BAB 2
LANDASAN TEORI

2.1 Teknologi Computer Vision

Computer vision adalah sebuah sistem otomatis analisis citra dan video oleh komputer dengan tujuan untuk memperoleh informasi dan pemahaman dari suatu objek.[18] Secara umum, tujuan dari pengembangan *computer vision* ini adalah untuk menduplikasi kemampuan mata manusia baik dalam hal penangkapan atau perekaman citra, analisis citra ataupun dalam pemahaman citra untuk diterapkan dalam sebuah mesin atau komputer. Dengan kata lain *computer vision* adalah kemampuan sebuah mesin atau komputer untuk melihat dan memahami sebuah citra dengan sama atau bahkan melebihi dari kemampuan penglihatan manusia asli.[19] Yang menjadi materi dan pengembangan dalam *computer vision* diantaranya :

1. Deteksi dan pengenalan objek (*Object Detection and Recognition*).
2. Gerakan (*Motion Detection and Tracking*).
3. Optimasi, peningkatan dan restorasi citra.
4. Pengukuran tingkat kemiripan citra.

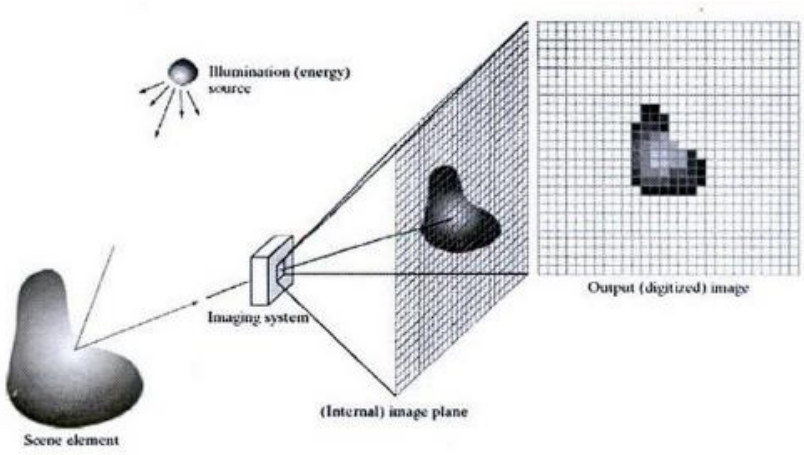
Salah satu bagian yang termasuk dalam computer vision adalah Pengolahan Citra (*Image Processing*). Pengoloahan citra merupakan tahapan utama yang harus dilakukan dalam *computer vision*. Didalam pengolahan citra terdapat 3 tahapan yang dilakukan diantaranya pengambilan citra, sampling, kuantisasi.

2.1.1 Pengolahan Citra

Pengolahan citra ialah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Tujuan utamanya memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer).[1] Dalam proses pengolahan citra, citra akan dianalisis sesuai dengan tujuan akhir yang diarapkan. Hasil dari analisis yang dilakukan adalah bagaimana mendeteksi pergerakan jari tangan manusia, pada prosesnya ada 3 materi terkait diantaranya pengambilan citra, sampling, dan kuantisasi.

2.1.1.1 Pengambilan Citra

Proses pengambilan citra atau disebut akuisisi citra adalah pemetaan suatu pandangan (*scene*) menjadi citra kontinu dengan menggunakan beberapa sensor. Sensor yang dipakai adalah sensor tunggal, sensor garis dan sensor larik. Sensor yang paling sering dipakai adalah sensor larik yaitu sensor yang tertanam di dalam kamera digital. Sensor ini berbentuk larik 2 dimensi. Sensor larik yang terdapat di dalam kamera digital disebut sensor CCD dengan ukuran sensor yang rata-rata mencapai 4.000 x 4.000 elemen.

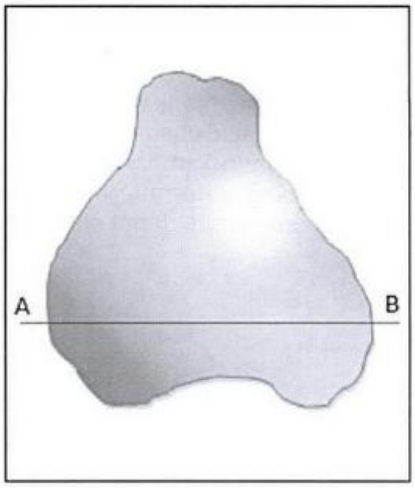


Gambar 2.1 Sensor Larik

Sensor ini akan menangkap setiap iluminasi yang dipantulkan oleh objek dan akan diproyeksikan ke dalam bidang citra. Secara bersamaan sensor larik akan menghasilkan keluaran yang setara dengan integral dari cahaya yang di terima setiap sensor.[2]

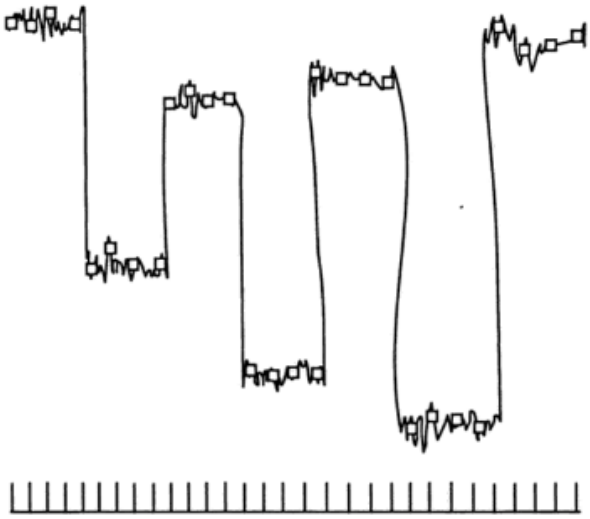
2.1.1.2 Sampling

Pada tahap ini adalah tahap dimana terjadinya proses digital pada koordinat x, y. Hasil dari sensor masih berupa citra kontinu yang merupakan fungsi kontinu f(x,y). Fungsi tersebut merupakan sinyal kontinu pada nilai x, y dan juga aplitudonya (intensitas). Nilai x, y yang kontinu akan diubah menjadi bentuk diskrit.



Gambar 2.2 Scan pada garis AB.

Proses sampling akan memberikan nilai koordinat diskrit pada setiap posisi di koordinat kontinu.



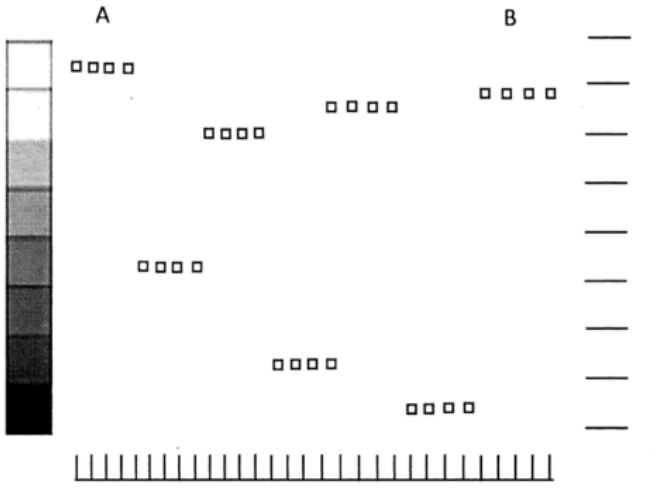
Gambar 2.3 Proses Sampling.

Koordinat diskrit ditandai dengan titik putih sepanjang sinyal analog.[2]

2.1.1.3 Kuantisasi

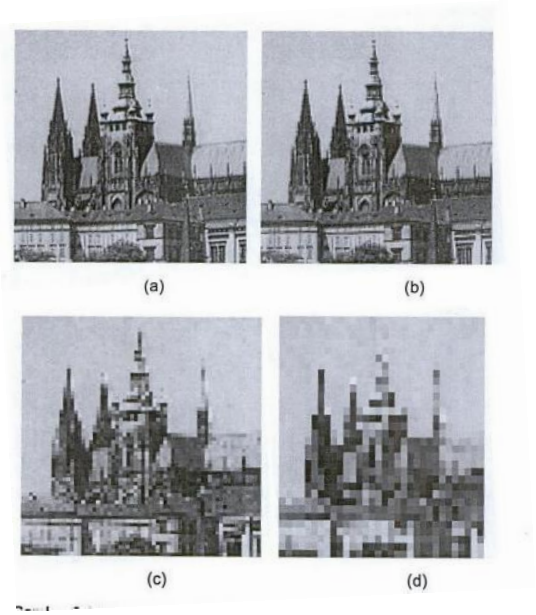
Proses kuantisasi adalah proses perubahan nilai amplitudo kontinu menjadi nilai baru yang berupa nilai diskrit. Nilai amplitudo yang dikuantisasi adalah nilai-nilai pada koordinat diskrit hasil proses sampling. Sinyal kontinu yang telah

dikuantisasi selanjutkan akan memiliki nilai dari 1 - 8 sesuai dengan nilai level digitalnya.[2]



Gambar 2.4 Hasil Sampling dan Kuantisasi

Ilustrasi perbedaan proses sampling dan kuantisasi di atas ditunjukan pada gambar berkut ini.



Gambar 2.5 Proses sampling, (a) Citra asli 256 x 256 pixel, (b) Sampling 128 x 128, (c) Sampling 64 x 64, (d) Sampling 32 x 32.

2.2 Gesture Recognition

Gesture recognition adalah ilmu komputer dan teknologi bahasa yang bertujuan untuk menggambarkan gerak tubuh manusia melalui algoritma matematis. Gerakan tersebut berasal dari gerak tubuh atau keadaan apapun akan tetapi pada umumnya berasal dari wajah dan tangan pengguna dapat berinteraksi dengan perangkat tanpa menyentuhnya secara fisik. *Gesture recognition* dapat diaplikasikan sebagai cara komputer untuk mengenali bahasa tubuh manusia sehingga antara komputer dengan pengguna sehingga membangun jembatan yang lebih mudah antara mesin dan manusia daripada antarmuka pengguna teks primitif atau bahkan GUI (antarmuka pengguna grafis), yang masih membatasi masukan pada *keyboard* dan *mouse*.

Dengan menggunakan konsep *gesture recognition*, pengguna mampu mengarahkan jari tangan ke layar komputer sehingga kursor akan bergerak sesuai dengan masukan yang di kenali oleh kamera. Ini bisa membuat perangkat input konvensional seperti *mouse*, *keyboard* dan bahkan layar sentuh menjadi tergantikan.k[23] Dalam *gesture recognition*, kamera membaca gerakan tubuh manusia dan mengkomunikasikan data ke komputer yang menggunakan isyarat sebagai masukan untuk mengendalikan perangkat atau aplikasi dengan demikian bertujuan menjadi salah satu cara yang digunakan adalah membantu orang yang mengalami gangguan fisik untuk berinteraksi dengan komputer, seperti menafsirkan bahasa isyarat.

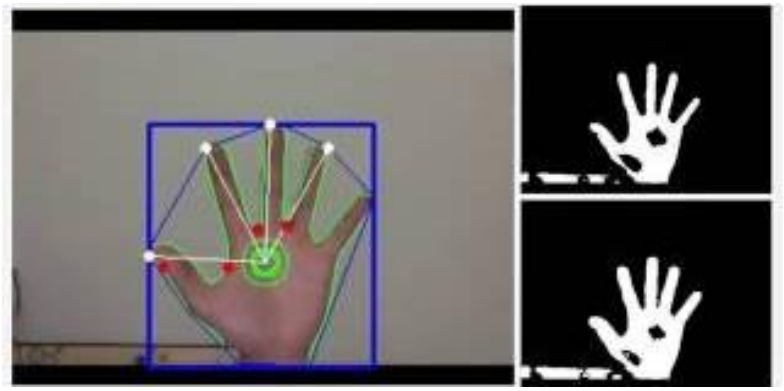
2.3 Metode Fingertip Tracking

Metode *fingertip tracking* merupakan tahapan yang akan dilakukan untuk melakukan pendekteksian jari tangan dan pelacakan jari tangan pengguna. Tahapan ini dibagi menjadi 2 yaitu *finger detection* dan *finger tracking*.

2.3.1 Finger Detection

Finger Detection adalah cara bagaimana agar dapat mengenali ibu jari dan jari telunjuk yang berbeda bentuk dan funsinya yang dilakukan oleh sistem berbasis *computer vision*. Untuk manusia melakukan pendeteksian objek sangatlah mudah tetapi merupakan tantangan tersendiri oleh *computer vision*.

Pada dasarnya pengembangan keilmuan *computer vision* adalah mengajarkan komputer untuk meniru bagaimana manusia melihat (*human vision*). Dengan cara mengumpulkan dan menafsirkan informasi visual dari lingkungan sekitarnya. Biasanya gambar pertama di proses pada tingkat yang lebih rendah untuk meningkatkan kualitas gambar, misalnya menghilangkan *noise*. Kemudian gambar di proses pada tingkat yang lebih tinggi misalnya mendeteksi pola atau bentuk, dan mencoba untuk menentukan apa yang ada di gambar.



Gambar 2.6 *Object Detection* (*Fingertp*).

Object Detection sering menggunakan metode yang bertanggung jawab untuk menemukan dan mengidentifikasikan keberadaan object dari kelas tertentu. Metode yang biasanya dipakai adalah menggunakan metode-metode deteksi tepi seperti Sobel, Prewitts atau Canni. Deteksi tepi dengan metode-metode tersebut sangat menguras sumber daya karena tingginya komputasi yang dilakukan picture element (*pixel*) demi *pixel*. Penggunaan deteksi tepi untuk pendeteksian objek tidak efektif pada objek bergerak seperti dalam format video dimana pergerakan *frame* dalam sebuah video dapat mencapai 20 *frame* per detik (fps). Metode deteksi tepi juga tidak efektif untuk diterapkan sebagai sistem pendeteksian objek pada *object tracking* atau *fast object tracking*. Oleh sebab itu membutuhkan metode-metode berikut untuk dapat mendeteksi objek yang bergerak.

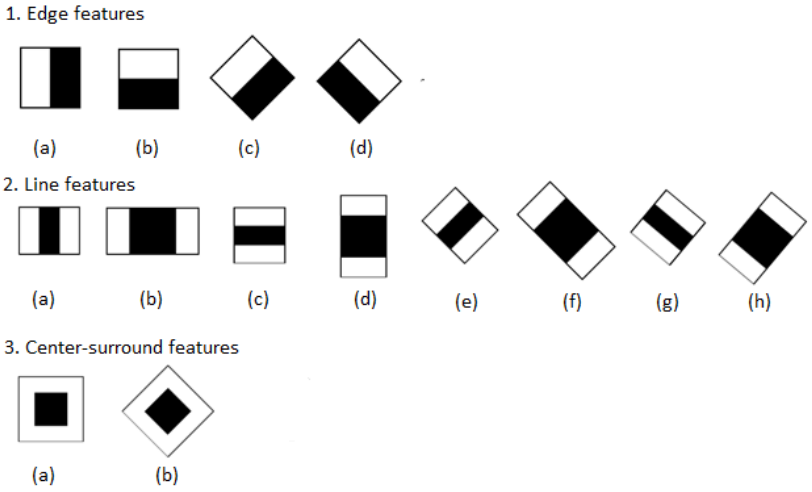
2.3.1.1 Haar Cascade Classifier

Secara umum, Haar-like Feature atau yang dikenal sebagai Haar Cascade Classifier digunakan dalam mendeteksi objek pada image digital. Nama *Haar* merujuk pada suatu fungsi matematika (Haar Wavelet) yang berbentuk kotak,

prinsipnya sama seperti pada fungsi Fourier. Awalnya pengolahan gambar hanya dengan melihat dari nilai RGB setiap *pixel*, namun metoda ini ternyata tidaklah efektif. Viola dan Jones kemudian mengembangkannya sehingga terbentuk Haar-like feature.

Haar-like feature merupakan *rectangular features* (fungsi persegi), yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. Prinsip Haar-like feature adalah mengenali objek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai *pixel* dari *image* objek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah *pixel* dalam persegi bukan setiap nilai *pixel* dari sebuah *image*.[3]

Haar-like feature memproses gambar dalam kotak-kotak dengan ukuran tertentu, dimana dalam satu kotak terdapat beberapa *pixel*. Per kotak itu pun kemudian di-proses dan didapatkan perbedaan nilai (*threshold*) yang menandakan daerah gelap dan terang. Sehinggan metode ini memerlukan data training yang berupa gambar berikut:



Gambar 2.7 Haar-like Feature

Tiga tipe kotak fitur:

a. Tipe *two-rectangle feature*
b. Tipe *three-rectangle feature*
c. Tipe *four-rectangel feature*

Pada gambar 2.7 terdapat 3 tipe kotak, dimana tipe pertama adalah *two-rectangle feature* yang terdiri dari 2 persegi, tipe kedua adalah *three-rectangle feature* yang terdiri dari 3 persegi dimana bagian tengah terdapat persegi berwarna hitam dan tipe yang terakhir *four-rectangle feature* yang terdiri dari 2 persegi dengan persegi berwarna hitam terdapat di dalam persegi putih. Tiap kotak-kotak digambar 2.7 mempunya nilai masing-masing yang diberi nama nilai Haar-like feature, yaitu perbedaan antara jumlah nilai-nilai *pixel gray level* dalam daerah kotak hitam dan daerah kotak putih. Nilai ini dapat dihitung dengan menggunakan persamaan berikut:

$$f(x) = sum_{black-rectangle}(pixel\ gray\ level) - sum_{white-rectangle}(pixel\ gray\ level) \quad (2.1)$$

2.3.1.2 Haar Classifier

Dari metode Haar-like feature dapat disebut sebagai metode yaitu meode Haar Classifier yaitu metode object detection yang membangun sebuah *boosted rejection cascade*, yang akan membuang data *training* negatif sehingga didapat suatu keputusan untuk menentukan data positif.

Haar Classifier merupakan metode *supervised learning*, yaitu membutuhkan data *trainning* untuk dapat mendeteksi objek-objek tertentu. Untuk itu Haar Classifier membutuhkan data postif (objek yang akan di deteksi) dan data negatif (bukan object yang di deteksi).

2.3.1.3 Cascade Classifier

Cascade classifier adalah sebuah rantai *stage classifier*, dimana setiap *stage classifier* digunakan untuk mendeteksi apakah di dalam *image sub-window* terdapat objek yang diinginkan (*object of interest*).[5]

Untuk mendapatkan nilai cascade, perlu dilakukan *training* cascade. Classifier dengan banyak fitur akan mendapat tingkat pendeteksian yang lebih tinggi serta error yang dibutuhkan untuk penghitungannya. Adapun atribu-atribut yang perlu diubah dan dicari nilai optimalnya adalah sebagai berikut:

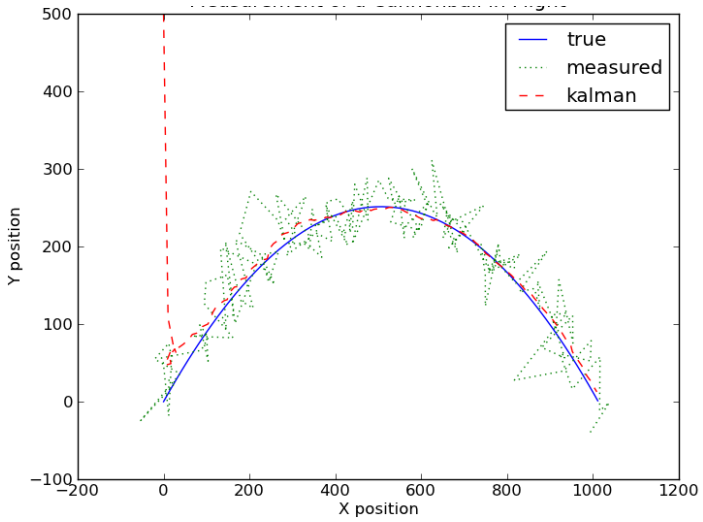
a. Jumlah *stage*,
b. Jumlah *feature*, dan

c. Nilai *threshold*.

Tiap *stage* mengurangi tingkat *false* positif serta meningkatkan tingkat pendeteksian. Sedangkan jumlah *stage* akan terus ditambahkan hingga target pendeteksian dan tingkat *false* positifnya ditemukan. Pada gambar diatas, dapat kita lihat bahwa tiap *sub-window* yang bernilai negatif (bukan objek yang dicari) akan langsung disisihkan dan secara otomatis tidak akan melanjutkan proses pencarian pada *sub-window* tersebut. sedangkan untuk *sub-window image* yang bernilai positif akan selalu dicek lagi dan lagi pada stage berikutnya hingga ditemukan objek yang dimaksud.[5]

2.3.2 Finger Tracking

Untuk melacak gerakan jari tangan menggunakan metode Kalman Filter. Secara singkat metode Kalman Filter mengasumsikan bahwa informasi terbaik diperolah dengan estimasi. Tujuannya adalah unuk menggunakan pengukuran yang diamati dari waktu ke waktu yang mengandung *noise* dan menghasilkan nilai yang cenderung lebih dekat dengan nilai sebenarnya. *Noise* merupakan efek samping dari penggunaan *sensor* kamera yang dipakai untuk mengumpulkan cahaya.



Gambar 2.8 O*bject Tracking* Menggunakan Kalman Filter

2.3.2.1 Kalman Filter

Kalman filter merupakan bagian dari Gaussian filter yang mengikuti dalil yang diberikan oleh Bayesian filter, ditemukan oleh Swerling (1958) dan Kalman (1960). Kalman filter menganggap bahwa objek bergerak pada distribusi Gauss atau normal. Dengan asumsi bahwa objek bergerak berdasarkan suatu persamaan linier dan memiliki variabel pengganggu yang diberikan secara acak, mengikuti aturan Gauss. Persamaan linier yang diberikan oleh Kalman filter dinyatakan dalam persamaan (2.2) dan (2.3).

$$x_t = A_t x_{t-1} + B_t u_{t-1} + \varepsilon_t \quad (2.2)$$

dimana $\mu_{t-1} = A_t x_{t-1} + B_t u_{t-1}$ adalah mean dan Σ_{t-1} kovarian dari ε_t

$$z_t = C_t x_t + \delta_t \quad (2.3)$$

Persamaan (2.2) menyatakan nilai state xt, dengan εt merupakan suatu variabel acak Gauss yang memiliki mean sama nol. At dan Bt merupakan konstanta yang diberikan. Kemudian untuk nilai observasi dinyatakan dalam persamaan (2.3), dengan δt merupakan suatu variabel acak Gauss yang memiliki mean sama nol. Ct merupakan konstanta yang diberikan. Kalman merepresentasikan nilai kepercayaan kedalam mean (μ) dan kovarian (Σ), sehingga untuk hitung nilai kepercayaan dapat dilakukan dengan cara yang serupa pada Bayes filter.

Hasil dari Kalman filter adalah nilai mean dan kovarian yang jika dilakukan proses rekursif, yaitu menyatakan kembali hasil dari Kalman filter sebagai nilai awal (μ_{t-1}, Σ_{t-1}) maka, akan membentuk A_t dan B_t yang baru, begitu juga dengan C_t. Sehingga dalam algoritma pelacakan kalman filter dinyatakan:

Algoritma Pelacakan_kalmanFilter

while(true)

$$\overline{\mu}_t = A_t \mu_t + B_t \mu_t$$

$$z_t = C_t x_t + \delta_t$$

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) bel(x_{y-1}) dx_{t-1}$$

$$bel(x_t) = \eta p(z_t|x_t)\overline{bel}(x_t)$$

$$Kalman_filter(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$$

endWhile.

2.4 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.[10]

2.4.1 Jenis-jenis *Video Game*

Berikut ini beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya dan mesin yang menjalankannya. Jenis-jenis game tersebut adalah: [11]

2.4.1.1 *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 2.9 Contoh *Game Pc* (Fronite Battle Royal)

2.4.1.2 *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 2.10 Perangkat *Game Console*



Gambar 2.11 Contoh-Contoh *Game Console* (Final Fantasy XV, Fifa 2017 dan Zelda)

2.4.1.3 *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 2.12 Perangkat *Game Arcade*



Gambar 2.13 Contoh *Game Arcade* (King Of Fighter)

2.4.1.4 *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 2.14 Contoh-Contoh Game Online (Dota 2 Dan Clash Of Clans).

2.4.2 Game Rhythm

Game rhythm atau disebut *game* ritme adalah *game* aksi bertema musik yang dimainkan dengan cara menuntun pemain menekan tombol yang disediakan sesuai dengan irama serta tempo dari musik yang dipilih di dalam *game* yang dimainkan untuk mencapai skor tertinggi.

Di beberapa *game* ritme terdapat *mode online* yang dapat dimanfaatkan untuk bermain secara *multiplayer* dengan pemain lain. Untuk memain *game* ini dapat menggunakan *gamepad* atau *joystick* yang digunakan sebagai inputan dalam bermain *game* ini.

2.5 Game Controller

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

2.5.1 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 2.15 *Joystick*.

2.5.2 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 2.16 *Mechanical Keyboard*

2.5.3 Mouse Macro

Mouse adalah alat/perangkat yang digunakan untuk mengatur perpindahan kursor atau memberikan perintah pada komputer. *Mouse Macro* adalah *mouse* yang memiliki fungsi sama dengan *mouse* biasa namun diciptakan khusus untuk *gaming*.

Karena memiliki fitur tombol tambahan yang berguna untuk memindahkan beberapa fungsi tombol *keyboard* ke dalam *mouse macro* sehingga mempermudah *gamers* dalam bermain *game*. *Mouse macro* juga biasanya dipakai untuk sejenis *game* ber-*genre* RPG dan FPS. Berikut adalah gambar bentuk dari *mouse macro*.



Gambar 2.17 *Mouse Macro*

2.6 UML (*Unified Modeling Language*)

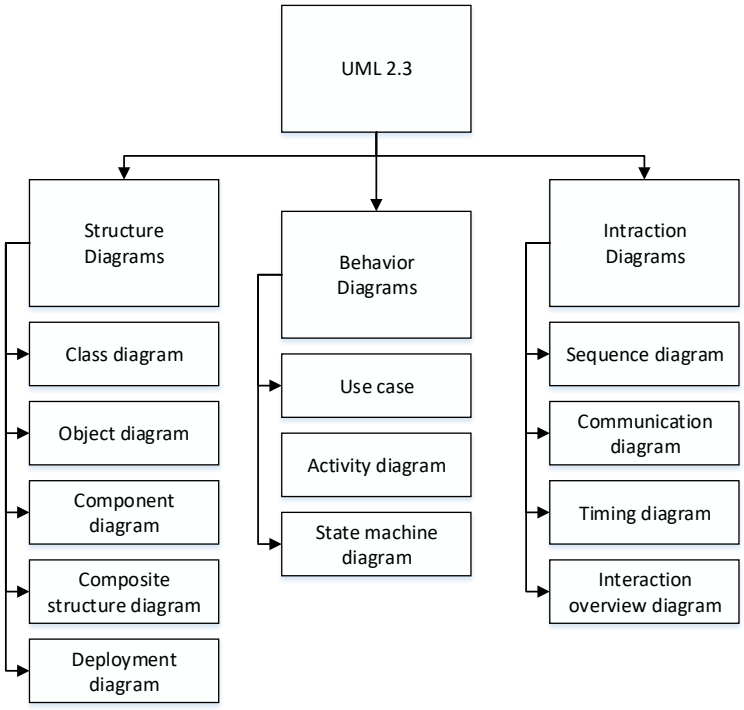
UML (*Unified Modelling Language*) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual.[21] Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.[22]

Pada versi UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori.[12] Berikut ini adalah pembagian berdasarkan pengelompokannya masing-masing:

1. *Stucture* diagrams yaitu kumpulan diagram yang menggambarkan stuktur dari sebuah sistem yang dibangun atau dikembangkan
2. *Behavior* diagrams yaitu kumpulan diagram yang menggambarkan tingkah laku sistem terhadap ruang lingkup sistemnya.

3. *Interaction* diagrams yaitu kumpulan diagram yang menggambarkan interkasi sistem satu dengan lainnya.

Gambar berikut adalah diagram dari pengelompokkan UML:



Gambar 2.18 Diagram UML [13]

2.6.1 Use Case

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendiskripsikan interaksi tipikaal antara para pengguna sistem dengan sistem itu sendiri, sengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. Terdapat dua entitas penting didalam *use case* disebut aktor dan *use case*.

2.6.2 Activity Diagram

Diagram aktivitas adalah diagram yang menggambarkan bagaimana sistem bekerja. Diagram aktivitas menggambarkan bagaimana sistem bekerja, bukan aktor yang melakukan.

2.6.3 Sequence Diagram

Diagram sekuen menggambarkan proses dimana proses tersebut mendeskripsikan waktu hidup dan pesan yang dikirimkan dan diterima antar objek.[13] Banyaknya diagram sekuen harus sesuai dengan banyak pendefinisian *use case*.

2.6.4 Class Diagram

Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sebuah sistem. Dalam sebuah kelas terdapat sebuah stuktur sebagai berikut:

1. Kelas *Main*

   Kelas *Main* berfungsi sebagai kelas yang mengeksekusi sistem ketika sistem pertama kali dijalankan.
2. Kelas *View*

   Kelas *View* berfungsi sebagai kelas yang mengatur sebuah tampilan.
3. Kelas *Controller*

   Kelas *Controller* mengangani proses bisnis yang bekerja disebuah sistem.
4. Kelas Model

   Kelas model berfungsi sebagai pembungkus data menjadi sebuah kesatuan yang diambil maupun yang akan disimpan.

2.7 Unity 3D

Unity 3D adalah software yang digunakan untuk membuat *video game* berbasis dua atau tiga dimensi dan dapat digunakan secara gratis, selain untuk membuat *game*, Unity 3D juga dapat digunakan untuk membuat konten yang interaktif lainnya seperti, visual arsitektur dan *real-time* 3D animasi, selain sebagai *game engine* Unity 3D juga dapat digunakan sebagai sebuah editor bagi *game* yang sudah ada.

Unity 3D dibuat dengan menggunakan bahasa perogram C++, tapi pengguna tidak perlu menggunakan bahasa C++ yang sulit, karena Unity 3D mendukung bahasa program lain seperti JavaScript, C#, dan Boo, Unity memiliki

kemiripan dengan *game engine* lainnya seperti, Blender *game engine*, Virtools, Gamestudio. Adapun kelebihan dari Unity 3D, Unity dapat dioperasikan pada *platform* Windows dan Mac Os dan dapat menghasilkan *game* untuk Windows, Mac, Linux, Wii, iPad, iPhone, google Android dan juga browser. Untuk browser, kita memerlukan sebuah *plugin*, yaitu Unity Web Player, sama halnya dengan *Flash Player* pada *Browser*. *Game* Unity 3D juga mendukung dalam pembuatan *game* untuk *console game* Xbox 360 dan PlayStation 3.

2.7.1 Sejarah Unity

Unity Technologies ditemukan oleh David Helgason, Nicholas Francis, dan Joachim Ante pada tahun 2004.[14] Pada tahun 2005 Unity merilis versi 1 di Apple's WWDC, tahun 2007 unity merilis versi 2.0 dan mengadakan acara unity tahunan konferensi developer, tahun 2008 unity versi iphone telah dirilis dan unity telah resmi menjadi penyedia *game* untuk Wii dan Cartoon Network memperkenalkan FusionFall dimana pengguna yang memainkan *game* tersebut lebih dari 8 juta orang.[15]

Pada tahun 2009 Unity merilis versi gratis di konferensi developer unity dan pada tahun yang sama unity mendapatkan pendanaan dari Sequoia Capital sebesar 5.5 juta dollar. Tahun 2010 Unity merilis versi 3 dimana terdapat unity asset store, unity untuk android developer dan membuat Naci (Unity native client) sebagai plugin chrome. Tahun 2011 Unity telah memiliki 100 karyawan dan membuka cabang diberbagai negara yaitu Kanda, Jepang, Korea dan Swedia, ditahun yang sama unity bekerjasama dengan Sony Ericsson membuat sebuah game di Xperia Play dan unity mendapatkan kembali pendanaan sebesar 12 juta dollar.

Unity merilis preview unity versi linux dan unity versi 4 pada tahun 2012 dan merilis unity 3.5 dengan berbagai fitur baru yaitu mekanik animasi yang memudahkan animator untuk menggerakan karakter dan pada tahun yang sama unity bekerjasama dengan adobe menjadi patner flash player. Tahun 2013 unity mengumumkan telah memiliki fitur 2D, unity cloud, mendukung blackberry 10 dan mendukung platform xbox dan tahun 2014 unity merilis unity versi 5 dengan

berbagai fitur terbarukan yaitu sound mixer, texture ligthing, unity ads dan lainnya.

2.8 Library OpenCV

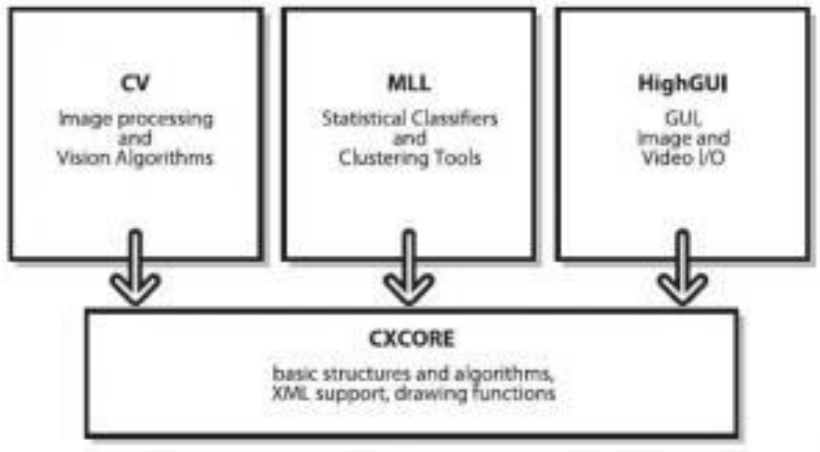
OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) *Library* yang sudah sangat familiar pada Pengolahan Citra *Computer Vision*. *Computer Vision* itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer 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.[9]

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 2.19 Struktur dan Konten OpenCV

2.9 Pemograman C#

Pemograman C# (C-Sharp) adalah sebuah bahasa pemograman yang berorientasi objek yang di kembangkan oleh Microsoft dan menjadi salah satu bahasa pemograman yang mendukung .NET programing melalui Visual Studio.[16] Bahasa ini diciptakan oleh designer Microsoft dikembangkan dibawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan Borland Delphi.[7]

C# merupakan bahasa pemrograman yang berorientasi objek (object-oriented) yang merepresentasikan konsep "objek" dengan suatu data *field* yang mempunyai suatu prosedur perintah *method*. Melalui konsep object-oriented ini aplikasi bisa dibangun dengan menggunakan beberapa objek yang akan saling berinteraksi satu dengan yang lainnya.

1. Class

   Class adalah sebuah template yang mendefinisikan bentuk suatu objek.[17] Data dan kode dispesifikasikan untuk membentuk sebuah objek. *Class* menjadi acuan deksripsi untuk membentuk sebuah objek.

2. Object

   Object adalah perumpamaan dari sebuah kelas, *Object* dapat mengakses attribut yang terdapat di *Class* sehingga attribut yang seharusnya ditulis berulang ulang dapat diakses kembali dengan properti yang sama.