

# ***BRUTE FORCE STRING MATCHING*** **PADA SISTEM DETEKSI KEBAKARAN** **DAN** **PENEBAANGAN HUTAN LIAR** **MENGGUNAKAN PENCITRAAN SATELIT DAN *BASE64 STRING***

**Amalfi Yusri Darusman**

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung  
e-mail : [if17023@students.if.itb.ac.id](mailto:if17023@students.if.itb.ac.id)

## **ABSTRAK**

Permasalahan kebakaran hutan dan penebangan hutan liar sekarang ini dijadikan sorotan karena merupakan salah satu masalah yang vital bagi permasalahan lingkungan dunia. Meskipun permasalahan tersebut dapat diselesaikan dengan menggunakan teknik konvensional, namun kadang perlu pendekatan yang lebih mengedepankan sisi otomatisasi untuk penyelesaian masalah tersebut, misalnya dengan sistem deteksi kebakaran dan penebangan hutan liar. Pada makalah ini implementasi teknik tersebut menggunakan algoritma *string matching* untuk mencocokkan *string* yang dihasilkan dari *pattern* gambar. Gambar akan di-*convert* menjadi *base64 string* lalu dicocokkan dengan *database base64 string* dari gambar yang telah di-*convert* sebelumnya. Algoritma *string matching* yang digunakan adalah *brute force* yang telah dimodifikasi untuk mengatur tingkat ketelitian dari teknik implementasi ini.

**Kata kunci :** otomatisasi, sistem deteksi kebakaran hutan dan penebangan hutan liar, *string matching*, *brute force*, *base64 string*, *convert*.

## **1. PENDAHULUAN**

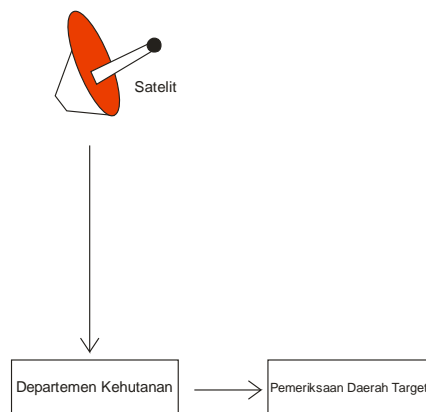
Kebakaran hutan dan penebangan hutan liar adalah salah satu masalah lingkungan yang serius, karena akibat yang ditimbulkan mempunyai imbas yang sangat besar, misalnya dengan berkurangnya luas hutan maka akibat dari *global warming* akan semakin terasa. Berbagai upaya dan sistem untuk menyelesaikan masalah ini sudah dilakukan dan dibangun, misalnya dengan penebangan

penebangan hutan, sistem patroli berkala di hutan, adanya hukum dan undang-undang untuk melindungi hutan, dan lain-lain. Namun ada kalanya upaya dan sistem tersebut meleset dari fungsinya sehingga masih terdapat kebocoran untuk permasalahan ini.

Lemahnya sistem konvensional untuk mendeteksi permasalahan ini yang menyebabkan timbulnya gagasan untuk menggunakan otomatisasi untuk meningkatkan kapasitas dan kemampuan sistem deteksi permasalahan ini. Diharapkan dengan adanya sistem otomatisasi tersebut sistem tidak akan lengah dan meleset untuk menjalankan fungsinya yang bersifat *real time*.

Teknologi satelit semakin banyak digunakan untuk berbagai kebutuhan manusia. Salah satunya adalah pencitraan bumi dari luar angkasa, untuk mencatat keadaan muka bumi dan menganalisisnya. Fungsi pencitraan tersebut dapat digunakan untuk sistem deteksi kebakaran dan penebangan hutan liar.

## **2. RANCANGAN SISTEM DETEKSI**

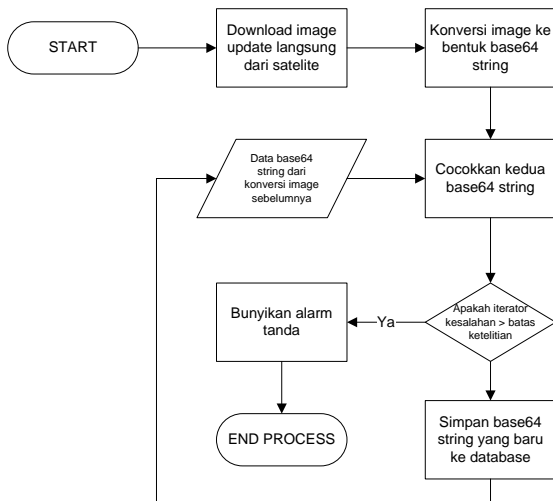


**Gambar 1.** rancangan sistem deteksi

Pada gambar di atas digambarkan rancangan sistem deteksi yang mungkin diimplementasikan, sistem akan diletakkan di Departemen Kehutanan yang terhubung langsung ke satelit. Sistem akan mengunduh data *image* secara berkala (tergantung pengaturan yang dilakukan) dan langsung melakukan konversi *image* ke *base64 string*. Setelah itu dilakukan pencocokan *string* antara data *base64 string* yang telah disimpan sebelumnya dengan hasil konversi *image*. Jika kesalahan pencocokan melebihi batas yang ditolerir maka alarm akan dibunyikan untuk member tanda, sedangkan jika kesalahan pencocokan tidak melebihi batas yang ditolerir maka hasil konversi akan disimpan ke dalam *database*. Jika alarm berbunyi maka akan dicek daerah yang dicitrakan oleh satelit apakah benar terdapat ancaman terhadap hutan atau hanya miscalculasi dari sistem.



Gambar 2. contoh gambar hasil pencitraan



Gambar 3. diagram proses sistem

### 3. BASE64 STRING

*Base64 string* adalah satu bentuk pengolahan string berbasis biner dengan representasi

karakter. *String* basis 64 digunakan pada komunikasi *dialup* antara sistem yang bekerja pada OS yang sama, misalnya Uuencode untuk UNIX, BinHex untuk TRS-80. *String* basis 64 menggunakan A-Z, a-z, dan 0-9 untuk 62 nilai pertama dan 2 nilai selanjutnya adalah tergantung dari sistem basis yang digunakan.

Contoh penggunaan string basis 64 adalah sebagai berikut :

*“Man is distinguished, not only by his reason, but by this singular passion from other animals, which is a lust of the mind, that by a perseverance of delight in the continued and indefatigable generation of knowledge, exceeds the short vehemence of any carnal pleasure.”*

Setelah dikonversi akan menjadi :

```

    "TWFuIG1zIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbm5IGJ5IGhpcyByZWZzb24sIGJ1dCBieSB0aGlzIHNoYm91dCBieSBvdGFyIGhpc3Npb24gZnJvbSBvdGhlc3Qgb2YgdGh1IG1pbmQsIHRoYXQgYnkYsBwZXXZlcmFuY2Ugb2YgZGVsaWdodCBpb29udGluZWVkaGFuZCBpbmRlZmF0aWdhYmx1IGdlbmVvYXRpb24gb2Yga25vd2xlZGdlLlCBleGN1ZWRzIHRoZSBzaG9ydCB2ZWwhbWVvY2Ugb2YgYW55IGNhc5hbCBwbGVhc3VyZS4="
  
```

Pada contoh di atas *Man* akan dikonversi menjadi *TWFu*. Pada format ASCII M,a,n akan disimpan dalam bentuk bytes 77, 97, 110, yaitu 01001101, 01100001, 01101110 dalam bilangan *base2*. Kemudian ketiga kelompok byte tersebut akan digabungkan menjadi 010011010110000101101110. Hasil gabungan tersebut akan dibagi menjadi 4 sehingga menghasilkan kelompok biner 010011, 010110, 000101, 101110 yang masing-masing merepresentasikan T,W,F, dan u.

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Gambar 4. gambar tabel karakter basis64

#### 4. KONVERSI IMAGE KE BASE64 STRING

Pada implementasi sistem ini akan digunakan *string* basis 64 untuk konversi *image*. Adapun *source code* untuk melakukan konversi tersebut adalah :

*Dalam Java*

```
public string ImageToBase64(Image image,
System.Drawing.Imaging.ImageFormat
format)
{
    using (MemoryStream ms = new
MemoryStream())
    {
        // Convert Image to byte[]
        image.Save(ms, format);
        byte[] imageBytes = ms.ToArray();

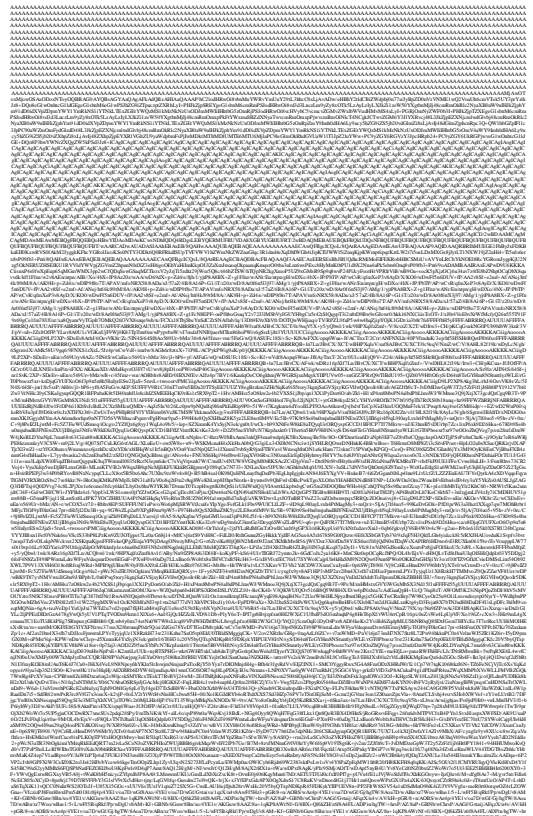
        // Convert byte[] to Base64 String
        string base64String =
Convert.ToBase64String(imageBytes);
        return base64String;
    }
}
```

Contoh implementasi kode tersebut pada gambar dibawah ini :



Gambar 5. contoh gambar yang akan dikonversi

Hasil konversi :



Pada contoh ini gambar yang digunakan untuk dikonversi tergolong sederhana, untuk gambar hasil pencitraan satelit pastinya akan lebih kompleks.

#### 5. ALGORITMA BRUTE FORCE YANG DIMODIFIKASI

Pada implementasi sistem ini terdapat modifikasi algoritma pencocokan string brute force untuk menghitung tingkat kesalahan dari pencocokan string.

*Dalam Java*

```
public static int sistemDeteksi (String
dariDatabase, String hasilKonversi){

    int m = dariDatabase.length();
    int n = hasilKonversi.length();
    int j = 0;
    int error = 0;
    for(int i = 0; I <= (n-m); i++){

        j = 0;
        if ((j < m) &&
(dariDatabase.charAt(i+j) ==
hasilKonversi.charAt(j))){
            j++;
        }
    }
}
```

```

else{
    error++;
}
}

return error;
}

```

Source code tersebut akan mengembalikan nilai jumlah *error* atau ketidaksamaan antara string yang diambil dari *database* dengan *string* hasil konversi. Semakin banyak jumlah *error* semakin besar kemungkinan alarm akan menyala.

## 6. PENCOCOKAN STRING DAN SISTEM DETEKSI

Dengan menggunakan algoritma *brute force* di atas maka akan didapatkan jumlah *error* yang terdapat dalam satu kali pencocokan.

Misalnya dicontohkan terdapat satu pencitraan satelit pada hutan di daerah X. Pencitraan pertama akan dikonversikan ke dalam bentuk *string* basis64, dan kemudian disimpan ke dalam *database*. Setelah selang waktu yang ditentukan, katakanlah tiap sepuluh menit sistem akan melakukan pengunduhan hasil pencitraan satelit dan langsung mengkonversinya menjadi *string* basis 64. Setelah itu, barulah dijalankan algoritma *brute force* untuk mengetahui jumlah *error* atau perubahan yang terdapat digambar. Jika jumlah *error* melebihi tingkat toleransi maka alarm akan berbunyi sedangkan jika tidak maka hasil pencitraan terakhir yang telah dikonversi akan disimpan ke *database* sebagai hasil pencitraan yang baru. Setelah selang sepuluh menit selanjutnya akan dilakukan proses yang sama hingga alarm berbunyi. Jika alarm berbunyi maka sistem harus di-*restart* dengan melakukan langkah pencitraan awal.

Jika ditulis secara prosedural proses yang terjadi adalah sebagai berikut :

1. Melakukan pengaturan terhadap tingkat toleransi *error* dan interval waktu pengunduhan sistem
2. Melakukan pengunduhan awal
3. Konversi gambar hasil pencitraan menjadi *string* basis64
4. Menyimpan hasil konversi ke basis data
5. Mengunduh hasil pencitraan selanjutnya setelah waktu interval

6. Melakukan *string matching* menggunakan algoritma *brute force* yang sudah dimodifikasi
7. Jika jumlah *error* melebihi angka toleransi maka alarm akan berbunyi. Jika jumlah *error* tidak melebihi maka kembali ke langkah nomor 5

## 7. KESIMPULAN

Penerapan teknik deteksi ini masih memiliki banyak kekurangan, yaitu :

1. Tingkat toleransi yang diberikan harus diturunkan secara matematis melalui berbagai penghitungan yang rumit untuk mendapatkan nilai toleransi yang paling ideal
2. Tingkat ketelitian interval sangat mempengaruhi. Jika interval sempit, maka komputasi yang dilakukan akan semakin banyak. Sedangkan jika interval lebar maka tingkat ketelitian akan semakin berkurang.
3. Daerah yang dicitrakan belum dispesifikkan nilai luasnya, sehingga efisiensi dan efektivitas sistem ini belum menjamin.
4. Sistem deteksi ini belum dapat menemukan titik tempat perubahan terjadi, hanya memberikan peringatan ancaman terhadap hutan.

Namun dibalik kekurangannya sistem ini mampu mendeteksi ancaman terhadap hutan yang bersifat universal, selama ada perubahan signifikan yang melebihi batas toleransi maka alarm akan berbunyi.

## 8. DAFTAR PUSTAKA

- [1] <http://www.dailycoding.com/Utils/Converter/ImageToBase64.aspx>, diakses pada 31 Desember 2009, 16.00 WIB
- [2] <http://en.wikipedia.org/wiki/Base64>, diakses pada 31 Desember 2009, 16.15 WIB
- [3] [http://www.dailycoding.com/Posts/convert\\_image\\_to\\_base64\\_string\\_and\\_base64\\_string\\_to\\_image.aspx](http://www.dailycoding.com/Posts/convert_image_to_base64_string_and_base64_string_to_image.aspx), diakses pada 31 Desember 2009, 16.00 WIB
- [4] Rinaldi Munir, "Diktat Kuliah Strategi Algoritmik", Program Studi Teknik Informatika ITB, 2006.