

IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK DENGAN FRAMEWORK TENSORFLOW PADA APLIKASI MOBILE PENDETEKSI PENYAKIT MELANOMA DENGAN MEMANFAATKAN WEBSERVICE FRAMEWORK FLASK

Yanes Rivki Yunius

Universitas Kristen Immanuel, Sleman, Yogyakarta, Indonesia

yanes@student.ukrimuniversity.ac.id

Abstrak

Melanoma Malignan atau disebut Melanoma adalah kanker kulit yang dapat berkembang pada melanosit, sel pigmen kulit yang berfungsi sebagai penghasil melanin. Menurut data National Cancer Institute Surveillance, Epidemiology, and End Result Program (SEER) jumlah kasus baru melanoma adalah 22,8 dari 100.000 dan jumlah kematian adalah 2,6 per 100.000 pria dan wanita pertahun [1]. Di bidang Informatika terdapat cara untuk mendeteksi penyakit, yaitu melalui deteksi citra digital atau foto. Salah satu teknologi yang digunakan untuk deteksi secara citra digital dapat menggunakan Convolution neural network(CNN).

Convolutional Neural Network adalah metode deep learning yang digunakan untuk citra visual. Preprocessing dalam algoritma ini tidak perlu dilakukan karena sudah masuk secara alamiah dalam arsitekturnya. Algoritma CNN dapat digunakan dengan library Tensorflow. Tensorflow adalah framework machine learning yang bekerja dalam skala besar dan dalam environment yang heterogenous. Webservice adalah sepotong atau sebagian informasi yang dapat diakses oleh siapa saja, kapan saja dan bersifat multiplatform[2].

Berdasarkan hasil pengujian yang dilakukan dengan menggunakan algoritma Convolutional Neural Network, hasil accuracy yang didapatkan adalah 0,93, lalu hasil prediksi dari 10 gambar melanoma mendapatkan akurasi rata – rata 0,877 dan prediksi dari 10 gambar non melanoma mendapatkan akurasi rata – rata 0,266.

Kata kunci: *convolutional neural network, melanoma, webservice*

Abstract

Melanoma Malignant is a skin cancer that evolves on melanocytes; a skin pigment that produces melanin. According to National Cancer Institute Surveillance, Epidemiology, and End Result Program (SEER) in 2015, the percentage of total new cases of melanoma reached 22.8% from 100.000, and the annual mortality rate was 2.6% per 100.000 both men and women [1]. Hence, it is crucial to detect this disease. There are several ways to detect the disease, one of which is by utilizing technology for image detection. One of the methods that can be used is deep learning.

Convolutional Neural Network (CNN) is a deep learning algorithm that is used for image detection. The pre-processing step in CNN does not need to be executed because this step is already a part of CNN's architecture. To develop this algorithm, we can use the help of Tensorflow framework. Tensorflow is a machine learning framework that works on a large

scale and in a heterogeneous environment. Web service is a part of the information or process which is accessible to everyone, whenever, and multiplatform[2].

As a result, CNN performed well by showing an accuracy of 93%. Besides, CNN shows average accuracy value in detecting melanoma and non-melanoma pictures with values of 87.7% and 26.6%, respectively.

Keyword: convolutional neural network, melanoma, web service

1. PENDAHULUAN

Kanker adalah penyakit yang menyerang semua kalangan, menurut data globocan menyebutkan di tahun 2018 terdapat 18,1 juta kasus baru dengan angka kematian sebesar 9,6 juta kematian, yang mana 1 dari 5 laki – laki dan 1 dari 6 perempuan di dunia mengalami kejadian kanker. Data tersebut juga menyatakan 1 dari 8 laki – laki dan 1 dari 11 perempuan, meninggal karena kanker [3]. Salah satu penyakit kanker kulit yang berbahaya adalah Melanoma, Melanoma Malignan atau disebut Melanoma adalah kanker kulit yang dapat berkembang pada melanosit, sel pigmen kulit yang berfungsi sebagai penghasil melanin. Gejala awal dari penyakit ini adalah kemunculan tahi lalat baru atau perubahan pada tahi lalat yang sudah ada. Hal ini dapat terjadi diseluruh tubuh, tetapi ada beberapa bagian tubuh yang sering mengalami kemunculannya yaitu wajah, tangan, punggung, kaki.

Menurut data *National Cancer Institute Surveillance, Epidemiology, and End Result Program* (SEER) jumlah kasus baru melanoma adalah 22,8 dari 100.000 dan jumlah kematian adalah 2,6 per 100.000 pria dan wanita pertahun, angka ini disesuaikan berdasarkan dari usia dan kasus kematian pada tahun 2011 – 2014 [1]. Tidak seperti di negara – negara yang memiliki jumlah kasus melanoma terbanyak, di Indonesia kasus melanoma menempati urutan ke 23 dengan jumlah kasus 1.392 pada tahun 2019 (*Global Cancer Observatory*. 2019). Meskipun melanoma dapat disembuhkan apabila dapat dideteksi sejak dini, masyarakat masih sulit mendeteksi gejala awal dari penyakit ini, karena melanoma biasanya muncul dalam berbagai bentuk, ukuran, dan warna.

Dari beberapa penelitian dibidang Informatika terdapat cara untuk mendeteksi penyakit, yaitu melalui deteksi citra digital atau foto dan hasil pengolahan wawancara. Salah satu teknologi yang digunakan untuk deteksi secara citra digital dapat menggunakan *Convolutional Neural Network (CNN)*.

Convolutional Neural Network adalah jenis metode deep learning yang digunakan untuk citra visual; citra visual akan diinput kedalam jaringan dan dengan menggunakan model jaringan 3 layer RGB (Red, Green, Blue). Algoritma ini akan mengenali citra visual tersebut. *Preprocessing* dalam algoritma ini tidak perlu dilakukan karena sudah masuk *secara* alamiah dalam arsitekturnya. Implementasi algoritma CNN dapat dilakukan dengan menggunakan *framework Tensorflow*. *Tensorflow* adalah *framework machine learning* yang bekerja dalam skala besar dan dalam environment yang *heterogenous*. *framework* ini juga digunakan untuk melakukan eksperimen *deep learning*, melatih model pada *dataset* yang berukuran besar, dan membuatnya layak diproduksi [4]. Namun, *tensorflow* tidak memiliki *user interface* yang dapat mempermudah user untuk melakukan prediksi dan hasil data prediksi tidak dapat diakses pada sistem operasi lain.

Sistem yang dapat digunakan untuk membantu proses prediksi agar dapat diakses oleh perangkat lain adalah *webservice*. *Webservice* adalah sepotong atau sebagian informasi atau

proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak terikat dengan system operasi atau bahasa pemrograman yang digunakan [2]. Sistem ini dapat dikembangkan dalam berbagai bahasa pemrograman seperti java, php, python, ruby, dan go.

Flask merupakan *Web Application Framework* yang ditulis menggunakan bahasa pemrograman python. Flask digunakan untuk mempermudah pengembangan *Web Application* [5]. *Framework* ini bergantung pada dua *external libraries* yaitu *jinja2 Template Engine* dan *WSGI Toolkit* [6]. *Framework* ini juga dapat diintegrasikan dengan *Tensorflow* sehingga dapat memudahkan user untuk memasukan data. *Output* dari webservice merupakan data json sehingga perlu aplikasi client untuk memvisualisasikan data json tersebut.

Sisi client yang akan dibuat untuk membantu visualisasi dari webservice ini dapat menggunakan aplikasi berbasis android. Android merupakan sistem operasi yang dikembangkan untuk perangkat mobile berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc. lalu kemudian dibeli oleh Google pada tahun 2005 [7].

Terdapat penelitian yang dilakukan oleh penelitian yang lain, yang pertama adalah penelitian yang dilakukan oleh Herwanto, dkk (2020) yang berjudul “Pengenalan Golongan Jenis Kendaraan Bermotor pada Ruas Jalan Tol Menggunakan CNN”. Tujuan penelitian ini adalah mengimplementasikan algoritma CNN pada sistem untuk mengklasifikasi jenis golongan kendaraan sehingga sistem dapat tahu berapa biaya yang dapat diambil dari saldo *e-money*. Hasil yang didapatkan dari pengujian ini menghasilkan akurasi sebesar 93,5% dan *f-score* sebesar 81,37%.

Penelitian kedua adalah Penelitian yang dilakukan oleh Wicaksono, dkk (2020) berjudul “Aplikasi Pendeteksi Penyakit Pada Daun Tanaman Apel Dengan Metode Convolutional Neural Network”. Tujuan penelitian ini adalah membuat aplikasi berbasis mobile menggunakan React Native dan Tensorflow untuk mengklasifikasikan penyakit yang menyerang tanaman apel menggunakan algoritma Convolutional Neural Network. Hasil yang didapatkan dari pengujian pada data training untuk epoch 50, 75, dan 100 menghasilkan rata – rata akurasi model sebesar 99,2% dan memiliki loss model sebesar 0,063. Dari pengujian pada data testing untuk epoch 50, 75, dan 100 menghasilkan rata – rata akurasi model sebesar 94,9% dan memiliki loss model sebesar 0,277.

Pada penelitian ini peneliti akan mengembangkan sistem pendeteksi penyakit melanoma menggunakan algoritma *Convolutional Neural Network* dengan *webservice application* sebagai pihak ketiga untuk menyajikan data prediksi CNN ke *client* dan dari sisi *client* menggunakan aplikasi android digunakan untuk dokter, yang diharapkan dapat membantu dokter memberi kesimpulan pada gambar yang berindikasi penyakit melanoma.

2. METODE PENELITIAN

2.1 Convolution Neural Network (CNN) dan Tensorflow

Convolutional Neural Network merupakan salah satu algoritma *deep learning* yang diklaim menjadi salah satu yang terbaik untuk model penyelesaian masalah pada object recognition. CNN dikembangkan dari *Multilayer Preceptron* (MLP) yang mana di desain untuk memproses dua dimensional data. Algoritma ini termasuk dalam *deep neural network* karena memiliki kedalaman network yang tinggi dan banyak diterapkan pada data gambar. Dalam kasus klasifikasi gambar pada penelitian tentang *virtual cortex* pada indra penglihatan kucing, MLP kurang cocok karena tidak menyimpan informasi spasial dari data gambar dan menganggap setiap piksel sebagai fitur independen yang menghasilkan hasil yang kurang baik [8].

Algoritma CNN memiliki 3 proses yang digunakan.

1. Convolutional Layers

Tahap ini bertujuan untuk menghapus garis lain pada gambar, sehingga yang didapatkan hanya garis vertikal dan horizontal.

2. Pooling Layers

Pooling yang digunakan adalah *max pooling*, *max pooling* membagi output menjadi beberapa bagian kotak dan mengambil nilai maksimum dari kotak yang telah dibagi. Hal ini bertujuan untuk mereduksi ukuran dari gambar agar dapat dengan mudah diganti dengan *convolutional layer* dengan *stride* yang sama dengan *pooling layer* yang sesuai. Bentuk dari *pooling* akan mereduksi *Feature Map* sampai 75% dari ukuran aslinya.

3. Fully Connected Layer

Fully Connected Layers adalah lapisan yang berisi activation neurons dari lapisan sebelumnya terhubung dengan *neuron* di lapisan berikutnya dan bertujuan untuk mengubah dimensi data sehingga data dapat diklasifikasikan secara linear.

Implementasi algoritma CNN dapat dilakukan dengan menggunakan *framework Tensorflow*. *Tensorflow* merupakan open-source software library, yang dikembangkan oleh tim Google Brain untuk mendukung komputasi pintar yang digunakan pada produk mereka. Komputasi pada *tensorflow* dapat dijalankan di berbagai perangkat, mulai perangkat seluler seperti smartphone dan tablet hingga ratusan sistem yang skala besar yang didistribusikan ke seluruh perangkat komputasi seperti kartu grafis (*GPU Card*) [8]. Berikut adalah implementasi algoritma CNN dengan menggunakan *framework Tensorflow*.

1. Proses Convolutional Layer dan Pooling Layer

Proses *convolutional layer* pertama memiliki *filters* bernilai 32, kernel size bernilai 5 x 5, dan input shape yang berisi image height merupakan tinggi gambar, image width merupakan lebar gambar, serta channels yang menandakan tipe yang digunakan, masing – masing memiliki nilai 60, 60, 3 yang telah didefinisikan ditahap load dataset. Nilai tiga pada channels berarti tahap convolution menggunakan data gambar RGB. Setelah melakukan tahap convolution layer pertama selanjutnya *MaxPooling2D* merupakan tahap *pooling layer* untuk mendapatkan filter pada *max pooling* untuk menentukan lebar dan tinggi yang akan digunakan pada tahap *convolution layer* dan *pooling layer* yang selanjutnya. Proses tersebut dapat dilihat pada Gambar 1.

```
model.add(Conv2D(layer_size, (5,5), input_shape=X_train.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
for l in range(conv_layer - 1):
    model.add(Conv2D(layer_size, (5, 5)))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
```

Gambar 1 proses *Convolutional layer* dan *Pooling layer*

2. Proses Fully Connected Layer

Proses pertama pada tahap ini adalah *flatten* untuk memisahkan data spasial dengan data yang tidak dapat digunakan pada tahap akhir jaringan, setelah tahap *flatten* tahap selanjutnya adalah *Dense*. *Dense* sendiri adalah proses *fully connected layer* yang bertujuan untuk mengubah dimensi data sehingga dapat diklasifikasikan secara linear. Tahap ini juga menggunakan *activation relu* yang berguna untuk meningkatkan sifat non-linear dari fungsi pengambilan keputusan dan semua jaringan tanpa

mempengaruhi bidang reseptif dari *Convolutional Layer*. Lalu fungsi *dropout* berfungsi mencegah *overfitting* pada tahap *fully connected layer*. Proses tersebut dapat dilihat pada Gambar 2.

```
model.add(Flatten())
for l in range(dense_layer):
    model.add(Dense(dense_layer))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
```

Gambar 2 proses *Fully Connected Layer*

3. Proses *training*

Proses terakhir yaitu proses training data selama 40 epoch, pada proses ini juga berlangsung proses validation data. Proses tersebut dapat dilihat pada Gambar 3.

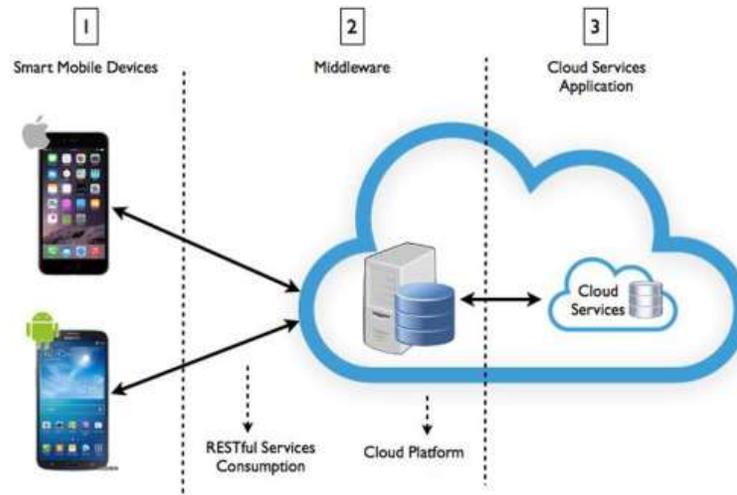
```
model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])

model.fit(X_train, y_train, epochs=40, validation_data=(X_test,
y_test), callbacks=[tensorboard])
```

Gambar 3 proses *Training*

2.2 Webservice

Webservice adalah antarmuka yang mendeskripsikan sekumpulan operasi yang dapat diakses dalam sebuah jaringan melalui pesan XML yang telah distandarkan [2]. Terdapat pengertian lain dari *Webservice* yaitu sepotong atau sebagian informasi atau proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak terikat dengan system operasi atau bahasa pemrograman yang digunakan [2]. Dapat disimpulkan *webservice* merupakan suatu sistem yang digunakan untuk menghubungkan sistem satu dengan sistem yang lainnya menggunakan media jaringan internet. Sekalipun teknologi ini hampir sama dengan *Application Programming Interface (API)*, web service lebih unggul karena dapat dipanggil dari jarak jauh melalui internet. Pemanggilan *webservice* dapat dipanggil dengan menggunakan diberbagai bahasa pemrograman dan platform, sementara API hanya dapat digunakan dalam platform tertentu. Penggambaran dari webservice dapat dilihat pada Gambar 4.



Gambar 4 Konsep webservice

Webservice memiliki 2 jenis, yaitu:

1. SOAP

SOAP (Simple Object Access Protocol) merupakan spesifikasi protokol untuk melakukan pertukaran informasi terstruktur pada implementasi *webservice* di jaringan komputer. Sistem ini menggunakan *Extensible Markup Language* (XML) sebagai format pesannya, dan biasanya bergantung pada protocol layer aplikasi lainnya, terutama *Hypertext Transfer Protocol* (HTTP) dan *Simple Mail Transfer Protocol* (SMTP), untuk transmisi dan negosiasi pesan.

2. REST

REST (Representation State Transfer) merupakan arsitektur software untuk sistem terdistribusi. REST telah berkembang sebagai model desain webservice yang dominan untuk saat ini. Istilah REST dikenalkan dan didefinisikan pada tahun 2000 oleh Roy Fielding dalam disertasi doktoralnya.

2.3 Flask

Flask merupakan sebuah web framework yang dikembangkan menggunakan bahasa python yang tergolong sebagai *microframework*. Flask memiliki fungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. *Framework* ini termasuk pada jenis *microframework* karena tidak ada memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Fungsi yang secara default yang belum terpasang pada flask seperti validasi form, database, dan sebagainya. Hal ini dikarenakan fungsi dan komponen – komponen tersebut sudah disediakan oleh pihak ketiga dan flask dapat menggunakan ekstensi yang membuat fitur dan komponen – komponen tersebut seakan diimplementasikan oleh flask sendiri. (Irsyad. 2018)

Fitur – fitur yang terdapat pada flask seperti built-in development server, debugger cepat, integrated support untuk unit test, kompatibel dengan mesin aplikasi Google, *RESTful request dispatching*, Jinja 2 templating, mendukung *secure cookies*, dan berbasis unicode. Flask juga memiliki kelebihan sebagai berikut:

1. Ringan untuk dijalankan karena mempunyai *core* yang sederhana dan desain modular.
2. Dapat menangani fungsi HTTP request dengan mudah.
3. API yang baik dan koheren.
4. Dokumentasi yang banyak dan terstruktur dengan baik, penuh dengan contoh yang dapat digunakan langsung.
5. Mudah untuk dipasang dan di-*deploy* untuk produksi.

6. Mudah untuk diperiksa secara menyeluruh.

2.4 Android

Android merupakan sebuah sistem operasi yang berbasis linux untuk telepon seluler seperti *smartphone*. Android bersifat *opensource* yang berarti terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri agar dapat digunakan oleh berbagai macam jenis *smartphone* [10].

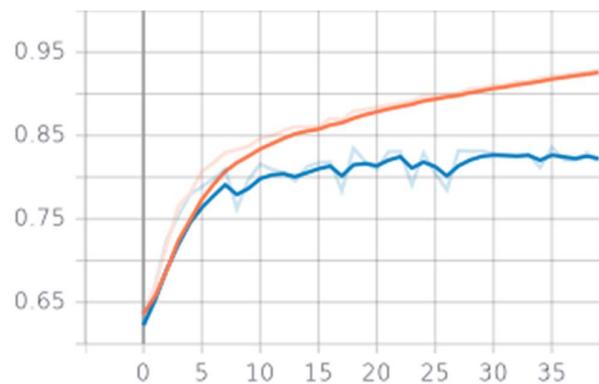
3. HASIL DAN PEMBAHASAN

Tahap pengujian pada *training data* dilakukan dengan memberi nilai pada variable *dense_layer*, *layer_sizes*, dan *conv_layers* yang berguna untuk mencari nilai *accuracy* dan *loss* yang terbaik. Nilai yang di *set* dapat dilihat pada Gambar 5.

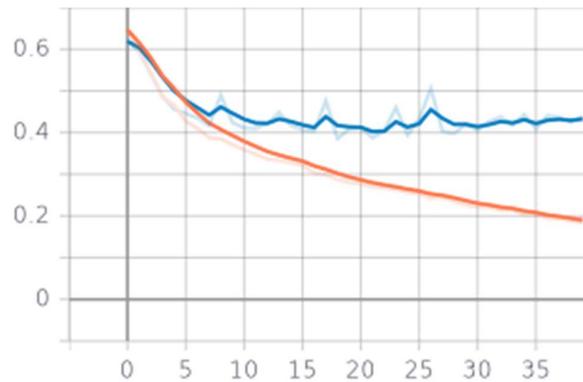
```
dense_layers = [0, 1, 2]
layer_sizes = [32, 64, 128]
conv_layers = [1, 2, 3]
```

Gambar 5 Nilai yang digunakan pada tahap *training data*

Dari hasil training data yang dilakukan didapatkan nilai *accuracy* dan *loss* sebesar 0,93 dan 0,16 dan validation data sebesar 0,82 dan 0,43 dengan kombinasi nilai *conv_layer* = 1, *layer_sizes* = 128, dan *dense_layers* = 0. Hasil data training dapat dilihat pada visualisasi grafik dengan sumbu y menunjukkan *accuracy* dan *loss* serta sumbu x menunjukkan epoch pada Gambar 6 dan Gambar 7.



Gambar 6 Grafik *accuracy* dari hasil *training data*



Gambar 7 Grafik *loss* dari hasil *training data*

Setelah mendapatkan *accuracy* dan *loss* dari training data, data tersebut disimpan kedalam file *result.model* sebagai acuan proses prediksi. Pengujian proses prediksi akan dilakukan dengan menggunakan 10 data gambar melanoma dan 10 data gambar non melanoma. Hasil pengujian proses prediksi dapat dilihat pada Tabel 1 dan Tabel 2.

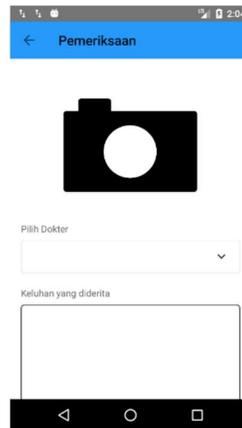
Tabel 1 Hasil pengujian proses prediksi dengan data melanoma

No	Jenis Penyakit	Hasil
1	Melanoma	0,87
2	Melanoma	0,99
3	Melanoma	0,10
4	Melanoma	0,99
5	Melanoma	0,99
6	Melanoma	0,99
7	Melanoma	0,99
8	Melanoma	0,87
9	Melanoma	0,99
10	Melanoma	0,99
Rata – rata		0,877

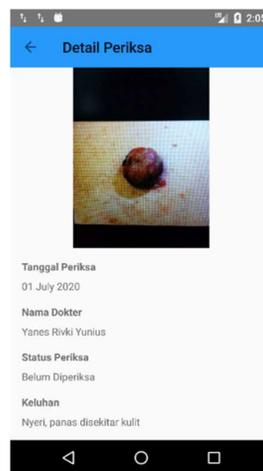
Tabel 2 Hasil pengujian proses prediksi dengan data non melanoma

No	Jenis Kelamin	Hasil
1	Non Melanoma	0,15
2	Non Melanoma	0,66
3	Non Melanoma	0,25
4	Non Melanoma	0,04
5	Non Melanoma	0,02
6	Non Melanoma	0,008
7	Non Melanoma	0,0002
8	Non Melanoma	0,0003
9	Non Melanoma	0,99
10	Non Melanoma	0,55
Rata – rata		0,266

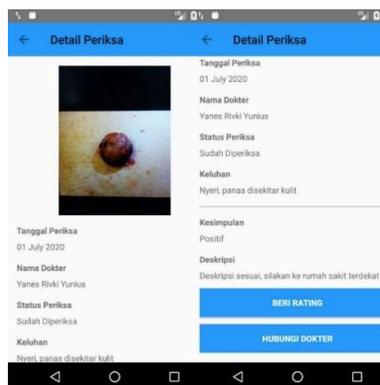
Berikut merupakan hasil dari sistem yang dikembangkan



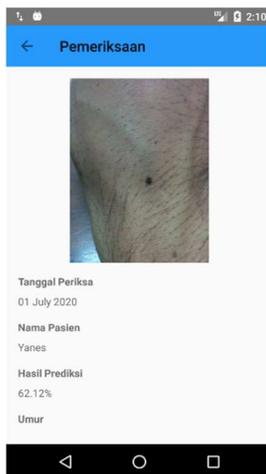
Gambar 8. Tampilan Fungsi Pemeriksaan



Gambar 9. Tampilan detail belum diperiksa



Gambar 10. Tampilan detail yang telah diperiksa



Gambar 11. Tampilan pemeriksaan dokter

4. KESIMPULAN

Berdasarkan dari hasil studi literatur, analisis, perancangan, implementasi dan pengujian pada sistem ini, maka kesimpulan yang didapat adalah sebagai berikut.

1. Pengembangan *webservice* menggunakan *flask* dimulai dengan melakukan analisis, perancangan, dan implementasi seperti membuat model sebagai perantara ke database, controller sebagai proses logic pada sistem, dan view sebagai menampilkan data berbentuk json agar dapat digunakan pada client (aplikasi android). Lalu membuat proses load dataset yang berguna untuk mengubah dataset gambar menjadi array dan proses training data untuk melakukan training pada dataset, Lalu mengintegrasikannya pada *webservice*.
2. Pengembangan aplikasi android untuk pasien dan dokter dimulai dengan analisis kebutuhan, perancangan sistem seperti membuat desain aplikasi dan implementasi seperti membuat tampilan, mengolah data dari *webservice* atau mengirim data ke *webservice*, serta mengimplementasikan arsitektur MVP dengan menggunakan bahasa pemrograman kotlin.

Pengujian yang dilakukan dengan menggunakan algoritma *Convolutional Neural Network* mendapatkan accuracy yang baik yaitu 0,93, lalu hasil prediksi dari 10 gambar melanoma mendapatkan akurasi 0,877 dan prediksi dari 10 gambar non melanoma mendapatkan akurasi 0,266.

5. SARAN

Adapun saran – saran yang dapat diberikan untuk pengembangan dan perbaikan sistem ini selanjutnya adalah sebagai berikut.

1. Melakukan pemotongan gambar sehingga mengurangi *noise* pada tahap prediksi, dikarenakan pada sistem sekarang gambar yang didapatkan terdapat background yang dapat mengurangi hasil prediksi.
2. Mendapatkan kualitas gambar yang lebih baik agar proses prediksi lebih optimal, dikarenakan pada sistem sekarang gambar yang dikirimkan oleh client langsung diolah tidak dilakukan proses pemfilteran gambar sehingga gambar yang diterima lebih baik.
3. Mengembangkan fitur chat pada sistem agar mempermudah proses konsultasi, pada sistem sekarang proses konsultasi hanya dapat dilakukan dengan telepon.

DAFTAR PUSTAKA

- [1]Noone AM, Howlader N, Krapcho M, Miller D, Brest A, Yu M, Ruhl J, Tatalovich Z, Mariotto A, Lewis DR, Chen HS, Feuer EJ, Cronin KA (eds). *SEER Cancer Statistics Review*. 2015. Sumber: https://seer.cancer.gov/csr/1975_2015/, diakses pada 6 September 2019.
- [2]Kreger, Heather. 2001. *Web Services Conceptual Architecture (WSCA 1.0)*. Sumber:https://www.researchgate.net/publication/235720479_Web_Services_Conceptual_Architecture_WSCA_10, diakses pada 8 juli 2020.
- [3]Kementrian Kesehatan. 2019. Hari Kanker Sedunia 2019. Departemen Kesehatan. Sumber: <http://www.depkes.go.id/article/view/19020100003/hari-kanker-sedunia-2019.html>, diakses pada 6 September 2019.
- [4]Pangestu, M. Afrizal & Bunyamin Hendra. 2018. *Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model*. Bandung: Jurnal Teknik Informatika dan Sistem Informasi 4. pp. 337 – 344.
- [5] A. F. Aslam, N. H. Mohammed dan P. Lokhande. 2015. *Efficient Way of Web Development Using Python and Flask*. International Journal of Advanced Research in Computer Science, vol. 6, 2015.
- [6]Pallets Team. 2010. *Flask Documentation, Pallets Team*. Sumber: <http://flask.pocoo.org/docs/1.0/>, Diakses 30 April 2020.
- [7]Maiyana, Efmi. 2018. *Pemanfaatan Android dalam Perancangan Aplikasi Kumpulan Doa*. Bukittinggi: Akademi Manajemen Informatika & Komputer.
- [8]Refianti, R., Mutiara, A. B., Priyandini, R. 2019. *Classification of Melanoma Skin Cancer using Convolutional Neural Network*. Depok: International Journal of Advanced Computer Science and Applications, Vol. 10, No. 3, 2019.
- [9]Irsyan, Rahardian. 2018. *Penggunaan Python Web Framework Flask Untuk Pemula*. Bandung: Institut Teknologi Bandung.
- [10]Sede, Deybi W.E. 2015. *Rancang Bangun Aplikasi Pemesanan Tiket Online Kapal Laut Berbasis Android*. Manado: E-Journal Teknik Informatika. Vol. 6, No. 1, 2015.