

DETEKSI API *REAL-TIME* DENGAN METODE THRESHOLDING RERATA RGB

Kartika Firdausy, Yusron Saudi, Tole Sutikno

Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan
Kampus III, Jln. Prof Soepomo, Janturan, Yogyakarta, Telp. (0274) 379418
Fax. (0274) 381523, email: kartikaf@indosat.net.id

Abstrak

Kebakaran di hutan maupun pemukiman mengakibatkan kerugian material maupun korban jiwa. Pendeteksian dini terhadap kebakaran merupakan salah satu upaya mengurangi kerugian tersebut. Pada penelitian ini akan dilakukan simulasi pendeteksian dini sumber api yang berasal dari lilin. Metode penelitian yang dilakukan adalah dengan pengambilan citra api secara real-time menggunakan webcam dan kemudian melakukan thresholding, sehingga diperoleh citra yang hanya mempunyai dua nilai derajat keabuan, dan selanjutnya dicari nilai rerata RGB untuk dibandingkan dengan nilai range acuan yang ada di database untuk menentukan apakah citra yang ditangkap webcam berupa api atau bukan. Hasil penelitian menunjukkan bahwa tingkat keberhasilan sistem deteksi api yang dirancang adalah 100 % dapat mendeteksi ada dan tidaknya api pada variasi jarak api ketika background-nya homogen.

Kata kunci: *deteksi api, webcam, thresholding, rerata RGB.*

1. PENDAHULUAN

Akhir-akhir ini sering kali kebakaran terjadi akibat belum adanya penanganan secara dini sehingga dapat memakan korban jiwa maupun material. Biasanya kebakaran dapat diketahui apabila api dan asap sudah membesar [3], jika sudah membesar akan ditemui kesulitan dalam memadamkannya belum lagi jika dihadapkan dengan kondisi lingkungan yang tidak mendukung seperti lokasi kebakaran yang sulit dicapai dengan mobil pemadam kebakaran. Oleh karena itu diperlukan suatu sistem yang dapat mendeteksi secara dini terjadinya kebakaran. Sistem deteksi api tersebut dapat berdasarkan citra yang dipantau secara terus-menerus, sehingga dimungkinkan penanggulangan lebih dini terhadap kebakaran.

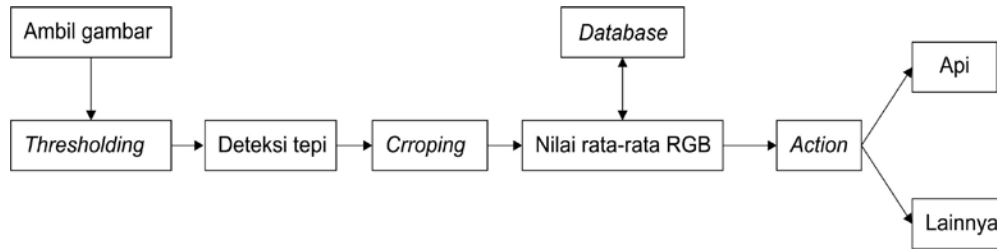
Salah satu penelitian tentang sistem proteksi bahaya kebakaran adalah sistem proteksi bahaya kebakaran berdasarkan monitoring suhu [3]. Penelitian ini hanya sebatas pembacaan suhu menggunakan sensor suhu LM 35. Sistem ini hanya akan bisa membaca suhu apabila mencapai batasan tertentu. Tingkat sensitivitas sensor sangat berpengaruh pada sistem ini.

Penelitian tentang pengolahan citra pada deteksi telur yang dilakukan yang dilakukan oleh Kusumadewi [2] hanya menggunakan citra statis yang telah diolah program komputer. Pada penelitian tersebut belum sampai pada deteksi yang berlangsung secara *real time*.

Dari dua penelitian terdahulu tersebut di atas, pada penelitian ini akan dikembangkan deteksi api menggunakan *webcam* sebagai pengganti sensor suhu dan berdasarkan citra dinamis yang ditangkap webcam secara realtime. Penelitian ini akan mengembangkan teknik pengolahan citra menggunakan bahasa pemrograman Borland Delphi 7.

2. METODE PENELITIAN


Secara garis besar rancangan sistem deteksi api *real-time* dengan metode *thresholding* rerata RGB menggunakan *webcam* ditunjukkan pada Gambar 1. Input sistem berasal dari citra api (dari lilin) yang ditangkap oleh *webcam* dalam format BMP. Citra api dalam format *true color* (BMP) tersebut kemudian diubah ke citra keabuan sehingga menjadi hanya bernilai 1 dan 0. Untuk mempermudah pengenalan api, maka dalam penelitian ini dibuatkan *database* untuk menyimpan citra yang dijadikan acuan dalam pengambilan keputusan.



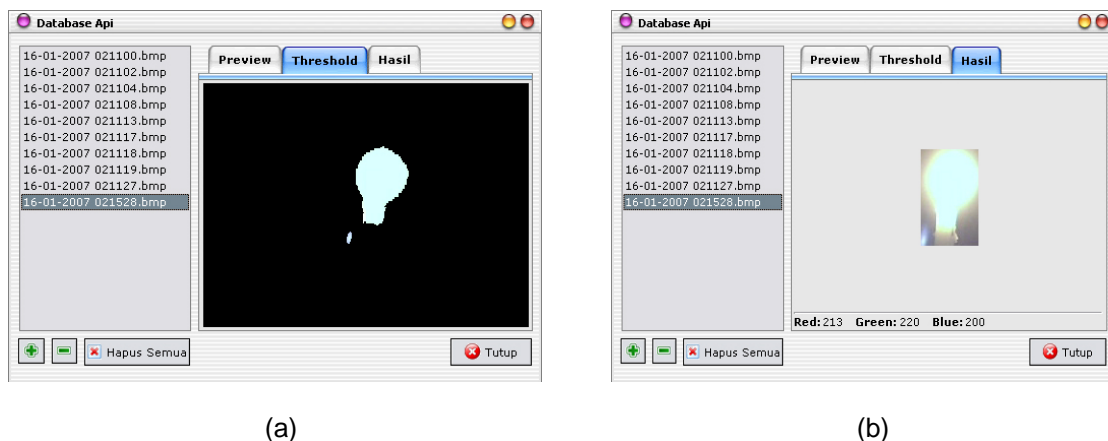
Gambar 1. Blok Diagram Deteksi Api menggunakan Webcam

Metode *thresholding* adalah salah satu teknik pengolahan citra yang digunakan untuk mengubah suatu citra dengan format *true color* menjadi citra biner, yang hanya memiliki 2 buah nilai (0 atau 1). Konversi citra hitam-putih ke citra biner dilakukan dengan operasi pengambangan (*Thresholding*). Operasi ini mengelompokkan nilai derajat keabuan setiap piksel ke dalam dua kelas, hitam dan putih. Pada citra hitam-putih terdapat 256 level, artinya mempunyai skala dari "0" sampai "256" atau $[0,255]$. Dalam hal ini nilai intensitas 0 menyatakan hitam, dan nilai intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih.

2.1. Menghitung Rerata RGB

Untuk menambahkan *database*, dilakukan dengan meng-klik icon , maka *database* citra yang dijadikan sebagai acuan untuk mendeteksi api akan bertambah. Pada *window* ini dapat ditampilkan citra hasil *capture* berukuran 320 x 240 piksel dengan format *true color* pada Tab Preview. Setelah citra api ditangkap, akan dilakukan proses binerisasi dengan metode *thresholding*. Pada proses binerisasi ini citra api akan dikonversi menjadi citra biner yaitu hanya memiliki nilai hitam (0) dan putih (1) saja. Citra api dengan nilai 1 yaitu putih dan latar belakang telur dengan nilai 0 yaitu hitam. Hasil *thresholding* dapat ditampilkan pada Tab Threshold seperti pada Gambar 2 (a) dengan ukuran citra masih 320 x 240 piksel.

Setelah didapatkan citra api dalam bentuk biner, maka akan dicari nilai batas atas, batas bawah, batas kiri, dan batas kanan dengan cara menyecan piksel api. Setelah didapatkan nilai-nilainya maka selanjutnya akan dilakukan proses *cropping* untuk mendapatkan citra api saja. Hasil *cropping* dapat di tampilkan dengan ukuran citra berbeda yang sesuai dengan batas-batasnya. Pada Tab Hasil (Gambar 2 (b)). Informasi jumlah rerata RGB juga ditampilkan untuk masing-masing komponen *Red*, *Green*, dan *Blue*.

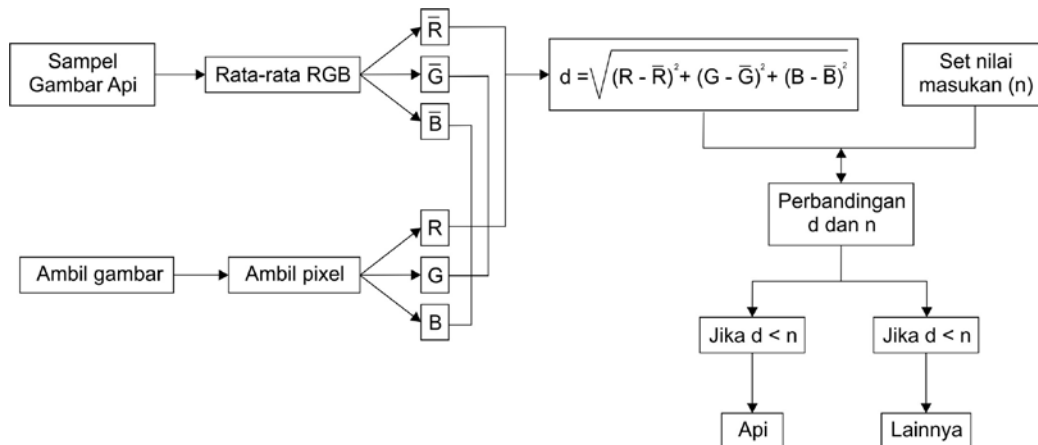


Gambar 2. Windows Database
(a). Citra setelah proses *thresholding* (b). Citra setelah proses *cropping*

Pada penelitian ini diambil 10 sampel api yang akan dijadikan sebagai citra acuan dalam *database* untuk menghitung jarak titik bobot (Persamaan 1) dengan citra yang baru diakuisisi *webcam*.

2.2. Menghitung Jarak Bobot

Nilai jarak bobot \leq toleransi akan dianggap api sedangkan nilai jarak bobot \geq toleransi akan dianggap lain. Proses pengambilan keputusan berdasarkan blok diagram segmentasi api yang terdapat pada Gambar 3, sedangkan persamaan yang digunakan adalah Persamaan (1).



Gambar 3. Blok Diagram Segmentasi Api[3]

$$\text{Jarak titik bobot } (d) = \sqrt{(R - \bar{R})^2 + (G - \bar{G})^2 + (B - \bar{B})^2} \quad (1)$$

dengan:

d = jarak titik bobot

R = nilai rerata *Red* pada *database*

\bar{R} = nilai rerata *Red* citra yang ditangkap

G = nilai rerata *Green* pada *database*

\bar{G} = nilai rerata *Green* citra yang ditangkap

B = nilai rerata *Blue* pada *database*

\bar{B} = nilai rerata *Blue* citra yang ditangkap

Berdasarkan Gambar 3, maka setiap ada citra baru maka proses pencarian titik bobot yang menggunakan persamaan 1. Apabila ada 10 sampel, maka program akan mencari nilai $d \leq n$ mulai dari file *database* yang pertama di-*capture* sampai file ke sepuluh. Apabila ditemukan nilai $d \leq n$ pada file *database* urutan yang kelima, maka proses pencarian akan dihentikan dan program akan memberikan *output* berupa informasi warna merah pada *display* jarak titik bobotnya.

Nilai toleransi (n) didapat dari beberapa kali percobaan, sehingga n pada penelitian ini sebesar 20. Untuk $n < 20$, program akan sulit untuk mengenal api hal ini didasarkan pada tingkat respons pengenalan api rendah sekali. Sedangkan untuk nilai $n > 20$, tingkat respons program sangat tinggi. Kedua hal tersebut dipengaruhi oleh jarak titik bobot yang dibandingkan dengan nilai n tersebut.

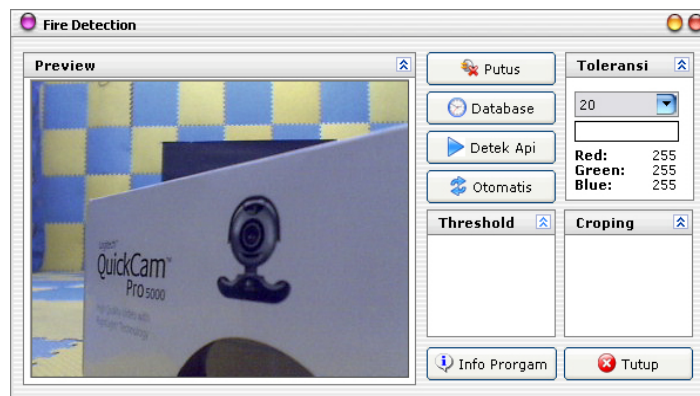
Jika proses pencarian nilai jarak titik bobot yang memenuhi $d \leq n$ tidak ditemukan maka program akan memberikan informasi warna putih pada *display* jarak titik bobot seperti Gambar 5 (b).

Pada proses yang berjalan secara otomatis, proses deteksi api dilakukan secara periodik. Akuisisi citra dilakukan setiap 1 detik kemudian dilanjutkan dengan proses deteksi api. Waktu yang dibutuhkan saat pengambilan keputusan sangat bergantung pada jumlah citra dalam *database*.

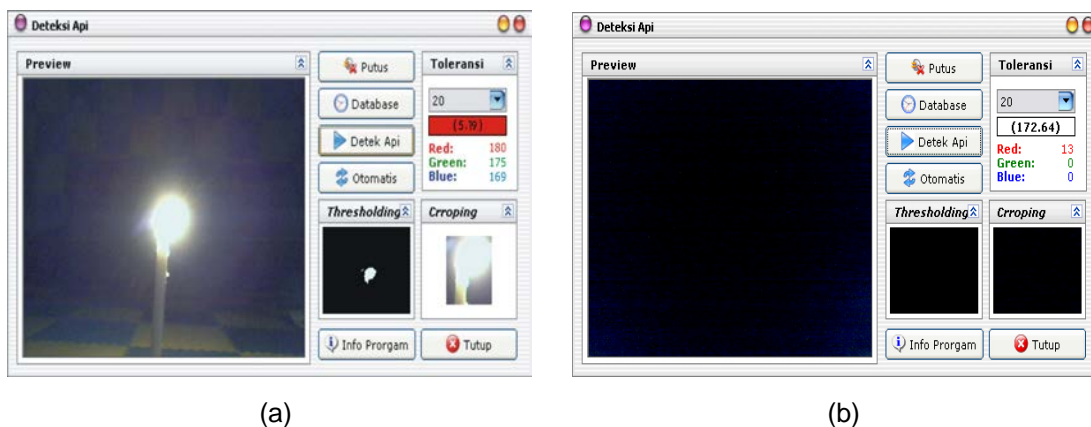
3. HASIL DAN PEMBAHASAN

Pada saat pertama kali, beberapa pola api harus diinputkan pada *database*. Dengan meng-klik menu *database*, maka akan menampilkan window *database* seperti pada Gambar 2. Apabila menu Detek Api dijalankan tanpa tersedianya *database*, maka program akan memberikan peringatan *Error* "Database api masih kosong".

Tampilan program terdiri dari beberapa menu yaitu menu Sambung, menu *Database*, menu Detek Api, menu Otomatis dan menu Tutup. Ketika menu Sambung di klik maka preview dari *webcam* bias ditampilkan dengan ukuran citra sebesar 320 x 240 pada format *true color*. Setelah *webcam* terkoneksi, maka menu *database*, menu Detek Api dan menu Otomatis akan aktif, sedangkan menu Sambung akan berganti menjadi menu Putus (Gambar 4).



Gambar 4. Tampilan Setelah *Webcam* Terkoneksi



(a)

(b)

Gambar 5. Tampilan *Output* Deteksi Api menggunakan *Webcam*

(a) Detek Api dengan Informasi adanya Api.

(b) Deteksi Api dengan Informasi Tidak adanya Api

Pengujian program dilakukan pada kondisi gelap (latar homogen) dengan menguji berbagai jarak api terhadap *webcam*. *Database* diisi dengan 10 sampel api dengan variasi jarak dan jumlah rata rata RGB dapat dilihat pada Tabel 1. Berikutnya akan ditangkap 10 gambar api

untuk dicari jarak titik bobotnya. Hasil deteksi api menggunakan *webcam* disajikan pada Tabel 2.

Tabel 1. *Database* dengan Variasi Jarak

Sampel	Jarak (centimeter)	Rerata RGB		
		Red	Green	Blue
1	25	213	220	200
2	50	229	233	209
3	75	179	180	166
4	100	224	232	207
5	125	230	239	212
6	150	237	241	213
7	175	174	167	155
8	200	236	234	209
9	250	184	184	171
10	300	249	252	224

Tabel 2. Hasil Deteksi Api menggunakan *Webcam*

Deteksi	Manual	Jarak (centimeter)	Rerata RGB			Jarak titik bobot	Hasil
			Red	Green	Blue		
1	Ada api	25	180	175	167	5,19	Ada api
2	Ada api	50	240	237	219	15,39	Ada api
3	Ada api	75	215	209	210	15,00	Ada api
4	Ada api	100	245	249	235	16,49	Ada api
5	Ada api	125	185	181	164	6,40	Ada api
6	Ada api	150	226	220	203	13,34	Ada api
7	Ada api	175	243	239	234	17,46	Ada api
8	Ada api	200	219	222	207	9,43	Ada api
9	Ada api	250	227	224	216	11,57	Ada api
10	Ada api	300	193	188	179	12,68	Ada api

Pada proses deteksi yang pertama yaitu pada jarak 25 centimeter, terdapat nilai rerata RGB masing-masing *Red* = 180, *Green* = 175, dan *Blue* = 167. Pencarian jarak bobot secara manual untuk membuktikan keluaran program dengan penerapan Persamaan (1). Pencarian jarak bobot untuk *database* pertama adalah sebagai berikut:

$$d = \sqrt{(R_1 - R)^2 + (G_1 - G)^2 + (B_1 - B)^2}$$

$$d = \sqrt{(213 - 180)^2 + (220 - 175)^2 + (200 - 167)^2}$$

$$d = \sqrt{(33)^2 + (45)^2 + (33)^2}$$

$$d = \sqrt{4203}$$

$$d = 64,83$$

Karena nilai $d > n$ maka pencarian jarak bobot akan dilanjutkan pada *database* kedua adalah sebagai berikut:

$$d = \sqrt{(R_2 - R)^2 + (G_2 - G)^2 + (B_2 - B)^2}$$

$$d = \sqrt{(229 - 180)^2 + (233 - 175)^2 + (209 - 167)^2}$$

$$d = \sqrt{(49)^2 + (58)^2 + (42)^2}$$

$$d = \sqrt{7529}$$

$$d = 86,76$$

Karena nilai d masih lebih besar dari n maka pencarian jarak bobot akan dilanjutkan pada *database* ketiga sebagai berikut:

$$d = \sqrt{(\overline{R}_3 - R)^2 + (\overline{G}_3 - G)^2 + (\overline{B}_3 - B)^2}$$

$$d = \sqrt{(179 - 180)^2 + (180 - 175)^2 + (166 - 167)^2}$$

$$d = \sqrt{(-1)^2 + (5)^2 + (-1)^2}$$

$$d = \sqrt{27}$$

$$d = 5,19$$

Karena nilai $d \leq n$ yaitu $5,19 < 20$, maka proses penghitungan jarak bobot dihentikan karena sudah mencapai nilai toleransi yang diinginkan. Program akan memberikan keputusan citra yang ditangkap merupakan api, demikian juga yang berlaku pada proses deteksi berikutnya. Apabila setelah dilakukan penghitungan pada seluruh *database* dan tidak ditemukan nilai $d \leq n$ maka program akan memberikan keputusan bahwa citra yang ditangkap bukan api.

Dari hasil pengujian di atas, antara pengamatan manual dengan hasil komputasi menunjukkan hasil yang sama, yaitu:

$$\frac{10}{10} \times 100\% = 100\%$$

Jadi hasil pengujian program dengan menggunakan 10 sampel api yang ada di *database* adalah 100 %.

4. SIMPULAN

Berdasarkan hasil dan pembahasan di atas dapat diambil simpulan bahwa tingkat keberhasilan sistem deteksi api yang dirancang adalah 100 % dapat mendeteksi ada dan tidaknya api pada variasi jarak api ketika *background*-nya homogen.

DAFTAR PUSTAKA

- [1] Jain, A.K., "Fundamentals of Digital Image Processing", Prentice Hall of India, 1989.
- [2] Kusumadewi, D.I., 2005, "Aplikasi Pengolahan Citra Menggunakan Metode Thresholding untuk Deteksi Kerusakan Telur", Skripsi S-1, Universitas Ahmad Dahlan, Yogyakarta.
- [3] Lalang, I.D., 2003, "Sistem Proteksi Bahaya Kebakaran Berdasarkan Monitoring Suhu", Tugas Akhir S-1, Universitas Islam Indonesia, Yogyakarta.
- [4] Schalkoff, R.J., "Digital Image Processing and Computer Vision", John Wiley & Sons, Inc., 1989.
- [5] Setyawardhana, dkk, 2006, "Robot Cerdas Pemadam Api menggunakan Kamera", Proceeding of Industrial Electronic Seminar EEPIS-ITS, Surabaya, Hal. 25-30