KLASIFIKASI PENYAKIT MALARIA BERDASARKAN CITRA SEL DARAH MENGGUNAKAN RESNET34

KERJA PRAKTEK

Diajukan untuk Memenuhi Persyaratan Akademik dalam

Menyelesaikan Pendidikan pada Program Studi

S1 Teknik Informatika Universitas Kristen Maranatha

Oleh

**Stefanus Hermawan**

**1772023**



**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS KRISTEN MARANATHA**

**BANDUNG**

**2020**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc40086685)

[BAB 1 PENDAHULUAN 5](#_Toc40086686)

[1.1 Latar Belakang 5](#_Toc40086687)

[1.2 Rumusan Masalah 6](#_Toc40086688)

[1.3 Tujuan Pembahasan 6](#_Toc40086689)

[1.4 Ruang Lingkup 6](#_Toc40086690)

[1.5 Sumber Data 6](#_Toc40086691)

[1.6 Sistematika Penyajian 7](#_Toc40086692)

[BAB 2 KAJIAN TEORI 8](#_Toc40086693)

[2.1 *Residual Neural Networks* 8](#_Toc40086694)

[*2.2* *Learning Rate Finder* 8](#_Toc40086695)

[*2.3* *FastAi* 9](#_Toc40086696)

[2.4 *Flask* 9](#_Toc40086697)

[2.5 Google Colaboratory 10](#_Toc40086698)

[2.6 Freelancer – Bootstrap Theme 10](#_Toc40086699)

[2.7 Dataset 11](#_Toc40086700)

[BAB 3 ANALISIS DAN RANCANGAN SISTEM 12](#_Toc40086701)

[*3.1* *Data Preprocessing* 12](#_Toc40086702)

[3.2 Pelatihan Model 12](#_Toc40086703)

[3.3 Cara Kerja Sistem 13](#_Toc40086704)

[3.4 Rancangan Tampilan Website 14](#_Toc40086705)

[3.4.1 Rancangan Tampilan Utama 14](#_Toc40086706)

[3.4.2 Rancangan Tampilan *Prediction Result* 15](#_Toc40086708)

[BAB 4 IMPLEMENTASI 17](#_Toc40086709)

[4.1 Implementasi Pelatihan Model 17](#_Toc40086710)

[4.2 Implementasi Website 17](#_Toc40086711)

[4.2.1 Struktur Folder Website 17](#_Toc40086713)

[4.2.2 Implementasi Tampilan Website 18](#_Toc40086714)

[BAB 5 PENGUJIAN 21](#_Toc40086716)

[5.1 Sub Topik Bab 5 21](#_Toc40086717)

[5.2 Pengujian *Black Box* 21](#_Toc40086718)

[5.3 Pengujian *White* *Box* 21](#_Toc40086719)

[5.4 *User* *Acceptance* *Test* 21](#_Toc40086720)

[5.5 *Benchmarking* 21](#_Toc40086721)

[BAB 6 SIMPULAN DAN SARAN 22](#_Toc40086722)

[6.1 Simpulan 22](#_Toc40086723)

[6.2 Saran 22](#_Toc40086724)

[DAFTAR PUSTAKA 23](#_Toc40086725)

# PENDAHULUAN

## Latar Belakang

Menurut data World Health Organization pada tahun 2017, terdapat 219 juta kasus penyakit malaria yang tersebar di 90 negara, diperkirakan penderita malaria yang meninggal sebanyak 435.000 kasus dan kasus terbanyak berada di wilayah Afrika yang tersebar sebanyak 92% kasus malaria dunia dan 93% kasus penderita yang meninggal [1].

Sedangkan di Indonesia, menurut Kementerian Kesehatan Republik Indonesia pada tahun 2017, terdapat 42% wilayah Indonesia yang belum terbebas dari penyakit malaria, dan jumlah kasus malaria tercatat sebanyak 261.617 kasus malaria secara nasional dengan kasus wilayah Papua, Papua Barat, dan NTT dengan total 10,7 juta penduduk [2].

Parasit malaria dapat diidentifikasi dengan cara memeriksa darah pasien dengan menggunakan mikroskop. Sebelum pemeriksaan spesimen, sel darah akan terlebih dahulu diwarnai agar parasit pada sel darah dapat dibedakan.

Citra sel darah merah diperoleh melalui penangkapan citra menggunakan kamera ponsel cerdas berbasis Android yang dilengkapi dengan optik mikroskop medan terang (*bright-field microscopy*) [3].

Maka dari itu dalam kerja praktek ini akan dibahas bagaimana cara mendeteksi penyakit malaria menggunakan Convolutional Neural Networks dari citra digital sel darah serta mengembangkan *website* sebagai tampilan antarmuka untuk mempermudah penggunaan bagi pengguna.

## Rumusan Masalah

Berdasarkan latar belakang masalah di atas, rumusan masalah yang dapat dirumuskan adalah:

1. Bagaimana cara model *Convolutional Neural Networks* untuk mendeteksi malaria?
2. Bagaimana cara mengembangkan *website* sebagai tampilan antarmuka pengguna untuk model *Convolutional Neural Networks*?
3. Bagaimana cara mengukur kinerja model dalam mendeteksi malaria?

## Tujuan Pembahasan

Adapun tujuan pembahasan dari Kerja Praktek ini adalah :

1. Menggunakan ResNet34 untuk membangun model *Convolutional Neural Networks* untuk mendeteksi malaria
2. Mengembangkan website sebagai tampilan antarmuka pengguna untuk model ResNet34
3. Melaksanakan penilaian kinerja terhadap model dengan sekumpulan data citra sel darah

## Ruang Lingkup

Ruang lingkup yang akan dibahas dalam laporan Kerja Praktek ini adalah :

1. Menggunakan model ResNet34 yang disediakan FastAi untuk melatih model
2. Mengembangkan website sebagai antarmuka untuk pengguna
3. Data untuk melatih dan mengukur kinerja model adalah citra sel darah yang diambil dari website U.S National Library of Medicine.

## Sumber Data

Dataset yang digunakan untuk melatih dan mengukur kinerja model adalah sekumpulan citra yang telah dikumpulkan dan telah diberi label oleh U.S National Library of Medicine [4].

Model yang digunakan adalah model ResNet34 [5] yang terdapat dalam pustaka FastAi [6].

## Sistematika Penyajian

Sistematika penyajian laporan kerja praktek ini diuraikan sebagai berikut. Bab pertama diisi dengan pendahuluan yang mencakup latar belakang masalah, rumusan masalah, tujuan pembahasan, ruang lingkup, sumber data, serta sistematika penyajian, bab kedua diisi kajian teori, bab ketiga diisi dengan analisis dan perancangan sistem model yang digunakan, bab keempat diisi dengan implementasi model yang telah dilatih, bab kelima diisi dengan pengujian model, dan pada bab keenam diisi dengan kesimpulan dan saran.

# KAJIAN TEORI

## *Residual Neural Networks*

C*onvolutional Neural Networks (CNN)* merupakan model yang cocok diterapkan dalam kasus klasifikasi, tetapi arsitektur CNN dengan lapisan (*layers)* yang semakin dalam menyebabkan penurunan kinerja yang cukup signifikan karena akan menyebabkan hilang atau membesarnya nilai *gradient* (*vanishing gradient*) [5].

Untuk mengatasi penurunan kinerja ini, Kaiming He, Xiangyu Zhang, Shaoqing Ren, dan Jian Sun pada tahun 2015 menemukan solusi untuk mengatasi penurunan kinerja model CNN dengan menerapkan *skip connection/shortcut* yang kemudian dipublikasikan dengan nama *Residual Neural Networks (ResNet)* [5].

ResNet menjadi pemenang dalam ImageNet Challenge 2015 karena mampu melatih jaringan syaraf sebanyak 152 lapisan dengan baik serta lebih mudah untuk dioptimalkan, dan memperoleh akurasi yang meningkat secara signifikan.

Dalam kerja praktek ini akan menggunakan *pre-trained model Residual Neural Networks 34* atau yang juga disebut ResNet34 yang disediakan oleh *library* FastAi.

## *Learning Rate Finder*

*Learning Rate* merupakan salah satu *hyperparameter* yang penting dan perlu ditentukan sebelum dilakukan proses *training* pada model karena hal ini akan mempengaruhi performa model secara signifikan ketika dilakukan proses *training. Learning rate* yang optimal akan menurunkan tingkat *loss function* sehingga *training* dapat mencapai tingkat akurasi yang optimal.

*Learning rate* yang diatur dengan angka yang sangat kecil, akan memperlambat proses *training* karena memakan waktu yang lama serta sumber daya komputasi yang besar, sedangkan jika diatur dengan angka yang terlalu besar akan menyebabkan tingkat akurasi yang rendah.

  
Gambar 1.1 Learning Rate [7]

Dalam kerja praktek ini akan menggunakan *Cyclical Learning Rates (CLR)* yang dikembangkan oleh Leslie Smith [8] yang dalam implementasi akan menggunakan fungsi yang telah tersedia dalam *library* FastAI. Dalam metode ini, *learning rate* tidak akan ditetapkan secara eksak tetapi akan diperbaharui secara berkala dengan variasi nilai yang telah ditentukan batasan berupa nilai minimum dan maksimumnya.

## *FastAi*

FastAi adalah *library deep learning* yang tersedia secara *open source* dan dapat diinstallsecara langsung dengan menggunakan *Package Manager* bahasa pemrograman Python [6]. FastAi dapat digunakan secara langsung untuk melatih model *deep learning* yang telah disediakan (*pre-trained)* serta memiliki dokumentasi yang mudah untuk dipahami.

Dalam kerja praktek ini, FastAi digunakan sebagai library utama untuk mengakses dan melatih pre-trained model ResNet34 untuk klasifikasi penyakit malaria.

## *Flask*

Flask merupakan *micro web* *framework* yang ditulis menggunakan bahasa pemrograman Python. Flask disebut *Micro Framework* karena Flask hanya menyediakan fungsi dasar dalam pengembangan web sehingga aplikasi web lebih mudah dikembangkan secara sederhana tetapi fungsionalitas lain juga dapat dikembangkan dan fungsionalitas tersebut dapat ditambahkan sebagai *extension* sesuai dengan kebutuhan [9].

Flask yang digunakan dalam kerja praktek ini untuk mengembangkan antarmuka pengguna website dan secara bersamaan dengan bahasa pemrograman HTML dan CSS akan dikoneksikan dengan model ResNet34 yang telah dilatih untuk mendeteksi malaria dari citra digital yang diunggah oleh pengguna.Flask merupakan *micro web* *framework* yang ditulis menggunakan bahasa pemrograman Python. Flask disebut *Micro Framework* karena Flask hanya menyediakan fungsi dasar dalam pengembangan web sehingga aplikasi web lebih mudah dikembangkan secara sederhana tetapi fungsionalitas lain juga dapat dikembangkan dan fungsionalitas tersebut dapat ditambahkan sebagai *extension* sesuai dengan kebutuhan [9].

Flask yang digunakan dalam kerja praktek ini untuk mengembangkan antarmuka pengguna website dan secara bersamaan dengan bahasa pemrograman HTML dan CSS akan dikoneksikan dengan model ResNet34 yang telah dilatih untuk mendeteksi malaria dari citra digital yang diunggah oleh pengguna.

## Google Colaboratory

Google Colaboratory merupakan produk dari Google Research yang memberikan akses kepada pengguna untuk menulis dan menjalankan kode berbasis bahasa pemrograman Python melalui *browser* yang cocok untuk menjalankan kode program seperti machine learning, analisis data serta kebutuhan untuk edukasi [10].

Secara teknis Google Colaboratory memberikan layanan aplikasi Jupyter Notebook bagi pengguna dan memberikan akses gratis ke sumber daya komputasi berupa GPU Nvidia K80s dan TPU yang memiliki batasan *timeout* 12 jam.

Dalam kerja praktek ini, Google Colaboratory digunakan untuk menjalankan kode program untuk melatih dan melakukan analisis terhadap model ResNet34 dan dataset sel darah merah.

## Freelancer – Bootstrap Theme

Freelancer adalah template HTML CSS dengan desain *One Page Website* yang memiliki fitur seperti bagian profil, bagian portofolio berbentuk *grid* yang responsive, *window modal* untuk menampilkan detail setiap item portfolio dan formulir kontak berbasis PHP [11].

Template Freelancer dikembangkan oleh David Miller selaku pemilik Blackrock Digital LLC menggunakan *framework* Bootstrap yang kemudian dirilis dibawah lisensi MIT secara *open source*.

Dalam kerja praktek ini, template Freelancer akan digunakan sebagai struktur tampilan utama *website* dan dilakukan penyesuaian sesuai dengan kebutuhan tampilan *website*.

## Dataset

Dataset yang digunakan dalam kerja praktek ini diambil dari website US National Library of Medicine. Citra digital sel darah diambil menggunakan mikroskop optik medan cahaya yang terpasang di perangkat Android dan diberikan label oleh peneliti Lister Hill National Center for Biomedical Communications (LHNCBC) yang merupakan bagian dari US National Library of Medicine (NLM).

Dataset citra sel darah merah yang telah dikumpulkan sebanyak 27.558 citra digital yang telah dibagi menjadi 2 label yaitu parasitized dan uninfected dengan jumlah citra sel darah merah yang berimbang.

Publikasi dataset yang dipakai dalam kerja praktek ini [12], He et al (2016) membandingkan beberapa pre-trained model Convolutional Neural Networks (CNN) seperti AlexNet, VGG-16, ResNet-50, Xception, DenseNet-121 serta model CNN yang dikustomisasi dengan tiga layer konvolusional (convolutional layers) dan dua layer yang terhubung penuh (fully connected layers). Hasil dari penelitian ini menunjukkan bahwa pre-trained model ResNet-50 memiliki tingkat akurasi paling tinggi.

# ANALISIS DAN RANCANGAN SISTEM

## *Data Preprocessing*

*Data Preprocessing* dataset citra sel darah merah dilakukan dengan membagi dataset citra sel darah merah menjadi 3 bagian yaitu 80% dataset digunakan dalam *training set,* 10% untuk *test set* dan 10% untuk *validation set* yang dipisahkan menjadi 3 *folder* secara acak dan berimbang menggunakan pustaka *split\_folders* [13]dengan ukuran *batch* sebesar 64.

Ukuran setiap citra pada dataset diubah menjadi 224 x 224 piksel, ditransformasikan dengan melakukan augmentasi data berupa citra sel darah yang diputar secara vertikal dan horizontal serta dilakukan normalisasi dataset berdasarkan statistika tiga channel RGB dari dataset ImageNet.

Semua proses *preprocessing* dilakukan menggunakan *library* FastAi yaitu fungsi *ImageDataBunch* [14] dan citra sel darah merah yang telah dikumpulkan oleh U.S National Library of Medicine [4] telah diberikan label berupa citra sel darah yang telah dipisahkan dalam 2 *folder* yaitu *parasitized* dan *uninfected*.

## Pelatihan Model

Setelah proses *data preprocessing* selesai dilakukan, kemudian dilanjutkan dengan menyiapkan model dengan cara memanggil fungsi *cnn\_learner* pada *library fastai* dan memasukkan parameter berupa data hasil *preprocessing*, arsitektur *pretrained* model ResNet34, dan metrik akurasi.

Sebelum dilakukan pelatihan pada model, dilakukan terlebih dahulu pencarian *learning rate* yang optimal dengan menggunakan fungsi *lr\_find*. Setelah *plot* berupa grafik hasil pencarian *learning rate* ditampilkan, pemilihan *learning rate* didasarkan dengan memilih *learning rate* yang mempunyai garis grafik sebelum tingkat *loss* naik dengan tajam.

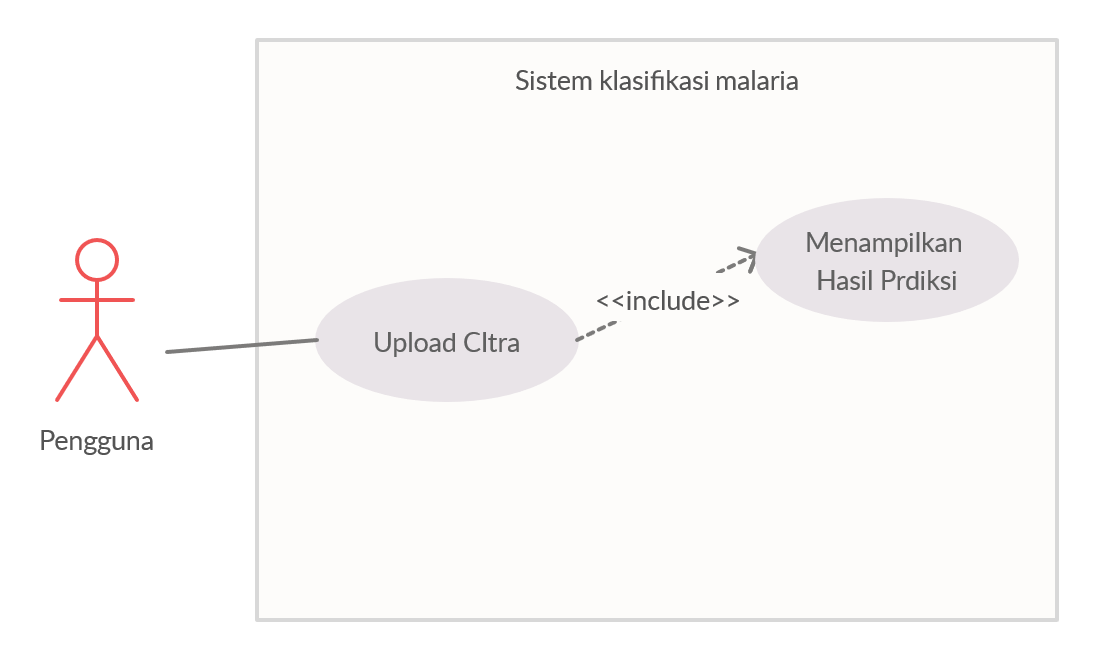
Proses pelatihan model dilakukan dengan menggunakan fungsi *fit\_one\_cycle* dengan parameter besaran *epoch* dan *learning rate* yang sebelumnya sudah ditentukan berdasarkan hasil pencarian.

## Cara Kerja Sistem

Sistem yang akan dikembangkan merupakan aplikasi web dengan menggunakan *framework* Flask dan FastAi sebagai *backend* aplikasi. Citra sel darah merah yang diunggah oleh pengguna akan disimpan ke dalam *folder* lalu akan langsung diproses oleh fungsi prediksi menggunakan fungsi pada *library* FastAi yang kemudian akan mengembalikan nilai hasil prediksi citra sel darah merah ke tampilan *website*.

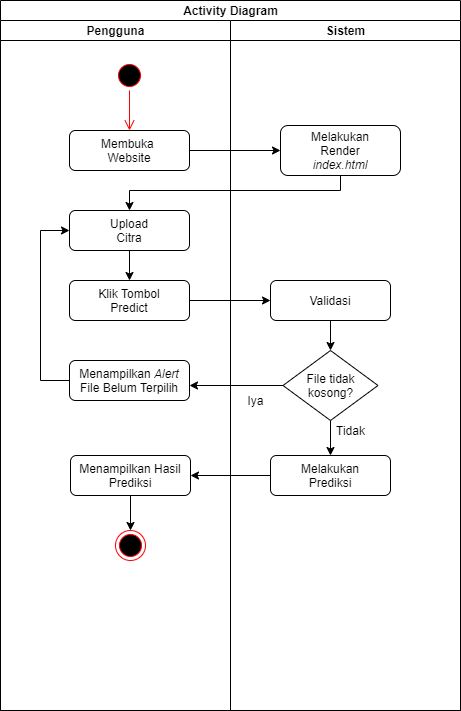
Kode program sistem akan dimasukkan ke dalam sebuah *file* yaitu *app.py. file app.py* akan bertugas untuk mengatur navigasi halaman *website* serta bertugas untuk memanggil fungsi untuk melakukan pengunggahan, menyimpanan, dan memprediksi *file* citra sel darah merah.

Gambar 3.1 menampilkan *use case diagram* dimana pengguna dapat mengunggah citra sel darah merah pada *website* yang kemudian disimpan ke dalam *folder* pada *server* dan kemudian akan menampilkan hasil prediksi dari citra sel darah merah yang diunggah.



Gambar 3.1 Use Case Diagram sistem

Gambar 3.2 menunjukkan *activity diagram* pada aplikasi *website* yang jika dijalankan, *app.py* akan melakukan *render* halaman *index.html* sebagai halaman utama *website*. Ketika pengguna melakukan pengunggahan *file*, maka sistem akan melakukan validasi berupa pemeriksaan apakah *file* yang diunggah sudah dipilih atau tidak. Jika pengguna belum memilih *file* untuk diunggah, maka sistem akan menampilkan peringan berupa meminta pengguna untuk memilih *file* yang akan diunggah. Kemudian sistem akan melakukan prediksi pada *file* citra yang telah diunggah dan menampilkan hasil prediksi pada halaman *index.html* berupa *popup window* yang menampilkan citra yang diprediksi, skor hasil prediksi, serta klasifikasi hasil prediksi.



Gambar 3.2 *Activity Diagram* Sistem

## Rancangan Tampilan Website

Rancangan tampilan antarmuka pengguna website dibuat sederhana dan menarik untuk mempermudah dalam pengunaan website ini.

## Rancangan Tampilan Utama

Pada Gambar 3.3 menunjukkan rancangan tampilan halaman utama *website* yang terdiri dari bagian untuk fitur utama yaitu klasifikasi citra digital malaria dan bagian *about.*

Pada bagian klasifikasi citra digital malaria, pengguna dapat melakukan pengunggahan citra digital sel darah merah kemudian dengan menekan tombol *predict* yang akan memunculkan tampilan *modal window* berupa hasil prediksi yang dilakukan.

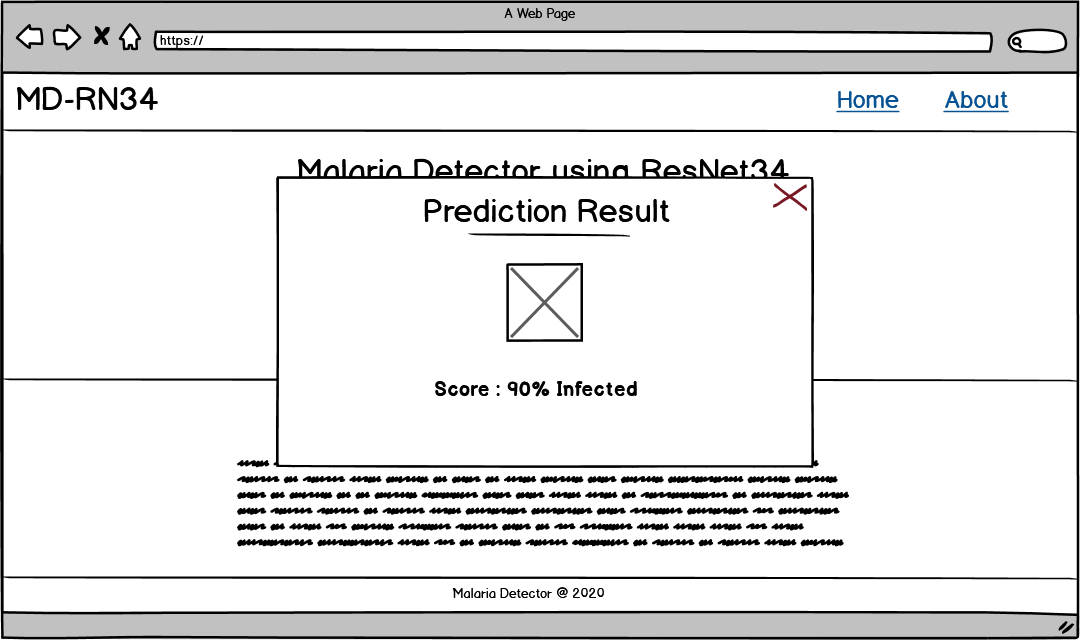
Pada bagian *about*, pengguna dapat membaca detail tentang tujuan dari *website* dibuat, penjelasan tentang penyakit malaria, detail tentang algoritma *ResNet34* serta hasil pelatihan model yang digunakan dalam *website*.

## 

Gambar 3.3 Rancangan Tampilan Utama

## Rancangan Tampilan *Prediction Result*

Gambar 3.4 menunjukkan hasil dari prediksi model *ResNet34* yang ditampilkan berupa *pop-up window* yang berisi citra digital sel darah yang telah diunggah dan skor hasil prediksi. Jika hasil prediksi menunjukkan kemungkinan terinfeksi lebih besar dari 50% maka *website* akan menampilkan hasil prediksi berupa teks “*XX% infected”* dan jika hasil prediksi menunjukkan kemungkinan tidak terinfeksi lebih besar dari 50% maka website akan menampilkan hasil prediksi berupa teks “*XX% uninfected*”.



Gambar 3.4 Tampilan Result

# IMPLEMENTASI

## Implementasi *Data Preprocessing*

## Pada Gambar 4.1 menunjukkan proses pembagian dataset citra sel darah merah secara acak menggunakan pustaka *split\_folder* yang bertujuan untuk menghilangkan bias pada saat pelatihan model agar menghindari *overfitting*.

A screenshot of a cell phone

Description automatically generated

Gambar 4.1 Split Dataset

Pada Gambar 4.2 menunjukkan proses transformasi dataset citra sel darah merah yang telah dibagi menjadi 3 bagian yaitu *training set* sebanyak 22.046 citra*, validations set* sebanyak 2.756 citra*,* dan *test set* sebanyak 2.756 citra. Pada *test set* tidak diberikan label karena digunakan sebagai data percobaan setelah melakukan pelatihan dan validasi untuk mengukur akurasi model pada data yang belum pernah dilihat sebelumnya.

A screenshot of a social media post

Description automatically generated

Gambar 4.2 Data Preprocessing

## Implementasi Pelatihan Model

Berikut merupakan proses pelatihan model yang dilakukan menggunakan Google Colaboratory dengan perangkat GPU yang memiliki batasan berupa *time-out* setelah 12 jam penggunaan.

Gambar 4.3 merupakan proses pendefinisian model ResNet34 dengan data yang telah dilakukan *preprocessing* sebelumnya serta melakukan pencarian *learning rate* untuk melakukan tahapan pelatihan model pertama.

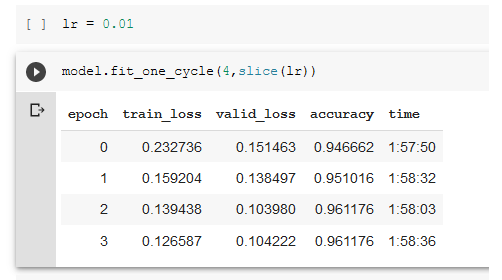
A picture containing screenshot

Description automatically generated

Gambar 4.3 Pencarian Learning Rate 1

Proses pelatihan model pada Gambar 4.4 dengan epoch sebanyak 4 dan *learning rate* berupa *slice* dengan nilai maksimal 0.01 yang ditentukan dengan cara memilih angka pada graf pencarian *learning rate* sebelum *loss* meningkat. Penentuan epoch sebesar 4 ditentukan berdasarkan dengan perkiraan waktu pelatihan yang dilakukan karena adanya batasan penggunaan GPU dari Google Colaboratory.

Hasil pelatihan model pada Gambar 4.4 menunjukkan *training loss* sebesar 12.65%, *validation loss* sebesar 10.42% dan akurasi sebesar 96.11%.



Gambar 4.4 Pelatihan Model 1

Setelah proses pelatihan pertama yang melatih unit dari *layer* terakhir dari model, pada Gambar 4.5 dilakukan *unfreeze* pada model dan dilakukan pencarian *learning rate* kembali sebelum dilakukan pelatihan tahap kedua. Proses pelatihan dengan melakukan *unfreeze* pada semua *layer* model agar semua *layer* pada model bisa dilatih untuk meningkatkan akurasi pada model.

A screenshot of a cell phone

Description automatically generated

Gambar 4.5 Pencarian Learning Rate 2

Pada Gambar 4.6 menunjukkan hasil pelatihan dengan *slice* *learning rate* antara 0.0001 sampai dengan 0.002 guna memperlambat proses pelatihan untuk mencapai akurasi yang lebih tinggi. Hasil dari pelatihan tahap kedua yaitu akurasisebesar 96.91%.

A screenshot of a cell phone

Description automatically generated

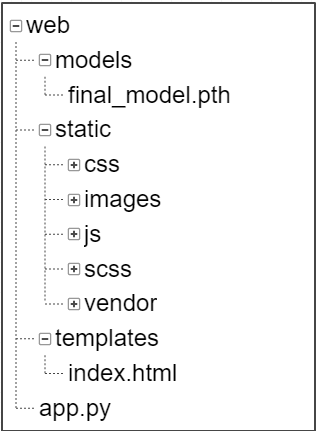
Gambar 4.6 Pelatihan Model 2

## Implementasi *Website*

## Implementasi *website* yang dilakukan mulai dari mengimplementasikan tampilan *website* yang sudah dirancang pada Bab 3 serta dilakukan implementasi rancangan sistem berupa kode program *backend* yang bertugas untuk menjalankan fungsi – fungsi untuk memprediksi dan mengembalikan nilai hasil prediksi.

## 

## Struktur Folder *Website*



Gambar 4.7 Diagram Struktur *Website*

Pada struktur *folder* *website* ini terdapat isian dari *folder* dan *file* sebagai berikut :

1. *Folder Models*

*Folder Models* berfungsi untuk menampung *file* model berekstensi *.pth* hasil pelatihan yang akan digunakan untuk memprediksi citra sel darah merah pada *website*.

1. *Folder Static*

*Folder Static* berfungsi untuk menampung *file* pendukung seperti *file* CSS, Javascript, serta terdapat folder *images* yang berfungsi untuk menampung citra sel darah merah yang diunggah oleh pengguna.

1. *Folder Templates*

*Folder Templates* berfungsi sebagai *folder* penampung dari *file* *index.html* yang berfungsi sebagai tampilan utama pada *website*.

1. *File app.py*

*File app.py* merupakan *file* yang berisi kode program untuk menjalankan fungsi utama pada *website* seperti fungsi yang bertugas untuk mengunggah, memprediksi serta nengembalikan nilai hasil prediksi citra sel darah merah.

## Implementasi *Backend Website*

## Fungsi yang dijalankan oleh aplikasi web ditunjukkan pada Gambar 3.2 kolom sistem memiliki 3 proses utama untuk memproses dan melakukan prediksi pada citra yang telah di unggah. Berikut ini adalah penjelasan kode program pada *file app.py* yang telah dipisah menjadi dua bagian agar dengan mudah untuk dijelaskan.

Kode program pada Gambar 4.8 menunjukkan pengimporan pustaka untuk mendukung program berjalan pada baris kode 1 – 6. Pada baris 8 – 10 dilakukan pendefinisan variabel *UPLOAD\_FOLDER* sebagai lokasi penyimpanan citra yang telah diunggah dan mendefinisikan variabel *app* sebagai Flask guna menjalankan fungsi – fungsi bawaan pada *framework* Flask, kemudian pada baris 12 – 21 dilakukan pendefinisian variabel *data* sebagai bentuk data citra yang akan digunakan untuk prediksi serta pendefinisian variabel *model* sebagai variabel yang memuat model yang telah dilatih untuk selanjutnya bisa melakukan prediksi pada citra yang telah diunggah.

A screenshot of a social media post

Description automatically generated

Gambar 4.8 Kode Program 1

Gambar 4.9 menunjukkan fungsi utama untuk menjalankan fungsi unggah citra dan fungsi prediksi. Baris 23 – 25 merupakan fungsi prediksi menggunakan fungsi pada variabel *model* yang sebelumnya telah didefinisikan dengan memasukkan parameter lokasi dari direktori citra. Fungsi *index()* pada baris 27 – 29 akan dijalankan untuk melakukan *render* halaman utama ketika pengguna mengunjungi *website*.

Fungsi *upload\_data()* pada baris 31 – 51 akan dijalankan ketika pengguna melakukan pengunggahan citra untuk diprediksi, jika berkas citra belum dipilih maka akan mengembalikan nilai *True* pada variabel *err* yang nantinya pada tampilan pengguna akan menunjukkan *alert* berupa teks peringatan citra belum dipilih untuk diunggah. Citra yang sudah diunggah akan disimpan kedalam *folder* penyimpanan dan tampilan utama *index.html* akan di-*render* serta akan mengembalikan nilai hasil prediksi berupa *class*, akurasi, serta lokasi penyimpanan citrayang telah diprediksi untuk ditampilkan kepada pengguna.

A screenshot of a cell phone

Description automatically generated

Gambar 4.9 Kode Program 2

## Implementasi Tampilan *Website*

## Pada Gambar 4.10 menampilkan halaman utama *website* yang berupa *navigation* *bar* yang berisi judul *website*, dua pilihan menu berupa menu *home* dan menu *about*. Menu *home* merupakan *section* untuk melakukan pengunggahan serta melakukan prediksi terhadap citra yang telah diunggah.

A screenshot of a cell phone

Description automatically generated

Gambar 4.10 Tampilan *Home Section*

Pada gambar 4.11 menampilkan *section home* yang disertai dengan tampilan peringatan terhadap pengguna agar memilih *file* citra sebelum melakukan pengunggahan dan prediksi.

A screenshot of a cell phone

Description automatically generated

Gambar 4.11 Tampilan *Alert*

Pada Gambar 4.12 menampilkan *popup window* berupa hasil prediksi yang terdapat citra yang diunggah, skor hasil prediksi serta klasifikasi hasil prediksi yang telah dilakukan oleh sistem.

A screenshot of a cell phone

Description automatically generated

Gambar 4.12 Tampilan Hasil Prediksi

Pada Gambar 4.13 menampilkan menu *about* yang menampilkan detail berupa penjelasan singkat tentang penyakit malaria, tujuan pembuatan *website*, dan model yang digunakan dalam melakukan prediksi pada citra sel darah merah.

A screenshot of a cell phone

Description automatically generated

Gambar 4.13 Tampilan *Section About*

# PENGUJIAN

## Pembahasan Pengujian

Pengujian yang dilakukan terhadap aplikasi web menggunakan metode *black box testing* dengan cara melakukan pengujian terhadap *input* dan *output* yang sesuai dengan yang diharapkan atau tidak.

Pada pengujian model hasil pelatihan dilakukan dengan melakukan prediksi pada *test set* dan melihat akurasi yang dihasilkan apakah berbanding jauh dengan akurasi pada hasil pelatihan model.

## Pengujian *Website*

Tabel 1 Pengujian Website

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Deskripsi Pengujian** | **Prosedur Pengujian** | ***Input*** | ***Output* yang Didapat** | **Hasil yang Didapat** | **Hasil** |
| 1 | Pengecekan Tampilan Utama | Menampilkan halaman *index* | Tidak Ada | Tampilan sesuai dengan yang diharapkan | Sesuai | Diterima |
| 2 | Prediksi tanpa memilih gambar untuk diunggah | Klik tombol Predict tanpa memilih gambar | Tidak Ada | Tampilan a*lert* sesuai dengan yang diharapkan | Sesuai | Diterima |
| 3 | Prediksi dengan mengunggah gambar | Memilih gambar dan klik tombol Predict | Gambar sel darah merah | Tampilan Popup Window hasil prediksi | Sesuai | Diterima |

## Pengujian Model

A screenshot of a cell phone

Description automatically generatedPada Gambar 5.1 menunjukkan akurasi pada hasil prediksi test set sebesar 97.22%. Jika dibandingkan dengan akurasi pelatihan model pada Gambar 4.6 yang sebesar 96.91% maka akurasi yang dihasilkan oleh hasil prediksi *test set* dapat diterima.

Gambar 5.1 Akurasi Prediksi Test Set

# SIMPULAN DAN SARAN

## Simpulan

Berdasarkan hasil pengerjaan dari kerja praktek ini, dapat disimpulkan bahwa pengklasifikasian terhadap citra sel darah merah yang memiliki parasit malaria sudah mampu dilakukan dengan cukup baik walaupun masih terjadi ketidakakuratan sebesar ±4% dan pengimplementasian aplikasi *website* untuk melakukan klasifikasi sudah mampu berfungsi dengan baik tetapi masih memiliki kekurangan pada tampilan *website* serta fitur yang terbatas pada *website*.

## Saran

Berdasarkan dari hasil yang telah dicapai dalam proses implementasi, hal yang dapat dilakukan untuk pengembangan lebih lanjut yaitu menambah jumlah epoch pada saat proses pelatihan model atau menggunakan arsitektur *Convolutional Neural Networks* lain seperti ResNet50 untuk mendapatkan akurasi yang lebih tinggi, meningkatkan tampilan *website* agar lebih mudah digunakan oleh pengguna, dan melakukan *hosting* pada aplikasi web supaya bisa diakses melalui internet.

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | WHO, “Malaria,” [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/malaria. [Diakses 10 Desember 2019]. |
| [2] | Biro Komunikasi dan Pelayanan Masyarakat, Kementerian Kesehatan RI, “Hari Malaria Sedunia, Pemerintah Perluas Wilayah Bebas Malaria,” 28 April 2018. [Online]. Available: http://www.depkes.go.id/pdf.php?id=18043000010. [Diakses 10 Desember 2019]. |
| [3] | Breslauer, David N et al. “Mobile phone based clinical microscopy for global health applications.” *PloS one* vol. 4,7 e6320. 22 Jul. 2009, doi:10.1371/journal.pone.0006320 |
| [4] | National Library of Medicine, “Malaria Datasets” [Online]. Available: https://lhncbc.nlm.nih.gov/publication/pub9932. [Diakses 10 Desember 2019]. |
| [5] | He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90. |
| [6] | Howard, Jeremy, and Sylvain Gugger. “Fastai: A Layered API for Deep Learning.” Information 11.2 (2020): 108. Crossref. Web. |
| [7] | Jeremy Jordan, “Setting the learning rate of your neural network. ,” [Online]. Available: https://www.jeremyjordan.me/nn-learning-rate/. [Diakses 25 Maret 2020]. |
| [8] | L. N. Smith, "Cyclical Learning Rates for Training Neural Networks," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, 2017, pp. 464-472. |
| [9] | Irsyad, Rahadian. (2018). Penggunaan Python Web Framework Flask Untuk Pemula. 10.31219/osf.io/t7u5r. |
| [10] | https://research.google.com/colaboratory/faq.html |
| [11] | https://github.com/BlackrockDigital/startbootstrap-freelancer |
| [12] | Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR. 2018. “Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images”. PeerJ 6:e4568 |
| [13] | https://pypi.org/project/split-folders/ |
| [14] | fastai, “vision.data | fastai,” [Online]. Available: https://docs.fast.ai/vision.data.html#ImageDataBunch. [Diakses 01 April 2020]. |
|  |  |
|  |  |
|  |  |
|  |  |