

Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow

by John Doe

Submission date: 09-Jun-2021 11:10AM (UTC+0700)

Submission ID: 1603233913

File name: Wajah_Menggunakan_Algoritma_Deep_Learning_dengan_TensorFlow.docx (1.33M)

Word count: 3650

Character count: 24512

Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow

Abstract—Seiring dengan meningkatnya klinik dermatologi, persaingan dalam metode dan teknologi untuk menyembuhkan kulit semakin kuat. Namun salah satu perawatan masalah kulit yang paling sering muncul saat ini adalah perawatan wajah berjerawat. Penelitian ini merupakan penelitian mengenai pengembangan aplikasi visi komputer berbasis web untuk melakukan pendeteksian jerawat pada foto wajah menggunakan algoritma *deep learning* dengan TensorFlow. Penelitian ini bertujuan untuk menghasilkan aplikasi yang dapat digunakan oleh klinik dermatologi sebagai asesmen perawatan kulit berjerawat. Segmentasi kulit wajah dilakukan dengan algoritma pendeteksian wajah dan pendeteksian landmark wajah. Proses pelatihan model pendeteksian jerawat menggunakan metode *transfer learning* dari model SSD ResNet50 V1 FPN 640x640. Hasil dari model pelatihan memiliki nilai total loss 0.147. Aplikasi sudah dapat mendeteksi jerawat-jerawat yang ada pada foto wajah, namun ada beberapa jerawat yang tidak terdeteksi dikarenakan jumlah dataset yang masih terbatas.

Keywords—visi komputer, jerawat wajah, deep learning, TensorFlow

I. PENDAHULUAN

Sekarang, memiliki kulit yang bagus menjadi tren dan berpengaruh di pergaulan masyarakat. Seiring dengan meningkatnya klinik dermatologi, persaingan dalam metode dan teknologi untuk menyembuhkan kulit semakin kuat. Namun salah satu perawatan masalah kulit yang paling sering muncul saat ini adalah perawatan wajah berjerawat. Sebagian besar remaja dan bahkan beberapa orang dewasa mengalami masalah ini.

Jerawat (*Acne vulgaris*) adalah masalah pada kulit atau kondisi kulit yang umum terjadi. Penyebab jerawat biasanya terjadi karena penyumbatan kelenjar minyak di kulit atau infeksi bakteri. Jerawat bisa ditandai dengan kulit bersisik merah (*seborrhea*), komedo, papula, nodul, bintil, dan jaringan parut [1, 2]. Ada dua jenis jerawat, yang pertama adalah lesi non-inflamasi, dan yang lainnya adalah lesi inflamasi. Pada kondisi yang parah, jerawat besar membengkak di bawah kulit dan meradang sehingga mengakibatkan benjolan menyakitkan yang menyebabkan jaringan parut. Meski tidak berbahaya, penderita sering mengalami gangguan emosi dan sosial karena ketidaknyamanan dan rasa malu yang ditimbulkan oleh jerawat. Jerawat muncul secara umum selama masa remaja, mempengaruhi sekitar 80-90% [3].

Secara tradisional, penilaian terhadap kulit berjerawat dilakukan oleh dokter kulit di lingkungan medis. Cara perawatan kulit berjerawat diresepkan oleh dokter kulit atau produk perawatan kulit yang dijual bebas [3, 4]. Perawatan awal untuk kulit wajah dan diagnosis jerawat adalah salah satu prosedur yang diperlukan. Metode yang umumnya dilakukan yaitu meminta dokter kulit secara manual menandai tempat jerawat pada lembar penanda sesuai dengan lokasi jerawat yang telah diamati. Hasil penandaan terkadang masih kurang akurat dan dapat menghabiskan waktu dan tenaga bagi dokter kulit untuk menentukan tingkat keparahan.

Penggunaan alat pada masalah kulit sudah digunakan oleh beberapa klinik dermatologi. Kebanyakan alat analisis kulit masih membutuhkan usaha manual untuk menentukan *region of interest* (ROI) sebelum dilakukan pengamanan dengan perangkat lunak, sebagian besar klinik menggunakan VISIA [5]. Hal ini memakan waktu dan tidak dapat diulang untuk area yang berbeda dan alat ini sangat mahal.

Berdasarkan permasalahan di atas, penelitian ini dilakukan untuk membuat suatu aplikasi untuk mendeteksi jerawat di wajah pasien secara otomatis tanpa cara manual dari foto wajah untuk menghitung atau mendiagnosis apakah area tersebut berjerawat atau tidak, sebagai asesmen perawatan kulit berjerawat. Selanjutnya hal ini akan dibahas pada bagian berikut. Bagian II, menjelaskan landasan teori yang menjadi dasar dari penelitian. Bagian III, menjelaskan tentang metode terkait yang diusulkan pada penelitian. Sedangkan bagian IV menjelaskan tentang hasil dan pembahasan dari penelitian. Selanjutnya yang terakhir bagian kesimpulan akan dibahas pada bagian V.

II. LANDASAN TEORI

A. Penelitian Terkait

Topik dari penelitian ini merupakan bagian dari bidang visi komputer tentang pendeteksian objek. Penelitian sebelumnya untuk melakukan pendeteksian jerawat menggunakan rentang warna HSV atau RGB untuk mendapatkan area kulit wajah dan objek jerawat. Sistem yang dikembangkan mencoba mendeteksi jerawat berdasarkan titik-titik yang terindikasi jerawat pada wajah, kemudian sistem akan menyeleksi berdasarkan warna, bentuk, dan luas area piksel. Metode lain juga diusulkan, proses dimulai dengan masukan gambar asli, *resize* gambar, segmentasi kulit, *sharpening*, seleksi ciri (luas, bentuk, warna) dan diakhiri proses *marking* objek jerawat [6, 7].

Penelitian yang dilakukan untuk pendeteksian jerawat menggunakan algoritma *machine learning* [8]. Penelitian ini menggunakan dua metode klasifikasi pembelajaran mesin. Akurasi rata-rata klasifikasi untuk membedakan bekas luka jerawat non-inflamasi dan inflamasi untuk metode FCM dan SVM linier adalah masing-masing 80% dan 66,6%. Pada klasifikasi kulit normal dari orang yang berjerawat dengan metode FCM akurasi performanya 100%.

Penggunaan rentang warna untuk pendeteksian objek tidak bisa berjalan dengan baik apabila warna kulit pasien terlalu bervariasi dan kondisi pencahayaan pengambilan foto yang berbeda-beda. Kami mengusulkan penggunaan algoritma pendeteksian wajah [9] dan landmark wajah [10, 11] untuk mendapatkan area wajah pasien. Sedangkan untuk mendeteksi jerawat menggunakan *deep learning* berbasis CNN pada TensorFlow.

20

B. Jerawat (*Acne Vulgaris*)

Jerawat adalah kondisi kulit yang terjadi ketika kelenjar sebaceous (penghasil minyak) di kulit dan di sepanjang batang rambut tersumbat dan meradang, serta terinfeksi bakteri. Jika jerawat membentuk pembengkakan yang kuat di bawah kulit, meradang dan bisa berkembang menjadi benjolan yang menyakitkan dan menyebabkan jaringan parut. Gejala yang dapat dilihat pada pasien berjerawat diantaranya sebagai berikut,

- *Whiteheads*
- Kulit berminyak meningkat
- Komedo - bintik hitam dengan pori-pori terbuka di bagian tengah
- Benjolan merah kecil
- Pembengkakan merah atau benjolan yang tampak berisi nanah (jerawat / pustula) - di wajah, dada, bahu, leher, dan / atau punggung atas
- Benjolan besar, meradang, merah, berisi cairan (nodul) di bawah kulit
- Benjolan besar, meradang, merah, lunak (kista) di bawah kulit yang bisa menjadi sebesar 2,5 cm.

Tempat munculnya jerawat biasanya terdapat di wajah, leher, dada, bahu, atau punggung. Tetapi jerawat paling sering terjadi pada wajah karena berbagai faktor etiologi dari jerawat. Meskipun jerawat bisa terjadi di mana saja di wajah, area jerawat yang dominan adalah dahi, pipi, dan dagu [1].

Secara tradisional, penilaian terhadap kulit berjerawat dilakukan oleh dokter kulit di lingkungan medis. Dokter kulit membuat diagnosis awal untuk membuat rencana perawatan yang efektif. Dokter akan mencatat jenis jerawat dan lokasi jerawat yang terdapat pada kulit. [4].

9

Jenis perawatan ⁹rgantung pada seberapa parah dan persistennya jerawat. Jerawat ringan dapat diobati dengan obat bebas (OTC), seperti gel, sabun, pembalut, krim, dan losion,

yang dioleskan ke kulit. Pengobatan untuk jerawat sedang hingga parah dilakukan oleh spesialis kulit atau dokter kulit [3].

C. Pendeteksian Landmark Wajah

Deteksi landmark wajah merupakan metode untuk mengekstrak fitur menonjol pada wajah. Landmark wajah mewakili titik fitur menonjol di wajah, seperti mata, hidung, dan mulut. Landmark wajah dapat digunakan untuk beragam tugas seperti pengenalan wajah, deteksi tatapan, pelacakan orang, pengenalan emosi, dan rias virtual. Sejauh ini, banyak penelitian telah dilakukan dengan tujuan mencapai ekstraksi landmark yang efisien dari gambar wajah. Proses pendeteksian dan pelacakan yang menggunakan terlalu banyak titik fitur landmark biasanya membutuhkan waktu pemrosesan yang berlebihan. Sebaliknya, apabila mengandalkan terlalu sedikit titik fitur tidak dapat menghasilkan pendeteksian yang akurat. Metode yang diusulkan telah mampu mendapatkan 68 titik fitur menonjol di wajah secara *real time*, menggunakan pustaka Dlib, dapat dilihat pada Gambar 1. Metode ini sudah dapat menyesuaikan posisi dari wajah sehingga dapat dijadikan acuan pengambilan area wajah [10, 11].



Gambar 1. Titik landmark wajah pada pustaka Dlib

D. Tensorflow Object Detection

TensorFlow adalah *platform open source end-to-end* untuk *machine learning* yang dibuat oleh tim Google Brain. TensorFlow memiliki perangkat, pustaka, dan sumber daya komunitas yang komprehensif dan fleksibel yang memungkinkan para peneliti dalam pengembangan *machine learning* dan tim pengembang dapat dengan mudah membangun dan menerapkan aplikasi yang didukung *machine learning*. Arsitekturnya yang fleksibel memungkinkan penerapan komputasi yang mudah di berbagai platform (CPU, GPU, TPU), dan dari perangkat seperti desktop, *cloud*, maupun perangkat *mobile* [12, 13].

TensorFlow menyediakan dukungan untuk mempermudah pengembangan aplikasi untuk pendeteksian objek yaitu melalui TensorFlow Object Detection API. TensorFlow Object Detection API merupakan *framework* yang dapat digunakan untuk membuat dan mengembangkan model *deep learning*

pendeteksian objek, memberikan kemudahan dalam membangun, melatih dan penerapan. Google telah menggunakan *framework* ini di berbagai produknya yang menggunakan visi komputer antara lain pencarian gambar, Google Assistant, deteksi wajah dan plat nomor kendaraan pada Google Streetview, Waymo atau *self driving car*, dan lain-lain.

TensorFlow Object Detection API juga menyediakan banyak model yang telah dilatih dengan dataset Common Object in Context (COCO) 2017 yang terdiri 300.000 gambar berisi 90 kategori objek. Model-model tersebut memiliki kecepatan dan tingkat akurasi masing-masing, sesuai dengan arsitektur modelnya [14, 15].

E. Aplikasi Web

Aplikasi web menjadi populer semenjak perkembangan teknologi internet karena bisa diakses dari berbagai jenis perangkat komputer melalui browser web. Aplikasi web disimpan di suatu remote server sehingga memberikan kemudahan dalam pembaharuan dan pemeliharaan aplikasi. Proses pendistribusian yang tidak memerlukan instalasi perangkat lunak pada kemungkinan ribuan komputer klien merupakan alasan kunci popularitasnya. Aplikasi web yang umum misalnya webmail, toko ritel, sistem kasir (POS), wiki, portal berita, weblog maupun sosial media [16].

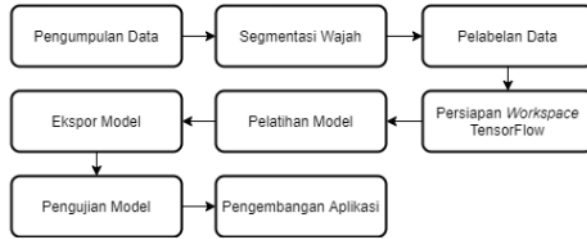
Banyak keuntungan yang diberikan oleh aplikasi berbasis web dibandingkan aplikasi berbasis desktop atau *mobile* sehingga sampai sekarang aplikasi berbasis web tetap diadopsi oleh berbagai perusahaan sebagai bagian dari strategi teknologi untuk menyediakan layanan, karena beberapa alasan:

- Akses informasi mudah,
- Setup server lebih mudah
- Informasi mudah didistribusikan
- Bebas platform, aplikasi dapat diakses dari berbagai jenis perangkat komputer atau sistem operasi

Terdapat beberapa jenis arsitektur aplikasi web secara umum digunakan dalam pengembangan, salah satunya adalah *single-page application* (SPA). Ini merupakan arsitektur aplikasi web yang tidak membutuhkan pemuatan halaman setiap saat terjadi interaksi pada aplikasi di dalam browser. Aplikasi terbagi menjadi dua bagian yaitu *frontend* dan *backend*. Dalam SPA, semua kode HTML, JavaScript, dan CSS yang diperlukan diambil oleh browser dengan pemuatan halaman tunggal. Permintaan ke server biasanya menghasilkan data mentah seperti XML, JSON atau HTML baru. Setelah data diterima, data diterjemahkan menggunakan JavaScript untuk memperbaharui area dari DOM [17].

III. METODOLOGI

Tahapan penelitian ini dapat dilihat pada diagram di Gambar 2, sebagai berikut:



Gambar 2. Tahapan Penelitian

A. Pengumpulan Data

Pengumpulan gambar pada penelitian ini dilakukan menggunakan kamera *smartphone*. Gambar berupa foto wajah orang-orang yang memiliki masalah jerawat pada wajah. Sudut pengambilan foto pada area wajah dari sisi depan, serong kanan dan serong kiri. Kondisi yang perlu dipastikan dalam pengambilan foto diantaranya,

- wajah tanpa *makeup*
- pencahayaan cukup
- latar belakang kontras dengan warna kulit
- tidak *backlight*
- fokus ke area wajah
- tidak menggunakan filter
- kamera min. 8MP.

Peneliti juga menggunakan gambar yang dikumpulkan dari Google dengan kata kunci 'wajah berjerawat' untuk memperbanyak data.

B. Segmentasi Wajah

Deteksi wajah dilakukan untuk mengambil ROI dari gambar berupa area kulit wajah. Algoritma pendeteksian wajah dan landmark wajah digunakan pada tahap ini untuk mendapatkan area kulit wajah secara otomatis. Hal ini dilakukan dengan menggunakan pustaka OpenCV dan Dlib.

Hasil dari pendeteksian wajah menggunakan pustaka OpenCV yang menerapkan Haar Cascades Classifier berupa array koordinat x dan y titik awal lokasi piksel wajah beserta lebar dan tinggi areanya. Hasil tersebut kemudian digunakan untuk mengambil gambar area wajah dengan melakukan *cropping* sehingga didapatkan gambar baru yang hanya berisi wajah. Gambar tersebut masih menyisakan latar belakang dari gambar lama sehingga perlu dilakukan segmentasi untuk area kulit wajah.

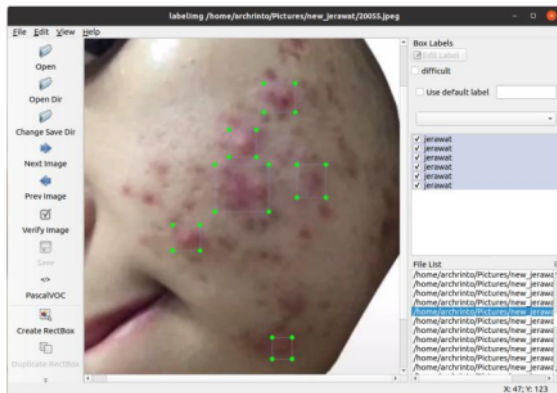
Segmentasi kulit wajah dilakukan dengan melakukan pendeteksian landmark wajah. Hasil dari pendeteksian menggunakan pustaka Dlib berupa 68 titik landmark wajah. Titik-titik tersebut kemudian diambil nomor 1 sampai 17 yang digunakan sebagai acuan untuk mendapatkan area kulit wajah.

Dari titik-titik tersebut kemudian dikonversi menjadi garis batas dengan menggunakan metode Convex Hull. Garis batas yang didapat kemudian dipakai untuk membuat *mask* gambar, berupa array bertipe data boolean berukuran $n \times m \times 1$. Area di dalam garis batas bernilai *true* yang mewakili area kulit wajah sedangkan di luar garis batas bernilai *false* yang mewakili area yang dianggap latar belakang. Proses *masking* kemudian dilakukan, menghasilkan area kulit wajah yang tersegmentasi, seperti Gambar 4.

C. Pelabelan Data

Pelabelan merupakan kegiatan memberikan label pada suatu dataset. Metode yang sering digunakan untuk melakukan pelabelan adalah secara manual dengan menggunakan bantuan tenaga ahli. Tahap ini dilakukan untuk memberikan label pada objek-objek jerawat yang terdapat pada gambar wajah. Penandaan pada gambar berupa koordinat area (x , y , *lebar* dan *tinggi*) yang mewakili lokasi kumpulan piksel dari objek. Setiap penandaan objek diberi label 'jerawat' sebagai nama dari kelas objek. Pelabelan dilakukan untuk masing-masing file gambar kulit wajah yang sudah tersegmentasi.

Alat yang digunakan untuk pelabelan data yaitu aplikasi yang bernama LabelImg, alat anotasi gambar grafis. LabelImg ditulis dengan bahasa Python dan menggunakan Qt sebagai antarmuka grafisnya. LabelImg mendukung pelabelan yang disimpan dengan format file PASCAL VOC, YOLO atau CreateML.



Gambar 3. Pelabelan objek dengan LabelImg

D. Persiapan workspace TensorFlow

TensorFlow memiliki spesifikasi ruang kerja yang perlu dipersiapkan untuk membuat model pendeteksian objek. Persiapan ini meliputi persiapan dataset, pembuatan LabelMap, pembuatan TensorFlow Record, dan konfigurasi *pipeline* pelatihan model. Data yang sudah diberi label kemudian dilakukan pemisahan data untuk mempersiapkan data pelatihan dan data pengujian, masing-masing 80% dan 20% dari dataset yang dimiliki. Pembuatan LabelMap dilakukan untuk memetakan setiap label atau kelas objek yang ada pada dataset. LabelMap ini digunakan dalam proses pelatihan dan pendeteksian. Dataset pelatihan dan pengujian kemudian

diubah ke dalam format yang disebut TFRecord. TensorFlow Record atau TFRecord merupakan format sederhana untuk menyimpan urutan catatan biner. Konfigurasi *pipeline* pelatihan model dilakukan untuk menyesuaikan sumber daya yang dimiliki [18].

E. Pelatihan Model

Pelatihan model adalah proses untuk mendapatkan model pendeteksian jerawat pada wajah. Pelatihan model menggunakan metode *transfer learning*. *Transfer learning* merupakan istilah pada *machine learning* yang mengacu pada pemanfaatan pengetahuan dari suatu bidang ke bidang yang lain, merupakan salah satu solusi apabila memiliki data yang sedikit. Model yang sebelumnya dilatih digunakan untuk mengekstrak fitur-fitur dari lapisan bawah jaringan saraf tiruan. Kemudian dari fitur-fitur tersebut dimanfaatkan untuk melatih model baru dengan tujuan pendeteksian jerawat.

Proses pelatihan perlu dipantau untuk mengetahui perkembangan dari model yang dilatih, dapat dilihat dari nilai *total loss*. TensorFlow menyediakan fitur untuk memantau dan memvisualkan laporan perkembangan pelatihan model melalui TensorBoard. Model pendeteksian yang bagus paling tidak mencapai nilai *total loss* mendekati 2 atau secara idealnya 1 atau kurang dari 1. Nilai *total loss* yang lebih rendah tentunya lebih baik, namun nilai *total loss* yang sangat rendah perlu dihindari karena model tersebut mungkin *overfitting* pada dataset, yang berarti bahwa model tersebut akan berkinerja buruk ketika diterapkan pada gambar di luar dataset. Lamanya proses pelatihan bergantung pada performa dari PC atau server yang digunakan [18].

F. Ekspor Model

Ekspor model dilakukan untuk mengambil model yang sudah selesai dari proses pelatihan. Grafik inferensi dari model diekstrak yang nantinya digunakan untuk melakukan deteksi objek.

G. Pengujian Model

Model yang telah terekstrak kemudian dilakukan pengujian dengan program Python. Program ini mengambil citra pengujian dan mencoba mendeteksi jerawat menggunakan model yang telah disimpan. Objek yang terdeteksi sebagai jerawat pada gambar diberi tanda kotak sehingga dapat dilihat pada gambar hasil deteksi. Kebenaran dari pendeteksian dilakukan dengan mencocokkan gambar hasil deteksi dengan pelabelan yang telah dilakukan sebelumnya, apakah model sudah dapat mendeteksi objek jerawat dengan tepat atau belum.

H. Pengembangan Aplikasi

Pengembangan aplikasi merupakan bagian dari upaya penerapan aplikasi untuk pendeteksian jerawat agar dapat digunakan oleh pengguna. Langkah-langkah yang diterapkan untuk mengembangkan aplikasi dalam penelitian ini adalah sebagai berikut:

1) *Identifikasi kebutuhan aplikasi*: Menganalisis kebutuhan aplikasi dari sisi kebutuhan program, menjelaskan

proses-proses yang dilakukan oleh program aplikasi yang dibutuhkan oleh pengguna.

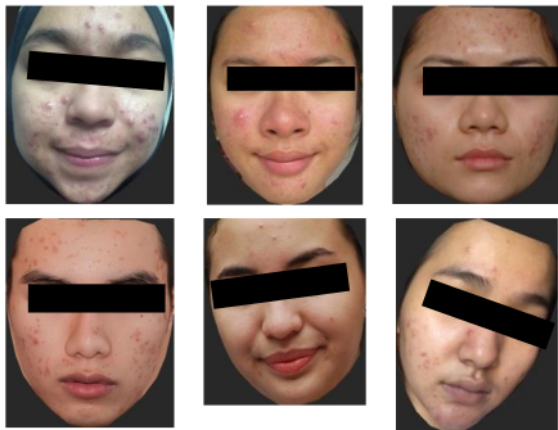
5
2) *Perancangan aplikasi*: Tahapan yang menjelaskan bagaimana aplikasi akan dibuat. Rancangan sistem berupa diagram arsitektur aplikasi, diagram alir (flowchart) aplikasi dan rancangan antarmuka (interface).

3) *Implementasi*: Tahap pembuatan aplikasi, sesuai rancangan yang telah dibuat. Bagian *frontend* aplikasi menggunakan Vue.js sebagai antarmuka pengguna dan bagian *backend* aplikasi menggunakan Python dengan framework Flask sebagai *web service*.

IV. HASIL DAN PEMBAHASAN

A. Pendeteksian Jerawat

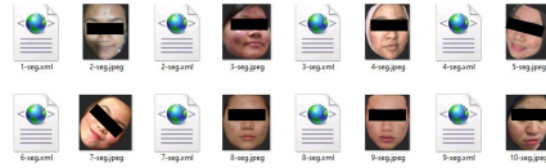
Gambar wajah orang yang memiliki masalah kulit jerawat yang telah dikumpulkan kemudian masuk ke tahap segmentasi kulit wajah untuk mendapatkan ROI. Gambar yang telah tersegmentasi kemudian disimpan dengan ukuran lebar 400 piksel dengan latar belakang gelap untuk memisahkan area kulit wajah pada gambar. Proses ini dilakukan otomatis menggunakan program, menghasilkan file-file gambar kulit wajah yang sudah tersegmentasi.



Gambar 4. Segmentasi wajah

Gambar kulit wajah yang tersegmentasi kemudian masuk ke tahap pelabelan objek jerawat. Pelabelan dilakukan dengan cara menandai area-area spesifik pada citra yang merupakan objek jerawat, dapat dilihat pada Gambar 3. Anotasi menggunakan format PASCAL VOC. Hasil dari proses pelabelan ini berupa file XML yang berisi daftar koordinat area untuk masing-masing gambar, dapat dilihat pada Gambar 5.

Setelah dataset jerawat diperoleh, dari sekian banyak data tersebut dipisahkan menjadi dataset pembelajaran dan dataset pengujian. Masing-masing dataset dipisahkan pada folder yang berbeda, *train* dan *test*.

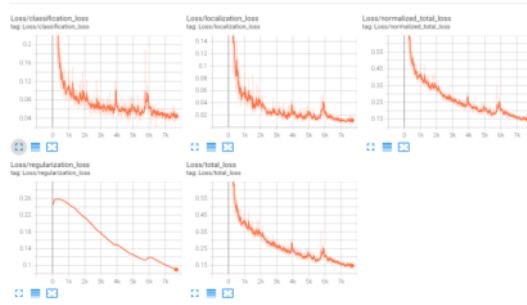


Gambar 5. Hasil pelabelan dengan LabelImg

TensorFlow memiliki banyak model yang sebelumnya dilatih untuk pendeteksian objek. Mempertimbangkan kinerja dan efisiensi komputasi, peneliti memilih menggunakan model SSD ResNet50 V1 FPN 640x640, karena memberikan *trade-off* yang relatif baik antara kinerja dan kecepatan.

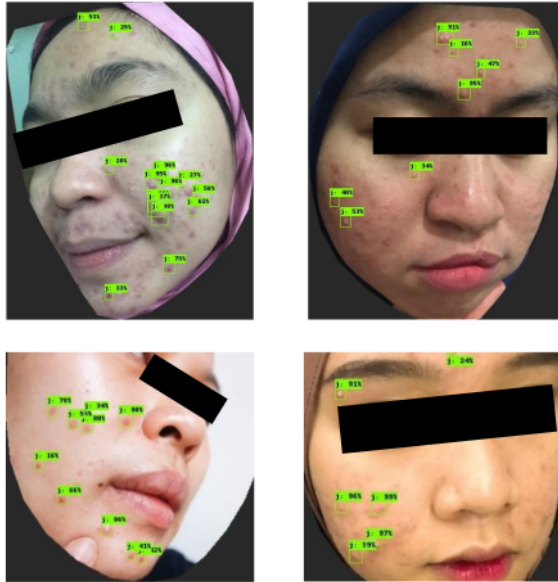
Langkah selanjutnya adalah menyiapkan *workspace* TensorFlow untuk pelatihan model pendeteksian objek. Persiapan meliputi pembuatan LabelMap, pembuatan TensorFlow Record, dan konfigurasi *pipeline* pelatihan model. Pembuatan LabelMap memetakan isi dari dataset yang dimiliki, file berisi daftar kelas-kelas objek. Konversi dataset ke dalam format TFRecord dilakukan menggunakan program Python yang menghasilkan file *train.record* dan *test.record* untuk masing-masing dataset. Kemudian konfigurasi *pipeline* pelatihan model yang meliputi pengaturan jumlah kelas objek, jumlah *batch size*, pengaturan evaluasi, pengaturan lokasi dataset, lokasi LabelMap, dan lokasi *checkpoint* dari model.

Setelah konfigurasi pipeline selesai kemudian dilanjutkan proses pelatihan model. Selama proses pelatihan nilai *total loss* dari model terus dipantau. Hasil dari proses pelatihan model ditunjukkan pada grafik Gambar 11. Grafik dari *classification loss* menunjukkan angka 0.04 dan *localization loss* menunjukkan angka 0.01 pada langkah ke 7687k sehingga *total loss* bernilai 0.147. Ini menunjukkan bahwa model sudah dapat mendeteksi lokasi jerawat dengan cukup akurat dari dataset yang dimiliki.



Gambar 6. Grafik pada TensorBoard

Model hasil pelatihan kemudian diuji untuk melakukan pendeteksian jerawat. Pengujian dilakukan menggunakan program Python dengan data pengujian sebagai *input*. Hasil pengujian menggunakan program menunjukkan bahwa model sudah dapat mendeteksi objek jerawat di area wajah dengan baik, dapat dilihat pada Gambar 7.



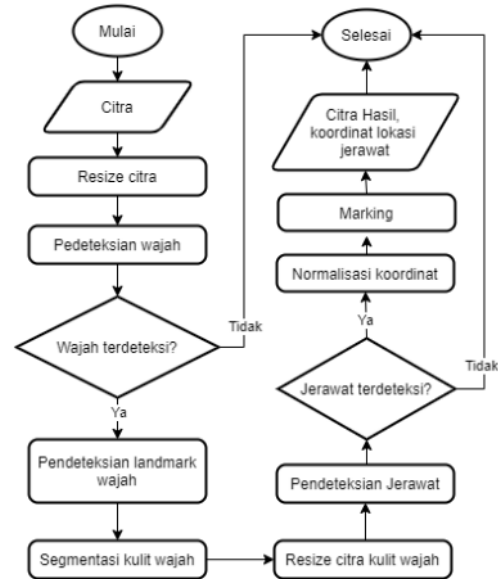
Gambar 7. Hasil pendeteksian jerawat

B. Pengembangan Aplikasi

Kebutuhan aplikasi meliputi kebutuhan program dan antarmuka aplikasi. Kebutuhan didasari oleh kebutuhan pengguna aplikasi yaitu untuk asesmen perawatan kulit berjerawat di klinik dermatologi. Berdasarkan identifikasi didapatkan kebutuhan umum aplikasi sebagai berikut,

- Aplikasi harus mampu mengunggah gambar atau mengambil foto dari kamera
- Aplikasi harus mampu melakukan segmentasi wajah
- Aplikasi harus mampu melakukan pendeteksian jerawat
- Aplikasi harus mampu mengirimkan dan menampilkan data hasil pendeteksian jerawat

Langkah selanjutnya adalah perancangan aplikasi. Perancangan arsitektur aplikasi menggunakan arsitektur *single-page application* (SPA). Aplikasi terbagi menjadi dua bagian yaitu *frontend* sebagai antarmuka di sisi pengguna dan *backend* di sisi server sebagai *web service*. Pertukaran data antara *frontend* dan *backend* menggunakan HTTP *methods* dan data berformat JSON. Perancangan alur program aplikasi dapat dilihat pada Gambar 8.



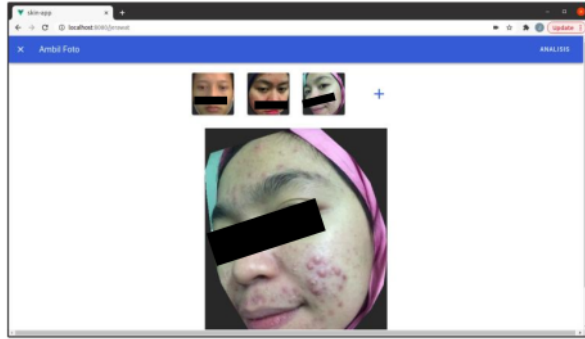
Gambar 8. Alur program aplikasi

Rancangan antarmuka dilakukan untuk merancang tampilan aplikasi yang nantinya digunakan pengguna. Pada aplikasi ini terdapat beberapa tampilan yaitu:

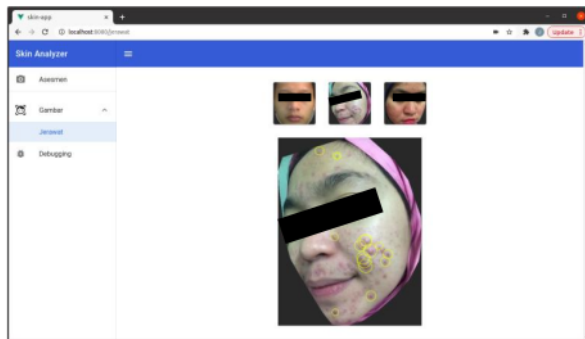
- Tampilan pengambilan gambar. Tampilan ini berfungsi untuk mengambil dan menampilkan gambar dari kamera pengguna. Pengambilan gambar dilakukan tiga kali untuk sisi wajah depan, serong kiri dan serong kanan.
- Tampilan hasil deteksi. Tampilan ini berfungsi untuk menampilkan hasil pendeteksian jerawat dari foto yang diambil.

Langkah selanjutnya adalah implementasi. Implementasi dimulai dari sisi *frontend* yaitu pembuatan antarmuka pengguna. Antarmuka dibuat menggunakan *framework* Javascript yaitu Vue.js dengan Vuetify sebagai penyedia komponen UI. Pada saat pengambilan gambar aplikasi menyediakan pilihan sumber yaitu dari kamera atau file.

Implementasi alur program aplikasi kemudian dilakukan di sisi *backend* sebagai *web service*. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan *framework* Flask. Di sisi *backend* aplikasi menyediakan *endpoint* untuk melakukan pendeteksian jerawat, *input* berupa foto atau gambar dan *output* berupa koordinat lokasi jerawat berformat JSON.



(a) Tampilan pengambilan gambar



(b) Tampilan hasil pendeteksian

Gambar 9. Implementasi Aplikasi

Pengembangan aplikasi menggunakan *web service* ini dapat dimanfaatkan untuk pengembangan aplikasi mobile, dengan mengkonsumsi *endpoint* yang ada. Arsitektur aplikasi dirancang untuk mendukung pengembangan aplikasi kedepannya, pendeteksian masalah kulit yang lain seperti keriput, kemerahan, dan kuli kusam. Hal tersebut dapat dilakukan dengan mengambahkan *endpoint* baru pada sisi *backend*.

V. KESIMPULAN

Aplikasi web untuk melakukan pendeteksian jerawat menggunakan algoritma *deep learning* pada TensorFlow sudah dapat mendeteksi jerawat pada foto wajah. Area kulit wajah diperoleh dengan efektif menggunakan pendeteksian wajah dikombinasikan dengan pendeteksian landmark wajah. Model pendeteksian jerawat diperoleh dari hasil *transfer learning* dari model SSD ResNet50 V1 FPN 640x640 yang dilatih dengan dataset jerawat, menghasilkan nilai *total loss* sebesar 0.147. Penggunaan model tersebut pada aplikasi sudah dapat mendeteksi jerawat di area kulit wajah dan menghitung jumlah jerawat. Namun, ada beberapa jerawat yang tidak terdeteksi dikarenakan jumlah dataset yang sedikit, perlu dataset lebih banyak. Model pendeteksian memiliki potensi untuk dilakukan peningkatan akurasi setelah aplikasi dirilis, aplikasi dapat mengumpulkan lebih banyak foto dari pengguna. Pelabelan

gambar baru kemudian bisa dilakukan kembali, ini bisa digunakan untuk mengulang pelatihan model.

DAFTAR PUSTAKA

- [1] health24. (2020). What is acne? Health24. <https://www.news24.com/health24/Medical/Acne/About-acne/Acne-20120721>.
- [2] Suva, M., "A Brief Review on Acne Vulgaris: Pathogenesis, Diagnosis, and Treatment", *Research & Reviews: Journal of Pharmacology*. Vol. 4, Issue 3, 2015.
- [3] MediLexicon International. (2017). Acne: Causes, treatment, and tips. *Medical News Today*. <https://www.medicalnewstoday.com/articles/107146>.
- [4] Acne: Diagnosis and treatment. American Academy of Dermatology. (n.d.). <https://www.aad.org/publications/seases/acne/derm-treat/treat>.
- [5] VISIA Skin Analysis, from <https://www.canfieldsci.com/imaging-systems/visia-complexion-analysis/>.
- [6] Prasetyo, D., Muhimmah, I., & Kurniawardhani, A. (2018). Aplikasi Pendeteksi Jerawat Di Wajah Dengan Menggunakan Teknik Pengolahan Citra Pada Foto.
- [7] Darmawan, A., Rositasari, A., & Muhimmah, I. (2020). The Identification System of Acne Type on Indonesian People's Face Image. *IOP Conference Series: Materials Science and Engineering*, 73(1). <https://doi.org/10.1088/1757-899X/803/1/012028>.
- [8] Alamdari, N., Tavakolian, K., Alhashim, M., & Fazel-Rezai, R. (2016). Detection and classification of acne lesions in acne patients: A mobile application. *IEEE International Conference on Electro Information Technology*, 2016-August, 739–743. <https://doi.org/10.1109/EIT.2016.7535331>.
- [9] Imran, W. T., & Lawi, A. (2020). Face image detection using haar cascade classifier. 1–5.
- [10] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1867–1874. <https://doi.org/10.1109/CVPR.2014.241>.
- [11] King, D. (2014). Real-Time Face Pose Estimation. *dlib C++ Library*. <http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>.
- [12] Tensorflow. TensorFlow. (n.d). <https://www.tensorflow.org/>.
- [13] What is TensorFlow? How it Works? Introduction & Architecture. guru99. (n.d.). <https://www.guru99.com/what-is-tensorflow.html>.
- [14] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January, 3296–3305. <https://doi.org/10.1109/CVPR.2017.351>.
- [15] Tensorflow Object Detection API. (n.d.). Retrieved from https://github.com/tensorflow/models/tree/master/research/object_detection.
- [16] Aplikasi web. Aplikasi web - Wikipedia bahasa Indonesia, ensiklopedia bebas. (2020). https://id.wikipedia.org/wiki/Aplikasi_web.
- [17] What is Application Architecture? Components, Models, and Types. Hackr.io. (2021). <https://hackr.io/blog/web-application-architecture-definition-models-types-and-more>.
- [18] TensorFlow 2 Object Detection API tutorial documentation. (2020). <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/>.



Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow

ORIGINALITY REPORT

16%

SIMILARITY INDEX

16%

INTERNET SOURCES

7%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

| | | |
|---|--|----|
| 1 | mti.binus.ac.id Internet Source | 2% |
| 2 | journal.uii.ac.id Internet Source | 2% |
| 3 | id.wikipedia.org Internet Source | 2% |
| 4 | arxiv.org Internet Source | 1% |
| 5 | dspace.uii.ac.id Internet Source | 1% |
| 6 | Alessio Bellino. "SEQUENCE: a remote control technique to select objects by matching their rhythm", Personal and Ubiquitous Computing, 2018 Publication | 1% |
| 7 | María Moncho Santonja, María Begoña Sanz Alamán, Beatriz Defez García, Ismael Lengua Lengua, Guillermo Peris-Fajarnés. "Analysis of segmentation methods for acne vulgaris | 1% |

images. Proposal of a new methodology applied to fluorescence images", Proceedings INNODOCT/19. International Conference on Innovation, Documentation and Education, 2019

Publication

| | | |
|----|---|------|
| 8 | www.mdpi.com Internet Source | 1 % |
| 9 | www.kafekepo.com Internet Source | 1 % |
| 10 | Masum Shah Junayed, Afsana Ahsan Jeny, Syeda Tanjila Atik, Nafis Neehal, Asif Karim, Sami Azam, Bharanidharan Shanmugam. "AcneNet - A Deep CNN Based Classification Approach for Acne Classes", 2019 12th International Conference on Information & Communication Technology and System (ICTS), 2019 Publication | 1 % |
| 11 | repozitorij.foi.unizg.hr Internet Source | 1 % |
| 12 | publikasi.dinus.ac.id Internet Source | <1 % |
| 13 | www.mayoclinic.org Internet Source | <1 % |
| 14 | github.com Internet Source | <1 % |

| | | |
|----|---|------|
| 15 | www.honestdocs.id Internet Source | <1 % |
| 16 | iopscience.iop.org Internet Source | <1 % |
| 17 | kc.umn.ac.id Internet Source | <1 % |
| 18 | repository.radenintan.ac.id Internet Source | <1 % |
| 19 | core.ac.uk Internet Source | <1 % |
| 20 | digilib.iain-palangkaraya.ac.id Internet Source | <1 % |
| 21 | eprints.undip.ac.id Internet Source | <1 % |
| 22 | fitnesfit.com Internet Source | <1 % |
| 23 | mulyadiveterinary.wordpress.com Internet Source | <1 % |
| 24 | qdoc.tips Internet Source | <1 % |
| 25 | www.spiedigitallibrary.org Internet Source | <1 % |

Exclude quotes Off

Exclude matches Off

Exclude bibliography On