan mereka seperti stasiun pemancar radio FM dan *smartphone* sebagai radio penangkap siarannya. Setiap satelit akan memberikan data posisi dan waktu satelit tersebut mencapai posisi yang dikirimkan. Dengan "mendengarkan" laporan dari satelit-satelit tersebut, sistem navigasi di dalam *smartphone* dapat memperhitungkan di mana posisinya di muka bumi ini. Tentunya, dibutuhkan minimal sekitar 4 satelit untuk dapat mengetahui posisi dengan baik. Makin banyak satelit yang didengarkan, makin akurat pula hasil perhitungannya. Sebagai acuan, GPS dapat mencapai akurasi lokasi hingga radius 2 meter saja di permukaan bumi.

Assisted GPS atau singkatnya A-GPS adalah sebuah sistem yang mempercepat TTFF (*Time To First Fix*) atau kecepatan menentukan posisi pertama kali. Rata-rata *smartphone* modern sudah menggunakan A-GPS untuk membantu sistem navigasi global di dalam smartphone "mengunci" satelit dengan cepat.

Satelit-satelit GPS/GLONASS/BeiDou akan selalu bergerak dan tidak berada dalam posisi yang sama. Akan tetapi, pergerakannya telah terjadwal rapi. Umpamakan saat ini ada satelit A, B, C, dan D di atas Jakarta. Di waktu yang berbeda, kondisi ini berubah dan satelit C, D, E, dan F yang berada di atas Jakarta.

Sistem navigasi di dalam *smartphone* harus mengetahui satelit mana yang relevan untuk perhitungan posisinya. Untuk mengetahui mana yang relevan, harus diketahui mana yang posisinya terdekat untuk mempermudah perhitungan posisi.

Oleh sebab itu, saat pertama kali dihidupkan, sistem navigasi pada *smartphone* akan berusaha mendengar ke semua satelit dan menunggu laporan lokasinya, agar tahu mana yang terdekat. Berusaha mendengarkan ke semua satelit dan memilihnya akan membutuhkan waktu yang lama sehingga, secara detail TTFF itu adalah waktu antara pertama kali sistem navigasi mencoba

mencari tahu posisi satelit-satelit tersebut, menentukan mana yang relevan, dan selesai memperhitungkan posisinya sendiri di muka bumi ini.

Dengan A-GPS yang memanfaatkan jaringan selular, sistem navigasi dalam *smartphone* akan mendapatkan informasi, satelit apa saja yang sedang berada di dekatnya dan relevan untuk diperhitungkan. Dengan demikian, TTFF akan menjadi lebih cepat. A-GPS bisa membantu dengan cepat karena stasiun-stasiun pemancar sinyal seluler di darat ini (atau kadang disebut tiang BTS), tentunya sudah memiliki kepastian lokasinya masing-masing. Sehingga, dengan mengirimkan laporan lokasinya sendiri dan data satelit serta keberadaannya secara akurat ke smartphone akan membuat sistem navigasi di smartphone dapat melakukan kalkulasi posisi dengan cepat.

Tentunya hal ini membutuhkan transfer data untuk memperoleh peta lokasi satelit dari operator. Tentunya, paket data yang aktif menjadi esensial di sini. sehingga, penentu lokasi tidak harus menggunakan paket data. Tetapi, jika ingin "mengunci" keberadaan posisi kita secara akurat dengan cepat, tetap dibutuhkan paket data, meski memang data yang ditransaksikan sangat kecil.

A-GPS dibutuhkan oleh sistem alat bantu ubinan untuk mencatatkan lokasi pelaksanaan survei ubinan agar lebih mudah ditelusuri kembali nantinya. Sistem alat bantu ubinan akan menyimpan titik latitude longitude yang dihasilkan oleh A-GPS pada database.

2.3.2.2 Kompas

Kompas merupakan sebuah perangkat penunjuk arah mata angin yang di dalamnya tertanam perangkat *magnetic* sebagai penala antara kutub utara dan kutub selatan. Pada perangkat android terdapat fitur kompas. Selain digunakan sebagai penunjuk arah mata angin, kompas juga digunakan sebagai alat navigasi lengkap layaknya nahkoda atau pilot pesawat dengan beberapa fungsi GPS-nya. Namun berbeda dengan kompas pada umumnya, kompas pada smartphone

android tidak menggunakan komponen *magnetic*. *Smartphone* hanya mengandalkan kode program dari aplikasi kompas tersebut.

Aplikasi di *smartphone* adalah hasil *compile* dari deretan kode program, tidak ada komponen fisik yang terpasang. Aplikasi kompas dapat bekerja layaknya kompas sungguhan karena sewaktu pertama kali melakukan instalasi kode-kode program yang ada di dalam aplikasi kompas meminta izin akses ke geo lokasi *smartphone*. Berikut beberapa cara, agar kompas dapat mendapatkan lokasi pengguna *smartphone*:

1. Melalui akun Google yang tertanam dalam *smartphone* terutama yang sudah terintegrasi oleh GPS.
2. Melalui GPS, alat penentu posisi yang dapat memberikan arah mata angin, juga dapat memberitahukan posisi koordinat pengguna *smartphone*.
3. Melalui operator selular, dengan adanya sambungan ke salah satu operator selular maka dapat diketahui letak koordinat *smartphone* berada walaupun tidak se presisi pada saat menggunakan GPS.
4. Sensor yang digunakan untuk mendapatkan posisi arah mata angin pada smartphone Android yaitu sensor accelerometer. Agar kompas dapat bergerak pada saat smartphone digerakkan.

Kompas akan ditampilkan pada sistem alat bantu ubinan untuk membantu pelaksana survei ubinan dalam menggambarkan bentuk lokasi pelaksanaan survei ubinan. Karena dalam penggambaran bentuk lokasi, perlu disesuaikan dengan arah mata angin yaitu utara menghadap ke atas.

2.4 Android SDK

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

Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform aplikasi-netral, Android memberikan kesempatan untuk membuat Aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Handphone/*Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah [10]:

1. Framework Aplikasi yang mendukung pengganti komponen dan reusable.
2. Mesin Virtual Dalvik di optimalkan untuk mobile
3. *Integrated browser* berdasarkan engine open source webKit
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional akselerasi hardware)
5. SQLite untuk menyimpan data
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung *hardware*).
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung *hardware*)
8. Kamera, GPS, kompas, dan *accelerometer* (tergantung *hardware*)
9. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

Setiap kali Google merilis versi baru dari Android, sebuah SDK yang sesuai juga dirilis. Untuk dapat menulis program dengan fitur terbaru, pengembang harus men-download dan menginstal SDK versi masing-masing untuk Ponsel tertentu. Agar sistem alat bantu dapat dijalankan pada semua versi, maka Android SDK dirasa perlu digunakan.

2.5 Android Studio

Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu

IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android.

Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut :

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

Jika dibandingkan dengan Android Studio memang dari sisi build lebih baik dibandingkan Eclipse, karena Android Studio menggunakan Gradle. Pada Android Studio, kita tidak perlu lagi dipusingkan dengan *dependencies package* berbeda dengan Eclipse. Satu hal tambahan lagi yang membuat Android Studio unggul adalah dukungan layout xml editor secara visual yang jauh lebih baik daripada Eclipse. Walaupun begitu, Android Studio saat ini masih dalam tahap beta dan belum mempunyai dukungan untuk NDK/Native Development Kit.

2.6 Java

Java adalah sebuah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems. Bahasa pemrograman Java adalah bahasa yang digunakan untuk menghasilkan aplikasi-aplikasi Java. Pada umumnya, bahasa pemrograman hanya mendefinisikan sintaks dan perilaku bahasa [13].

Pada saat program Java dikompilasi, ia akan dikonversi ke bentuk bytecode, yang merupakan bahasa mesin yang portable. Selanjutnya bytecode

tersebut dijalankan di *Java Virtual Machine* (JVM). Meskipun JVM dapat diimplementasikan langsung di perangkat keras, namun biasanya diimplementasikan dalam bentuk program perangkat lunak yang mengemulsi mesin (komputer) dan digunakan untuk menginterpretasi bytecode [13].

Guna mencakup lingkungan-lingkungan aplikasi yang berbeda, Sun mendefinisikan 3 (tiga) edisi Java, yaitu [13]:

1. J2ME (*Java 2 Micro Edition*)
   Edisi ini ditujukan bagi lingkungan dengan sumber daya terbatas, seperti *smartcard*, ponsel dan PDA.
2. J2SE (*Java 2 Standard Edition*)
3. Edisi ini ditujukan bagi lingkungan *workstation*, seperti pembuatan aplikasi-aplikasi desktop.
4. J2EE (*Java 2 Enterprise Edition*)
   Edisi ini ditujukan bagi lingkungan internet atau aplikasi terdistribusi dalam skala besar.

Pada sistem alat bantu ubinan, smartphone android dijadikan media dalam menjalankan sistem sehingga bahasa pemrograman yang sesuai yaitu bahasa pemrograman java.

2.7 Pemrograman Berorientasi Objek

Di dalam membangun sebuah sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah programmer di dalam mendesain program dalam bentuk objek-objek dan hubungan antara objek tersebut untuk kemudian dimodelkan dalam sistem nyata [14].

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi

berorientasi objek merupakan suatu cara bagaimana system perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktivitas
   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reusable).
2. Kecepatan pengembangan
   Karena system yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean
3. Kemudahan pemeliharaan.
   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Adanya konsistensi
   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Meningkatkan kualitas perangkat lunak
   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan system (system perangkat lunak, system informasi, atau system lainnya). Pendekatan berorientasi objek akan

memandang system yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antarkelas sampai abstraksi system. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

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

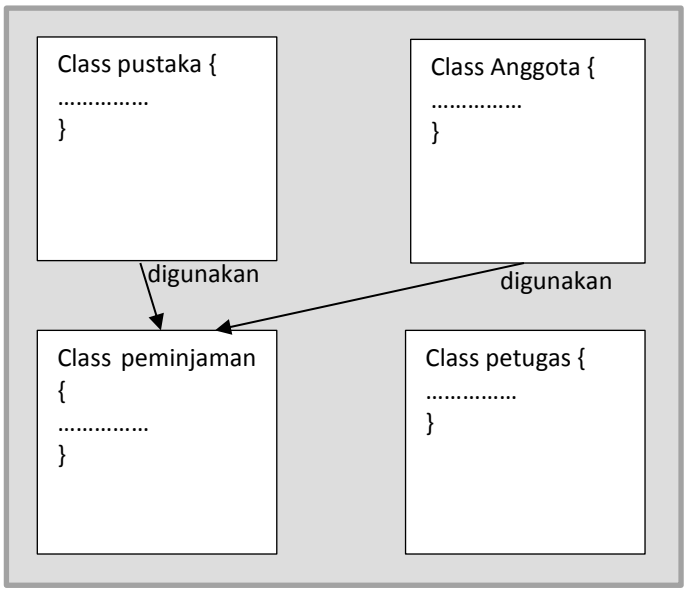
Sistem berorientasi objek merupakan sebuah system yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah system yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam system tersebut dapat mewarisi atribut dan sifat dan komponen lainnya. Dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (class)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi static dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut). Kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek. Ilustrasi dari sebuah kelas dapat dilihat pada gambar berikut.



Sumber gambar: Buku Rekayasa Perangkat Lunak oleh Rosa A.S dan M. Shalahuddin (2013) [8]

Gambar 2.4 Ilustrasi Kelas

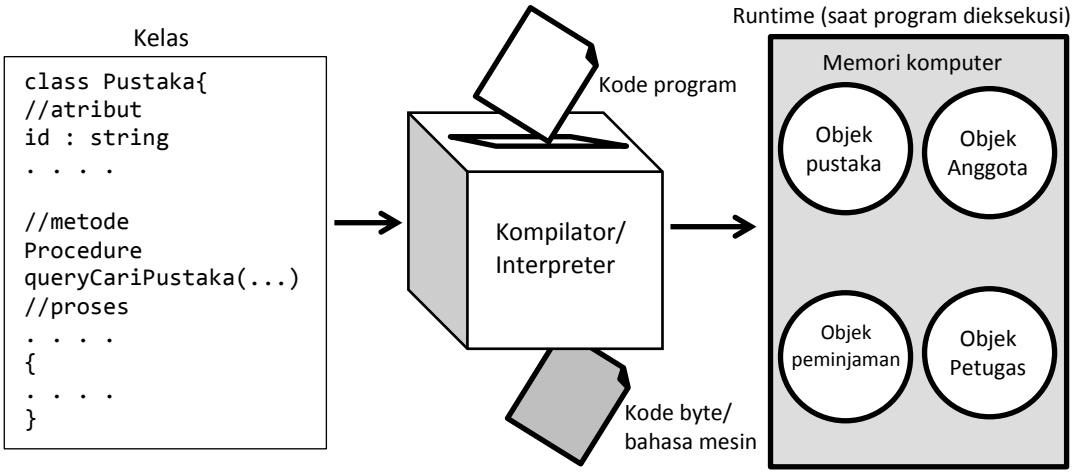
Kelas secara fisik adalah berkas atau file yang berisi kode program, dimana kode program merupakan semua hal yang terkait dengan nama kelas.

2. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal - hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi maka akan di buat sebuah objek. Objek dilihat dari segi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehingga sebuah objek hanya ada saat sebuah objek hanya ada saat sebuah program dieksekusi, jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat

sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.



Sumber gambar: Buku Rekayasa Perangkat Lunak oleh Rosa A.S dan M. Shalahuddin (2013) [8]

Gambar 2.5 Ilustrasi Kelas dan Objek

3. Metode (method)

Operasi atau metode atau method pada sebuah kelas hamper sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.

Metode atau operasi dapat berasal dari event, aktivitas atau aksi keadaan, fungsi, atau kelakuan dunia nyata. Atau kelakuan dunia nyata. Contoh metode atau operasi misalnya read, write, move copy, dan sebagainya. Kelas sebaiknya memiliki metode get dan set untuk setiap atribut agar konsep enkapsulasi tetap terjaga. Metode get digunakan untuk memberikan akses kelas ini dalam mengakses atribut, dan set adalah metode yang digunakan untuk mengisi atribut, agar kelas ini tidak mengakses atribut secara langsung.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaiknya bersifat privet untuk menjaga konsep enkapsulasi.

5. Abstraksi (*abstraction*)

Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah *interface* dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat

mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu. Metode pada antarmuka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka. Antarmuka atau *interface* biasanya digunakan agar kelas yang lain tidak mengakses langsung ke suatu kelas, mengakses antarmukanya.

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permisalan pada permasalahan lainnya yang melibatkan objek tersebut. Misalkan dalam sebuah aplikasi peminjaman buku diperlukan kelas anggota, maka ketika membuat aplikasi penyewaan VCD, kelas anggota ini bisa digunakan kembali dengan sedikit perubahan untuk aplikasi penyewaan VCD tanpa harus membuat dari awal kembali.

10. Generalisasi dan Spesialisasi

Menunjukkan hubungan antara kelas dan objek uang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antar Objek

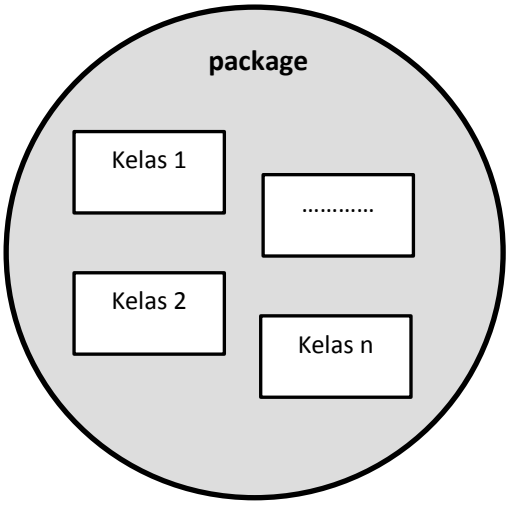
Komunikasi antar-objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

Kemauan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

13. Package

Package adalah sebuah container atau kemasan yang dapat di gunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda. Ilustrasi dari sebuah package dapat dilihat pada gambar berikut [8].



Sumber gambar: Buku Rekayasa Perangkat Lunak oleh Rosa A.S dan M. Shalahuddin (2013) [8]

Gambar 2.6 Package

Pemrograman berorientasi objek digunakan agar lebih mudah dalam perawatan perangkat lunak, pemrograman berorientasi objek akan mudah dalam merawat dan memodifikasi kode yang sudah ada. Objek yang baru dapat dibuat tanpa mengubah kode yang sudah ada. Dalam membangun aplikasi *frontend* sistem alat bantu ubinan menggunakan pemrograman berorientasi objek.

2.8 UML (*Unified Modelling Language*)

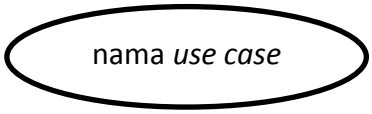
Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [14]. Untuk membuat sebuah program berorientasi objek, UML cocok digunakan untuk merancang model-model yang akan dibangun nantinya.

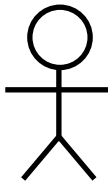
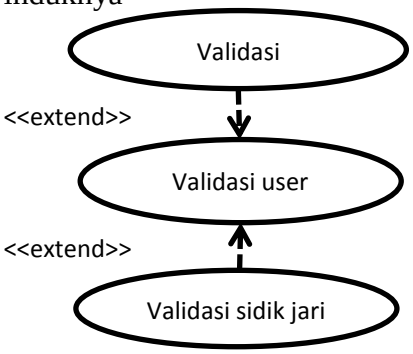
UML memiliki beberapa diagram antara lain : *use case diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram, deployment diagram* [14]. Berikut ini penjelasan untuk beberapa diagram yang akan digunakan dalam penelitian ini:

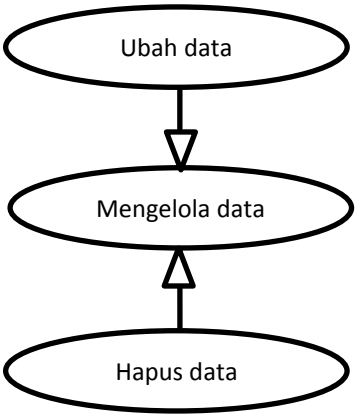
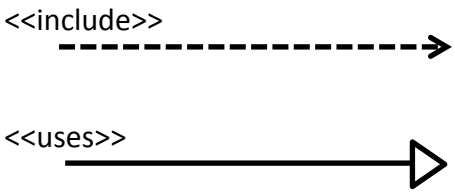
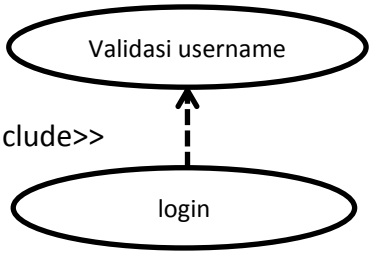
2.8.1 Use Case Diagram

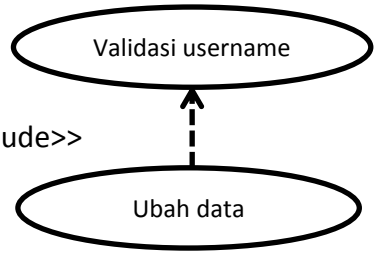
Use case diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah *use case* merepresentasikan interaksi antara aktor dengan sistem. *Use case* sangat menentukan karakteristik sistem yang sedang dibuat. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [14]. Berikut adalah simbol-simbol yang ada pada *use case diagram:*

Tabel 2.1 Simbol-simbol yang ada pada diagram use case [8]

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case

Simbol	Deskripsi
Aktor/*actor*  Nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, sehingga walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / *association* 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
Ekstensi / *extend*  	Relasi *use case* tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi obyek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan. Misalnya arah panah mengarah pada *use case* yang ditambahkan, biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya 

Simbol	Deskripsi
Generalisasi / *generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya:
	 arah panah mengarah pada use case yang menjadi generalisasi nya (umum).
Menggunakan / *include* / *uses* 	Relasi use case tambahan ke sebuah use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: *Include* berarti *use case* yang ditambahkan akan selalu dipanggil saat *use case* tambahan dijalankan, misal pada kasus berikut: 

Simbol	Deskripsi
	Include berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:  Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

2.8.2 Class Diagram

Class merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan objek yang berhubungan satu sama lain seperti *containment*, asosiasi, dan lain-lain [14]. Berikut adalah simbol-simbol yang ada pada *class diagram.*

Tabel 2.2 Simbol-simbol yang ada pada class diagram [8]

Simbol	Deskripsi
Kelas **nama_kelas** +atribut +operasi()	Kelas pada struktur sistem

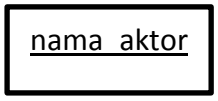
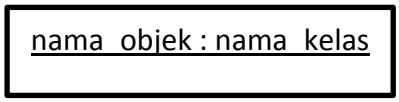
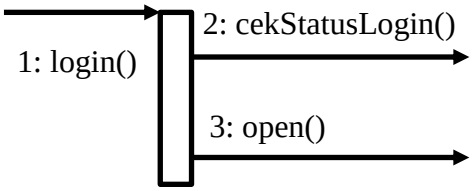
Simbol	Deskripsi
Antarmuka / *interface*  **nama_interface**	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi / *association* 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *directed association* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*.
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *dependency* 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi / *aggregation* 	Relasi antarkelas dengan maksa semua bagian (*whole part*)

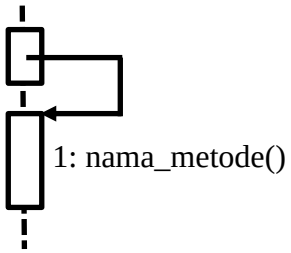
2.8.3 Sequence Diagram

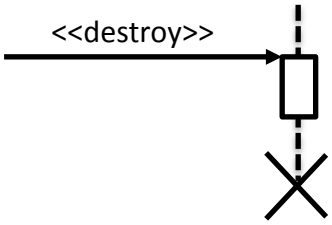
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [14].

Berikut adalah simbol-simbol yang ada pada *sequence diagram*:

Tabel 2.3 Simbol-simbol yang ada pada sequence diagram [8]

Simbol	Deskripsi
Aktor  Nama aktor Atau  Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, sehingga walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup / *lifeline* 	Menyatakan kehidupan suatu objek, untuk menggambarkan kelas dan objek.
Objek  	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya
Waktu aktif 	 Maka cekStatusLogin() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif

Simbol	Deskripsi
Pesan tipe create <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1: nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe send 1: masukan()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, sesuai dengan kelas objek yang berinteraksi.
Pesan tipe return 1: keluaran()	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

Simbol	Deskripsi
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

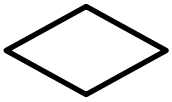
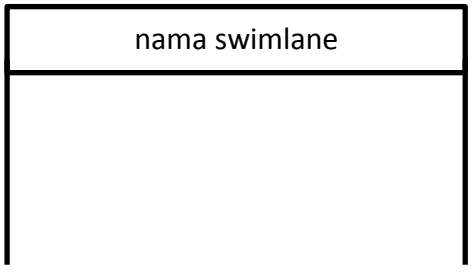
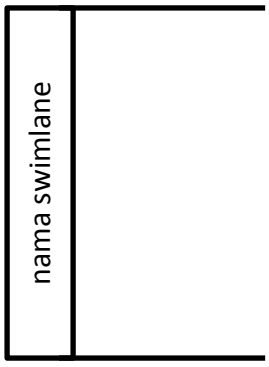
2.8.4 Activity Diagram

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi [14].

Activity Diagram merupakan state diagram khusus, dimana sebagian besar state adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya state sebelum (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas [14]. Berikut adalah simbol-simbol yang ada pada *activity diagram*:

Tabel 2.4 Simbol-simbol yang ada pada *activity diagram* [8]

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal

Simbol	Deskripsi
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

2.9 Pemrograman Terstruktur

Analisis terstruktur adalah salah satu pendekatan formal pertama untuk analisis sistem informasi. Analisis ini terfokus pada aliran data dari proses bisnis dan perangkat lunak. Analisis ini disebut *process oriented*. Analisis terstruktur sederhana dalam konsep. Pada analisis menggambarkan serangkaian proses dalam bentuk diagram alir data (*data flow diagram*) yang menggambarkan proses yang ada atau yang diusulkan bersama-sama dengan input, dan file mereka [15].

Dalam membangun *backend* pada sistem alat bantu ubinan, diperlukan sebuah aplikasi yang dapat mengelola seluruh data yang ada pada database yang akan dibangun menjadi sebuah aplikasi website dan perancangan aplikasinya menggunakan pemrograman terstruktur. Perancangannya menggunakan DFD (*Data Flow Diagram*).

2.10 DFD (*Data Flow Diagram*)

Rancangan model logika (*logical model*) dari sistem informasi lebih menjelaskan kepada user bagaimana nantinya fungsi-fungsi di sistem informasi secara logika akan bekerja. Model logika dapat digambar dengan menggunakan diagram arus data (*data flow diagram*). DFD menggambarkan arus data dari suatu sistem informasi, baik sistem lama maupun sistem baru secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut berada [16].

Tabel 2.5 Simbol-simbol DFD

De Marco and Yourdan Symbols	*Gane and Sarson Symbols*	Nama	Keterangan
		Source (Kesatuan Luar)	Kesatuan luar adalah kesatuan di luar sistem yang akan memberikan input atau menerima output dari sistem, dapat berupa orang, organisasi, sumber informasi lain ataupun penerima akhir dari suatu laporan

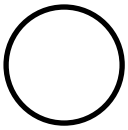
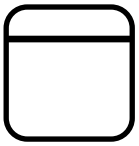
De Marco and Yourdan Symbols	*Gane and Sarson Symbols*	Nama	Keterangan
		Proses	Proses merupakan kegiatan atau pekerjaan yang dilakukan oleh orang atau mesin komputer, dimana aliran data masuk ditransformasikan ke aliran data keluar
		Data Flow (Arus Data)	Arus data menunjukkan arus data, dapat berupa masukan untuk sistem atau hasil proses sistem.
		Data Store (Simpanan Data)	Notasi ini dapat mendefinisikan file atau basis data seperti tape magnetic, disk, dan model DBMS lain, atau mendefinisikan bagaimana penyimpanan diimplementasikan dalam sistem komputer.

Diagram konteks (*context diagram*) adalah diagram tingkat atas, merupakan diagram dari sebuah sistem yang menggambarkan aliran data yang masuk dan keluar dari sistem yang menggambarkan aliran data yang masuk dan keluar dari sistem dan yang masuk dan keluar dari entitas luar.