BAB II

LANDASAN TEORI

2.1 Shalat

Dalam mengerjakan shalat kita harus megetahui aturan/ rukun shalat atau tata cara mengerjakan shalat yang benar. Berikut adalah penjelasan mengenai shalat.

2.1.1 Pengertian Shalat

Menurut H. Fachrurazi (2002 : 27) pengertian shalat menurut makna adalah doa. Yang dimaksud disini adalah ibadah yang tersusun dari beberapa perkataan dan perbuatan yang dimulai dengan *takbir* dan sudahi/diakhiri dengan salam, serta memenuhi syarat-syarat yang telah ditentukan.

Firman Allah SWT :

"Kerjakanlah Shalat, sesungguhnya shalat itu mencegah perbuatan keji dan munkar. (Al-Ankabut ayat 45)

Perintah shalat hendaklah ditanamkan di dalam hati dan jiwa anak-anak mulai dari kecil, dengan cara yang cermat.

Sabda Rasulullah SAW :

"Perintahkanlah anak-anakmu mengerjakan shalat di waktu usia mereka menginjak tujuh tahun, dan engkau boleh memukulnya bila mereka enggan mengerjakan shalat ddi waktu mereka meningkat usia sepuluh tahun". (H.R. Abu Dawud)

Dasar kewajiban shalat di antaranya adalah firman Allah SWT :

"Dan dirikanlah shalat, keluarkanlah zakat dan tunduklah/ruku'lah bersama orang-orang yang ruku'." (Q.S. Al-Baqarah : 43)

2.1.2 Pengertian Shalat Fardhu

Shalat Fardhu Yaitu shalat yang diwajibkan Allah SWT kepada hamba-hamba-Nya sesuai batasan-batasan yang telah dijelaskan-Nya, baik melalui perintah maupun larangan. Dalam hal ini adalah shalat 5 waktu dalam sehari semalam.

2.1.3 Waktu Shalat Fardhu

Dalam melaksanakan ibadah shalat, semua diatur dalam Kitab Suci Al-Qur'an dan hadist Nabi Muhammad SAW.

Allah SWT berfirman :

"…Sesungguhnya shalat itu adalah fardhu yang ditentukan waktunya atas orang-orang yang beriman." (Q.S. An-Nisa' : 103)

Allah SWT juga berfirman dalam Al'quran yang artinya :

"Dirikanlah sholat ketika gelincir matahari hingga waktu gelap malam dan dirikanlah sholat subuh sesungguhnya sholat subuh itu adalah disaksikan (keistimewaannya)". (Q.S. Al-Isra' : 78)

Dalil diatas menjelaskan untuk mengalokasikan waktu-waktu shalat. Berikut ini merupakan rincian waktu-waktu shalat fardhu :

1. Shalat Shubuh

   Waktunya diawali saat *Fajar Shiddiq* sampai matahari terbit (syuruk). *Fajar Shiddiq* ialah terlihatnya cahaya putih yang melintang mengikut garis lintang ufuk di sebelah Timur akibat pantulan cahaya matahari oleh atmosfer.

2. Shalat Dzuhur

   Disebut juga waktu Istiwa (*zawaal*) terjadi ketika matahari berada di titik tertinggi. Istiwa juga dikenal dengan sebutan Tengah Hari (*midday*/*noon*). Pada saat Istiwa, mengerjakan ibadah shalat (baik wajib maupun sunnah) adalah haram. Waktu Zuhur tiba sesaat setelah Istiwa, yakni ketika matahari telah condong ke arah Barat.

3. Shalat Ashar

   Menurut Mazhab Syafi'i, Maliki, dan Hambali, waktu Ashar diawali jika panjang bayang-bayang benda melebihi panjang benda itu sendiri. Sementara Madzab Imam Hanafi mendefinisikan waktu Ashar jika panjang bayang-bayang benda dua kali melebihi panjang benda itu sendiri. Waktu Ashar dapat dihitung dengan algoritma tertentu yang menggunakan trigonometri tiga dimensi.

4. Shalat Maghrib

   Diawali saat matahari terbenam di ufuk sampai hilangnya cahaya merah di langit Barat.

5. Shalat Isya Diawali dengan hilangnya cahaya merah (*syafaq*) di langit Barat, hingga terbitnya Fajar Shiddiq di Langit Timur.

Sebelum manusia menemukan hisab/perhitungan falak/astronomi, pada zaman Rasulullah waktu shalat ditentukan berdasarkan observasi terhadap gejala alam dengan melihat langsung matahari. Lalu berkembang dengan dibuatnya Jam Surya atau Jam Matahari serta Jam Istiwa atau seing disebut Tongkat Istiwa dengan kaidah bayangan matahari.

Sumber : http://rukyatulhilal.org/waktu-shalat/index.html/ 16 Mei 2012

Syarat sah shalat lima waktu yaitu :

1. Islam
2. Suci dari hadast, haid dan nifas, juga suci seluruh anggota badan, pakaian, dan tempat.
3. Berakal dan balig (dewasa)
4. Menutup aurat

   Bagi laki-laki : auratnya antara pusat sampai lutut

   Bagi perempuan :seluruh badan, kecuali muka dan kedua telapak tangan.
5. Mengetahui masuknya waktu shalat yang telah ditentukan
6. Menghadap kiblat
7. Mengerti mana yang rukun dan mana yang sunnah.

Adapun rukun Shalat adalah sebagai berikut :

1. Niat, karena mengikuti perintah Allah semata
2. Berdiri bagi orang yang kuasa
3. Takbiratul ihram (membaca *Allaahu Akbar*)
4. Memmbaca surat Al-Fatihah setiap rakaat

5. Ruku' serta tuma'ninah (diam sebentar)
6. I'tidal serta tuma'ninah (diam sebentar)
7. Sujud dua kali serta tuma'ninah (diam sebentar)
8. Duduk di antara dua sujud serta tuma'ninah (diam sebentar)
9. Duduk tasyahud akhir serta tuma'ninah (diam sebentar)
10. Membaca tasyahud akhir
11. Membaca shalawat Nabi ketika tasyahud akhir
12. Membaca salam yang pertama sambil berpaling ke kanan
13. Menertibkan rukun, artinya meletakkan rukun pada tempatnya menurut susunan yang telah di tetapkan.

2.1.4 Pengertian Shalat Sunnah

Shalat sunnah (shalat nafilah) adalah shalat tambahan diluar shalat fardhu, bila dikerjakan akan mendapat pahala tetapi bila ditinggalkan tidak berdosa. Shalat sunnah terbagi dua yaitu:

1. Shalat sunnah yang dilaksanakan secara berjamah. Shalat sunnah jenis ini status hukumnya adalah *muakkad,* contohnya: shalat idul fitri, idul adha.

   Sumber:http://www.nurulilmi.com/component/content/article/17-fiqih/589-macam-macam-sholat.html/ 28 Mei 2012
2. Shalat sunnah yang dikerjakan secara munfarid (sendiri-sendiri). Status hukumnya ada yang *muakkad* seperti : shalat sunnah rawatib. Shalat sunnah rawatib adalah shalat sunat yang di kerjakan sebelum dan sesudah shalat fardhu. Ada pula yang status

hukumnya sunnah biasa (ghairu muakkad) seperti: shalat tahiyatul masjid, shalat dhuha, shalat witir, dan lain-lain.

2.1.5 Menghitung Waktu Shalat

Dalam menentukan waktu shalat wajib lima waktu di suatu tempat pada tanggal tertentu, ada beberapa parameter yang mesti diketahui :

1. Koordinat lintang tempat tersebut (L). Daerah yang terletak di sebelah utara garis khatulistiwa (ekuator) memiliki lintang positif. Sebaliknya, untuk yang disebelah selatan lintangnya negatif.
2. Koordinat bujur tempat tersebut (B) atau *longitude*. Daerah yang terletak disebelah timur Greenwich memiliki bujur positif.
3. Zona waktu tempat tersebut (Z). Daerah yang terletak di sebelah timur Greenwich memiliki Z positif.
4. Ketinggian lokasi dari permukaan laut (H). Ketinggian lokasi dari permukaan laut (H) menentukan waktu kapan terbit dan terbenamnya matahari. Tempat yang berada tinggi di atas permukaan laut akan lebih awal menyaksikan matahari terbit serta lebih akhir melihat matahari terbenam, dibandingkan dengan tempat yang lebih rendah. Satuan H adalah meter.
5. Tanggal (D), Bulan (M) dan Tahun (Y). Merupakan parameter yang diperlukan untuk waktu shalat pada tanggal tersebut. Dari tanggal, bulan dan tahun selanjutnya di hitung nilai Julian Day (JD). Dengan rumus sebagai berikut:

JD=1720994,5+INT 365,25×Y +INT 30,6001 M+1 +B+D (1)

Keterangan :

INT : Lambang nilai *integer* (bilangan bulat)

Jika M > 2, maka M dan Y tidak berubah.

Jika M = 1 atau M = 2, maka M +12 dan Y dikurangi 1

$$\mathrm{B} = 2 + \mathrm{INT}\left(\frac{\mathrm{A}}{4}\right) - \mathrm{A}, \text{dimana } \mathrm{A} = \mathrm{INT}\left(\frac{\mathrm{Y}}{100}\right)$$

Nilai JD berlaku untuk pukul 12.00 UT atau saat tengah hari di Greenwich. Untuk JD yang digunakan dalam perhitungan yaitu JD lokasi tempat yang ingin ditentukan waktu shalat. Diperoleh dari JD pukul 12.00 UT waktu Greenwich dikurangi dengan Z/24, dimana Z adalah zona waktu lokal tersebut.

6. Sudut Deklinasi Matahari (Delta). Deklinasi matahari (Delta) untuk satu tanggal tertentu dapat dihitung dengan menggunakan rumus berikut:

   Delta = 0,37877 + 23,264 * sin(57,297 * T – 79,547) + 0,3812 * sin(2 * 57,297 * T – 82, 682) + 0,17132 * sin(3 * 57,297 * T – 59,722) (2)

   Keterangan :

   T adalah sudut tanggal, dengan rumus, T = 2 * PI * (JD -2451545) /365,25

7. *Equation of Time* (ET). *Equation of Time* untuk satu tanggal tertentu dapat dihitung dengan rumus :

   ET = (- (1789 + 237 * U) * sin(L0) – (7146 – 62 * U) * cos(L0) + (9934 – 14 * U) * sin(2 * L0) – (29 + 5 * U) * cos(2 * L0) + (74 + 10 * U) * sin(3 * L0) + (320 – 4 * U) * cos(3 * L0) – 212 * sin(4 * L0))/1000 (3)

Keterangan :

L0 adalah Bujur rata-rata matahari, L0 = 280,46607 + 36000,7698 * U.

U = (JD – 2451545)/36525.

8. *Altitude* matahari waktu Shubuh dan Isya. Shubuh saat fajar menyingsing pagi disebut *dawn astronomical twilight* yaitu ketika langit tidak lagi gelap dimana atmosfer bumi mampu membiaskan cahaya matahari dari bawah ufuk. Sementara Isya' disebut *dusk astronomical twilight* ketika langit tampak gelap karena cahaya matahari di bawah ufuk tidak dapat lagi dibiaskan oleh atmosfer. Nilai *altitude* matahari berasal dari ketika langit berubah dari gelap menjadi mulai terang, ketika fajar menyingsing di pagi hari dan menyebar secara horisontal dengan seragam. *Altitude* matahari sangat menentukan metode perhitungan waktu shalat, dimana perbedaan 1 derajat dapat memberikan perbedaan waktu sekitar 4 menit. Terdapat beberapa pendapat mengenai nilai altitude matahari

Tabel 2.1 Altitude Matahari Saat Shubuh dan Isya

Organisasi	Sudut Shubuh	Sudut Isya	Regional
Indonesia	20 derajat	18 derajat	Indonesia
Universitas Sience Islam, Karaci	18 derajat	18 derajat	Pakistan, Bangladesh, India, Afghanistan, sebagian Eropa
Amerika Utara	15 derajat	15 derajat	Sebagian Amerika Serikat, Kanada, sebagian Inggris
Liga Muslim Dunia	18 derajat	17 derajat	Eropa, sebagian Amerika
Komite Umm Al-Qura'an	18.5 derajat	18.5 derajat	Semenanjung arab
Mesir	19.5 derajat	17.5 derajat	Afrika, siria, Irak, libanon, Malaysia, sebagian Amerika

9. Tetapan panjang bayangan ashar, dalam hal ini terdapat dua pendapat berbeda. pendapat mazhab Imama Syafi'i menyatakan panjang bayangan benda saat

ashar adalah tinggi benda di tambah panjang bayangan saat zuhur. Sementara mazhab Imam Hanafi menyatakan panjang bayangan benda saa ashar sama dengan dua kali inggi benda di tambah panjang bayangan saat zuhur.

Setiap parameter sangat menentukan datangnya waktu shalat, bila salah satu parameter kurang akurat maka maka ketepatan datangnya waktu shalat akan sebanding.

Waktu shalat dapat ditentukan dengan menggunakan rumus-rumus pergerakkan matahari dengan tepat. Berikut adalah rumus waktu shalat.

a. Shubuh = waktu zuhur – (12/PI) * acos((sin(-1 * sudut altitude matahari subuh) – sin(delta) * sin(L)) / (cos(delta) * cos(L))
b. Zhuhur = 12 + Z – B / 15 – ET/60
c. Ashar = waktu zuhur + (12/PI) * acos((sin(atan(1/MA + tan (abs (L-delta))))) – sin delta * sin (L))/(cos(delta)*cos(L)))

   Dimana MA merupakan mazhab yang digunakan, MA sama dengan 1 untuk mazhab imam syafi'i, dan MA 2 sama dengan untuk mazhab imam hanafi.
d. Magrib = waktu zuhur + (12/PI) * acos((sin((-0,8333 - 0,0347 * H ^ 0,5)) – sin (delta) * sin (L)) /(cos(delta) * cos(L))).
e. Isya' = waktu zhuhur + (12/PI) * acos((sin(-1 * sudut altitude matahari isya') - sin(delta)* sin (L)) / cos(delta)* cos(lintang)))

2.2 Java

Dibawah ini adalah penjelasan mengenai pengertian bahasa pemograman java dan sistem operasi java

2.2.1 Pengertian Java

sebuah teknologi yang diperkenalkan oleh Sun Microsystems pada pertengahan tahun 1990. Menurut definisi dari Sun, Java adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan. Kita lebih menyukai menyebut Java sebagai sebuah teknologi dibanding hanya sebuah bahasa pemrograman, karena Java lebih lengkap dibanding sebuah bahasa pemrograman konvensional. Teknologi Java memiliki tiga komponen penting, yaitu:

a. Programming-language specification
b. Application-programming interface
c. Virtual-machine specification

Sumber : http://www.ombar.net/2010/02/pengertian-java.html / 1 Agustus 2012

2.2.2 Sistem Operasi Java

Sistem operasi biasanya ditulis dalam sebuah kombinasi dari kode bahssa C dan assembly, terutama disebabkan oleh kelebihan performa dari bahasa tersebut dan memudahkan komunikasi dengan perangkat keras. Satu kesulitan dalam merancang sistem basis bahasa adalah dalam hal proteksi memori, yaitu memproteksi sistem opeasi dari pemakai program yang sengaja memproteksi pemakai program lainnya. Sistem operasi tradisional mengaharapkan pada tampilan perangkat keras untuk menyediakan proteksi memori. Sistem basis

bahasa mengandalkan pada tampilan keamanan dari bahasa. Sebagai hasilnya, sistem basis bahasa menginginkan pada alat perangkat keras kecil, yang mungkin kekurangan tampilan perangkat keras yang menyediakan proteksi memori. Sumber : http://www.ombar.net/2010/02/pengertian-java.html /1 Agustus 2012

2.3 Aplikasi

Berikut ini merupakan penjelasan mengenai pengertian aplikasi, juga penggolongan aplikasi serta defenisi rancang bangun perangkat lunak

2.3.1 Pengertian Aplikasi

Aplikasi merupakan sekumpulan elemen yang saling berinteraksi dan saling berkaitan antara satu dengan yang lainnya dalam melakukan suatu kegiatan secara bersama-sama untuk mencapai tujuan tertentu.

Pada Wikipedia (id.wikipedia.org) aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan langsung utnuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer.

2.3.2 Penggolongan Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain :

1. Perangkat lunak perusahaan (*enterprise*).
2. Perangkat lunak infrastruktur perusahaan.
3. Perangkat lunak informasi kerja.
4. Perangkat lunak media dan hiburan.
5. Perangkat lunak pendidikan.
6. Perangkat lunak pengembangan media.

Pada pengertian umumnya, aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Sumber : http://tiaseptianawidi.blogspot.com/2012/03/pengertian aplikasi.html/ 5 Juni 2012

2.3.3 Defenisi Rancang Bangun

Definisi awal dari rancang-bangun perangkat lunak diutarakan oleh Fritz Bauer dalam suatu konferensi, dimana pada saai itu dijadikan sebagai pokok materi yang akan dibahas. Definisinya adalah : *The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.* (Roger S Pressman, Software Engineering A Practitioner's Approach: Hal 19 : Singapore 1987)

Dari sekian banyak definisi yang telah diusulkan, semua definisi tersebut mengacu pada pentingnya rancang-bangun dalam pengembangan software. Rancang-bangun perangkat lunak merupakan suatu pertumbuhan diluar perangkat keras dan rancang-bangun sistem yang meliputi tiga kunci pokok yaitu : cara, alat dan prosedur yang memungkinkan manajer untuk mengendalikan proses pengembangan software dan menyediakan praktisi dengan tujuan untuk membangun perangkat lunak yang berkualitas.

Sumber : http://irhamnaa-latu.blogspot.com/ 2 Agustus 2012

2.4 Kiblat

Dalam melaksanakan ibadah shalat, yang harus di ketahui ummat muslim sebelum menunaikan shalat adalah arah kiblat. Berikut ini merupakan penjelasan yang berhubungan dengan kiblat.

2.4.1 Pengertian Kiblat

Menurut Ivan Michael Siregar 2011 Kiblat berasal dari bahasa Arab *"Qiblah"* adalah arah yang merujuk ke suatu tempat dimana bangunan Ka'bah di Masjidil Haram, Makkah, Arab Saudi. Ka'bah juga sering disebut dengan *Baitullah* (Rumah Allah). Menghadap arah kiblat merupakan suatu masalah yang penting dalam syariat Islam. Menurut hukum syariat, menghadap ke arah kiblat diartikan sebagai seluruh tubuh atau badan seseorang menghadap ke arah Ka'bah yang terletak di Makkah yang merupakan pusat tumpuan umat Islam bagi menyempurnakan ibadah-ibadah tertentu.

Pada awalnya, kiblat mengarah ke Baitul Maqdis atau Masjidil Aqsa Jerussalem di Palestina, namun pada tahun 625 M ketika Nabi Muhammad SAW hijrah ke Madinah, arah Kiblat berpindah ke arah Ka'bah di Makkah hingga kini atas petunjuk wahyu dari Allah SWT. Menghadap ke arah kiblat menjadi syarat sah bagi umat islam yang hendak menunaikan shalat baik shalat fardhu lima waktu sehari semalam atau shalat- shalat sunat yang lain. Kaidah dalam menentukan arah kiblat memerlukan suatu ilmu khusus yang harus dipelajari atau sekurang-kurangnya meyakini arah yang dibenarkan sesuai dengan syariat.

Firman Allah SWT

"Dan dari mana saja kamu keluar (datang), maka palingkanlah wajahmu ke arah Masjidil Haram." (Q.S. Al-Baqarah : 149)

Rasulullah SAW. juga bersabda,

"Jika kamu hendak shalat, maka sempurnakanlah wudhu lalu menghadaplah kearah kiblat dan kemudian bertakbirlah." (H.R. Muslim)

2.4.2 Menentukan Arah Kiblat

Untuk menentukan arah kiblat ada beberapa hal yang harus di ketahui terlebih dahulu, diantaranya adalah sebagai berikut :

2.4.2.1 Menentukan Posisi Kiblat

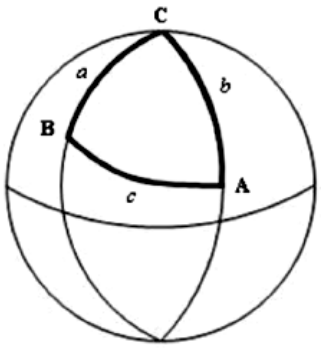
Bola (*sphere)* adalah benda tiga dimensi yang unik dimana jarak antara setiap titik di permukaan bola dengan titik pusatnya selalu sama. Karena bumi sangat mirip dengan bola, maka cara menentukan arah dari satu tempat (misalnya masjid) ke tempat lain (misalnya Ka'bah) dapat dilakukan dengan mengandaikan bumi seperti bola.

Setiap titik di permukaan bumi dapat dinyatakan dalam duat koordinat, yaitu bujur (*longitude*) dan lintang (*latitude*). Semua titik yang memiliki bujur nol terletak pada garis *meridian* Greenwich (setengah lingkaran besar yang menghubungkan kutub utara dan selatan dan melewati Greenwich). Sementara itu semua titik yang memiliki lintang nol terletak pada garis ekuator (khatulistiwa).

Bujur timur terletak di sebelah timur Greenwich, sedangkan bujur barat terletak di sebelah barat Greenwich. Sesuai kesepakatan umum, bujur timur bernilai positif, sedangkan bujur barat bernilai negatif. Sementara itu semua titik yang terletak di sebelah barat Greenwich. Sesuai kesepakatan umum, bujur timur bernilai positif, sedangkan bujur barat bernilai negatif. Sementara itu semua titik yang terletak di sebelah utara ekuator disebut lintang utara, demikian juga untuk titik di selatan ekuator disebut lintang selatan. Lintang utara bernilai positif, sedangkan lintang selatan bernilai negatif.

2.4.2.2 Segitiga Bola

Segitiga bola adalah segitiga di permukaan bola yang sisi-sisinya merupakan bagian dari lingkaran besar. Berbeda dengan segitiga linier atau segitiga biasa yang kita kenal, segitiga bola memiliki tiga sudut dalam satuan derajat busur dan tiga sisi berbentuk garis yang berdimensi panjang seperti meter atau centimeter, sehingga segitiga bola seluruh elemennya hanya dalam satuan derajat busur, karena hanya tiga sudut dan tiga sisi berbentuk busur atau lengkungan bagian dari bola langit atau bola bumi. Lihat gambar berikut :



Gambar 2.1 Segitiga Bola ABC

Sudut segitiga bola ABC adalah A, B dan C kemudian sisi-sisi dihadapan sudut bola masing-masing adalah a, b dan c. Pada segitiga bola terdapat beberapa persyaratan yang diperlukan untuk menguji apakah hasil perhitungan sudah konsisten atau belum juga untuk melihat apakah komponen sudut dan sisi-sisinya sudah merupakan segitiga bola atau bukan, persyaratannya antara lain :

a. Jumlah sudut A, B dan C harus lebih dari 180^o dan kurang dari 540^o($180^o < A+B+C < 540^o$)
b. Jumlah sisi-sisi a, b dan c harus lebih dari 0^o dan kurang dari 360^o($0^o < a+b+c < 360^o$)

c. Jarak sudut (panjang busur) antara sebuah lingkaran besar dan kutubnya adalah 90^o

d. a + b > c; a + c > b dan b + c > a

e. Bila a = b maka A=B, bila a=c maka A = C, bila b = c maka B = C dan sebaliknya.

f. Bila a>b maka A>B; bila a > c maka A > C ; bila b > c maka B > C dan sebaliknya.

2.4.2.3 Rumus Segitiga Bola

Rumus segiga bola diantaranya adalah sebagai berikut :

Rumus cos:

Untuk sudut bola

$$\text{Cos } B = -\cos A \cos C + \sin A \sin C \cos b$$

$$\text{Cos } A = -\cos B \cos C + \sin B \sin C \cos a$$

$$\text{Cos } C = -\cos B \cos A + \sin B \sin A \cos c$$

Untuk sisi – sisi segitiga bola

$$\text{Cos } a = \cos b \cos c + \sin b \sin c \cos A$$

$$\text{Cos } b = \cos a \cos c + \sin a \sin c \cos B$$

$$\text{Cos } c = \cos a \cos b + \sin a \sin b \cos C$$

Rumus sin:

$$\frac{\sin A}{\sin a} = \frac{\sin \mathrm{B}}{\sin b} = \frac{\sin \mathrm{C}}{\sin c}$$

Rumus Analogi Gauss atau De Lambre:

$$\frac{\sin ½(A - B)}{\cos ½C} = \frac{\sin ½(a - b)}{\cos ½c}$$

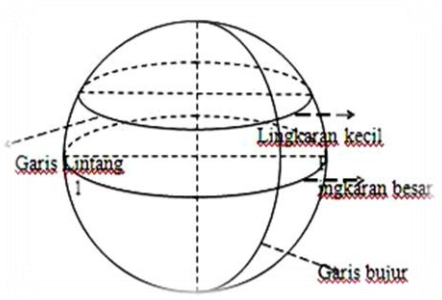
$$\frac{\cos ½(A-B)}{\sin ½C} = \frac{\cos ½(a+b)}{\sin ½c}$$

$$\frac{\sin ½(A+B)}{\cos ½C} = \frac{\sin ½(a-b)}{\cos ½c}$$

$$\frac{\cos ½(A+B)}{\cos ½C} = \frac{\sin ½(a+b)}{\cos ½c}$$

2.4.2.4 Mengenal Tata Koordinat Geografis Bola Bumi

Pada bola langit/bola bumi terdapat tata koordinat geografis antara lain; garis lintang (Φ), garis bujur (λ), lingkaran kecil, lingkaran besar dan lain-lain.



Gambar 2.2 Tata Koordinat Geografis Bola Bumi

Garis Bujur (λ) =0^o adalah meridian standar melewati greenwich di timur Greenwich BT atau Bujur Timur, di barat BB atau Bujur Barat Garis Lintang (Φ)=0^o adalah katulistiwa, Kutub Utara = 90^o Kutub Selatan = -90^o.

Berikut diberikan tabel koordinat geografis tempat di bola bumi :

Tabel 2.2 Tabel koordinat geografis tempat di bola bumi

Lingkaran Dasar	Ekuator Bumi (Khatulistiwa)
Lingkaran Kutub	Bujur (Meridian)
Titik Acuan	Lintang:Khatulistiwa(0^o) Bujur (Meridian):Greenwich(0^o)
Koordinat Pertama	Bujur atau Meridian (λ) Ke arah timur Greenwich atau BT Ke arah barat Greenwich atau BB
Koordinat Kedua	Lintang tempat (Φ) Ke arah selatan = - atau LS atau S Ke arah utara = - atau LU atau U Kutub Utara = 90^o atau 90^o U atau 90^o LU Kutub Selatan = -90^o atau 90^o S atau 90^o LS

Contoh:

1. Jakarta (106^o 49’ BT, 6^o 10’ LS), berarti Jakarta terletak pada garis bujur 106^o 49’ di timur Greenwich dan di garis lintang 6^o 10’ di selatan Khatulistiwa
2. Bandung (107^o5’ BT, -6^o49’ LS), berarti Bandung terletak 107^o5’ di timur Greenwich dan di garis lintang -6^o 49’ di selatan Khatulistiwa.
3. Mekkah (39^o5’ BT, 21^o25’ LU), berarti Mekkah terletak 39^o5’ di timur Greenwich dan di garis lintang 21^o 25’ di Utara Khatulistiwa

Sumber : http://www.mancikarang.sch.id/daftar-artikel/133-menghitung-arah-kiblat-dengan-rumus-segitiga-bola.html/ 10 Mei 2012

2.5 Android

Dalam membangun sebuah aplikasi *operating system* yang berbasis android ada beberapa hal yang perlu dipahami terlebih dahulu sebagai pendukung untuk membangun sebuah aplikasi yang akan dibangun.

2.5.1 Pengertian Android

Menurut Nazruddin Safaat 2011 Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

2.5.2 Sejarah dan Perkembangan Android

Awalnya, Google Inc. membeli android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. kemudian untuk mengembangkan android dibentuklah Open Handset Alience, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standart terbuka pada perangkat selular. Di lain pihak, google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat selular. Di dunia ini terdapat dua jenis distributor system operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Service (GSM) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distributor (OHD).

2.5.3 Komponen Aplikasi

Fitur penting Android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya Android berbeda dari sistem-sistem lain, Android tidak memiliki satu tampilan utama program seperti fungsi main() pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

a. Activities

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity*

ditentukan oleh posisinya dalam tumpukan *acitivity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* Android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

1) **Active,** setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
2) **Paused,** dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused.* Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak

dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, Android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.

3) **Stopped,** ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped. Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive.*

4) **Inactive,** kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah ditiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali. Kondisi transisi ini sepenuhnya ditangani oleh manajer memori Android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

b) Services

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan

pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

c) Intents

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

d) Broadcast Receivers

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Batre sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default* pada perangkat. Sama halnya dengan *service, Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

e) Content Provider

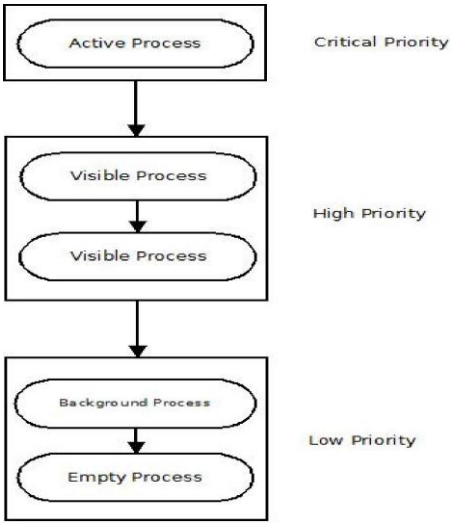
Content Providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database

SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.5.4 Siklus Hidup Aplikasi Android

Siklus hidup aplikasi Android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan perioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya Android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Gambar dibawah menunjukkan prioritas dari aplikasi.



Gambar 2.3 Prioritas Aplikasi Berdasarkan *Activity*

2.5.5 Versi Android

Adapun versi-versi android yang pernah di rilis adalah sebagai berikut :

1. **Android versi 1.1**

   Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. **Android versi 1.5 (Cupcake)**

   Pada pertengah mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengubah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan

terhubung secara otomatis ke headset Bluetooth, animasi layar dan keyboard pada layar yang dapat disesuaikan dengan system.

3. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibandingkan sebelumnya, penggunaan baterai indicator dan konrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapu, kamera, camcorder dan galeri yang diintegrasikan, CDMA/EVDO,802.1x,VPN, Gestures dan Text-to-speech engine, kemampuan dial kontak, teknologi *text to change speech* tidak tersedia pada semua ponsel, pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 (Éclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Éclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP , digital Zoom dan Bluetooth 2.1.

5. Android versi 2.2 (Froyo: frozen Yoghurt)

Pada bulan Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Android inilah yang sekarang sangat banyak beredar di pasaran,

salah satunya adalah di pakai di samsung FX tab yang sudah ada di pasaran.

6. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan webM, efek audio baru (reverb, equalization, headphone virtualization dan bebas boost) dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesai untuk tablet. Honeycomb juga mendukung multi processor dan juga akselerasi perangkat hardware untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan sudah hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus.

8. Android versi 4.0 (ICS : ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan control, terpadu kontak jaringan social, perangkat tambahan fotografi, mencari email secara offline dan berbagi informasi menggunakan NFC.

2.5.6 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Seprti : Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan sebagainya. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasiskan Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royality*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.

3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database* SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.6 GPS (Global Positioning System)

Dalam menentukan sebuah tata letak, koordinat posisi kita harus mengetahui terlebih dahulu yang berhubungan dengan GPS. Adapun yang berkaitan dengan GPS adalah sebagai berikut :

2.6.1 Pengertian GPS

GPS adalah singkatan dari *Global Positioning System*, system ini digunakan untuk menentukan posisi pada permukaan bumi dengan bantuan sinkronisasi sinyal satelit. System ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima yang ada di bumi, dan digunakan untuk menentukan posisi, kecepatan, arah, dan waktu.

Sumber:http://itokwrote.wordpress.com/2007/08/01/teknologi-gps-global positioning-system/ 27 Mei 2012

2.6.2 Cara Kerja GPS

Setiap satelit mentransmisikan dua sinyal yaitu L1 (1575.42 MHz) dan L2 (1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal pseudo-random yaitu kode P (Protected) dan kode C/A (coarse/aquisition). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehingga penerima (perangkat GPS) dapat mengidentifikasi sinyal dari setiap satelit. Pada saat fitur ”Anti-Spoofing” diaktifkan, maka kode P akan dienkripsi dan selanjutnya dikenal sebagai kode P(Y) atau kode Y.

Perangkat GPS yang dikhususkan buat sipil hanya menerima kode C/A pada sinyal L1 (meskipun pada perangkat GPS yang canggih dapat memanfaatkan sinyal L2 untuk memperoleh pengukuran yang lebih teliti.

Perangkat GPS menerima sinyal yang ditransmisikan oleh satelit GPS. Dalam menentukan posisi, kita membutuhkan paling sedikit 3 satelit untuk penentuan posisi 2 dimensi (lintang dan bujur) dan 4 satelit untuk penentuan posisi 3 dimensi (lintang, bujur, dan ketinggian). Semakin banyak satelit yang diperoleh maka akurasi posisi kita akan semakin tinggi. Untuk mendapatkan sinyal tersebut, perangkat GPS harus berada di ruang terbuka. Apabila perangkat GPS kita berada dalam ruangan atau kanopi yang lebat dan daerah kita dikelilingi oleh gedung tinggi maka sinyal yang diperoleh akan semakin berkurang sehingga akan sukar untuk menentukan posisi dengan tepat atau bahkan tidak dapat menentukan posisi.

2.6.3 Kelebihan dan Kekurangan GPS

Teknologi GPS merupakan teknologi yang sangat fenomenal dalam bidang penentuan posisi karena mampu memberikan informasi mengenai posisi secara *real-time* dan kontinu, dimana saja dan kapan saja. Di samping itu, ada beberapa hal yang membuat GPS sangat baik dalam sistem pelacakan :

1. Meliputi wilayah yang luas.
2. Tidak terpengaruh topografis.
3. Solusi Tracking System di Bidang Militer
4. Memberikan ketelitian akurasi yang cukup.
5. Penggunaan tidak dikenakan biaya

Kekurangan GPS yaitu :

1. Penggunaan GPS untuk mengetahui posisi yang mengandalkan setidaknya tiga satelit ini tidak selamanya akurat.
2. Terkadang, dibutuhkan satu satelit untuk memperbaiki sinyal yang diterima. Ketidakakuratan posisi yang ditunjukkan
3. GPS ini dipengaruhi oleh posisi satelit yang berubah dan adanya proses sinyal yang ditunda. Kecepatan sinyal GPS ini juga seringkali berubah karena dipengaruhi oleh kondisi atmosfer yang ada. Selain itu, sinyal GPS juga mudah berinteferensi dengan gelombang elektromagnetik lainnya

2.7 UML

UML (*Unified Modeling Language*) disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka Anda hanya membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain.

UML menurut Yasmi Afrizal, Wahyuni memaparkan bahwa UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan atau di hasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa

model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

2.7.1 Diagram UML

Suatu sistem yang kompleks harus dapat dipandang dari sudut yang berbeda-beda sehingga didapat pemahaman secara menyeluruh. Untuk keperluan tersebut UML menyediakan 9 jenis diagrams yang dapat dikelompokkan berdasarkan sifatnya; Statis atau Dinamis. Kesembilan diagram ini tidak mutlak digunakan atau dibuat sesuai kebutuhan. Pada pemodelan UML dimungkinkan menggunakan diagram lain sejauh diagram tersebut dapat membantu pemahaman mendalam tentang suatu sistem atau perangkat lunak. Ke-9 diagram tersebut adalah sebagai berikut:

1. *Class Diagrams* (Statis)

   Diagram ini memperlihatkan objek-objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instantiasi statis dari segala sesuatu yang dijumpai dari pada diagram kelas.

2. *Object Diagrams* (Statis)

   Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi. Diagram ini umum ditemui pada pemodelan sistem berorientasi objek. Meski sifatnya statis, sering pula memuat kelas-kelas aktif.

3. *Use-Case Diagrams* (Statis)

   Diagram ini memperlihatkan himpunan Use-Case dan Actor-Actor (jenis khusus dari kelas). Diagram ini penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

4. *Sequence Diagrams* (Dinamis)

   Diagram sequence (urutan) adalah diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

5. *Collaboration Diagrams* (Dinamis)

   Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan (*message*).

6. *State Chart Diagram* (Dinamis)

   Diagram ini memperlihatkan state-state pada sistem; memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi, dan terutama penting pada pemodelan sistem-sistem reaktif.

7. *Activity Diagrams* (Dinamis)

   Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagrams* (Statis)

   Diagram ini memperlihatkan organisasi serta kebergantunngan pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka, serta kolaborasi-kolaborasi.

9. *Deployment Diagrams* (Statis)

   Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*), memuat simpul-simpul atau node beserta komponen-komponen yang ada didalamnya. Deployment Diagrams berhubungan erat dengan diagram komponen dimana Deployment Diagrams memuat satu atau lebih komponen-komponen.

Dalam membangun aplikasi, jenis diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram, sequence diagram, activity diagram, class diagram, deployment diagram dan component diagram.*

2.7.2 Kelebihan dan Kekurangan UML

Kelebihan UML dibandingkan dengan bahasa pemodelan yang lain yaitu:

1. Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam pembangunan system
2. Dibandingkan dengan SSAD, waktu pengembangan, level organisasi, ketangguhan dan penggunaan kembali (re-use) kode program lebih tinggi dibandingkan dengan metode OOAD
3. Tidak ada pemisahan antara fase desain dan analisi, sehingga meningkatkan komunikasi antara user dan developer dari awal hingga akhir pembangunan system
4. Analis dan programmer tidak dibatasi dengan batasan implementasi system.
5. Relasi obyek dengan entitas (thing) umumnya dapat di-mapping dengan baik seperti kondisi pada dunia nyata dan keterkaitan dalam system. Hal ini memudahkan dalam memahami desain
6. Memungkinkan adanya perubahan dan kepercayaan diri yang tinggi terhadap kebenaran software yang membantu untuk mengurangi resiko pada pembangunan system yang kompleks
7. Encapsulasi data dan method, memungkinkan penggunaan kembali pada proyek lain, hal ini akan memperingan proses desain, pemrograman dan reduksi harga.

8. OOAD memungkinkan adanya standarisasi obyek yang akan memudahkan memahami desain dan mengurangi resiko pelaksanaan proyek.
9. Dekomposisi obyek, memungkinkan seorang analis untuk memecah masalah menjadi pecahan-pecahan masalah dan bagian-bagian yang di-manage secara terpisah. Kode program dapat dikerjakan bersama-sama

Sedangkan kekurangan UML antara lain:

1. Pada awal desain OOAD, system mungkin akan sangat simple
2. Pada OOAD lebih focus pada coding dibandingkan dengan SSAD.
3. Pada OOAD tidak menekankan pada kinerja team seperti pada SSAD.
4. Pada OOAD tidak mudah untuk mendefinisikan class dan obyek yang dibutuhkan system
5. Seringkali pemrograman berorientasi obyek digunakan untuk melakukan analisis terhadap fungsional site, sementara metode OOAD tidak berbasis pada fungsional system
6. OOAD merupakan jenis manajemen proyek yang tergolong baru, yang berbeda dengan metode analisis dengan metode terstruktur. Konsekuensinya adalah team developer butuh waktu yang lebih lama untuk berpindah ke OOAD karena mereka sudah menggunakan SSAD dalam waktu yang lama
7. Metodologi pengembangan system denga OOAD menggunakan konsep re-use.

2.8 Google API

Google API atau antarmuka pemrograman aplikasi adalah sekumpulan perintah, fungsi, dan protokol dari Google yang dapat digunakan oleh *programmer* saat membangun perangkat lunak untuk sisitem operasi tertentu. Layanan API yang sediakan Google cukup banyak diantaranya :

1. Google Latitude

   Google Latitude merupakan layanan yang memungkinkan pengguna untuk berbagi lokasi tempat dimana pengguna tersebut berada. Walaupun demikian, fitur ini tidak akan menunjukkan lokasi spesifik tempat pengguna berada tapi memberikan informasi bahwa pengguna sedang berada di suatu lokasi pada waktu tertentu. *Google Latitude* dapat diakses melalui beberapa perangkat *mobile*, seperti : Perangkat *mobile* berbasiskan OS Android, seperti Samsung, Sony Ericson, BlackBerry dan sebagainya.

2. Google Maps

   Google maps adalah peta online atau membuka peta secara online. kini dapat dilakukan secara mudah melalui servis gratis dari Google ini. bahkan, servis ini menyediakan API)Application Programming Interface) yang memungkinkan developer lain untuk memanfaatkan aplikasi ini di aplikasi buatannya. Tampilan Google Maps pun dapat dipilih, berdasarkan foto asli atau peta gambar rute saja.

Sumber http://jayaputrasbloq.blogspot.com/2011/06/definisi-atau-pengertian-istilah-google.html / 2 Juni 2012