BAB II
TEORI PENUNJANG

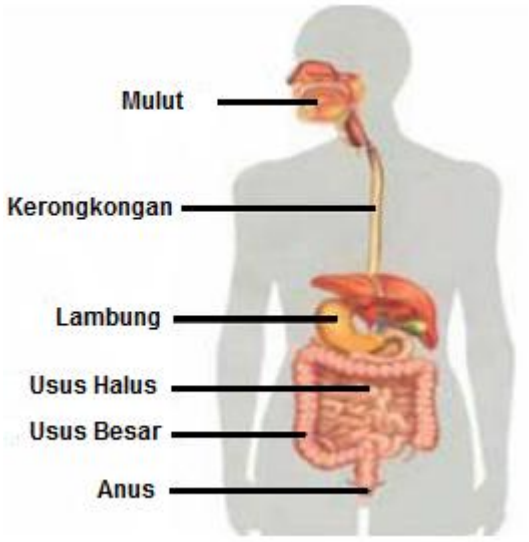
2.1 Organ Tubuh Manusia

Tubuh manusia terdiri dari satu sistem yang didalamnya terdiri atas organ-organ. Organ tubuh manusia tersebut terdiri atas berbagai macam jaringan yang dibangun atas sel-sel [1].

Organ adalah bagian tubuh yang tersusun dari jaringan-jaringan. Organ memiliki fungsinya masing-masing [1].

2.1.1 Organ Pencernaan

Organ pencernaan adalah alat penunjang sistem pencernaan pada manusia. Sistem pencernaan pada manusia terdiri atas beberapa organ yang berawal dari mulut, kerongkongan, lambung, usus halus, usus besar dan anus [1]. Penjelasan tersebut tampak pada Gambar 2.1.

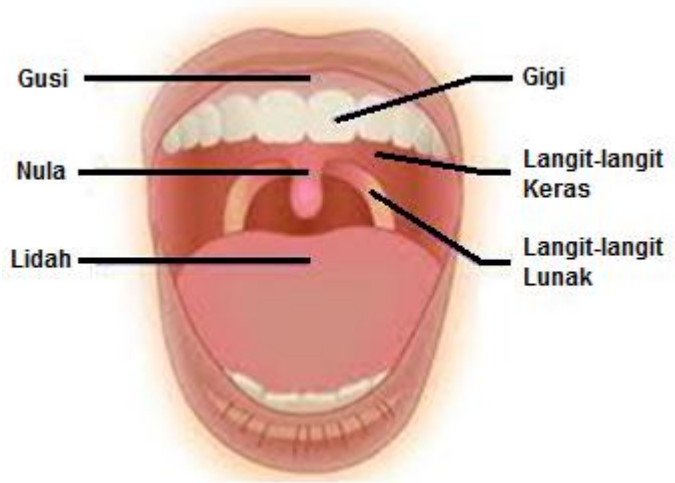


Gambar 2.1 *Sistem pencernaan manusia [1]*

2.1.1.1 Mulut

Mulut merupakan organ pencernaan yang pertama yang bertugas dalam proses pencernaan makanan. Di dalam mulut makanan akan dicerna secara sadar menjadi bagian-bagian kecil. Di dalam rongga mulut terdapat gigi, gusi, nula, lidah, langit-langit lunak, dan langit-langit keras [1].

Penjelasan tersebut tampak pada Gambar 2.2.

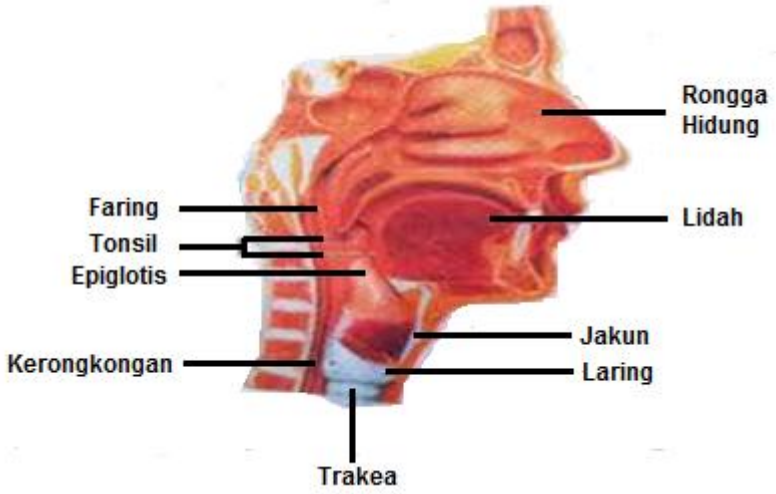


Gambar 2.2 *Organ pencernaan mulut [1]*

2.1.1.2 Kerongkongan

Kerongkongan adalah tabung yang menghubungkan faring (tenggorokan) dengan lambung. Kerongkongan terletak di antara trakea (pipa udara) dan tulang belakang, melewati bawah leher, menembus diafragma sedikit di sebelah kiri garis tengah, dan bergabung dengan ujung atas perut. Pada orang dewasa, kerongkongan memiliki panjang sekitar 25 cm. Ketika seseorang menelan, otot dinding kerongkongan berkontraksi untuk mendorong makanan ke dalam lambung. Kelenjar lapisan lendir kerongkongan menjaga lorong tetap lembab dan memudahkan menelan [1].

Penjelasan tersebut tampak pada Gambar 2.3.



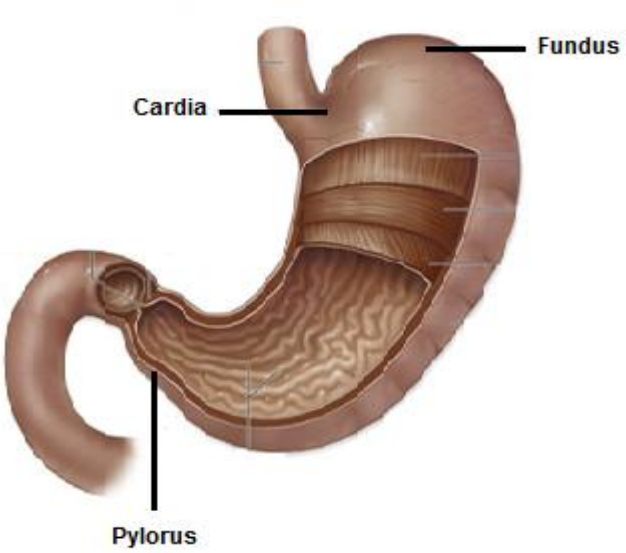
Gambar 2.3 *Organ pencernaan kerongkongan [1]*

2.1.1.3 Lambung

Lambung merupakan suatu kantung yang terletak di dalam rongga perut sebelah kiri atas. Lambung mengeluarkan getah lambung yang mengandung: *pepsine*, berguna mengubah zat putih telur menjadi *pepton*. *Asam klorida (HCL)*, yang berguna untuk membunuh bibit penyakit yang masuk bersama makanan. *Rennin* berguna untuk mengubah susu menjadi *kasein* [2]. Lambung terdiri dari tiga bagian yaitu:

1. *Kardia*, lambung bagian atas, daerah pintu masuknya makanan dari kerongkongan itu sendiri.
2. *Fundus*, lambung bagian tengah dan bentuknya membulat.
3. *Pilorus*, lambung bagian bawah, lambung bagian ini adalah daerah yang berhubungan dengan usus dua belas jari.

Penjelasan tersebut tampak pada Gambar 2.4.



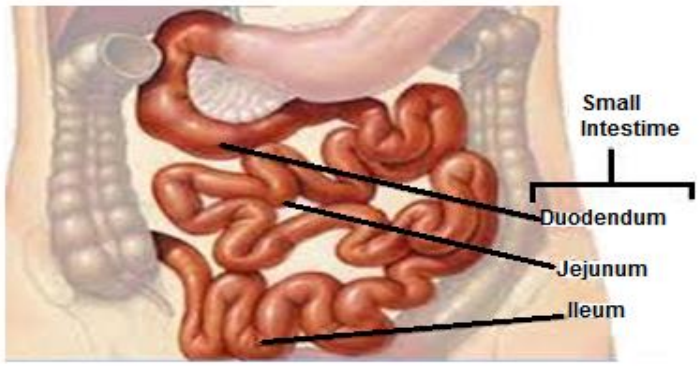
Gambar 2.4 *Organ pencernaan lambung [2]*

2.1.1.4 Usus Halus

Usus halus adalah untuk bertanggung jawab menyerap sebagian besar nutrisi yang ditemukan dalam makanan. Pada saat makanan dicerna mencapai usus halus, telah secara mekanis dipecah menjadi cairan. Usus halus terbagi menjadi tiga bagian yaitu usus dua belas jari (*duodenum*), usus kosong (*jejunum*), dan usus penyerapan (*ileum*) [1].

Usus dua belas jari dan usus kosong berperan penting dalam pencernaan makanan secara kimiawi. Pencernaan makanan berakhir di *ileum*. Disini makanan yang telah dicerna akan diserap dinding *ileum* [1].

Penjelasan tersebut tampak pada Gambar 2.5.

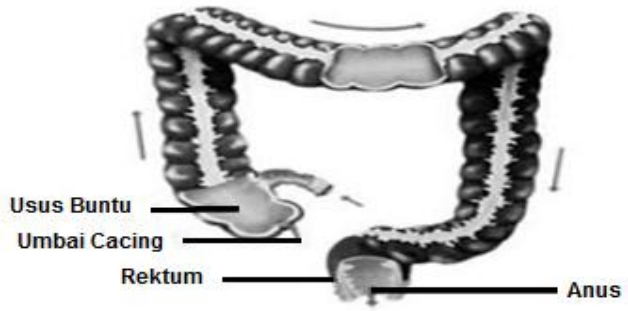


Gambar 2.5 *Organ pencernaan usus halus [1]*

2.1.1.5 Usus Besar dan Anus

Usus besar berfungsi zat-zat yang tidak dapat diserap di usus halus selanjutnya akan masuk ke usus besar, di usus besar ini terjadi penyerapan air dan pembusukan sisa-sisa makanan oleh bakteri pembusuk. Pembusukan dilakukan oleh bakteri yang hidup di anus. Akhirnya sisa makanan akan dikeluarkan dalam bentuk kotoran *(feces)* melalui anus [1].

Penjelasan tersebut tampak pada Gambar 2.6.



Gambar 2.6 *Organ pencernaan usus besar [1]*

2.1.2 Organ Vital

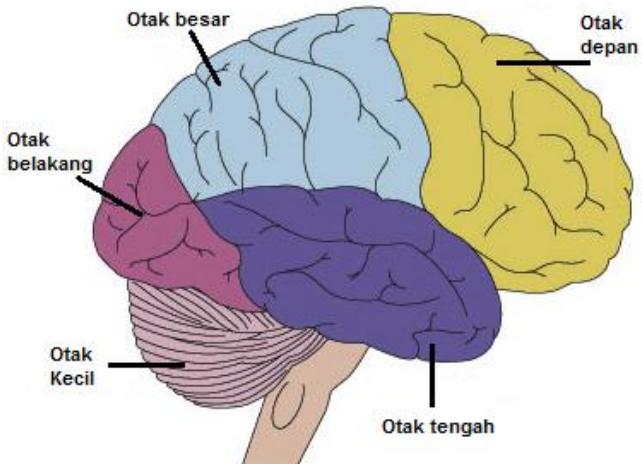
Organ vital pada manusia terdiri dari otak, jantung, paru-paru, hati, dan ginjal. Jika salah satu organ tubuh vital tersebut mengalami gangguan atau penurunan fungsi, maka akan mempengaruhi kinerja organ tubuh lainnya. Organ vital tersebut bekerja dengan bantuan jaringan organ lainnya [1].

2.1.2.1 Otak

Otak adalah pusat dari sistem saraf yang memiliki fungsi mengatur gerakan, perilaku, dan fungsi tubuh *homeostatis*, melatih emosi, memori/ingatan, dan juga motorik [2]. Otak manusia terdiri atas lima bagian yaitu:

1. Otak depan
2. Otak besar
3. Otak tengah
4. Otak belakang
5. Otak kecil

Penjelasan tersebut tampak pada Gambar 2.7.



Gambar 2.7 *Organ vital otak [2]*

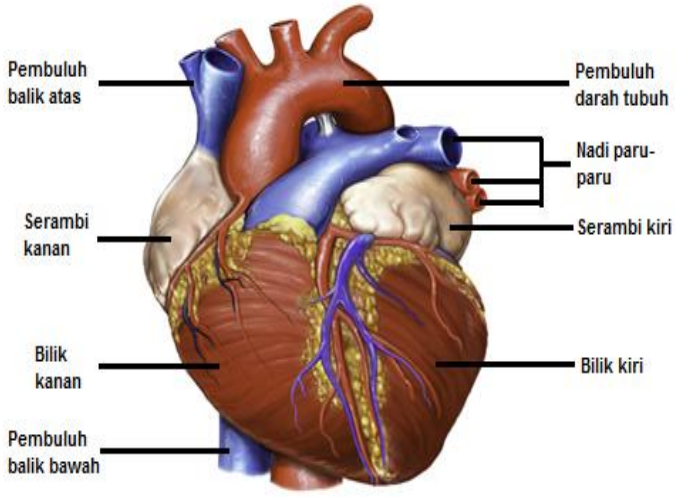
2.1.2.2 Jantung

Jantung merupakan salah satu organ tubuh manusia yang sangat penting, tersusun atas otot-otot jantung yang bekerja diluar kesadaran kita. Jantung berfungsi sebagai pemompa darah ke seluruh tubuh yang mempunyai letak dan bagian-bagian sebagai berikut [2].

a. Jantung terletak di dalam rongga dada aga sedikit ke kiri dan terbungkus kantong pelindung yang disebut pericardium.
b. Jatung besarnya kira-kira sebesar kepalan tangan.
c. Jantung terdiri dari empat ruangan, dua ruangan atas *(serambi)* dan dua ruangan bawah *(bilik)*
d. Ruangan atas sebelah kiri disebut *serambi* kiri kanan *(atrium kiri)* dan sebelah kanan disebut *serambi* kanan *(atrium kanan)*.
e. Ruangan bawah jantung sebelah kanan disebut bilik kanan *(ventrikel kanan)* dan sebelah kiri disebut bilik kiri *(ventrikel kiri)*.
f. Pada jantung terdapat suatu celah berpintu diantara serambi dan bilik yang disebut katup jantung. *Katup* jantung berfungsi sebagai penyekat agar darah dari bilik dan dari serambi tidak tercampur.
g. Ruangan atas *(serambi)* menerima darah yang kembali ke jantung dari pembuluh balik besar.

h. Bilik kanan berfungsi memompa darah ke paru-paru, sedangkan bilik kiri memompa darah ke seluruh tubuh selain paru-paru.
i. Peredaran darah kecil adalah peredaran darah dari jantung ke paru-paru kembali ke jantung.

Penjelasan tersebut tampak pada Gambar 2.8.



Gambar 2.8 *Organ vital jantung [2]*

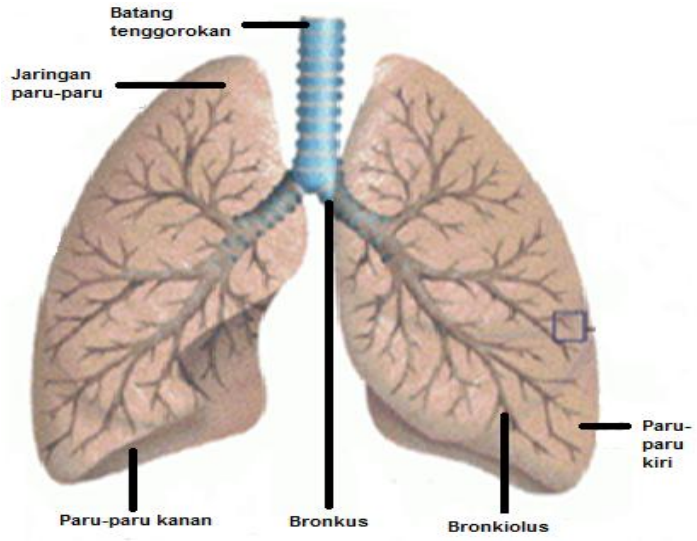
2.1.2.3 Paru- paru

Selain jantung, paru-paru juga merupakan organ tubuh yang sangat penting sebagai alat pernapasan. Manusia memiliki sepasang paru-paru yang terletak di dalam rongga dada. Di dalam paru-paru, saluran pernapasan bercabang-cabang dan tiap-tiap ujungnya berakhir pada gugus kantong udara. Pada kantong-kantong udara inilah tempat pertukaran *gas oksigen* (O_2) dan *karbondioksida* (CO_2) [2].

a. Paru-paru terbungkus oleh selaput tipis dan licin yang disebut *pleura*.
b. Selaput licin memungkinkan paru-paru dapat bergerak pada waktu kita bernapas, tanpa menimbulkan kerusakan kedua paru-paru itu.
c. Semua sel tubuh kita membutuhkan *oksigen* untuk hidup dan tumbuh.
d. *Oksigen* diperoleh dari udara pada waktu bernapas.
e. Paru-paru bertugas menyerap oksigen.
f. *Oksigen* dari udara dipindahkan ke dalam darah dan diangkut ke seluruh tubuh.

g. Gas yang dilepaskan dari proses pernapasan manusia adalah *karbondioksida (CO_2)*.
h. Udara yang masuk melalui hidung mengakibatkan manusia dapat mencium bau.
i. Aliran udara yang melalui tenggorokan digunakan untuk menghasilkan suara, sehingga kita dapat berbicara.
j. Pernapasan menghasilkan sejumlah air dalam sistem tubuh untuk membantu mengurangi panas tubuh.

Penjelasan tersebut tampak pada Gambar 2.9.



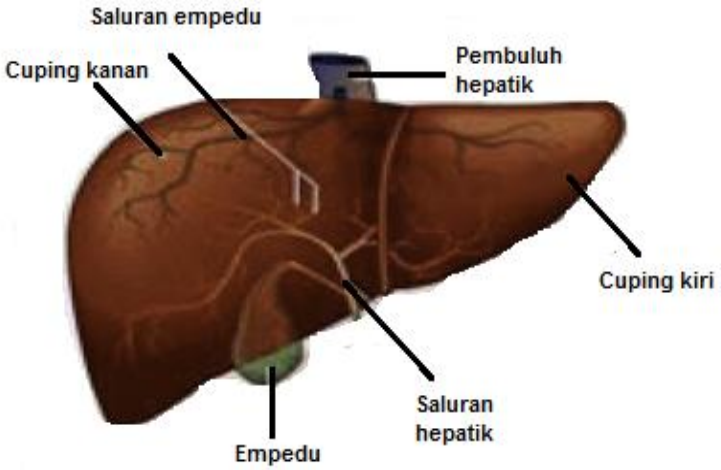
Gambar 2.9 *Organ vital paru-paru [2]*

2.1.2.4 Hati

Hati sebagai salah satu organ tubuh manusia yang terletak di rongga perut sebelah kanan, tepat dibawah rongga dada, pada orang dewasa beratnya kira-kira dua *kilogram* dan berwarna kemerah-merahan [2]. Sebagai organ penting, hati mempunyai fungsi sebagai berikut:

a. Hati sebagai penawar racun yang masuk ke tubuh kita.
b. Menghasilkan cairan *empedu* yang penting untuk proses pencernaan. Cairan *empedu* berwarna cokelat kekuningan. Kurang lebih setengah liter cairan *empedu* setiap hari dihasilkan oleh hati.

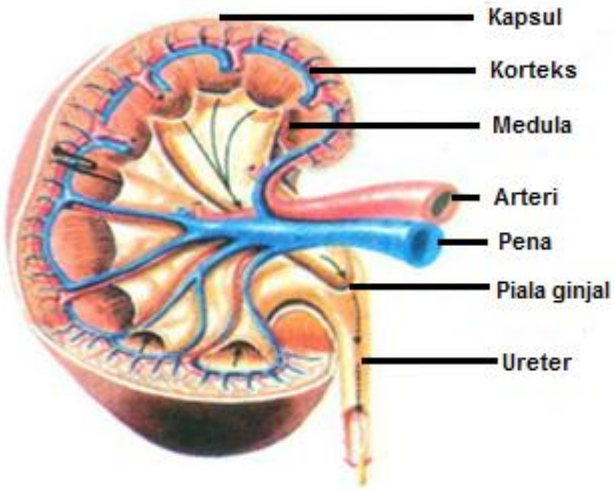
Penjelasan tersebut tampak pada Gambar 2.10.



Gambar 2.10 *Organ vital hati [2]*

2.1.2.5 Ginjal

Ginjal adalah organ utama yang bertugas membuang racun yang larut dalam air. Ginjal terdiri dari pembuluh darah *arteri* dan *vena interlobular, renal korteks, renal medulla, arteri* dan *vena interlobar, renal arteri, renal pelvis,* dan *kortex* [2]. Penjelasan tersebut tampak pada Gambar 2.11.



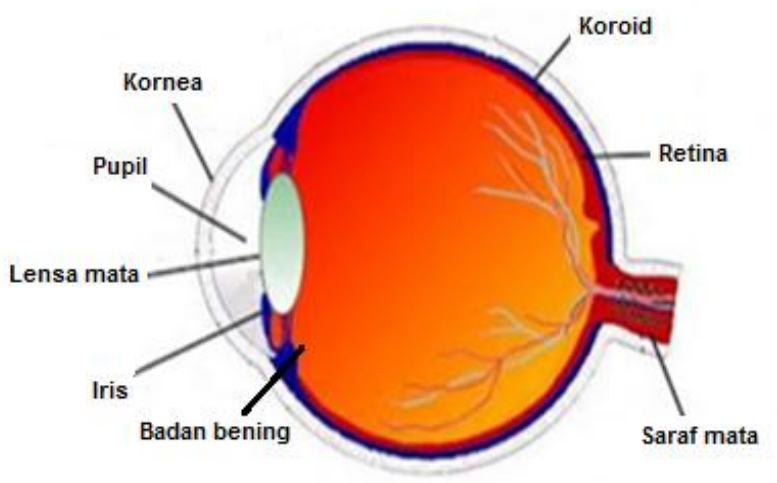
Gambar 2.11 *Organ vital ginjal [2]*

2.1.3 Alat Indera

Makhluk hidup mempunyai ciri-ciri yang salah satunya adalah peka terhadap rangsangan. Manusia memiliki alat untuk menerima rangsangan yang datang dari luar. Alat ini disebut alat indera. Berbagai rangsangan diterima oleh alat indera kita lalu disampaikan ke otak. Rangsangan tertentu ditanggapi oleh indera tertentu, misalnya cahaya ditanggapi oleh indera penglihatan dan seterusnya. Kita mempunyai lima alat indera, yaitu indera penglihatan (mata), indera pendengaran (telinga), indera pembau (hidung), indera pengecap (lidah), serta indera peraba (kulit) [2]. Semua itu disebut panca indera. Berikut penjelasan dari masing-masing indera:

2.1.3.1 Indera Penglihat (Mata)

Mata merupakan salah satu alat indra yang tak ternilai harganya. Mata terdiri atas bola mata yang terletak di dalam lekuk mata. Dalam bola mata terdiri atas bagian-bagian penting dalam penglihatan, yaitu: *selaput pelangi (iris), pupil, selaput tanduk (kornea), lensa mata, badan bening, selaput jala (retina), saraf mata* [2]. Penjelasan tersebut tampak pada Gambar 2.12.



Gambar 2.12 *Organ mata [2]*

Berikut penjelasan dari bagian-bagian mata:

a. *Selaput pelangi (iris)*
    1) *Selaput pelangi* memberikan pola warna pada mata kita.

2) Bagian tengah *iris* terdapat anak mata *(pupil)* yang berfungsi mengatur cahaya yang masuk ke mata.
3) *Pupil* akan mengecil jika cahaya yang diterima mata terlalu banyak, dan akan membesar jika cahaya yang diterima terlalu sedikit.

b. *Selaput tanduk (kornea)*
1) *Kornea* bersifat transparan dan tidak berpembuluh darah.
2) Berfungsi meneruskan cahaya yang masuk mata ke bagian *retina*.

c. *Lensa mata*
1) Bersifat *elastic*, dapat mencembung dan memipih.
2) Berfungsi memusatkan cahaya yang masuk ke mata agar bisa jatuh tepat di *retina*.

d. *Badan bening*
1) Merupakan bagian paling luas, tepatnya di belakang *lensa*.
2) *Badan bening* berisi *zat* seperti agar-agar.
3) Berfungsi meneruskan cahaya dari lensa mata ke *retina*.

e. *Selaput jala mata (retina)*
1) Bersifat sangat peka terhadap cahaya yang masuk ke mata, karena mempunyai sel-sel saraf penerima.
2) Berfungsi meneruskan rangsangan cahaya ke saraf mata.

f. *Saraf mata (saraf optik)*
1) Berfungsi meneruskan rangsangan cahaya yang diterimanya kesusunan saraf pusat di otak
2) Setelah rangsangan diterima otak, barulah kita bisa melihat.
3) *Kelenjar* air mata, menghasilkan air mata yang berguna untuk membasahi *kornea*.

2.1.3.2 Indera Pendengar (Telinga)

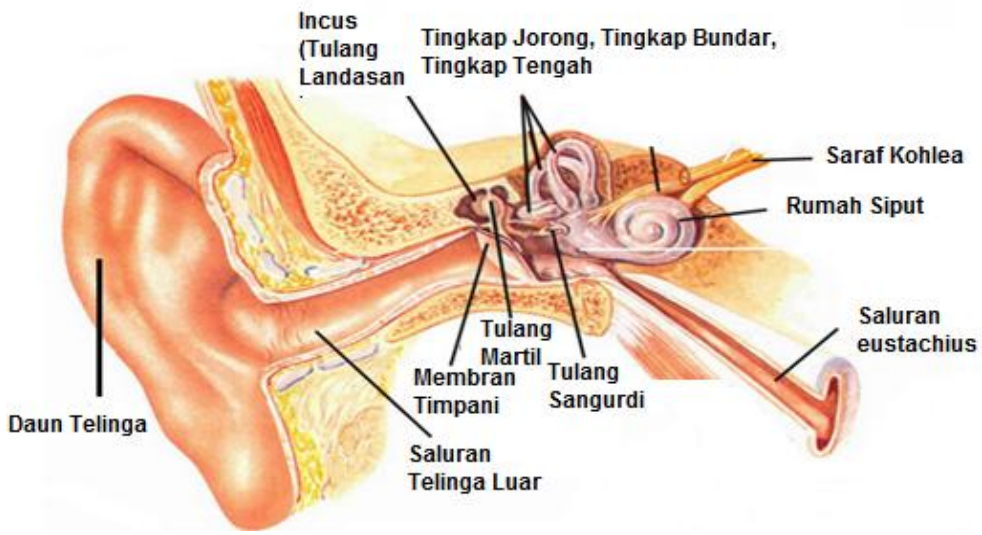
Kita dapat mendengar suara radio, suara music, dan suara kicau burung karena kita mempunyai alat pendengaran yaitu telinga. Telinga terdiri tiga bagian, yaitu telinga luar, telinga tengah, dan telinga dalam [2]. Berikut penjelasan dari bagian-bagian telinga:

a. Telinga Luar, terdiri dari:

1) Terdiri dari daun telinga dan saluran pendengaran.
2) Berfungsi sebagai penangkap getar suara.

b. Telinga Tengah, terdiri dari:
1) Gendang telinga, berfungsi menangkap getaran suara untuk diteruskan ke tulang-tulang pendengaran.
2) Tulang-tulang pendengaran, menerima getaran dari gendang telinga untuk diteruskan ke telinga dalam.
3) Saluran *Eustachius*, sebagai penghubung ruang telinga dan rongga mulut.

c. Telinga Dalam, terdiri dari:
1) *Tingkap jorong* dan rumah siput. Di dalam rumah siput ada cairan *limfa*. Getaran dari telinga tengah diterima *tingkap jorong* dan *rumah siput*. Cairan pada *rumah siput* ikut bergetar.
2) Saraf pendengaran, getaran rumah siput merangsang ujung-ujung saraf untuk diteruskan ke otak [2]. Sehingga kita dapat mendengar suara. Penjelasan tersebut tampak pada Gambar 2.13.



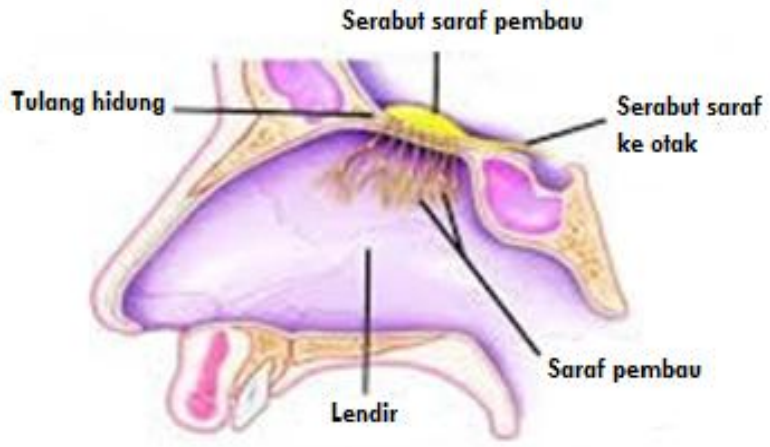
Gambar 2.13 *Bagian-bagian telinga [2]*

2.1.3.3 Indera Pembau (Hidung)

Hidung selain untuk jalan pernapasan juga sebagai indera pembau. Bau yang harum dan busuk dapat tercium jika *uap zat* yang berbau sampai ke dalam hidung [2].

a. Rangsangan yang ditimbulkan *uap zat* tersebut diterima oleh ujung-ujung urat saraf yang merupakan indera pembau. Rangsangan bau tersebut diteruskan ke otak, sehingga kita dapat mencium bau.
b. Indera pembau ini terletak pada *selaput lendir* di rongga hidung bagian atas, pada *kerang hidung* atas dan pada permukaan atas kerang hidung tengah.
c. Di dalam rongga hidung terdapat bulu hidung dan *selaput lendir*. Bulu hidung dan *selaput lendir* berguna untuk menahan kotoran yang terbawa udara yang kita hirup.

   Penjelasan tersebut tampak pada Gambar 2.14.



Gambar 2.14 *Bagian-bagian hidung [2]*

2.1.3.4 Indera Pengecap (Lidah)

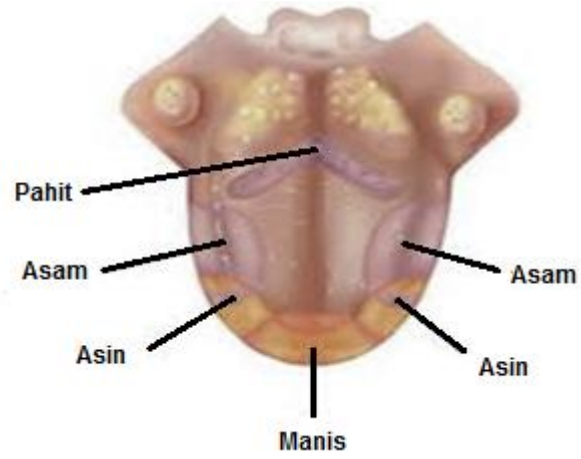
Apabila kita makan atau minum, kita dapat merasakan makanan atau minuman itu. Hal ini dikarenakan manusia memiliki indera pengecap yaitu lidah. Berikut penjelasan mengenai indera pengecap [2].

a. Permukaan lidah yang kasar karena penuh *bintil-bintil*. Pada bintil-bintil itu terdapat ujung-ujung saraf pengecap. Makanan dan minuman di

dalam mulut kita, akan merangsang ujung-ujung saraf pengecap dan rangsangan itu diteruskan ke otak. Oleh karena itu, kita dapat mengecap (merasakan) makanan atau minuman.

b. Selain untuk mengecap, lidah berguna untuk mengatur agar makanan di dalam mulut tercampur dengan air liur terkunyah dengan sebaik-baiknya.
c. Bagian-bagian lidah yang peka terhadap rasa tertentu adalah sebagai berikut:
    1) Ujung lidah peka terhadap rasa manis.
    2) Pangkal lidah peka terhadap rasa pahit.
    3) Tepi lidah peka terhadap rasa asam.
    4) Bagian ujung lidah samapai aga ke samping peka terhadap rasa asin.

Penjelasan tersebut tampak pada Gambar 2.15.



Gambar 2.15 *Bagian-bagian lidah [2]*

2.1.3.5 Indera Peraba (Kulit)

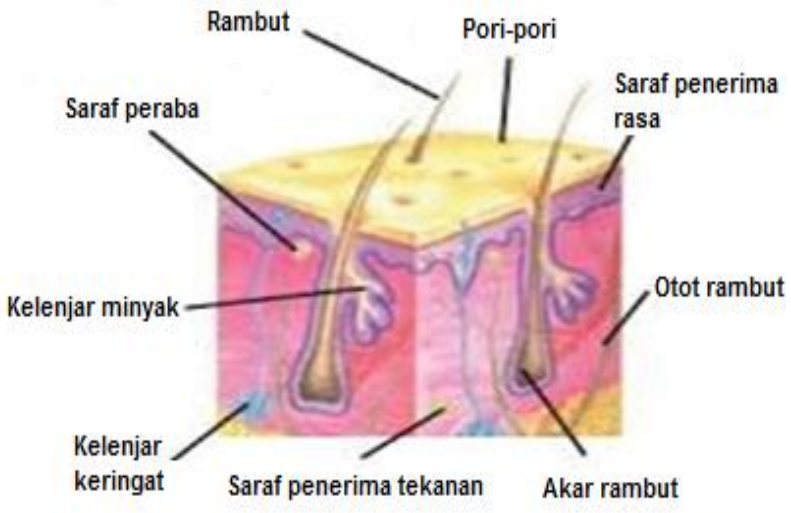
Tubuh kita seluruhnya dilapisi oleh kulit. Selain berfungsi sebagai pelindung, kulit juga berfungsi sebagai indera peraba [2]. Berikut ini penjelasan mengenai fungsi kulit sebagai indera peraba:

a. Sebagai indera peraba, kulit bertugas menerima rangsangan. Pada lapisan kulit tersebar ujung-ujung saraf peraba.
b. Bila kita memegang benda maka akan menimbulkan rangsang. Rangsang diterima oleh ujung-ujung saraf peraba, untuk diteruskan ke otak.

Dengan demikian kita dapat merasakan halus atau kasar sebuah permukaan benda. Selai itu kita dapat merasakan panas, dingin atau nyeri.

c. Bagian kulit yang paling peka terhadap rangsangan yaitu bagian yang terdapat pada bibir dan ujung jari.
d. Ujung-ujung saraf peraba juga terdapat pada dinding alat-alat dalam, misalnya dinding usus. Oleh karena itu, kita dapat merasakan sakit atau nyeri pada alat-alat tersebut.

Penjelasan tersebut tampak pada Gambar 2.16.



Gambar 2.16 *Bagian-bagian kulit [2]*

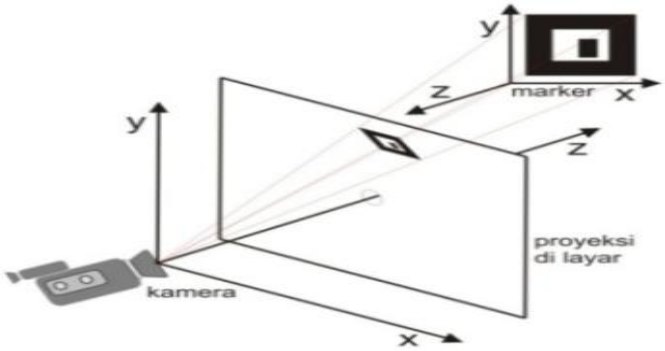
2.2 *Augmented Reality*

Augmented Reality (AR) adalah teknologi yang memungkinkan komputer untuk menampilkan objek *virtual* secara tepat di sebuah objek nyata secara *real time*. Sistem AR pertama kali dikembangkan di Sutherland pada tahun 1965, dan sampai sekarang terus berkembang pesat di berbagai bidang seperti kedokteran, manufaktur, hiburan, dan lain-lain. Sampai saat ini perkembangan AR terus terjadi, namun fokus perkembangan AR sendiri secara garis besar terbagi dalam 3 bidang yaitu [3] :

a. Teknologi *Tracking*
b. Teknologi penampilan
c. Teknologi interaksi

2.2.1 Lingkungan *Augmented Reality*

Pada sistem AR sistem koordinat yang dipakai adalah model *pinhole camera* atau kamera lubang jarum. Dimana pada model ini sumbu z positif berada di depan dan yang menjadi acuan adalah posisi marker jika dilihat dari kamera. Jika dilihat pada gambar 2.18, terlihat marker dan kamera masing-masing memiliki orientasi posisi yang berbeda. Baik marker maupun kamera menggunakan *system right handed* (sumbu z positif didepan) dan hasil penangkapan gambar dari kamera di proyeksikan ke *viewplane* menggunakan proyeksi perspektif [3]. Penjelasan tersebut tampak pada Gambar 2.17.



Gambar 2.17 *Sistem koordinat lingkungan AR [3]*

Dalam menampilkan objek 3D yang sesuai dengan posisi dan orientasi marker, perlu diperhitungkan hasil proyeksi yang diterima *viewplane* (bidang proyeksi di layar) untuk kemudian ditampilkan. Selain proyeksi pada bidang 2D, dalam pergeseran marker maupun kamera perlu diperhatikan perubahaan posisi dan rotasi dalam sistem koordinat 3D. posisi dan orientasi dari marker didapat dari hasil *tracking* marker yang ditransformasi dengan operasi translasi dan rotasi, sedangkan posisi dan orientasi yang ada pada proyeksi di layar didapat dari perhitungan transformasi proyeksi perspektif [3].

2.2.2 Metode *Occlusion Based Detection*

Occlusion Based Detection adalah metode untuk mendeteksi ada tidaknya *occlusion* dalam penampilan objek 3D. Secara sederhana *occlusion based detection* hanya mendefinisikan kejadian dimana satu marker tidak terdeteksi karena tertutup benda lain [3].

Jika ada n objek yang diwakili matrik O, maka akan dihasilkan matrik O1, O2, . . . , On yang merupakan posisi proyeksi objek-1, objek-2, . . . , objek-n dilayar. Deteksi *occlusion* akan dilakukan dengan pengecekan 2 objek misal dipilih objek-1 terhadap objek-2 maka akan dilakukan pengecekan syarat pertidaksamaan *point clipping* berikut:

$$x = (O2x - \text{Batas } x) \leq (O1x) \leq (O2x + \text{Batas } x) \qquad (2.1)$$

$$y = (O2y - \text{Batas } y) \leq (O1y) \leq (O2y + \text{Batas } y) \qquad (2.2)$$

Keterangan:

x = Panjang marker

y = Lebar marker

O1x = Titik pada sumbu x objek satu

O1y = Titik pada sumbu y objek satu

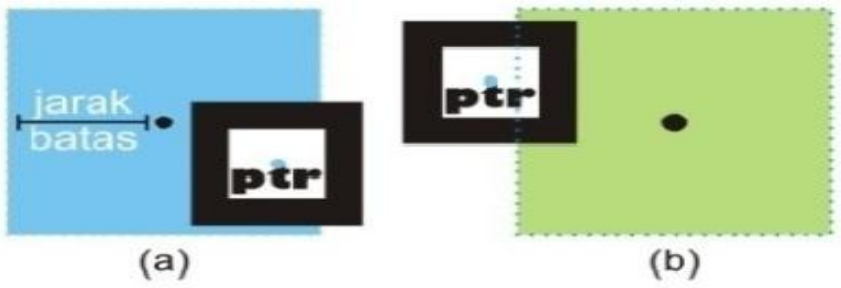
O2x = Titik pada sumbu x objek dua

O2y = Titik pada sumbu y objek dua

Batas = Jarak terhadap batas marker

Hasil deteksi ini berupa nilai kebenaran yang merupakan dasar pendefinisian *event* dari interaksi *occlusion based detection* jika persamaan (2.1) dan (2.2) terpenuhi [3].

Interaksi *occlusion based detection* adalah sebuah desain interaksi eksosentris dimana dalam mendefinisikan *event* untuk menghasilkan aksinya menggunakan metode *occlusion based detection* diatas. Berikut adalah Gambar 2.18.



Gambar 2.18 *(a) Terjadi event (b) Tidak terjadi event [3]*

Jika titik biru ditengah marker ptr adalah objek O1 dan titik hitam adalah objek O2 maka gambar 2.18. (a) dikatakan terjadi *event* karena memenuhi persamaan (2.1) dan (2.2) yaitu koordinat O1 (x,y) ada di dalam batas area O2. Sedangkan gambar 2.18. (b) tidak terjadi *event* karena hanya memenuhi pertidaksamaan (2.2) (nilai O1 y ada dalam batas O2 y) namun tidak memenuhi pertidaksamaan (2.1) [3].

2.3 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc*. membeli *AndroidInc*. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone* [4].

2.3.1 Keunggulan Android

Android dipuji sebagai "*platform mobile* pertama yang Lengkap, Terbuka, dan Bebas" [4].

1. Lengkap (*Complete Platform*)

   Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun software dan memungkinkan untuk peluang mengembangkan aplikasi.
2. Terbuka (*Open Source Platform*)

   Platform Android disediakan melalui *lisesnsi open source*. Pengembang dapat dengan bebas mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
3. *Free* (*Free Platform*)

   Android adalah platform/aplikasi yang bebas untuk develop. Tidak ada *lisensi* atau biaya *royalti* untuk dikembangan pada platform android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian.

Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.3.2 Android Versi 4.1 *(Jelly Bean)*

Android versi 4.1 *(Jelly Bean)* diluncurkan pada tanggal 9 Juli 2012, hal-hal yang direvisi dari sebelumnya adalah kemampuan seperti berikut: meningkatkan input *keyboard*, desain baru fitur pencarian, *UI* yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Sistem operasi Android *Jelly Bean* 4.1 muncul pertama kali dalam produk tablet *Asus*, yakni *Google Nexus 7* [4].

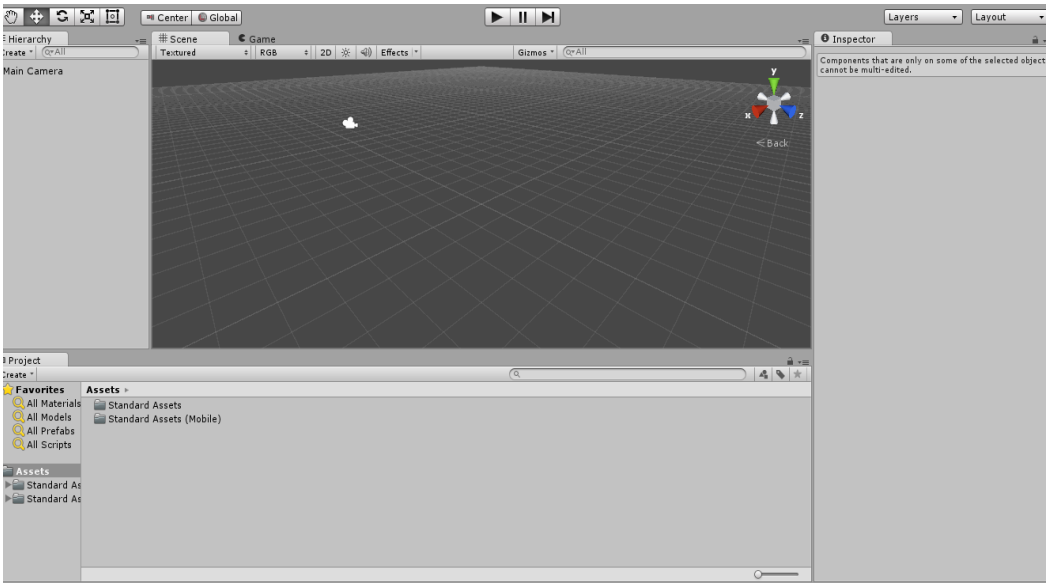
2.4 *Unity 3D*

Aplikasi *Unity 3D* adalah *game engine* merupakan sebuah *software* pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat suatu game, meskipun tidak selamanya harus untuk game. Contohnya adalah seperti materi pembelajaran untuk simulasi membuat SIM. Kelebihan dari *game engine* ini adalah bisa membuat game berbasis 3D maupun 2D, dan sangat mudah digunakan [5].

Unity 3D merupakan *game engine* yang ber-multiplatform. *Unity 3D* mampu di *publish* menjadi *Standalone* (.exe), berbasis web, berbasis web, Android, IoS Iphone, XBOX, dan PS3. Walau bisa *dipublish* ke berbagai *platform*, *Unity 3D* perlu *lisensi* untuk dapat dipublish ke *platform* tertentu. Tetapi *Unity 3D* menyediakan untuk *free user* dan bisa di *publish* dalam bentuk *Standalone (.exe)* dan *web*. Untuk saat ini *Unity 3D* sedang dikembangkan berbasis *Augment Reality* (AR) [5].

Untuk mengaktifkan *lisensi*, *Unity 3D* perlu adanya *lisensi*. Sebagai contoh ketika ingin mengaktifkan *free user*, langkah pertama adalah mendownload softwarenya secara gratis pada *web http://unity3d.com/*. Setelah selasai instalasi, maka *Unity 3D* meminta untuk terhubung dengan internet untuk aktifasinya. Lalu selesai *Unity 3D* akan *automatis run* ke program. Untuk langkah selanjutnya *Unity 3D* tidak perlu lagi memerlukan koneksi internet saat

menjalankan aplikasinya [5]. Berikut contoh tampilan *Unity 3D* tampak pada Gambar 2.19.



Gambar 2.19 *Tampilan unity 3D*

2.5 *Vuforia*

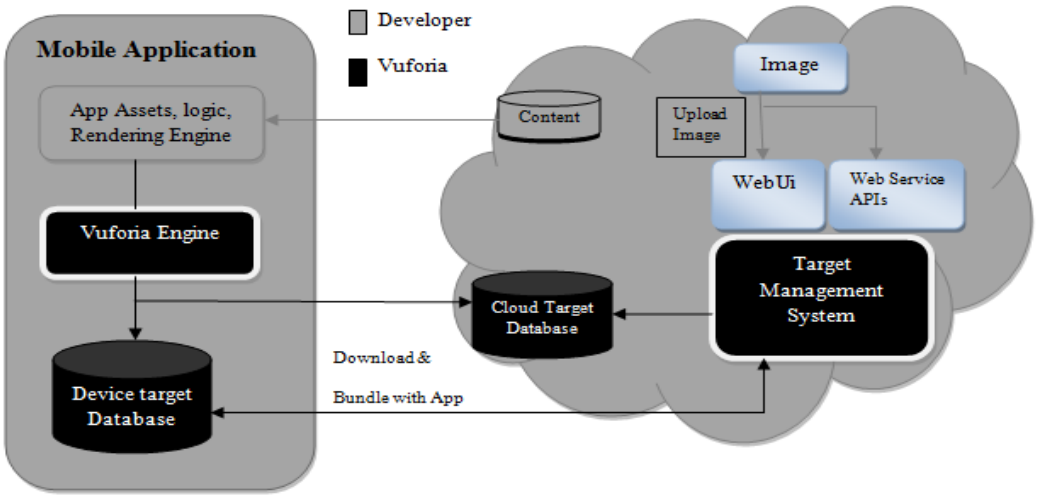
Vuforia merupakan *software* untuk *Augmented Reality* yang dikembangkan oleh Qualcomm, yang menggunakan sumber yang konsisten mengenai *computer vision* yang fokus pada *image recognition*. *Vuforia* mempunyai banyak fitur-fitur dan kemampuan, yang dapat membantu pengembangan untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknikal [6].

Dengan *support* untuk *iOS*, Android, dan *Unity 3D*, *platform Vuforia* mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis *smartphone* dan *tablet* [6].

Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain:

1. Teknologi *computer vision* tingkat tinggi yang mengijinkan *developer* untuk membuat efek khusus pada *mobile device.*
2. Terus-menerus mengenali *multiple image.*
3. *Tracking* dan *Detection* tingkat lanjut.
4. Solusi pengaturan *database* gambar yang *fleksibel*.

Penjelasan tersebut tampak pada Gambar 2.20.



Gambar 2.20 *Struktur vuforia [6]*

2.6 Bahasa Pemrograman C#

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti (*Java, Delphi, Visual Basic,* dan lain-lain) dengan beberapa penyederhanaan. Menurut standar *ECMA-334 C# Language Specification*, nama C# terdiri atas sebuah huruf Latin *C (U+0043)* yang diikuti oleh tanda pagar yang menandakan angka *# (U+0023)*. Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik *(U+266F)*, dan tanda pagar *# (U+0023)* tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam *keyboard* standar [7].

2.7 Pemograman Berorientasi Objek *(Object Oriented)*

Metode berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metode berorientasikan objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada

penerapan prinsip-prinsip pengolahan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas, analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [8].
Berikut beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek:

- Bahasa pemrograman Smalltalk
- Bahasa pemrograman Eiffel
- Bahasa pemrograman C++
- Bahasa Pemrograman C#
- Bahasa pemrograman (web) PHP
- Bahasa pemrograman Java

2.7.1 *Unified Modelling Language* (UML)

Unified Modeling Language (UML) adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [8].

2.7.2 Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967. Bahasa pemrograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada *developer* pengembang bahasa pemrograman berorientasi objek selanjutnya [8].

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS. Secara aktual, penggunaan bahasa pemrograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian di saat itu. Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shaler dan

Stephen Mellor (Shaler-Mellor, 1988) dan Peter Coad dan Edward Yourdon (Coad-Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, William Lorensen, Frederick Eddy, William Premerlani (Rumbaugh-Blaha-Premerlani-Eddy-Lorensen, 1991), dan masih banyak lagi [8].

Buku terkenal yang juga berkembang selanjutnya adalah karangan Ivar Jacobson (Jacobson, 1992) yang menerangkan perbedaan pendekatan yang fokus pada *use case* dan proses pengembangan. Sekitar lima tahun kemudian muncul buku yang membahas mengenai metodologi berorientasi objek yang diikuti dengan buku-buku yang lainnya. Didalamnya juga membahas mengenai konsep, definisi, notasi, terminologi, dan proses mengenai metodologi berorientasi objek [8].

Karena banyaknya metodologi-metodologi yang berkembang pesat itu, maka muncul ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators* (CRC) dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainya dimana James R. Rumbaigh, Grandy Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan *Rational Software Corporation* menghasilkan bahasa yang disebut dengan *Unified Modeling Language* [8].

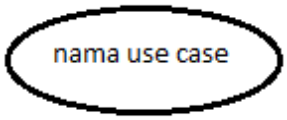
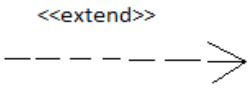
2.7.3 Diagram UML

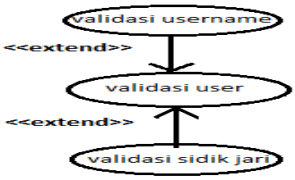
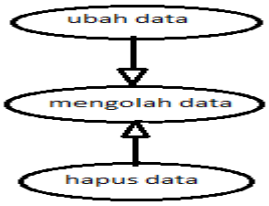
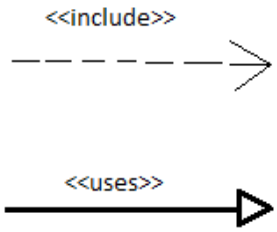
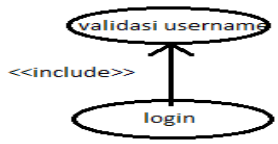
Berikut akan dijelaskan diagram yang akan digunakan dalam pembangunan aplikasi pembelajaran organ tubuh menggunakan *augmented reality*, yaitu *use case diagram, class diagram, sequence diagram*, dan a*ctivity diagram* [8].

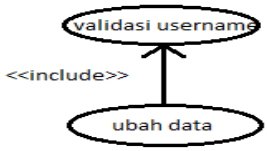
2.7.3.1 *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan pemodelan untuk (*behavior)* aplikasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak mengguankan fungsi-fungsi itu [8]. Penjelasan tentang simbol-simbol *use case* dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Definisi simbol use case diagram [8]*

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal *frase* nama *use case*.
Aktor/*actor* 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal *frasa* nama aktor.
Asosiasi/association 	Komunikasi antara aktor dan use case yang berpartisipasi pada *use case* atau *use case* memiliki interasi dengan aktor.
Ekstensi/*extend* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambhan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misal

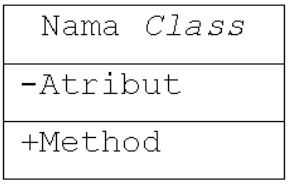
	 arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya.
Generalisasi/*generalization* 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya:  arah panah mengarah pada *use case* yang menjadi generalisasinya (umum).
Menggunakan/*include/uses* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. Ada dua sudut pandang yang cukup besar mengenai *include* di *use case*: • *Include* berarti *use case* yang ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan, misal pada kasus berikut: 

	• *Include* berarti use case yang tambahan akan selalu melakukan pengecekan apakah *use case* yang ditambahkan telah dijalakan sebelum *use case* tambahan dijalankan, missal pada kasus berikut:  Kedua *interpretasi* di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan *interpretasi* yang dibutuhkan.

2.7.3.2 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [8].

- Atribut, merupakan variable-variabel yang dimiliki oleh satu kelas.
- Operasi, atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sesuai. Contoh *class diagram* dapat dilihat pada Gambar 2.21.

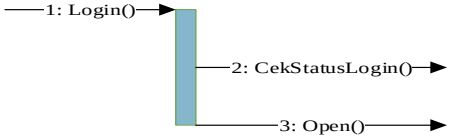


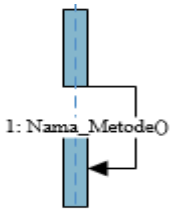
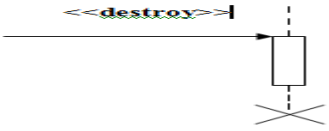
Gambar 2.21 *Contoh class diagram [8]*

2.7.3.3 *Sequence Diagram*

Sequence diagram menggambarkan *behavior* objek pada *use case* dengan mendeskripsikan waktu hidup dan pesan yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *sequence diagram* juga dibutuhkan melihat skenario yang ada pada *use case* [8]. Penjelasan tentang simbol-simbol *sequence diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Definisi simbol sequence diagram [8]*

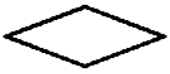
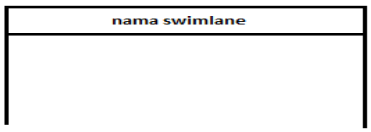
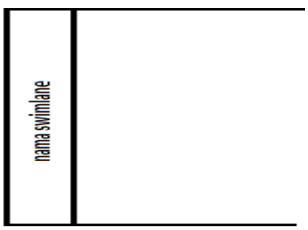
Simbol	Deskripsi
Aktor  Nama Aktor Atau Nama Aktor Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal *frasa* nama aktor.
Garis hidup/*lifeline* 	Menyatakan kehidupan suatu objek.
Objek Nama Objek: Nama Kelas	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya  Maka cekStatusLogin() dan open() dilakukan

	didalam metode login() Aktor tidak memiliki waktu aktif.
Pesan *tipe create* <<Creat>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan *tipe call* 1: Nama_Metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan *tipe send* 1: Masukan	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
Pesan *tipe retrun* 1: Keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan *tipe destroy* 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*.

2.7.3.4 *Activity Diagram*

Diagram aktivitas atau a*ctivity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan adalah diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *aktor*. Tabel 2.3 adalah penjelasan tentang simbol-simbol *activity diagram* [8].

Tabel 2.3 *Definisi simbol activity diagram [8]*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/*join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.8 Pengujian *Alpha*

Alpha adalah status *developmen* pada *software* yang mungkin belum terdapat seluruh fitur yang direncanakan dan belum stabil, tetapi dapat menunjukan atau memberikan gambaran dari hasil akhir sebuah software. *Versi alpha* dapat di katakan sebuah *versi review* yang akan di-*test* secara intensif oleh *internal* penguji/*tester* tetapi dapat diberikan juga pada penguji *eksternal*, misalkan user yang ingin mencoba *versi alpha* tersebut. Tahapan *alpha* bisa berkembang dari *alpha* satu, *alpha* dua dan seterusnya tergantung dari hasil pengujian dan tindakan dari hasil pengujian tersebut seperti perbaikan bug atau error, penambahan, penyesuaian dan sebagainya dari *software* tersebut [9].

2.9 Pengujian *Beta*

Beta adalah sebuah status *developmen* pada *software* setelah status *alpha* telah dinyatakan selesai. *Beta* dapat dikatakan sebuah *versi software* yang telah lulus *test*/pengujian oleh tim *tester*/penguji *internal*. Pada *versi beta software* ditujukan kepada pengguna atau calon pengguna untuk mencoba *software* tersebut dengan tujuan untuk mendapatkan *feedback* yang dialamin penggunan saat menggunakan *software* tersebut (seperti apakah terjadi bug, error, ribet dan sebagainya) [9].

Jadi pada tahap ini proses perbaikan bug dan error masih terus dilakukan oleh pihak pembuat *software* dan masih dapat terjadi penambahan, penggantian, penyesuaian *feature* dalam skala kecil. Tahap *beta* dapat berkembang dari beta satu, beta dua dan seterusnya tergantung dari hasil *feedback* pengguna yang berperan sebagai penguji dan *software versi beta* tidak direkomendasikan untuk digunakan keperlukan produksi. Sebagai contoh *software versi beta* dapat berupa *download* gratis *software beta*. Pada umumnya versi beta terbagi menjadi *open beta* yang berarti secara umum siapa saja dapat mencoba *versi software* tersebut dan *close beta* yang artinya adalah *sofware* tersebut hanya dirilis untuk sekolompok individu saja yang bertujuan untuk mencoba *software* tersebut [9].