



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Malaria

Malaria adalah parasit *Plasmodium* yang ditularkan melalui gigitan nyamuk *anopheles* betina. Terdapat 5 parasit yaitu *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium ovale*, *Plasmodium malariae* dan *Plasmodium knowlesi* (Dave, dkk., 2017).

1. Malaria *falciparum*

Disebabkan oleh *Plasmodium falciparum*. Gejala demam timbul intermiten dan dapat kontinyu. Jenis malaria ini paling sering menjadi malaria berat yang menyebabkan kematian.

2. Malaria *Vivax*

Disebabkan oleh *Plasmodium vivax*. Gejala demam berulang dengan interval bebas demam 2 hari.

3. Malaria *Ovale*

Disebabkan oleh *Plasmodium ovale*. Manifestasi klinis biasanya bersifat ringan. Pola demam seperti pada malaria *Vivax*.

4. Malaria *Malariae*

Disebabkan oleh *Plasmodium malariae*. Gejala demam berulang dengan interval bebas demam 3 hari.

5. Malaria *Knowlesi*

Disebabkan oleh *Plasmodium knowlesi*. Gejala demam menyerupai malaria *Falciparum*.

Penyakit malaria dapat didiagnosis dengan cara pemeriksaan mikroskopik, *Quantitative Buffy Coat*, *Polymerase Chain Reaction* (PCR), serta *Rapid Diagnostic Tests* (RDT) (Kusuma, dkk., 2014).

1. *Quantitative Buffy Coat*

Quantitative buffy coat menggunakan *fluorochrome acridine orange* sebagai pewarna sehingga parasit bisa dideteksi dengan mikroskop *fluorescence*. Pengecatan ini dapat digunakan untuk mendeteksi dan menghitung dengan mikroskop jumlah parasit di hapusan darah dan di lapisan eritrosit (*buffy coat*) pada sampel darah yang sudah disentrifugasi. Cara ini masih relatif mahal untuk banyak tempat, karena kurangnya teknologi sentrifugasi, dan perlu modifikasi pada mikroskop.

2. *Polymerase Chain Reaction* (PCR)

PCR dapat mendeteksi *parasitemia* yang rendah dan identifikasi semua spesies malaria. PCR memerlukan personil terlatih, peralatan khusus, reagen labil, dan lingkungan pemeriksaan yang khusus. Saat ini PCR tidak banyak digunakan untuk diagnosis malaria.

3. *Rapid Diagnostic Tests* (RDT)

Kemampuan RDT pada umumnya ada 2 jenis yaitu mampu mendiagnosis infeksi *Plasmodium falciparum* (*single*) dan mampu mendiagnosis infeksi-infeksi *Plasmodium falciparum* dan *non falciparum* (*combo*). Pemeriksaan RDT memiliki beberapa kekurangan diantaranya ialah tidak mampu menghitung densitas parasitemia, dan kemampuannya kurang optimal pada parasitemia yang rendah. Kualitas alat diagnosis RDT sangat dipengaruhi oleh transportasi dan penyimpanan alat diagnostik.

4. Pemeriksaan Mikroskopik

Pada pemeriksaan mikroskopik, darah yang akan digunakan untuk membuat preparat diambil dari ujung jari manis untuk pasien dewasa, sedangkan pada bayi diambil dari jempol kaki. Terdapat 2 bentuk sediaan yang digunakan yaitu hapusan darah tebal dan hapusan darah tipis. Hapusan darah tebal biasa digunakan untuk mencari parasit malaria. Hapusan darah tipis untuk pemeriksaan malaria dibuat dengan cara yang sama pembuatan hapusan darah rutin untuk evaluasi hematologis. Pemeriksaan mikroskopik akan memberikan informasi mengenai ada tidaknya parasit malaria, menentukan jenis parasit, stadium plasmodium, dan kepadatan parasitemia. Pada pemeriksaan mikroskopik dinyatakan tidak terkena penyakit malaria apabila sudah dilakukan 3 kali pemeriksaan dan hasilnya ialah negatif. Pemeriksaan mikroskopik harus dilakukan oleh tenaga laboratorik yang berpengalaman dalam pemeriksaan parasit malaria dan membutuhkan waktu antara 20-60 menit.

Malaria dapat dicegah dengan upaya meningkatkan kewaspadaan terhadap risiko malaria, mencegah gigitan nyamuk, pengendalian vektor, dan kemoprofilaksis. Pencegahan gigitan nyamuk dapat dilakukan dengan menggunakan kelambu berinsektisida, repelem, kawat kasa nyamuk, dan lain-lain.

2.2 Sel Darah Merah

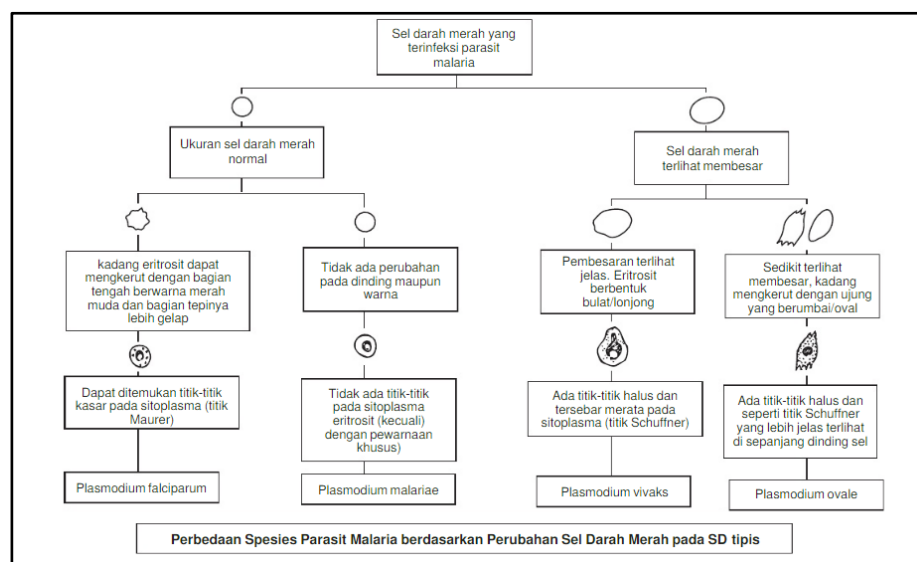
Sel darah merah (*eritrosit*) merupakan salah satu sel darah dengan jumlah paling banyak dibandingkan dengan sel darah lainnya. Sel darah merah matang berbentuk cakram bikonkaf dengan struktur sel yang tidak lengkap dengan tebal 1,5-2,5 mikron. Darah merah normal mempunyai volume 80- 96 femoliter (1fL =

10^{-15} liter) dengan diameter kira-kira 7-8 mikron, sama dengan inti limfosit kecil. Diameter sel darah merah dapat diukur dengan membandingkan sel darah merah dengan inti limfosit kecil dalam bidang yang sama atau berdekatan (Setiawan, 2014).

Sel darah merah hanya terdiri dari membran dan sitoplasma tanpa inti sel. Sel darah merah yang berukuran lebih besar dari inti limfosit kecil pada hapusan darah tepi disebut makrositik. Sel darah merah yang berukuran lebih kecil dari inti limfosit kecil disebut mikrositik.

2.2.1 Sel Darah Merah Malaria

Sel darah merah yang sudah terinfeksi dengan malaria mengalami perubahan (Kementerian Kesehatan Republik Indonesia, 2017). Perubahan tersebut dapat dilihat melalui ukuran sel darah merah yang terinfeksi dapat terlihat membesar atau normal. Pada sitoplasma *eritrosit* yang terinfeksi dapat ditemukan titik *Schuffner* dan *Maurer*.



Gambar 2.1 Sel darah merah malaria (Kementerian Kesehatan Republik Indonesia, 2017:22)

2.3 Klasifikasi Citra

Menurut Apriyanti (2015) klasifikasi citra terdiri dari dua kategori yaitu:

1. *Supervised Classification*

Pengelompokkan piksel-piksel dalam citra yang sebelumnya dianalisis terlebih dahulu untuk menentukan beberapa daerah contoh, kemudian nilai-nilai piksel dalam daerah contoh yang memiliki nilai yang sejenis dikelompokkan.

2. *Unsupervised Classification*

Unsupervised classification merupakan kebalikan dari *supervised classification*, yaitu nilai-nilai piksel dikelompokkan terlebih dahulu oleh komputer menjadi beberapa kelas.

2.4 Artificial Neural Network

Artificial Neural Network (ANN) adalah teknik yang menirukan syaraf manusia yang merupakan bagian fundamental dari otak. Metode ini menggunakan elemen perhitungan *non-linear* dasar yang disebut *neuron* yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan ANN (Russel dan Norvig, 2009:727). *Neuron* memiliki beberapa komponen seperti *input signal* yang dikenal sebagai *dendrites* dan *output signal* yang dikenal sebagai *axons* (Jones, 2015:251).

2.4.1 Komponen Artificial Neural Network

Neural Network terdiri dari beberapa komponen (Saxena, 2017) :

1. *Input Layer*, merupakan bagian *layer* pertama yang berfungsi sebagai menerima semua *input*. Data yang dimasukkan ke dalam *input layer* akan dikirim ke *hidden layer*.

2. *Hidden Layer*, berfungsi untuk memproses data dari *layer* sebelumnya dan bertanggung jawab untuk mengekstraksi fitur yang diperlukan dari *input* data. Sebuah *neural network* tidak perlu memiliki *hidden layer*, *hidden layer* dirancang apabila dibutuhkan saja.
3. *Output Layer*, merupakan *layer* terakhir yang bertugas untuk mengumpulkan dan mentransmisikan data sesuai dengan cara yang telah dirancang.
4. *Connection and weights*, setiap node akan terhubung dengan *node* dari lapisan berikutnya dan setiap koneksi akan memiliki bobot (Ognjanivski, 2015).

2.4.2 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi matematis yang berguna untuk membatasi dan menentukan jangkauan *output* suatu *neuron* (Ervina, 2018).

1. Fungsi Sigmoid

Fungsi *Sigmoid* memiliki nilai *output* pada *range* nol sampai satu $[0,1]$ dan didefinisikan sebagai berikut:

$$f(x) = \frac{1}{1 + e^{-x}} \quad \dots(2.1)$$

2. Fungsi Rectified Linear Unit (ReLU)

Nilai *output* dari *neuron* dinyatakan 0 jika *input*-nya adalah negatif. Jika nilai *input* dari fungsi aktivasi adalah positif, maka *output* dari *neuron* adalah nilai *input* aktivasi itu sendiri.

$$f(x) = \max(0, x) \quad \dots(2.2)$$

2.4.3 Arsitektur Artificial Neural Network

Terdapat 3 arsitektur utama pada ANN, ketiga arsitektur tersebut merupakan (Tony, dkk., 2019):

1. *Perceptron*

Merupakan *neural network* yang paling sederhana, dimana *input layer* langsung terhubung dengan *output layer* dan tidak terhubung sebaliknya.

2. *Multi-Layer Feedforward*

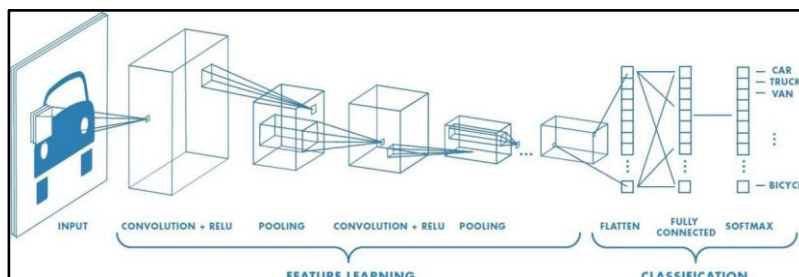
Pada *multi-layer feedforward* terdapat *hidden layer*, dimana *input layer* akan terhubung langsung dengan *hidden layer* dan tidak sebaliknya lalu *hidden layer* terhubung langsung dengan *output layer* dan tidak sebaliknya.

3. *Recurrent Network*

Pada dasarnya *recurrent network* memiliki sifat yang sama dengan *feedforward network*, tetapi pada *recurrent network* terdapat satu syarat yaitu harus memiliki satu *feedback loop*.

2.5 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan metode pengembangan dari metode *Artificial Neural Network* untuk memproses data. Pada CNN digunakan operasi konvolusi sebagai pengganti operasi perkalian matriks (Goodfellow, dkk., 2016:9). CNN adalah *neural network* dimana sinyal dimasukkan ke dalam *convolutional pooling layer* (*convpool layer*) dan *output* dari lapisan terakhir akan dimasukkan ke dalam *fully connected layer* dan *softmax layer* (Venkatesan dan Li, 2017).



Gambar 2.2 Arsitektur *convolutional neural network* (Saha, 2018)

2.5.1 Feature Learning

Feature learning terdiri dari beberapa *layer* yaitu:

1. *Convolutional Layer*

Convolutional layer berfungsi untuk mengekstraksi fitur dari sebuah citra. Proses di dalam *convolutional layer* dilakukan dengan mengambil *sub-region* seukuran *filter* lalu dilakukan operasi konvolusi antara *sub-region* dengan *filter* (Alcander, 2019). Operasi konvolusi dapat dituliskan sebagai berikut.

$$(I * f)_{xy} = \sum_{i=1}^h \sum_{j=1}^w f_{ij} * I_{x_i+i-1,y+j-1} \quad \dots(2.3)$$

Variabel I merupakan *input* citra dua dimensi dan f merupakan *filter* berukuran $h \times w$. Pengambilan *sub-region* pertama dengan *sub-region* berikutnya diatur dengan nilai yang biasa disebut dengan *strides*. Berikut merupakan komponen-komponen utama pada *convolutional layer* (Patterson & Gibson, 2017):

a. *Filter*

Filter adalah fungsi yang memiliki lebar dan tinggi lebih kecil dari pada lebar dan tinggi volume *input*. *Filter* diaplikasikan pada lebar dan tinggi data masukan dengan cara *sliding window*. *Filter* juga diterapkan untuk setiap kedalaman masukan. *Filter* menghitung keluaran dengan menghasilkan *dot product* dari *filter* pada wilayah masukan.

b. *Activation Map*

Dengan menggeser setiap *filter* sepanjang dimensi spasial dari data masukan akan menghasilkan keluaran dengan dua dimensi yang disebut dengan *activation map* untuk setiap *filter*.

c. *Layer-specific Hyperparameters*

- *Filter size*, setiap *filter* memiliki daerah spasial yang kecil berdasarkan lebar dan tinggi dari *filter size*.
- *Stride*, *hyperparameter* ini menentukan seberapa jauh pergeseran dari pergerakan *filter window* setiap pengaplikasian fungsi *filter*.
- *Zero-padding*, *hyperparameter* ini dapat mengontrol besarnya spasial dari volume keluaran, hal ini digunakan untuk mempertahankan ukuran spasial dari volume masukan dalam volume keluaran.
- *Output Depth*, ukuran kedalaman pada *output* ditentukan dari seberapa besar kedalaman tinggi dan lebar dari *filter* yang digunakan.

2. *Pooling Layer*

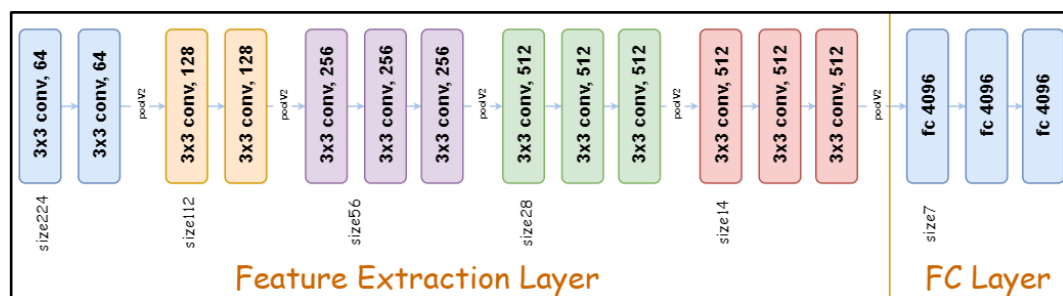
Pooling layer digunakan untuk mengurangi parameter dan jumlah perhitungan yang dibutuhkan untuk *training network* (Murphy, 2016). Terdapat dua jenis *pooling layer*, yaitu *max pooling* dan *average pooling* (Saha, 2018). *Max pooling* mengembalikan nilai tertinggi dari bagian gambar yang dicakup oleh *filter* atau *kernel*. *Average pooling* mengembalikan nilai rata-rata dari bagian gambar yang dicakup oleh *filter* atau *kernel*.

2.5.2 Fully Connected Layer

Fully connected layer pada umumnya digunakan pada *layer* akhir CNN. Pada *fully connected layer*, seluruh *neuron* akan saling terhubung satu sama lain seperti pada *feedforward neural network* dan dilakukan proses klasifikasi yang hasilnya akan menjadi *output* dari CNN (Tarawan, 2019).

2.5 Pre-trained Model VGG16

Pada penelitian ini menggunakan *pre-trained* model dengan arsitektur *Visual Geometry Group 16* (VGG16). Arsitektur VGG16 dirancang oleh Visual Geometry Group yang berasal dari Universitas Oxford. VGG16 memiliki jumlah *layer* sebanyak 16 *layer* yang terdiri dari 1 neuron masukan, 5 blok konvolusi yang terdiri dari *layer convolution*, ReLu, dan *pooling*, dan 3 *dense layer* (Adinegoro, 2019). Gambar 2.3 memperlihatkan arsitektur dari VGG16. Dengan menggunakan implementasi *pre-trained* model yang disediakan oleh Keras, *pre-trained* model digunakan sebagai *feature extractor* tanpa *fine-tuning* (Kieffer, dkk., 2017). *Pre-trained* model yang disediakan telah dilatih dengan menggunakan dataset ImageNet (Akçay, 2016).

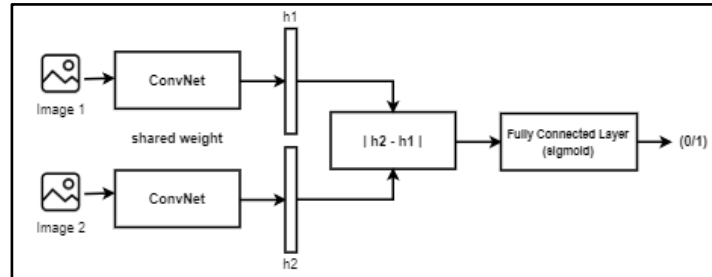


Gambar 2.3 Arsitektur VGG16

2.6 Siamese Convolutional Neural Network

Siamese Convolutional Neural Network (SCNN) digunakan untuk mempelajari kesamaan metrik dengan cara melatih jaringan dengan menggunakan CNN yang memiliki bobot yang sama (LeCun, dkk., 2005). Data *input* yang dimasukkan ke dalam sistem berupa gambar yang berpasangan dan label. SCNN mempunyai dua buah *input* untuk membandingkan dua pola dan mempunyai satu buah *output* berupa nilai dari kesamaan antara kedua pola (LeCun, dkk., 1994). Pada SCNN dilakukan beberapa proses seperti *feature learning*, *flatten*, dan perhitungan jarak

antar dua gambar untuk mengetahui kesamaan antar kedua gambar. Pada penelitian ini, perhitungan jarak dilakukan dengan menggunakan *manhattan distance*. Alur kerja SCNN dapat dilihat pada Gambar 2.4.



Gambar 2.4 Alur kerja *Siamese Convolutional Neural Network*

2.7 Manhattan Distance

Pengukuran *distance* dibutuhkan untuk mengukur kesamaan atau perbedaan suatu data. Salah satu *distance function* yang dapat digunakan yaitu *manhattan distance*. *Manhattan distance* dibutuhkan untuk pengukuran jarak antar dua *image* yang digunakan pada saat pembuatan data *pair*. Cara perhitungannya yaitu total nilai absolut dari selisih antara koordinat 1 dan koordinat 2 (Hoetama, 2018). Berikut merupakan persamaan *manhattan distance*.

$$dist(x, y) = |x_{ik} - x_{jk}| \quad \dots(2.4)$$

Keterangan:

$dist(x, y)$: Jarak manhattan *distance*

(x_i) : *record* ke-*i*

(x_j) : *record* ke-*j*

k : urutan data(1,2,3, ... n)

2.8 Metrik Evaluasi

Metrik evaluasi yang sering digunakan untuk klasifikasi adalah sebagai berikut:

1. *Confusion Matrix*

Confusion Matrix adalah *tools* yang digunakan untuk evaluasi model klasifikasi untuk memperkirakan objek yang benar atau salah. Sebuah *matrix* dari prediksi yang akan dibandingkan dengan kelas yang asli dari masukkan atau dengan kata lain berisi informasi nilai aktual dan prediksi pada klasifikasi (Utami, 2016).

		Predicted class	
		Negatives	Positives
Actual class	Negatives	<i>a</i>	<i>b</i>
	Positives	<i>c</i>	<i>d</i>

Gambar 2.5 *Confusion matrix* (Novaković, dkk., 2017)

2. *Accuracy*

Accuracy ialah ukuran kinerja paling intuitif dan yang akan dihitung hanya jumlah pengamatan yang diprediksi dengan benar terhadap total pengamatan. Perhitungan akurasi dapat dilakukan dengan cara membagi jumlah data yang diklasifikasi secara benar dengan total *sample data testing* yang akan dilakukan pengujian (Utami, 2016).

$$Accuracy = \frac{a+d}{a+b+c+d} \quad \dots(2.5)$$

Keterangan:

a : *True Negatives*

d : *True Positives*

b : *False Negatives*

c : *False Positives*