

Journal of Applied Mechanical Engineering and Green Technology

Journal homepage: <https://ojs2.pnb.ac.id/index.php/JAMETECH>
p-ISSN: 2655-9145; e-ISSN: 2684-8201

Deteksi api kebakaran berbasis *computer vision* dengan algoritma YOLO

I Gede Suputra Widharma^{1*}, Putri Alit Widyastuti Santiary¹, I Nengah Sunaya¹,
I Ketut Darminta¹, I Gde Nyoman Sangka¹ dan Putu Ardy Wahyu Widiatmika¹

¹Jurusan Teknik Elektro, Politeknik Negeri Bali, Kampus Bukit Jimbaran, Badung, Bali, Indonesia
*Email: suputra@pnb.ac.id

Abstrak

Kebakaran adalah suatu yang terjadi akibat adanya api yang tidak ditangani dengan baik dan cepat. Kejadian ini sangat berbahaya mengingat berpotensi menimbulkan kerugian materil dan korban jiwa. Berbagai sistem pendeteksi api kebakaran telah dikembangkan untuk mendeteksi dan mencegah kebakaran. Selama ini banyak penelitian pendeteksi api menggunakan sensor api dan asap, dan lainnya, tetapi hal itu saja tidak cukup. Dibutuhkan alat yang sekaligus bisa mendeteksi api dan mengetahui bagaimana kondisi keadaan saat terjadi kebakaran. Semakin berkembangnya teknologi, penggunaan *computer vision* dimanfaatkan untuk mendeteksi api kebakaran secara *real-time*. Selain dapat mendeteksi api, sistem juga dapat memberikan gambaran keadaan terkini saat terjadi kebakaran melalui kamera. Pada penelitian dirancang deteksi api kebakaran berbasis *computer vision*. Dimana menggunakan algoritma *You Only Look Once* (YOLO) untuk mendeteksi api kebakaran serta disertai notifikasi telegram dan buzzer. Pada pengujian deteksi api menggunakan 4 sumber api yang berbeda yaitu lilin obor kecil, obor besar dan sabut kelapa. Dari data pengujian deteksi api di berbagai *background* mendapatkan hasil nilai *accuracy* sebesar 0.8, *precision* sebesar 1 dan *recall* 0.8 pada siang hari. Sedangkan pada malam hari mendapatkan hasil *accuracy* sebesar 0.96, *precision* sebesar 1 dan *recall* 0.96. Tingkat nilai *accuracy* dan *recall* pada pengujian berbagai jarak menghasilkan sistem deteksi api akan mengalami penurunan seiring jauhnya jarak api dengan kamera. Untuk nilai *precision* menghasilkan nilai 1 di berbagai jarak pada siang dan malam hari. Itu berarti ketepatan hasil klasifikasi sebesar 100%. Nilai *precision* yang stabil tersebut dipengaruhi oleh pembacaan deteksi api, dimana sistem tidak pernah mendeteksi objek lain sebagai api.

Kata kunci: YOLO, computer vision, deteksi, kebakaran

Abstract: *Fire is something that occurs as a result of a fire that is not handled properly and quickly. This incident is very dangerous considering the potential to cause material losses and loss of life. Various fire detection systems have been developed to detect and prevent fires. So far, many kinds of research on fire detection use fire and smoke sensors, and others, but that alone is not enough. It takes a tool that can detect a fire at the same time and know-how the conditions are when a fire occurs. With the development of technology, the use of computer vision is used to detect fires in real-time. Besides being able to detect fires, the system can also provide an overview of the current state of the fire through the camera. In this study, a computer vision-based fire detection was designed. Where it uses the You Only Look Once (YOLO) algorithm to detect fires and is accompanied by telegram notifications and buzzers. In the fire detection test, 4 different fire sources were used, namely small torch candles, large torches, and coconut fiber. From the test data for fire detection in various backgrounds, the accuracy value is 0.8, precision is 1 and recall is 0.8 during the day. While at night, the accuracy is 0.96, precision is 1 and recall is 0.96. The level of accuracy and recall values in testing various distances results in a fire detection system that will decrease as the distance between the fire and the camera is far away. The precision value produces a value of 1 at various distances during the day and night. That means the accuracy of the classification results is 100%. The stable precision value is influenced by fire detection readings, where the system never detects other objects as fire.*

Keywords: YOLO, computer vision, detection, fire

Penerbit @ P3M Politeknik Negeri Bali

1. Pendahuluan

Kebakaran merupakan suatu peristiwa yang timbul dikarenakan adanya api yang tidak terkontrol atau tidak terkawal. Kejadian ini sangat berbahaya karena berpotensi menimbulkan korban jiwa dan kerugian materil. Apabila api

sudah membesar dan tidak terkendali akan sangat sulit untuk cepat dipadamkan, apalagi jika daerah kebakaran tersebut jauh dan sulit dijangkau oleh mobil pemadam kebakaran. Penyebab kebakaran dapat dibedakan menjadi 3 kelas, yaitu Kelas A yang disebabkan oleh benda yang

mudah terbakar yaitu kayu, kertas, karet, dll. Sedangkan untuk Kelas B yang disebabkan oleh cairan yang mudah terbakar seperti bensin, oli, spiritus, dan cairan lain yang mudah terbakar. Pada Kelas C dimana disebabkan oleh konslet pada aliran listrik [1]. Terjadinya api menimbulkan kebakaran disebabkan ketiga elemen oksigen, panas dan bahan bakar saling bertemu satu sama lain, apabila ketiga unsur tersebut tidak bertemu maka titik api tidak akan terjadi [2]. Dari paparan tersebut untuk mengantisipasi adanya kerugian materil dan korban jiwa yang disebabkan oleh bencana kebakaran maka dibutuhkan sistem untuk mendeteksi kebakaran sebagai peringatan sedini mungkin terhadap indikasi kebakaran. Selama ini banyak penelitian pendeteksi api menggunakan sensor api dan asap, dan lainnya, tetapi hal itu saja tidak cukup. Dibutuhkan alat yang sekaligus bisa mendeteksi api dan mengetahui bagaimana kondisi keadaan saat terjadi kebakaran.

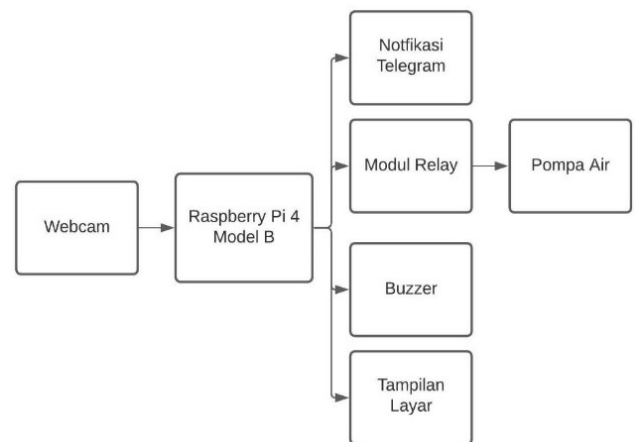
Deteksi api merupakan salah satu teknologi yang saat ini dikembangkan seiring perkembangan teknologi yang sangat cepat dan diikuti dengan perkembangan di bidang *Artificial Intelligence* (AI) atau Kecerdasan Buatan. Salah satu cabang AI salah satunya yaitu *computer vision*. *Computer vision* tentunya tidak lepas dari penggunaan library OpenCV. OpenCV merupakan *library open source* yang tujuannya dikhususkan untuk melakukan pengolahan citra [3]. Penggunaan *computer vision* dalam alat pendeteksi kebakaran sangat penting, karena dengan menggunakan *computer vision* petugas akan mengetahui keadaan ruangan atau gedung yang terbakar. Informasi keadaan tersebut dapat berguna bagi tim pemadam untuk mengetahui kondisi lapangan dan bisa menentukan rencana/tindakan yang akan diambil. Sampai sekarang sudah banyak penelitian-penelitian yang mengarah kepada permasalahan ini. Salah satunya sebagai berikut, A. Zarkasi, et al [4] dengan judul *Implementation Of Fire Image Processing for Land Fire Detection Using Color Filtering Method* dan Hatekar.A, et al [5] dengan judul *Fire Detection on a Surveillance System Using Image Processing*. Di dalam kedua penelitian tersebut membahas tentang seberapa akurat pendeteksi api dengan menggunakan metode filter warna atau RGB. Dari penelitian yang dilakukan Zarkasi, didapatkan hasil api terdeteksi jika jarak kamera dengan api di bawah 7 meter. Kelemahan dari kedua penelitian tersebut adalah menurunnya tingkat akurasi pendeteksi api jika digunakan di luar ruangan. Pada penelitian A. R. Warsil, et all dengan judul *Pembuatan Pendeteksi Obyek Dengan Metode You Only Look Once (YOLO) untuk Automated Teller Machine (ATM)* [6], membahas tentang pendeteksi benda-benda yang terlarang ketika berada di ATM seperti helm. Penelitian ini menggunakan metode *You Only Look Once* (YOLO) yang dapat mengenal obyek-obyek yang telah ditentukan.

Pada penelitian ini dibuat sistem pendeteksi api sedini mungkin di dalam ruangan berbasis *image processing* yang menggunakan Raspberry Pi B+ dengan memanfaatkan kamera untuk mengambil gambar. Untuk mendeteksi api, metode *image processing* yang digunakan yaitu metode YOLO (*You only look once*). Dalam penerapannya alat ini nantinya bisa mendeteksi api yang berada di dalam ruangan ataupun luar ruangan. Obyek api yang terdeteksi kamera nantinya akan dicocokkan dengan dataset yang sudah disiapkan. *Dataset* ini merupakan hasil *training* AI, dimana AI diajarkan untuk mengenal api dengan berbagai gambar

dan latihan pengenalan yang berulang-ulang. Jika gambar tangkapan kamera memiliki kecocokan dengan api, maka api yang terdeteksi akan ditandai. Setelah api terdeteksi, pompa air dan buzzer akan aktif sehingga api dapat dipadamkan dengan air dan para penghuni gedung akan mengetahui adanya kebakaran. Dengan pendeteksi api berbasis *computer vision* diharapkan bisa menjadi deteksi dini terjadinya kebakaran dan sekaligus dapat memberikan gambaran terkini keadaan pada gedung/ruangan yang terjadi kebakaran. Sehingga dapat dijadikan acuan untuk membuat keputusan/rencana untuk petugas keamanan gedung/pemadam kebakaran berdasarkan kondisi terkini.

2. Metode dan Bahan

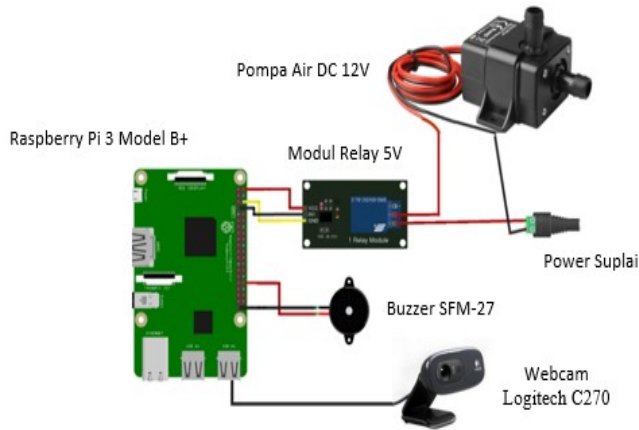
Pada penelitian ini, metode yang digunakan untuk mendeteksi api kebakaran dengan menggunakan metode YOLOv3-tiny dengan menggunakan *pre-trained weight* dan model dari darknet *open source* YOLO. Sehingga perlu menyesuaikan kebutuhan *software* dan *hardware* dalam mengimplementasikan YOLOv3-tiny dari darknet untuk *dataset* yang digunakan dalam penelitian. Proses uji coba dilakukan dengan melakukan proses pengujian metode dengan menguji beberapa kondisi. Proses uji coba dilakukan untuk mengetahui batasan deteksi api. Sedangkan proses evaluasi dilakukan untuk mengetahui tingkat keakuratan kinerja model dan untuk menganalisis hasil uji coba validasi. Validasi kinerja program dilakukan dengan menghitung seberapa besar nilai akurasi klasifikasi objek.



Gambar 1. Diagram Blok Perancangan Hardware

Pada Gambar 1 merupakan diagram blok perancangan *hardware*, terdapat perangkat yang digunakan yaitu menggunakan kamera webcam, Raspberry Pi 4 Model B sebagai otak untuk mengendalikan kerja sistem. Tangkapan gambar dari kamera webcam akan diterima oleh Raspberry Pi 4 Model B sebagai data. Data yang yang diterima dari kamera akan diolah dengan menggunakan *image processing* OpenCV untuk memberikan *output*. Pada bagian *ouput* terdiri atas buzzer dan pompa air. Dalam penerapannya alat ini nantinya bisa mendeteksi api yang berada di dalam ruangan ataupun luar ruangan. Cara kerja alat yaitu, webcam akan menangkap gambar sekitar dan tangkapan gambar tersebut akan diterima oleh Raspberry Pi 4 Model B. Data tersebut akan diolah menggunakan *image processing* dan dicocokkan dengan *dataset* yang sudah dilatih di Google Colab. Jika gambar tangkapan kamera

memiliki kecocokan dengan api, maka api yang terdeteksi akan ditandai. Setelah itu Raspberry Pi 4 Model B akan menghidupkan buzzer dan menutup anak kontak dari modul relai sehingga pompa air akan aktif sehingga api dapat dipadamkan dengan air dan para penghuni gedung akan mengetahui adanya kebakaran. Untuk notifikasi, *User* akan dikirimkan notifikasi melalui aplikasi telegram sehingga dapat mengetahui adanya kebakaran. Ketika api tidak terdeteksi lagi, maka pompa air dan buzzer akan mati. Pada Gambar 2 disajikan *wiring* komponen.

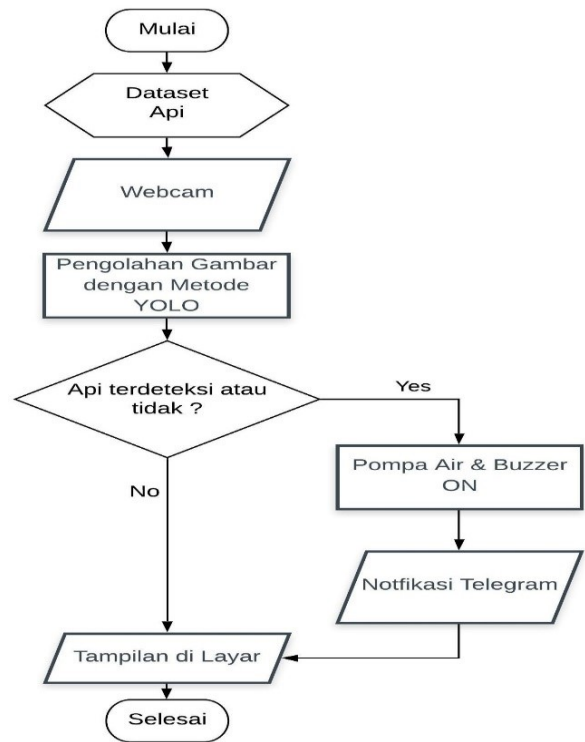


Gambar 2. Komponen wiring

Pada pembuatan alat ini terdapat beberapa komponen-komponen yang digunakan yaitu:

- Raspberry Pi 4 Model B
- Modul relai 5V 1 channel
- Buzzer SFM-27 DC 3-24V
- Webcam logitech C270
- Pompa air TJ1600
- Power suplai
- Kabel jumper dupont dan kabel NYAF 1.55mm

Berdasarkan flowchart pada Gambar 3, webcam merupakan alat yang digunakan sebagai alat pengambil gambar yang nantinya akan dideteksi. Kemudian hasil dari pengambilan gambar tersebut diproses dengan menggunakan metode YOLO (*You Only Look Once*) dengan menggunakan *dataset* yang digunakan untuk melakukan *transfer learning* atau pencocokan gambar yang sudah dilatih pada *dataset*. Disini terdapat kondisi yang pertama adalah apabila ada api yang terdeteksi dan yang kedua adalah apabila api tidak terdeteksi. Apabila api terdeteksi, maka proses selanjutnya adalah menghitung keakuratan gambar yang ditangkap oleh kamera. Setelah proses perhitungan selesai maka hasil yang sudah didapatkan nantinya akan ditampilkan dilayar dan buzzer akan berbunyi. Lalu notifikasi untuk *user* akan dikirim melalui telegram. Namun jika api tidak terdeteksi maka tampilan kondisi yang ditangkap kamera akan ditampilkan di layar. Alat ini dirancang untuk ditempatkan pada ruangan/pabrik yang terdapat kategori benda-benda penyebab kebakaran yaitu kelas A (kayu, kertas, karet, dan lain-lain) dan kelas B (bensin, oli, solar, spiritus, dan cairan lain). Untuk memaksimalkan jangkauan kamera pada ruangan, kamera akan diletakkan pada sudut ruangan. Alat ini diharapkan sebagai pendeteksi api untuk mengetahui kebakaran, sehingga kejadian kebakaran dapat diminimalisir.



Gambar 3. Flowchart sistem

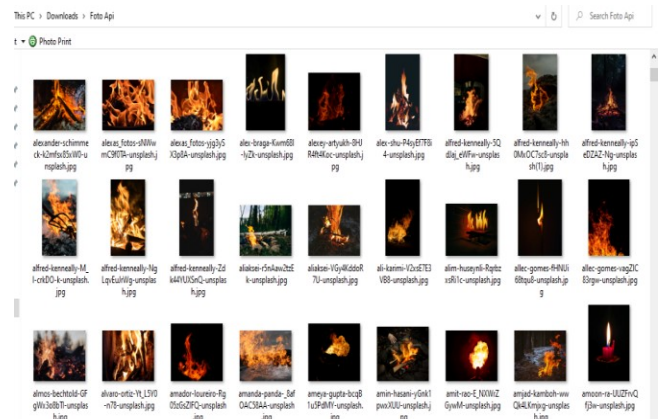
3. Hasil dan Pembahasan

Hasil dan pembahasan dibagi menjadi 4 bagian yaitu:

- Melatih dataset deteksi api dengan menggunakan metode YOLO
- Nilai *accuracy*, *precision*, *recall* pendeteksi api pada background berbeda di skenario siang hari dan malam hari
- Pengaruh jarak terhadap akurasi deteksi api
- Pengujian bot telegram

3.1. Melatih dataset deteksi api

Untuk mendeteksi api dengan menggunakan metode YOLO, dilakukan dalam beberapa tahap yaitu mengumpulkan gambar/*image* api. Gambar/*image* didapatkan dari hasil *download* dari internet dan mengumpulkan foto api dari kamera. Gambar api yang dikumpulkan akan di-*labelling* menggunakan aplikasi *labelimg*. Gambar dan hasil label akan digunakan untuk melatih *dataset* yang dilakukan Google Colab (Gambar 4).



Gambar 4. Kumpulan gambar api

a. Mendownload gambar/images

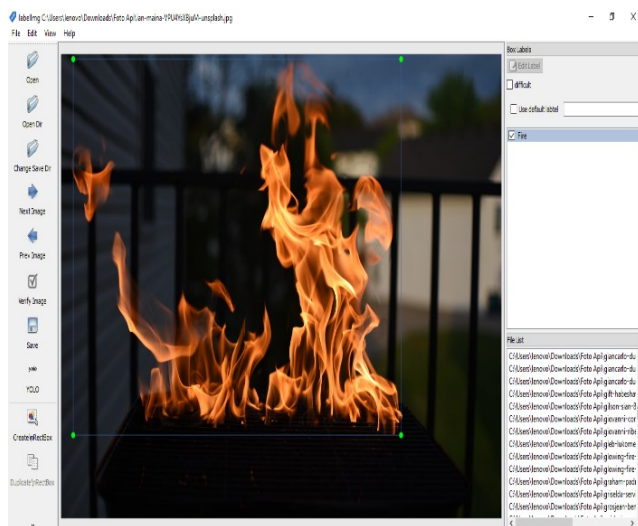
Gambar/images yang digunakan yaitu berbagai macam gambar api. Gambar yang digunakan untuk membuat *dataset* ini yaitu 1600 buah yang didapatkan secara manual menggunakan webcam dan hasil *download* dari internet. Perbandingan jumlah gambar hasil manual dan hasil *download* yaitu 50:50. Gambar-gambar yang sudah dikumpulkan ditempatkan dalam satu *folder* yang sama.

b. Proses anotasi

Sebelum melakukan proses pelatihan dan pengujian, diperlukan data anotasi. Anotasi merupakan proses memberikan label pada gambar dengan cara memberikan *bounding box* atau kotak batas beserta nama *class* pada objek disetiap citra. Anotasi data diawali dengan menggambar kotak pembatas (*bounding box*) tersebut dalam suatu *file* gambar. Proses pembuatan ini menggunakan aplikasi LabelImg untuk memberi anotasi pada gambar/images. Kelas yang diatur dalam penelitian ini hanya terdapat satu buah yaitu *Fire*.

Hasil *output* dari anotasi tersebut adalah sebuah file data yang memuat informasi letak kotak batas berupa kordinat anotasi dalam format file *.txt*. Gambar api dan hasil anotasi diletakkan pada satu folder yang sama dan diekstrak menjadi file *.ZIP*

Agar *training* data dihasilkan maksimal dan model yang dihasilkan memiliki akurasi yang tinggi, pelabelan yang dilakukan pada gambar/images harus melabel objek yang tidak blur atau terlihat jelas. Semakin jelas dan objek semakin tinggi juga akurasinya. Dalam pelabelan gambar, sudut pandang objek dan variasi bentuk yang beragam di dalam *dataset* juga harus diperhatikan agar pendeteksian nantinya memiliki akurasi yang tinggi. Berikut merupakan proses memberikan label dan hasil *labelling* pada masing-masing gambar.



Gambar 5. Kumpulan Gambar Api

c. Melatih dataset pada google colab

Proses *training* atau proses latih adalah proses dimana *Neural Network* akan dilatih untuk mempelajari suatu pola dari suatu objek yang ingin dideteksi yang nantinya dapat digunakan untuk melakukan pengenalan deteksi objek api dengan tingkat akurasi yang tinggi. Langkah pertama mengaktifkan GPU pada google colab. Setelah itu

menghubungkan Google Collab ke Google Drive dengan koding seperti pada Gambar 6.

```
from google.colab import drive
drive.mount('/content/gdrive')
!ln -s /content/gdrive/My\ Drive/ /mydrive
!ls /mydrive
```

Gambar 6. Koding menghubungkan Google Collab ke Google Drive

3.2. Hasil pengujian *accuration*, *precision* dan *recall*

Pengujian akurasi, presisi dan *recall* berdasarkan skenario yang berbeda bertujuan untuk mengetahui performa deteksi api jika dihadapkan dengan kondisi yang berbeda. Hasil dari skenario dibagi menjadi dua yaitu skenario saat siang hari dan malam hari. Data yang dikumpulkan masing-masing sebanyak 75 data. Pada data hasil pengujian deteksi api pada siang hari dilakukan dengan *background* dan sudut pandang yang berbeda-beda. Data yang didapati *bounding box* dapat terbentuk di sekeliling objek api dengan menampilkan nilai *confidence*-nya.

Sedangkan pada malam hari, data didapatkan dengan melakukan deteksi dengan sudut yang berbeda-beda. Berdasarkan hasil deteksi, api yang berhasil dideteksi memperoleh nilai *confidence* berkisar antara 20.61%-96.32%. Warna api yang berhasil ditangkap kamera saat malam hari memiliki perbedaan jika dibandingkan pada siang hari, dimana pada malam hari api memiliki warna api dominan putih. Hal ini dikarenakan kemampuan webcam yang terbatas sehingga tangkapan gambar yang dihasilkan mengalami *overexposure*. *Overexposure* adalah keadaan dimana image/gambar yang dihasilkan lebih cerah karena terlalu banyak sinar/cahaya yang mengenai sensor dari kamera.

Pada pengujian jarak api bertujuan untuk mengetahui seberapa jauh jarak api yang bisa dideteksi dan pengaruh jarak terhadap akurasi deteksi alat. Pengujian dilakukan siang dan malam hari di ruangan setengah terbuka. Sumber api yang digunakan yaitu lilin, obor kecil, obor besar dan hasil pembakaran sabut kelapa. Sumber api diuji saat siang dan malam hari dari berbagai jarak dimulai dari 0.25m-7m. Pada masing-masing jarak dilakukan 15 kali pengujian. Pada siang hari deteksi api lilin jarak maksimal yang bisa dideteksi hanya sampai 1 meter dan untuk malam hari bisa sampai 3.5 meter. Api obor kecil pada siang hari bisa mendeteksi api sampai jarak 4 meter dan pada saat malam bisa sampai 7 meter. Api obor besar pada siang hari bisa mendeteksi api sampai jarak 5.5 meter dan pada saat malam bisa sampai 7 meter. Sedangkan untuk api hasil pembakaran sabut kelapa dapat dideteksi sampai 7 meter pada siang hari.

3.3. Pembahasan

Menganalisa hasil dari pengujian adalah sesuatu yang penting karena menggambarkan seberapa baik sistem dalam melakukan klasifikasi data. Pengukuran kinerja menggunakan *confusion matrix* dimana hasil dari deteksi api akan digolongkan menjadi empat bagian yaitu *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN). Berdasarkan hasil tersebut akan diperoleh nilai *accuracy*, *precision*, dan *recall*. *Accuracy* merupakan pengukuran seberapa benar sebuah sistem dapat

mengklasifikasi dari keseluruhan. *Precision* merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Sedangkan *Recall* merupakan pengukuran untuk data dengan klasifikasi positif yang benar oleh sistem.

3.3.1. Pengujian akurasi, presisi dan recall berbagai background

Pengujian ini dilakukan untuk mengetahui nilai *accuracy*, *precision*, dan *recall* dari deteksi api pada background yang berbeda pada siang hari dan malam hari. Seluruh gambar yang didapat dikelompokkan menjadi 2 kelompok, yaitu hasil deteksi saat siang hari dan malam hari. Dari 2 kelompok data tersebut akan dilakukan perhitungan nilai *accuracy*, *precision*, dan *recall* yang kemudian hasil perhitungan semua nilai *accuracy*, *precision*, dan *recall* dirata-ratakan. Berikut perhitungan nilai *accuracy*, *precision*, dan *recall* dari 2 kelompok tersebut:

a. Siang hari

Pengujian sistem yang dilakukan terhadap deteksi api mendapatkan hasil seperti Tabel 1. Hasil deteksi api pada siang hari berjumlah 75 data. Dari 75 data, api yang dikenali sebagai api berjumlah 60 data, bukan api yang dikenali sebagai api tidak ada, api yang dikenali sebagai bukan api berjumlah 15 data, dan bukan api yang dikenali sebagai bukan api tidak ada. Maka dapat dibuat *confusion matrix* untuk deteksi api pada siang hari seperti pada Tabel 1.

Tabel 1. Hasil Deteksi pada Siang Hari

Terdeteksi	Real	
	Api	Bukan Api
Api	60	0
Bukan Api	15	0

Accuracy yang didapatkan adalah 0,8. Nilai tersebut menunjukkan bahwa data terklasifikasi secara benar sebesar 80 %. *Precision* merupakan perbandingan jumlah data yang kategori positif yang diklasifikasikan secara benar oleh sistem dan keseluruhan data yang terklasifikasi positif. Nilai *precision* yang dihasilkan adalah 1. Nilai tersebut menunjukkan bahwa ketepatan hasil klasifikasi adalah sebesar 100 %. *Recall* adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar. Nilai *recall* yang dihasilkan adalah 0,8. Nilai tersebut menunjukkan bahwa tingkat keberhasilan mengenali suatu data dari seluruh data yang dikenali adalah sebesar 80%.

b. Malam Hari

Pengujian sistem yang dilakukan terhadap deteksi api mendapatkan hasil seperti Tabel 4.2. Hasil deteksi api pada malam hari berjumlah 75 data. Dari 75 data, api yang dikenali sebagai api berjumlah 72 data, bukan api yang dikenali sebagai api tidak ada, api yang dikenali sebagai bukan api berjumlah 3 data, dan bukan api yang dikenali sebagai bukan api tidak ada. Maka dapat dibuat *confusion matrix* untuk deteksi api pada siang hari seperti Tabel 2. *Accuracy* yang didapatkan adalah 0,96. Nilai tersebut menunjukkan bahwa data terklasifikasi secara benar sebesar 96 %.

Precision merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Nilai *precision* yang dihasilkan adalah 1. Nilai tersebut menunjukkan bahwa ketepatan hasil klasifikasi adalah sebesar 100 %.

Tabel 2. Hasil Deteksi pada Malam Hari

Terdeteksi	Real	
	Api	Bukan Api
Api	72	0
Bukan Api	3	0

Recall adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar. Nilai *recall* yang dihasilkan adalah 0,96. Nilai tersebut menunjukkan bahwa tingkat keberhasilan mengenali suatu data dari seluruh data yang dikenali adalah sebesar 96%.

3.3.2. Analisa keseluruhan

Dari pengolahan data nilai *accuracy*, *precision* & *recall* pada percobaan deteksi api pada background yang berbeda-beda pada siang hari dan malam hari diatas maka dapat dibuat tabel nilai perbandingan *accuracy*, *precision* dan *recall* pada deteksi api saat siang hari dan malam hari seperti Tabel 3.

Dari pengujian sistem yang dilakukan pada siang hari dan malam hari, rata-rata *accuracy* yang didapatkan adalah 0,88. Nilai tersebut menunjukkan bahwa data terklasifikasi secara benar sebesar 88 %.

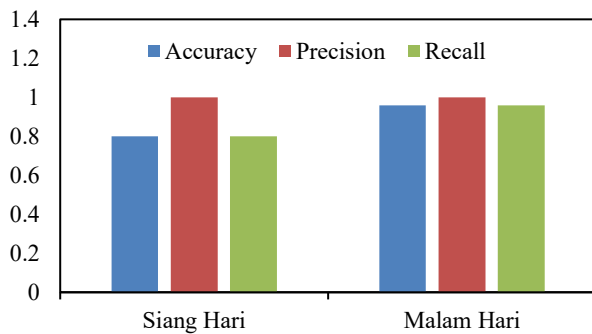
Tabel 3. Perbandingan deteksi pada siang dan malam hari

No	Kondisi	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
1	Siang Hari	0,8	1	0,8
2	Malam Hari	0,96	1	0,96
	Rata-rata	0,88	1	0,88

Precision merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Nilai rata-rata *precision* yang dihasilkan adalah 1. Nilai tersebut menunjukkan bahwa ketepatan hasil klasifikasi adalah sebesar 100 %. *Recall* adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar. Sedangkan nilai rata-rata *recall* yang dihasilkan adalah 0,88. Nilai tersebut menunjukkan bahwa tingkat keberhasilan mengenali suatu data dari seluruh data yang dikenali adalah sebesar 88%. Pada pengujian ini tidak ditemukan deteksi api yang mendeteksi api yang bukan api (*false positive*). Dari data tersebut maka dapat digambar grafik seperti yang disajikan pada Gambar 7.

Dari grafik pada Gambar 7, *accuracy* deteksi api yang tertinggi adalah saat malam hari. Hal ini disebabkan karena intensitas cahaya pada saat malam hari lebih rendah dibandingkan dengan siang hari. Warna api yang berhasil ditangkap kamera saat malam hari memiliki perbedaan jika dibandingkan pada siang hari, dimana pada malam hari api memiliki cahaya lebih tegas namun warna api yang dihasilkan putih dengan pinggiran merah. Warna api yang dihasilkan lebih dominan putih dikarenakan kemampuan webcam yang terbatas sehingga tangkapan gambar yang dihasilkan mengalami *overexposure*. *Overexposure* adalah keadaan dimana *image*/gambar yang dihasilkan sangat

cerah karena terlalu banyak sinar/cahaya yang mengenai sensor dari kamera.



Gambar 7. Grafik Perbandingan Accuracy, Precision & Recall

4. Kesimpulan

Dari hasil penelitian dan pembahasan dapat disimpulkan:

1. Sistem pendeteksian api berhasil mendeteksi api dengan algoritma YOLO
2. Nilai *accuracy*, *precision* dan *recall* pada berbagai background dari percobaan siang hari berturut-turut yaitu 0.8, 1 dan 0.8. Sedangkan pada malam hari nilai *accuracy*, *precision* dan *recall* yang dihasilkan berturut-turut yaitu 0.96, 1 dan 0.96. Percobaan pada malam hari memiliki nilai yang lebih bagus dibandingkan pada saat siang hari dikarenakan background disekitar api berwarna gelap dan cahaya yang ditimbulkan api menjadi sangat kuat. Hal tersebut mengakibatkan deteksi api menjadi sangat akurat.
3. Tingkat nilai *accuracy* dan *recall* pada pengujian berbagai jarak menghasilkan sistem deteksi api akan mengalami penurunan seiring jauhnya jarak api dengan kamera. Untuk nilai *precision* menghasilkan nilai 1 di berbagai jarak pada siang dan malam hari. Itu berarti ketepatan hasil klasifikasi sebesar 100 %. Nilai *precision* yang stabil tersebut dipengaruhi oleh pembacaan deteksi api, dimana sistem tidak pernah mendeteksi objek lain sebagai api. Pendeteksian tingkat *accuracy*, *precision* dan *recall* juga sangat dipengaruhi dengan besar kecilnya sumber api dan dataset yang dilatih. Jika sumber api semakin besar maka tingkat *accuracy* dan *recall* dari deteksi api juga semakin besar. Sedangkan untuk dataset, semakin bervariasi gambar yang dilatih maka ketepatan deteksi api juga semakin tinggi.

Ucapan Terima Kasih

Ucapan terimakasih atas bantuan atau dukungan dari teman sejawat, mahasiswa dan teknisi yang telah membantu menyelesaikan penelitian ini.

Daftar Pustaka

- [1] S. Y. Pradana, F. Utamingrum, and W. Kurniawan, "Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis Sms Gateway Pada Raspberry Pi", *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12), 2018, 7183–7191, 2018.
- [2] Kelvin, P. E. Yuliana and S. Rahayu, "Pemetaan Lokasi Kebakaran Berdasarkan Prinsip Segitiga Api Pada Industri Textile," 2015.
- [3] B. Santoso and R. P. Kristianto, "Implementasi Penggunaan Opencv Pada Face Recognition Untuk Sistem Presensi Perkuliahan Mahasiswa," *Sistemasi*, vol. 9, no. 2, 2020, 352.
- [4] A. Zarkasi, S. Nurmaini, D. Stiawan, Firdaus, Abdurahman, and C. Deri Amanda, "Implementation of fire image processing for land fire detection using color filtering method," *J. Phys. Conf. Ser.*, 1196(1), 2019.
- [5] A. Hatekar, S. Manwani, G. Patil, and A. Parekh, "Fire Detection on a Surveillance System using Image Processing," *Int. J. Eng. Res.*, 6(5), 2017, 121–125.
- [6] A. R. Warsil, M. S. Ghozali, and M. B. Mustafa, "Pembuatan Pendeteksi Obyek dengan Metode You Only Look Once (Yolo) Untuk Automated Teller Machine (Atm)," *Maj. Ilm. UNIKOM*, 17(1), 2019, 69–76.
- [7] A. S. A. Putra and S. M. Nasri, "Evaluasi Penerapan Keselamatan Kebakaran Menggunakan Computerized Fire Safety Evaluation System (Cfses) Pada Gedung Pendidikan Dan Laboratorium Fakultas Ilmu Keperawatan Universitas Indonesia Tahun 2014," 2014.
- [8] A. S. Pratikno, "Implementasi Artificial Intelligence dalam Memetakan Karakteristik, Kompetensi, dan Perkembangan Psikologi Siswa Sekolah Dasar Melalui Platform Offline Conference," *Isbn*, pp. 18–27, 2018.
- [9] D. H. Fathoni, "Rancang Bangun Sistem Smart CCTV untuk Efektivitas Energi Berbasis Yolo CNN Dan Android di Laboratorium Otomasi PPNS," 2019.
- [10] I.G.S. Widharma, I.N. Sunaya, "Aplikasi Sistem Akuisisi Data Pada Sistem Fire Alarm Berbasis Sistem Mikrokontroler," *Jurnal Logic*, 14(2), 2017, 126-130.
- [11] R. H. Pramestya, "Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode Yolo Berbasis Citra Digital," *Master Thesis*, 2018.
- [12] A. Rohim, Y. A. Sari, and Tibyani, "Convolution neural network (cnn) untuk pengklasifikasian citra makanan tradisional," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, 3(7), 2019, 7038–7042.
- [13] A. R. Arganata, "Analisis Perhitungan Bibit Ikan Gurame Menggunakan Webcam Dengan Metode YOLO (You Only Look Once)." 2020.
- [14] R. R. Putra and F. Antony, "Sistem Computer Vision Pengenalan Pola Angka dan Operator Matematika Pada Permainan Kartu Angka Berbasis Jaringan Syaraf Tiruan Perceptron," *J. Ilm. Inform. Glob.*, 9(1), 2018.
- [15] M. S. Hidayatulloh, "Sistem Pengenalan Wajah Menggunakan Metode Yolo (You Only Look Once)," pp. i-43, 2021.
- [16] M. R. Tharhan, "Rancang Bangun Aplikasi Deteksi Kerusakan Jalan Menggunakan Algoritma YOLO," *Politek. Negeri Jakarta*, 2020.