

Monitoring Suhu *Chiller* Berbasis *Internet of Things* Menggunakan *Thermal Camera*

Nurul Akbar Arlan, Muhammad Rivai, dan Harris Pirngadi
Departemen Teknik Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: muhammad_rivai@ee.its.ac.id

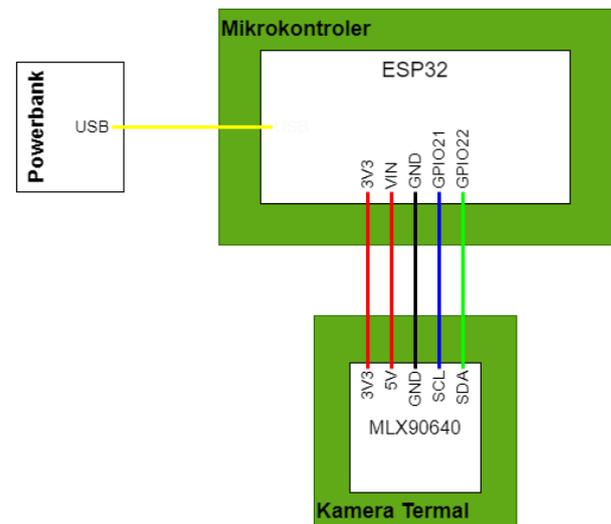
Abstrak—*Cold chain* dideskripsikan sebagai alat dan proses yang melakukan suatu cara untuk memastikan produk terjaga suhunya hingga sampai ke konsumen. Produk ini bersifat mudah busuk, rentan perubahan suhu, dan memiliki jangka waktu yang pendek. *Cold chain logistic* merupakan gabungan dari empat sistem yang bekerja sama yaitu *precooling*, gudang pendingin, transportasi dengan pendinginan, dan pemasaran. Kelemahan sistem pemantauan yang digunakan sekarang ialah menggunakan sensor yang tidak merepresentasikan suhu satu ruangan. Pada penelitian ini melibatkan kamera termal untuk mendeteksi suhu dari produk-produk yang diletakkan dalam satu ruangan *chiller*. Sistem ini menggunakan *computer vision* untuk menampilkan gambar termal yang ditangkap oleh kamera termal, *object detection* dengan model YOLOv5 untuk mendeteksi objek yang akan dipantau, dan *Internet of Things* untuk pengiriman datanya. Alat ini dibuat menggunakan modul kamera termal MLX90640 yang terhubung dengan mikrokontroler ESP32 dan dikirim ke komputer untuk dapat mendeteksi objek dan pemantauan suhu dengan sistem peringatan dini. Hasil percobaan menunjukkan bahwa kamera termal dapat mengukur suhu ruangan *chiller* dengan persamaan polinomial orde dua setelah dilakukan proses kalibrasi. Tahap pelatihan model YOLOv5 menghasilkan nilai *mean average precision* (mAP) terbaik sebesar 0.7, *object loss* terkecil sebesar 0.013, dan *box loss* terkecil sebesar 0.016 dengan *precision* sebesar 0.976 dan *recall* sebesar 0.965. Sistem peringatan dini berupa alarm yang aktif ketika suhu objek atau ruangan melebihi batas yang telah diatur. Data-data suhu dikirimkan via *Internet of Things* (IoT) ke *server Thingspeak* untuk ditampilkan pada komputer.

Kata Kunci—*Chiller*, *Computer Vision*, *Internet of Things*, Kamera Termal.

I. PENDAHULUAN

PASAR *cold chain* merupakan sebuah kombinasi konsep dari rantai pasok makanan dan pendistribusian makanan dengan suhu yang terkontrol. kebanyakan produk yang dikategorikan di dalam pasar *cold chain* adalah produk yang bersifat mudah busuk, rentan terhadap suhu, dan memiliki jangka waktu yang pendek sehingga membutuhkan perlakuan khusus untuk mempertahankannya agar tetap segar hingga sampai ke konsumen. Permintaan *frozen food* pada tahun-tahun ini terus meningkat dengan rata-rata 6.9% per tahun. Pada 2014 konsumsi *frozen food* hanya mencapai sekitar 5 juta ton dan di tahun selanjutnya naik hingga 5.5 juta ton hingga tahun 2018. Tercatat konsumsi *frozen food* di Indonesia mencapai 6.6 juta ton yang memperlihatkan pertumbuhan terus meningkat [1].

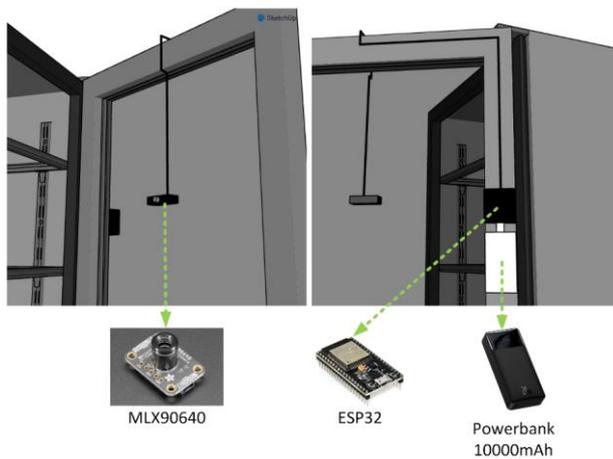
Proses logistik pada *cold chain* meliputi empat sistem yakni *precooling*, gudang pendingin, transportasi dengan pendinginan, dan pemasaran untuk pendistribusian produk. Secara umum waktu membusuk akan berlipat ganda setiap kenaikan 10°C bila produk ini memiliki umur relatif 100% pada 0°C dengan kerugian 1%; pada 10°C umur relatif menurun menjadi 33% dan kerugian meningkat 3%; pada



Gambar 1. Rangkaian seluruh sistem monitoring suhu *chiller*.

20°C umur relatif menurun menjadi 13% dan kerugian meningkat 8%; pada 30°C umur relatif menurun menjadi 7% dan kerugian meningkat 14%; dan pada 40°C umur relatif menurun menjadi 4% dan kerugian meningkat 25% [2–4]. Untuk produk susu, suhu sangat perlu dijaga pada 4°C untuk memastikan kualitas dari produknya dan mencegah adanya pertumbuhan mikro bakteri yang dapat membuatnya membusuk [5]. Untuk vaksin pada *chiller* dapat disimpan pada suhu 2-8°C agar dapat bertahan selama beberapa hari. Bila suhu vaksin tidak di dalam jangkauan suhu yang seharusnya maka vaksin harus dibuang [6]. Karena itu diperlukan keterampilan manajemen *cold chain* untuk menghindari kerugian besar akibat kerusakan makanan dan menjaga kualitas produk yang dibutuhkan.

Pada penelitian ini telah membuat sebuah sistem yang dapat melakukan pemantauan suhu dari produk-produk *cold chain* pada proses logistik. Studi ini berfokus membuat alat *monitoring* produk *cold chain* dengan menggunakan kamera termal untuk mendeteksi suhu dari produk-produk *cold chain* antara lain susu di dalam botol minuman dan obat-obatan di dalam vial. Kamera termal digunakan karena dapat mendeteksi suhu di banyak titik sehingga lebih merepresentasikan satu ruangan *chiller* dibandingkan sensor *thermocouple* yang hanya dapat mendeteksi satu titik saja [7]. Kamera termal dapat mendeteksi suhu dari jarak yang lebih jauh dan dapat menunjukkan distribusi suhu dari tiap objeknya. Gambar yang diperoleh kamera termal diproses oleh mikrokontroler ESP32, lalu data dikirim ke komputer untuk pendeteksian objek di dalam ruangan *chiller* menggunakan metode *You Only Look Once* (YOLO). Objek yang dideteksi akan diukur suhunya dan diperiksa apakah sesuai dengan batas suhu yang diizinkan. Sistem akan memberi peringatan dini bila terdapat produk yang suhunya



Gambar 2. Ilustrasi perancangan keseluruhan sistem.

tidak sesuai.

II. METODE

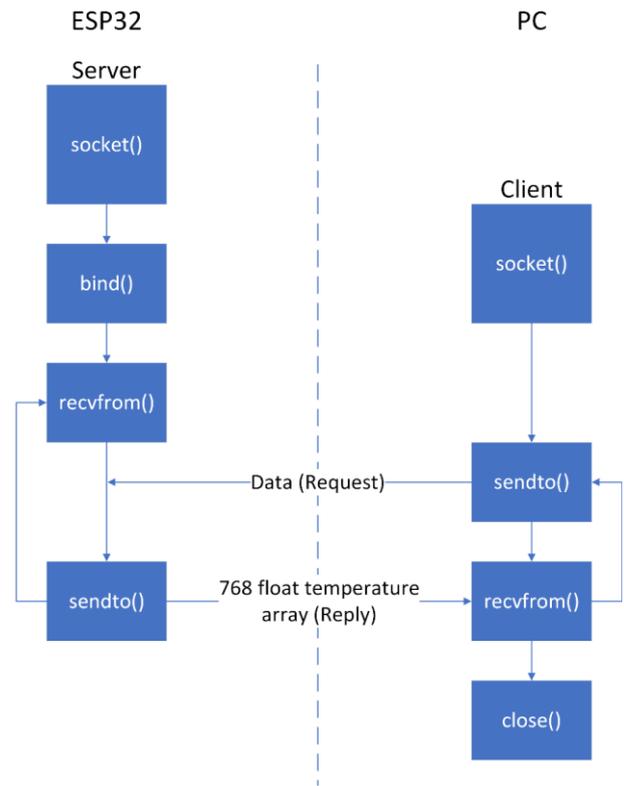
A. Perancangan Seluruh Sistem

Sistem yang digunakan pada penelitian ini dibuat dengan kamera termal MLX90640 diletakkan di dalam ruangan *chiller* dihadapkan ke objek yang akan dideteksi dan dihubungkan menggunakan kabel dengan desain tahan air ke mikrokontroler ESP32 yang diletakkan di luar ruangan kulkas. Gambar 1 menunjukkan sistem yang terdiri dari kamera termal yang terhubung dengan mikrokontroler ESP32 dengan komunikasi I2C dengan pin SDA terhubung dengan pin GPIO21 dan pin SCL terhubung dengan pin GPIO22. Komponen kapasitor elektrolit diperlukan agar mikrokontroler ESP32 dapat di-*upload* secara benar. Kemudian kamera termal dimasukkan ke dalam sebuah wadah yang ditutup dengan lem agar air tidak dapat masuk ke dalamnya. Kamera termal MLX90640 dihubungkan ke mikrokontroler ESP32 melalui kabel yang dilapisi dengan *heatsink* agar dapat dimasukkan ke dalam *chiller*. Kamera termal ditempel di dalam *chiller* menghadap ke objek *cold chain* dan mikrokontroler ESP32 diletakkan di luar ruangan *chiller* dengan *powerbank* sebagai sumber tegangan. Rancangan ini diimplementasikan seperti yang diilustrasikan pada Gambar 2.

B. Pengujian dan Kalibrasi Modul Kamera Termal

Kamera termal adalah perangkat yang dapat mengukur suhu objek dan menampilkannya sebagai gambar visual. Ini berguna untuk menemukan kebocoran panas, dan mengidentifikasi area yang berpotensi merusak karena panas yang berlebih. Kamera termal beroperasi dengan cara memindai lingkungan dan mengukur panas yang dipancarkan oleh objek yang ada di sekitarnya. Kamera termal menangkap radiasi inframerah yang mewakili radiasi suhu. Radiasi yang dipancarkan benda diubah menjadi sinyal elektronik untuk menghasilkan gambar yang dapat dilihat oleh mata manusia yang disebut juga sebagai termogram [8–10]. Lensa yang digunakan pada kamera termal adalah sebuah material khusus yaitu Germanium, kristal safir, atau bahan yang lebih ekonomis seperti *Chalcogenide*.

Pengujian dilakukan dengan mencoba kamera termal dengan mikrokontroler yang kemudian divisualisasikan hasil



Gambar 3. Datagram perpindahan data dari ESP32 ke PC.

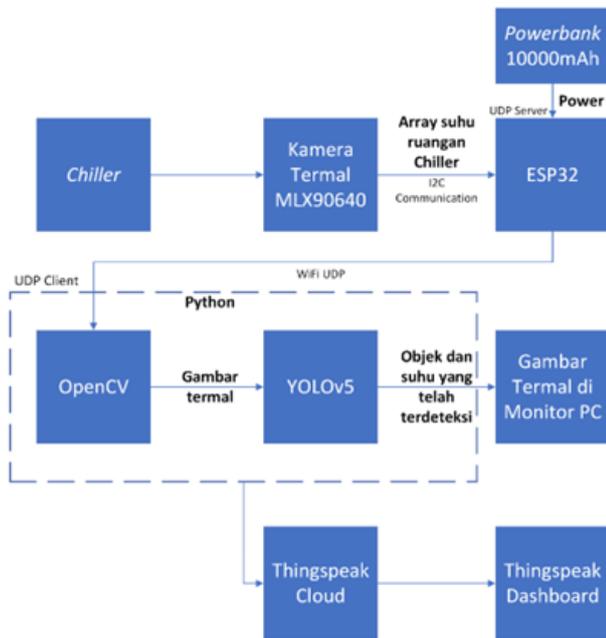
suhu dengan aplikasi Processing IDE. Kemudian kalibrasi dilakukan dengan meletakkan kamera termal diarahkan menghadap objek yang memiliki suhu yang rata. Pada proses kalibrasi ini menggunakan air yang diberikan es batu. Sensor suhu DS18B20 digunakan untuk tujuan ini. Proses kalibrasi ini dilakukan untuk memperoleh ketepatan pembacaan suhu yang dihasilkan oleh kamera termal. Kamera termal MLX90640 akan dioperasikan dengan mikrokontroler ESP32, sedangkan sensor DS18B20 akan dioperasikan terpisah dengan mikrokontroler Arduino Nano.

Proses kalibrasi kamera termal dilakukan terhadap perubahan suhu dan variasi jarak dari air ke kamera. Secara detail kalibrasi ini dilakukan dengan menghadapkan kamera termal ke air yang dicampur dengan es batu sejauh 10 hingga 40 cm. Kemudian suhu dinaikkan setiap 5°C. Data suhu sensor DS18B20 diplot pada sumbu-y dan data suhu kamera termal MLX90640 diplot pada sumbu-x. Persamaan kalibrasi didapatkan dengan menggambar garis *trendline* pada grafik tersebut.

C. Perancangan dan Pengujian Neural Network

Pada penelitian ini digunakan *machine learning* yang mempelajari suatu data sehingga dapat membuat keluaran sebuah pemahaman dari data yang dimasukkan [11]. Untuk mendapatkan informasi dari gambar maka memerlukan metode *computer vision* yang merupakan kombinasi dari *image processing* dan *pattern recognition*. Tujuan utama dari proses *computer vision* adalah untuk mengekstraksi data dari gambar, membuat model, dan informasi dari gambar [12]. Untuk mengenali pola dari citra biasanya digunakan *Convolutional Neural Network (CNN)* yang merupakan salah satu konsep dari *deep learning* [13].

YOLO adalah sebuah algoritme yang dikembangkan oleh Ultralytics yang berfungsi untuk melakukan *object detection*



Gambar 4. Diagram blok keseluruhan system.

secara *real-time*. YOLO juga dapat dilatih dengan menggunakan *custom dataset*. Akan tetapi model ini memiliki keterbatasan dalam mendeteksi objek yang kecil, memiliki struktur yang sangat besar, dan membutuhkan komputasi yang sangat kompleks [14]. Pada penelitian ini digunakan YOLO versi 5 atau YOLOv5. Algoritme ini menggunakan struktur CSP-Darknet53 yang terbaru sebagai *backbone*. Lalu pada *neck* menggunakan *Path Aggregation Network* (CSP-PAN) dengan *Spatial Pyramid Pooling Fast* (SPPF). Model ini menggunakan *head* yang sama yang digunakan oleh YOLOv3. Bagian *head* menghasilkan tiga output *feature map* untuk mencapai *multi scale prediction* [15].

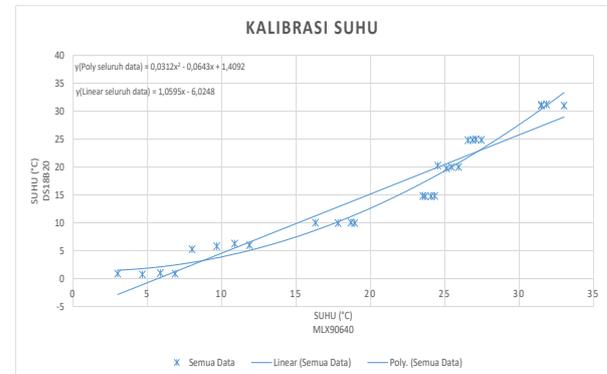
Untuk mendapatkan model yang sesuai dengan dataset maka dilakukan pelatihan dengan melakukan pengumpulan *custom dataset*, Anotasi *dataset*, dan *transfer learning* YOLO. Pengumpulan *custom dataset* dilakukan dengan cara mengumpulkan dataset yang berupa gambar dari ruangan *chiller* yang berisi botol kemasan yang berisi air dan susu dengan beberapa variasi ukuran. *Dataset* ini memiliki 2 kelas yaitu botol dan vial. Kemudian dari dataset yang telah dikumpulkan, dilakukan proses anotasi dan pemberian label pada objek kelas yang akan dideteksi pada gambar dataset. Setelah pemrosesan anotasi data kemudian dilakukan proses augmentasi data. Augmentasi data adalah sebuah proses manipulatif untuk mengubah bentuk dan posisi citra aslinya. Tujuannya adalah untuk menghasilkan variasi gambar yang lebih beragam sehingga komputer dapat belajar dan mengenali citra dari berbagai kondisi yang berbeda-beda. *Transfer learning* dilakukan pada *pre-trained model* dengan *hyperparameter Batch Size* sebesar 16, *Epoch* sebanyak 500, dan *Network size* sebesar 640. Pengujian ini dilakukan dengan membandingkan model YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, dan YOLOv5x agar dapat dilihat perbandingan dari ke 5 model ini sehingga dapat dipilih model yang terbaik.

D. Perancangan dan Pengujian Internet of Things

Internet of Things (IoT) dideskripsikan sebagai benda-benda yang saling terhubung melalui suatu jaringan secara

Tabel 1.
Perbandingan *error* persamaan kalibrasi linear dan polinomial.

Jarak (cm)	MAPE (%)	
	Kalibrasi Linear	Kalibrasi Polinomial
10	76,6	24,6
20	50,9	29,8
30	26,8	26,1
40	21,7	34,9
MAPE Seluruh Jarak (%)	44,0	28,9



Gambar 5. Grafik *trendline* linear dan polinomial orde 2 pada seluruh data kalibrasi.

nirkabel. IoT berinteraksi tanpa adanya campur tangan manusia. IoT memiliki arsitektur unik yang dapat dibagi menjadi tiga lapisan operasi. Setiap lapisan memiliki fungsi yang berbeda dan harus terhubung untuk bekerja. Lapisan *perception* digunakan untuk mengumpulkan data. Lapisan *application* tidak memiliki standar struktural umum untuk aplikasinya. Berbagai jenis program memiliki banyak batasan pada level ini. Lapisan *network* mengacu pada transfer data, dimana IoT berfungsi seperti lapisan jaringan TCP/IP [16].

Sistem IoT pada alat ini digunakan untuk mengirimkan *array* suhu dari mikrokontroler ESP32 ke komputer. Pada tahap ini dilakukan perancangan dan uji IoT dengan jaringan *Wireless Fidelity* (WiFi) melalui soket *User Datagram Protocol* (UDP). UDP adalah adalah lapisan jalur protokol di dalam struktur sistem TCP/IP. UDP biasa digunakan untuk mengirim pesan ke *hosts* di dalam jaringan *IP address*. Pesan dikirim dalam bentuk *datagrams* di dalam paket [16]. UDP memiliki latensi yang lebih rendah karena tidak memerlukan proses *acknowledgments*. Biasanya UDP digunakan untuk pengiriman data yang dapat menolerir *data loss*. Karena itu UDP sangat direkomendasikan untuk penggunaan *video streaming* tetapi memerlukan mekanisme tambahan agar lebih dapat diandalkan dan menangani masalah *congestion control* [17].

Pada penelitian ini mikrokontroler ESP32 akan mengirim data *array* dari kamera termal ke komputer melalui jaringan WiFi dengan port tertentu, lalu komputer akan difungsikan sebagai *client* untuk menerima data kamera. Komputer dan modul yang digunakan harus terhubung dengan jaringan WiFi yang sama. Pengujian IoT dilakukan dengan mencoba mengirim *request* suatu pesan dari komputer ke mikrokontroler ESP32, kemudian mikrokontroler mengirim data suatu pesan ke komputer, dan komputer akan menampilkan data yang diterima. Kemudian dilakukan juga pengujian pengiriman data *array* suhu dari kamera termal melalui jaringan WiFi dengan UDP ke komputer dari mikrokontroler ESP32. Jadi komputer akan mengirim *request* suatu pesan ke mikrokontroler dan setelah itu mikrokontroler

Tabel 2.
Perbandingan *metrics* hasil *training* YOLOv5.

Dataset	Model	Epoch	Metrics			
			Precision	Recall	mAP50	mAP50:95
Tanpa Interpolasi	YOLOv5n	251	0,97777	0,97267	0,98015	0,68806
	YOLOv5s	233	0,9811	0,96284	0,97585	0,68902
	YOLOv5m	185	0,97802	0,96342	0,97566	0,68356
	YOLOv5l	163	0,9741	0,96462	0,97538	0,69018
	YOLOv5x	171	0,97742	0,96907	0,96766	0,68878
Dengan Interpolasi	YOLOv5n	374	0,96463	0,94525	0,96812	0,68275
	YOLOv5s	226	0,98004	0,96267	0,97772	0,69669
	YOLOv5m	226	0,9658	0,96624	0,98032	0,69728
	YOLOv5l	192	0,98012	0,96117	0,98272	0,68917
	YOLOv5x	204	0,97603	0,96536	0,976	0,70017

akan mengirimkan data *array* suhu kamera termal dalam bentuk *byte*. Data *array byte* yang diterima oleh komputer dikonversi menjadi *float*. Rancangan transfer data dari mikrokontroler ESP32 ke komputer melalui protokol UDP digambarkan pada datagram Gambar 3.

E. Pengujian Seluruh Sistem

Pengujian seluruh sistem merupakan penggabungan modul yang telah diuji dan diintegrasikan menjadi satu sistem. Kamera termal akan menangkap suhu dari objek dan ruangan di dalam kulkas. Setelah itu data *array* suhu ruangan kulkas dikirim ke komputer melalui jaringan WiFi melalui protokol UDP dengan sistem IoT yang telah diuji sebelumnya. Kemudian data *array* divisualisasikan dan dilakukan *inference* dengan model YOLOv5 untuk mendeteksi objek. Kemudian suhu objek diukur dengan menghitung rata-rata suhu yang ditangkap oleh kamera termal di dalam *bounding box* dari hasil *inference*. Kemudian semua sistem yang telah uji diimplementasikan pada *chiller* Sanyo SR-F5.

Pada sistem ini juga dibuat sebuah *dashboard* Thingspeak agar sistem dapat dipantau dari luar jaringan WiFi yang terkoneksi pada sistem. Pada *dashboard* ini akan ditampilkan informasi suhu dari *chiller* dan objek. Blok diagram dari sistem ini ditunjukkan pada Gambar 4.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian dan Kalibrasi Modul Kamera Termal

Proses kalibrasi kamera termal dengan membandingkan suhu kamera termal MLX90640 dengan sensor DS18B20 yang dihubungkan dengan mikrokontroler Arduino Nano diarahkan ke air yang diberikan es di dalam ember. Perbandingan dilakukan dengan memvariasikan jarak antara kamera termal dengan permukaan air dan mengambil data setiap perubahan suhu pada sensor DS18B20. Kemudian dilakukan kalibrasi dengan memvariasikan jarak dengan pencatatan setiap perubahan suhu agar didapat persamaan kalibrasi nilai suhu kamera termal dengan sensor DS18B20. Pengujian ini dilakukan pada jarak 10 hingga 40 cm.

Data-data yang didapatkan diplot dalam sebuah grafik pada Gambar 5 untuk digambarkan sebuah *trendline* linear dan polinomial orde 2. Dari hasil *trendline* didapat dua persamaan linear (1) dan polinomial orde dua (2), yang dinyatakan sebagai:

$$y_1 = 1,0595x - 6,0248 \quad (1)$$

$$y_2 = 0,0312x^2 - 0,0643x + 1,4092 \quad (2)$$

Kemudian kedua persamaan ini diuji dengan suhu kamera

termal dan dibandingkan nilai rata-rata *error*-nya dengan menggunakan persamaan *Mean Absolute Percentage Error* (MAPE) (3).

$$MAPE = 100\% \times \frac{1}{n} \sum_{i=1}^n |y_i - x_i| \quad (3)$$

Dari Tabel 1 diperoleh rata-rata *error* terkecil adalah hasil dari substitusi suhu kamera termal MLX90640 pada persamaan kalibrasi polinomial orde dua. Sehingga persamaan ini mewakili karakteristik suhu dari kamera termal.

B. Hasil Perancangan dan Pengujian Neural Network

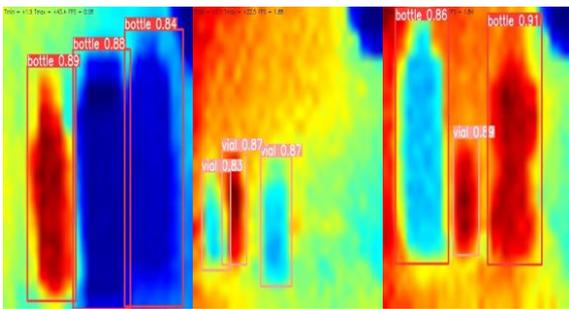
Dataset diambil dengan menggunakan kamera termal MLX90640 yang kemudian divisualisasikan dengan OpenCV di Python menjadi citra dengan *colormap JET*. Total dataset yang digunakan ialah 1013 gambar dengan persebaran 70% untuk data *training*, 15% untuk data *validate*, dan 15% untuk data *test*. Kemudian dari hasil pelatihan yang dilakukan dengan model yang berbeda didapatkan hasil *metrics* dari model *Transfer Learning*.

Berdasarkan Tabel 2, model YOLOv5x yang dilatih dengan dataset interpolasi memiliki *mean average precision* dengan *Intersect Over Union* 50-95% (mAP50:95) terbesar bernilai 0,7, *precision* bernilai 0,976, *recall* bernilai 0,965, dan *mean average precision* dengan *intersect over union* 50% (mAP50) bernilai 0,976. Sehingga pada penelitian ini digunakan YOLOv5x dengan interpolasi. Beberapa hasil pengujian yang dilakukan dengan model yang dipilih terlihat pada Gambar 6.

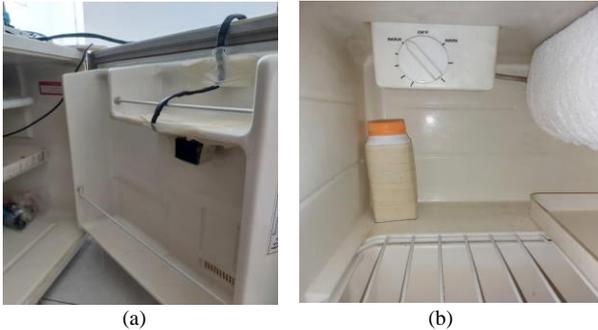
C. Hasil Perancangan dan Pengujian Internet of Things

Pada penelitian ini, mikrokontroler ESP32 mengirimkan data ke komputer melalui WiFi menggunakan socket UDP. Mikrokontroler bertindak sebagai *server* sedangkan komputer sebagai *client*. Mikrokontroler ESP32 terkoneksi ke internet melalui jaringan WiFi dan mendapatkan *IP address* yang digunakan untuk mengoneksikan komputer dengan mikrokontroler. Komputer mengirimkan data *request* sebuah pesan ke mikrokontroler ESP32, kemudian mikrokontroler mengirimkan data *float* berbentuk *byte* dan diterima komputer. Data *byte* yang diterima oleh komputer akan diubah menjadi *float*. Data tersebut harus berjumlah 768 nilai suhu dalam sebuah frame agar gambar yang diperoleh tidak cacat.

Selain pengiriman data dari mikrokontroler ESP32 ke



Gambar 6. Gambar hasil pengujian model pada data *test*.



Gambar 7. (a) Letak kamera termal, dan (b) objek yang dideteksi di dalam *chiller*.

komputer, juga dilakukan pengiriman data ke *server* Thingspeak dari PC agar dapat ditampilkan pada *dashboard* Thingspeak. *Dashboard* Thingspeak dibuat untuk memperlihatkan data rata-rata suhu tiap kelas objek, jumlah objek, dan rata-rata suhu *chiller* di luar jaringan WiFi yang digunakan pada sistem ini. Sehingga pengguna dari sistem ini dapat memantau suhu *chiller* tanpa perlu masuk ke jaringan WiFi yang sama.

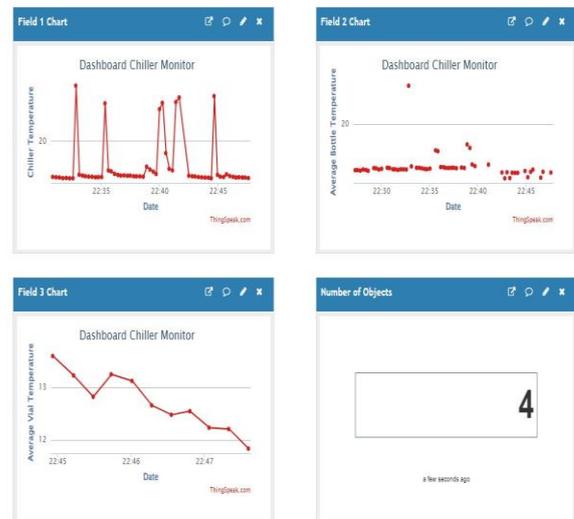
D. Hasil Pengujian Seluruh Sistem

Hasil dari pengujian seluruh sistem dilakukan dengan menggabungkan sistem kamera termal, *neural network*, dan IoT. Kemudian kamera termal diletakkan di dalam ruangan *chiller* yang dihadapkan ke objek seperti ditunjukkan pada Gambar 7. Deretan nilai suhu yang diperoleh kamera termal MLX90640 dikirimkan dalam bentuk *byte* melalui jaringan WiFi dengan socket UDP ke komputer. Data suhu tersebut kemudian divisualisasikan menjadi gambar termal. Citra ini dijadikan gambar *binary large object* untuk dilakukan *inference* ke model YOLOv5 sehingga diperoleh data *bounding box* berupa koordinat, tinggi, dan lebar botol yang dideteksi. Data-data yang didapatkan diaplikasikan dengan fungsi *Non-maximum Suppression* untuk menghilangkan *bounding box* yang *overlap*.

Rata-rata suhu objek dibandingkan dengan nilai batas suhu yang didapat setelah melakukan observasi, dimana menurut ASEAN-Japan Transport Partnership produk minuman harus disimpan pada suhu 5-18°C [1]. Sedangkan menurut WHO merekomendasikan vial yang berisi obat-obatan harus disimpan pada suhu 2-8°C [18]. Bila suhu objek lebih besar dari pada nilai batas suhu maka sistem akan menyalakan alarm. Lalu sistem menghitung rata-rata suhu ruangan *chiller* yang kemudian dibandingkan juga dengan batas suhu. Alarm akan dinyalakan bila rata-rata suhu ruangan *chiller* lebih besar dari batas suhu yang ditetapkan. Tahap terakhir dari sistem adalah menampilkan *Dashboard* Python yang berisi gambar termal, hasil *object detection*, rata-rata suhu ruangan *chiller*, rata-rata suhu tiap objek, dan peringatan bila terdapat

Channel Stats

Created: *a day ago*
Last entry: *less than a minute ago*
Entries: 121



Gambar 8. Gambar *dashboard* Thingspeak menampilkan data suhu.

suhu anomali. Dari hasil ini data jumlah objek, rata-rata suhu tiap objek, dan rata-rata suhu *chiller* dikirim ke *server* Thingspeak agar dapat ditampilkan pada komputer. Tampilan *dashboard* di komputer ini dapat dilihat pada Gambar 8.

IV. KESIMPULAN

Pada penelitian telah dilakukan rancang bangun sistem monitoring suhu *chiller* menggunakan kamera termal, *computer vision*, YOLO, dan IoT. Kamera termal MLX90640 dapat digunakan untuk mengukur suhu permukaan objek setelah dilakukan proses kalibrasi. Karakteristik kamera ini cenderung mempunyai persamaan polinomial orde 2. Model YOLOv5x dengan dataset interpolasi memiliki *metrics* paling baik dengan menghasilkan nilai mAP50:95 sebesar 0,7; *object loss* terkecil sebesar 0,013; *box loss* terkecil sebesar 0,016; *precision* sebesar 0,976; dan *recall* sebesar 0,965. Sistem menggunakan jaringan WiFi melalui UDP untuk mengirimkan data *array* suhu dari kamera termal MLX90640 dengan mikrokontroler ESP32 sebagai *server* ke PC sebagai *client* secara *realtime*. Sistem akan memberikan peringatan berupa suara bila nilai suhu objek yang diperoleh bernilai lebih besar dari nilai batas suhu yang ditentukan. Data suhu juga dikirimkan ke *server* Thingspeak untuk ditampilkan pada komputer sehingga sistem dapat dipantau dari jauh.

DAFTAR PUSTAKA

- [1] P. T. C. I. Consult, "A cold chain study of Indonesia," *cold Chain agri-food Prod. ASEAN. ERIA Res. Proj. Rep. FY2018*, no. 11, pp. 101–147, 2019.
- [2] J.-W. Han, M. Zuo, W.-Y. Zhu, J.-H. Zuo, E.-L. Lü, and X.-T. Yang, "A comprehensive review of cold chain logistics for fresh agricultural products: Current status, challenges, and future trends," *Trends Food Sci. & Technol.*, vol. 109, pp. 536–551, 2021.
- [3] I. Masudin, A. Ramadhani, D. P. Restuputri, and I. Amallynda, "The effect of traceability system and managerial initiative on Indonesian food cold chain performance: A Covid-19 pandemic perspective," *Glob. J. Flex. Syst. Manag.*, vol. 22, no. 4, pp. 331–356, 2021.
- [4] J. Gillespie *et al.*, "Real-time anomaly detection in cold chain transportation using IoT technology," *Sustainability*, vol. 15, no. 3, p. 2255, 2023.
- [5] N. Burke, K. A. Zacharski, M. Southern, P. Hogan, M. P. Ryan, and C. C. Adley, "The dairy industry: process, monitoring, standards, and

- quality," *Descr. food Sci.*, vol. 162, 2018.
- [6] M. L. Fahrni *et al.*, "Management of COVID-19 vaccines cold chain logistics: a scoping review," *J. Pharm. policy Pract.*, vol. 15, no. 1, p. 16, 2022.
- [7] A. Y. Prinanto, M. Rivai, and R. Setiawan, "Diagnostics of Magnetron Transmitter System using Thermal Camera and Neural Network," *1st International Conference on Information System & Information Technology (ICISIT)*, Yogyakarta, 27-28 July 2022, pp. 73-78.
- [8] A. Chastity and M. Rivai, "Implementasi kamera termal pada pemadam api otomatis," *J. Tek. ITS*, vol. 9, no. 1, pp. A138--A143, 2020.
- [9] I. S. Jati and M. Rivai, "Implementasi thermal camera pada pengaturan pendingin ruangan," *J. Tek. ITS*, vol. 8, no. 2, pp. A66--A71, 2020.
- [10] F. Aufar, M. A. Murti, and M. H. Barri, "Design of Non-Contact Thermometer Using Thermal Camera For Detecting People With Fever," *International Conference on Computer Science and Engineering (IC2SE)*, Padang, 16 November 2021, vol. 1, pp. 1-5.
- [11] E. Alpaydin, *Introduction to Machine Learning*, 4th ed. Cambridge: MIT press, 2020.
- [12] V. Wiley and T. Lucas, "Computer vision and image processing: a paper review," *Int. J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 29-36, 2018.
- [13] M. P. Batubara, M. Rivai, and T. Mujiono, "Sistem inspeksi panas kabel menggunakan kamera termal dan neural network untuk pencegahan potensi kebakaran," *J. Tek. ITS*, vol. 11, no. 3, pp. A210--A216, 2022.
- [14] Y. Yao, D. Pei, Q. Ren, and T. Dan, "Research On Lightweight Flame Detection Algorithm Based On Improved YOLOv5," *International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, Changcun, 20-22 May 2022, pp. 1128-1132.
- [15] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, 2022.
- [16] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, "Securing The Internet of Things (IoT): A Security Taxonomy for IoT," *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications & 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, New York, 1-3 August 2018, pp. 163-168.
- [17] S. Kim, S. Shin, and J. Moon, "UDP-Based Extremely Low Latency Streaming," *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, 8-10 January 2022, pp. 94-99.
- [18] G. Goldwood and S. Diesburg, "The effect of cool water pack preparation on vaccine vial temperatures in refrigerators," *Vaccine*, vol. 36, no. 1, pp. 128-133, 2018.