

## Identifikasi Persediaan Makanan di dalam Lemari Pendingin Berbasis *Raspberry Pi* dan *Deep Learning*

Faisal Candrasyah Hasibuan<sup>1</sup>, Andri Ulus Rahayu<sup>2</sup>

Program Studi S1 Teknik Komputer Fakultas Teknik Elektro Universitas Telkom, Bandung  
Jurusan Teknik Elektro Universitas Siliwangi, Tasikmalaya

<sup>1</sup>faicanhasfcb@telkomuniversity.ac.id

<sup>2</sup>andriulusr@unsil.ac.id

**Intisari** — Sistem ini dibuat atas dasar permasalahan yang terjadi dalam kehidupan sehari-hari, salah satunya yaitu tidak terpantaunya persediaan bahan makanan di lemari pendingin. Ketika dibutuhkan suatu bahan makanan dari lemari pendingin dan ternyata tidak ada, maka akan menjadi masalah. Oleh karena itu, dibuatlah sebuah sistem yang mampu mengidentifikasi objek makanan di dalam lemari pendingin. Masukan dari sistem ini berupa foto objek makanan yang diambil menggunakan *Raspberry Pi* Camera dan terhubung langsung dengan *Raspberry Pi* di dalam lemari pendingin. Setelah diproses dengan algoritma pembelajaran mesin, maka keluaran yang dihasilkan berupa identifikasi objek makanan yang terdapat di dalam lemari pendingin tersebut. Objek makanan yang diuji berupa pisang, mentimun, brokoli, dan jeruk. Dari hasil pengujian, terlihat bahwa program mengidentifikasi objek dengan benar pada objek pisang dan jeruk yang ditunjukkan dengan *confidence level* tertinggi sebesar 56,98% dan 45,88%. Identifikasi objek mentimun dikenali sebagai zucchini dengan *confidence level* tertinggi sebesar 78,61%. Adapun identifikasi objek paling rendah terdapat pada objek brokoli dengan *confidence level* kurang dari 1%.

**Kata kunci** — *Deep Belief Network*, *Deep Learning*, Lemari Pendingin, *Machine learning*, *Raspberry Pi*.

**Abstract** — This system is made based on problems that occur in everyday life, one of which is the lack of monitoring of food supplies in the refrigerator. It will be a problem when there is a need for a food item from the refrigerator and it does not exist. Therefore, a system is made that can identify food objects in the refrigerator. The input from this system is in the form of photos of food objects taken using the *Raspberry Pi* Camera and connected directly to the *Raspberry Pi* in the refrigerator. After being processed with machine learning algorithms, the resulting output identifies food objects in the refrigerator. The food objects tested were bananas, cucumbers, broccoli, and oranges. The test results show that the program correctly identified the object on the banana and orange object, which was indicated by the highest confidence levels of 56.98% and 45.88%, respectively. The identification of the cucumber object was recognized as zucchini with the highest confidence level of 78.61%. The lowest object identification was found in broccoli, with a confidence level of less than 1%.

**Keywords**— *Deep Belief Network*, *Deep Learning*, *Deep Belief Network*, *Machine learning*, *Raspberry Pi*, Refrigerator.

## I. PENDAHULUAN

Sistem ini dibuat atas dasar permasalahan yang terjadi dalam kehidupan sehari-hari. Salah satunya yaitu tidak terkontrolnya persediaan bahan makanan di lemari pendingin. Pengguna harus melihat secara langsung untuk mengetahui sisa persediaan bahan makanan yang disimpan[1]. Masalah akan timbul jika baru diketahuinya persediaan makanan yang dibutuhkan telah habis atau ada bahan makanan yang sudah terlalu lama disimpan. Oleh karena itu, dibuatlah sebuah sistem kecerdasan buatan yang mampu mengidentifikasi objek makanan di dalam lemari pendingin.

Beberapa penelitian yang menggunakan algoritma kecerdasan buatan, pembelajaran mesin, dan pengenalan citra, di antaranya ada yang memanfaatkan jaringan syaraf tiruan untuk mengidentifikasi tingkat kematangan buah pisang[2]. Ada pula yang memanfaatkan *deep belief network* untuk pengenalan plat nomor kendaraan [3]. Raspberry Pi digunakan sebagai pengendali dan pemroses utama karena sudah paket lengkap, memiliki kemampuan GPIO seperti Arduino, tetapi memiliki prosesor *multicore* yang mendukung komputasi kompleks. Penelitian-penelitian yang menggunakan Raspberry Pi untuk keperluan sistem otomatisasi ataupun pembelajaran mesin seperti sistem pemberi pakan kucing otomatis yang dapat dimonitor melalui aplikasi *mobile*[4], membedakan benda berdasarkan warna dan bentuk[5], pemrosesan gambar untuk sistem kamera keamanan[6], dan pendeteksi sidik jari[7].

Masukan dari sistem ini adalah foto objek makanan yang diambil menggunakan Raspberry Pi Camera yang terhubung langsung dengan Raspberry Pi di dalam lemari pendingin. Keluaran yang dihasilkan berupa identifikasi objek makanan yang terdapat di dalam lemari pendingin tersebut. Harapan lebih lanjut dari sistem ini adalah terhubung langsung dengan sistem belanja daring yang ada di supermarket. Ketika persediaan makanan yang ada di lemari pendingin habis, lemari pendingin tersebut dapat memesan bahan makanan secara otomatis ke supermarket langganan pengguna.

## II. TINJAUAN PUSTAKA

### A. Pembelajaran Mesin (*Machine Learning*)

Pembelajaran mesin (*machine learning*) merupakan cabang dari keilmuan kecerdasan buatan, adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor dan basis data. Sistem pembelajar dapat memanfaatkan contoh data untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya. Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variabel yang diamati. Fokus besar penelitian pembelajaran mesin adalah cara mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua perilaku yang mungkin, dari semua masukan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan yang disebut data pelatihan. Oleh karena itu, pembelajar harus merampatkan perilaku dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru.

Algoritma dalam pembelajaran mesin dapat dikelompokkan berdasarkan masukan dan keluaran yang diharapkan dari algoritma. Ada beberapa tipe algoritma, di antaranya adalah pembelajaran terarah (*supervised learning*), pembelajaran tidak terarah (*unsupervised learning*), pembelajaran semi terarah (*semi supervised learning*), dan *reinforcement learning* [8].

*Supervised learning* adalah teknik pembelajaran mesin dengan membuat suatu fungsi dari data latihan. Data latihan terdiri dari pasangan nilai masukan dan keluaran yang diharapkan dari masukan yang bersangkutan. Tugas dari *supervised learning* adalah untuk memprediksi nilai fungsi untuk nilai semua masukan yang ada.

Sebagai contoh, ada data luas rumah ( $x$ ) dan harga ( $y$ ), lalu dimasukkan dalam grafik  $x$  dan  $y$ -nya. Setelah itu, dibuat regresi antara  $x$  dan  $y$ -nya. Setelah membuat regresi, dapat diprediksi dari hasil regresi harga rumah dengan luas tertentu.

Teknik ini menggunakan prosedur yang berusaha untuk mencari partisi dari sebuah

pola. *Unsupervised learning* mempelajari cara sebuah sistem dapat belajar untuk merepresentasikan pola masukan dalam cara yang menggambarkan struktur statistik dari keseluruhan pola masukan. Berbeda dari *supervised learning*, *unsupervised learning* tidak memiliki target keluaran yang eksplisit atau tidak ada pengklasifikasian masukan.

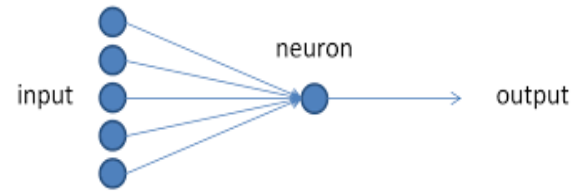
Dalam *machine learning*, teknik *unsupervised* sangat penting. Hal ini dikarenakan cara bekerjanya mirip dengan cara bekerja otak manusia. Dalam melakukan pembelajaran, tidak ada informasi dari contoh yang tersedia. Oleh karena itu, *unsupervised learning* menjadi esensial.

### B. Artificial Neural Network (ANN)

Secara umum, Neural Network (NN) atau Artificial Neural Network (ANN) adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan jaringan syaraf manusia. ANN ini merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Secara sederhana, ANN adalah sebuah alat pemodelan data statistik non-linear. ANN dapat digunakan untuk memodelkan hubungan yang kompleks antara masukan dan keluaran untuk menemukan pola-pola pada data.

Secara mendasar, sistem pembelajaran merupakan proses penambahan pengetahuan pada ANN yang sifatnya kontinuitas, sehingga pada saat digunakan, pengetahuan tersebut akan dieksploitasikan secara maksimal dalam mengenali suatu objek. *Neuron* adalah bagian dasar dari pemrosesan suatu Neural Network. Di bawah ini merupakan bentuk dasar dari suatu neuron.

Pada Gambar 1 di atas, bagian Input merupakan masukan yang digunakan baik saat pembelajaran maupun dalam mengenali suatu objek. Weight, beban yang selalu berubah setiap kali diberikan input sebagai proses pembelajaran. Processing Unit merupakan tempat berlangsungnya proses pengenalan suatu objek berdasarkan pembebanan yang diberikan. Output, keluaran dari hasil pengenalan suatu objek.



Gbr. 1 Bentuk dasar neuron [9]

### C. Deep Learning

*Deep learning* adalah salah satu algoritma pembelajaran mesin yang berbasis tidak terarah (*unsupervised learning*) dan terdiri dari banyak level dalam fitur dan ekstraksi data. Fitur yang lebih tinggi didapat dari fitur yang lebih rendah untuk membentuk sebuah representasi hierarki. Terdapat beberapa jenis arsitektur *deep learning*. Namun, yang akan dibahas di sini adalah arsitektur yang digunakan dalam sistem identifikasi objek pada penelitian ini.

#### 1) Deep Belief Network

*Deep Belief Network* adalah sebuah model yang generatif dan terdiri dari beberapa lapisan unit tersembunyi. DBN dapat dilihat sebagai sebuah modul *learning* sederhana yang dibentuk di setiap *layer*. DBN dapat digunakan untuk data pra-latih DNN secara generatif dengan menggunakan bobot belajar (*learned weight*) sebagai bobot awal (*initial weights*). *Backpropagation* atau algoritma diskriminatif lainnya kemudian dapat diaplikasikan untuk *fine tuning* bobot-bobot tersebut. Hal ini sangat membantu terutama ketika data latih yang tersedia sangat terbatas. Data pra-latih ini berada di area yang dekat dengan bobot optimal jika dibandingkan dengan inisialisasi acak. Hal ini membuat kapabilitas pemodelan meningkat dan konvergensi pada fase *fine tuning* menjadi lebih cepat.

Sebuah DBN dapat dilatih dalam *unsupervised* secara efektif, *layer per layer* yang mana setiap *layer*-nya biasanya dibentuk dari *Restricted Boltzman Machines* (RBM). RBM berbentuk graf tidak berarah (*undirected*), model berbasis energi generatif dengan *layer input* dan *layer tersembunyi* tunggal. Hubungan hanya ada di antara unit *layer input* yang terlihat dan unit yang

tersembunyi di *layer* tersembunyi. Tidak ada hubungan antara yang terlihat dengan yang terlihat (*visible-visible*) atau yang tersembunyi dengan yang tersembunyi (*hidden-hidden*).

RBM dapat ditumpuk (*stacked*) dan dilatih dengan pendekatan *greedy* untuk membentuk Deep Belief Networks (DBN). DBN adalah model grafis yang belajar untuk mengekstrak representasi hirarkis yang mendalam dari data latih. Model tersebut membentuk *joint distribution*  $x$  diantara *observed vector* dan  $l$  *hidden layer*  $h^k$  seperti ditunjukkan dalam persamaan:

$$P(x, h^1, \dots, h^l) = \left( \prod_{k=0}^{l-2} P(h^k | h^{k+1}) \right) P(h^{l-1}, h^l) \quad (1)$$

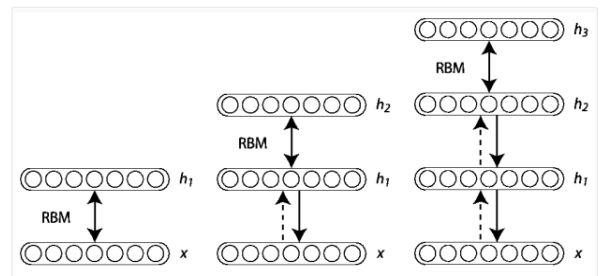
yang mana  $x = h^0$ ,  $P(h^{k-1} | h^k)$  adalah sebuah *conditional distribution* untuk unit terlihat yang dikondisikan dalam unit tersembunyi di RBM sebagai  $P(h^{l-1}, h^l)$  level  $k$ , dan sebagai *joint distribution visible-hidden* di level atas RBM.

Prinsip *greedy layer-wise unsupervised training* dapat diaplikasikan ke DBN dengan RBM sebagai *building block* di tiap *layer*-nya. Prosesnya sebagai berikut.

1. Latih *layer* pertama sebagai RBM yang memodelkan *raw input*  $x = h^0$  sebagai *layer* yang terlihat.
2. Gunakan *layer* pertama untuk mendapatkan representasi input yang akan digunakan sebagai data untuk *layer* kedua. Dua solusi umum tercipta. Representasi ini bisa dipilih sebagai *mean activation*  $p(h^{\{1\}} = 1 | h^{\{0\}})$  atau contoh dari  $p(h^{(1)} | h^{(0)})$ .
3. Latih *layer* kedua sebagai RBM, ambil data yang telah ditransformasi (sampel atau *mean activations*) sebagai contoh *training* (untuk *layer* RBM yang terlihat).
4. Ulangi langkah 2 dan 3 sesuai jumlah *layer* yang diperlukan, di setiap iterasi lakukan penyebaran ke atas (*propagating upward*) baik sampel maupun nilai rata-rata.
5. *Fine-tune* semua parameter dari arsitektur *deep* ini sehubungan dengan proxy untuk DBN-loglikelihood, atau sehubungan dengan kriteria pembelajaran terarah

(setelah menambahkan perlengkapan pembelajaran extra untuk mengonversi representasi yang telah dipelajari menjadi prediksi terarah, contohnya sebuah *classifier liner*).

Berikut adalah ilustrasi dalam melatih sebuah *Deep Belief Network* dengan membangun multi *Restricted Boltzmann Machines* ditumpuk di setiap atasnya. Setiap *layer* terdiri dari banyak *node* yang berkontribusi untuk *layer* selanjutnya.



Gbr. 2 Training Deep Belief Network [10]

Mengapa algoritma tersebut bekerja? Ambil contoh sebuah DBN yang memiliki 2 *layer* dengan *hidden layer*  $h^1$  dan  $h^2$  (dengan parameter *weight*  $W^{(1)}$  dan  $W^{(2)}$ , didirikan untuk derivasi rinci) yaitu  $\log p(x)$  yang dapat ditulis sebagai :

$$\begin{aligned} \log p(x) &= KL(Q(h^{(1)} | x) || p(h^{(1)} | x)) \\ &+ H_{Q(h^{(1)} | x)} \\ &+ \sum_h Q(h^{(1)} | x) (\log p(h^{(1)})) \\ &+ \log p(x | h^{(1)}) \end{aligned} \quad (2)$$

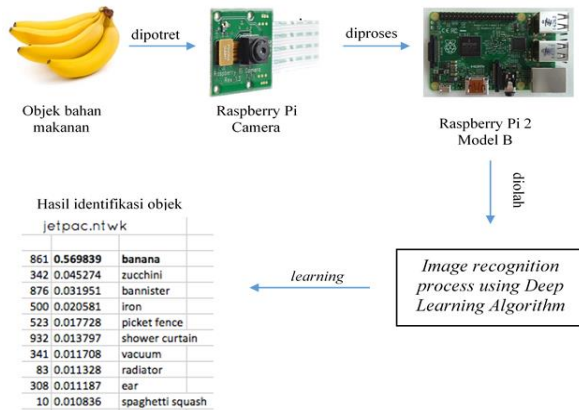
$$\begin{aligned} \log p(x) &= KL(Q(h^{(1)} | x) || p(h^{(1)} | x)) \\ &+ H_{Q(h^{(1)} | x)} \\ &+ \sum_h Q(h^{(1)} | x) (\log p(h^{(1)})) \\ &+ \log p(x | h^{(1)}) \end{aligned}$$

$KL(Q(h^{(1)} | x) || p(h^{(1)} | x))$  merepresentasikan perbedaan KL di antara posterior  $Q(h^{(1)} | x)$  dari RBM pertama jika berdiri sendiri, dan probabilitas  $p(h^{(1)} | x)$  untuk *layer* yang sama tetapi didefinisikan oleh keseluruhan DBN. Sebagai contoh, mempertimbangkan *prior*  $p(h^{(1)}, h^{(2)})$

didefinisikan oleh level atas RBM.  $H_{Q(h^{(1)}|x)}$  adalah entropi dari distribusi  $Q(h^{(1)}|x)$ .

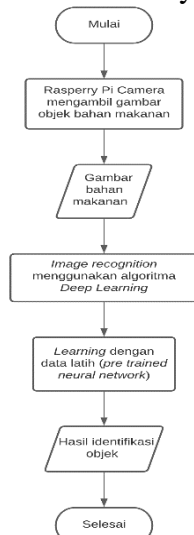
### III. METODE

Berikut gambaran umum sistem yang menunjukkan alur berjalannya sistem dari mulai *input*, proses, dan *output*.



Gbr. 3 Gambaran umum sistem

Dari gambar di atas, dapat dilihat bahwa Raspberry Pi Camera berperan sebagai perangkat *input* sistem yang dibuat. Kamera ini akan mengambil gambar yang ada di lemari pendingin yang merupakan objek yang akan diolah datanya. Kemudian, Raspberry Pi 2 akan mengolah gambar yang sudah diambil. Dilakukan proses *image recognition* dengan menggunakan algoritma *Deep Learning* untuk mengidentifikasi objek makanan. Setelah melalui proses *learning*, objek makanan akan teridentifikasi dengan tingkat kepercayaan (*confidence level*) tertentu. Berikut merupakan *flowchart* sistem yang dibuat.



Gbr. 4 Flowchart sistem

### IV. HASIL DAN PEMBAHASAN

Percobaan dilakukan dengan memasukkan Raspberry Pi, Raspberry Pi Camera, dan *power bank* sebagai catu daya, ke dalam lemari pendingin bersamaan dengan objek bahan makanan.



Gbr. 5 Implementasi sistem di dalam lemari pendingin



Gbr. 6 Set Raspberry Pi



Gbr. 7 Kamera Raspberry Pi

Kemudian dilakukan uji program pada 4 objek bahan makanan: pisang, mentimun, brokoli, dan jeruk. Pertama-tama, gambar objek bahan makanan diambil dengan menggunakan kamera Raspberry Pi. Kemudian, gambar tersebut diolah menggunakan algoritma *Deep Learning* dengan arsitektur Deep Belief Network. Terdapat 2 data latih (*pre trained neural networks*) yang digunakan, masing-masing bernama *jetpac.ntwk* dan *ccv2012.ntwk*.



Kedua data latih ini diadopsi dari ImageNet [11] yang terdiri dari 999 dan 1000 data.

Objek pertama yang diuji adalah pisang. Berikut merupakan gambar yang diambil dari kamera dan hasil identifikasi objek pisang. Pisang menempati urutan pertama objek dengan tingkat kepercayaan paling tinggi yang dikenali sebagai pisang. Dengan menggunakan data latih *jetpac.ntwk* didapatkan *confidence level* pisang sebesar 56,98%; sedangkan *ccv2012.ntwk* sebesar 36,88%.

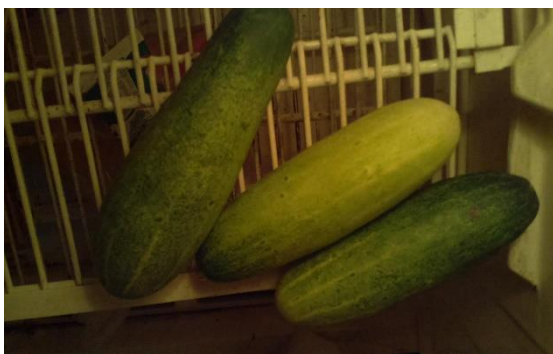


Gbr. 8 Hasil potret objek pisang

jetpac.ntwk		ccv2012.ntwk	
861	0.569839 banana	322	0.368846 banana
342	0.045274 zucchini	738	0.081959 zucchini
876	0.031951 bannister	570	0.053678 radiator
500	0.020581 iron	908	0.042067 bath towel
523	0.017728 picket fence	746	0.029723 shower curtain
932	0.013797 shower curtain	827	0.028204 swab
341	0.011708 vacuum	836	0.023038 sweatshirt
83	0.011328 radiator	793	0.013164 suit
308	0.011187 ear	717	0.010681 bannister
10	0.010836 spaghetti squash	835	0.010132 cardigan

Gbr. 9 Hasil identifikasi objek pisang

Berikut ini hasil pengujian program pada mentimun sebagai objek kedua. Dapat dilihat bahwa objek diidentifikasi paling tinggi sebagai zukini, padahal seharusnya mentimun. Zukini memang memiliki bentuk seperti mentimun. Mentimun memiliki *confidence level* tertinggi kedua setelah Zukini di data latih *ccv2012.ntwk* dengan 5,87%.

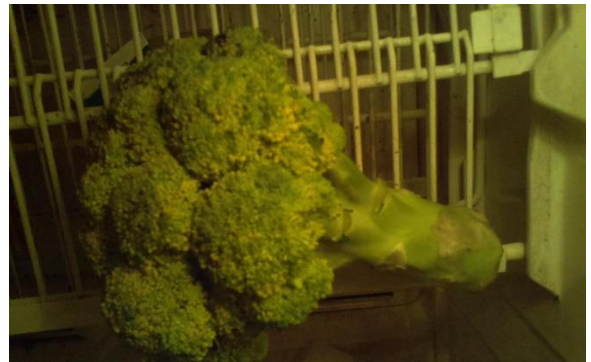


Gbr. 10 Hasil potret objek mentimun

jetpac.ntwk		ccv2012.ntwk	
342	0.098825 zucchini	738	0.786118 zucchini
787	0.073409 green mamba	742	0.058774 cucumber
845	0.059466 jackfruit	739	0.044942 spaghetti squash
523	0.054030 picket fence	740	0.019571 acorn squash
861	0.039254 banana	322	0.012329 banana
585	0.035547 artichoke	317	0.009287 Granny Smith
10	0.035221 spaghetti squash	741	0.008309 butternut squash
308	0.032584 ear	774	0.004652 spindle
860	0.032338 head cabbage	743	0.004245 artichoke
570	0.031100 cucumber	827	0.003409 swab

Gbr. 11 Hasil identifikasi objek mentimun

Objek ketiga yang diuji adalah brokoli. Pada pengujian ini, hasil identifikasi terbilang jauh dari yang sebenarnya. Objek diyakini dikenali sebagai tirai kamar mandi. Brokoli hanya memiliki *confidence level* sebesar 0,44% dan 0,96% di data latih *jetpac.ntwk* dan *ccv2012.ntwk*.



Gbr. 12 Hasil potret objek brokoli

jetpac.ntwk		ccv2012.ntwk	
932	0.089147 shower curtain	746	0.328960 shower curtain
585	0.077925 artichoke	721	0.104471 picket fence
394	0.066606 greenhouse	570	0.027328 radiator
533	0.064439 picket fence	847	0.023815 groom
874	0.049406 tennis ball	358	0.016378 cliff
836	0.027558 pot	870	0.014442 mitten
338	0.022567 worm fence	903	0.013792 window shade
90	0.022489 book jacket	737	0.011888 cauliflower
860	0.019243 head cabbage	717	0.011375 bannister
967	0.016267 cauliflower	470	0.010971 green lizard
861	0.015607 banana	701	0.009866 prison
171	0.015273 pitcher	885	0.009587 coral fungus
123	0.013984 wine bottle	736	0.009564 broccoli
779	0.012979 prison	854	0.008810 poncho
275	0.012698 corn	827	0.008614 swab
553	0.012501 custard apple	789	0.008594 vestment
869	0.012372 barrel	911	0.008509 window screen
286	0.012110 chime	719	0.008366 dam
731	0.011605 guacamole	738	0.008098 zucchini
231	0.011253 rain barrel	910	0.007000 gown
328	0.010691 acorn squash	723	0.006890 stone wall
787	0.009906 green mamba	997	0.006783 stole
44	0.008863 broom	711	0.006233 fountain
941	0.008692 castle	329	0.006209 rapeseed
83	0.008289 radiator	850	0.006162 broom
965	0.006924 golf ball	921	0.006009 maze
395	0.006883 shoji	70	0.005712 West Highland white terrier
958	0.006792 lakeside	499	0.005590 tree frog
54	0.006527 megalith	851	0.005306 mosquito net
308	0.006160 ear	323	0.005248 jackfruit
426	0.006048 window shade	795	0.004647 feather boa
88	0.005885 Granny Smith	722	0.004619 worm fence
277	0.005772 beer glass	683	0.004328 greenhouse
18	0.005734 lampshade	599	0.003846 spider web
647	0.005603 groom	716	0.003456 megalith
342	0.005445 zucchini	888	0.003352 toilet tissue
572	0.005225 swing	725	0.003319 sliding door
503	0.005161 window screen	720	0.003316 chainlink fence
3	0.005021 wool	873	0.003292 vase
570	0.004837 cucumber	700	0.003277 castle
729	0.004455 broccoli	813	0.003251 lampshade

Gbr. 13 Hasil identifikasi objek brokoli

Pengujian terakhir dilakukan pada jeruk. Jeruk dapat diidentifikasi dengan baik di data latih *ccv2012.ntwk*. Hal ini ditunjukkan dengan *confidence level* sebesar 45,88%.



Gbr. 14 Hasil potret objek jeruk.

jetpac.ntwk			ccv2012.ntwk		
565	0.653928	butternut squash	318	0.458803	orange
739	0.072812	jack-o'-lantern	319	0.148665	lemon
903	0.063168	bagel	739	0.124909	spaghetti squash
183	0.042747	guinea pig	741	0.122865	butternut squash
701	0.011112	vizsla	323	0.023338	jackfruit
10	0.010669	spaghetti squash	325	0.021703	pomegranate
942	0.010391	golden retriever	950	0.018437	wooden spoon
826	0.008826	Rhodesian ridgeback	767	0.011185	bagel
448	0.008476	mushroom	337	0.008512	maraca
870	0.006811	basenji	336	0.006338	gong
328	0.006770	acorn squash	322	0.004973	banana
284	0.005848	boxer	840	0.004224	ping-pong ball
427	0.005242	redbone	320	0.003935	fig
221	0.005105	Pekinese	317	0.003532	Granny Smith
316	0.004529	French loaf	740	0.003036	acorn squash
788	0.004486	Angora	745	0.002872	mushroom
531	0.003995	proboscis monkey	326	0.002040	acorn
624	0.003921	ocarina	341	0.001957	cello
365	0.003890	hamster	785	0.001685	teddy
518	0.002556	dumbbell	340	0.001510	banjo
806	0.002168	pretzel	755	0.001381	croquet ball
550	0.002133	Band Aid	863	0.001355	dough
845	0.001993	jackfruit	452	0.001305	anemone fish
257	0.001961	standard poodle	156	0.001250	hamster
874	0.001937	tennis ball	513	0.001224	strainer
331	0.001836	Great Dane	702	0.001222	grocery store
476	0.001819	wig	872	0.001175	French loaf
121	0.001704	dough	980	0.001147	bolete
747	0.001630	bolete	342	0.000679	violin
247	0.001436	bath towel	999	0.000671	dumbbell
170	0.001398	ice lolly	675	0.000523	spatula
147	0.001335	clog	100	0.000466	guinea pig
832	0.001272	piggy bank	546	0.000425	abacus
977	0.001107	lemon	798	0.000417	drumstick
553	0.001095	custard apple	978	0.000397	clog
989	0.001051	shower cap	232	0.000390	balloon
864	0.000903	Saint Bernard	670	0.000382	frying pan
324	0.000848	conch	891	0.000370	ladle
162	0.000750	hog	335	0.000367	drum
326	0.000715	corgi	334	0.000328	chime
654	0.000710	wooden spoon	671	0.000318	wok
230	0.000690	hen-of-the-woods	845	0.000318	punching bag
893	0.000685	dingo	520	0.000317	scale
141	0.000667	earthstar	591	0.000308	jack-o'-lantern
178	0.000644	orange	897	0.000296	crate

Gbr. 15 Hasil identifikasi objek jeruk

## V. KESIMPULAN

Dari hasil pengujian, terlihat bahwa program mengidentifikasi objek dengan benar pada objek pisang dan jeruk yang ditunjukkan dengan *confidence level* tertinggi sebesar 56,98% dan 45,88%. Identifikasi objek mentimun dikenali sebagai zukini dengan *confidence level* tertinggi sebesar 78,61%. Adapun identifikasi objek paling rendah terdapat pada objek brokoli dengan *confidence level* kurang dari 1%.

## REFERENSI

- [1] D. A. Muktiawan, "Rancang Bangun Monitoring Alat Penyimpanan Kebutuhan Pokok Melalui Android Berbasis Mikrokontroler," in *Prosiding SNRT (Seminar Nasional Riset Terapan)*, 2016, vol. 5662, pp. 9–10, doi: ISSN 2541-5670 (Online).
- [2] T. Meilany Siregar, L. A. Harahap, and A. Rohanah, "IDENTIFIKASI KEMATANGAN BUAH PISANG (*Musa paradisiaca*) DENGAN TEKNIK JARINGAN SYARAF TIRUAN (Identification of Banana Maturity (*Musa paradisiaca*) with Artificial Neural Network)," *J. Rekayasa Pangan dan Pertan.*, vol. 3, no. 2, pp. 261–265, Aug. 2015, Accessed: May 20, 2021. [Online]. Available: <http://download.garuda.ristekdikti.go.id/article.php?article=1434485&val=4140&title=IDENTIFICATION OF BANANA MATURITY MUSA PARADISIACA WITH ARTIFICIAL NEURAL NETWORK>.
- [3] M. Michael, F. Tanoto, E. Wibowo, F. Lutan, and A. Dharma, "Pengenalan Plat Kendaraan Bermotor dengan Menggunakan Metode Template Matching dan Deep Belief Network," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 1, pp. 27–36, 2019, doi: 10.30812/matrik.v19i1.475.
- [4] N. Anggraini, D. F. Rahman, L. K. Wardhani, and N. Hakiem, "Mobile-based monitoring system for an automatic cat feeder using Raspberry Pi," *Telkomnika (Telecommunication Comput. Electron. Control.)*, vol. 18, no. 2, pp. 1038–1046, Apr. 2020, doi: 10.12928/TELKOMNIKA.V18I2.14819.
- [5] F. Humani, K. Adi, and C. E. Widodo, "Aplikasi Pengolahan Citra Pada Raspberry Pi Untuk Membedakan Benda Berdasarkan Warna Dan Bentuk," *Youngster Phys. J.*, vol. 5, no. 4, pp. 157–162, Oct. 2016, Accessed: May 20, 2021. [Online]. Available: <https://ejournal3.undip.ac.id/index.php/bfd/article/view/14055>.
- [6] R. Arif Candra, D. Nur Ilham, E. Sipahutar, D. Politeknik Aceh Selatan, and D. Politeknik ATI Padang, "METHOMIKA: Jurnal Manajemen Informatika & Komputerisasi Akuntansi PERANCANGAN SMART SECURITY CAMERA DENGAN MODEL IMAGE

- PROCESSING MENGGUNAKAN RASPBERRY PI,” *METHOMIKA J. Manaj. Inform. Komputerisasi Akunt.*, vol. 3, no. 2, pp. 105–111, 2019.
- [7] J. Sapes and F. Solsona, “Fingerscanner: Embedding a fingerprint scanner in a raspberry pi,” *Sensors (Switzerland)*, vol. 16, no. 2, Feb. 2016, doi: 10.3390/s16020220.
- [8] J. W. G. Putra, “Pengenalan Konsep Pembelajaran Mesin dan Deep Learning,” *Comput. Linguist. Nat. Lang. Process. Lab.*, vol. 4, pp. 1–235, 2019, [Online]. Available: <https://www.researchgate.net/publication/323700644>.
- [9] G. Ciaburro, *MATLAB for Machine Learning*, vol. 1, no. 1. Packt, 2018.
- [10] C. C. Aggarwal, “Restricted Boltzmann Machines,” *Neural Networks and Deep Learning*, pp. 235–270, 2018.
- [11] S. V. Lab, S. University, and P. University, “ImageNet,” <https://image-net.org/index.php>, 2020. <https://image-net.org/index.php> (diakses 27 Maret, 2021).
- [12] D. Krisrenanto, M. Rivai, and F. Budiman, “Identifikasi Jumlah dan Tingkat Aktivitas Orang Berbasis Pengolahan Citra Menggunakan Raspberry Pi,” *J. Tek. ITS*, vol. 6, no. 1, p. A-110-A-115, 2017, doi: 10.12962/j23373539.v6i1.21397.
- [13] L. Liu *et al.*, “Deep Learning for Generic Object Detection: A Survey,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.
- [14] B. Petrovska, I. Stojanovic, and T. Atanasova-Pacemska, “Classification of Small Sets of Images with Pre-trained Neural Networks,” *Int. J. Eng. Manuf.*, vol. 8, no. 4, pp. 40–55, Jul. 2018, doi: 10.5815/ijem.2018.04.05.
- [15] Y. Rizk, N. Hajj, N. Mitri, and M. Awad, “Deep belief networks and cortical algorithms: A comparative study for supervised classification,” *Appl. Comput. Informatics*, vol. 15, no. 2, pp. 81–93, Jul. 2019, doi: 10.1016/j.aci.2018.01.004.