

BAB 2

LANDASAN TEORI

2.1 Perdarahan Otak

Otak merupakan bagian dari sistem saraf pusat yang tersusun dari sumsum tulang belakang. Otak memiliki 100-200 milyar sel aktif yang saling berhubungan dan bertanggung jawab atas fungsi mental dan intelektual manusia (Leonard L. Berry, 1998). Otak dilindungi oleh berbagai lapisan, seperti duramater dan cairan serebrospinal, yang juga dapat ditemukan di sumsum tulang belakang. Dari luar, otak juga dilindungi oleh tengkorak, yang terdiri dari berbagai jenis tulang yang saling berhubungan. Namun, otak sangat rawan mengalami kerusakan dan perdarahan.

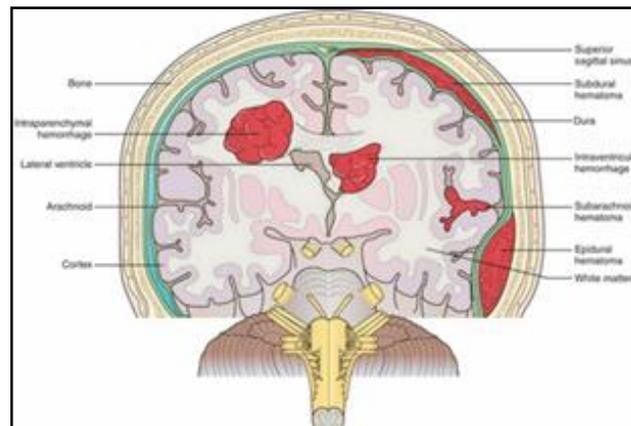


Gambar 2.1 Otak Normal (<https://radiopedia.org>)

Fong (2016) memperlihatkan bahwa di Hongkong, perdarahan otak menjadi penyakit keempat yang paling fatal karena hampir 3.000 orang meninggal setiap tahunnya. Terdapat banyak faktor penyebab perdarahan otak dan area di sekitarnya, antara lain cedera kepala, kelainan pembuluh darah, hemofilia, aneurisma (pembengkakan pembuluh darah), hipertensi dan lain sebagainya. Prof.Hafid (2010) mengatakan “orang-orang dengan penyakit hipertensi harus waspada dengan ancaman

aneurisma”. Goldstein (2006) (Goldstein L.B., 2006) menjelaskan bahwa hipertensi dapat mengakibatkan pecahnya maupun menyempitnya pembuluh darah otak yang mengakibatkan perdarahan otak dan apabila pembuluh darah otak menyempit maka aliran darah ke otak akan terganggu dan sel-sel otak akan mengalami kematian.

Apabila perdarahan menjadi tidak dapat dikendalikan, selain menyebabkan matinya sel otak dan terhentinya banyak fungsi tubuh, perdarahan otak juga dapat menyebabkan kelumpuhan sementara atau permanen, serta hilangnya kemampuan tertentu, seperti kemampuan motorik.



Gambar 2.2 Perdarahan Otak (Fong, 2016)

Menurut (Graham, 1995), terdapat 4 macam tipe pendarahan pada otak, antara lain:

a. Pendarahan Otak Epidural

Perdarahan ini menyebabkan adanya penumpukan darah pada ruang epidural, yaitu durameter dan tabula interna. Pendarahan otak epidural paling sering terjadi pada frontal dan temporal. Sumber pendarahan berasal dari arteri Meningea Media yang disebabkan oleh fraktur tulang (med.unhas.ac.id).



Gambar 2.3 Perdarahan Otak Epidural (<https://radiopedia.org/>)

Gejala dari perdarahan epidural, yakni kebingungan, vertigo, mengantuk atau perubahan tingkat kewaspadaan, membesarnya pupil di satu mata, sakit kepala, mual dan muntah. Berdasarkan Ashis P, et al. (2016), gejala biasanya terjadi dalam beberapa menit sampai beberapa jam setelah cedera kepala dan menunjukkan situasi yang darurat.

b. Pendarahan Otak Subdural

Penggumpalan darah yang terjadi di antara duramater dan subaraknoid, biasanya meliputi pendarahan vena (Graham, 1995). Pendarahan subdural paling sering akibat cedera kepala berat. Jenis subdural adalah salah satu yang paling mematikan dari semua cedera kepala. Perdarahan mengisi area otak dengan sangat cepat, mengompresi jaringan otak, sehingga dapat mengakibatkan kematian. Tergantung pada ukuran perdarahan dan dimana darah menekan pada otak.



Gambar 2.4 Perdarahan Otak Subdural (<https://radiopedia.org/>)

Gejala yang timbul antara lain, bicara bingung atau tidak jelas, masalah dengan keseimbangan atau berjalan, sakit kepala, vertigo, kejang atau kehilangan kesadaran, mual dan muntah (Stippler, 2016). Perdarahan ini terbagi atas akut (gejala timbul 3 hari pertama setelah cedera), sub-akut (hari ke 4-20), kronik (timbul gejala > 3minggu).

c. Perdarahan Otak Subarachnoid

Pendarahan subarachnoid adalah proses pecahnya pembuluh darah di ruang yang berada dibawah arakhnoid (subaraknoid). Prevalensi terjadinya perdarahan subarachnoid dapat mencapai hingga 33.000 orang per tahun di Amerika Serikat. Perdarahan subarachnoid memiliki puncak insidens pada usia ekitar 55 tahun untuk laki-laki dan 60 tahun untuk perempuan. Lebih sering dijumpai pada perempuan dengan rasio 3:2 (Setyopranoto, 2012). Perdarahan subarachnoid (Student Med, 2011)ditandai dengan adanya ekstrasvasasi darah ke rongga subarachnoid yaitu rongga antara lapisan dalam (*piamater*) dan lapisan tengah (*arakhnoid matter*) yang merupakan bagian selaput yang membungkus otak (*meninges*).



Gambar 2.5 Perdarahan Otak Subarachnoid (<https://radiopedia.org/>)

Gejala perdarahan subarachnoid (PERDOSSI, 2011) dapat berupa nyeri kepala yang mendadak dan kemudian hilang dengan sendirinya (30-60%), nyeri kepala

disertai mual, nyeri tengkuk dan fotofobia (40-50%), dan beberapa penderita mengalami serangan seperti “disambar petir”.

d. Perdarahan Otak Intraparenchymal

Pendarahan ini menyerang area yang homogen dan konfluen yang terdapat di dalam parenkim otak. Menurut Graham (1995), penyebab pendarahan bukan karena benturan antara parenkim otak dengan tulang tengkorak, tetapi disebabkan oleh gaya akselerasi dan deselerasi akibat trauma yang menyebabkan pecahnya pembuluh darah yang terletak lebih dalam, yaitu di parenkim otak atau pembuluh darah kortikal dan subkortikal. Umumnya terjadi pada pendarahan spontan.

Perdarahan intraparenkim menyumbang sekitar 8-13% dari semua stroke dan disebabkan oleh berbagai kelainan. Sehingga resiko kematian atau kecacatan utama lebih tinggi daripada stroke iskemik atau perdarahan subaraknoid. Perdarahan intraserebral dan edema yang menyertainya dapat mengganggu atau menekan jaringan otak yang berdekatan, yang menyebabkan disfungsi neurologis. Perpindahan substansial parenkim otak dapat menyebabkan peningkatan tekanan intrakranial (ICP) dan sindrom herniasi yang berpotensi fatal.



Gambar 2.6 Perdarahan Otak Intraparenchymal (<https://radiopedia.org/>)

e. Perdarahan Otak Intraventricular

Perdarahan intraventricular mengacu pada adanya darah di sistem ventrikel serebral. Ini terkait dengan morbiditas yang signifikan karena risiko hidrosefalus obstruktif. Perdarahan Intraventricular dapat dibagi menjadi perdarahan primer atau sekunder, dengan perdarahan primer jauh lebih jarang daripada perdarahan sekunder:

1. Primer

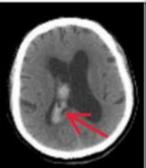
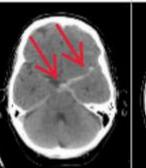
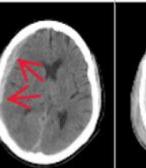
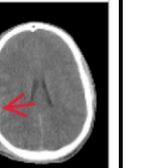
Darah di ventrikel dengan sedikit (jika ada) darah parenkim.

2. Sekunder

Terdapat komponen ekstrasventrikuler yang besar (misalnya parenkim atau subarachnoid) dengan ekspansi sekunder ke dalam ventrikel. Pada orang dewasa, perdarahan intraventrikel sekunder biasanya disebabkan oleh perdarahan intraserebral (biasanya perdarahan ganglia basalis hipertensi) atau perdarahan subaraknoid dengan refluks ventrikel.



Gambar 2.7 Perdarahan Otak Intraventricular (<https://radiopedia.org/>)

	Intraparenchymal	Intraventricular	Subarachnoid	Subdural	Epidural
Location	Inside of the brain	Inside of the ventricle	Between the arachnoid and the pia mater	Between the Dura and the arachnoid	Between the dura and the skull
Imaging					
Mechanism	High blood pressure, trauma, arteriovenous malformation, tumor, etc	Can be associated with both intraparenchymal and subarachnoid hemorrhages	Rupture of aneurysms or arteriovenous malformations or trauma	Trauma	Trauma or after surgery
Source	Arterial or venous	Arterial or venous	Predominantly arterial	Venous (bridging veins)	Arterial
Shape	Typically rounded	Conforms to ventricular shape	Tracks along the sulci and fissures	Crescent	Lentiform
Presentation	Acute (sudden onset of headache, nausea, vomiting)	Acute (sudden onset of headache, nausea, vomiting)	Acute (worst headache of life)	May be insidious (worsening headache)	Acute (skull fracture and altered mental status)

Gambar 2.8 Jenis-jenis Perdarahan Otak (<https://www.kaggle.com/>)

2.2 Visi Komputer

Teknologi visi komputer (*computer vision*) merupakan salah satu bidang dari teknologi *Artificial Intelligence* yang memuat kumpulan metode untuk mendapatkan, memproses, menganalisis suatu gambar atau dalam makna lain, visi komputer merupakan kumpulan metode-metode yang digunakan untuk menghasilkan angka-angka atau simbol-simbol yang didapat dari gambar tersebut. Inti dari teknologi visi komputer ini adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan (Sonka, Hlavac, & Boyle, 2015).

Menurut (Szeliski, 2010), kemampuan manusia didapatkan dengan mendeskripsikan dunia yang manusia lihat dalam bentuk satu atau lebih citra dan membentuk properti seperti bentuk, cahaya dan distribusi warna yang akan dianalisis menggunakan basis teoritis dan algoritma untuk menghasilkan pemahaman visual.

2.3 Citra

Citra merupakan suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra terbagi menjadi 2 (Sutoyo, 2019), yaitu:

2.3.1 Citra Analog

Citra analog adalah citra yang bersifat *continue* seperti gambar pada monitor televisi, foto sinar X, dan lain-lain. Citra analog tidak dapat direpresentasikan dalam computer, sehingga tidak bias diproses di computer secara langsung.

2.3.2 Citra Digital

Citra digital merupakan representative dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan *sampling* dan kuantisasi. Sampling menyatakan ukuran *pixel* pada citra dan kuantisasi menyatakan jumlah warna yang ada pada citra.

2.4 Pencitraan Medis

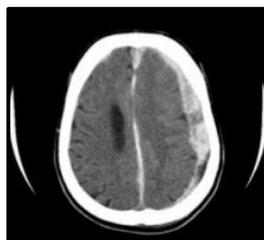
Kemampuan digitalisasi citra telah mengevolusi dunia medis dengan adanya citra medis (*medical image*) yang merupakan metode visualisasi tubuh manusia untuk mendukung penanganan medis berupa menganalisa dan mendiagnosis penyakit dari para pasien dalam ilmu kedokteran. Tugas utama yang dimiliki citra medis, yakni memberikan peta anatomi yang jelas sehingga memungkinkan untuk mengidentifikasi ketidaknormalan pada fisiologi tubuh (Wolbarst, Capasso, & Wyant, 2013). Menurut *European Society of Radiology* (European, 2015), pencitraan medis memiliki peran untuk melakukan prediksi, diagnosis dan pengobatan terhadap suatu penyakit yang diderita oleh para pasien.

2.4.1 CT Scan

Computed Tomography Scanner (CT Scan) telah terbukti sangat *valueable* sebagai alat diagnostik medis sehingga penemu *CT Scan* dianugrahi Penghargaan Nobel dalam Kedokteran 1979 dan menempati peringkat sebagai salah satu dari lima perkembangan medis terpenting dalam 50 tahun terakhir (RadiologyInfo.org For patients, 2020). Pemeriksaan *CT Scan* telah menjadi instrumen penting dalam pencitraan medis untuk melengkapi sinar-X dan ultrasonografi medis yang menghasilkan gambar organ, tulang, dan pembuluh darah di tubuh manusia (dr. Michael Andrianus Sundah, 2020).

2.4.2 CT Scan Kepala

CT Scan kepala merupakan prosedur untuk mendapatkan gambaran dari berbagai sudut kecil dari tulang, tengkorak dan otak sehingga dapat mempermudah dalam menemukan ketidaknormalan atau penyakit pada kepala. *CT Scan* kepala digunakan untuk melakukan pengecekan terhadap gejala-gejala seperti sakit kepala yang tiba-tiba berat, sakit kepala yang tidak membaik dengan istirahat, sakit kepala yang disertai demam, kejang, muntah, kehilangan koordinasi atau perubahan dalam penglihatan, ucapan atau kewaspadaan. Hal ini dapat disebabkan oleh cedera kepala, stroke, kejang, tumor otak ataupun perdarahan otak.



Gambar 2.9 Contoh Hasil CT Scan Kepala

Selain itu, dapat pula digunakan untuk melakukan pemeriksaan pasca operasi pada bagian kepala.

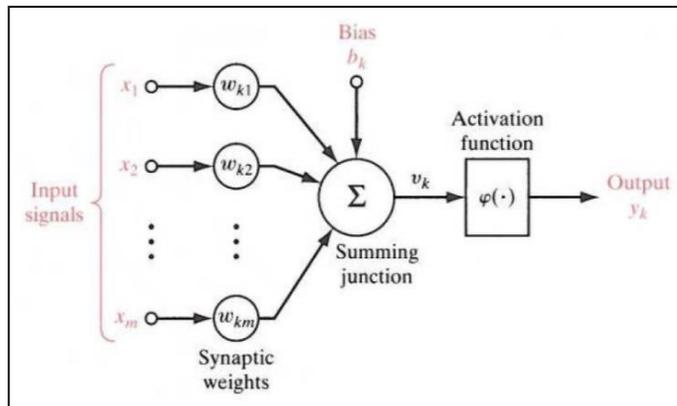
2.4.3 Metode CT Window

Dalam Metode CT Window atau disebut juga *windowing* dikenal 2 *window*, yaitu *Window Width* yang merupakan rentang nilai *computed tomography* yang dikonversi menjadi *gray scale* untuk ditampilkan ke TV Monitor. dan *Window Level* yang merupakan nilai tengah *window* yang digunakan untuk menentukan densitas gambar yang dihasilkan dalam penampilan gambar. Nilai-nilai tersebut memiliki satuan HU (*Hounsfield Unit*). HU merupakan satuan dimensi yang digunakan dalam mengukur *radio density* oleh para radiologi untuk menginterpretasikan citra *CT Scan*.

$$HU = Pixel_{value} * Rescale_{slope} + Rescale_{intercept} \quad \dots(2.1)$$

2.5 Artificial Neural Network

Artificial Neural Network Artificial (ANN) atau Jaringan Syaraf Tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi (Widiputra, 2016). Menurut Widiputra (2016), *Neural Network* terdiri dari sejumlah besar elemen pemrosesan informasi (*neuron*) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya adalah masalah klasifikasi ataupun prediksi.



Gambar 2.10 Struktur ANN (<https://socs.binus.ac.id/2012/07/26/konsep-neural-network/>)

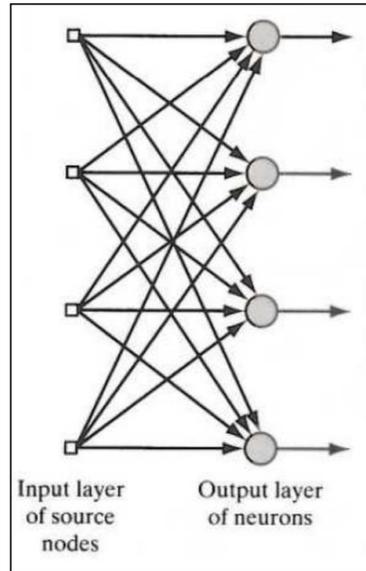
Struktur ANN secara mendasar, terdiri dari *input*, *output* dan fungsi aktivasi (*activation function*). Proses pada ANN dimulai dari input yang diterima beserta dengan nilai bobot dari setiap input yang ada. Nilai input tersebut yang ada akan dijumlahkan oleh suatu fungsi penjumlahan (*summing function*) yang dilambangkan dengan sigma (Σ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap *neuron*, dimana akan dilakukan perbandingan hasil penjumlahan dengan *threshold* (nilai ambang) tertentu. Jika nilai melebihi *threshold*, maka aktivasi *neuron* akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, *neuron* akan diaktifkan. Setelah aktif, *neuron* akan mengirimkan nilai *output* melalui bobot-bobot *output*-nya ke semua *neuron* yang berhubungan.

Menurut Haykin(2009), secara umum, ada 3 jenis arsitektur dari Jaringan Saraf Tiruan, yaitu:

1. *Single-Layer Feedforward Networks*

Jaringan Saraf Tiruan dengan satu *layer* memiliki sebuah *input layer* dari *node* sumber di mana informasi diproyeksikan ke *output layer* dari *neuron* tapi tidak bisa

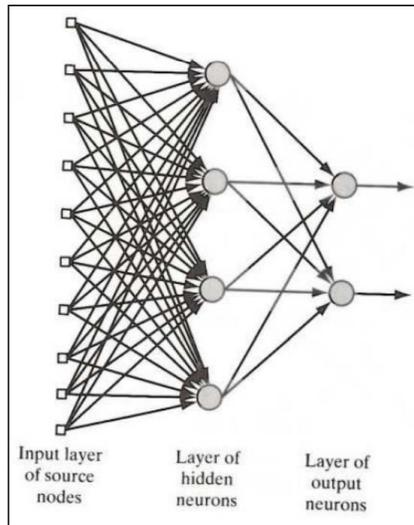
sebaliknya. Dengan kata lain, jaringan ini adalah tipe *feedforward*. Pada jaringan ini, *input layer* dari *node* sumber tidak dihitung karena tidak ada perhitungan yang dilakukan.



Gambar 2.11 Single-Layer Feedforward Networks (<http://library.binus.ac.id/eColls/eThesisd/doc/Bab2/2013-1-01024-IF%20Bab2001.pdf>)

2. *Multilayer Feedforward Networks*

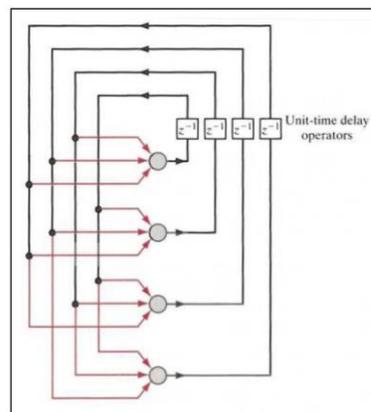
Jaringan ini bersifat *feedforward* yang memiliki satu atau lebih *hidden layer* yang berfungsi untuk mengintervensi antara *input* eksternal dan *output* dari jaringan dalam cara yang berguna. Sumber *node* di *input layer* dari jaringan menyediakan masing-masing elemen dari pola aktivasi (*vector input*), yang merupakan sinyal *input* yang diaplikasikan ke *neuron* di *layer* kedua (*hidden layer* pertama). Sinyal *output* dari *layer* kedua digunakan sebagai *input* ke *layer* ketiga, dan seterusnya sampai ke sisa dari jaringan. Gambar di bawah menunjukkan *multilayer network* yang terdiri dari 10 *input layer*, 4 *hidden layer*, dan 2 *output layer*.



Gambar 2.12 Multilayer Feedforward Network (<https://machinelearning.mipa.ugm.ac.id/2018/05/24/artificial-neural-network-ann/>)

3. Recurrent Network

Jaringan Saraf Tiruan *Recurrent* memiliki paling tidak satu *feedback loop* yang berpengaruh sangat besar pada kemampuan pembelajaran dari jaringan dan juga performanya. *Feedback loop* melibatkan penggunaan dari cabang-cabang yang terbentuk dari elemen *unit-time delay* (Z^{-1}).

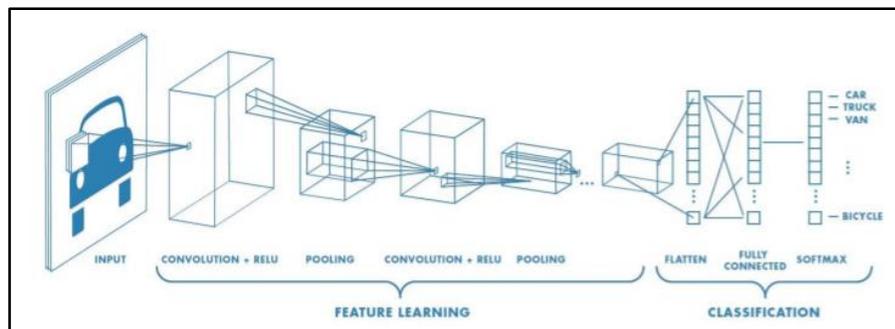


Gambar 2.13 Recurrent Networks (<https://machinelearning.mipa.ugm.ac.id/2018/05/24/artificial-neural-network-ann/>)

2.6 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu metode yang tercakup di dalam kelas *Feedforward Neural Network* merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi, seperti gambar atau suara. CNN termasuk dalam jenis *Deep Neural Network* karena kedalamana jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP tidak menyimpan informasi spasial dari data citra dan menganggap setiap *pixel* adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik.

Lapisan-lapisan CNN memiliki susunan *neuron* 3 dimensi (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran lapisan sedangkan kedalaman mengacu pada jumlah lapisan. Sebuah CNN dapat memiliki puluhan hingga ratusan lapisan yang masing-masing belajar mendeteksi berbagai data dua dimensi. Pengolahan citra dapat dimulai sebagai fitur yang sangat sederhana, seperti kecerahan dan tepi atau meningkatkan kompleksitas pada fitur yang secara unik menentukan objek sesuai ketebalan lapisan (Mathworks, 2017).



Gambar 2.14 Lapisan Convolutional Neural Network

Secara umum tipe lapisan pada CNN dibagi menjadi dua, yaitu:

1. Layer ekstraksi fitur (*feature extraction layer*)

Layer ini terletak pada awal arsitektur tersusun atas beberapa lapisan dan setiap lapisan tersusun atas *neuron* yang terkoneksi pada daerah lokal (*local region*) dari lapisan sebelumnya. Lapisan jenis pertama adalah *convolutional layer* dan lapisan kedua adalah *pooling layer*. Setiap lapisan diberlakukan fungsi aktivasi dengan posisinya yang berselang-seling antara jenis pertama dengan jenis kedua. Lapisan ini menerima *input* gambar secara langsung dan memprosesnya hingga menghasilkan *output* berupa vektor untuk diolah pada lapisan berikutnya.

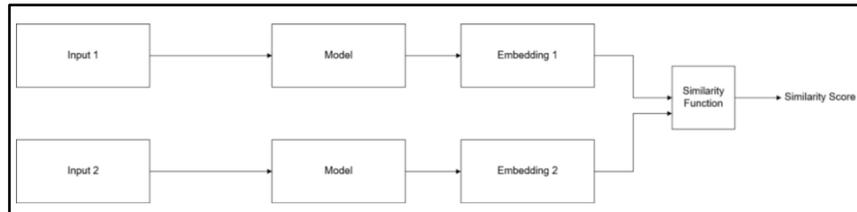
2. Layer klasifikasi (*classification layer*)

Layer klasifikasi tersusun atas beberapa lapisan dan setiap lapisan tersusun atas *neuron* yang terkoneksi secara penuh (*fully connected*) dengan lapisan lainnya. *Layer* ini menerima *input* dari hasil keluaran *layer* ekstraksi fitur gambar berupa vektor kemudian ditransformasikan dengan tambahan beberapa *hidden layer*. Hasil keluaran berupa akurasi kelas untuk klasifikasi.

2.7 Siamese Network

Dalam penelitian dan aplikasi *Deep Learning*, *Siamese Network* semakin populer. Jaringan siam atau *Siamese Network* adalah jaringan saraf yang berisi dua atau lebih komponen *subnetwork* yang identik. *Siamese Network* pertama kali ditemukan oleh Bromley and LeCun untuk verifikasi tanda tangan (Bromley, Guyon, LeCun, Säcker, & Shah, 1994).). Input Siamese network dapat berupa: data numerik

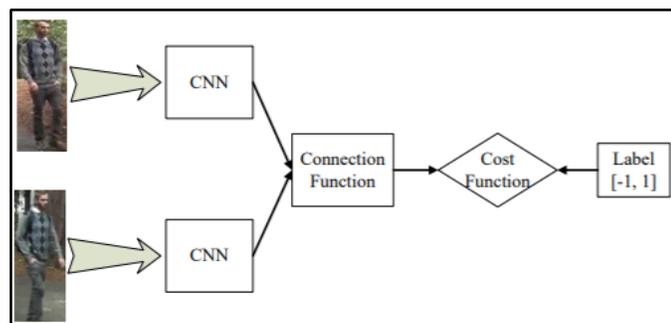
(*subnetwork* biasanya dibentuk oleh lapisan *fully connected*), data gambar (*subnetwork* berasal dari CNN) atau bahkan data berurutan seperti kalimat atau sinyal waktu (RNN sebagai *subnetwork*).



Gambar 2.15 Siamese Network

2.8 Siamese Convolutional Neural Network

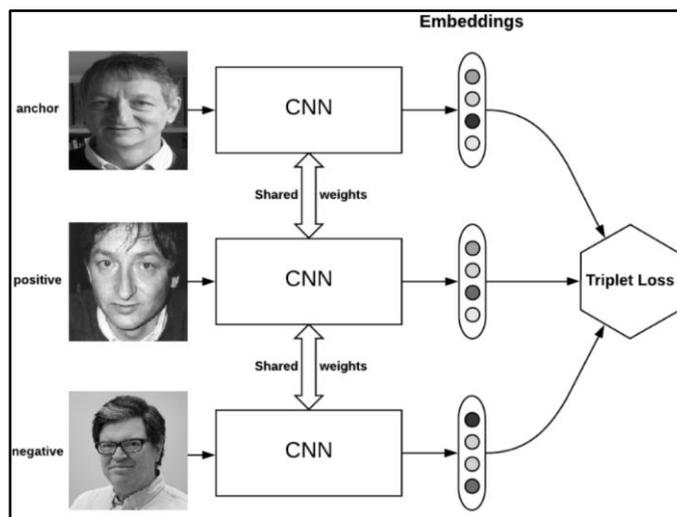
Metode *Siamese Network* dengan *Convolutional Neural Network* dipopulerkan pada tahun 2015 untuk *face recognition* (Koch, Zemel, & Salakhutdinov, 2015). Penggunaan *Siamese Network* dengan input gambar sering digabungkan dengan menggunakan *Convolutional Neural Network* (CNN) sebagai jaringan saraf yang identik. Hal ini dikarenakan, *siamese network* dapat mengatasi perbedaan jumlah penyebaran data yang tidak seimbang antar kelas, tidak seperti CNN yang dapat memberikan efek negatif dimana sebagian besar prediksi yang dihasilkan mengikuti jumlah data terbanyak pada kelas (Hensman & Masko, 2015).



Gambar 2.16 Siamese Convolutional Neural Network (Yi, 2014)

Siamese convolutional neural network menerapkan dua buah *convolutional neural network* dengan arsitektur yang sama, baik *subnetwork* yang identik maupun bobotnya. Masukan dari arsitektur ini berupa dua buah gambar yang berpasangan. Keluaran dari kedua input akan diberikan nilai berdasarkan kesamaan fitur satu sama lain dengan menggunakan perhitungan *cost function*. Keluaran yang dihasilkan dalam *siamese network* berupa *binary classification* yang langsung mengklasifikasi masukan berada pada kelas yang sama atau tidak.

2.8.1 Siamese Convolutional Neural Network Menggunakan Triplet Loss



Gambar 2.17 Triplet Network

Learning dengan *Siamese Network* sangat sering digunakan dengan fungsi *triplet loss*. Model dengan menggunakan fungsi *triplet loss* ini, sering disebut dengan *triplet network*, yaitu jaringan dimana bobot yang sama digunakan kembali untuk menghitung hasil untuk berbagai sampel. Model ini membutuhkan 3 *input (triplet pairs)* pada saat *training* yang terdiri dari *anchor*, *positive* dan *negative*. *Anchor* merupakan gambar yang ingin dilatih, *positive* adalah gambar yang memiliki kelas

yang sama dengan *anchor* sedangkan *negative* adalah gambar yang memiliki kelas yang berbeda dengan *anchor*. Setiap *triplet pairs* akan menyebar melalui jaringan dan *cost function* akan dihitung berdasarkan 3 embeddings yang dihasilkan. Untuk meminimalkan *cost function* harus meminimalisasi jarak antara *anchor* dan *positive* sekaligus memaksimalkan jarak ke *negative*.

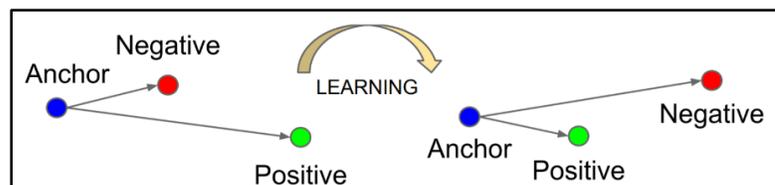
2.8.2 Euclidean distance

Euclidean distance digunakan untuk menentukan tingkat kesamaan (*similarity degree*) atau ketidaksamaan (*dissimilarity degree*) dua vektor fitur. Tingkat kesamaan berupa nilai dan berdasarkan nilai tersebut dua vektor fitur akan dikatakan mirip atau tidak. Perhitungan *euclidean distance* akan dilakukan dengan menghitung akar dari kuadrat perbedaan 2 vektor (Putra, 2010).

$$d_{i,j} = \sqrt{\sum_{k=1}^n (X_{ij} - X_{jk})^2} \quad \dots(2.2)$$

2.8.3 Triplet Loss Function

Triplet loss merupakan salah satu algoritma yang diterapkan dalam OpenFace sebagai salah satu komponen dari *stochastic gradient descent* pada saat training (Amos et al., 2016).



Gambar 2.18 Triplet Loss Function

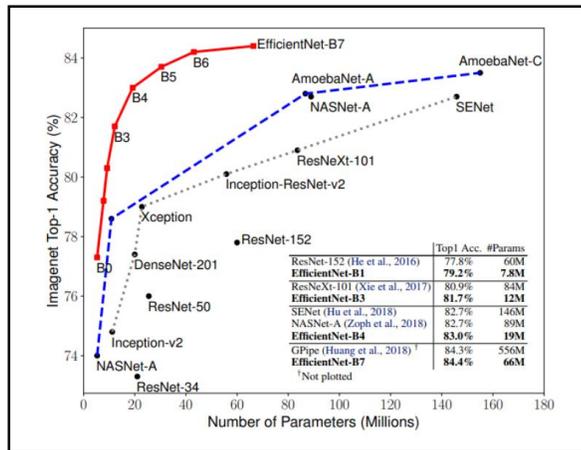
Triplet Loss bertujuan agar jarak antara *anchor* dengan *positive images* menjadi lebih dekat daripada dengan *negative images* (Schroff et al., 2015). Perhitungan Triplet Loss dapat dilihat, sebagai berikut.

$$L = \max(d(a, p) - d(a, n) + \alpha, 0) \quad \dots(2.3)$$

Dimana $d(a,p)$ merupakan jarak antara *anchor* dengan *positive*, $d(a,n)$ merupakan jarak antara *anchor* dengan *negative* dan pada notasi tersebut terlihat adanya parameter α . Parameter ini dikenal sebagai *triplet margin* yang pada penerapannya bernilai 0,3. Hal ini bertujuan agar selisih antara *anchor-positive pair* dengan *anchor-negative pair* bernilai relatif jauh, dimana pada umumnya akan memberikan performa yang baik dalam proses *training*. Sebaliknya, *triplet pairs* yang memiliki nilai diluar *triplet margin* tersebut akan diabaikan karena memiliki resiko yang dapat menyebabkan proses *training* menjadi gagal (*collapsed*) dengan terjadinya *early convergence*.

2.9 Arsitektur EfficientNet

EfficientNet merupakan salah satu keluarga arsitektur *neural network* yang dirilis oleh Google pada tahun 2019. Arsitektur *EfficientNet* mampu meningkatkan akurasi hingga 6% dengan jumlah parameter yang jauh lebih sedikit dari model CNN biasa, sehingga arsitektur ini sangat direkomendasikan karena telah terbukti lebih baik daripada arsitektur lainnya, seperti *DenseNet*, *Inception* ataupun *ResNet* dalam bidang *image classification* dan dapat berjalan secara signifikan dengan cepat. Perbandingan *EfficientNet* dengan arsitektur CNN lainnya terlihat pada Gambar 6.12.



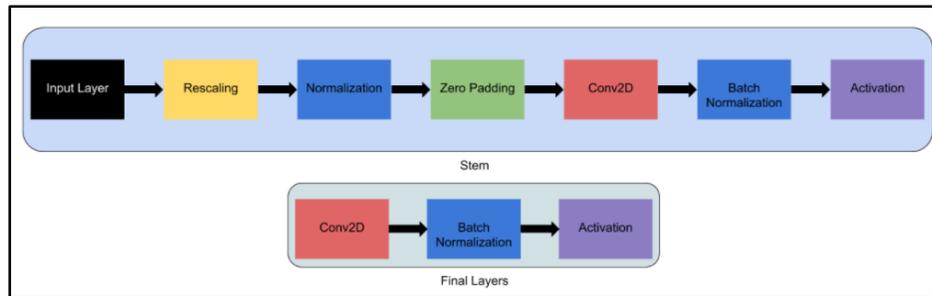
Gambar 2.19 Perbandingan Akurasi Arsitektur (<https://morioh.com/p/ebb64276117e>)

EfficientNet memiliki 8 model, antara lain: EfficientNet-B0, EfficientNet-B1, EfficientNet-B2, EfficientNet-B3, EfficientNet-B4, EfficientNet-B5, EfficientNet-B6, dan EfficientNet-B7. Kedelapan model *EfficientNet* dari B0 sampai B7 memiliki *input shape* yang berbeda-beda. Berikut daftar *input shape* yang diharapkan setiap model:

Base model	resolution
EfficientNetB0	224
EfficientNetB1	240
EfficientNetB2	260
EfficientNetB3	300
EfficientNetB4	380
EfficientNetB5	456
EfficientNetB6	528
EfficientNetB7	600

Gambar 2.20 Input shape model EfficientNet

Kedelapan model *EfficientNet* masing-masing terdiri dari 2 bagian, yaitu *Stem* dan *Final Layers* (<https://morioh.com/p/ebb64276117e>). *Stem* terdiri dari 7 blok yang memiliki jumlah sub-blok yang berbeda, sesuai dengan jenis arsitekturnya.



Gambar 2.21 Layer Arsitektur EfficientNet

2.10 Principal Component Analysis

Principal Component Analysis atau PCA merupakan salah satu teknik statistik yang secara linier mentransformasikan sekelompok variabel data asli menjadi sekelompok variabel data substansial yang tidak terkorelasi (Jolliffe,1986). PCA bertujuan mengurangi dimensi data dengan mempertahankan sebanyak mungkin informasi dari dataset yang asli, sehingga semakin sedikit data, algoritma akan semakin cepat, dan akurasi menjadi lebih tinggi. Dengan menggunakan PCA, variabel yang tadinya sebanyak n variabel akan direduksi menjadi k variabel baru (*principal component*) dengan jumlah k lebih sedikit dari n dan dengan hanya menggunakan k *principal component* akan menghasilkan nilai yang sama dengan menggunakan n variabel.

Penentuan jumlah komponen utama (*principal component*) yang akan digunakan dapat ditentukan dengan melihat total varians yang lebih dari 80%. Pada penelitian ini, PCA digunakan sebagai ekstraksi fitur dan penentuan jumlah komponen utama dengan melihat total varians yang dapat menyerap keragaman sebesar 95%.

2.11 StandardScaler

StandardScaler merupakan *class* dari *sklearn* untuk melakukan normalisasi data agar data yang digunakan tidak memiliki penyimpangan yang besar. *StandardScaler* menghilangkan mean dan menskalakan ke variansi dengan asumsi data terdistribusi normal (gauss) untuk semua fitur, sehingga setiap fitur akan memiliki nilai rata-rata 0 dan standar deviasi 1.

$$z = \frac{x-u}{s} \quad \dots(2.4)$$

Dengan u adalah mean sampel dan s adalah standar deviaso sampel.

2.12 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* merupakan suatu pendekatan untuk mencari kasus dengan melakukan perhitungan antara kasus baru dengan kasus lama berdasarkan pada pencocokan bobot dari sejumlah fitur yang ada. (Kusrini dan Luthfi, 2009)

K-Nearest Neighbor atau KNN adalah metode yang menggunakan algoritma *supervised* untuk mengklasifikasikan objek berdasarkan data pembelajaran yang memiliki jarak terdekat dengan objek tersebut. Data *train* diestimasikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan setiap fitur dari data. Penentuan dekat atau jauhnya tetangga, biasanya dihitung berdasarkan jarak *Euclidean*.

2.13 Metrik evaluasi

Dalam *deep learning*, terdapat banyak metrik evaluasi yang dapat digunakan untuk mengukur kinerja dari model yang dihasilkan, antara lain:

1. Confusion Matrix

Confusion matrix atau disebut juga *error matrix* yang pada dasarnya memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya. *Confusion Matrix* berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui.

Aktual	Classified as	
	+	-
+	True positives (TP)	False negatives (FN)
-	False positives (FP)	True negatives (TN)

Gambar 2.22 Confusion Matrix

2. Accuracy

Accuracy menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. Nilai *accuracy* dihitung dari rasio antara sampel yang diprediksi benar dengan total jumlah dari sampel (Tharwat, 2018). Nilai akurasi dapat dihitung menggunakan persamaan 2.4, sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad \dots(2.5)$$

Dari persamaan tersebut, terdapat empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion matrix*, yaitu:

a. True Positive (TP)

True Positive merupakan nilai ketika melakukan prediksi positif dan benar positif.

b. True Negative (TN)

Nilai *True Negative* merupakan nilai ketika prediksi yang dilakukan negatif dan benar negatif.

c. False Positive (FP)

False Positive adalah kebalikan dari *True Positive*, yaitu ketika melakukan prediksi positif namun prediksi tersebut salah.

d. False Negative (FN)

False Negative adalah kebalikan dari *True Negative*, yaitu ketika melakukan prediksi negative namun prediksi tersebut salah.

3. F1 Score

F1 Score menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan dari suatu model. Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya (Rosanndy, 2016).

Nilai *precision* dapat dihitung dengan persamaan berikut:

$$Precision = \frac{TP}{TP+FP} \quad \dots(2.6)$$

Recall atau *sensitivity* adalah proporsi kasus positif yang sebenarnya yang diprediksi positif secara benar (Rosanndy, 2016). Nilai *recall* dapat dihitung menggunakan persamaan berikut:

$$Recall = \frac{TP}{TP+FN} \quad \dots(2.7)$$

Perhitungan nilai *F1 Score* dapat dihitung melalui persamaan yang dijabarkan, sebagai berikut:

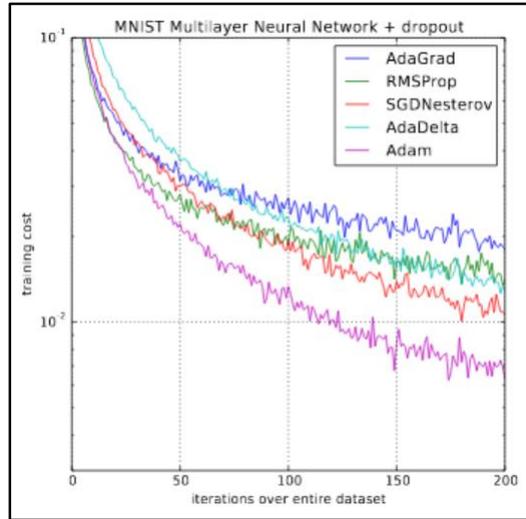
$$\begin{aligned} F1\ Score &= \frac{2 \times precision \times recall}{precision+recall} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned} \quad \dots(2.8)$$

2.14 Adam Optimizer

Optimisasi Adam diperkenalkan oleh Diederik Kingma dari OpenAI dan Jimmy Ba dari *University of Toronto* dalam *paper ICLR 2015* (Diederik P. Kingma, 2019). Adam merupakan metode *learning rate* adaptif, dimana Adam menghitung *learning rate* individu untuk parameter yang berbeda. Adam dapat dikatakan sebagai algoritma optimisasi pengganti untuk *stochastic gradient descent* untuk training model dalam *deep learning* yang menggabungkan sifat-sifat terbaik dari algoritma *AdaGrad* dan *RMSProp* untuk memberikan algoritma yang lebih optimal dimana dapat menangani gradien yang menyebar dan memiliki *noise*. Adam relatif mudah dikonfigurasi di mana parameter konfigurasi *default* bekerja dengan baik pada sebagian besar masalah yang dihasilkan saat *training*.

Nama “Adam” berasal dari “*Adaptive Moment Estimation*” karena Adam menggunakan estimasi gradien momen pertama dan kedua untuk mengadaptasi *learning rate* untuk setiap bobot jaringan saraf sehingga dapat mencapai hasil yang baik dengan cepat. Dalam *paper* berjudul “*Adam: A Method for Stochastic Optimization*”, menunjukkan perbandingan antara Adam dan *optimizer* lainnya saat

training menggunakan algoritma *Multilayer Neural Network* untuk pengenalan digit MNIST memberikan hasil yang ditunjukkan sebagai berikut.

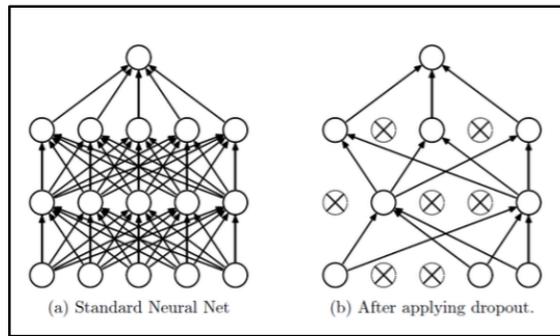


Gambar 2.23 Perbandingan antara Adam dan optimizer lainnya

2.15 Dropout

Dropout adalah teknik yang digunakan untuk mengurangi *overfitting* di dalam *neural network* dengan mencegah adaptasi kompleks pada data *training*. *Dropout* mengacu pada penghapusan unit secara acak dan sementara, baik di dalam *hidden layer* atau *visible layer* dan semua keterhubungan antara *input* dan *output*. Setiap *node* diberikan nilai probabilitas dengan antara 0 sampai 1.

Misalnya, setiap unit *hidden layer* diatur dengan probabilitas 0,5, hal ini berarti terdapat 50% unit dari masing-masing unit yang akan dipertahankan dan kemungkinan 50% akan di *drop* dari *neural network*.



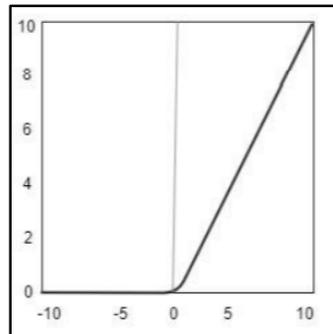
Gambar 2.24 Dropout

2.16 ReLu (*Rectified Linear Unit*)

ReLU merupakan fungsi aktivasi yang dirumuskan sebagai berikut:

$$ReLU(x) = \max(x, 0) \quad \dots(2.9)$$

Dimana x merupakan input neuron. Aktivasi ReLu pada dasarnya merupakan fungsi aktivasi *non-linear* yang paling sederhana. Fungsi aktivasi ReLu digunakan untuk mengubah nilai *minus* (di bawah 0) menjadi nilai 0 sehingga nilai keluaran dari aktivasi bernilai positif (Zhang, Lipton, Li, & Smola, 2019)



Gambar 2.25 Fungsi ReLu

2.17 Python

Python diciptakan pada tahun 1990 oleh Guido Van Rossum di Belanda dengan mengambil nama dari acara televisi favorit Guido, yaitu *Monty Python's Flying Circus*.

Kemudian python digunakan sebagai bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif dan memiliki dokumentasi yang luas.

Bahasa pemrograman Python merupakan salah satu bahasa pemrograman yang bersifat interpreter, interaktif, *object-oriented* dan dapat beroperasi hampir di seluruh *platform*, seperti *Windows*, *UNIX*, *Mac* dan lainnya. Memiliki sintaks yang jelas dan elegan, dikombinasikan dengan menggunakan *module* siap pakai, struktur data tingkat tinggi dan efisien menjadi salah satu keunggulan Python, yakni mudah untuk dipelajari.



Gambar 2.26 Logo Python

Sebagai pemrograman yang berorientasi pada objek (OOP), data yang terdapat dalam python berbentuk objek yang telah dibuat dari kelas (*class*). Selain itu, python juga dapat dikatakan sebagai alat ampuh yang mampu mengembangkan perangkat lunak yang dapat digunakan kembali.

2.18 Library Keras

Keras adalah API *neural network* tingkat tinggi, dibangun dengan Python dan dapat berjalan pada Tensorflow, CTNK atau Theano yang disebut juga Keras *Backends*. Keras membantu jalannya *deep learning* dan kecerdasan buatan.



Gambar 2.27 Keras

Fitur-fitur yang terdapat pada Keras, yaitu:

1. Keras merupakan antarmuka tingkat tinggi yang menggunakan TensorFlow dan Theano sebagai *backend*-nya
2. Keras dapat berjalan lancar di kedua CPU dan GPU
3. Keras mendukung hampir semua jaringan saraf yang selanjutnya dapat dikombinasikan untuk membangun model yang lebih kompleks
4. Keras adalah kerangka kerja berbasis Python yang membuatnya mudah untuk dideteksi dan dipelajari.
5. Keras menyediakan beberapa model untuk CNN, seperti: *interception resnet v2*, *nasnet* dan *densenet* (Li, Li, & Li, 2017)

2.19 Library TensorFlow



Gambar 2.28 TensorFlow

Library ini dikembangkan oleh Google Brain Team dan diperkenalkan pada tahun 2016. Library TensorFlow adalah sebuah *framework* komputasional untuk membuat model *machine learning*. Selain itu, TensorFlow juga menyediakan optimisasi dan algoritma untuk training, sehingga *library* ini dapat mendukung pengembangan *deep learning* (Abadi, 2016). Terdapat berbagai *toolkit* dalam TensorFlow yang memungkinkan pembuatan model pada tingkat abstraksi yang diinginkan, penggunaan API dengan tingkat yang lebih rendah untuk membuat model dengan menentukan serangkaian operasi matematis. Untuk menentukan arsitektur, seperti regresi linear atau jaringan neural dapat menggunakan API dengan tingkat yang lebih tinggi.