BAB 2
LANDASAN TEORI

2.1. Game

Game merupakan sebuah bentuk seni dimana penggunanya disebut dengan pemain (*player*), diharuskan membuat keputusan-keputusan dengan tujuan mengelola sumber daya yang diperoleh dari kesempatan-kesempatan bermain (*token*) miliknya untuk mencapai tujuan tertentu. *Video game* adalah bentuk *game* yang interaksi umumnya melibatkan media *video* dan *audio*.

Menurut Andang Ismail terdapat dua pengertian *game* (permainan). Pertama, *game* (permainan) adalah sebuah aktifitas bermain yang murni mencari kesenangan tanpa mencari menang atau kalah. Kedua, permainan diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan, namun ditandai pencarian menang-kalah. [6]

2.1.1. Jenis-jenis *Video Game*

Berikut ini beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya. Jenis-jenis game tersebut adalah: [7]

2.6.1.1. *Game PC*

Game PC adalah *game* yang dimainkan pada PC (*Personal Computer*) yang memiliki kelebihan yaitu tampilan antarmuka yang baik untuk *input* maupun *output*. *Output* visual berkualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa. Kekurangannya adalah spesifikasi komputer yang sangat bervariasi antar satu komputer dengan komputer yang lainnya menyebabkan beberapa *game* dapat ditampilkan dengan baik pada satu komputer tetapi tidak berjalan dengan baik pada komputer yang lainnya. Berikut adalah gambar contoh *game pc*:

Gambar 2.1 Contoh *Game PC* (Fortnite Battle Royale)

2.6.1.1. *Game Console*

Game console adalah *game* yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah seperti PlayStation, Xbox, Nintendo, Wii dan lain-lain. Berikut adalah gambar dari perangkat *game console* dan contoh *game console*.

Gambar 2.2 Perangkat *Game Console*

Gambar 2.3 Contoh *Game Console* (FIFA 2017)

2.6.1.2. *Game Arcade*

Game arcade adalah *game* yang dijalankan pada mesin dengan *input* dan *output audio* visual yang telah terintegrasi dan tersedia ditempat-tempat umum. Beikut adalah perangkat dari *game arcade* dan contoh *game arcade*.

Gambar 2.4 Contoh *Game Arcade*

2.6.1.3. *Game Online*

Game online adalah *game* yang hanya dapat dimainkan secara *online* melalui koneksi LAN atau koneksi internet. Untuk *game online* dapat dimainkan diperangkat manapun dengan kondisi perangkat tersebut dapat terkoneksi internet. Berikut adalah gambar contoh dari *game online*.

Gambar 2.5 Contoh *Game Online*

2.2. Game Controller

Untuk bermain *game* disediakan alat *controller* fisik yang dapat di bagi menjadi beberapa jenis diantaranya adalah *joystick, mechanical keyboard*, dan *mouse macro*.

2.7.1. *Joystick*

Joystick adalah alat input komputer yang berwujud tuas atau tongkat dan dapat bergerak ke segala arah, sedangkan ada juga *joystick* berbentuk kotak atau persegi terbuat dari plastik dilengkapi dengan tuas dan tombol-tombol yang akan mengatur gerak suatu objek dalam komputer. Alat ini dapat mentransmisikan arah sebesar dua atau tiga dimensi ke komputer.

Alat ini umumnya digunakan sebagai pelengkap untuk memainkan permainan *video* yang dilengkapi lebih dari satu tombol. Biasanya perangkat *joystick* digunakan pada sega dan playstation namun bisa juga diterapkan pada *game* PC. Kegunaan *joystick* pun tergantung pada *genre game* yang dimainkan.

Pada *genre arcade* atau *adventure* bisa menggunakan *joystick rumble*, pada *genre racing* atau balap mobil bisa menggunakan *joystick racing wheel*, kemudian jika ingin memainkan *game simulasi* pesawat terbang bisa menggunakan *Joystick Extreme 3D Pro*. Berikut adalah gambar bentuk dari *joystick.*

Gambar 2.6 *Joystick*

2.7.2. *Mechanical Keyboard*

Keyboard adalah alat input perangkat keras pada komputer yang berbentuk papan dengan berbagai macam fungsi perintah. *Keyboard* dapat digunakan untuk bermain *game* pada jenis *game* PC (komputer). Biasanya para pemain

menggunakannya untuk *game* sejenis RPG (*role playing game*) dan FPS (*first person shooter*).

Sedangkan *mechanical keyboard* atau *keyboard* mekanikal adalah *keyboard* yang diciptakan khusus untuk pemain dengan kualitas tinggi menggunakan teknologi *spring activated* dalam *switch keyboard*. *Mechanical keyboard* lebih unggul dalam segala hal (*Switch, framing, functionality, type print methods, key construction, PCB board, LED lightning* (*sharpness, brightness, adjustability*)).

Kelebihan itu yang membuat para pemain merasa lebih nyaman menggunakannya karena kulaitasnya lebih baik dibandingkan dengan *rubber dome keyboard* atau *keyboard* biasa. Namun *keyboard* biasa pun suduh cukup baik untuk pemain pemula. Berikut adalah gambar bentuk dari *mechanical keyboard*.

Gambar 2.7 *Mechanical Keyboard*

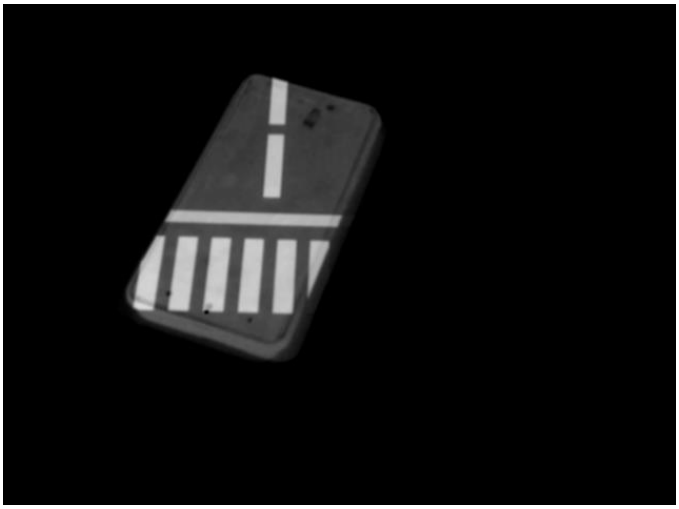
2.3. Pengolahan Citra

Computer vision adalah otomatisasi dalam pengambilan informasi dari sebuah citra. Model 3D, Posisi Kamera, Deteksi dan Rekognisi Objek adalah jenis jenis yang informasi nya bisa diolah dengan pengolahan citra. Pengolahan citra tidak hanya berisi tentang rekognisi dan deteksi objek pada citra, namun di dalamnya berisi ada juga programming, matematika dan modelling dicampur menjadi satu. [8] Boyle dan Thomas (1988) menyatakan bahwa pengolahan

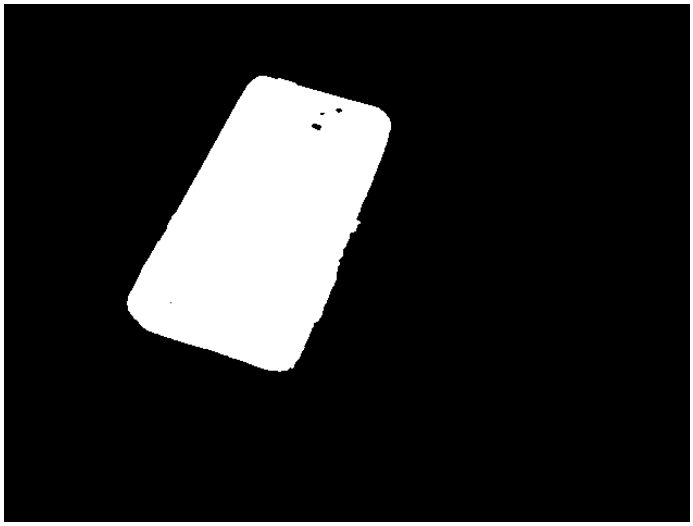
menghadirkan *low level processing* yang dijadikan sebagai algoritma pengolahan citra yang dapat disebut purely. [9]

2.4. *Thresholding*

Threshold adalah batas, di mana kita menetapkan nilai, apakah di atas atau di bawah. Metode ini digunakan biasanya untuk pemisahan objek dengan selain objek, biasanya menghasilkan dua buah warna yaitu hitam dan putih. Batasan biasanya berupa tipe data float antara 0 sampai 1. 0 adalah hitam dan 1 adalah putih. Jika data citra melebihi batas yang ada, maka akan dimasukan menjadi golongan 1 (berwarna putih) jika dibawah atau kurang dari batas yang ada maka akan dibuat masuk ke golongan 0 (berwarna hitam). [10] Menurut Gonzales dan Woods, 2002) Mengatur nilai intensitas semua piksel yang lebih besar dari nilai threshold T sebagai latar depan dan yang lebih kecil dari nilai threshold T sebagai latar belakang untuk partisi citra, adalah langkah yang digunakkan untuk melakukkan thresholding. Sebelum melakukkan thresholding, perlu dilakukkan grayscaling terlebih dahulu. Berikut adalah langkah – langkah thresholding yang bisa bekerja pada citra yang memiliki noise. [11] Berikut adalah gambar asli yang diambil dari video dan hasil *thresholding*-nya.



Gambar 2.8 Gambar Asli



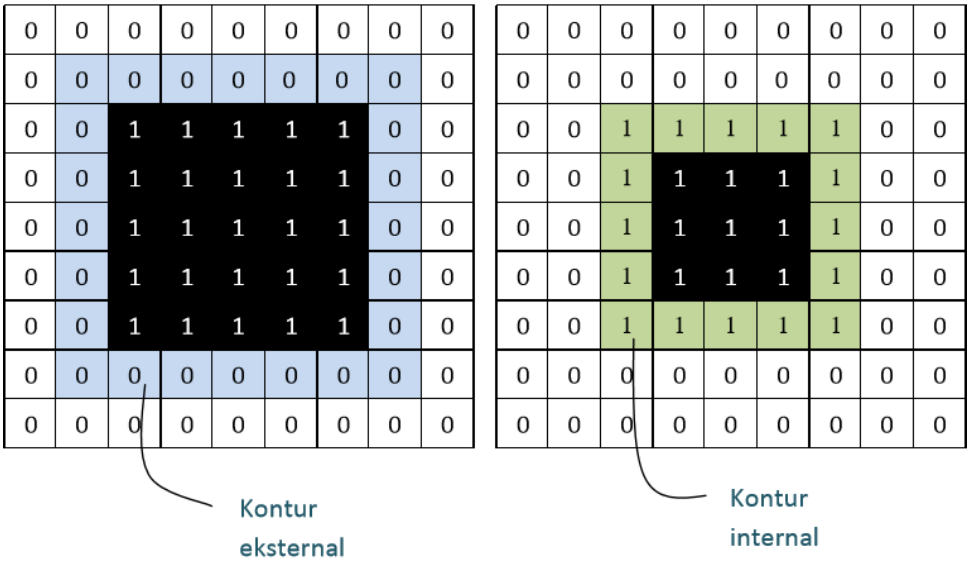
Gambar 2.9 Hasil dari Thresholding Gambar 2.8 Gambar Asli

2.5. *Contour Detection*

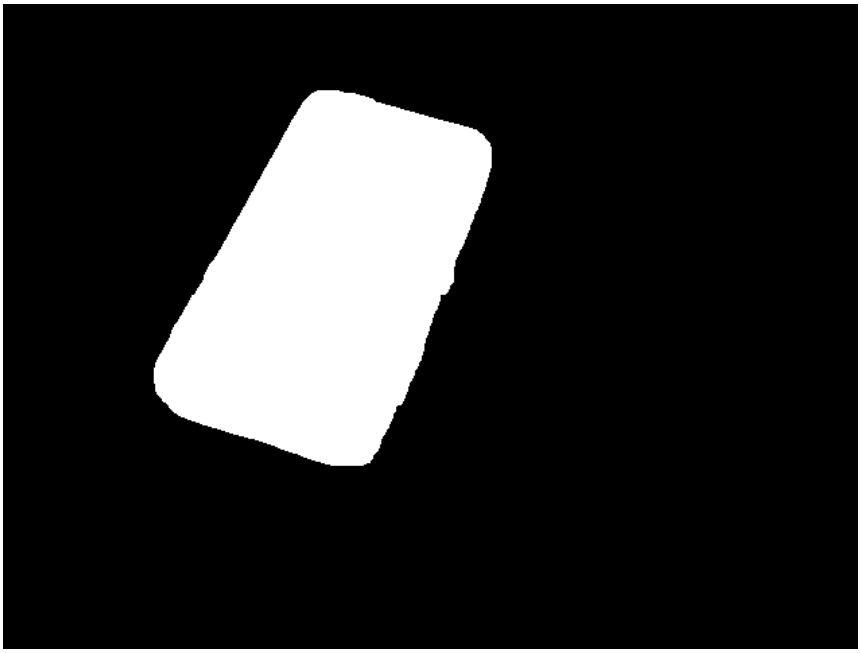
Contour adalah garis batas terluar dari objek yang mempunyai intensitas warna yang sama dan terdeteksi oleh computer. Dalam penelitian yang melibatkan computer vision, contour adalah hal yang paling sering digunakan sebagai alat untuk menganalisa sebuah bentuk. [12] Dapat kita lihat hasil dari thresholding pada gambar 6, menunjukan bahwa tepi dari piksel manusia yang terdeteksi pada gambar masih belum optimal. Hal ini akan mempengaruhi proses deteksi blob. Untuk mengoptimalkan piksel yang terdeteksi pada hasil thresholding maka digunakan deteksi contour, agar piksel menjadi mendekati bentuk manusia yang ada dalam video. Berikut adalah langkah – langkah dan contour yang ditemukkan dari hasil *thresholding* dengan *contour detection.*

Gambar 2.10 Langkah – langkah Deteksi Contour

Contour dibagi menjadi dua jenis, yaitu kontur eksternal dan kontur internal, piksel bagian kontur eksternal terletak di bagian luar objek sedangkan kontur internal ada di dalam objek.



Gambar 2.11 Kontur Eksternal (Kiri) Kontur Internal (Kanan)



Gambar 2.12 Hasil deteksi contour dari thresholding

2.6. Kinect

Gambar 2.13 Fitur-fitur pada Kinect

Kinect adalah produk dari *Microsoft* yang pada awalnya dikususkan untuk *Xbox 360*, dimana memperkenalkan teknologi *motion gaming* sebagai fitur utamanya. *Motion Gaming* maksudnya adalah membuat pemain ketika berinteraksi dengan permainan tanpa menggunakan *controller*. Sehingga ketika bermain, pemain cukup menggunakan gerakan tangan atau bagian tubuh lainnya [13]. Pada *Kinect* terdapat beberapa fitur seperti pada Gambar 2.13 Fitur-fitur pada Kinect yaitu:

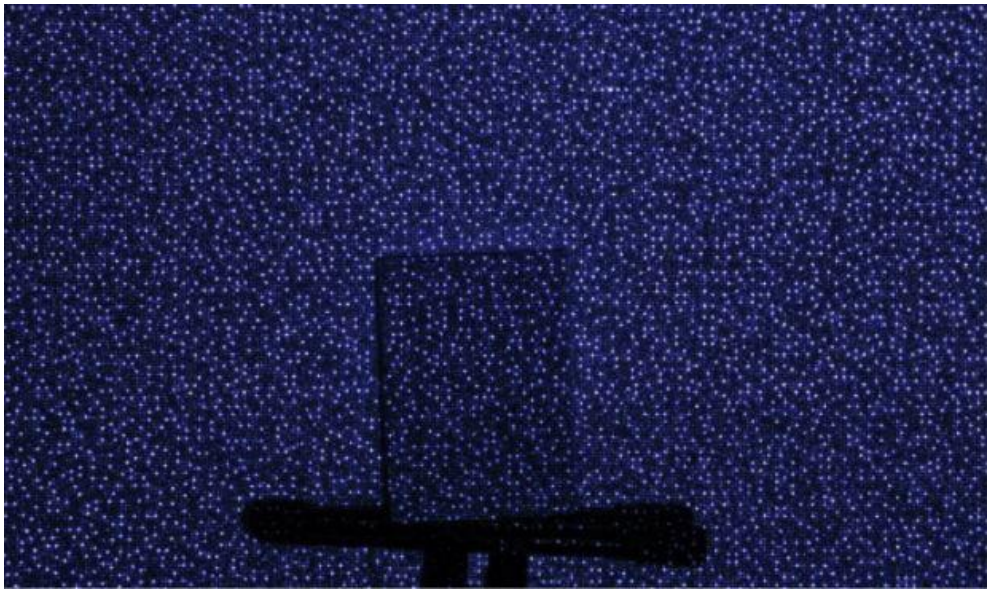
1. *3D Depth Sensor*
2. *RGB Camera*
3. *Motorized Tilt*
4. *Multi-Array Microphone*
5. *Skeleton Tracking*

Kinect juga memiliki beberapa kekurangan pada *optical lenses*, sensor pada *Kinect* dapat bekerja dengan baik pada kondisi berikut:

1. *Horizontal viewing angle 57º*
2. *Vertical viewing angle 43º*
3. Jarak ideal saat menggunakan *Kinect* adalah 1,2 meter sampai 2,5 meter
4. *Depth range*: 400mm sampai 800mm
5. Suhu: 3º sampai 35º

2.6.1. 3D Depth Sensor

3D Depth Sensor terdiri atas kombinasi *infrared laser projector* dan *monochrome CMOS* yang dapat digunakan untuk model tiga dimensi dari gambar yang ditangkap dan disesuaikan dengan kondisi *ambient light* (penyesuaian cahaya dengan lingkungan). Objek terdeteksi dengan baik apabila ada diantara *IR projector* dan *CMOS sensor*. Jarak minimum dengan objek adalah 1,2 meter. [4, 14]



Gambar 2.14 Titik-titik *Infrared* pada dinding [15]

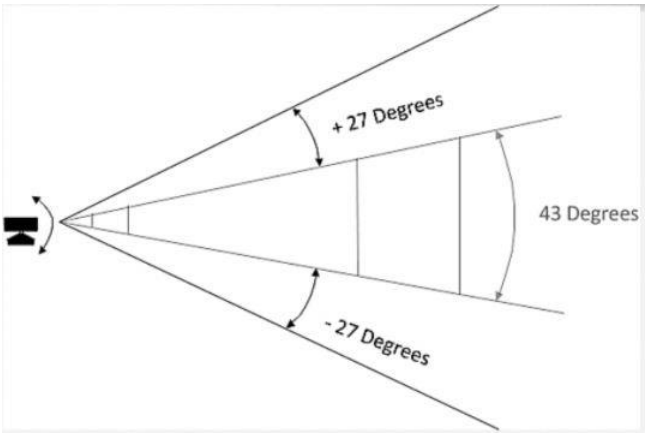
2.6.2. RGB Camera

Sebuah kamera yang dapat menampung 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. [4] Selain itu juga berfungsi sebagai pengenal wajah dengan fitur lainnya. [14]

RGB camera kinect memiliki pilihan resolusi 640 x 512 piksel dengan *framerate* 30 Hz dan 1280 x 1024 piksel dengan *framerate* 15 Hz, yang dikirim melalui USB (*Universal Serial Bus*). [16] Keunggulan *RGB camera* dari *kinect* yaitu memiliki *white balance*, *black refrence*, dan *color saturation* yang sangat baik. [17]

2.6.3. Motorized Tilt

Motorized Tilt pada *kinect* digunakan untuk menyesuaikan posisi kamera dengan objek pada saat mencari objek. *Motorized Tilt* dapat bergerak ke atas hingga 27 derajat dan ke bawah hingga 27 derajat. [15]



Gambar 2.15 Sudu pergerakan kinect [15]

2.6.4. Multi-Array Microphone

Kinect mempunyai empat buah *microphone array* dengan *sampling* 16 KHz dan resolusi 24 bit. [16] Semua *microphone* yang terpisah diatur dengan menggunakan pola linear. [18]

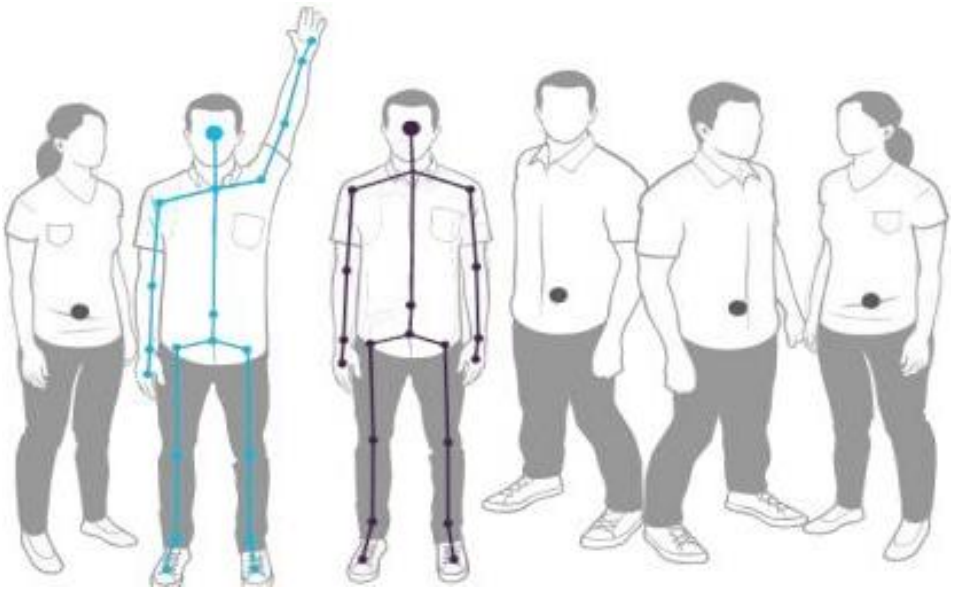
2.6.5. Skeleton Tracking

Skeleton Tracking merupakan fitur yang disediakan oleh *Kinect SDK*, fitur ini dapat melacak titik sendi utama tubuh manusia dan digunakan untuk mendeteksi bagian tubuh manusia dan merepresentasikannya dalam bentuk *skeletal* (tulang). [4] Ada empat tahap untuk dapat mempresentasikan bagian tubuh manusia menjadi bentuk *skeletal* yaitu estimasi pose, merepresentasikan kembali gerakan, *horizontal symmetry*, dan recognisi menggunakan HMM. [19]

Penggunaa *skeleton tracking* tidak lepas dari *depth sensor* yang memetakan objek-objek sendi utama tubuh manusia berdasarkan jarak dengan data latih yang ada. Data latih tersebut terdiri kisaran 100.000 *frame* pose manusia yang sedang berdiri. Oleh karena itu, *Kinect* tidak dapat mendeteksi posisi kaki manusia dengan pada saat duduk. [14]

2.6.6. Microsoft Kinect SDK

Pada tahun 2011 *Microsoft* merilis *Kinect Development Kit (SDK)*. *Kinect SDK* mempunyai fitur yang paling lengkap untuk berinteraksi dengan sensor yang ada pada *Kinect*. Dan bisa estimasi posisi enam orang secara bersamaan yang didepan kinect, tapi hanya dua orang yang dapat terdeteksi secara sempurna dan dibuat *skeletal*. [16]



Gambar 2.16 Hanya mendeteksi dua orang secara sempurna [4]

2.7. OpenCV

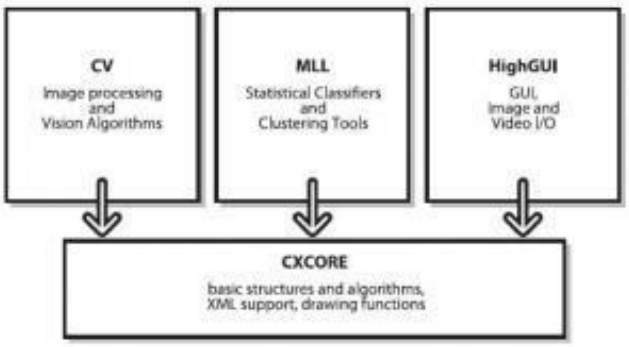
OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) *Library* yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek.

Beberapa pengimplementasian dari *Computer Vision* adalah *Face Recognition*, *Face Detection*, *Face/Object Tracking*, *Road Tracking*, dll. OpenCV adalah *library Open Source Computer Vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk image/video. [20]

OpenCV sendiri terdiri dari 5 *library*, yaitu :

a. CV : untuk algoritma *Image processing* dan *Vision*.
b. ML : untuk *machine learning library.*
c. Highgui : untuk GUI, *Image* dan *Video* I/O.
d. CXCORE : untuk struktur data, *support* XML dan fungsi-fungsi grafis.
e. CvAux.

Struktur dan Konten OpenCV :



Gambar 2.17 Struktur OpenCV

2.8. Object-Oriented Analisys and Design (OOAD)

Analisis dan Desain Berorientasi Objek (*Object Oriented Analysis and Design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Alasan mengapa harus memakai metode berorientasi objek yaitu karena perangkat lunak itu sendiri yang bersifat dinamis, di mana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat.

Tujuannya untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap analisis perancangan dan implementasi relatif sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisanya berbeda-beda. Hal itu menyebabkan transisi antar tahap pengembangan menjadi kompleks. Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah

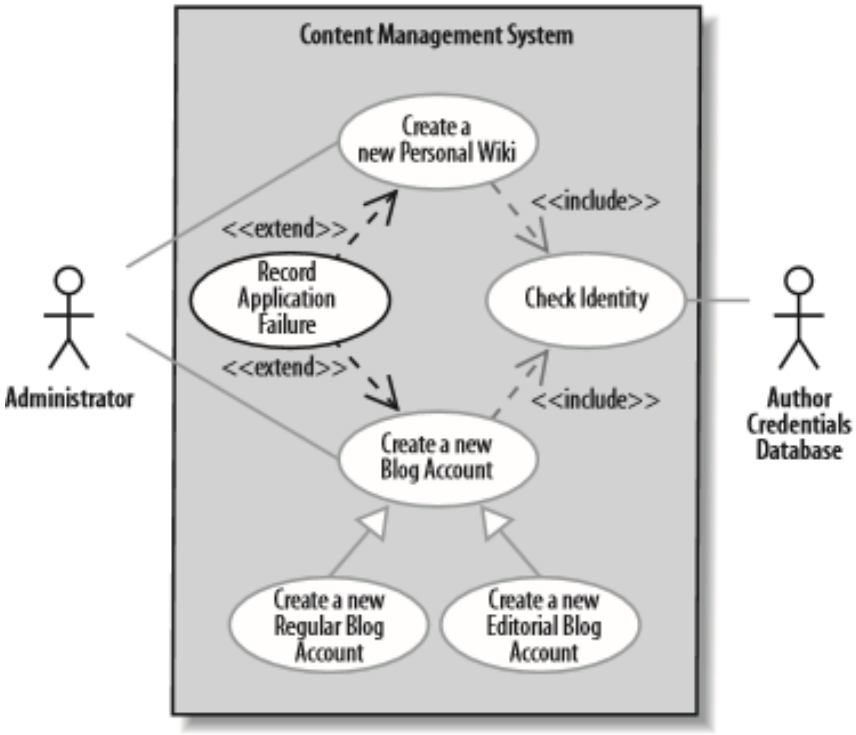
objeknya bukan fungsinya. Beda ceritanya dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance* dan tentunya UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam Bahasa pemrograman berorientasi objek. Selain itu UML merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan software agar bias menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan Testing.

Dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah sebuah Bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, melakukan spesifikasi, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*) [21].

Dokumentasi UML menyediakan 10 macam diagram untuk membuat model aplikasi berorientasi objek yang 5 di antaranya adalah [22]:

2.8.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *use case diagram* ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan bentuk dari sebuah pekerjaan tertentu, misalnya *login* ke dalam sistem, *posting* dan sebagainya, sedangkan seorang *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [23]. Contoh *use case* dapat dilihat pada Gambar 2.18 *Use Case*.



Gambar 2.18 *Use Case*

2.8.2. Use Case Scenario

Sebuah diagram yang menunjukkan use case dan aktor mungkin menjadi titik awal yang bagus, tetapi tidak memberikan detail yang cukup untuk desainer sistem untuk benar-benar memahami persis bagaimana sistem dapat terpenuhi. Cara terbaik untuk mengungkapkan informasi penting ini adalah dalam bentuk penggunaan *use case scenario* berbasis teks per *use case*-nya [23].

2.8.3. Activity Diagram

Activity Diagram adalah sebuah tahapan yang lebih focus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity diagram* ini sendiri memiliki struktur yang mirip dengan *flowchart* atau data flow diagram pada perancangan terstruktur. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram* [23].

2.8.4. Sequence Diagram

Sequance diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan di antaranya di dalam sebuah *use case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan gari putus dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *sequence* diagram adalah memberikan gambaran detail dari setiap *use case diagram* yang dibuat sebelumnya [23].

2.8.5. Class Diagram

Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class,* paket dan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [23].