BAB II

TEORI PENUNJANG

Bab ini akan membahas tentang teori dasar dari sistem pemainan dan teori mengenai sistem *kinect.*

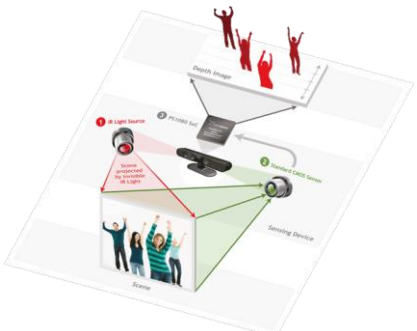
2.1 Teori Sistem *Kinect*

Kinect adalah produk dari *Microsoft* yang awalnya dibuat khusus untuk perangkat permainan Xbox 360, dimana memperkenalkan teknologi *motion gaming* sebagai fitur utamanya. *Motion gaming* maksudnya adalah membuat pemain dapat berinteraksi ketika bermain *game* tanpa menggunakan *gamepad*. Sehingga melalui *kinect*, pemain dapat bermain *game* cukup hanya dengan menggunakan gerakan tangan atau gerakan tubuh lainnya [14]. Terdapat beberapa perangkat inti yang berada di dalam *Kinect* yaitu:

1. *3D Depthsensor*
2. *RGB Camera*
3. *Motorized Tilt*
4. *Multi-Array Microphone*

2.1.1 *3D Depth Sensor*

3D Depth sensor bertujuan untuk mendapatkan data video dalam kondisi tiga dimensi di dalam kondisi *ambient light* (menyesuaikan kondisi pencahayaan yang ada di lingkungan tersebut). Agar objek dapat terdeteksi dengan baik maka objek harus berada di antara IR *projector* dan *CMOS sensor.* Jarak minimum *kinect* terlhadap objek, kurang lebih 1,5 meter.



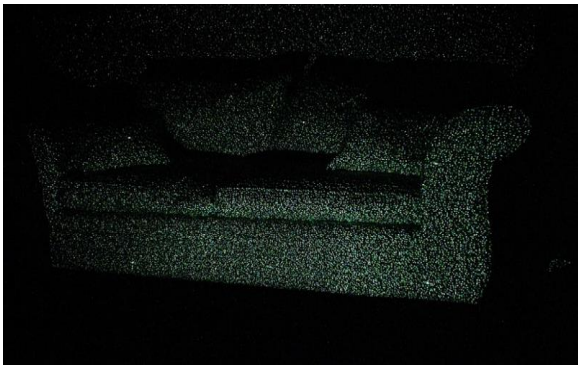
Gambar 2.1 Cara kerja *3D depth sensor* pada perangkat sensor *kinect.*

2.1.1.1 *Infrared Laser Projector*

Infrared laser projector berfungsi dalam men-transmisikan cahaya *invisible near-infrared* ke seluruh bagian ruangan yang terjangkau oleh *sensor kinect*. Cahaya *near–infrared* tidak dapat dilihat secara kasat mata dan menggunakan laser kelas 1 yang aman untuk tubuh manusia [10]. Untuk dapat melihat cahaya *near*-infrared maka dibutuhkan *camcorder* atau kamera yang memiliki fitur *night vision.*



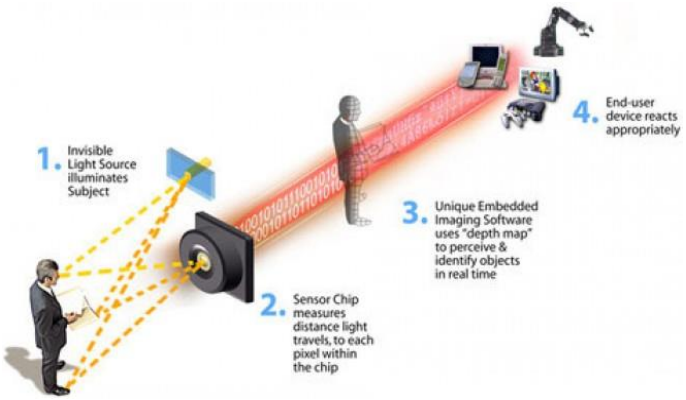
Gambar 2.2 Foto sofa dengan *infrared laser projector,* secara kasat mata.



Gambar 2.3 Foto sofa dengan *infrared laser projector,* saat menggunakan *camcorder night vision*.

2.1.1.2 *Monochrome CMOS Sensor*

Fungsi dari *monochrome CMOS sensor* adalah mengukur waktu penerbangan cahaya setelah terpantul oleh objek didepannya. Sistem ini bekerja layaknya sebuah sonar, bila diketahui berapa lama waktu yang dibutuhkan untuk cahaya tersebut kembali, maka dapat diketahui jarak suatu objek dari sensor [5]. Hal ini dapat dilihat pada gambar 2.4.



Gambar 2.4 Cara kerja sensor *kinect* [5]

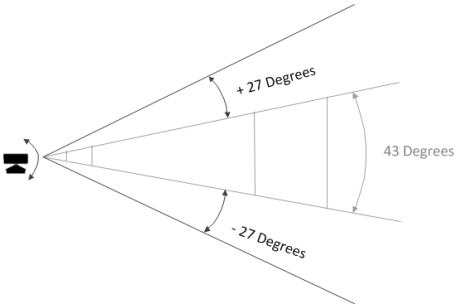
2.1.2 Kamera *RGB (Red, Green, and Blue)*

Sebuah *RGB camera* dapat memberikan tiga komponen warna dasar yaitu merah, hijau, dan biru. Jenis kamera ini menggunakan sensor CMOS (*Complementary metal-oxide-semiconductor*) untuk memperoleh tiga sinyal warna. *RGB camera* digunakan untuk mendapatkan warna gambar yang sangat akurat.

Menurut Kramer [11] *RGB camera* di *kinect*, beroperasi di 30Hz dan dapat menghasilkan beberapa pilihan resolusi dari 640 X 512 hingga 1280 X 1024 piksel, yang akan dikirim menggunakan USB (*Universal Serial Bus).* Dan terdapat beberapa keunggulan *RGB camera* di *kinect* yaitu memiliki *white balance, black refrence,* dan *color saturation* yang sangat baik.

2.1.3 Motor Penggerak

Menurut Singh [21] Motor penggerak *(Motorized tilt)* di dalam *kinect* digunakan untuk menyesuaikan posisi kamera dalam mencari objek. *Motorized tilt* dapat menggerakan dan meningkaatan jarak interaksi pada +27 hingga -27 derajat. Hal ini dapat dilihat pada gambar 2.5.



Gambar 2.5 Jarak interaksi *kinect.*

2.1.4 *Multi-array Microphone*

Menurut Catuhe [6] *multi-array microphone* terdiri dari empat buah mikrofon yang terpisah dan diatur dengan pola linear. Dengan menggunakan struktur ini dapat menghasilkan fungsi-fungsi sebagai berikut:

1. *Noise suppresion*

   Dengan adanya fungsi *noise suppresion*, maka tingkat dapat mengurangi suara yang tidak kita inginkan.
2. *Acoustic echo cancellation* (AEC)

   Fungsi dari *acoustic echo cancellation* untuk membuat kualitas suara yang didapat menjadi lebih baik.
3. *Beamforming and source localization*

   Dengan adanya fungsi ini, maka setiap mikrofon dapat menerima suara tertentu dalam waktu berbeda, sehingga dapat menentukan arah sumber suara.

2.1.5 Cara Kerja *Kinect*

Agar *kinect* dapat mendeteksi tubuh manusia maka dilakukan perhitungan kedalaman peta menggunakan proyektor inframerah dan sensor inframerah yang memancarkan *speckle pattern*, teknik ini biasanya disebut teknik cahaya terstruktur. Prinsip umum cahaya terstruktur adalah memproyeksikan pola yang dikenal (*scene*) dan menyimpulkan kedalaman dari deformasi pola tersebut.

Kinect Menggabungkan cahaya terstruktur dengan dua teknik visi komputer klasik, yaitu :

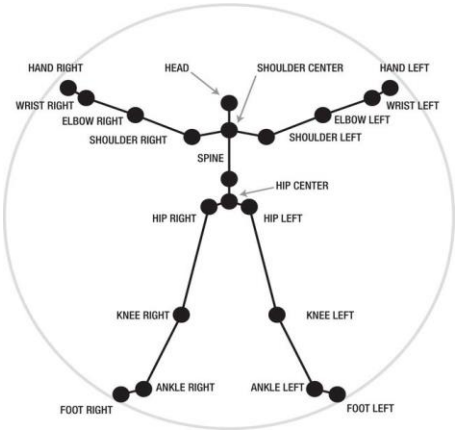
1. Kedalaman dari fokus.

   Kedalaman dari fokus menggunakan prinsip objek yang terlihat kabur (*blur*) maka akan terlihat jauh, dan sebaliknya jika objek yang terlihat tidak kabur (*blur*) maka akan terlihat dekat.
2. Kedalaman dari stereo.

   Kedalaman dari stereo menggunakan paralaks. Jika seseorang melihat suatu *scene* dari sudut lain, benda yang dekat akan terlihat bergeser ke samping lebih dari benda yang berada pada tempat yang jauh.

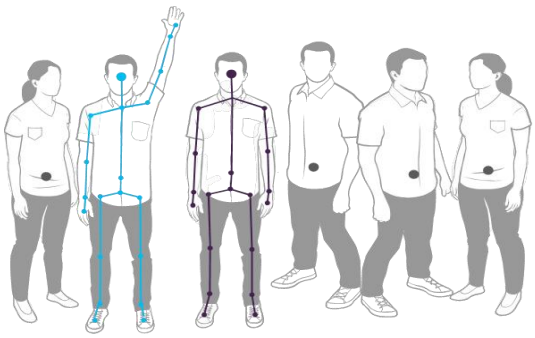
Dalam mendeteksi posisi tubuh dilakukan dengan dua tahap. Pertama mengubah kedalaman *image* menjadi gambar bagian tubuh, dan kedua mengubah gambar bagian tubuh menjadi kerangka *(skeletal)*. Untuk dapat mendeteksi kerangka (*skeletal tracking*) digunakanlah *Natural User Interface* (NUI) yang terdapat dalam *kinect SDK*.

Skeletal Tracking adalah untuk memungkinkan sensor *kinect* dapat mencari titik sendi utama pada tubuh manusia. Teknologi dibalik *skeletal tracking* ini sendiri adalah penggunaan *depth sensor*. *Depth sensor* awalnya akan memetakan objek yang ditangkap kemudian gambar tersebut akan dibandingan dengan data hasil *training* yang sebelumnya. *Data training* tersebut dibuat oleh para peneliti dengan menggunakan 100.000 *frame* gambar objek manusia yang diambil dengan posisi yang berbeda-beda [1].



Gambar 2.6 Titik-titik sendi yang terdapat pada *skeletal tracking.*

Skeletal Tracking dapat mendeteksi hingga enam orang di depan *kinect*, tetapi hanya dua orang yang akan terdeteksi sempurna. Seperti gambar 2.7 dibawah



Gambar 2.7 Hanya dapat mendeteksi dua orang secara sempurna.

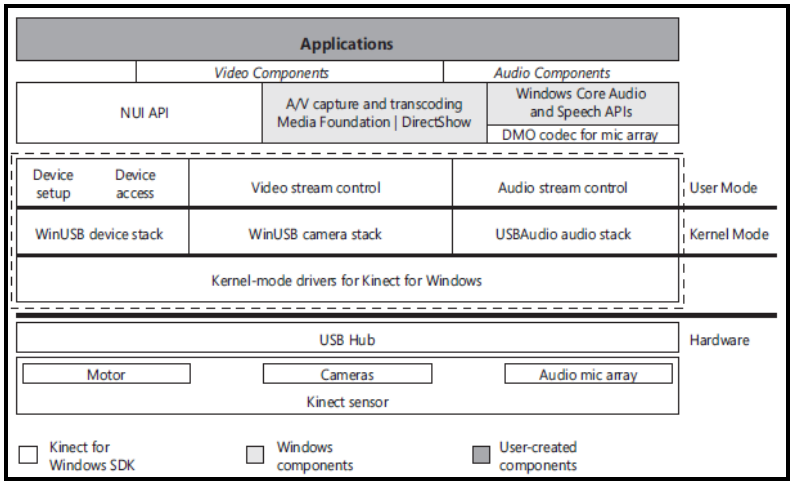
2.1.6 *OpenNI (Open Natural Interaction)*

Berdasarkan web resmi dari *Microsoft*, *OpenNI* adalah pustaka (*library*) yang membantu menghubungkan perangkat *kinect* ke *Unity3D* pada komputer.

OpenNI adalah *user interface* standar untuk algoritma pengolahan data *sensor3D*. *OpenNI* terbuka untuk semua (*open source)*. Tujuan *OpenNI* adalah untuk mendefinisikan tipe data (kedalaman peta, warna peta, pose, dll) dan sebuah *user interface* untuk modul yang dapat menghasilkan *OpenNI* sensor dan *OpenNI* kerangka algoritma, untuk pengembang pihak ketiga.

2.1.7 *Kinect* SDK *(Software Development Kit)*

Menurut Catuhe [6], *Kinect* SDK dapat menghitung secara lengkap dalam melakukan pelacakan kerangka (*skeletal tracking*) di depan sensor yang hanya menggunakan *raw stream*. Didalam SDK sudah terdapat NUI *(Natural User Interface)*, API *(Application Programming Interface)* dan *driver* Microsoft *Kinect* agar dapat mengintergrasikan *kinect* terhadap Microsoft Windows.



Gambar 2.8 Arsitektur SDK *Kinect.*

Dapat dilihat pada gambar 2.8, *stream* ditransmisikan ke komputer mengunakan *USB hub*. NUI API mengumpulkan *raw data* dan mempersiapkannya kepada *application*, tetapi SDK juga terintegrasi di dalam Windows.

- *Audio, speech,* dan media API digunakan untuk aplikasi seperti Microsoft *Speech*.

- *DirectX Media Object (DMO)* digunakan pada aplikasi seperti *DirectShow* atau *Media Foundation*.

2.2 *Game*

Game adalah suatu aktivitas pemecahan masalah yang didekati dengan cara yang menyenangkan. Namun tidak semua pemecahan masalah dikatakan *game*. Suatu pemecahan masalah yang dilakukan dengan cara bermain yang menyenangkan dapat dikatakan suatu *game* [20].

Game merupakan hal yang sangat rumit. ketika orang bermain *game*, mereka dihibur dengan aktif berpartisipasi. Meskipun digunakan untuk hiburan interaktif untuk mengacu pada komputer dan video *game*, setiap *game* bersifat interaktif jika Anda berpartisipasi di dalamnya, meskipun itu adalah softbol ataupun catur [19].

2.2.1 *Game Design*

Menurut Rolling dan Adams [19], *game design* adalah proses dari mengimajinasikan sebuah *game*, menjelaskan cara *game* itu bekerja, mendeskripsikan elemen yang menyusun permainan itu (konseptual, fungsional, artistik, dan yang lainnya) dan mentransmisikan informasi pada kelompok yang akan membangun permainan tersebut.

Game dapat dikategorikan berdasarkan *gameplay* menjadi beberapa *genre*, yaitu sebagai berikut:

- *Action Games*, perancangan *game* dengan genre ini bervariasi dimulai dari 2D sampai 3D. Kemampuan pemain diuji berdasarkan *reaction time* dan *hand-eye coordination* di bawah tekanan.

- *Strategy Games*, merupakan *game* yang memerlukan analisa dan perencanaan beserta pertimbangan terhadap langkah-langkah yang akan diambil selanjutnya untuk memenangkan permainan.

- *Role Playing Games*, adalah sebuah *game* di mana pemain dapat membayangkan dan berempati terhadap karakter yang dikendalikan.

- *Sport Games*, adalah permainan olahraga yang direalisasikan ke dalam bentuk *game*. Sebuah *sport games* memiliki peraturan yang sama atau mendekati dengan permainan olahraga asli. Umumnya *sport games* memiliki *match mode* di mana pemain dapat bertanding dengan komputer atau pemain lainnya. Intinya, yang menjadi fondasi dari *sport games* adalah sebuah pertandingan.

- *Vehicle Simulations*, *game* dengan g*enre* ini berusaha untuk menciptakan suasana seolah seseorang dapat mengendalikan sebuah kendaraan. Kendaraan tersebut dapat berupa mobil, pesawat, kapal selam atau kendaraan imajinasi. Ciri-ciri dari *vehicle simulations* adalah dapat memberikan gambaran bagaimana menjalankan sebuah kendaraan asli dengan memainkan *game* tersebut.

- *Construction and Management Simulation*, merupakan *game* yang menggambarkan sebuah proses. Tujuan dari permainan bukanlah mengalahkan musuh atau orang lain, akan tetapi permainan lebih menitik beratkan pada pencapaian yang dilakukan.

- *Adventure Games*, *game* dengan genre ini tidak menuntut pemain untuk bertindak dengan cepat, berbeda dengan *game* dengan genre *action game*. *Adventure games* memiliki sifat-sifat menelesuri, mengumpulkan objek dan menyelesaikan *puzzle*.

- *Artificial Life*, *m*erupakan *game* yang menggambarkan siklus hidup.

- *Puzzle Games*, adalah *genre* di mana pemain hanya perlu menyelesaikan *puzzle*. Terkadang *puzzle games* tidak menggabungkan *puzzle* dengan *storyline* atau tujuan yang lebih besar.

Element yang ada pada *game* adalah sebagai berikut:

- *Game Concepts*, adalah tahap di mana ide untuk sebuah *game* berasal dan bagaimana cara untuk membentuknya.

- *Game's setting and world*, merupakan tempat terjadinya *gameplay* dan berlangsungnya *game*. Dimensi dari *game world* adalah dimensi fisik, dimensi temporal, dan dimensi lingkungan.

- *Storytelling and Narrative*, merupakan cara membuat cerita dan cara menyeimbangkan antara keinginan desainer dan pemain.

- *Character Design*, merupakan proses pembuatan karakter khususnya yang akan merepresentasikan pemain.

- *Gameplay*, merupakan tantangan yang terjadi dan bagaimana mengatasinya.

- *Internal economy or core mechanics of a game*, merupakan aliran sumber daya seperti uang, poin, amunisi yang ada dalam *game*.

2.2.2 *Game Balance*

Permainan yang seimbang adalah permainan di mana faktor penentu utama keberhasilan pemain adalah tingkat kemampuan pemain itu sendiri. Hal ini tidak berati bahwa peristiwa acak tidak dapat terjadi, tetapi hal tersebut mengimplikasikan bahwa pemain yang lebih baik biasanya harus lebih sukses dari pemain yang buruk kecuali dia bernasib buruk untuk waktu yang lama. Keseimbangan permainan dibagi menjadi dua yaitu, statis dan dinamis [19]. *Game balance* dapat dibagi :

1. *Static Balance*

   Keseimbangan jenis ini berkaitan dengan aturan permainan dan bagaimana aturan-aturan tersebut berinteraksi satu sama lain. Aturan- aturan ini tidak bergantung oleh waktu (*time invariant*). Dengan kata lain, aturan-aturan ini bisa langsung dituliskan untuk membantu permainan.

2. *Dynamic Balance*

   Keseimbangan dinamis mencakup bagian pembukaan, pertengahan dan akhir dari sebuah analisis *game* yang klasik pada skala yang lebih baik. Berbeda dengan keseimbangan statis, harus mempertimbangkan interaksi pemain dengan sistem yang seimbang secara statis. Proses ini tentang perubahan titik keseimbangan sejalan dengan waktu dan interaksi pemain [19].

Tujuan dari *balancing* adalah untuk mengatur sebuah sistem yang seimbang untuk dieksplorasi oleh pemain. Maka dari itu, *gameplay* semestinya diatur untuk menyediakan hal-hal berikut :

- Memberikan tantangan yang konsisten.
- Memberikan pemain pengalaman bermain yang adil dari sisi pemain.
- Menghindari stagnasi.
- Menghindari hal yang tidak perlu.
- Memungkinkan pengaturan dari tingkat kesulitan.

2.2.3 *Game Engine*

Menurut Meigs [13], *game* engine adalah pada dasarnya sekumpulan modul C++ yang besar dan kompleks yang mengendalikan beberapa hal seperti input/output pemain, *rendering system*, sistem animasi, sistem audio, dan keseluruhan literasi logika permainan utama, yang akan memberitahukan sistem target apa yang harus dilakukan dan bagaimana melakukannya.

2.3 *Unity*

Menurut Blackman [3], *Unity* merupakan alat pengembangan *game* yang didesain untuk membiarkan pemakainya fokus dalam mengembangkan *game* yang bagus. *Unity* merupakan pilihan yang tepat bagi studio kecil, pengembang *indie*, dan bagi mereka yang ingin membuat *game* mereka sendiri. *Unity* memiliki basis pengguna yang banyak dan komunitas pemakai yang aktif di mana memungkinkan semua orang awam bertukar pengalaman dari para veteran.

Unity membuka peluang untuk Windows pada musim semi tahun 2009. Sejak saat itu, *Unity* terus memberikan dukungan pada iPhone, Android, iPad, dan Wii dan dukungan pengembangan pada Xbox 360 dan PS3. Adapter awal dari *engine Unity* cenderung untuk berpindah dari *Flash* dan *Director*, membuat *scripting* menjadi lebih mudah diterapkan. Ada banyak bahan untuk membuat *game* memakai Unity, meliputi *first person shooting game, racing game, platform jumper*, dan sejenisnya.

2.4 Visual Studio 2010

Microsoft Visual Studio 2010 merupakan sebuah IDE *(Integrated Development Environment)* yang dikembangkan oleh Microsoft. IDE ini mencakup semua bahasa berbasis .NET framework yang dikembangkan oleh Microsoft. Keunggulan Microsoft Visual Studio 2010 ini antara lain adalah support untuk Windows 7, editor baru dengan WPF *(Windows Presentasion Foundation),* dan banyak peningkatan fitur lainnya.



Gambar 2.9 Visual Studio 2010.

2.5 Kuisioner Menggunakan Skala Likert

Skala likert menurut Nur [9] merupakan metode untuk mengukur persepsi pengguna untuk menyatakan setuju dan tidak setuju terhadap subjek, objek, atau kejadian tertentu.