BAB II
LANDASAN TEORI

Pada bab ini akan dijelaskan konsep dasar dan teori-teori yang mendukung pengembangan aplikasi . Selain itu, pada bab ini juga dijelaskan tools-tools yang dipakai dalam pembangunan aplikasi pencarian rumah kontrakan dan kosan.

1.1 Rumah Sewa

Rumah merupakan sebuah bangunan, tempat manusia tinggal dan melangsungkan kehidupannya, disamping itu rumah juga merupakan tempat berlangsungnya proses sosialisasi. Rumah adalah bangunan yang berfungsi sebagai tempat tinggal atau hunian (UU No4 Tahun 1992 Tentang Perumahan dan Pemukiman).

Menurut kamus besar bahasa indonesia sewa berarti pemakaian sesuatu dengan membayar uang sewa dan menyewa berarti memakai dengan membayar uang sewa, sewa-menyewa adalah persetujuan antara pihak yang menyewakan dengan pihak penyewa, pihak yang menyewakan menyerahkan barang yang hendak disewa kepada pihak penyewa untuk dinikmati sepenuhnya.

kesimpulan dari pengertian rumah sewa adalah bangunan yang ber'ungsi sebagai rumah tinggal yang dipakai atau di man'atakan denganmembayar uang sewa dimana didalamnya ada persetujuan antara pihak yang menyewakan dengan pihak penyewa.

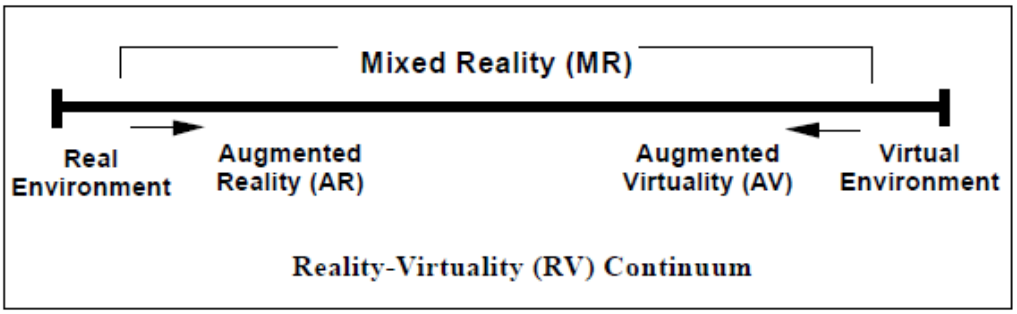
Akan tetapi seiring dengan berjalannya waktu pemanfaatan atau kebutuhan akan rumah sewa ini semakin marak terjadi di kota-kota besar, hal inilah kemudian yang mulai menjadi kebutuhan terutama bagi mereka mahasiswa yang berada di perantauan, bagi keluarga yangmeman'aatkan untuk hanya sekedar tempat berteduh, dan atau bahkan hanya sebagai tempat singgah untuk sekedar menghilangkan akan penat nya rutinitas sehari-hari si penyewa rumah. [3]

1.2 Augmented Reality

Augmented reality merupakan teknologi yang dapat memvisualisasikan elemen-elemen digital baik 2D maupun 3D ke dalam lingkungan nyata melalui sebuah *webcam*, kamera *smartphone* dan kacamata [1].

1.2.1 Aristektur Augmented Reality

Pada tahun 1994, Milgram dan Kishino merumuskan kerangka kemungkinan penggabungan dan peleburan dunia nyata dan dunia maya ke dalam sebuah ***v**irtuality continuum.* Dalam kerangka tersebut, *augmented reality* lebih dekat ke sisi kiri yang menjelaskan bahwa lingkungan bersifat nyata dan benda bersifat maya. Sebaliknya *augmented virtuality* lebih dekat ke sisi kanan dalam kerangka tersebut, yang menjelaskan bahwa lingkungan bersifat maya dan benda bersifat nyata. Sehingga jika terjadi penggabungan antara a*ugmented reality* dengan *virtual reality* akan tercipta *mixed reality* [4]



Gambar 2. 1 Kerangka *Augmented Reality* [4]

1.2.2 Konsep Augmented Reality

Augmented reality memiliki 2 metode yang sangat signifikan berkembang dalam beberapa tahun belakangan ini yaitu *marker based tracking* dan *markerless augmented reality.*

1.2.3 Marker Based Tracking

Augmented reality pada *marker based tracking* merupakan metode yang dikembangkan sejak tahun 80an. *Marker* merupakan ilustrasi dari hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi *marker* dan menciptakan dunia *virtual* 3D yaitu titik (0,0,0) dan 3 sumbu yaitu X,Y,dan Z [5].



Gambar 2. 2 Macam bentuk marker *augmented reality* [20]

1. Barcode standards

Terdapat beberapa jenis marker yang pada umumnya digunakan pada implementasi Marker-based AR, diantaranya adalah

2D-barcode yang umumnya berbentuk persegi atau persegi panjang. 2D-barcode terdiri dari warna hitam dan putih. Warna putih pada 2D-barcode menyimpan data bit berupa 1, sedangkan warna hitam menyimpan data berupa bit 0.

Pada 2D-barcode, biasanya terdapat penanda berupa object dengan garis hitam dan putih yang lebih tebal dan menggumpal agar 2D-barcode dapat lebih mudah dibaca. 2D-barcode bersifat lebih fleksibel dalam penerapan ukuran barcode. QR Code, DataMatrix, dan PDF417 merupakan contoh 2D-barcode. [6]

2. Circular markers

Penanda yang berbentuk lingkaran pada sisi luarnya, semakin banyak lingkaran di dalam suatu circular marker maka akan semakin tepat posisi objek AR di gambarkan. Circular marker pada umumnya digunakan pada implementasi AR yang membutuhkan akurasi yang tepat, tetapi memungkinkan pemrosesan informasi secara offline. Seperti 2D-Barcode, circular markers hanya terdiri dari warna hitam dan putih.

3. Image markers

Penanda yang berupa berupa gambar yang menggunakan warna natural sebagai marker. Image marker biasanya dikelilingi oleh suatu frame tertentu untuk membantu deteksi dan rotasi marker tersebut. Image

markers pada umumnya digunakan jika marker tidak ingin dihilangkan pada saat suatu objek AR ditanamkan kedalam foto yang telah diproses.

1.2.4 Markerless Augmented Reality

Markerless Augmented Reality merupakan salah satu metode *Augmented Reality* tanpa menggunakan frame marker sebagai obyek yang dideteksi. Dengan adanya *Markerless Augmented Reality*, maka, penggunaan marker sebagai tracking object yang selama ini menghabiskan ruang, akan digantikan dengan gambar, atau permukaan apapun yang berisi dengan tulisan, logo, atau gambar sebagai tracking oject (obyek yang dilacak) agar dapat langsung melibatkan obyek yang dilacak tersebut sehingga dapat terlihat hidup dan interaktif, juga tidak lagi mengurangi efisiensi ruang dengan adanya marker.

Terdapat perbedaan antara pelacakan berbasis marker (marker based tracking) dan pelacakan markerless (markerless tracking). Pada pelacakan berbasis marker posisi kamera dan orientasi kamera dhitung dengan marker yang telah ditetapkan. Sementara pelacakan markerless, menghitung posisi antara kamera/pengguna dan dunia nyata tanpa referensi apapun, hanya menggunakan titik-titik fitur alami (edge, corner. garis atau model 3D). Metode Markerless memerlukan langkah priorti manual, serta modelatau gambar referensi untuk inisialisasi. [7]



Gambar 2. 3 Markerless *Augmented Reality* [4]

1. Face Tracking

Ciri pada wajah setiap manusia berbeda-beda, namun pasti setiap manusia memilki mata, hidung dan mulut. Penentuan titik koordinat bentuk pola wajah menggunakan algoritma Viola-Jones. Dengan algoritma Viola-Jones, teknik ini dapat mengenali pola pada mata, hidung dan mulut dan mengabaikan objek sekitarnya sehingga objek digital dapat diimpelementasikan pada wajah melalui sebuah kamera.

2. 3D Object Tracking

Berbeda dengan *Face Tracking* yang hanya mengenali pola wajah secara umum, teknik *3D object Tracking* dapat mengenali semua bentuk benda yang ada disekitarnya seperti televisi, handphone, mobil, rumah dan lain-lain.

3D object Tracking menggunakan hubungan geometris sebagai pendeteksi bentuk objek 3D. Hubungan geometris digunakan karena pada *3D object Tracking* hanya menggunakan fungsi kamera untuk mendeteksi benda-benda nyata yang ada di sekitar sebagai sebuah *marker*. Ada 2 Metode pengembangan *3D object Tracking* yang dipakai yaitu *bottom-up approach*(BUA) dan *top-down approac*h(TDA). 2 metode ini cocok untuk dipakai dalam penentuan titik koordinat bentuk objek 3D berbasis kamera.

3. Motion Tracking

Metode *marker*less dengan teknik ini yaitu dengan menangkap gerakan yang dilakukan pengguna. Salah satu contoh Augmented Reality dengan menggunakan teknik ini yaitu pada film Avatar. Dunia film sudah menggunakan teknologi ini dengan sangat ekstensif karena dapat mensimulasikan gerakan secara real-time.

Motion tracking lebih dekat ke dunia *virtual reality*, karena di sini hasil yang tampak hampir keseluruhan dunia virtual. Dalam penerapannya, *motion tracking* memerlukan sebuah alat pendeteksi gerakan dari tubuh penggunanya. Salah satu alat tersebut adalah *Head Mounted Display* (HMDs), alat ini berfungsi sebagai indra penglihatan dunia *virtual*. Metode pengembangan yang dipakai dalam *motion tracking* yaitu metode TDA. Metode ini dibutuhkan karena *inertial orientation sensors* yang menjadi solusi dalam menangkap sebuah gerakan.

4. GPS Based Tracking

Pengembangan teknik ini lebih diarahkan pada *smartphone,* karena teknologi GPS dan kompas yang tertanam pada *smartphone* tersebut. Dengan memanfaatkan fitur GPS yang berfungsi sebagai penentu lokasi pengguna pada saat itu berada sehingga lokasi terdekat yang ingin dituju dapat dilihat melalui implementasi augmented reality.

Teknik ini berguna sebagai pemandu selayaknya fungsi GPS, namun dilengkapi dengan *marker* informasi arah yang dituju. Dalam implementasinya, teknik ini mengharuskan tersambungnya koneksi GPS dan kebutuhan paket data yang ada pada *smartphone*, karena data-data lokasi yang dimiliki GPS memiliki akses

langsung dari satelit agar cepat mendeteksi wilayah yang telah dijadikan sebuah objek *marker* informasi pada *Augmented Reality*. Akses internet memiliki fungsi sebagai pemanggilan data-data berupa *latitude, longitude,* serta informasi yang mendukung setiap lokasi yang disimpan pada *server* sehingga beban ukuran aplikasi dapat diminimalisir [8]

Teknik GPS *based tracking* sebenarnya membutuhkan peran kompas dan akselerometer sebagai pengatur ukuran layar secara *horizontal* dan *vertical* agar marker lokasi dapat dilihat ketika kamera *handset* berada posisi yang sesuai dengan lokasi tersebut. Namun ketika *handset* tidak berada dalam sudut pandang lokasi tersebut maka *marker* tersebut tidak akan tampak. Kebanyakan teknik GPS *based tracking* sudah memiliki *engine* pembantu (AR *browser*) yang telah dikembangkan oleh beberapa perusahaan sehingga mempermudah untuk mengembangkan teknik ini sesuai dengan keinginan. Pada tugas akhir ini digunakan AR Mixare *Browser* sebagai engine untuk mengembangkan aplikasi menjadi lebih mudah digunakan.

a. Penentuan Lokasi Jarak Terdekat

Dalam menentukan jarak lokasi tujuan terdekat dibutuhkan persamaan yang mampu menkalkulasikan antara 2 lokasi yaitu lokasi pengguna berada dan lokasi yang ada disekitar pengguna. Lokasi terdekat biasanya menjadi tujuan utama pengguna untuk mencapai lokasi tersebut, sehingga untuk mendapatkan jarak terdekat lokasi tersebut dengan pengguna digunakan metode *Great Circle Distance*. Great Circle Distance adalah metode untuk melakukan menentukan jarak terpendek antara 2 lokasi di muka bumi. Salah satu persamaan untuk menghitung greatest distance antara 2 titik lokasi yaitu dengan menggunakan persamaan haversine. Persamaan Haversine merupakan persamaan yang dapat mengkalkulasikan jarak antara 2 titik lokasi di muka bumi. Perhitungan dari rumus ini juga cukup akurat dimana rumus ini mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi.

b. Persamaan Haversine

Metode Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (longitude) dan garis bujur (lattitude) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang .

Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367,45 km, dan lokasi dari 2 titik di koordinant bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut:

```
x = (lon2-lon1) * cos ((lat1+lat2)/2);
y= (lat2-lat1);
d= sqrt(x*x+y*y)*R
```

Keterangan:

x = Longitude (Lintang)
y = Lattitude (Bujur)
d = Jarak
R = Radius Bumi = 6371 km
1 derajat = 0.0174532925 radian

Lokasi 1: lon1= 119.800801, lat1= -0.790175

Lokasi 2: lon2= 119.8428 , lat2= -0.8989

lat1 = -0.790175 * 0.0174532925 radian = -0.013791155 radian

lon = 119.800801 * 0.0174532925 radian = 2.090918422 radian

lat2 = -0.8989 * 0.0174532925 radian = -0.01569 radian

lon2 = 119.8428 * 0.0174532925 radian= 2.091651 radian

x = (lon2-lon1) * cos ((lat1+lat2)/2)

= (2.091651-2.090918422) * cos ((-0.013791155 + -0.01569)/2)

= 0.0007329412

y = (lat2-lat1)

= (-0.01569- (-0.013791155))

= -0.001897609

d = sqrt (x*x + y*y) * R

= sqrt((0.0007329412 * 0.0007329412) + (-0.001897609 * -
0.001897609)) * 6371

= sqrt (0.0000041381) * 6371

=12.96012927km

1.2.5 Location Based Service

Location Based Service (LBS) atau layanan berbasis lokasi adalah sebuah layanan informasi yang dapat di akses dengan perangkat bergerak melalui jaringan dan mampu menampilkan posisi secara geografis keberadaan perangkat bergerak tersebut. *Location Based Service* dapat berfungsi sebagai layanan untuk mengidentifikasi lokasi dari seseorang atau suatu objek tertentu, seperti menemukan lokasi SPBU terdekat lainya. [9]

1.2.5.1 Unsur Utama LBS

Dua unsur utama dari *Location Based Service* adalah:

a. *Location Manager* (API Map) : Menyediakan perangkat bagi sumber atau *source* untuk LBS, *Application Programing Interface* (API) Maps menyediakan fasilitas untuk menampilakan atau memanipulasi peta paket ini berada pada "com.google.android.maps"
b. *Location Provider* (API Location) : Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat API *Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi real-time API *Location* berada pada perangkat android yaitu dalam paket "android.location". Lokasi, perpindahan, serta kedekatan dengan lokasi tertentu dapat ditentukan melalui *Location Manager*.

1.2.5.2 Komponen LBS

Terdapat lima komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain

1. Piranti Mobile

   adalah salah satu komponen penting dalam LBS ini berfungsi sebagai alat bantu (tool) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Piranti mobile yang dapat digunakan bisa berupa PDA, smartphone, laptop. Selain itu, piranti mobile dapat juga. [10]
2. Jaringan Komunikasi

   Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data data yang dikirim oleh pengguna dari piranti mobilenya untuk kemudian

dikirimkan ke penyedia layanan dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna. berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

3. Komponen Positioning

   (Penunjuk Posisi/Lokasi), Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pengolah/pemroses yang akan menentukan posisi pengguna layanan saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaringan komunikasi mobile atau juga menggunakan *Global Positioning System* GPS).

4. Penyedia layanan dan aplikasi, merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data sesuai dengan permintaan, dan masih banyak lagi yang lainnya. [10]

1.2.5.3 Cara Kerja LBS

Untuk menggambarkan cara kerja LBS, aplikasi LBS akan mencarikan informasi mengenai lokasi SPBU yang berada di sekitar posisi pengguna.

1. Fungsi pencarian telah diaktifkan, posisi pengguna sebenarnya dari perangkat mobile diperoleh dari Positioning Service. Hal ini dapat dilakukan baik oleh perangkat menggunakan GPS sendiri atau layanan posisi jaringan yang berasal dari provider (Cell Tower). Setelah itu perangkat mobile pengguna mengirimkan permintaan informasi, yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke gateway telekomunikasi.
2. Gateway memiliki tugas untuk bertukar pesan di antara jaringan komunikasi selular dan internet. Oleh karena itu dia mengetahui alamat web dari beberapa aplikasi server dan rute permintaan ke spesifik server tertentu. Gateway akan menyimpan juga informasi tentang perangkat mobile yang telah meminta informasi.
3. Aplikasi server membaca permintaan dan mengaktifkan layanan yang terkait.

4. Kemudian, *service* menganalisis lagi pesan dan memutuskan mana informasi dan posisi pengguna diperlukan untuk menjawab permintaan pengguna.
5. Selanjutnya service akan menemukan bahwa informasi lokasi SPBU.
6. Setelah sekarang semua informasi *service* akan melakukan buffer spasial dan query routing untuk mendapatkan beberapa SPBU terdekat. Setelah itu hasil dikirim kembali ke pengguna melalui internet, gateway dan jaringan mobile.
7. Kemudian, informasi mengenai SPBU akan disampaikan kepada pengguna baik dalam bentuk peta digital. [10]

1.2.5.4 Sitem satelit GPS

Untuk menginformasikan posisi user, 24 satelit GPS yang adadi orbit sekitar 12,000 mil di atas kita. Bergerak konstan bergerakmengelilingi bumi 12 jam dengan kecepatan 7,000 mil per jam.Satelit GPS berkekuatan energi sinar matahari, mempunyai bateraicadangan untuk menjaga agar tetap berjalan pada saat gerhanamatahari atau pada saat tidak ada energi matahari. Roket penguatkecil pada masing-masing satelit agar dapat mengorbit tepat padatempatnya.

Satelit GPS adalah milik Departemen Pertahanan (Department of Defense) Amerika, adapun hal-hal lainnya:

1. Nama satelit adalah NAVSTAR
2. GPS satelit pertama kali adalah tahun 1978
3. Mulai ada 24 satelit dari tahun 1994
4. Satelit di ganti tiap 10 tahun sekali
5. GPS satelit beratnya kira-kira 2,000 pounds
6. Kekuatan transmiter hanya 50 watts atau kurang

Satelit-satelit GPS harus selalu berada pada posisi orbit yangtepat untuk menjaga akurasi data yang dikirim ke GPS reciever,sehingga harus selalu dipelihara agar posisinya tepat. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Islan, DiegoGarcia, Kwajalein dan Colorado Spring. Stasiun bumi tersebut selalumemonitor posisi orbit jam jam satelit dan di pastikan selalu tepat.

1.2.5.5 Signal Sateleit GPS

Satelite GPS mengirim sinyal dalam dua frekuensi. L1 dengan1575.42 Mhz dengan membawa dua status pesan dan pseudo-random code untuk keperluan perhitungan wakt. L2 membawa1227.60 MHz dengan menggunakaan presesi yang

lebih akuratkarena untuk keperluan militer. Daya sinyal radio yang dipancarkan hanya berkisar antara 20-50 Watts. Ini tergolong sangat rendah mengingat jarak antara GPS dan satelit sampai 12.000 mil. Sinyal dipancarkan secara line of sight (LOS), dapat melewati awan, kaca tapi tidak dapat bendapadat seperti gedung, gunung.

GPS yang digunakan untuk publik akan memantau frekuensi L1 pada UHF (Ultra High Frequency) 1575,42 MHz. Sinyal L1 yangdikirimkan akan memiliki pola-pola kode digital tertentu yang disebut sebagai pseudorandom. Sinyal yang dikirimkan terdiri dari dua bagian yaitu kode Protected (P) dan Coarse/Acquisition (C/A). Kode yang dikirim juga unik antar satelit, sehingga memungkinkansetiap receiver untuk membedakan sinyal yang dikirim oleh satusatelit dengan satelit lainnya. Beberapa kode Protected (P) juga adayang diacak, agar tidak dapat diterima oleh GPS biasa. Sinyal yangdiacak ini dikenal dengan istilah Anti Spoofing, yang biasanyadigunakan oleh GPS khusus untuk keperluan tertentu seperti militer Ada sinyal frekuensi berkekuatan lemah yang di tambahkanpada kode L1 yang memberikan informasi tentang orbit satelit,clock corectionnya dan status sistem lainnya.

1.2.5.6 Cara Kerja GPS

Setiap daerah di atas permukaan bumi ini minimalterjangkau oleh 3-4 satelit. Pada prakteknya, setiap GPS terbarubisa menerima sampai dengan 12 chanel satelit sekaligus. Kondisilangit yang cerah dan bebas dari halangan membuat GPS dapatdengan mudah menangkap sinyal yang dikirimkan oleh satelit.Semakin banyak satelit yang diterima oleh GPS, maka akurasi yangdiberikan juga akan semakin tinggi.Cara kerja GPS secara logik ada 5 langkah:

1. Memakai perhitungan “triangulation” dari satelit.
2. Untuk perhitungan “triangulation”, GPS mengukur jarakmenggunakan travel time sinyal radio.
3. Untuk mengukur travel time, GPS memerlukan memerlukanakurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan pasti posisisatelit dan ketingian pada orbitnya.

5. Terakhir harus menggoreksi delay sinyal waktu perjalanan diatmosfer sampai diterima reciever.

Satelit GPS berputar mengelilingi bumi selama 12 jam didalam orbit yang akurat dia dan mengirimkan sinyal informasi kebumi. GPS reciever mengambl informasi itu dan denganmenggunakan perhitungan "triangulation" menghitung lokasi userdengan tepat. GPS reciever membandingkan waktu sinyal di kiirimdengan waktu sinyal tersebut di terima. Dari informasi itu didapatdiketahui berapa jarak satelit. Dengan perhitungan jarak jarak GPSreciever dapat melakukan perhitungan dan menentukan posisi userdan menampilkan dalam peta elektronik.

Sebuah GPS reciever harus mengunci sinyal minimal tigasatelit untuk memenghitung posisi 2D (latitude dan longitude) dantrack pergerakan. Jika GPS reciever dapat menerima empat ataulebih satelit, maka dapat menghitung posisi 3D (latitude, longitudedan altitude). Jika sudah dapat menentukan posisi user, selanjutnyaGPS dapat menghitung informasi lain, seperti kecepatan, arah yangdituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit danmatahari terbenam dan masih banyak lagi.Satelit GPS dalam mengirim informasi waktu sangat presesikarena Satekit tersebut memakai jam atom. Jam atom yang adapada satelit jalam dengan partikel atom yang di isolasi, sehinggadapat menghasilkan jam yang akurat dibandingkan dengan jambiasa.Perhitungan waktu yang akurat sangat menentukan akurasiperhitungan untuk menentukan informasi lokasi kita. Selain itusemakin banyak sinyal satelit yang dapat diterima maka akansemakin presesi data yang diterima karena ketiga satelit mengirimpseudo-random code dan waktu yang sama.Ketinggian itu menimbulkan keuntungan dalam mendukungproses kerja GPS, bagi kita karena semakin tinggi maka semakin bersih atmosfer, sehingga gangguan semakin sedikit dan orbit yangcocok dan perhitungan matematika yang cocok. Satelit harus teptappada posisi yang tepat sehingga stasiun di bumi harus terusmemonitor setiap pergerakan satelit, dengan bantuan radar yangpresesi salalu di cek tentang altitude, posision dan kecepatannya.

1.2.5.7 Bagai Mana Signal Bisa Menemukan Lokasi

Apa hubungan antara sinyal yang dikirimkan oleh satelitdengan cara GPS menentukan lokasi? Sinyal yang dikirimkan olehsatelit ke GPS akan digunakan untuk menghitung waktu perjalanan(travel time). Waktu perjalanan ini sering juga

disebut sebagai Time of Arrival (TOA). Sesuai dengan prinsip fisika, bahwa untukmengukur jarak dapat diperoleh dari waktu dikalikan dengan cepat rambat sinyal. Maka, jarak antara satelit dengan GPS juga dapat diperolehdari prinsip fisika tersebut. Setiap sinyal yang dikirimkan oleh satelit akan juga berisi informasi yang sangat detail, seperti orbit satelit, waktu, dan hambatan di atmosfir. Satelit menggunakan jam atom yang merupakan satuan waktu paling presisi. Untuk dapat menentukan posisi dari sebuah GPS secara duadimensi (jarak), dibutuhkan minimal tiga buah satelit. Empat buah satelit akan dibutuhkan agar didapatkan lokasi ketinggian (secara tiga dimensi). Setiap satelit akan memancarkan sinyal yang akanditerima oleh GPS receiver. Sinyal ini akan dibutuhkan untukme nghitung jarak dari masing-masing satelit ke GPS. Dari jarak tersebut, akan diperoleh jari-jari lingkaran jangkauan setiap satelit. Lewat perhitungan matematika yang cukup rumit, interseksi (perpotongan) setiap lingkaran jangkauan satelit tadi akan dapat digunakan untuk menentukan lokasi dari GPS di permukaan bumi.

1.2.6 Vuforia

Vuforia adalah SDK yang disediakan oleh Qualcomm untuk membantu para developer membuat aplikasi-aplikasi *Augmented Reality* (AR) di mobile phones (iOS, Android). SDK Vuforia sudah sukses dipakai di beberapa aplikasi-aplikasi mobile untuk kedua platform, AR Vuforia memberikan cara berinteraksi yang memanfaatkan kamera mobile phones untuk digunakan sebagai perangkat masukan, sebagai mata elektronik yang mengenali penanda tertentu, sehingga di layar bisa ditampilkan perpaduan antara dunia nyata dan dunia yang digambar oleh aplikasi. Dengan kata lain, Vuforia adalah SDK untuk computer vision based AR. [11]

1.3 Android

Aplikasi android terdiri dari beberapa fungsi dasar seperti mengedit catatan, memutar file musik, membunyikan alarm, atau membuka kontak telepon. Fungsi-fungsi tersebut dapat diklasifikasikan ke dalam empat komponen android yang berbeda seperti ditunjukkan pada , klasifikasi tersebut berdasarkan kelas-kelas dasar java yang digunakan [12]:

1. Applications dan Widgets

Applications dan Widgets ini adalah layer dimana berhubungan dengan aplikasi saja, dimana biasanyadownload aplikasi dijalankan kemudian dilakukan instalasi dan jalankan aplikasi tersebut.

2. Applications Frameworks

   Applications frameworks ini adalah layer di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *contect-providers* yang berupa sms dan panggilan telepon.

3. Libraries

   Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, Layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL.

4. Android Run Time

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

5. Linux Kernel

   Linux Kernel adalah layer dimana inti dari operating system dari Android itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi android lainnya. Linux kernel yang digunakan android adalah linux kernel release 2.6.

tentang versi Android (versi - nama kode - api - tanggal peluncuran) :

Android 1.0 Angel Cake (API level 1) 23 September 2008
Android 1.1 Battenberg (API level 2) 9 Februari 2009
Android 1.5 Cupcake (API level 3) 30 April 2009
Android 1.6 Donut (API level 4) 15 September 2009
Android 2.0 Eclair (API level 5) 26 Oktober 2009
Android 2.0.1 Eclair (API level 6) 3 Desember 2009
Android 2.1 Eclair (API level 7) 12 Januari 2010
Android 2.2 - 2.2.3 Froyo (API level 8) 20 Mei 2010
Android 2.3 - 2.3.2 Gingerbread (API level 9) 6 Desember 2010
Android 2.3.3 - 2.3.7 Gingerbread (API level 10) 9 Februari 2011

Android 3.0 Honeycomb (API level 11) 22 Februari 2011
Android 3.1 Honeycomb (API level 12) 10 Mei 2011
Android 3.2 Honeycomb (API level 13) 15 Juli 2011
Android 4.0 - 4.0.2 Ice Cream Sandwich (API level 14) 19 Oktober 2011
Android 4.0.3 - 4.0.4 Ice Cream Sandwich (API level 15) Desember 16, 2011
Android 4.1 Jelly Bean (API level 16) 9 Juli 2012
Android 4.2 Jelly Bean (API level 17) 13 November 2012
Android 4.3 Jelly Bean (API level 18) 24 Juli 2013
Android 4.4 KitKat (API level 19) 31 Oktober 2013

1.4 JavaScript Object Notation (JSON)

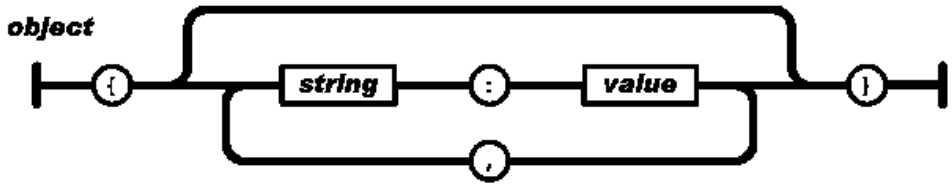
JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data [13].
JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau associative array.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemprograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemprograman yang juga berdasarkan pada struktur data ini [13].

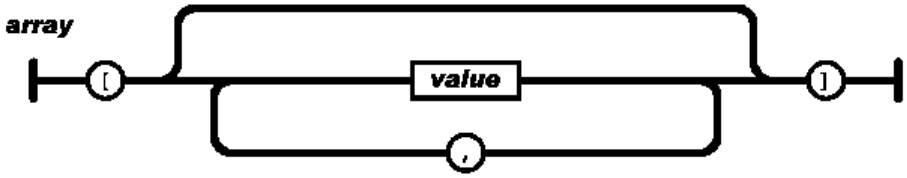
JSON menggunakan bentuk sebagai berikut:

Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan { (kurung kurawal buka) dan diakhiri dengan } (kurung kurawal tutup). Setiap nama diikuti dengan : (titik dua) dan setiap pasangan nama/nilai dipisahkan oleh , (koma).



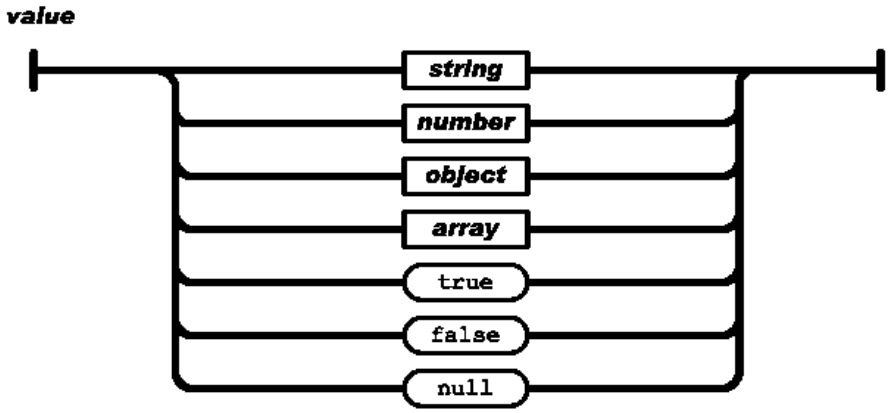
Gambar 2. 4 Objek Dalam JSON

Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).



Gambar 2. 5 Array JSON

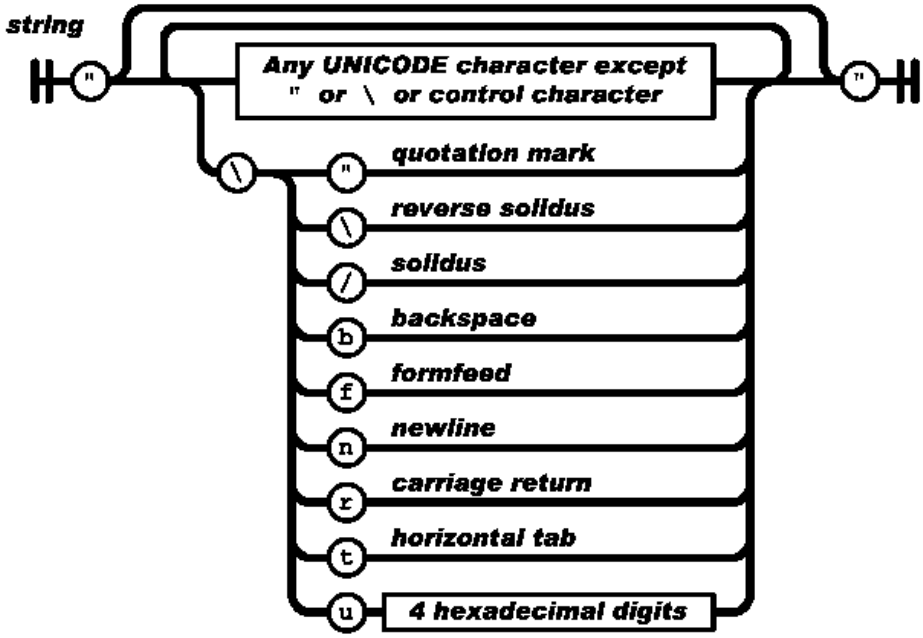
Nilai (value) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau true atau false atau null, atau sebuah objek atau sebuah larik. Struktur-struktur tersebut dapat disusun bertingkat.



Gambar 2. 6 Value JSON

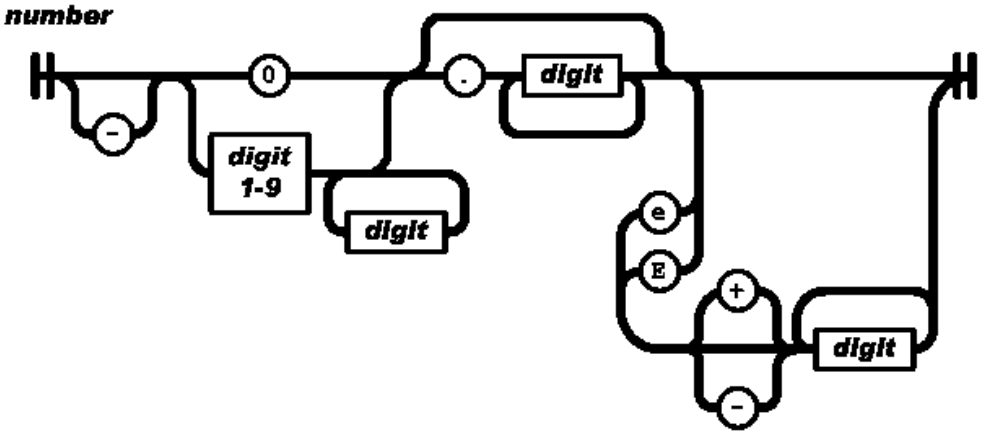
String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan backslash escapes "\"

untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string. String sangat mirip dengan string C atau Java.



Gambar 2. 7 String JSON

Angka adalah sangat mirip dengan angka di C atau Java, kecuali format oktal dan heksadesimal tidak digunakan.



Gambar 2. 8 Angka JSON

Spasi kosong (whitespace) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detil encoding yang secara lengkap dipaparkan oleh bahasa pemprograman yang bersangkutan.

2.5. *Object-oriented programming* (OOP)

Pemograman berorientasi objek (*Object Oriented Programming* – OOP) merupakan satu ide luar biasa dalam bidang pemograman yang setelah diterapkan ternyata melahirkan bahasa yang lebih maju seperti: C++, Java, Python, dsb. Perkembangan dari OOP inipun melahirkan konsep pemograman yang lain Windows atau GUI (*Graphical User Interface*) yang kemudian populer dengan bahasa visual, seperti Visual Basic, Delphi, Visual C, Visual Java, dsb [12].

Para Pemogram yang terbiasa dengan konsep bahasa prosedural seperti BASIC, PASCAL, dan C, biasanya emerlukan suatu pengenalan ke dalam konsep objek agar kemudian dapat merancang program berorientasi objek.

2.5.1. Konsep Objek

Objek dalam pengertian sehari-hari adalah benda, baik secara fisik dapat diketahui keberadaannya maupun yang bersifat khayal (virtual). Mobil Toyota, komputer Mugen, totebook Toshiba, hotel, restoran, aatau suatu gagasan dan sebagainya adalah objek.

Objek dalam pemograman adalah suatu komponen dari suatu sistem yang dikembangkan, misalnya: objek matematis, objek jendela (*window*), objek tombol (*button*), objek kotak gambar (*picture box*), objek kotak teks (*text box*), dsb. Analogi objek pada pemograman prosedural adalah “fungsi” atau “prosedur” yang bekerja mandiri menghasilkan sesuatu dalam suatu proses pertukaran data dengan program utama.

Setiap objek memiliki suatu kelompok, misalnya Toyota, Daihatsu, dan Honda adalah termasuk kelompok “mobil”. Kelompok ini dalam konsep OOP disebut “kelas” atau “*class*”.

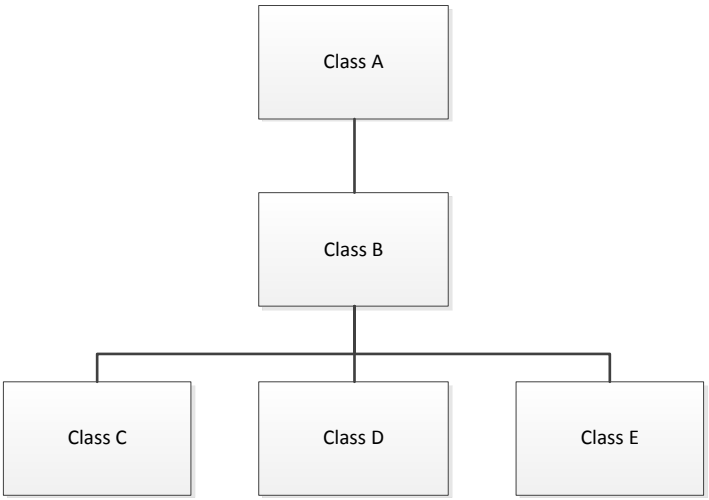
Kelas (*class*) pada hakikatnya sama dengan “cetak biru” atau “*blue print*” bagi suatu bangunan, kemudian rumah atau gedung yang dikembangkan berdasarkan cetak biru tersebut adalah objek (*object*).

Setiap kelas merupakan suatu kesatuan yang terpisah (*encapsulation*) dari kelas lainnya. Kelas/objek harus memiliki *interface* agar bisa berkomunikasi dengan kelas/objek lain.

Dalam suatu kelas terdapat dua hal utama, yaitu sifat-sifat (*attribute*) dan fungsi (*method*) atau kelakuan (behavior). Misalnya, kelas mobil memiliki atribut:

warna body, jumlah tempat duduk, ukuran ban, dan lain sebagainya, kemudian terdapat fungsi seperti pengapian, pengisian aki, percepatan (*accelarator*), rem, dan sebagainya.

Sifat-sifat dari suatu kelas bisa diwariskan (*inheritance*) ke subkelasnya. Misal, subkelas mobil sedan, mewarisi attribute dari kelas mobil. Demikian pula fungsi-fungsi kelas dapat diwariskan bila perlu. Kelas yang mewariskan sifat-sifat disebut *super-class*. Setiap subkelas dapat mengembangkan sendiri sifat serta kelakuannya sebagai tambahan dari sifat dan kelakuan utama yang diwarisi dari *super-class*.



Gambar 2. 9 *Super-Class*

Class A merupakan *super-class* dari class B. Class B merupakan subkelas dari class A. Class B merupakan superkelas dari class C, D, dan E. Class C, D dan E merupakan subkelas dari class B.

Ada beberapa konsep yang terkandung dalam pembahasan objek, antara lain:

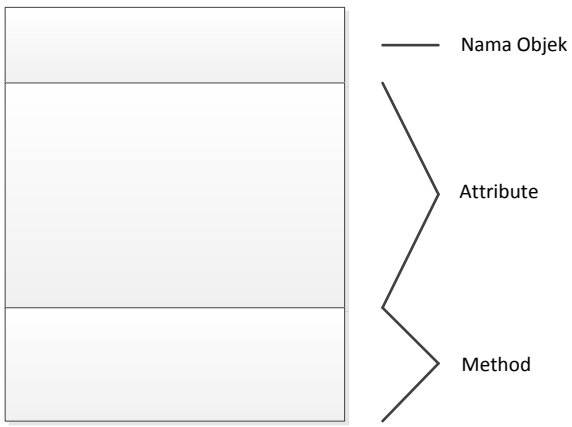
1. **Abstraction**: Karakteristik mendasar yang dimiliki oleh sebuah objek atau entitas yang membedakannya dari objek atau entitas lain.
2. **Attribute**: Sifat-sifat atau data yang dimiliki ole suatu objek.
3. **Method/Behavior**: fungsi atau kelakuan yang dimiliki oleh suatu objek.
4. **Encapsulation**: Penyembunyian data/informasi yang merupakan kesatuan dari suatu objek sehingga objek ini dapat dimanfaatkan tanpa harus tahu secara detil bagaimana objek tersebut berkelakuan. Enkapsulasi membungkus detil dari atribut dan method/fungsi yang dimiliki objek.

5. **Inheritance**: karakteristik suatu objek dapat diwariskan kepada objek lain, baik seluruh maupun sebagian.
6. **Polymorphism**: suatu nama fungsi bisa memiliki kelakuan yang berbeda-beda, merupakan suatu kemampuan dari objek untuk menyembunyikan berbagai implemantasi dari suatu nama fungsi. Polymorphism biasa juga disebut sebagai method overriding.
7. **Interface**: antarmuka yang dimiliki oleh sebuah objek. Suatu objek dapat berhubungan dengan objek lain hanya melalui interface yang disediakan.

Implementasi dari suatu kelas/objek sering dijumpai beberapa macam tipe akses (*access modifiers*) yang antara lain sebagai berikut:

1. *Public*: tidak ada pembatasan akses
2. *Protected*: akses dibatasi hanya untuk objek yang berasal dari kelas yang sama
3. *Private*: akses hanya untuk di dalam kelas itu saja.

Representasi grafis dari suatu objek dapat digambarkan sebagai satu kotak yang terdiri atas tiga bagian, yaitu: nama objek, attribute, dan method/proses.



Gambar 2. 10 Kelas

2.5.2. Objek Pada Desain Program

Pada rancangan program perlu ditetapkan beberapa tata cara dalam mendefinisikan “kelas” atau “objek” beserta attribute dan methodnya.

Suatu objek dapat didefinisikan sebagai salah satu implementasi dari suatu kelas. Cara mendefinisikannya adalah sebagai berikut:

Object nama-kelas nama-objek:

Bahasa yang berorientasi objek pada umumnya memiliki pustaka objek (*object library*) dimana terdapat berbagai kelas yang dapat dimanfaatkan oleh programmer ketika membangun suatu sistem perangkat lunak.

Proses *inheritance* atau pewarisan atribut dan metode suatu kelas ke subkelasnya dapat dilakukan melalui perintah **extends nama_kelas**.

2.5.3. Objek Graphical User Interface

Bahasa program beorientasi visual, seperti visual basic, visual c, visual java, delphi, visual dbase, visual foxpro, dan sebagainya. Pada hakikatnya bahasa-bahasa tersebut berorientasi objek, dimana tersedia objek-objek untuk visualisasi pada layar monitor yang populer dengan istilah *Graphical User Interface* (GUI), seperti objek jendela (*window*, *panel*, atau *form*), objek teks (*textbox*, *label*), objek daftar pilihan (*list*, *combo-box*, *spin-box*), objek tombol (*button*), dsb. Setiap objek memiliki atribut (*properties*) dan method (*function*). Objek-objek dapat berkomunikasi melalui *interface*, dan *events*. Events adalah kejadian/program yang diaktifkan melalui satu peristiwa, misalnya karena mouse ditekan (*on_Click*), atau isi suatu kotak data_entry diubah (*on_Change*), suatu jendela atau suatu file dibuka (*on_Open*), atau ditutup (*on_Close*), dsb.

Suatu pustaka objek bisa memutar berbagai macam objek, misalkan pustaka objek jendela (*window tool kit*) bisa memuat kelas-kelas untuk objek-objek yang disebutkan di atas. Untuk menggunakan objek-objek tersebut maka pustakanya harus dipanggil terlebih dahulu, misalnya dengan intruksi **import** seperti contoh berikut:

import window;	memanggil semua objek dari pustaka window
import window.panel;	memanggil hanya objek panel saja
import window.list;	memanggil hanya objek list saja

Beberapa fungsi atau method yang dapat diterapkan pada List adalah:

addItem(pilihan)	:	fungsi untuk mengisi list dengan pilihan-pilihan
getItem(posisi)	:	fungsi untuk memperoleh yteks pilihan pada posisi yang diberikan

countItems()	:	fungsi untuk mengetahui berapa banyak pilihan dalam list
getSelectedIndexes()	:	fungsi untuk mengetahui indeks(nomor) dari pilihan tertentu
getSelectedItem()	:	fungsi untuk mengetahui pilihan yang saat ini terpilih
getSelectedItems()	:	fungsi untuk mengetahui semua pilihan yang terpilih
select(nomor)	:	pilih pilihan pada nomor yang diberikan
select(pilihan)	:	Pilih pilihan yang sama dengan string pilihan yang diberikan

2.6. Java

Java adalah bahasa yang berorientasi objek, bersumber dari C++, namun dibuat agar lebih mudah dari C++ dengan membuang beberapa unsur yang sulit, seperti pointer dan alokasi memori, kemudian menambahkan beberapa unsur penting sehingga bahasa ini bisa diterima pada berbagai macam mesin (*multiplatform* [12]. Java disimpulkan dalam satu kalimat: "Java: a simple, object oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language" [The Java Language: A White Paper, Sun's Microsystem].

Tahun 1995, saat dunia internet mulai berbunga, maka Sun Microsystem mengumumkan bahasa Java SDK versi 1.0 dalam suatu konfersi sebagai suatu "*freeware*", terbuka bagi siapa saja. Sejak itulah java berkembang dan banyak digunakan dalam tampilan Web sebagai bentuk dinamisasi dan interaksi dari Web.

Jenis pustaka yang tersedia cukup banyak, beberapa di antaranya yang penting adalah:

java.io	pustaka untuk fungsi input-output
java.lang	pustaka yang memuat elemen dasar bahasa Java
java.aw	pustaka yang membuat objek-objek virtual

Beberapa tipe data dasar yang dikenal dalam bahasa Java:

boolean : tipe data logik, true atau false
char : tipe data karakter tunggal
byte : bilangan bulat -128 s/d 127
short : bilangan bulat -32768 s/d 32767

int : bilangan bulat 32bit (>2 milyar)

long : bilangan bulat 64bit (sangat besar)

float : bilangan biasa 32 bit

double : bilangan biasa 64 bit

string : deretan karakter

Java dipersiapkan agar bisa dijalankan pada berbagai macam komputer, maka intruksi input-outputnya sedikit lebih rumit dari intruksi bahasa lain. Penanganan input-output umumnya berbeda pada jenis komputer yang berbeda, sehingga JAVA mensyaratkan untuk memeriksa kegiatan input-output sebelum melaksanakan operasi input-output.

2.7. Basis Data

Basis data adalah suatu koleksi data yang saling berhubungan secara logis dan menggambarkan integrasi antara suatu tabel dengan tabel lainnya, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi [14].

Berikut ini merupakan alasan dari penggunaan database:

1. Padat.
   Tidak perlu lagi membuat arsip kertas dalam ukuran besar.
2. Kecepatan
   Mesin dapat mendapatkan kembali dan mengubah data jauh lebih cepat daripada yang manusia yang dapat lakukan.
3. Mengurangi pekerjaan yang membosankan
   Rasa bosan dari proses memelihara arsip – arsip berupa kertas dapat dikurangi.
4. Aktual
   Informasi yang terbaru dan akurat selalu tersedia disetiap waktu ketika dibutuhkan.

Terdapat delapan keuntungan dengan menggunakan pendekatan database, yaitu:

1. Redundansi dapat dikurangkan
2. Ketidakkonsistenan dapat dihindari
3. Data dapat dibagikan.
4. Standar-standar dapat diselenggarakan.
5. Pembatasan keamanan dapat diselenggarakan.

6. Integritas dapat dipertahankan.
7. Keperluan yang bertentangan dapat diseimbangkan.
8. Tersedianya dukungan untuk transaksi.

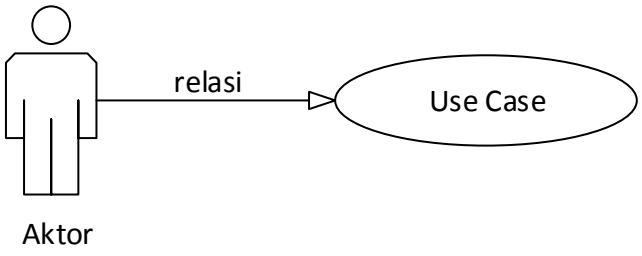
Pada umumnya data dalam database bersifat integrated dan shared. Maksud dari integrated adalah database merupakan penggabungan beberapa file data yang berbeda, dengan membatasi pengulangan baik keseluruhan file ataupun sebagian. Pengertian shared artinya adalah data individu dalam database dapat digunakan secara bersamaan antara beberapa pengguna yang berbeda.

2.8. Unified Modeling Language (UML)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. Berikut diagram-diagram tersebut :

1. Use Case Diagram

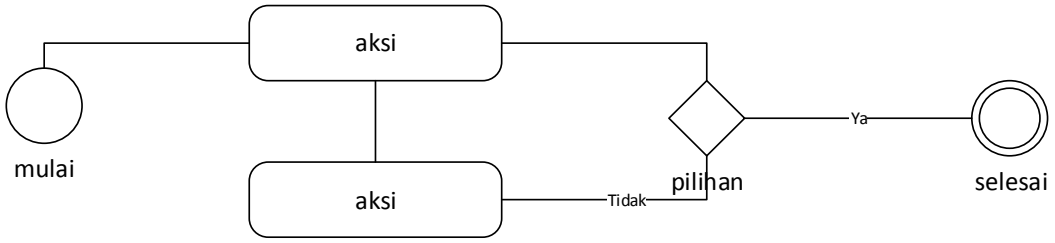
Use Case Diagram merupakan gambaran actor, use case dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Setiap use case terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case. Dapat dilihat pada gambar 2.4.



Gambar 2. 11 Notasi Utama Diagram *Use Case*

2. Activity Diagram

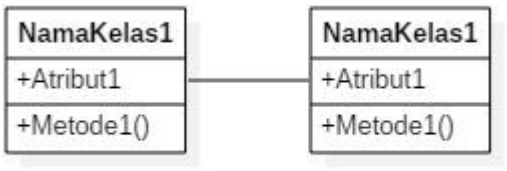
Activity Diagram merupakan gambaran aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas. Dapat dilihat pada gambar 2.5.



Gambar 2. 12 Notasi *Activity Diagram*

3. Class Diagram

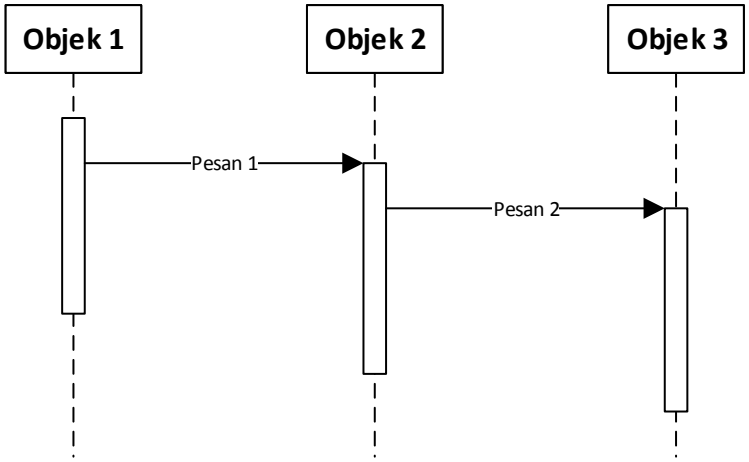
Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. Class memiliki tiga area pokok yaitu nama (dan *stereotype*), atribut, metoda. Dapat dilihat pada gambar 2.6.



Gambar 2. 13 Notasi *Class Diagram*

4. Sequence Diagram

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya sequence diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram. Notasi utama dari sequence diagram dapat dilihat pada gambar 2.7.



Gambar 2. 14 Notasi Sequence Diagram

2.9. Perangkat Lunak Pendukung

2.9.1. NetBeans

NetBeans merupakan salah satu IDE yang dikembangkan dengan bahasa pemrograman java. NetBeans mempunyai lingkup pemrograman java terintergrasi dalam suatu perangkat lunak yang di dalamnya menyediankan pembangunan pemrograman GUI, text editor, complier, dan interpreter. NetBeans adalah sebuah perangkat lunak open source sehingga dapat digunakan secara gratis untuk keperluan komersial maupun nonkomersial yang didukung oleh Sun Microsystem.

Beberapa bahasa yang didukung oleh NetBeans IDE ini terdiri dari Java, C/C++, PHP, XML, HTML, Javadoc, Javascript, JSP, dan masih banyak lagi. Selain itu, pembaca dapat memasang plugin, maupun modul yang bisa didapatkan di komunitas untuk mendukung bahasa lain agar dapat dijalankan di NetBeans IDE ini.

2.9.2. Wamp Server

WAMP Server adalah singkatan dari Windows, Apache, MySQL dan PHP. WAMP Server adalah aplikasi yang menggabungkan antara Apache, MySQL dan PHP untuk menjalankan fungsinya sebagai server yang berdiri sendiri. Program WAMP tersedia dalam GNU (General Public License) dan gratis. Program WAMP merupakan web server yang mudah digunakan yang dapat mampu melayani halaman dinamis.

2.9.3. SQLyog

SQLyog adalah aplikasi client MySQL yang sangat populer digunakan di Indonesia. Aplikasi ini memiliki banyak fitur yang memudahkan pengguna melakukan administrasi maupun melakukan pengolahan data MySQL.

SQLyog ini aplikasi yang digunakan untuk menghubungkan satu komputer dengan komputer lain, agar satu komputer dengan komputer dapat saling mengakses dapat dikatakan aplikasi adalah aplikasi yang bisa digunakan untuk menerapkan client server.

Dengan kombinasi PHP dan MySQL menjadi salah satu standar untuk pengembangan Web, kebutuhan untuk frontend yang baik untuk MySQL meningkat. Bagi banyak orang, tugas menciptakan dan memanipulasi database MySQL adalah tugas yang menakutkan. Hal ini sering sulit untuk menguasai perintah kompleks yang dibutuhkan oleh konsol MySQL dan MySQL Admin. Untungnya ada harapan. Karena popularitas yang semakin meningkat dari MySQL dan PHP, banyak alat pihak ketiga sedang dikembangkan khusus untuk platform ini.

2.10. Pengujian Aplikasi

Tujuan utama dari pengujian aplikasi ini adalah untuk meyakinkan kepada pelanggan bahwa sistem tersebut layak untuk digunakan dan memenuhi kinerja persyaratan [15]. Berikut ini merupakan penjelasan mengenai pengujian :

2.10.1. Black Box Testing (Pengujian Kotak Hitam)

Black Box Testing adalah pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinsikan [16].

Metode Black Box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Black Box dapat menemukan kesalahan dalam kategori berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam strutur data atau akses basisdata eksternal
4. Inisialisasi dan kesalahan terminasi

5. validitas fungsional
6. kesensitifan sistem terhadap nilai input tertentu
7. batasan dari suatu data

2.10.2. Wawancara

Wawancara adalah suatu proses komunikasi dengan tujuan yang serius dan ditetapkan terlebih dulu yang dirancang untuk mempertukarkan perilaku dan melibatkan tanya jawab [16].

Namun definisi tsb agak terbatas, karena wawancara membatasi wawancara dengan tujuan yang serius. Wawancara juga telah menjadi bentuk hiburan yang populer seperti disiarkan televisi dan radio.

1.5 Diagram Konteks

Diagram Konteks adalah data flow diagram tingkat atas (DFD Top Level), yaitu diagram yang paling tidak detail, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan ke luar sistem dan ke dalam dan ke luar entitas-entitas eksternal. (CD menggambarkan sistem dalam satu lingkaran dan hubungan dengan entitas luar. Lingkaran tersebut menggambarkan keseluruhan proses dalam sistem) .

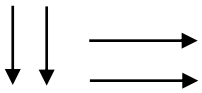
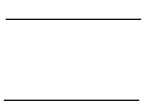
1.6 Data Flow Diagram

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut, dan asal, tujuan, dan penyimpanan dari data tersebut. DFD terdiri dari empat komponen utama diantaranya adalah :

1. Entitas.
2. Proses.
3. Data Store.
4. Data Flow.

DFD dapat dikatakan merupakan breakdown dari diagram konteks. Analisis mengenai sistem yang lebih detail dapat dilakukan dengan melakukan breakdown pada DFD level 1, 2 dan seterusnya. Berikut adalah Tabel 2.1 Simbol Diagram Konteks dan DFD.

Tabel 2. 1 Simbol Diagram Konteks dan DFD

Simbol	Nama Simbol	Fungsi
	Entitas Luar	Menggambarkan entitas eksternal yang berhubungan dengan sistem
	Sistem (Konteks) / Proses (DFD)	Menggambarkan proses yang ada dalam suatu sistem
	Aliran Data/Informasi	Menggambarkan aliran data antar proses, data store dan entitas luar
	Data Storage	Mengambarkan tempat penyimpanan data di dalam sistem

1.7 Spesifikasi Proses

Spesifikasi Proses adalah tabel yang berisi keterangan atau deskripsi dari semua proses yang terdapat pada DFD. Logika proses yang ditulis pada spesifikasi proses harus dituliskan secara jelas baik menggunakan bahasa deskriptif atau pseudecode tetapi tidak boleh mengkombinasikan kedua bahasa tersebut . Berikut adalah Tabel 2.2 Contoh Spesifikasi Proses.

Tabel 2. 2 Contoh Spesifikasi Proses

No	Proses	Keterangan
1.	No Proses	Menyatakn no proses
	Nama Proses	Menyatakan nama proses
	Source (Sumber)	(Menyatakan sumber data input menuju proses)
	Input	(menyatakan isi data yang masuk ke proses)
	Output	(menyatakan informasi yang keluar dari proses)
	Destination	(menatakan tujuan informasi output dari proses)
	Logika Proses	(menyatakan algoritma dari proses)

1.8 Kamus Data

Kamus data adalah tabel yang berisi deskripsi dari data yang mengalir pada DFD, Penjelasan struktur data dapat berupa field tiap data harus sama dengan yang sudah dimodelkan di ERD. Berikut adalah Tabel 2.3 Contoh Kamus Data.

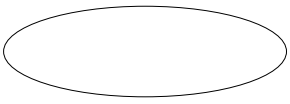
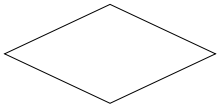
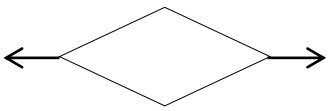
Tabel 2. 3 Contoh Kamus Data

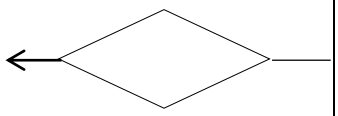
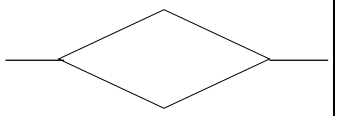
Nama aliran data	Data Siswa Nama dari data yang digunakan
Where used/How used	Daftar pada poin proses apa data digunakan
Keterangan	Uraian singkat dari data yang digunakan
Struktur Data	Daftar komponen data yang ada pada data yang digunakan
Deskripsi	Jenis data dalam representasi komputer untuk masing-masing data.

1.9 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (entity) serta hubungan (relationship) antar entitas-entitas tersebut dengan menggunakan beberapa notasi. Komponen-komponen pembentuk ERD dapat di lihat pada Tabel 2.4 Komponen-komponen ERD .

Tabel 2. 4 Komponen-komponen ERD

Notasi	Komponen	Keterangan
	Entitas	Individu yang mewakili suatu objek dan dapat dibedakan dengan objek yang lain
	Atribut	Properti yang dimiliki oleh suatu entitas, dimana dapat mendeskripsikan karakteristik darri entitas tersebut
	Relasi	Menunjukkan hubungan diantara sejumlah entitas yang berbeda
	Relasi 1 : 1	Relasi yang menunjukkan bahwa setiap entitas pada himpunan entitas pertama berhubungan dengan paling banyak satu entitas pada himpunan entitas kedua

Notasi	Komponen	Keterangan
	Relasi 1 : N	Relasi yang menunjukkan bahwa hubungan antara entitas pertama dengan entitas kedua adalah satu banding banyak atau sebaliknya. Setiap entitas dapat berelasi dengan banyak entitas pada himpunan entitas yang lain
	Relasi N : N	Hubungan ini menunjukkan bahwa setiap entitas yang pertama dapat berhubungan dengan banyak entitas pada himpunan entitas yang kedua, demikian juga sebaliknya

1.10 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp. Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja query cepat, dan mencukupi untuk kebutuhan database perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat open source (tidak berbayar). MySQL merupakan database yang pertama kali didukung oleh bahasa pemrograman script untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pembangun aplikasi web yang ideal.