

Klasifikasi Pakaian Berdasarkan Gambar Menggunakan Metode YOLOv3 dan CNN

Michael Christianto Wujaya, Leo Willyanto Santoso
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
E-Mail: mc.wujaya@gmail.com, leow@petra.ac.id

ABSTRAK

Pakaian merupakan salah satu dari kebutuhan primer manusia yang sangat dibutuhkan karena memiliki banyak fungsi. Tidak hanya untuk menutup dan melindungi tubuh, tetapi juga untuk tampil gaya dan *stylish*. Media masa, internet, dan media sosial merupakan tempat utama masyarakat dalam mencari inspirasi untuk tampil *fashionable*. Namun terkadang sulit untuk menentukan jenis suatu pakaian sehingga mudah dicari. Oleh karena itu sebuah program yang mampu membedakan dan mengklasifikasikan pakaian akan sangat membantu.

Metode yang digunakan adalah *You Only Look Once* untuk mendeteksi objek dari pakaian pada gambar yang setelah itu akan dipotong, dan hasilnya akan di olah oleh *Convolutional Neural Network* dengan model arsitektur ResNet50 untuk klasifikasi. Dalam proses pelatihan dari ResNet50 akan dilakukan tuning pada berbagai macam hal yang diantaranya adalah *learning rate*, *dropout*, *epoch*, *dense layer*, *freeze layer*, dan *data augmentation*. Setelah itu dilakukan pencarian gambar dengan menggunakan *k-nearest neighbor*.

Penelitian ini akan melakukan klasifikasi pada pakaian dalam suatu gambar yang dikenakan oleh model pada gambar. Akurasi rata-rata yang diperoleh dengan menggunakan ResNet50 yang telah di setel adalah 86,44%.

Kata Kunci: *Residual Network, You Only Look Once, Klasifikasi Gambar, Pakaian*

ABSTRACT

Clothing is one of the primary human needs and have many functions. It's function not solely to cover and protect the wearer, but also to look stylish. Mass media, the internet, and social media are the main place for people to find inspiration to look fashionable. But sometimes it is difficult to determine the type of clothing so it will be easy to find. Therefore, a program that is able to differentiate and classify clothes will be a great help.

The method we used are You Only Look Once to detect the clothing object from an image. The output of detection will be cropped and the result will be processed and classified by Convolutional Neural Network using ResNet50 architecture. In the training process of ResNet50, various things will be tuned which is learning rate, dropout, epoch, number of dense layer and its value, freezing layer,

and data augmentation. Then program will search similar image using k-nearest neighbor.

The result of this study will classify clothes in an image that is worn by the model in the image. The average accuracy obtained using the fine-tuned ResNet50 is 86.44%.

Keywords: *Residual Network, You Only Look Once, Image Classification, Clothing.*

1. PENDAHULUAN

Belanja *online* dan *e-commerce* saat ini mengalami perkembangan yang sangat pesat. Diperhitungkan dalam empat tahun terakhir pertumbuhan *e-commerce* di Indonesia mencapai 500 persen. Pertumbuhan yang luar biasa ini diprediksi transaksi *e-commerce* akan mencapai angka 12 miliar pada tahun 2020 [8]. Sebagian besar pertumbuhan yang terjadi disebabkan oleh transaksi produk dalam kategori *fashion* baik berupa produk kecantikan maupun pakaian jadi. Berdasarkan data yang ada pada salah satu *e-commerce* dimana transaksi kategori *fashion* mencapai 70% lebih banyak dibandingkan kategori lainnya [2]. Tidak hanya itu, internal dari *e-commerce* juga menyatakan bahwa salah satu dari lima kategori terpopuler adalah *fashion* [4].

Perkembangan yang pesat dari sistem penjualan menjadi salah satu pendukung penyebaran dunia *fashion* di Indonesia. Banyaknya transaksi yang terjadi ini bukan hanya dikarenakan pakaian merupakan kebutuhan primer bagi manusia untuk menutup dan melindungi dari sekitar, tetapi juga untuk tampil gaya dan *stylish*. Media massa, internet, dan sosial media menjadi tempat dimana orang-orang dapat mencari inspirasi untuk tampil *fashionable*. Namun menemukan produk yang diinginkan bukanlah hal yang mudah. Hal ini dikarenakan penggunaan kata kunci pada pencarian tidak sepenuhnya akurat dalam mengungkapkan permintaan konsumen terhadap suatu produk.

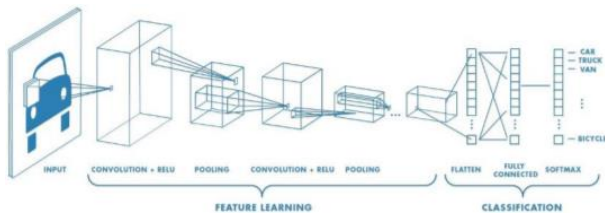
Pada penelitian ini, pendekatan dalam melakukan klasifikasi produk pakaian pada suatu gambar/foto dilakukan dengan metode YOLOv3 dan CNN dengan menggunakan arsitektur ResNet50. Penggunaan YOLOv3 dilakukan untuk pencarian posisi dan letak dari produk *fashion* pada gambar. Hasil dari deteksi ini nantinya akan digunakan untuk melakukan klasifikasi dengan menggunakan CNN. Klasifikasi pada CNN ini akan dilakukan dengan salah satu arsitektur dari CNN yang memiliki keunggulan dalam performa akurasi yaitu ResNet-50. Hasil dari klasifikasi nantinya akan digunakan untuk pencarian gambar menggunakan *k-nearest neighbor*.

2. DASAR TEORI

2.1 Convolutional Neural Network

Convolutional neural network (ConvNet/CNN) merupakan algoritma *deep learning* yang dapat menerima input berupa gambar dan dapat melakukan pengenalan terhadap berbagai aspek/objek dari suatu gambar. Pengenalan ini nantinya akan diklasifikasikan untuk membedakan satu gambar dengan yang lainnya dari suatu kategori seperti perabot, binatang, kendaraan, dsb [5]. Pengenalan ini dilakukan dengan memproses data pada serangkaian *pixel* yang tersusun dalam *grid* yang ada pada gambar. Setiap *pixel* ini sendiri memiliki nilai yang menunjukkan tingkat terang dan warna.

Secara teknis proses CNN terdiri dari proses pelatihan dan *testing*. Gambar yang diinputkan akan melalui serangkaian *layer* untuk mengklasifikasi objek di dalamnya. Rangkaian ini terdiri dari *convolutional layers*, *pooling layer*, dan *fully connected layer*. Arsitektur dari CNN dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur CNN[5]

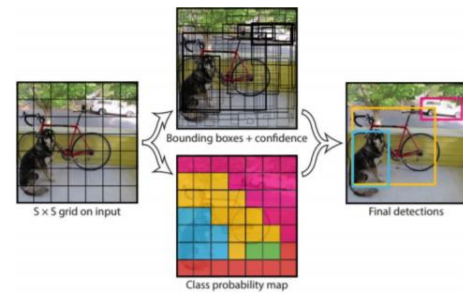
Convolution layer merupakan *layer* pertama untuk melakukan ekstraksi fitur dari suatu gambar. *Layer* ini nantinya akan mempelajari fitur yang ada dari hubungan setiap *pixel* dengan *filter matrix* yang umumnya 3×3 . Dimana nantinya matriks ini akan melakukan yang disebut dengan *stride* atau pergeseran untuk melakukan konvolusi di setiap petak pada gambar. Proses konvolusi dari suatu gambar dengan *filter* yang berbeda dapat melakukan operasi seperti deteksi tepi, *blur*, dan penajaman gambar.

Rectified Linear Unit (ReLU) merupakan operasi untuk mengenalkan *nonlinearity* pada CNN. Tujuan dari ReLU adalah untuk mengubah nilai data *negative* menjadi 0. *Pooling layer* adalah proses yang dilakukan untuk pengecilan matriks yang dilakukan ketika gambar terlalu besar. Hal ini dilakukan untuk mengurangi dimensi pada setiap *map* namun tetap mempertahankan informasi yang penting. *Pooling* ini memiliki beberapa tipe seperti *max pooling*, *average pooling*, dan *sum pooling*.

Sementara *fully-connected layer* merupakan sebuah lapisan yang menghubungkan semua *neuron* dari lapisan sebelumnya menjadi seperti jaringan syaraf. Lapisan ini digunakan sebagai pengolahan data untuk melakukan klasifikasi.

2.2 You Only Look Once

You only look once (YOLO) merupakan jaringan untuk melakukan deteksi objek pada suatu gambar. Mendeteksi objek terdiri dari menentukan lokasi pada gambar dimana terdapat objek tertentu. YOLO menerapkan *single neural network* pada gambar dan akan membagi gambar menjadi wilayah-wilayah yang kemudian memprediksi *bounding box*. Setiap *bounding box* ditimbang probabilitasnya untuk mengklasifikasikan sebagai objek atau bukan [7]. Diagram dari algoritma YOLO dapat dilihat pada Gambar 2.



Gambar 2. Diagram model algoritma YOLO[7]

Beberapa pengembangan sudah dilakukan pada YOLO, salah satunya adalah YOLOv3 yang dibuat dengan *classifier network* baru yang lebih baik dari sebelumnya. Pada YOLOv3 prediksi *bounding box* dilakukan menggunakan dimensi kluster sebagai *anchor box*. Setiap *bounding box* akan diprediksi dengan empat buah koordinat oleh jaringan [3]. Setiap *box* akan memprediksi kelas dari *bounding box* dengan menggunakan *multilabel classification*. Prediksi nilai suatu objek dilakukan berdasarkan nilai pada setiap *box* menggunakan regresi logistik tanpa dengan *softmax*. Pada masa *training* dilakukan dengan menggunakan *binary cross-entropy loss* pada prediksi kelas. Formulasi ini membantu ketika memindahkan area yang kompleks seperti *Open Images Dataset*. Penggunaan *softmax* memaksa setiap *box* harus memiliki hanya satu kelas. Sedangkan *multilabel* memberikan pendekatan lebih baik pada *model data*.

Pada YOLOv3 prediksi *box* dilakukan pada tiga skala. Sistem akan mengekstraksi fitur dari dari setiap skala dengan menggunakan konsep fitur dari 11 jaringan piramida. Pada ekstraksi fitur yang ada ditambahkan beberapa *layer* konvolusi. Pada eksperimen dengan COCO, YOLOv3 memprediksi 3 *box* pada setiap skala, jadi *tensor* $N \times N \times (3 \times (4+1+80))$ untuk mengimbangi 4 buah *bounding box*, 1 prediksi objek, dan 80 prediksi kelas. Mengambil *feature map* dari dua *layer* sebelumnya dan *up-sample* 2x. Dengan mengambil *feature map* dari sebelumnya pada jaringan dan menggabungkan dengan sampel yang telah *up-sample*, akan memungkinkan untuk mendapat lebih banyak informasi semantik. Informasi yang dihasilkan akan lebih halus dari sebelumnya.

	Type	Filters	Size	Output
1x	Convolutional	32	3×3	256×256
	Convolutional	64	$3 \times 3 / 2$	128×128
	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			128×128
2x	Convolutional	128	$3 \times 3 / 2$	64×64
	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			64×64
8x	Convolutional	256	$3 \times 3 / 2$	32×32
	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Residual			32×32
8x	Convolutional	512	$3 \times 3 / 2$	16×16
	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Residual			16×16
4x	Convolutional	1024	$3 \times 3 / 2$	8×8
	Convolutional	512	1×1	
	Convolutional	1024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1000	
	Softmax			

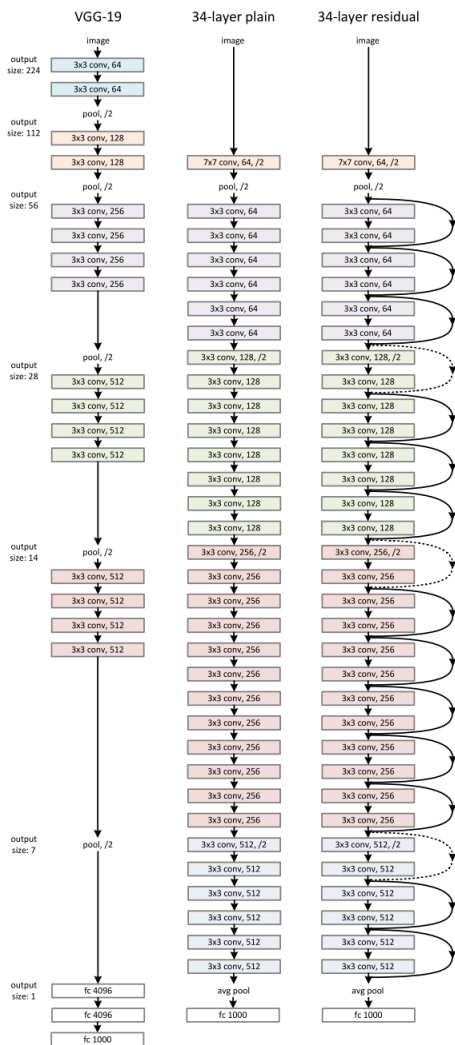
Gambar 3. Lapisan konvolusi pada YOLOv3[8]

Ekstraksi fitur pada YOLOv3 dilakukan dengan menggabungkan pendekatan dari jaringan yang digunakan pada YOLOv2, Darknet-19, dan lapisan residu. Pada jaringan ini berhasil menggunakan 3 x 3 dan 1 x 1 lapisan konvolusi, tapi dengan memiliki koneksi langsung dan lebih besar. Lapisan konvolusi dari YOLOv3 dapat dilihat pada Gambar 3. Karena pada memiliki 53 lapisan konvolusi sehingga dapat disebut dengan Darknet-53 [8].

Performa dari pendekatan ini sudah dicoba pada ImageNet. Darknet-53 berjalan setara dengan state-of-the-art classifier tapi memiliki operasi floating point lebih sedikit dan lebih cepat. Darknet-53 juga sudah terbukti 1.5x lebih cepat dari ResNet-101 dimana Resnet memiliki layer terlalu banyak dan tidak efisien.

2.3 Deep Residual Network

Deep Residual Network (ResNet) merupakan salah satu dari arsitektur pada convolutional neural network yang diusulkan oleh Kaimin He, Xiangyu Zhang, Shaoqin Ren, dan Jian Sun pada tahun 2015. Arsitektur ini dibangun untuk mengatasi permasalahan pada pelatihan deep learning dimana pelatihan umumnya memakan waktu yang cukup lama dan terbatas pada jumlah lapisan tertentu. ResNet memberikan solusi dengan menerapkan skip connection atau shortcut. Skip connection dilakukan pada dua sampai tiga layer yang mengandung ReLu [1].



Gambar 4. Perbandingan jaringan biasa dengan ResNet[1]

Residual block pada ResNet dapat dilakukan apabila dimensi data input sama dengan dimensi data output. Setiap blok dari ResNet terdiri dari dua layer seperti pada ResNet 18 dan ResNet 34 atau tiga layer seperti pada ResNet 50, ResNet 101, ResNet 152. ResNet merupakan arsitektur dari CNN, sehingga layer-layer yang ada pada ResNet terdiri dari convolutional layer, pooling layer, dan fully connected layer (fc). Perbandingan dari jaringan biasa dengan ResNet dapat dilihat pada Gambar 4.

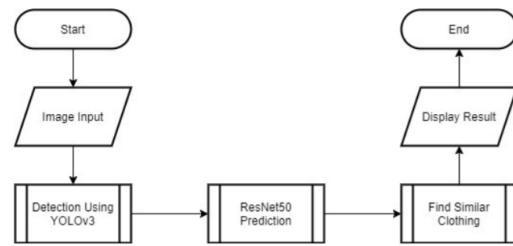
2.4 K-Nearest Neighbor

K-Nearest Neighbor adalah sebuah metode untuk klasifikasi terhadap data berdasarkan dari pembelajaran data yang sudah di klasifikasikan sebelumnya. Proses pembelajaran ini dilakukan pada mayoritas dari kategori tetangga K terdekat. Proses ini dilakukan dengan cara mencari pola dalam data training yang memiliki kemiripan dengan menghitung jarak perbedaannya. Dekat atau jauhnya dihitung berdasarkan Euclidian Distance dimana perhitungan berkaitan dengan teorema pythagoras. Dimana formula dari Euclidian adalah mencari jarak antar kedua titik dalam dua ruang dimensi [6].

3. DESAIN SISTEM

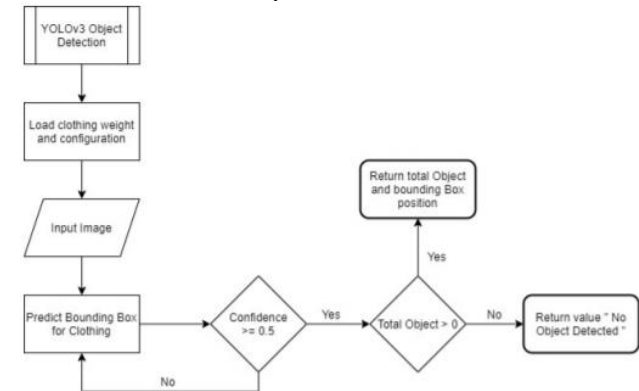
3.1 Analisis Sistem

Pertama sistem akan menerima input data dari user yang berupa suatu gambar dari pakaian. Gambar tersebut nantinya akan diproses oleh sistem untuk melakukan deteksi pakaian pada gambar dengan menggunakan YOLOv3. Setelah itu gambar yang objeknya telah diprediksi menggunakan YOLO akan diproses lagi untuk dirubah ukurannya menjadi 224x224. Setelah berhasil gambar akan di proses menggunakan ResNet50 yang telah dilatih sebelumnya. Setelah proses selesai hasil dari klasifikasi akan dilakukan pencarian gambar pada kelasnya berdasarkan warna pada pakaian itu dengan menggunakan k-nearest neighbor. Arsitektur dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur sistem

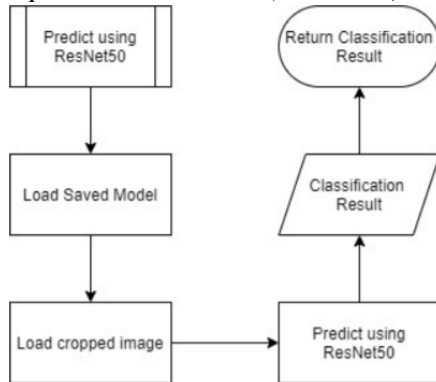
3.1.1 Proses You Only Look Once



Gambar 6. Flowchart proses YOLOv3

Proses yang terjadi pada bagian *You Only Look Once* ini merupakan proses untuk mendeteksi objek pada gambar yang di-input. Pada proses ini dapat dilihat pada Gambar 6. Objek yang dideteksi adalah pakaian yang ada pada gambar. Pakaian yang terdeteksi berupa jenis pakaian yaitu atasan, bawahan, maupun pakaian terusan. Dimana prediksi akan dilakukan ke keseluruhan area gambar untuk diprediksi *class* dan *bounding box*-nya. Selanjutnya akan dilakukan pengecekan tingkat keyakinan, dimana objek dengan *bounding box* yang memiliki tingkat *confidence* yang tinggi akan diambil dan di-label sebagai objek. Setelah ojek dengan *confidence* yang tinggi ditemukan akan dipotong dan hasilnya akan dilanjutkan dengan ResNet untuk mengolah lebih lanjut.

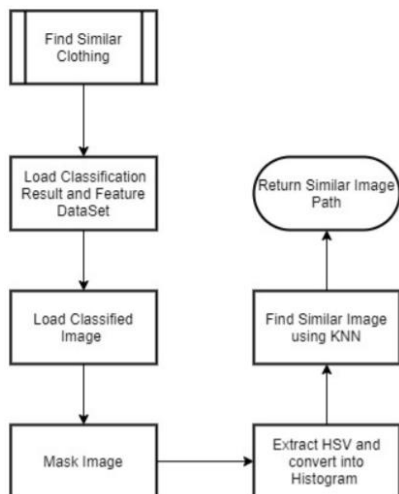
3.1.2 Deep Residual Network (ResNet50)



Gambar 7. Flowchart proses ResNet50

Setelah gambar yang di-input diproses dengan menggunakan *You Only Look Once* selanjutnya akan diproses dengan menggunakan ResNet50. Alur dari proses kerja dapat dilihat pada Gambar 7. Hal pertama yang akan dilakukan ketika suatu gambar telah masuk adalah mengubah ukuran menjadi 224×224 pixel. Setelah gambarnya telah diubah ukurannya akan langsung diolah dengan menggunakan ResNet50 dan nantinya hasil dari proses ini akan memberikan *output* berupa klasifikasi dari pakaian yang ada pada gambar *input*.

3.1.3 Pencarian Gambar



Gambar 8. Flowchart proses pencarian

Setelah ResNet berhasil melakukan klasifikasi terhadap gambar. Langkah selanjutnya adalah melakukan pencarian untuk gambar yang mirip. Hal ini dilakukan dengan memberikan *masking* berbentuk elips terlebih dahulu pada gambar agar proses ekstraksi

fitur warna lebih baik. Setelah di berikan *masking*, proses selanjutnya adalah melakukan ekstraksi fitur HSV dan merubahnya menjadi *histogram data* untuk memudahkan pencarian. Pencarian akan dilakukan dengan menggunakan *k-nearest neighbor* dengan menggunakan *metric Euclidian*. Alur program dapat dilihat pada Gambar 8.

3.2 Desain Sistem

Pada bagian ini akan dijelaskan tentang *user interface* dari program yang telah dibuat. Program ini akan berupa aplikasi *web* yang dibuat dengan satu halaman untuk melakukan *input* gambar dan juga mengeluarkan hasilnya. Prosesnya berupa *user* akan *upload* gambar yang ingin diklasifikasikan. Pada halaman yang sama nanti akan ditampilkan hasil dari klasifikasi dan gambar dari pakaian pada kelas itu beserta produk yang memiliki warna yang sama. Format *input* dari *user* nantinya adalah *file* yang memiliki ekstensi .JPG.

4. IMPLEMENTASI SISTEM

Sistem yang dibuat diimplementasikan pada komputer dengan spesifikasi sistem operasi *windows 10*, RAM 8gb, CPU *Intel Core i7-7700HQ* dan GPU *NVIDIA GeForce GTX 950M*.

Bahasa pemrograman yang digunakan pada penelitian ini adalah *python* dengan versi 3.8. Penelitian ini dibantu dengan *framework tensorflow 2.2*. Adapun beberapa *library* yang digunakan untuk mendukung sistem ini ialah *flask*, *CV2*, *numpy*, *pickle*, *scikit-learn*, dan *matplotlib*.

5. ANALISA DAN PENGUJIAN

5.1 Pengujian Sistem

Pengujian sistem dilakukan dengan mengevaluasi performa dari sistem *training* dan *testing* yang telah dibuat. Pencatatan akan dibedakan berdasarkan jenis pakaian yang diprediksi dan diklasifikasikan.

5.1.1 Pengujian Metode YOLOv3

Pengujian dengan metode YOLO dilakukan terhadap 23 kelas pakaian yang berbeda. Dengan Jumlah iterasi atau *epoch* sebesar 6000. Untuk mendapatkan akurasi prediksi pakaian dilakukan dengan menggunakan rumus (1) dan hasil akurasi dapat dilihat pada Tabel 1.

$$Akurasi = \frac{\text{Jumlah gambar terdeteksi}}{\text{Jumlah total gambar}} \times 100 \quad (1)$$

Tabel 1. Pengujian prediksi dengan YOLOv3

Class	Jumlah Gambar	Terdeteksi	Akurasi
Kaos lengan panjang	867	773	89.16%
Kaos lengan pendek	1196	990	82.78%
Kaos tanpa lengan	868	665	76.61%
Polo	1008	914	90.67%
Kemeja lengan pendek	687	554	80.64%
Kemeja lengan panjang	1041	943	90.59%
Blouse	1083	792	73.13%

Hoodies	1058	1042	98.49%
Sweater	860	719	83.60%
Jaket kulit	495	453	98.49%
Jaket denim	886	764	86.23%
Cardigan	921	793	86.10%
Coat	968	920	95.04%
Blazer	1035	913	88.21%
Jeans	947	533	56.28%
Short	992	714	71.98%
Trouser	706	464	65.72%
Skirts	762	520	68.24%
Track pants	499	411	82.36%
Legging	1149	702	61.10%
Dress	893	689	77.16%
Jumpsuit	931	815	87.54%
Romper	800	673	84.13%

5.1.2 Pengujian Model ResNet50

Pengujian pada model ResNet50 dilakukan dengan melihat performa dari akurasi, presisi, *recall* dan *f-score*. Dalam proses pengujian konfigurasi yang digunakan adalah konfigurasi yang telah diuji dan memiliki akurasi terbaik. Konfigurasi ini merupakan konfigurasi dari model yang telah di *fine-tune*. Total dari rata-rata akurasi yang diberikan adalah 86,44%. Hasil dari akurasi, presisi, *recall* dan *f-score* setiap kategori dari hasil model yang telah diberlakukan *fine-tuning* dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian *fine-tuned* ResNet50

Kategori	Akurasi	Presisi	Recall	F-Score
Blazer	87.11%	0.79	0.87	0.83
Blouse	71.13%	0.71	0.71	0.71
Cardigan	72.95%	0.73	0.73	0.73
Coat	86.60%	0.81	0.87	0.84
Dress	69.23%	0.61	0.69	0.65
Hoodies	98.47%	0.95	0.98	0.97
Jaket denim	86.01%	0.86	0.86	0.86
Jaket kulit	86.99%	0.81	0.87	0.84

Jeans	90.23%	0.94	0.90	0.92
Jumpsuit	89.18%	0.94	0.89	0.92
Kaos lengan panjang	75.00%	0.81	0.75	0.78
Kaos lengan pendek	93.73%	0.95	0.94	0.78
Kemeja lengan panjang	95.61%	1.00	0.96	0.98
Kemeja lengan pendek	99.33%	0.90	0.99	0.95
Legging	93.43%	0.93	0.93	0.93
Polo	97.86%	0.99	0.98	0.98
Rok	94.70%	0.92	0.95	0.93
Romper	58.90%	0.71	0.59	0.64
Short	96.95%	0.94	0.97	0.95
Sweater	55.62%	0.65	0.56	0.60
Tank top	95.27%	0.94	0.95	0.95
Track pants	96.81%	0.90	0.97	0.93
Trouser	90.76%	0.98	0.91	0.94

5.1.3 Pengujian Klasifikasi Pada Gambar Blur

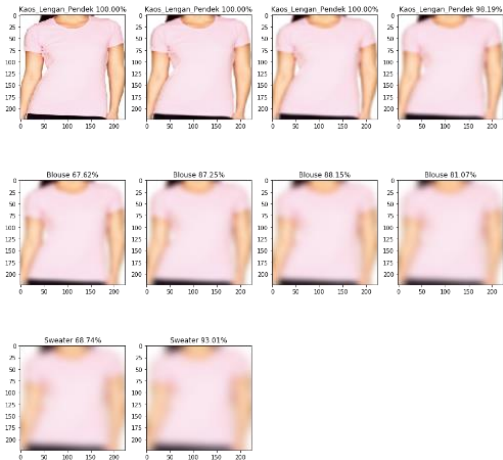
Pengujian ini dilakukan pada model ResNet50 dalam melakukan klasifikasi pada gambar yang telah diberikan efek *blur*. Gambar *blur* ini merupakan gambar dari *dataset test* yang sama namun diaplikasikan *blur* buatan. Gambar *blur* ini dibuat dengan menggunakan salah satu fungsi pada PIL dengan nilai 1 sampai 10. Hasil perhitungan dari klasifikasi *blur* dapat dilihat pada Tabel 3. Tabel 3 berisikan total dari akurasi keseluruhan pada setiap *value blur* yang ada.

Tabel 3. Hasil pengujian pada gambar *blur*

	Akurasi	Presisi
1	83.67%	84.28%
2	73.95%	78.51%
3	61.20%	72.18%
4	50.29%	64.70%
5	43.47%	59.80%
6	39.07%	56.00%
7	35.76%	-
8	32.87%	-
9	30.48%	-
10	28.78%	-

Hasil dari deteksi memiliki penurunan yang cukup drastis untuk setiap nilai dari *blur*-nya. Efektifitas dari klasifikasi menurun lebih

dari 10% dari setiap *level* dari *blur*. Berdasarkan hasil ini dapat dilihat bahwa efektivitas deteksi hanya mampu mentolerir gambar yang memiliki nilai *blur* 2 sampai 4. Contoh hasil klasifikasi dapat dilihat pada Gambar 9.



Gambar 9. Hasil klasifikasi pada gambar yang di-blur

5.1.4 Pengujian Pada Pencarian Gambar

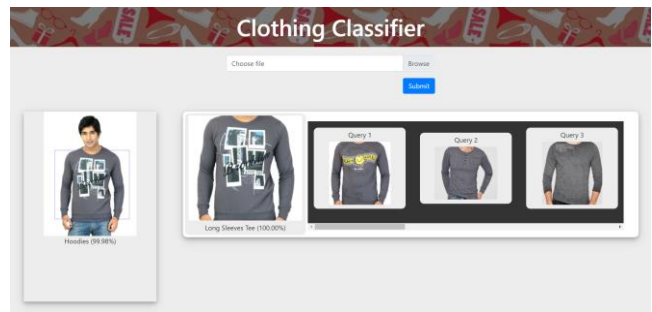
Pada penelitian ini proses proses training dari ResNet50 dilakukan berdasarkan 23 kategori yang telah diatur dan tidak dipisahkan setiap warnanya. Untuk itu hasil pencarian yang dilakukan hanya dengan menggunakan *output* dari klasifikasi memberikan gambar yang sama dengan kategori hasil deteksi namun warnanya tidak sesuai. Untuk itu diperlukan ekstraksi warna tambahan yang dilakukan dengan menggunakan *library* pada *opencv*. Dilakukan dengan cara memberikan *masking* untuk memfokuskan pada objek pakaian setelah itu dilakukan ekstraksi *histogram* warna. Hasil dari pencarian dapat dilihat pada Gambar 10.



Gambar 10. Flowchart proses pencarian

5.2 Pengujian Aplikasi

Pengujian Aplikasi dilakukan dengan melakukan prediksi dan klasifikasi berdasarkan *user interface* yang telah dibuat. Pengujian akan dilakukan pada setiap kategori dengan gambar pada *dataset testing* yang diambil secara acak. Sistem aplikasi ini memerlukan waktu sekitar 30 detik untuk dijalankan dalam melakukan deteksi dan klasifikasi. Berikut adalah tampilan *user-interface* hasil klasifikasi yang dilakukan pada kaos lengan panjang. Tampilan dari *user-interface* dapat dilihat pada Gambar 11.



Gambar 11. Gambar dari *user-interface*

6. KESIMPULAN

Berdasarkan hasil pengamatan dan pengujian terhadap sistem, dapat disimpulkan beberapa hal sebagai berikut:

- Berdasarkan hasil pengujian pada sistem dan aplikasi dapat disimpulkan bahwa YOLOv3 yang dilatih dari *pre-trained* untuk mendeteksi lokasi pakaian mendapat tingkat akurasi 82.5% menggunakan *dataset* yang telah dipilah dari *dataset* yang ada.
- Hasil deteksi YOLOv3 dengan *dataset* yang ada memiliki kekurangan belum mampu mendeteksi semua objek pakaian pada gambar. Terkadang hanya mampu mendeteksi satu atau dua saja.
- Hasil dari klasifikasi ResNet50 setelah di-*tuning* yang dilakukan pada *dataset test* mampu memberikan hasil yang cukup baik dengan tingkat akurasi 86.44%, presisi 86.05%, *recall* 86.16% dan *f-score* 86.11%.
- Berdasarkan pengujian model dari ResNet50 yang telah di-*tuning* hanya mampu mentolerir gambar yang memiliki nilai *blur* pada *PIL* dari 1 sampai 4 yang tergolong *blur* ringan.

Saran yang dapat diberikan untuk menyempurnakan dan mengembangkan aplikasi ini lebih lanjut antara lain:

- Penambahan *dataset* yang lebih bervariasi dan memiliki resolusi yang baik untuk klasifikasi terutama pada kategori *sweater* dan *romper*.
- Meneruskan aplikasi dengan menambahkan klasifikasi objek lain selain pakaian.
- Menggunakan *pre-trained* model lain selain ResNet50 untuk dapat mencari akurasi yang lebih baik.

7. DAFTAR PUSTAKA

- [1] He, K., Zhang, X., Ren, S., & Sun, J. 2015. Deep Residual Learning for Image Recognition.
- [2] Jatmiko, B. P. 2019. Fashion Jadi Produk Terlaris dan Menguntungkan bagi E-Commerce. Kompas. Retrieved from <https://money.kompas.com/read/2019/07/12/193100126/fashion-jadi-produk-terlaris-dan-menguntungkan-bagi-e-commerce>
- [3] Liu, R., Yan, Z., Wang, Z., & Ding, S., 2019. An Improved YOLOV3 for Pedestrian Clothing Detection, 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2019, pp. 139-143, doi: 10.1109/ICSAI48974.2019.9010512.
- [4] Mahadhirka, W. A. 2020. Ini Kategori Produk yang Paling Dicari di E-Commerce 2019. Kompas. Retrieved from <https://money.kompas.com/read/2020/01/09/165909226/ini-kategori-produk-yang-paling-dicari-di-e-commerce-2019>

- [5] Prabhu. 2018. Understanding of Convolutional Neural Network (CNN) — Deep Learning. Retrieved from <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>
- [6] Ramadhani, R. D. 2019. Memahami K-Nearest Neighbor (KNN) Dengan R . Retrieved from <https://medium.com/@16611129/memahami-k-nearest-neighbor-knn-dengan-r-de5280439053>
- [7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. You Only Look Once : Unified, Real-Time Object Detection
- [8] Redmon, J., & Farhadi, A. 2018. YOLOv3: An Incremental Improvement
- [9] Widiarini, A. D. 2019. Terus Tumbuh Signifikan, Begini Masa Depan E-Commerce di Indonesia. Kompas. Retrieved from <https://money.kompas.com/read/2019/12/10/110500326/terus-tumbuh-signifikan-begini-masa-depan-e-commerce-di-indonesia>