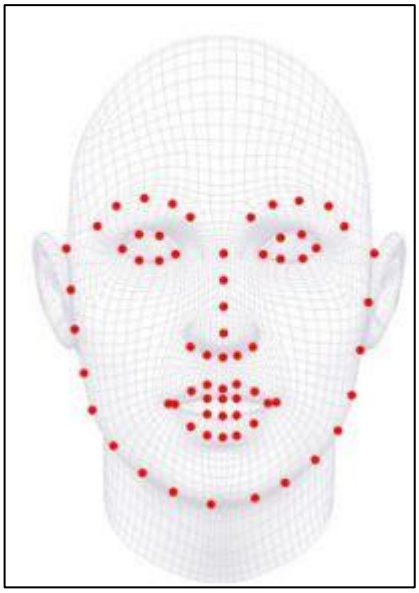
BAB 2

LANDASAN TEORI

2.1 Facial Features

Facial features ditemukan dengan menemukan *landmark*, sebuah *landmark* merepresentasikan sebuah titik *distinguishable* yang ada di mayoritas gambar dalam konsiderasi, contohnya lokasi pupil mata kiri (Milborrow dan Nicolls, 2008). Sebuah set *landmark* membuat sebuah bentuk, dimana bentuk itu di representasikan sebagai vektor yang isinya semua pasangan variabel x dan y dari semua titik di dalam *shape* tersebut (Milborrow dan Nicolls, 2008). Titik-titik *landmark* ini disebut juga dengan istilah *facial annotation* (Sagonas, dkk, 2008).



Gambar 2.1 Contoh 68 *facial annotation* pada *dataset* iBUG 300-W

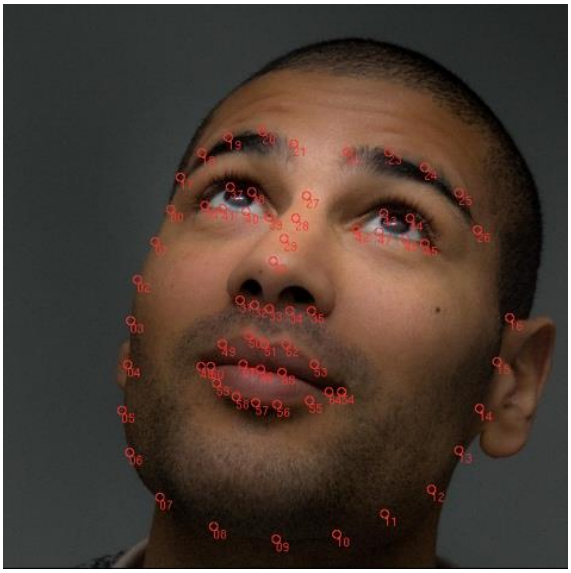
(Sagonas dkk., 2016)

Untuk dapat mengetahui posisi *facial annotation* pada satu muka, komputer dapat menggunakan *pre-trained neural network* untuk mengekstrak posisi *facial annotation* dalam suatu *frame* wajah. *Neural network* yang dibutuhkan komputer dapat dilatih secara *manual*, atau diambil dari *library* yang sudah ada.

2.2 Liveness Detection

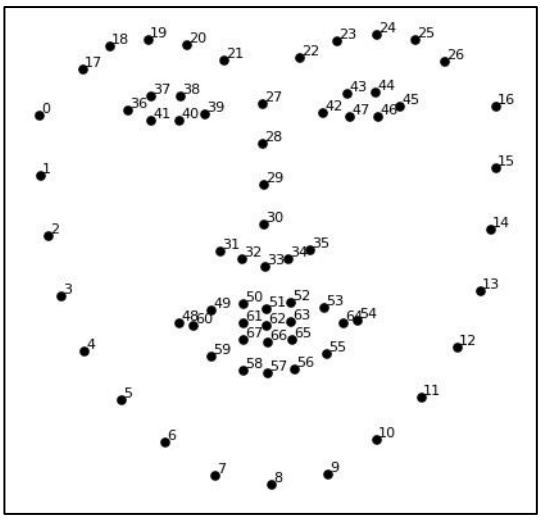
Liveness detection menentukan apakah wajah seseorang itu asli atau di sintesis dengan cara mengukur *liveness* (aktivitas) wajah (Parveen dkk., 2015). Beberapa metode yang dapat dipakai dalam *liveness detection* adalah dengan melakukan analisis terhadap *texture*, analisis terhadap *frequency* (Kim dkk., 2012), dan dengan mengukur pergerakan *facial features* (koordinat wajah) di gambar/*frame* yang diberikan (Parveen dkk., 2015). Menurut Deniz dkk. (2007), meminta user untuk melakukan sebuah aksi seperti senyuman, aksi-aksi tersebut di definisikan dengan cara melakukan sebuah kalkulasi terhadap rumus-rumus matematika dengan memakai koordinat wajah (*facial annotation*) di dalam sebuah kumpulan *frame*.

Dataset iBUG 300-W menggunakan 68 *facial annotation* sesuai dengan paper penelitian (Sagonas dkk., 2013).



Gambar 2.2 Hasil deteksi 68 *facial annotation* oleh Dlib pada *dataset* iBUG 300-W. (King, 2017)

Berikut adalah pengukuran 68 titik *facial annotation* (koordinat wajah) dari iBug-300 dataset menggunakan *library* Dlib.

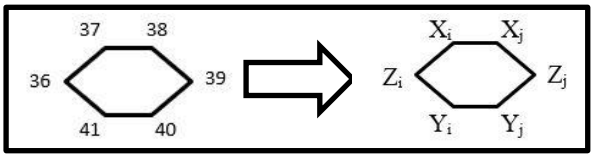


Gambar 2.3 Contoh 68 *facial annotation* dalam Dlib (Korshunov dkk., 2018)

Setidaknya ada tiga rumus untuk indikator yang dipakai oleh *liveness detection* dalam metode *life sign*, rumus pertama untuk mengukur *Eye Aspect Ratio* (EAR) (Awasekar dkk., 2018), rumus kedua untuk mengukur *Mouth Aspect Ratio* (MAR)

(Awasekar dkk., 2018), dan rumus ketiga untuk mengukur α dan γ (Wang dkk., 2018).

Rumus EAR menghitung keterbukaan mata dengan mengukur koordinat-koordinat pada mata didalam sebuah *frame* (Awasekar dkk., 2018). Untuk illustrasinya ada di gambar berikut.

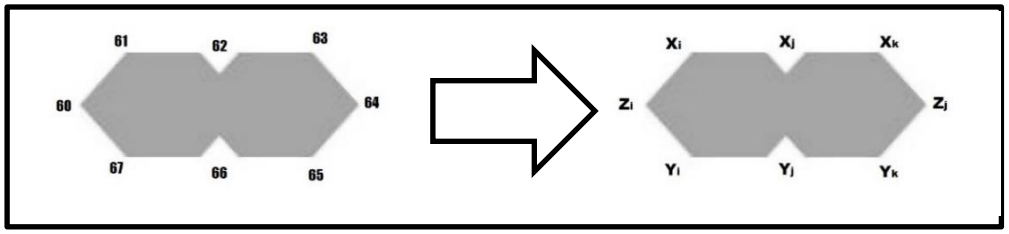


Gambar 2.4 Analogi mata

Menurut Awasekar, Ravi, Doke dan Shaikh (2018), pengukuran mata sebagai berikut:

$$\text{EAR} = \frac{|(Xi+Xj)-(Yi+Yj)|}{2*|Zi-Zj|} \quad (2.1)$$

Sementara itu, rumus MAR menghitung keterbukaan mulut dengan mengukur koordinat-koordinat pada mulut (Awasekar dkk., 2018), dalam kasus ini semua titik koordinat wajah terdalam dari bagian mulut (tidak menghitung bibir, hanya bukaan mulut). Illustrasinya ada di gambar berikut.

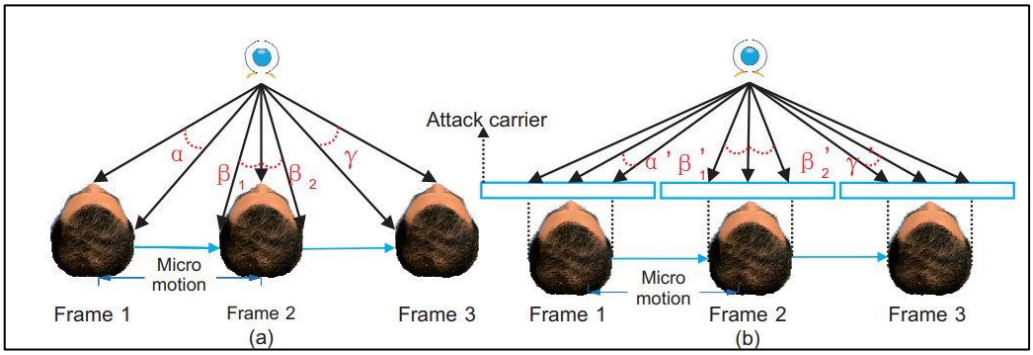


Gambar 2.5 Analogi mulut

Menurut Awasekar dkk, (2018), pengukuran mulut adalah sebagai berikut:

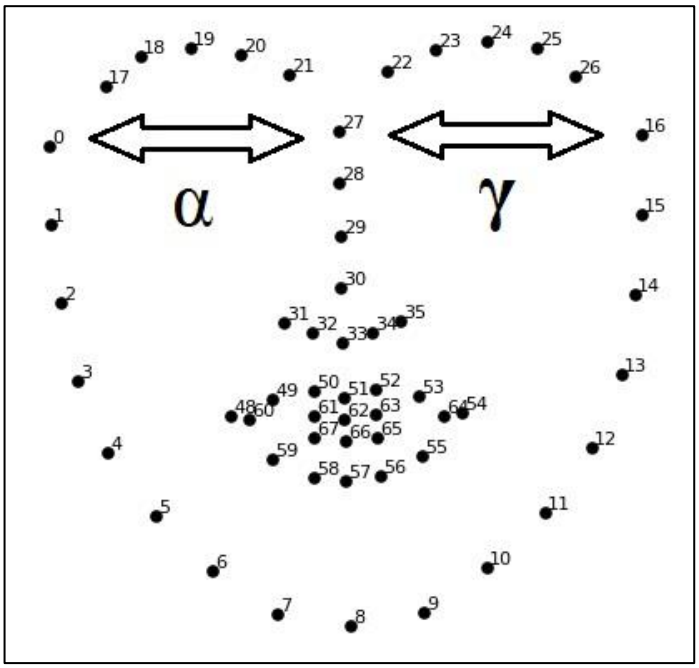
$$\text{MAR} = \frac{|(Xi+Xk)-(Yi+Yk)|}{2*|Zi-Zj|} \quad (2.2)$$

Untuk rumus ketiga, α dan γ adalah sudut yang dihasilkan untuk dalam mengukur sudut muka (Wang dkk., 2018), dijelaskan lebih *detail* oleh gambar berikut.



Gambar 2.6 Sudut wajah. Wang, Z. dkk. (2018)

Menurut Wang, Z. dkk. (2018), sudut β_1 adalah sudut antara telinga kiri dan hidung dan sudut β_2 adalah sudut antara telinga kanan dan hidung. Setelah seseorang mengganti posisi wajah atau memutar kepala, hasil penghitungan baru β_1 adalah γ dan hasil penghitungan baru β_2 adalah α. Dalam kasus wajah manusia, bila wajah tersebut bergerak ke kiri atau memutar kepala ke kiri, maka α akan lebih besar dari β_2 ($\alpha > \beta_2$) dan begitu juga sebaliknya bila wajah tersebut bergerak ke kanan atau memutar kepala ke kanan, maka v akan lebih besar dari β_1 ($\gamma > \beta_1$). Pada kasus *printing attack*, hal ini terbalik, yaitu ($\alpha < \beta_2$) dan ($\gamma < \beta_1$). Ilustrasi sudut pergerakan α dan γ dapat dilihat di gambar berikut ini.



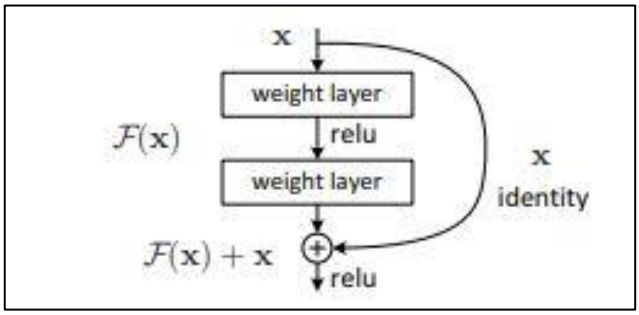
Gambar 2.7 Illustrasi sudut pergerakan α dan γ

2.3 Neural Network

Ada beberapa tipe *neural network*, yang akan dibahas ada dua, yaitu *Residual Neural Network* (ResNet), dan *Convolutional Neural Network* (CNN).

2.3.1 ResNet

Residual Neural Network (ResNet) adalah sistem *machine learning* berbasis *neural network* yang menggunakan *residual learning*. Residual learning dapat melakukan lompatan terhadap suatu layer untuk menuju layer selanjutnya (He, K dkk. 2015). ResNet terbentuk dengan cara melatih *neural network* dengan *residual learning*.



Gambar 2.8 Residual learning. (He dkk., 2015)

ResNet yang dipakai disini adalah ResNet *pre-trained* dlib dengan 29 *conv layer* yang merupakan ResNet-34 (He dkk., 2016) yang dimodifikasi (King, 2019). ResNet ini mempunyai *mean error* sebesar 0.993833 dengan standar deviasi 0.00272732 di LFW benchmark (King, 2019). Neural network untuk mendeteksi wajah adalah *pre-trained* Mmod model dengan dataset public yaitu ImageNet, AFLW, Pascal VOC, the VGG dataset, WIDER, and face scrub (King, 2019). Neural network untuk mengenali wajah adalah *pre-trained face landmark* model dengan menggunakan dataset iBUG 300-W (Antonakos dkk., 2016) (King, 2019).

2.3.2 CNN

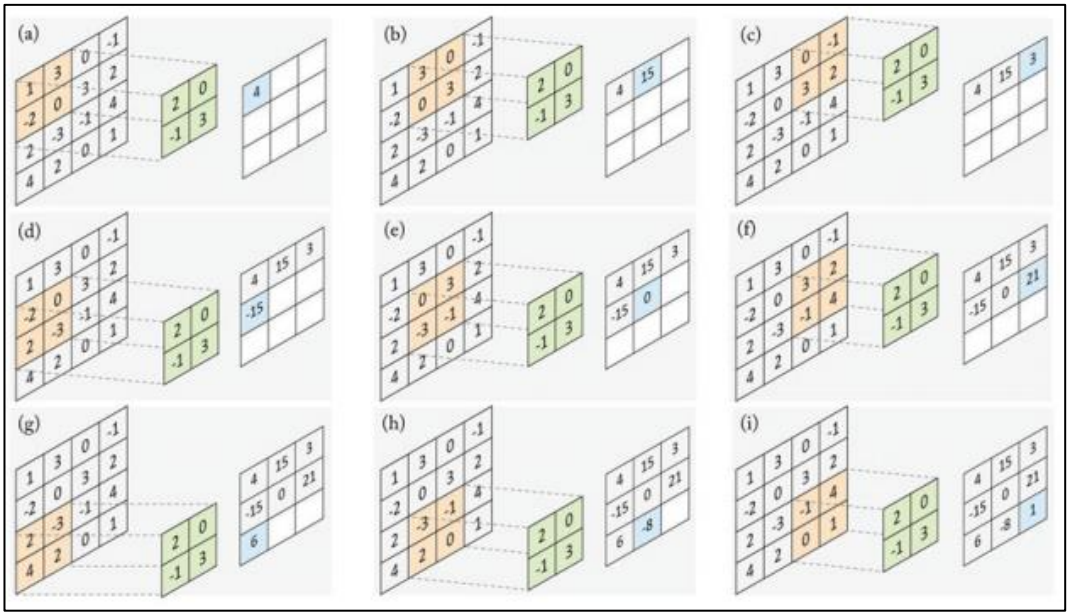
Convolutional Neural Network (CNN) secara historis menjadi yang paling sukses dari semua jenis *neural network*. Mereka digunakan secara luas untuk pengenalan gambar, deteksi objek / lokalisasi, dan bahkan pemrosesan teks (Aggarwal dkk, 2018).

Convolutional layer adalah komponen terpenting dari CNN, terdiri dari satu set filter yang *convolted* dari sebuah *input* untuk menghasilkan *output* (Khan dkk, 2018). Filter yang dimaksud disini adalah satu tabel yang berisi angka diskrit untuk mewakilkan beban, seperti filter 2x2 di gambar berikut.

2	0
-1	3

Gambar 2.9 Contoh tabel filter 2x2 (Khan dkk, 2018)

Angka-angka beban tersebut didapatkan dengan cara melakukan *training*, yaitu di initialisasi secara acak, lalu dengan pasangan *input-output*, angka-angka beban ini di-*tune* seiring iterasi yang berjalan (Khan dkk, 2018).



Gambar 2.10 Contoh iterasi convolutional layer (Khan dkk, 2018)

Gambar diatas mewakili satu iterasi dalam input-output dengan beban yang diberikan, dibagi menjadi 9 kali penghitungan (a-i) karena hasilnya matrix 3x3. Tiap iterasi, Tabel angka dikiri adalah tabel *input* (kasus ini 4x4) (*input* yang dipakai adalah yang ditandai merah), tabel hijau ditengah adalah tabel angka beban, dan tabel di kanan adalah tabel *output* (output yang dihasilkan per proses adalah yang diwarnai biru).

Pre-trained Convolutional Neural Network dari *resource* umum seperti ImageNet sering kali digunakan secara off-the-shelf untuk aplikasi dan kumpulan data lain (Aggarwal, 2018).

2.4 Spoofing Attack

Sebuah *attack* adalah suatu usaha untuk menipu sistem biometrik (Erdoğmuş dan Marcel, 2014). Menurut Erdoğmuş dan Marcel (2014), secara garis besar ada dua jenis *attack* di dalam sistem biometrik, yaitu langsung dan tidak langsung. Di dalam suatu *attack* secara langsung, seorang penyamar dapat mengganti karakteristik biometrik yang dipunyai untuk menghindari identifikasi (*obfuscation*), atau mengklaim sebuah identitas seorang *authorized user* dengan cara berpose sendiri (*zero-effort attack*) atau dengan cara mempresentasikan sifat biometrik yang dipalsukan dari *user* tersebut (*spoofing* atau *presentation attack*) (Erdoğmuş dan Marcel, 2014).

Menurut Erdoğmuş dan Marcel (2014), *spoofing attack* berpotensi sebagai tipe *attack* yang paling berbahaya karena tidak memerlukan kemampuan pemrograman yang ahli seperti *attack* tidak langsung. Tidak seperti *zero-effort attack*, *spoofing attack* membawa ancaman yang besar terhadap keamanan, apalagi dengan mempertimbangkan *false acceptance rate,* dari sistem biometrik (Erdoğmuş dan Marcel, 2014).

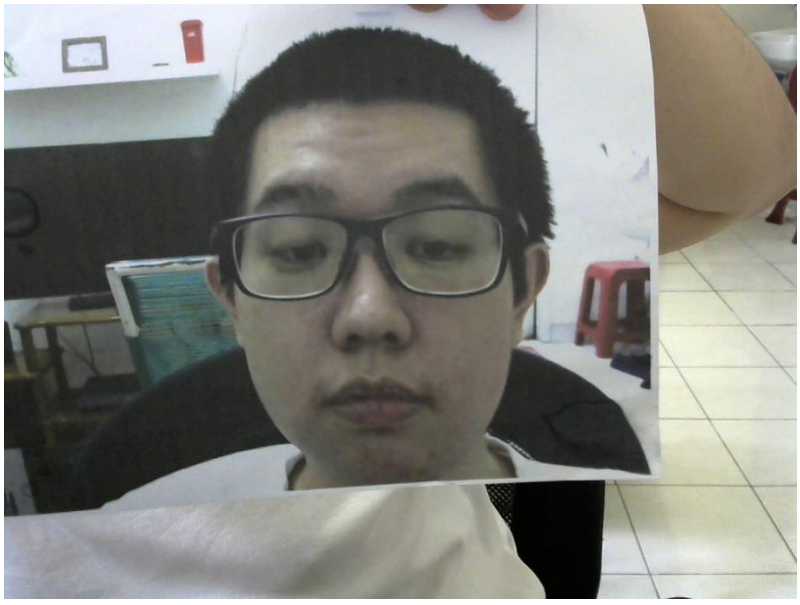
Adapun *spoofing attack* yang umum di dalam pengenalan wajah, yaitu *photo attack, video attack,* 3D *mask attack* (Hernandez-Ortega, dkk., 2019). Jenis *photo attack* dan *video attack* adalah yang paling umum digunakan karena tingkat

eksposisi wajah yang sangat tinggi seperti foto dan video yang tersebar di sosial media dan rekaman CCTV, dan biaya rendah untuk kamera, *printer* dan layar beresolusi tinggi (Hernandez-Ortega, dkk., 2019).

2.4.1 Photo Attack

Photo attack adalah sebuah *attack* yang dilakukan dengan cara menampilkan sebuah foto orang yang bersangkutan ke sensor pengenalan wajah (Hernandez-Ortega, dkk., 2019).

Photo attack dapat dilakukan dengan cara melakukan printing terhadap wajah orang lain ke dalam kertas untuk ditampilkan (printed attack) atau ditampilkan ke layar elektronik (Hernandez-Ortega, dkk., 2019). Mendapatkan foto user asli dapat dilakukan dengan mudah berkat sosial media, lalu mencetaknya kedalam sebuah foto cetak (printed attack) adalah hal yang mudah dan juga murah (Hernandez-Ortega, dkk., 2019).



Gambar 2.11 Contoh photo attack.

2.4.2 Video Attack

Video attack adalah *attack* yang merekam wajah orang lain sebagai video dan menampilkannya di layar elektronik, lebih maju dibandingkan *photo attack* (Hernandez-Ortega, dkk., 2019). Tidak seperti *photo attack* yang hanya menampilkan wajah dan tekstur, *video attack* juga memperlihatkan gerakan dinamis (kedipan mata, pergerakan mulut, dkk) sehingga lebih sulit dideteksi (Hernandez-Ortega, dkk., 2019).

Gambar 2.10 Contoh video attack.

2.5 Akurasi dan F-score

Berdasarkan Lipton dkk, (2014), berikut tabel kebenaran dalam prediksi awal dan hasil aktual.

Tabel 2.1 Tabel kebenaran (Lipton dkk., 2014)

Kondisi	Aktual Positif	Aktual Negatif
Prediksi Positif	True Positive	False Positive
Prediksi Negatif	False Negative	True Negative

Menurut Lipton dkk, (2014), tabel tersebut menjelaskan nilai kebenaran dengan memperhitungkan prediksi awal dan hasil aktual dari suatu subjek yang diteliti, berikut definisi-definisi dari *true positive, false positive, true negative, dan false negative.*

- True positive (TP) = angka kasus positif yang dianggap positif oleh sistem.
- False positive (FP) = angka kasus negatif yang dianggap positif oleh sistem.
- True negative (TN) = angka kasus negatif yang dianggap negatif oleh sistem.
- False negative (FN) = angka kasus positif yang dianggap negatif oleh sistem.

Menurut Sokolova, Japkowicz dan Szpakowicz (2006), akurasi dapat diukur dengan rumus berikut

$$\text{Akurasi} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{FP}+\text{TN}+\text{FN}} \quad (2.3)$$

Secara matematis (Lipton dkk., 2014) presisi dan *recall* dapat dihitung dengan formula berikut:

$$\text{Presisi (p)} = \frac{\text{TP}}{\text{TP}+\text{FP}} \quad (2.4)$$

$$\textit{Recall (r)} = \frac{\text{TP}}{\text{TP}+\text{FN}} \quad (2.5)$$

Sementara untuk F-score (Lipton dkk., 2014) dapat diukur dengan rata-rata antara presisi dan *recall*

$$\text{F-score} = \frac{2}{(1/r + 1/p)}$$

$$\text{F-score} = \frac{2TP}{2TP+FP+FN} \tag{2.6}$$