BAB 2
LANDASAN TEORI

2.1 Sensor

Menurut berbagai sumber, sensor didefinisikan sebagai berikut:

1. Menurut Kenny "Sensor adalah perangkat yang mengubah suatu properti fisik menjadi sebuah sinyal elektronik"[2]
2. Menurut Zook, Bonne & Samad "Sensor adalah perangkat yang merubah rangsangan fisik atau masukan menjadi keluaran yang dapat dibaca"[2]

2.2 Sensor *Accelerometer*

"*Accelerometer*modern tidak lain adalah MEMS *(micro electro mechanical system)* berskala kecil. *Accelerometer* adalah suatu alat untuk mengukur percepatan, mendeteksi dan mengukur getaran, atau untuk mengukur percepatan gravitasi (inklinasi)" [2].

Pengukurannya bisa secara analog maupun digital. *Accelerometer* dapat digunakan untuk mengukur percepatan baik statis maupun dinamis. *Accelerometer* akan mengalami percepatan dalam kisaran dari -1g sampai +1g (9.8m/s^2), dan hingga kemiringan 180°.

Pada *smartphoneAccelerometer* merupakan sensor yang bisa membaca pergerakan sehingga dapat mengubah tampilan layar dari posisi*landscape* ke *portrait* atau sebaliknya dengan cukup memiringkan badan ponsel secara otomatis. Sensor *Accelerometer* terlihat pada Gambar 2.2 :



Gambar 2. 1 *Sensor Accelerometer*

Jenis – jenis *Accelerometer*:

a. Kapasitif

   Teknologi yang digunakan lempengan logam atau mesin mikro yang menghasilkan kapasitansi. Perubahan kapasitansi berkaitan dengan percepatan.

b. Piezoelektrik

   Teknologi yang digunakan kristal piezoelektrik yang sensitif terhadap percepatan (tekanan) dan mengubahnya menjadi muatan listrik.

c. Piezoresistif

   Teknologi yang digunakan lempeng atau mesin mikro yang tahan terhadap perubahan percepatan.

d. *Hall Effect*

   Teknologi yang digunakan gerakan diubah jadi sinyal listrik dengan mengukur perubahan yang terjadi pada medan magnet.

e. *Magnetoresistive*

   Teknologi yang digunakan resistivitas dari material yang berubah ketika berada di medan magnet.

f. *Head transfer*

   Teknologi yang digunakan lokasi dari material yang mengalami perubahan suhu dilacak selama terjadi percepatan dengan mengukur temperatur.

Accelerometer dapat diimplementasikan pada beberapa bidang Salah satu pengguaan *Accelerometer* yang sangat umum yaitu dalam sistem *airbag* yang terdapat pada kendaraan, khususnya mobil. *Accelerometer* ini digunakan untuk mendeteksi penurunan percepatan yang sangat besar yang biasanya terjadi ketika terjadinya tabrakan antar kendaraan.

Sport Watch, berupa jam tangan olahraga yang juga dapat menghitung berapa banyak langkah yang telah lakukan, *Accelerometer* juga digunakan untuk menghitung kecepatan dan jarak dari si pelari yang menggunakannya. *Accelerometer* banyak digunakan untuk menghitung percepatan dan penurunan

percepatan dari sebuah kendaraan. *Accelerometer* membantu untuk mengevaluasi performansi dari mesin dan sistem percepatan dan juga *breaking system* (sistem penurunan percepatan).

Kecepatan yang biasa ditampilkan pada kendaraan anda umumnya didapatkan dari penggunaan *Accelerometer*. Selain itu juga biasa digunakan untuk menghitung vibrasi pada kendaraan, mesin, bangunan, dan sistem keamanan pada kendaraan (*safety installation*). *Accelerometer* juga dapat mengkalkulasi percepatan yang diakibatkan oleh gravitasi bumi. *Accelerometer* yang menghitung gravitasi secara spesifik digunakan pada *gravimetry*, disebut sebagai *gravimeter*. *Notebook* atau *laptop* juga dilengkapi dengan *Accelerometer* untuk mengevaluasi goncangan yang dirasakan oleh *laptop* tersebut.

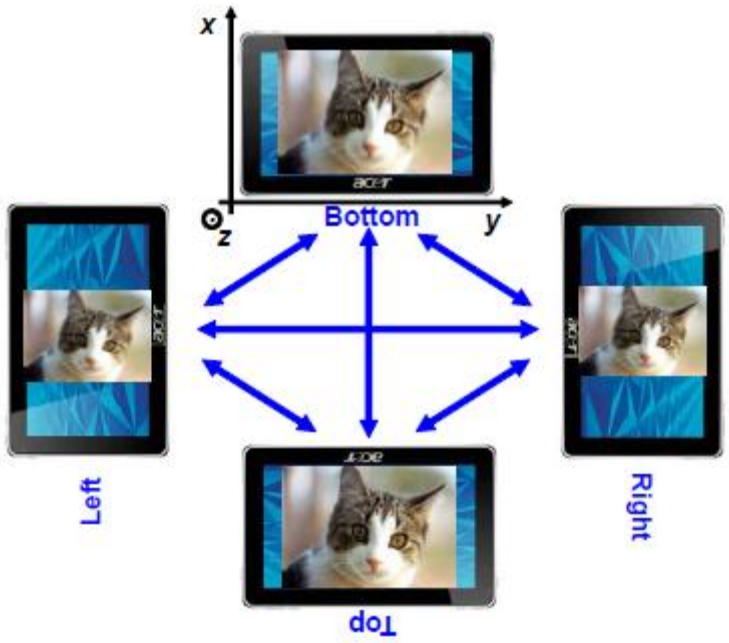
Accelerometer pada *laptop* biasanya digunakan pada sistem *Sudden Motion Sensor*, yang biasa digunakan untuk mendeteksi jatuhnya *laptop*. Jika kondisi pada saat jatuh terdeteksi, *hard diskdrive* yang ada akan diproteksi sehingga tidak terjadi *data loss*. Sekarang ini juga terdapat *notebook* yang menggunakan *Accelerometer* untuk secara otomatis mengubah arah layar (menjadi miring ataupun terbalik) sesuai dengan arah monitor tersebut ditegakkan (*portrait* atau *landscape*).

Terdapat juga sejumlah ponsel yang menggunakan *Accelerometer* untuk mengubah lagu yang dimainkan (*Track Switching*). *Camera recorder* menggunakan *Accelerometer* untuk menstabilkan gambar. Kamera digital menggunakan *Accelerometer* untuk menu pilihan anti blur ketika mengambil gambar. Baru-baru ini Apple.Inc memperkenalkan sebuah gebrakan dengan mengkombinasikan 2 sensor gerakan yaitu antara *Accelerometer* dan *Gyroscope* pada sebuah perangkat ponsel. Ini akan menyempurnakan fitur dari ponsel yang hanya menggunakan *Accelerometer* dalam mendeteksi gerakan. Dengan kombinasi ini maka akan didapatkan 6 sumbu gerakan yaitu 3 sumbu linier (atas-bawah, kanan-kiri, depan-belakang) dan 3 sumbu rotasi (rotasi *roll,pitch and yaw* seperti pada gambar 2). 1 keunggulan lagi dari kombinasi ini adalah akan didapatkan *output* gambar yang tiap detil

gerakannya lebih halus dari pada perangkat ponsel yang hanya menggunakan *Accelerometer*.[2]

Memilih Portrait dan Landscape Mode

Penggunaan volume tinggi pertama accelerometer dalam produk konsumen adalah untuk beralih tampilan antara orientasi landscape dan portrait. Sebuah tablet PC, yang ditunjukkan pada Gambar 2.2, memiliki empat kemungkinan orientasi layar berlabel di sini sebagai Bawah, Atas, Kanan dan Kiri, menurut lokasi tepi bawah teks atau citra yang ditampilkan.



Gambar 2. 2*Portrait dan Landscape Layar Orientasi*

Masalah ini agak lebih kompleks dari itu akan muncul pada pandangan pertama sejak pemetaan langsung antara orientasi layar dan membaca accelerometer tidak ergonomis cocok. Secara khusus, pengguna melihat layar dalam salah satu dari empat mode yang ditunjukkan pada Gambar 8 mengharapkan bahwa mode tampilan tetap tidak berubah sebagai tablet diputar ke orientasi datar. Pembacaan accelerometer yang dihasilkan dari tablet yang datar di atas meja sehingga dapat dikaitkan dengan salah satu dari empat orientasi layar.

Solusinya adalah mesin negara sederhana di mana pembacaan accelerometer mengarah ke transisi antara orientasi layar daripada langsung mendefinisikan orientasi layar. Sederhana aturan transisi keadaan tercantum di bawah ini.

1. (|G_{pz}|< 0.5g)dan(|G_{px}|< 0.5g)dan(|G_{py}|< 0.4g) : orientasikeatas
2. (|G_{pz}|< 0.5g)dan(|G_{px}|<-0.5g)dan(|G_{py}|< 0.4g) : orientasikebawah
3. (|G_{pz}|< 0.5g)dan(|G_{py}|< 0.5g)dan(|G_{px}|< 0.4g) : orientasikekanan
4. (|G_{pz}|< 0.5g)dan(|G_{py}|<-0.5g)dan(|G_{px}|< 0.4g) : orientasikekiri

Pada *smartphone* berbasis android terdapat sensor manager yang berfungsi untuk mengaktifkan sensor *accelerometer* dalam mencari nilai koordinat x,y,z dengan memiringkan *handphone*

Tabel 2. 1 Sensor Manager

```
sensorManager=(SensorManager)getSystemService(SENSOR_SERVICE);

        sensor=sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
                sensorManager.registerListener(new SensorEventListener() {

                        @Override
                        publicvoid onSensorChanged(SensorEvent event) {
                                currentX=event.values[0];
                                currentY=event.values[1];
                                currentZ=event.values[2];
                                Log.d("X,Y,Z", String.valueOf(round(currentX))+",
"+round(currentY)+", "+round(currentZ));
                        }
```

Jika sensor *accelerometer* telah diaktifkan maka dapat mencari nilai koordinat kemiringan dimana *values* [0] untuk nilai *currentX* dan *values* [1] untuk nilai *currentY* dan *values* [2] untuk nilai *currentZ*

2.3 *Handphone*

Telepon seluler (ponsel) atau telepon genggam (telgam) atau *handphone*(HP) atau disebut pula adalah perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, namun dapat dibawa ke mana-mana (portable, *mobile*) dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel (nirkabel;

wireless). Saat ini Indonesia mempunyai dua jaringan telepon nirkabel yaitu sistem GSM (*Global System for Mobile Telecommunications*) dan sistem CDMA (*Code Division Multiple Access*). Badan yang mengatur telekomunikasi seluler Indonesia adalah Asosiasi Telekomunikasi Seluler Indonesia (ATSI)

2.4 Java

Bahasa Java dikembangkanoleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang-barang elektronik seperti televise, *VCR*, *toaster* dan mesin-mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program – program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program – program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.[3]

Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*
3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas

2.4.1 Kelebihan dan Kekurangan Java

Setelah membahas mengenai pengertian java, kemudian akan membahas mengenai kelebihan dan kekurangan java.

2.4.1.1 Kelebihan Java

Kelebihan Java yang pertama tentu saja multi *platform*. Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda. Hal ini sesuai dengan slogannya yang udah dibahas sebelumnya.

Yang kedua adalah OOP *(Object Oriented Programming)*. Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah. Kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas Java. Setiap hal pasti memiliki kelebihan dan kekurangan.

2.4.1.2 Kekurangan Java

Kekurangan Java yang adalah penggunaan memori yang cukup banyak, lebih besar daripada bahasa tingkat tinggi sebelum generasi Java. Namun hal ini memang sesuai dengan fitur beragam yang dimiliki oleh Java. Masalah memori ini juga tidak dialami oleh semua pengguna aplikasi Java. yang sudah menggunakan perangkat keras dengan teknologi terbaru tidak merasakan kelambatan dan konsumsi memori Java yang tinggi. Lain halnya dengan yang menggunakan teknologi lama atau komputer yang sudah berumur tua lebih. Namun apapun kelemahan yang dimiliki Java, faktanya adalah Java merupakan bahasa pemrograman yang populer dan digunakan di seluruh dunia saat ini.

2.5 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan untuk digunakan oleh bermacam peranti bergerak (*mobile device*). Hal ini memungkinkan para pengembang menulis kode terkelola (*managed code*) dalam bahasa pemrograman Java, mengontrol peranti via perpustakaan Java yang dikembangkan Google. Android merupakan *subset* perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, middleware, dan aplikasi inti yang di release oleh Google. Sedangkan Android SDK *(Software Development Kit)* menyediakan Tools dan API yang diperlukan untuk Aplikasi Android nantinya tidak akan berjalan langsung di atas *kernel* sistem operasi namun berjalan di atas Dalvik,

sebuah *virtual machine* yang khusus dirancang untuk digunakan pada sistem embedded mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemograman Java.[4]

Android memiliki berbagai fitur seperti :

a. *Framework* Aplikasi yang mendukung penggantian komponen dan *reusable*

b. Mesin virtual Dalvik dioptimalkan untuk perangkat *mobile*

c. *Integrated browser* berdasarkan *engine open source WebKit*

d. Grafts yang dioptimalkan dan didukung oleh *library* grafis 2D, grafis 3D berdasarkan spesifikasi OpenGL

e. *SQLite* untuk penyimpanan data

f. Media support yang mendukung audio, video dan berbagai format gambar (MPEG4,MP3, AAC, AMR, JPG,PNG,GIF)

g. GSM, *Bluetooth,* EDGE, 3G dan *WiFi* (*hardware dependent*)

h. Kamera, GPS, Kompas dan *Accelerometer*

i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging,* serta plugin untuk Eclipse IDE.

Selain itu menggunakan Android sebagai Operasi Sistemnya memiliki biaya lisensi lebih murah dan sifatnya yang semi *open source*. Tidak hanya itu, Android tentunya akan *support* dengan berbagai layanan dari Google.

2.5.1 Karakteristik Android

Andorid merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, *middleware,* dan aplikasi init yuang di-release oleh Google. Pada tulisan sebelumnya, kita mengenal SDK (*Software Development Kit*). SDK adalah suatu tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform atau linkungan Android. Pengembangan aplikasi Android menggunaka bahasa pemrograman Java. Seperti kita ketahui, SDK ini dikembangkan oleh OHA (*Open Handsate Alliance*). Organisasi OHA ini terdiri atas Googl, Intel, Motorola, Qualcomm,

T-Mobile, dan NVIDIA. SDK dapat kita gunakan pada beberapa IDE (*Integrated Development and Environment*) – software untuk membuat suatu program – akan tetapi, pada tulisan selanjutnya, saya akan menggunakan Eclipse karena OHA secara resmi membuat plugin untuk IDE Eclipse.

2.5.2 Arsitektur Android

Arsitektur Android menunjukkan komponen-komponen utama yang terdapat pada sistem operasi Android. Berikut ini arsitektur Android:



Gambar 2.3 Arsitektur Android

1. **Linux Kernel**

   Android bukan Linux, akan tetapi Android dibangun diatas Linux *Kernel*, yaitu versi 2.6 sehingga kehandalannya bisa dipercaya. Untuk inti sistem *service* Linux yang digunakan seperti keamanan, manajemen memori, proses manajemen, *network*, dan *driver* model. Seperti yang terlihat digambar (Gambar II.3), Linux *Kernel* menyediakan *Driver* Layar, Kamera, *Keypad*, WiFi, *Flash Memory*, *Audio*, dan IPC (*Interprocess Communication*) untuk

mengatur aplikasi dan keamanan. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software stack*-nya.

2. **Libraries**

   Android menyertakan *libraries* C/C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *Developer* aplikasi melalui *Framework* Aplikasi Android. Beberapa inti *libraries* tercantum dibawah ini:

   a. *System C Library* – Variasi dari implementasi BSD-berasal pelaksana sistem standar C *library* (*libc*), sesuai untuk perangkat *embedded* berbasis Linux.
   b. *Media Libraries* – *PacketVideo* berdasarkan OpenCORE; *library* medukung pemutaran rekaman dan populer banyak format audio dan video, serta *file* gambar, termasuk MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
   c. *Surface Manager* – mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.
   d. LibWebCore – mesin *web modern* yang *powerfull* yang baik *browser* Android *embedded web*.
   e. 3D *Libraries* – sebuah pelaksana berdasarkan openGL ES 1,0 API; perpustakaan baik menggunakan perangkat keras akselerasi 3D (apabila tersedia) atau yang disertakan, sangat optimal 3D *software rasterizer*.
   f. FreeType – bitmap dan vektor *font rendering*.
   g. SQLite – mesin *database* yang kuat dan ringan, dan penghubung tersedia untuk semua aplikasi

3. **Android Runtime**

   Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat pada *core libraries* bahasa pemrograman Java. Setiap

aplikasi menjalankan prosesnya sendiri dalam Android, dengan masing-masing instan dari mesin virtual Dalvik (*Dalvik VM*). Dalvik dirancang agar perangkat dapat menjalankan multiple VMs secara efisien. Mesin Virtual Dalvik mengeksekusi *file* dalam Dalvik *executable* (.dex), sebuah format yang dioptimalkan untuk memori yang kecil.

4. *Framework* Aplikasi

Pengembang memiliki akses penuh menuju API *framework* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah. Setiap aplikasi dapat memanfaatkan kemampuan ini (sesuai dengan batasan keamanan yang didefinisikan oleh *framework*). Mekanisme yang sama memungkinkan komponen untuk diganti oleh pengguna.

Semua aplikasi merupakan rangkaian set layanan dan sistem, termasuk:

a. *Views* yang kaya dan extensible yang dapat digunakan untuk membangun aplikasi, termasuk list, grids, kotak teks, tombol, dan bahkan sebuah *embedded web*.
b. *Content Provider* yang memungkinkan aplikasi untuk mengakses data (seperti dari daftar kontak telepon) atau membagi data tersebut.
c. *Resource Manager*, yang menyediakan akses ke kode nonsumber lokal seperti string, gambar, dan tata letak *file*.
d. *Notifications Manager* yang memungkinkan semua aplikasi menampilkan *alert* yang bisa dikostumisasi di dalam status bar.
e. *Activity Manager* yang mengelola siklus hidup aplikasi dan menyediakan navigasi umum *backstack*.

5. *Applications*

Android telah menyertakan aplikasi inti seperti *email client*, SMS, kalender, peta, *browser*, kontak, dan lainnya. Semua aplikasi tersebut ditulis dengan menggunakan bahasa pemrograman Java. Pada *layer* inilah *developer* atau kita menempatkan aplikasi yang dibuat. Yang istimewa adalah pada Android semua aplikasi baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan pada *layer* aplikasi dengan menggunakan *library* API yang sama. Ini berarti semua aplikasi yang dibuat untuk Android akan memiliki akses yang setara dalam mengakses seluruh kemampuan *handset*, tanpa membedakan apakah itu merupakan aplikasi inti atau aplikasi pihak ketiga. Dalam kata lain dengan *platform* Android ini, *Programmer* atau *Developer* secara penuh akan bisa mengkostumisasi perangkat Androidnya.[4]

2.5.3 Perkembangan Android

Android telah banyak melakukan perkembangan, berikut adalah perkembangan dari android:

a. 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.

b. Android versi 1.5 (*Cupcake*)

Pada pertengahan 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, mengunggah 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 sistem.

c. Android versi 1.6 (*Donu*t)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang dintegrasikan; 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.

d. Android versi 2.0/2.1 (*Eclair*)

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

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

e. Android versi 2.2 (*Froyo: Frozen Yoghurt*)

Pada 20 Mei 2010, Android versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 *JavaScript*

engine yang dipakai *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi *Android Market.*

f. 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 *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

g. 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 didesain untuk tablet. *Honeycomb* juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (*hardware*) untuk *grafis*. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah *Motorola Xoom*. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama *Eee Pad Transformer* produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

h. 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 kontrol, terpadu kontak jaringan sosial, perangkat tambahan *fotografi*, mencari *email*

secara *offline*, dan berbagi informasi dengan menggunakan NFC.[4]

2.5.4 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows *Mobile*, BlackBerry, Java *Mobile* Edition, Linux *Mobile* (LiM), dan banyak lagi. 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 system 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 system menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.

5. Dukungan grafis dan suarat 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.

2.5.5 Kekurangan Android

Tak bisa dipungkiri disamping kelebihan tentu saja ada kekurangan dari sistem operasi ini.

1. Koneksi Internet yang terus menerus – Yups, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif.Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan.
2. Iklan – Aplikasi di **Ponsel Android** memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

2.6 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform (platform-independent)*. Berikut ini adalah sifat dari Eclipse:[6]

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. *Mulit-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.
4. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.[3]

2.6.1 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah *kernel*, yang mengangkat *plug-in*. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari *plug-in* yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan *Rich Client Platform* (RCP). Berikut ini adalah komponen yang membentuk RCP:

a. *Core platform*
b. OSGi
c. SWT (*Standard Widget Toolkit*)
d. JFace
e. *Eclipse Workbench*

Secara standar Eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (*Plug-in Development Environment*) untuk mengembangkan *plug-in* baru. Eclipse beserta *plug-in*-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik[4]. Jadi, Eclipse tidak saja untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal *plug-in* yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat *plug-in* CDT (*C/C++ Development Tools*).

Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, *plug-in* UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat *plug-in* sesuai dengan keinginannya.

2.6.2 Android SDK

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di *release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman Java. Sebagai *platform* aplikasi-netral, android member anda kesempatan unutk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Hadphone/*Smartphone*. Beberapa fitur-fitur android yang paling penting adalah:[4]

a. *Framework* : aplikasi yang mendukung pengganti komponen dan reusable.
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*
c. Integrated *Browser* verdasarkan engine *open source* WebKit.
d. Grafis yang dioptimalkan dan didukung oleh *libraries* grafis 2D, grafis 3Dberdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi *hardware*)
e. SQLite untuk penyimpanan data.

f. Media *Support* yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung *Hardware*)
g. *Bluetooth*, EDGE, 3G, dan WiFi (tergantung *hardware*)
h. Kamera, GPS, Kompas, dan *Accelerometer* (tergantung *hardware*)
i. Lingkungan *Development* yang lengkap dan termasuk pernagkat *emulator*, *tools* untuk *debugging*, profil dan kinerja memori, dan *plugin* untuk IDE Eclipse.

2.6.3 Android Development Tools (ADT)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse *Intergrated Development Environment* (IDE) yang dirancang untuk memberikan lingkungan yang terpadu di mana untuk membangun aplikasi Android. ADT memperluas kemampuan Eclipse untuk membiarkan para *developer* lebih cepat dalam membuat proyek baru Android, membuat aplikasi UI, menambahkan komponen berdasarkan Android *Framework* API, *debug* aplikasi dalam pengunaan Android SDK, dan membuat file APK untuk mendistribusikan aplikasi[4]

2.7 *Tools-Tools* yang digunakan

2.7.1 Konsep Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan mendaji lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antara objek tersebut untuk kemudian dimodelkan dalam sistem nyata.

Suatu perusahaan software yaitu Rational Software, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified*

Modelling Language (UML) sebagai bahasa standar dalam *Objak Oriented Analysist Design* (OOAD).[5]

2.7.1.1 Unified Modelling Language

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan *design* ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, erincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur pemodelan entity-relationship.

2.7.1.2 Use Case Diagram

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case* adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

2.7.1.3 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* yang diletakkan diantara obyek-obyek ini di dalam *use case*. *Sequence* diagram menambahkan dimensi waktu pada interaksi diantara obyek. Pada diagram ini *participant* diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. *Life line participant* diurutkan dari setiap *participant*. Kotak kecil pada *lifelin*e menyatakan *activation,* yaitu menjalankan salah satu operation dari *participant*. State bisa ditambahkan dengan menambahkannya sepanjang life line.

2.7.1.4 Activity Diagram

Activity diagram seperti sebuah *flow chart*. *Activity* diagram menunjukkan tahapan, pengambilan keputusan dan percabangan. Diagram ini sangat berguna untuk menunjukkan *operation* sebuah obyek dan proses bisnis. Kelebihan *activity* diagram dibandingkan *flowchart* adalah kemampuannya dalam menampilkan aktivitas parallel.

2.7.1.5 Class Diagram

Kotak adalah notasi UML untuk class. Nama, *attribute, operation* dan *responsibility* dari *class* ada pada kotak tersebut. *Stereotype* bisa dipergunakan untuk mengorganisasikan daftar *attribute* dan *operation.* Dalam beberapa kasus, kadangkala hanya perlu ditampilkan sebagain saja dari *attribute* dan *operation.*

Tipe *attribute* dan nilai default bisa dimunculkan sebagaimana pada *operation.* Untuk mengurangi ambiguitas pada pendeskripsian *class, constraint* bisa ditambahkan. Bahkan kalau perlu bisa ditambahkan *attribute* notes ke dalam kotak tersebut.[5]

2.8 Pengenalan Teorema Trigonometri

Trigonometri (dari bahasa yunani *trigonon* = tiga sudut dan *metro* = mengukur) adalah sebuah cabang matematika yang berhadapan dengan sudut

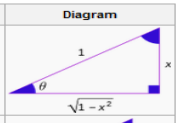
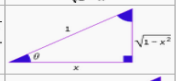
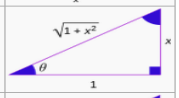
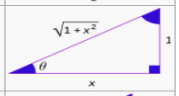
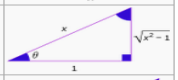
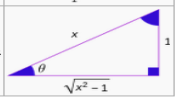
segitiga dan fungsi trigonometrik seperti sinus, cosinus, dan tange. Trigonometri memiliki hubungan dengan geometri, meskipun ada ketidaksetujuan tentang apa hubungannya. bagi beberapa orang, trigonometri adalah bagian dari geometri.

Matematikawan Yunani Hipparchus sekitar 150 SM menyusun tabel trigonometri untuk menyelesaikan segitiga. Matematikawan Yunani lainnya, Ptolemy sekitar tahun 100 mengembangkan penghitungan trigonometri lebih lanjut. Matematikawan SilesiaBartholemaeus Pitiskus menerbitkan sebuah karya yang berpengaruh tentang trigonometri pada 1595 dan memperkenalkan kata ini ke dalam bahasa Inggris dan Perancis.

Ada banyak aplikasi trigonometri. Terutama adalah teknik triangulasi yang digunakan dalam astronomi untuk menghitung jarak ke bintang-bintang terdekat, dalam geografi untuk menghitung antara titik tertentu, dan dalam sistem navigasi satelit. Bidang lainnya yang menggunakan trigonometri termasuk astronomi (dan termasuk navigasi, di laut, udara, dan angkasa), teori musik, akustik, optik, analisis pasar finansial, elektronik, teori probabilitas, statistika, biologi, pencitraan medis/*medical imaging* (*CAT scan* dan *ultrasound*), farmasi, kimia, teori angka (dan termasuk kriptologi), seismologi, meteorologi, oseanografi, berbagai cabang dalam ilmu fisika, survei darat dan geodesi, arsitektur, fonetika, ekonomi, teknik listrik, teknik mekanik, teknik sipil, grafik komputer, kartografi, kristalografi.

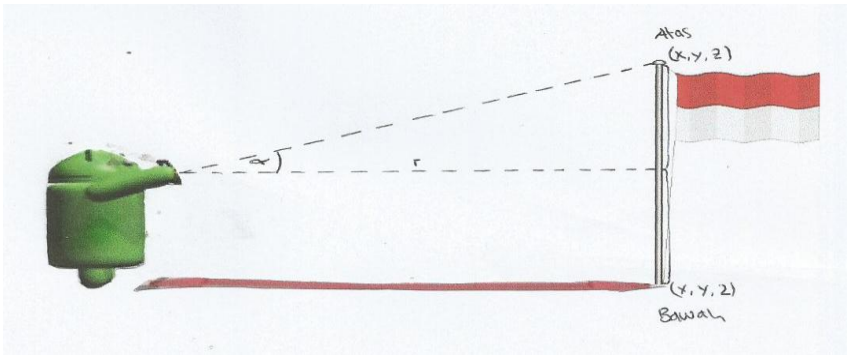
Trigonometri terdiri dari sinus (sin), cosinus (cos), tangens (tan), cotangens (cot), secan (sec) dan cosecan (cosec). Trigonometri merupakan nilai perbandingan yang didefinisikan pada koordinat kartesius atau segitiga siku-siku.[6]

Tabel 2.2 Sudut-sudut Trigonometri

θ	$\sin\theta$	$\cos\theta$	$\tan\theta$	Diagram
$\arcsin x$	$\sin(\arcsin x) = x$	$\cos(\arcsin x) = \sqrt{1-x^2}$	$\tan(\arcsin x) = \frac{x}{\sqrt{1-x^2}}$	
$\arccos x$	$\sin(\arccos x) = \sqrt{1-x^2}$	$\cos(\arccos x) = x$	$\tan(\arccos x) = \frac{\sqrt{1-x^2}}{x}$	
$\arctan x$	$\sin(\arctan x) = \frac{x}{\sqrt{1+x^2}}$	$\cos(\arctan x) = \frac{1}{\sqrt{1+x^2}}$	$\tan(\arctan x) = x$	
$\operatorname{arccot} x$	$\sin(\operatorname{arccot} x) = \frac{1}{\sqrt{1+x^2}}$	$\cos(\operatorname{arccot} x) = \frac{x}{\sqrt{1+x^2}}$	$\tan(\operatorname{arccot} x) = \frac{1}{x}$	
$\operatorname{arcsec} x$	$\sin(\operatorname{arcsec} x) = \frac{\sqrt{x^2-1}}{x}$	$\cos(\operatorname{arcsec} x) = \frac{1}{x}$	$\tan(\operatorname{arcsec} x) = \sqrt{x^2-1}$	
$\operatorname{arccsc} x$	$\sin(\operatorname{arccsc} x) = \frac{1}{x}$	$\cos(\operatorname{arccsc} x) = \frac{\sqrt{x^2-1}}{x}$	$\tan(\operatorname{arccsc} x) = \frac{1}{\sqrt{x^2-1}}$	

1. Gambar 2.4 memperlihatkan simulasi pada saat pengguna melakukan pengambilan sudut atas pada objek, pada saat pengambilan sudut, posel tidak boleh bergeser agar nilai yang diambil akurat. Berikut rumus mencari alpha setalah mendapatkan nilai koordinat kemiringan ponsel pada titik atas objek.

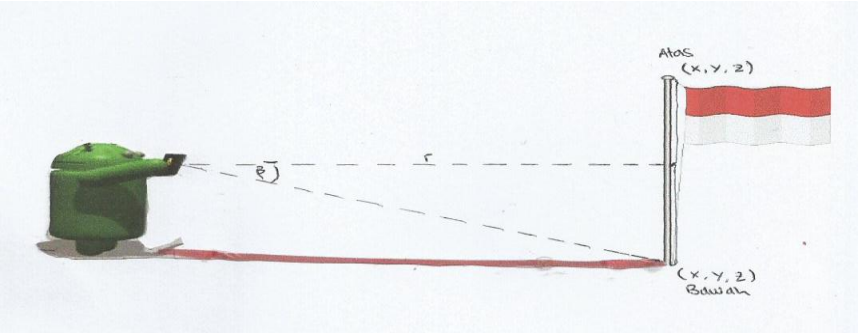
$$\alpha = \text{acos}\left(\frac{Vatas_x}{\sqrt{Vatas_x^2 + Vatas_y^2 + Vatas_z^2}}\right) \qquad (2.1)$$



Gambar 2.4 Simulasi Pengambilan Sudut Atas Objek

2. Gambar 2.5 memperlihatkan simulasi pada saat pengguna melakukan pengambilan sudut bawah pada objek, pada saat pengambilan sudut, posel tidak boleh bergerak agar nilai yang diambil akurat. Berikut rumus mencari beta setalah mendapatkan nilai koordinat kemiringan ponsel pada titik bawah objek.

$$\beta = \text{acos}\left(\frac{Vbawa\ h_x}{\sqrt{Vbawa\ h_x^2 + Vbawa\ h_y^2 + Vbawa\ h_z^2}}\right) \qquad (2.2)$$



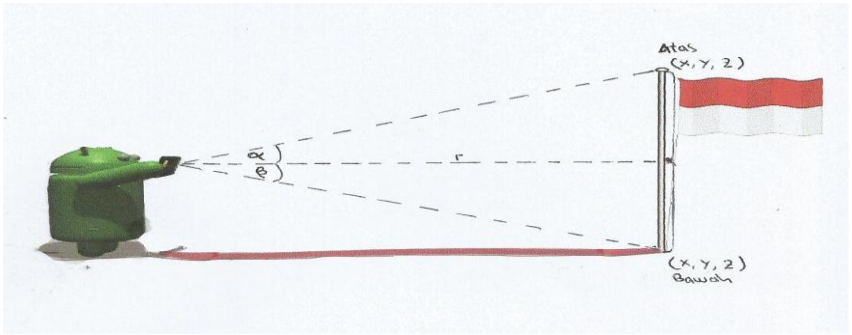
Gambar 2.5 Simulasi Pengambilan Sudut Bawah Objek

3. Setelah mendapatkan nilai alpha dan beta, maka dapat dicari tinggi atas dan tinggi bawah dengan memasukan jarak objek secara manual (r). Berikut rumus untuk mencari

   Tinggi atas = tan $(\alpha) * r$

   Tinggi bawah = tan $(\beta) * r$

4. Gambar 2.6 Simulasi setelah mendapatkan tinggi atas dan tinggi bawah. Berikut rumus yang diterapkan untuk mancari tinggi objek. (Tinggi objek = tinggi atas + tinggi bawah)



Gambar 2.6 Simulasi Untuk Mencari Tinggi Objek

2.9 Pengukuran Kesalahan Peramalan MAD

Sebuah notasi matematika dikembangkan untuk menunjukkan periode waktuyang lebih spesifik karena metode kuantitatif peramalan sering kali memperlihatkandata runtun waktu. Huruf *Y* akan digunakan untuk menotasikan sebuah variabel runtun waktu meskipun ada lebih dari satu variabel yang ditunjukkan. Periode waktubergabung dengan observasi yang

ditunjukkan sebagai tanda. Oleh karena itu, $Y_t$menunjukkan nilai dari runtun waktu pada periode waktu *t.*

Notasi matematika juga harus dikembangkan untuk membedakan antarasebuah nilai nyata dari runtun waktu dan nilai ramalan. *A* akan diletakkan di atassebuah nilai untuk mengindikasi bahwa hal tersebut sedang diramal. Nilai ramalanuntuk Y_t adalah $\hat{Y}_t$. Ketepatan dari teknik peramalan sering kali dinilai denganmembandingkan deret asli Y_1 , Y_2 , … dengan deret nilai ramalan $\hat{Y}_1, \hat{Y}_2, \ldots$[7]

NOTASI DASAR PERAMALAN

$$et = Yt - \hat{Y}t \qquad (2.3)$$

Notasi peramalan dapat diringkas sebagai berikut:

Y_t : nilai data time series pada periode *t*

$\hat{Y}_t$: nilai ramalan dari $\hat{Y}_t$

$e_t = Y_t - \hat{Y}_t$: sisa atau kesalahan ramalan.

Beberapa metode lebih ditentukan untuk meringkas kesalahan (error) yangdihasilkan oleh fakta (keterangan) pada teknik peramalan. Sebagian besar daripengukuran ini melibatkan rata-rata beberapa fungsi dari perbedaan antara nilai aktualdan nilai peramalannya. Perbedaan antara nilai observasi dan nilai ramalan ini seringdimaksud sebagai residual.

Persamaan di bawah ini digunakan untuk menghitung error atau sisa untuktiap periode peramalan.

Satu metode untuk mengevaluasi metode pengukuran tinggi objek menggunakan jumlah kesalahan-kesalahan yang absolute. *The Mean Absolut Deviation* (MAD) mengukur ketepatan tinggi objek dengan merata- rata kesalahan dugaan (nilai absolute masing-masing kesalahan). MAD paling berguna ketika orang yang menganalisa ingin mengukur kesalahan dalam mengukur tinggi objek dalam unit yang sama sebagai deret asli.[7]

$$\boldsymbol{MAD} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} |Y_t - \hat{Y}_t| \qquad (2.4)$$

2.10 Perancangan Kuesioner

Perancangan kuesioner dilakukan sebelum melakukan pengumpulan data dengan tujuan untuk mendapatkan kuesioner yang baik. Kuesioner yang digunakan dalam penelitian ini adalah kuesioner yang mempunyai bentuk terstruktur langsung, dimana alternative jawaban responden telah disusun sedemikian rupa sehingga responden dan memberikan sekumpulan alternatif yang sifatnya *mutually exclusive* yaitu hanya satu alternatif yang dapat dipilih dan *mutually exhhastive* yaitu kumpulan alternatif yang diberikan sudah mencakup semua kemungkinan sesuai dengan responden pada pertanyaan yang diajukan. Kuesioner yang dpergunakan pada penelitian ini terdiri dari dua jenis skor pengukuran yaitu : [8]

1. Skor kuesioner untuk data umum

Pengukuran data umum dalam penelitian ini menggunakan skala norminal, dimana angka yang diberikan kepada objek mempunyai arti sebagai label saja, dan tidak menunjukkan tingkatan maupun urutan antara kategori-katagori dalam ukuran itu.

Tabel 2. 3 Skor Kuesioner

Ketegori Jawaban	Bobot
Sangat Setuju	5
Setuju	4
Kurang Setuju	3
Tidak Setuju	2
Sangat Tidak Setuju	1

2. Perhitungankuesioner

Data dianalisis dengan menghitung rata-rata jawaban berdasarkan scoring setiap jawaban dari responden kemudian dijumlahkan lalu digambarkan secara kontinum, perhitungan dapat dilakukan secara manual menggunakan penskoran skala likert yaitu :

Jumlah skor = Jumlah responden x Skor bobot (2.5)

Hasil = Jumlah skor \ Sample responden (2.6)