

## BAB II

### LANDASAN TEORI

#### 2.1 Menghitung Kecepatan Menggunakan Kinect

Untuk menghitung kecepatan menggunakan Kinect dibutuhkan sebuah masukan berupa posisi objek dan waktu yang ditempuh objek. (Catuhe, 2012 dan Microsoft, 2012). Nilai masukan dapat diperoleh dari *Runtime Tracking* atau *Depth Tracking*. Cara mendapatkan masukan tersebut dengan membuat sebuah *threshold*. Titik U sebagai posisi dimana *user* berdiri dan titik K adalah posisi Kinect, sedangkan titik U1 sebagai marker awal pukulan dan U2 sebagai marker akhir dari pukulan. Jarak antara titik U – K adalah 1,5 meter sedangkan jarak titik U1 – U2 adalah 50 cm.

Jarak optimal dan hasil tangkapan terbaik pada Kinect adalah sejauh 1,2 meter sampai 3 meter (Catuhe, 2012), oleh karena itu akan digunakan setengah dari jarak optimal tersebut agar Kinect berfungsi dengan normal sehingga mendapatkan hasil terbaik dan optimal kecepatan pukulan dari tangan *user*.



Gambar 2.1 Jarak *threshold*

#### 2.2 Energi Kinetik

Setiap sistem fisik menyimpan sebuah energi, untuk mengetahui berapa besarnya energi yang tersimpan dalam sebuah benda maka diperlukan persamaan khusus untuk menghitungnya. Secara umum adanya energi dalam sebuah benda dapat diketahui dengan mengamati perubahan sifat dari benda atau sistem

tersebut. Dikarenakan bentuk energi yang terlalu banyak maka secara umum energi dapat didefinisikan sebagai kapasitas sebuah benda untuk melakukan usaha.

Energi Kinetik merupakan energi yang dimiliki benda karena gerakannya. Dapat juga didefinisikan sebagai usaha yang dibutuhkan untuk menggerakkan suatu benda dengan massa tertentu dari posisi diam sampai pada kecepatan tertentu.

Dalam fisika, energi kinetik biasa dituliskan dengan persamaan berikut

$$E_k = \frac{1}{2}mv^2 \dots\dots\dots(1)$$

Dimana Energi Kinetik ( $E_k$ ) adalah setengah dari massa benda dikalikan dengan kuadrat kecepatan benda tersebut. Willem's Gravesande telah melakukan percobaan untuk membuktikan rumus tersebut dengan cara menjatuhkan benda kedalam tanah liat dari hasil percobaan tersebut dia menyimpulkan bahwa kedalaman dari lubang tanah liat yang dihasilkan benda yang jatuh tersebut berbanding lurus dengan kuadrat kecepatan.

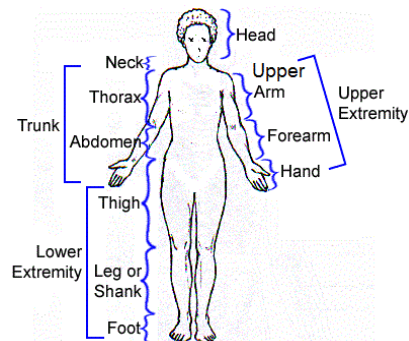
Energi Kinetik memiliki satuan internasional yaitu Joule dan dilambangkan dengan kapital J, atau bisa juga ditulis Newton meter (N.m) (Zainuri, 2007)

Karena setiap benda yang bergerak memiliki energi kinetik, maka nilai dari energi kinetik tersebut yang akan digunakan untuk merepresentasikan kekuatan sebuah pukulan.

### 2.3 Bagian Tubuh Manusia

Tubuh manusia terdiri dari beberapa bagian besar, dan setiap bagian tersebut memiliki massanya masing-masing. Menurut situs ExRx (ExRx, 2013) yang

berdasarkan hasil studi de Leva mengemukakan bahwa tubuh manusia terbagi atas beberapa bagian dan memiliki berat seperti tabel berikut:



Gambar 2.2 Human Segments

Tabel 2.1 Persentase berat anggota tubuh (ExRx, 2013)

<i>Percentages of Total Body Weight</i>			
<b>Segment</b>	<b>Males</b>	<b>Females</b>	<b>Average</b>
Head & Neck	6.94	6.68	6.81
Trunk	43.46	42.58	43.02
Upper Arm	2.71	2.55	2.63
Forearm	1.62	1.38	1.5
Hand	0.61	0.56	0.585
Thigh	14.16	14.78	14.47
Shank	4.33	4.81	4.57
Foot	1.37	1.29	1.33

lalu jika dibuat tabel rata-rata maka hasilnya sebagai berikut:

Tabel 2.2 Persentase rata-rata berat anggota tubuh (ExRx, 2013)

<i>de Leva's Segment Weight data</i>			
<b>Segment</b>	<b>Quantity</b>	<b>Percent</b>	<b>Extension</b>
Head	1	6.810	6.81
Trunk	1	43.020	43.02
Total Arm	2	4.715	9.43
Total Leg	2	20.370	40.74
Total Percent:			100

Dapat dilihat dari tabel 2.2 bahwa berat sebuah tangan manusia adalah 4.715 persen dari berat tubuhnya. Maka jumlah persen ini nantinya akan dihitung dengan massa masukan pada rumus energi kinetik.

Dikarenakan perbedaan penamaan antara massa dan berat, apabila dalam ilmu fisika massa merupakan berat dalam ilmu biologi ataupun masyarakat umum dimana satuannya sama-sama gram atau Kilogram. Maka untuk selanjutnya penyebutan massa dan berat adalah memiliki maksud yang sama hanya berbeda bidang studinya saja.

#### 2.4 Microsoft Xbox Kinect

Microsoft Xbox 360 Kinect atau biasa disebut Kinect, pada awalnya memiliki nama Project Natal. Kinect adalah produk dari Microsoft yang memperkenalkan teknologi *motion gaming* sebagai fitur utamanya. Kinect membuat pemain dapat berinteraksi dengan konsol Xbox 360 tanpa bantuan *game controller*. Menggunakan Kinect, pemain dapat bermain Xbox 360 cukup hanya dengan menggunakan gerakan anggota tubuhnya.

Teknologi Kinect diciptakan dengan tujuan untuk memperluas peminat konsol Xbox 360 diluar batas kalangan gamer. Saat ini Kinect bersaing ketat dengan Playstation Move milik Sony dan Wii MotionPlus milik Nintendo. Kinect pertama kali dirilis pada tanggal 4 November 2010 di wilayah USA.

Kinect dibuat berdasarkan sistem software dari developer Rare yang merupakan bagian dari Microsoft Game Studios. Sedangkan teknologi kamera yang digunakan oleh Kinect dibuat dari developer Israel, PrimeSense.

Kinect dilengkapi dengan kamera RGB, *Depth Sensor*, *Multi-Array Microphone* untuk menangkap dan mengenali suara, dan dilengkapi sebuah *Tilt motor* agar bisa menyesuaikan derajat tangkapan kamera. Teknologi *Depth Sensor* Kinect merupakan sensor tiga dimensi (3D) untuk mengenali gerakan pemain.

Sensor ini dapat mengenali sampai enam orang sekaligus, namun hanya dua pemain berstatus aktif yang dapat dideteksi gerakannya oleh Kinect. *Depth Sensor* terdiri dari sebuah proyektor *Infra-Red* (IR) yang dikombinasikan dengan sensor monokrom CMOS. Inilah yang dapat membuat Kinect melihat dalam bentuk 3D dalam keadaan cahaya apapun. Kalkulasi jarak antara obyek yang ditangkap dengan Kinect diperoleh berdasarkan sinar IR tersebut. Semakin pendek jaraknya, maka semakin bersinar poin yang ditangkap sensor. Jarak tangkap *Depth Sensor* dapat diatur. Kinect mampu mengkalibrasikan sensor secara otomatis berdasar pola permainan berada, termasuk benda-benda yang berada disekitar pemain.

Inovasi utama dari Kinect adalah kemampuannya yang lebih maju dalam mengenali wajah, gerakan dan suara. Kinect menghasilkan video dengan 30Hz *frame rate*. Dengan *video stream* RGB pada VGA yang beresolusi 11-bit (640 x 480 pixel dengan tingkat sensitivitas 2048). Sensor memiliki daerah pandang angular dengan sudut 57 derajat pada bidang horizontal dan 43 derajat pada bidang vertikal. Sensor ini juga dapat dimiringkan hingga 27 derajat ke atas ataupun ke bawah.

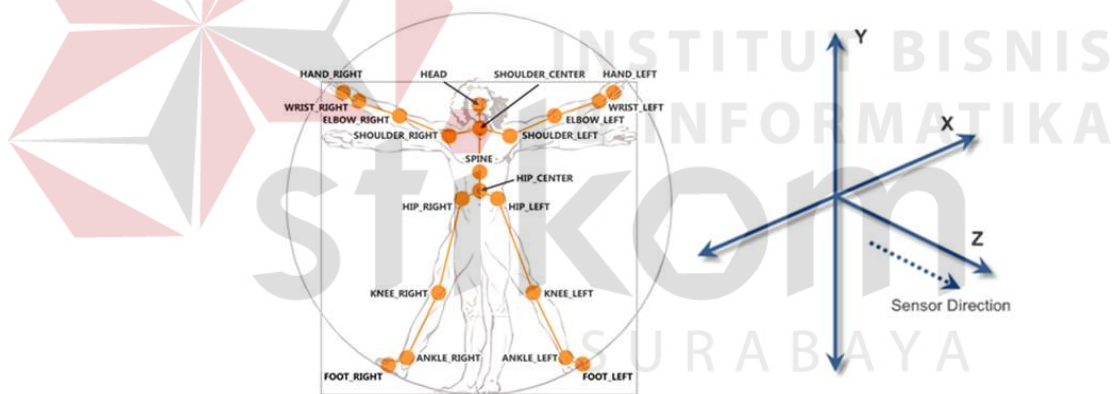
Kinect mampu menangkap dan mendeteksi gerakan tubuh secara akurat, pemain hanya menggunakan tubuh untuk berinteraksi dengan *dashboard* konsol Xbox 360, bermain game, bahkan untuk mengakses fitur-fitur Xbox Live.

Mekanisme kerja Kinect menangkap dan mengenali pergerakan tubuh dari pemain adalah sebagai berikut:

1. Posisi pemain harus menghadap Kinect terlebih dahulu.
2. Kemudian Kinect menangkap titik-titik persendian pada tubuh yang disebut *Joint*. Ada 20 titik pada tubuh yang dapat terdeteksi oleh Kinect meliputi

kepala, bahu tengah, bahu kanan, bahu kiri, siku kanan, siku kiri, pergelangan tangan kanan, pergelangan tangan kiri, tangan kanan, tangan kiri, tulang punggung, pinggul tengah, pinggul kanan, pinggul kiri, pinggul tengah, lutut kanan, lutut kiri, pergelangan kaki kanan, pergelangan kaki kiri, kaki kanan dan kaki kiri.

3. *Joint* yang tertangkap oleh sensor tadi dikonversi menjadi titik koordinat 3D (x,y,z) bertipe *float*. Sehingga setiap pemain melakukan gerakan, maka titik koordinat dari *joint* yang bersangkutan juga turut berubah pula. Sumbu x merepresentasikan gerakan pemain secara horizontal, sumbu y merepresentasikan gerakan pemain secara vertikal, sumbu z merepresentasikan jauh dekatnya posisi pemain dari Kinect.



Gambar 2.3 *Joints* dan *Orientation* pada Kinect

(Webb and Ashley, 2012 dan Catuhe,2012)

## 2.5 Microsoft Kinect SDK

Kinect for Windows SDK adalah *toolkit* pemrograman untuk developer aplikasi. Hal ini memungkinkan para akademik dan komunitas untuk mengakses

kemampuan yang ditawarkan oleh perangkat Microsoft Kinect yang terhubung ke komputer dengan OS Windows 7.

SDK ini dilengkapi driver, API untuk *raw sensor streams*, *skeletal tracking*, dokumentasi instalasi dan resource lainnya. SDK ini juga menyediakan kemampuan-kemampuan Kinect bagi para developer yang akan membuat aplikasi dengan C++, C# maupun Visual Basic dengan menggunakan Microsoft Visual Studio 2010.

SDK ini mencakup beberapa fitur diantaranya:

1. Raw Sensor Streams

Akses ke *raw data stream* dari *depth sensor*, *color camera sensor* dan empat *microphone array* memungkinkan developer untuk membangun proyek mereka menggunakan *low-level stream* yang dihasilkan oleh Kinect.

2. Skeletal Tracking

Kemampuan untuk melacak gambar kerangka satu atau dua orang yang bergerak dalam bidang Kinect sehingga mudah untuk membuat aplikasi berbasis gerakan.

3. Kemampuan audio yang canggih

Kemampuan pemrosesan audio yang canggih termasuk *acoustic noise suppression* dan *echo cancellation*, *beam formation* untuk mengidentifikasi sumber suara dan terintegrasi dengan Windows Speech Recognition API.

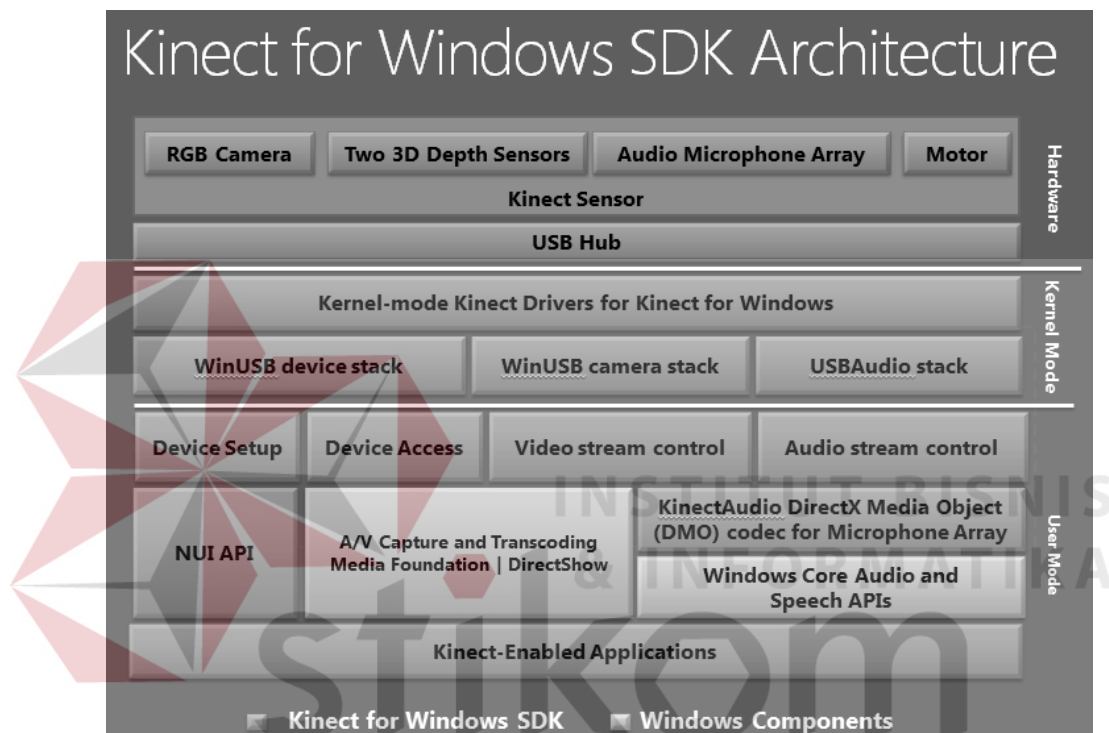
4. Contoh kode dan dokumentasi

SDK mencakup lebih dari 100 halaman dokumentasi teknis. Selain file bantuan, dokumentasi termasuk panduan rinci untuk sampel yang disediakan dengan SDK.

## 5. Instalasi mudah

Instalasi SDK berlangsung cepat, tidak memerlukan konfigurasi yang rumit dan ukuran installer kurang dari 300MB.

Kinect SDK memiliki arsitektur sebagai berikut, agar aplikasi yang dibuat menggunakan SDK bisa berkomunikasi dengan Kinect. (Microsoft, 2012).



Gambar 2.4 Arsitektur Kinect

## 2.6 Bahasa Pemrograman C#

C# sering dianggap sebagai bahasa penerus C++ atau versi canggih, karena ada anggapan bahwa tanda # adalah perpaduan 4 tanda tambah yang disusun sedemikian rupa sehingga membentuk tanda pagar. Akan tetapi, terlepas dari benar tidaknya anggapan tersebut, C# adalah sebuah bahasa pemrograman yang berorientasi pada objek yang dikembangkan oleh Microsoft dan menjadi salah



satu bahasa pemrograman yang mendukung .NET programming melalui Visual Studio.

C# didasarkan pada bahasa pemrograman C++. C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti Visual Basic, Java, Delphi dan tentu saja C++. C# memiliki kemudahan syntax seperti Visual Basic, dan tentu saja ketangguhan seperti Java dan C++. Kemiripan-kemiripan ini tentunya memudahkan programmer dari berbagai latar belakang bahasa pemrograman tidak perlu waktu yang lama untuk menguasainya, karena bagaimanapun juga C# lebih sederhana dibandingkan bahasa-bahasa seperti C++ dan Java.

C# didesain oleh program designer dari Microsoft, Anders Hajlsberg. Sebelum bekerja pada Microsoft, Anders bekerja di borland, tempat dia menulis Pascal compiler. Sebelum mengembangkan C#, Anders mengetahui berbagai macam kekurangan pada bahasa C++, Delphi, Java, dan Smaltalk, karena itu Anders menciptakan bahasa C# yang lebih tangguh. Hal ini juga menjelaskan mengapa C# memiliki kemiripan dengan beberapa bahasa tersebut dan menjadi salah satu alasan mengapa C# memiliki lebih banyak nilai tambah apabila digunakan untuk mengembangkan aplikasi Kinect menggunakan SDK milik Microsoft dibandingkan menggunakan bahasa C++ ataupun VB.Net. (Microsoft, 2013).

## **2.7 UML**

Notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbough, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon, dan lainnya. Jacobson menulis tentang pendefinisian persyaratan – persyaratan sistem yang

disebut use case. Juga mengembangkan sebuah metode untuk perancangan sistem yang disebut Object Oriented Software Engineering (OOSE) yang berfokus pada analisis. Booch, Rumbaugh, dan Jacobson biasa disebut dengan tiga sekawan (tree amigos). Semuanya bekerja di Rational Software Corporation dan berfokus pada standarisasi dan perbaikan ulang UML.

Penggabungan beberapa metode menjadi UML dimulai 1993. Setiap orang dari tiga sekawan di Rational mulai menggabungkan idenya dengan metode – metode lainnya. Pada akhir tahun 1995 Unified Method versi 0.8 diperkenalkan. Unified Method diperbaiki dan diubah menjadi UML pada tahun 1996, UML 1.0 disahkan dan diberikan pada Object Technology Group (OTG) pada tahun 1997, dan pada tahun itu juga beberapa perusahaan pengembang utama perangkat lunak mulai mengadopsinya. Pada tahun yang sama OTG merilis UML 1.1 sebagai standar industri.

