

# Content-Based Image Retrieval Menggunakan Metode Block Truncation Algorithm dan Grid Partitioning

Duman Care Khrisne <sup>1)</sup>, Mohamad David Yusanto <sup>2)</sup>

<sup>1</sup> Teknik Elektro UNIVERSITAS UDAYANA  
Denpasar, Bali, Indonesia  
E-mail : duman\_lx14@yahoo.com

<sup>2</sup> Teknik Informatika STMIK STIKOM INDONESIA  
Denpasar, Bali, Indonesia  
E-mail : david\_loe@yahoo.com

## Abstrak

Dampak *Image-Sharing* melalui Internet, menyebabkan semakin meningkatnya *volume* dari basis data yang berisikan data gambar di Internet. Sehingga dalam melakukan pencarian gambar tidak dapat dilakukan secara manual. Para ahli mengemukakan dua teknik dalam proses temu-kembali gambar, yaitu *Text-Based Image Retrieval* dan *Content-Based Image Retrieval*. Penggunaan *Content-Based Image Retrieval* digunakan untuk mengatasi kelemahan *Text-Based Image Retrieval* yang membutuhkan sumber daya manusia yang banyak saat mendeskripsikan sebuah gambar.

*Content-Based Image Retrieval* melakukan pencarian gambar berdasarkan pendekatan konten yang dimiliki gambar sendiri. Metode *Block Truncation Algorithm (BTA)* dan *Grid Partitioning* digunakan sebagai teknik ekstraksi ciri sebuah gambar uji, jarak *Euclidean* digunakan dalam pencocokan gambar uji terhadap gambar yang terdaftar dalam basis data. Hasil pengujian dinilai berdasarkan perbandingan antara *Recall* dan *Precision* yang disajikan dalam grafik analisis *ROC (Receiver Operating Characteristics)*, hasilnya gabungan kedua metode yang diusulkan memiliki nilai presisi 53% dan *recall* 11,8% dan dalam daerah analisis *ROC* dengan nilai 'Good.'

**Kata kunci:** *Content-Based Image Retrieval, Block Truncation Algorithm, Grid Partitioning, analisis ROC.*

## 1. Pendahuluan

Saat ini, setiap orang dengan mudah dapat mengunggah (*upload*) serta melakukan pertukaran foto atau gambar melalui aplikasi media sosial secara *online* kapanpun dan dimanapun. Semakin seringnya masyarakat meng-*upload* foto serta

gambar, menyebabkan semakin meningkatnya *volume* dari *database* yang berisikan data gambar di internet. Sehingga dalam melakukan pencarian gambar tidak dapat dilakukan secara manual. Para ahli mengemukakan dua teknik dalam proses temu-kembali gambar, yaitu *text-based image retrieval* dan *content-based image retrieval*.

Teknik pencarian gambar berdasarkan teks (*text-based image retrieval*) merupakan teknik yang telah digunakan untuk pencarian gambar setelah periode 1970 [1]. Teknik ini akan mencari sesuai dengan teks sebagai deskripsi dari gambar, dengan kata lain gambar terlebih dahulu dideskripsi dengan teks agar dapat dilakukan pencarian. Dalam mendeskripsikan sebuah gambar diperlukan sumber daya dalam jumlah besar, mengingat meningkatnya aktifitas mengunggah foto atau gambar melalui media sosial *online*. Namun, muncul sebuah kendala pada saat memberikan teks untuk mendeskripsikan gambar. Subyektifitas mempengaruhi seseorang dalam mendeskripsikan gambar berdasarkan perspektifnya. Sehingga tidak sedikit teks dari gambar tidak sesuai dengan deskripsi gambarnya [2]. Disamping itu jika kita menggunakan input yang berupa teks, maka kita harus mengetahui kata kunci yang benar-benar tepat agar gambar yang kita inginkan dapat ditampilkan. Sedangkan suatu gambar itu mempunyai *interest* yang sangat bervariasi.

Berdasarkan permasalahan di atas, dalam penelitian kali ini akan merancang sebuah aplikasi pencarian gambar dengan menggunakan teknik *content-based image retrieval* sebagai penunjang dari *text-based image retrieval*. Teknik *content-based* melakukan pencarian gambar berdasarkan konten yang terkandung dalam sebuah gambar, seperti bentuk, warna serta tekstur dari gambar. Dengan pencarian berdasarkan konten yang ada pada gambar dapat memperkecil kerancauan dalam proses pencarian gambar, meskipun kita merubah nama/judul gambar.

Dikarenakan pencarian tidak berdasarkan teks dari gambar, tapi berdasarkan bentuk, warna serta tekstur dari gambar.

## 2. Teori Penunjang

### 2.1 Content-Based Image Retrieval

*Content Based Image retrieval* (CBIR), teknik yang menggunakan fitur gambar (*visual contents*) dalam melakukan pencarian gambar dalam *database* gambar yang besar. Pada awal 1990-an perkembangan teknologi internet dan teknologi sensor *image* digital terjadi dengan sangat pesat. Sebagai hasilnya jumlah produksi gambar digital oleh para peneliti, akademisi, militer, industri, kesehatan dan para pengguna lainnya berkembang dengan sangat pesat. Sedangkan pencarian gambar berdasarkan teks sudah tidak mampu lagi secara optimal menyelesaikan permasalahan ini. Sehingga diperlukan suatu metode baru dalam melakukan *image retrieval*.

Pada tahun 1992, *National Science Foundation of the United States* mengadakan *workshop* mengenai pengembangan metode baru dalam *image database management system*. Dalam seminar itu dikemukakan metode baru yang lebih efektif dan presisi dalam merepresentasikan informasi dari fitur gambar yaitu berdasarkan properti atau fitur yang terkandung dalam gambar itu sendiri. Sejak tahun 1997 penelitian dan publikasi dibidang *content-based image retrieval* seperti ekstraksi fitur, *indexing*, *image database* manajemen berkembang dengan pesat, contohnya QBIC produk IBM, Virage produk Virage Inc, Netra produk Synapse dan beberapa teknik dan sistem mengenai aplikasi CBIR lainnya. [3]

### 2.2 Block Truncation Algorithm

*Block Truncation Algorithm* (BTA) adalah algoritma mendapatkan fitur warna dari sebuah citra berwarna, citra berwarna dibagi berdasarkan komponen penyusun warna R (*Red*), G (*Green*) dan B (*Blue*), rata-rata dari setiap komponen warna dijadikan patokan untuk memisah komponen warna menjadi dua H (*High*) dan L (*Low*) dimana H untuk *pixel-pixel* dalam citra yang memiliki nilai lebih tinggi dari rata-rata nilai *pixel* dalam suatu komponen warna dan L untuk *pixel-pixel* dalam citra yang memiliki nilai lebih rendah dari rata-rata nilai *pixel* dalam suatu komponen warna. Jadi warna dari sebuah gambar membentuk 6 kelompok RH, RL, GH, GL, BH dan BL.

Momen-momen dari kelompok inilah yang menjadi fitur warna dari BTA. Adapun langkah-langkah dalam metode ini sebagai berikut. [4]

1. Membagi gambar menjadi komponen warna merah, hijau dan biru.
2. Mencari nilai rata-rata di masing-masing komponen.
  - Rata-rata komponen merah
  - Rata-rata komponen hijau
  - Rata-rata komponen biru
3. Membagi setiap komponen gambar untuk mendapatkan nilai RH, RL, GH, GL, BH, BL dari gambar. RH didapatkan dengan hanya mengambil nilai komponen merah di semua piksel pada gambar yang di atas nilai rata-rata merah sedangkan RL hanya mengambil nilai komponen merah di semua piksel pada gambar yang di bawah nilai rata-rata merah. Demikian pula untuk mendapatkan nilai GH, GL, BH, dan BL.
4. Menerapkan *color moments* disetiap pembagian komponen yaitu RH, RL, GH, GL, BH, BL.
5. Menggunakan algoritma *clustering* untuk menemukan *cluster*.

### 2.3 Grid Partitioning

Segmentasi merupakan teknik untuk membagi suatu gambar menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut. Grid Partitioning merupakan metode sederhana untuk melakukan segmentasi gambar dengan menggunakan *grid* (kotak) persegi panjang yang sifatnya *fixed-size* serta *slides over* (dapat tumpang tindih) pada gambar. Untuk setiap *grid* persegi panjang dari gambar, fitur vektornya akan diekstrak. Ukuran dari persegi panjang dapat bervariasi untuk membuat versi multi skala dari *grid partitioning*. Penggabungan antara *overlapping* dan multi skala memungkinkan untuk mengatasi perubahan posisi dan perubahan skala dari suatu gambar. Dengan menggunakan *grid* mendapatkan beberapa keuntungan. Kinerja dari *grid* persegi panjang lebih baik daripada metode *Region Segmentation*. Selain itu, terjadi penurunan yang signifikan dalam waktu komputasi yang diperlukan untuk segmentasi gambar. [5]

### 2.4 Eucliden Distance

Dalam matematika, *Euclidean Distance* adalah jarak diantara dua buah obyek atau titik. *Euclidean Distance* dapat digunakan untuk mengukur kemiripan (*matching*) sebuah obyek dengan obyek yang lain. Jarak *Euclidean* dapat dianggap sebagai jarak yang paling pendek antar dua poin poin, maka dari itu dalam proyek akhir ini digunakan fungsi jarak *Euclidean* dan pada dasarnya sama halnya dengan persamaan Pythagoras ketika digunakan di dalam dua dimensi. Secara matematis dapat dituliskan di dalam persamaan berikut. [6]

$$d(i, j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 \dots + |x_{ip} - x_{jp}|^2}$$

Ket :

$d(i, j)$  = nilai jarak

$x_i$  = nilai fitur 1

$x_j$  = nilai fitur 2

## 2.5 Ekstraksi Fitur Warna

*Color Moment* merupakan ukuran yang dapat digunakan untuk membedakan gambar berdasarkan warna. Basis dari *Color Moment* terletak pada asumsi bahwa distribusi warna pada gambar dapat diartikan sebagai distribusi probabilitas. Distribusi probabilitas ditandai dengan sejumlah momen unik (misalnya distribusi normal dibedakan berdasarkan nilai rata-rata (*mean*) dan varians. Jika warna sebuah gambar mengikuti distribusi probabilitas tertentu, momen dari distribusi tersebut kemudian dapat digunakan sebagai fitur untuk mengidentifikasi gambar berdasarkan warna. Stricker dan Orengo (1995) menggunakan tiga momen pusat distribusi warna gambar dimana  $p_{ij}^k$  merupakan nilai komponen warna  $k$  dari piksel gambar  $ij$  dengan  $P$  adalah tinggi gambar serta  $Q$  merupakan lebar gambar. tiga momen tersebut adalah *Mean* (Rata-rata), Standar Deviasi dan *Skewness*. [7]

a. MOMENT 1 – Mean

$$E_k = \frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q p_{ij}^k$$

(1)

*Mean* dapat diartikan sebagai nilai rata-rata warna gambar.

b. MOMENT 2 – Standar Deviasi

$$SD_k = \sqrt{\frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (p_{ij}^k - E_k)^2}$$

(2)

Standar Deviasi merupakan akar pangkat dua dari varian sebuah distribusi.

c. MOMENT 3 – Skewness

$$S_k = \left( \frac{1}{PQ} \sum_{i=1}^P \sum_{j=1}^Q (p_{ij}^k - E_k)^3 \right)^{\frac{1}{3}}$$

(3)

*Skewness* dapat dipahami sebagai derajat asimetri suatu distribusi.

## 2.6 Precision, Recall dan Accuracy

Salah satu penerapan prinsip relevansi yang sejak dahulu digunakan dalam pengembangan sistem IR

(*Information Retrieval*) adalah penggunaan ukuran *recall and precision*. Menurut Lancaster yang dievaluasi kembali oleh David Power, efektifitas dari suatu sistem temu kembali informasi adalah kemampuan dari sistem itu untuk memanggil berbagai dokumen dan suatu basis data sesuai dengan permintaan pengguna. Ada dua hal penting yang biasanya digunakan dalam mengukur kemampuan suatu sistem temu kembali informasi yaitu rasio atau perbandingan dari perolehan (*recall*) dan ketepatan (*precision*) [8].

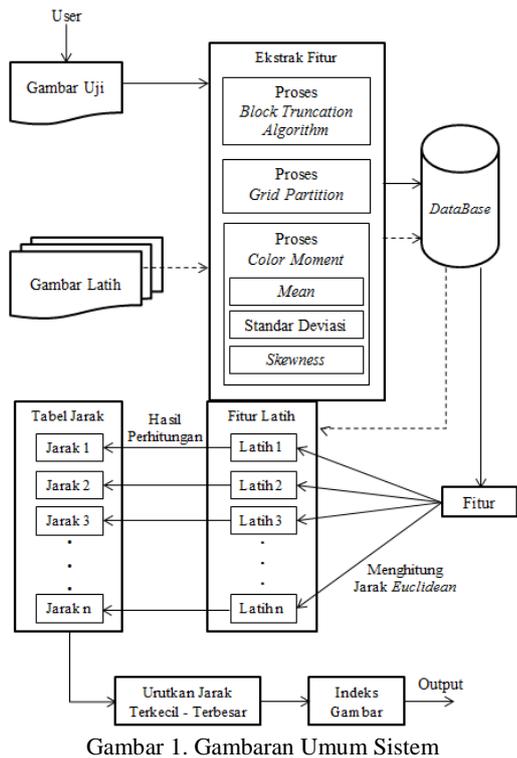
*Precision* dapat diartikan sebagai kepersisan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu). Jika seseorang mencari informasi di sebuah sistem, dan sistem menawarkan beberapa dokumen, maka kepersisan ini sebenarnya juga adalah relevansi. Artinya, seberapa persis atau cocok dokumen tersebut untuk keperluan pencari informasi, bergantung pada seberapa relevan dokumen tersebut bagi si pencari.

*Recall* adalah proporsi jumlah dokumen yang dapat ditemukan-kembali oleh sebuah proses pencarian di sistem IR. Rumusnya: Jumlah dokumen relevan yang ditemukan / Jumlah semua dokumen relevan di dalam koleksi. Lalu, *precision* adalah proporsi jumlah dokumen yang ditemukan dan dianggap relevan untuk kebutuhan si pencari informasi.

## 3. Gambaran Umum Sistem

Penjelasan dari Gambar 1 sebagai berikut.

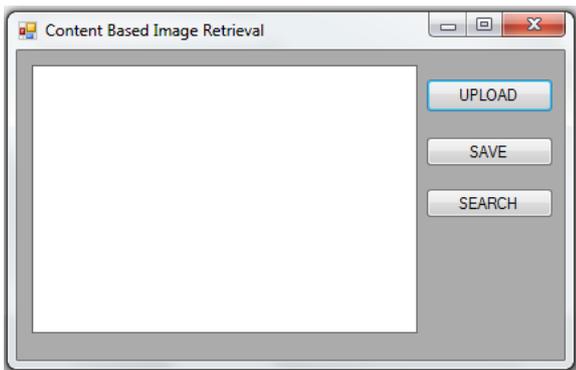
1. Masukkan (*input*) gambar latih yang digunakan sebagai data gambar dalam *database*.
2. Mengekstrak fitur warna dari gambar latih, kemudian hasil fitur tersebut disimpan ke dalam *database*.
3. Memasukkan gambar uji sebagai indikator dalam pencarian data.
4. Mengekstrak fitur warna dari gambar uji.
5. Melakukan proses perhitungan jarak (*euclidean distance*) antara hasil fitur dari gambar uji dengan fitur gambar latih.
6. Hasil perhitungan jarak akan dimasukkan dalam *database*.
7. Mengurutkan jarak dari yang terkecil sampai terbesar.
8. Melakukan proses indeks gambar berdasarkan urutan jarak dari yang terkecil sampai terbesar.
9. Hasil dari indeks gambar tersebut yang akan ditampilkan dalam *interface*.



Gambar 1. Gambaran Umum Sistem

#### 4. Implementasi Sistem

Rancangan program diimplementasikan menggunakan bahasa pemrograman Visual Studio C#. Dengan pengembangan berupa windows application. Form utama menampilkan form untuk melakukan upload gambar, save gambar dan ciri gambar, serta melakukan query gambar. User dapat melihat gambar pada picture box.



Gambar 2. Form Utama

Saat gambar sukses ter-upload proses ekstraksi ciri gambar menggunakan BTA dilakukan dengan membagi elemen warna sebuah gambar menjadi dua bagian dengan rata-rata nilai setiap elemen warna didapat dengan kode pada gambar 3.

```
byte* p = (byte*)(void*)Scan0;
byte red, green, blue;
int nOffset = stride - Img.Width * 3;
for (int y = 0; y < Img.Height; y++)
{
    for (int x = 0; x < Img.Width; x++)
    {
        blue = p[0];
        green = p[1];
        red = p[2];

        Br = Br + blue;
        Gr = Gr + green;
        Rr = Rr + red;

        p += 3;
    }
    p += nOffset;
}

Br = Math.Round(Br / size);
Gr = Math.Round(Gr / size);
Rr = Math.Round(Rr / size);
```

Gambar 3. Kode mencari nilai rata-rata elemen warna

Setelah mendapatkan nilai rata-rata sebagai pemisah, maka setiap elemen warna dipisahkan menjadi masing-masing dua elemen warna baru High dan Low untuk setiap elemen warna Red, Green dan Blue tahap ini diimplementasikan menggunakan kode pada gambar 4

```
p = (byte*)(void*)Scan0;
nOffset = stride - Img.Width * 3;
for (int y = 0; y < Img.Height; ++y)
{
    for (int x = 0; x < Img.Width; ++x)
    {
        blue = p[0];
        green = p[1];
        red = p[2];

        if (blue >= Br)
        {
            Bh.Add(blue);
        }
        else
        {
            Bl.Add(blue);
        }

        if (green >= Gr)
        {
            Gh.Add(green);
        }
        else
        {
            Gl.Add(green);
        }

        if (red >= Rr)
        {
            Rh.Add(red);
        }
        else
        {
            Rl.Add(red);
        }

        p += 3;
    }
    p += nOffset;
}
Img.UnlockBits(bmData);
```

Gambar 4. Kode pemisahan elemen warna menjadi high dan low

Setelah mendapatkan enam elemen warna, maka ciri dari sebuah gambar didapatkan dengan mengekstraksi Color Moment, yaitu mean, standar deviasi dan skewness dari keenam elemen warna

dalam sebuah gambar, Gambar 5 menunjukkan penggalan kode program untuk melakukan ekstraksi *moment mean* sebuah gambar.

Selain mencari ciri dari keseluruhan gambar ciri gambar setelah *Grid Partitioning* juga harus diekstraksi *Grid Partitioning* membagi gambar menjadi 3 bagian, yaitu atas, tengah dan bawah. Tiap bagian ini kemudian akan dilakukan proses *Color Moment* kembali. Pembagian dilakukan berdasarkan ukuran tinggi dari gambar, gambar dibagi menjadi 3 bagian yang memiliki ukuran yang sama.

```

for (int i = 0; i < Rl.Count; i++)
{
    byte point = Convert.ToByte(Rl[i]);
    sumRl = sumRl + point;
}
MRl = sumRl / size;

for (int i = 0; i < Rh.Count; i++)
{
    byte point = Convert.ToByte(Rh[i]);
    sumRh = sumRh + point;
}
MRh = sumRh / size;

for (int i = 0; i < Gl.Count; i++)
{
    byte point = Convert.ToByte(Gl[i]);
    sumGl = sumGl + point;
}
MGl = sumGl / size;

for (int i = 0; i < Gh.Count; i++)
{
    byte point = Convert.ToByte(Gh[i]);
    sumGh = sumGh + point;
}
MGh = sumGh / size;

for (int i = 0; i < Bl.Count; i++)
{
    byte point = Convert.ToByte(Bl[i]);
    sumBl = sumBl + point;
}
MBl = sumBl / size;

for (int i = 0; i < Bh.Count; i++)
{
    byte point = Convert.ToByte(Bh[i]);
    sumBh = sumBh + point;
}
MBh = sumBh / size;
    
```

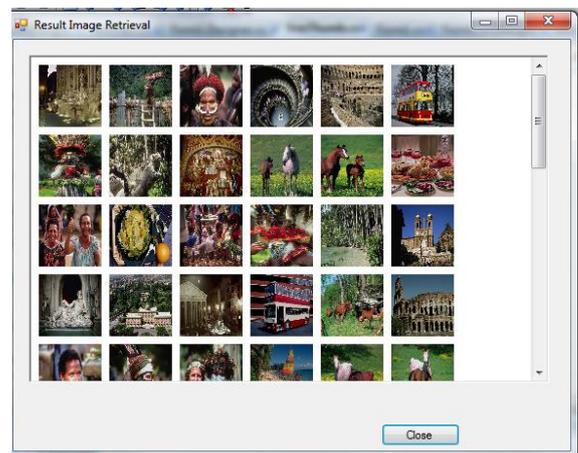
Gambar 5. Kode ekstaksi *moment mean*

Sampai tahap ini *user* diberi tiga buah pilihan, melakukan upload terhadap gambar yang berbeda, menyimpan gambar dan ciri gambar yang telah didapatkan dan yang paling penting untuk melakukan *query Content-Based Image Retrieval*.

*Query* dilakukan dengan membandingkan nilai-nilai ciri pada gambar uji dengan ciri yang dimiliki gambar pada basis data gambar (gambar latih). Dalam proses ini, dilakukan proses perhitungan jarak *euclidean* dengan 10 kali perbandingan yaitu :

1. Membandingkan ciri keseluruhan gambar uji dengan ciri keseluruhan dari setiap gambar latih. (keseluruhan – keseluruhan).
2. Membandingkan ciri bagian atas gambar uji dengan ciri bagian atas dari setiap gambar latih. (atas – atas).
3. Membandingkan ciri bagian atas gambar uji dengan ciri bagian tengah dari setiap gambar latih. (atas – tengah).

4. Membandingkan ciri bagian atas gambar uji dengan ciri bagian bawah dari setiap gambar latih. (atas – bawah).
5. Membandingkan ciri bagian tengah gambar uji dengan ciri bagian atas dari setiap gambar latih. (tengah – atas).
6. Membandingkan ciri bagian tengah gambar uji dengan ciri bagian tengah dari setiap gambar latih. (tengah – tengah).
7. Membandingkan ciri bagian tengah gambar uji dengan ciri bagian bawah dari setiap gambar latih. (tengah – bawah).
8. Membandingkan ciri bagian bawah gambar uji dengan ciri bagian atas dari setiap gambar latih. (bawah – atas).
9. Membandingkan ciri bagian bawah gambar uji dengan ciri bagian tengah dari setiap gambar latih. (bawah – tengah).
10. Membandingkan ciri bagian bawah gambar uji dengan ciri bagian bawah dari setiap gambar latih. (bawah – bawah).



Gambar 6. Hasil Pencarian Gambar

Setelah melakukan perbandingan, *query* menggunakan data uji menampilkan hasil pencarian gambar dengan mengurutkan gambar-gambar yang memiliki jarak rata-rata jarak *euclidean* paling kecil sampai dengan jarak paling besar. Jarak *euclidean* terjauh telah diatur sehingga tidak semua gambar akan dimunculkan dalam sebuah *query*. Gambar 6 menunjukkan sebuah hasil pencarian gambar, gambar hasil pencarian pada *form* ini hanya berupa *thumbnail*, jika salah satu *thumbnail* gambar dipilih akan menampilkan gambar dalam *form view image*, seperti tampak pada gambar 7.



Gambar 7. Form View Image

Form *View Image* akan menampilkan gambar dalam bentuk ukuran aslinya serta nama dari gambarnya.

### 5. Analisis Hasil Pengujian

Data yang digunakan dalam penelitian ini adalah gambar digital yang diambil dari <http://wang.ist.psu.edu/~jwang/test1.tar>. Dalam file test1.tar terdapat database gambar yang berisikan 1000 gambar digital. 700 data gambar digunakan untuk data latih dan 300 data gambar digunakan untuk data uji. Dari hasil pengujian menggunakan data uji didapatkan hasil sebagai berikut:

Tabel 1. Tabel Hasil Pengujian

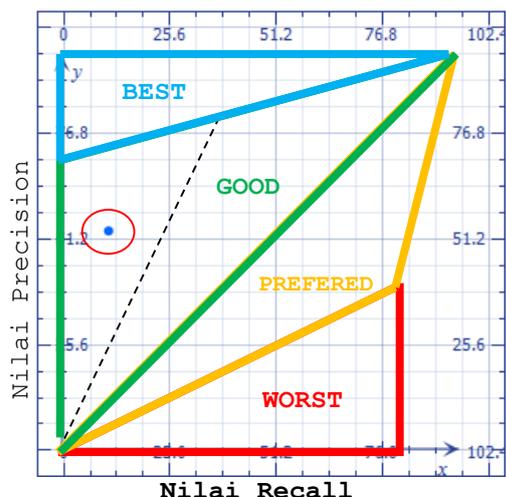
Gambar (Kelas)	Precision (%)	Recall (%)	Accuracy (%)
Orang Afrika dan Desa	53	12	89
Pantai	34	12	88
Gedung dan Bangunan	38	10	88
Bus	45	15	89
Dinosaur	100	22	92
Gajah	46	13	88
Bunga	53	12	90
Kuda	88	9	91
Gunung dan Glasier	15	5	87
Makanan	58	8	90
<b>AVERAGE</b>	<b>53</b>	<b>11,8</b>	<b>89,2</b>

Dari hasil pengujian pada Tabel 1 dapat dilihat bahwa didapatkan nilai rata-rata sebagai berikut.

- a. Precision : 53%
- b. Recall : 11,8%
- c. Accuracy : 89,2%

Dapat diartikan performa aplikasi ini adalah tingkat ketepatan dalam proses menemukan gambar berdasarkan pendekatan warna bernilai 53%, tingkat perolehan gambar yang relevan adalah 11,8%, dan tingkat keakuratan aplikasi ini bernilai 89.2%.

Dari hasil tersebut nilai *precision* serta *recall* ke dalam grafik analisis *ROC* [8] sebagai berikut.



Gambar 8. Grafik perbandingan nilai precision dan recall

Dari Gambar 8 nilai *precision* dan *recall* hasil pengujian pada penelitian ini masuk ke area analisis *ROC* bernilai **GOOD**.

### 6. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian sistem adalah sebagai berikut.

1. Dalam perancangannya, aplikasi temu kembali gambar (*Content-Based Image Retrieval*) dengan metode *Block Truncation Algorithm* (BTA) dan *Grid Partitioning* (GP) dilakukan dengan beberapa tahapan penelitian.
  - a. Memasukan gambar latih yang akan digunakan sebagai gambar *database*.
  - b. Mengekstraksi fitur dari gambar latih yang telah dimasukkan, kemudian akan disimpan pada *database*.
  - c. Memasukan gambar uji sebagai gambar *query* untuk pencarian gambar.
  - d. Mengekstraksi fitur dari gambar *query*.
  - e. Menghitung jarak *euclidean* antara gambar *query* dan gambar *database*.
  - f. Mengurutkan hasil perhitungan jarak *euclidean* dari yang rendah hingga yang tinggi.
  - g. Menampilkan hasil pengurutan tersebut dalam *interface* sistem.
2. Performa dari aplikasi berdasarkan nilai presisi, akurasi dan *recall* dapat disimpulkan sebagai berikut. Nilai rata-rata yang didapatkan masing-masing adalah :
  - a. Presisi = 53%
  - b. Akurasi = 89,2%
  - c. *Recall* = 11,8%.

Dapat diartikan performa aplikasi ini adalah tingkat ketepatan dalam proses menemukan gambar berdasarkan pendekatan warna bernilai 53%, tingkat perolehan gambar yang relevan adalah 11,8%, dan tingkat keakuratan aplikasi ini bernilai 89.2%.

3. Penggunaan metode *Grid Partitioning* (GP) [5] dalam proses optimalisasi hasil temu kembali gambar nilai presisi berhasil bertambah dari nilai 48,6% menjadi 53% dan nilai *recall* menjadi lebih kecil dengan dari nilai 47,4% menjadi 11,8% dibandingkan dengan penelitian sebelumnya [4].
4. Nilai tertinggi didapatkan pada kelas dinosaurus, sedangkan nilai terkecil didapatkan pada kelas gunung dan glasier. Perbedaan nilai disetiap gambar dikarenakan tingkat *foreground* dan *backgorund* gambar dalam setiap kelas yang berbeda. Semakin mudah dibedakan antara *foreground* dengan *backgorund*, maka hasil dari sebuah gambar semakin relevan, dan sebaliknya.
5. Terjadi hubungan tarik-ulur (*trade-off*) antara presisi dengan *recall*, dengan kondisi nilai presisi yang tinggi dan nilai *recall* yang rendah. Sehingga dapat diartikan aplikasi dapat mengembalikan gambar dan hampir semuanya relevan, tetapi sejumlah besar gambar relevan lain terabaikan.

## Referensi

- [1] Rahmanti, Farah Zakiyah., Entin, Martiana., Ramadijanti, Nana. 2010. **Pengelompokan Gambar Berdasarkan Warna dan Bentuk Menggunakan FGKA (Fast Genetic Kmeans Algorithm) Untuk Pencocokan Gambar.** Surabaya : Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh November.
- [2] Daud, Imam Abu. 2008. **Anotasi Automatis Citra Menggunakan Statitital Machine Translation Untuk Temu Kembali Citra.** Bogor : Institut Pertanian Bogor.
- [3] Alfatah , Edwin Ferdian. 2009. **Klasifikasi Ikan Dengan Menggunakan Hierarchical Clustering.** Surabaya : Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh November.
- [4] Silakari, Sanjay., Motwani, Mahesh., Maheshwari, Manish . 2009. **Color Image Clustering using Block Truncation Algorithm.** IJCSI International Journal of Computer Science Issues, Vol. 4, No. 2.
- [5] Pham, Trong-ton. 2006. **Automatic Image Annotation : Towards a Fusion of Region-based and Saliency-based Models.** Universite Pierre Et Marie Curie Master IAD.
- [6] Bagus, Bayu. 2007. **Image Database Menggunakan Sistem Content Based Image Retrieval Dengan Ekstraksi Fitur Terstruktur.** Surabaya : Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh November.
- [7] Maheshwari, Manish., Silakari, Sanjay., Motwani, Mahesh. 2012. **Content Based Color Image Clustering.** International Journal of Computer Applications (0975 – 8887) Vol 57, No.15.
- [8] Power, David M W, **Evaluation : From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.** Journal Of Machine Learning Technologies 2 (1) : 37-63, 2011

## Penulis

**Duman Care Khrisne, S.T., M.T,** Dosen di Jurusan Teknik Elektro Universitas Udayana, Bukit, Jimbaran-Bali.

**Mohamad David Yusanto,** Mahasiswa di STMIK STIKOM INDONESIA, Jln. Tukad Pakerisan 97, Denpasar-Bali.