Implementasi Algoritma Runut Balik dalam Pengenalan Citra Wajah pada Basis Data

Restu Arif Priyono / 13509020
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509020@std.stei.itb.ac.id

***Abstract*—Pencarian citra wajah yang tersimpan pada basis data dapat dilakukan dengan mengimplementasikan algoritma *backtracking* (runut balik) untuk menyesuaikan elemen yang terdapat pada suatu citra yang berada dalam basis data dengan fungsi pembatas berupa elemen yang terdapat pada citra wajah yang akan dicari. Basis data yang digunakan untuk menyimpan gambar wajah dibagi menjadi beberapa kategori dan upa kategori[1]. Kategori ini dibedakan berdasarkan suatu perhitungan rentang nilai tertentu. Diasumsikan bahwa setiap citra wajah memiliki karakteristik unik yang digunakan sebagai parameter yang dibutuhkan dalam pencarian citra wajah. Dengan menggunakan cara ini, suatu kategori mengarahkan kepada upa kategori lainnya yang pada akhirnya merujuk pada gambar yang sedang dicari.**

***Index Terms*— basis data, runut balik, pengenal wajah, pemrosesan gambar.**

I. PENDAHULUAN

Sistem pengenal wajah (*face recognition*) adalah sebuah pengaplikasian komputasi yang bekerja untuk mengidentifikasi sebuah citra digital, kemudian mencocokkannya dengan citra yang berada pada basis data.

Pengenalan wajah ini sudah digunakan secara luas pada bidang keamanan dan bidang-bidang lainnya. Penelitian mengenai algoritma pengenalan wajah ini juga sudah banyak dilakukan untuk meningkatkan efisiensi pencarian citra wajah ini. Sehingga, perkembangan algoritma dalam pencarian citra wajah ini meningkat secara signifikan beberapa dekade belakangan ini menjadi semakin baik[1].

Beberapa algoritma yang biasanya digunakan untuk menghadapi masalah pengenal wajah ini adalah *local binary pattern* (LBP) dan *sparse representation based classification* (SRC), atau gabungan antara keduanya[3] yang diperkenalkan oleh Rui Min dan Jean-Luc Dugelay yang terbukti memiliki kompleksitas algoritma $O(n^{1/2})$. Algoritma lainnya yang bisa digunakan untuk menyelesaikan pencarian citra wajah digital dengan citra digital yang berada di basis data adalah dengan menerapkan algoritma *divide and conquer*. Selain algoritma yang telah disebutkan, terdapat juga algoritma *backtracking* (runut balik) yang bisa digunakan untuk menyelesaikan pencarian citra wajah, seperti yang akan dijelaskan pada bagian selanjutnya.

Algoritma runut balik ini merupakan algoritma rekursif dan bisa juga iteratif, yang memeriksa kandidat solusi, apakah kandidat tersebut tidak melanggar fungsi pembatas, karena suatu pencarian akan terhadap suatu cabang akan dihentikan, dan kembali untuk mencari di cabang selanjutnya segera setelah suatu elemen atau *node* tidak memenuhi fungi pembatas.

Ketika mencari suatu citra wajah, terdapat beberapa parameter yang dijadikan fungsi pembatas, seperti jarak antara pupil mata kiri dengan pupil mata kanan, jarak antara pupil dengan ujung hidung, ukuran bibir, jarak antara dua tulang pipi, dan sebagainya[1].

Penjelasan pada makalah ini akan difokuskan kepada bagaimana parameter-parameter yang telah disebutkan dicocokkan dengan citra pada basis data yang juga menyimpan parameter-parameter tersebut. Bagaimana suatu citra dikompresi, diidentifikasi, atau cara mendapatkan parameter-parameter yang telah disebutkan sebelumnya tidak akan dibahas dalam makalah ini.

II. DASAR TEORI

a. Struktur Basis Data Penyimpanan Citra Wajah

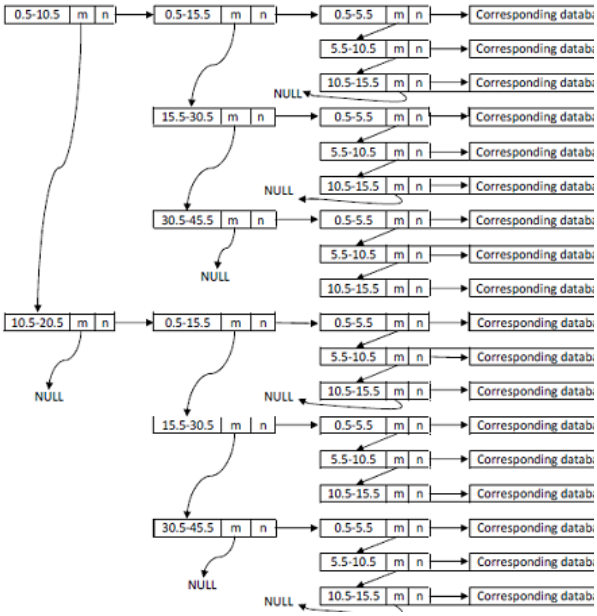
Sebelum melanjutkan pembahasan mengenai bagaimana suatu citra wajah dapat dicari dalam suatu basis data, terdapat beberapa hal yang perlu diperhatikan agar penjelasan selanjutnya lebih terarah.

setiap data citra yang terdapat dalam basis data memiliki elemen yang disimpan. Untuk menyederhanakan hal ini, dalam makalah ini akan diambil contoh elemen citra yang digunakan, yaitu:

a. Jarak antara pupil mata kiri dan pupil mata kanan,

b. Jarak antara pupil dengan ujung hidung,
c. Ukuran bibir.

Ketiga parameter di atas disimpan dalam tipe *number*. Asumsi yang digunakan adalah setiap orang memiliki nilai yang unik dari parameter-parameter tersebut. Kemudian, citra yang berada pada basis data dikelompokan sesuai dengan parameter-parameter ini sedemikian rupa sehingga minimal terdapat tiga kali pengelompokkan data citra. Sebagai gambaran, berikut ini adalah deskripsi struktur basis data yang digunakan.



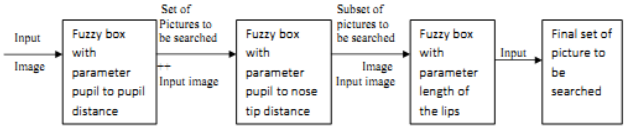
Pada gambar di atas, terdapat empat buah kelompok kotak, masing-masing kotak memiliki tiga buah bagian. Bagian pertama berisi interval angka yang menunjukkan parameter pertama. M menunjukkan *pointer* yang mengarahkan kepada parameter selanjutnya pada tingkat parameter yang sama (*pass* pertama). Kemudian n menunjukkan *pointer* yang mengarahkan kepada parameter yang lain dengan suatu interval tertentu (*pass* kedua), dan seterusnya hingga yang ditunjuk adalah data citra sesungguhnya pada basis data yang juga menyimpan informasi mengenai parameter-parameter yang dicari.

Namun sebelum suatu data citra dicocokkan dengan data citra yang berada di basis data, suatu mekanisme harus dilakukan untuk membuat data citra tersebut dapat dibandingkan. Hal ini dilakukan karena pada saat citra wajah seseorang diambil, akan terjadi perbedaan ukuran antara citra wajah yang baru diambil dengan data citra wajah yang telah disimpan di dalam basis data. Namun, pada dasarnya perbandingan jarak yang direpresentasikan dalam parameter yang telah kita bahas sebelumnya selalu sama. Untuk mencocokkan parameter ini, dibutuhkan rumus yang digunakan untuk mendapatkan nilai perbandingan ini. Misalkan panjang dari citra di dalam basis data dilambangkan dengan L, dan jarak antara pupil mata kiri dan kanan dilambangkan dengan P, sedangkan panjang citra wajah yang dideteksi adalah L1 dan jarak antara pupil kiri dan kanan dari citra wajah yang dideteksi adalah P1. Maka diperoleh hubungan sebagai berikut,

$$\frac{L}{L1} = \frac{P}{P1}$$

$$\mathrm{P} = P1 * \frac{L}{L1}$$

Sebagai penjelasan awal, berikut ini adalah gambaran dari pencocokkan citra wajah yang dideteksi dengan citra wajah yang terdapat di dalam basis data.



b. Algoritma Backtracking

a. Elemen Algoritma Backtracking

Agar suatu algoritma dapat dikatakan algoritma backtracking, terdapat beberapa elemen yang harus dimilikinya, yaitu sebagai berikut.

- Solusi Persoalan
  Solusi dinyatakan sebagai suatu vector dengan n-tuple[4]
  $X = (x_1, x_2, x_3, \ldots, x_n)$, dimana x_i adalah elemen dari S.

- Fungsi Pembangkit Nilai
  Misalkan terdapat suatu nilai xk, kemudian fungsi pembangkit nilai ini dinyatakan sebagai T(k). T(k) membangkitkan nilai untuk x_k, yang merupakan komponen dari vector solusi.

- Fungsi Pembatas
  Misalkan fungsi pembatas ini dinyatakan sebagai $B(x_1, x_2, x_3, \ldots, x_k)$
  Fungsi ini umumnya berupa pernyataan benar atau salah. Fungsi ini mengembalikan nilai benar ketika nilai dari x_i mengarah ke solusi. Jika bernilai benar, maka akan dilakukan pembangkitan nilai untuk x_{k+1}, dan jika mengembalikan nilai yang salah, maka nilai x_i yang bersangkutan tidak akan dimasukkan ke dalam perhitungan lagi.

b. Prinsip Kerja Algoritma Backtracking
Algoritma backtracking sebenarnya merupakan perluasan dari algoritma DFS (aturan pencarian

mendalam), sehingga cara kerja algoritma ini dapat digambarkan dengan suatu pohon dengan lintasan yang diambil dari akar ke daun. Terdapat beberapa terminology yang hendaknya diperhatikan dalam menggunakan algoritma ini, yaitu simpul hidup yang merupakan simpul-simpul yang sudah dibangkitkan, simpul-E atau *expand node* yang merupakan simpul hidup yang sedang diperluas. Simpul-E yang sedang diperluas mengakibatkan bertambah panjangnya lintasan. Simpul-E inilah yang akan "dibunuh" atau tidak diperhitungkan lagi ketika suatu lintasan yang sedang dibentuk tidak mengarah ke solusi. Fungsi yang digunakan untuk membunuh simpul-E ini adalah fungsi pembatas. Jika pembentukkan lintasan berujung pada simpul yang mati, maka proses pencarian selanjutnya adalah membangkitkan simpul anak yang masih belum diperiksa. Ketika seluruh anak tidak mengarah ke solusi, pencarian akan melakukan backtracking / runut balik ke simpul orang tuanya, kemudian membangkitkan simpul anak yang lain. Simpul-simpul anak yang baru inilah yang menjadi simpul-E. Pencarian dihentikan ketika solusi telah ditemukan atau tidak ada lagi simpul hidup yang bisa dibangkitkan untuk melakukan runut balik.

c. Pseudocode Algoritma Runut Balik
Algoritma Backtracking dapat dilakukan dengan dua cara, yaitu dengan cara rekursif atau cara iteratif.

Berikut ini adalah pseudocode skema umum algoritma runut balik.

Versi rekursif:

```
Procedure BacktrackRekursif(input k : integer)
  {
  prosedur yang digunakan untuk mencari semua solusi persoalan dengan cara backtracking skema rekursif
  Masukan: k yang bertipe integer, yaitu komponen vektor solusi, x[k]
  Keluaran: solusi x = (x[1], x[2], ..., x[n])
  }

Algoritma
  for each x[k] yang belum dicoba sedemikian rupa sehingga (x[k] ← T[k]) and B (x[1], x[2], ..., x[k]) = true do
      if (x[1], x[2], ..., x[k]) adalah lintasan dari akar ke daun then
          cetakSolusi(x) {basis}
      end if
      BackTrackRekursif(k+1) {backtrack, untuk x[k+1]}
end for
```

Sedangkan untuk skema umum algoritma runut balik versi iteratif adalah sebagai berikut:

```
procedure BacktrackIter(input k : integer)
{
  prosedur yang digunakan untuk mencari semua solusi persoalan dengan cara backtracking skema iteratif
  Masukan: k yang bertipe integer, yaitu komponen vektor solusi, x[k]
  Keluaran: solusi x = (x[1], x[2], ..., x[n])
}

Deklarasi
  n : integer

Algoritma
  n ← 1
  while n > 0 do
      if (x[n] belum dicoba sedemikian sehingga x[k] ← T(k)) and (B(x[1], x[2], ..., x[n])) then
          if (x[1], x[2], ..., x[n]) adalah lintasan dari akar ke daun then
              cetakSolusi(x)
          end if
          k ← k + 1 {indeks anggota tuple berikutnya}
      else {x[1], x[2], ..., x[n] tidak mengarah pada simpul solusi}
          n ← n - 1 {backtracking ke simpul sebelumnya}
      end if
  end while
{n = 0}
```

III. IMPLEMENTASI ALGORITMA RUNUT BALIK DALAM PENCARIAN CITRA WAJAH PADA BASIS DATA

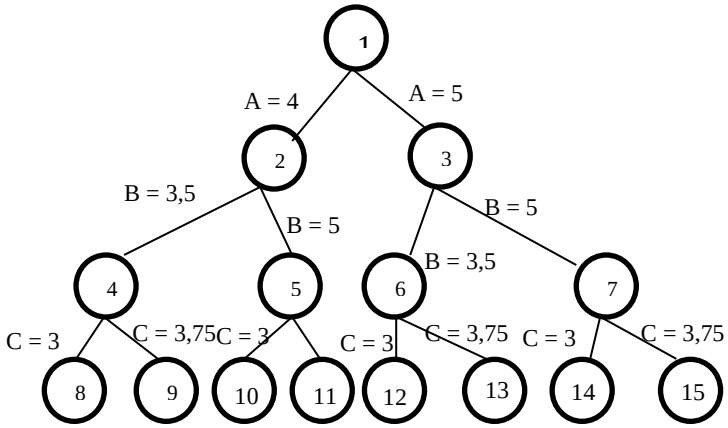
Algoritma runut balik cukup sederhana untuk diterapkan dalam pencocokkan citra wajah pada basis data. Berikut ini adalah pengaplikasian parameter yang digunakan untuk mencocokkan citra dengan elemen dari algoritma runut balik:

- Solusi Persoalan
  Misalkan solusi persoalan dilambangkan dengan X. maka X = {jarak antar pupil = 5 cm; jarak pupil dengan ujung hidung = 3,5 cm; ukuran bibir = 3,75 cm}.

- Fungsi Pembangkit Nilai
  Fungsi pembangkit nilai ini yang akan meneruskan ke pengujian pada simpul berikutnya selama tidak melanggar batasan.

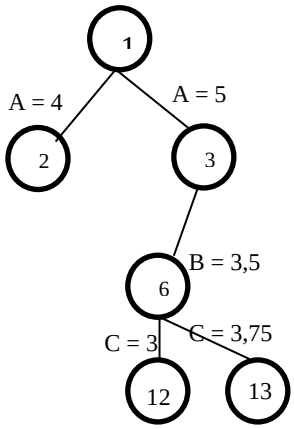
- Fungsi Pembatas
  Fungsi pembatas pada kasus ini adalah ketika

jarak antar pupil tidak sama dengan 5 cm, atau jarak pupil dengan ujung hidung tidak sama dengan 3, 5 cm, atau ukuran bibir tidak sama dengan 3,75 cm.



Misalkan A adalah jarak antara pupil kiri dan kanan, B adalah jarak pupil dengan ujung hidung, dan C adalah ukuran bibir. Simpul 8, 9, 10, 11, 12, 13, 14, dan 15 adalah data citra yang disimpan pada basis data.

Pada kasus di atas, kondisi simpul akhir yang dikunjungi untuk memperoleh hasil jarak antar pupil = 5 cm; jarak pupil dengan ujung hidung = 3,5 cm; ukuran bibir = 3,75 cm adalah:



Pertama-tama, pengujian dilakukan pada jarakan antara pupil kiri dan kanan. Ketika didapatkan jarak 4 cm, fungsi pembatas diaktifkan karena tidak sesuai dengan nilai yang diminta sehingga simpul baru dalam jalur ini tidak diaktifkan, tetapi akan dilakukan runut balik kembali ke simpul 1. Setelah itu, fungsi pembangkit nilai mengarahkan pada simpul 3. Nilai A tidak melanggar nilai jarak antar pupil, sehingga dilanjutkan dengan simpul ke-6 yang memiliki jarak antara pupil dan ujung hidung adalah 3,5 cm yang juga tidak mengaktifkan fungsi pembatas, sehingga mengaktifkan fungsi pembangkit nilai menuju simpul 12. Namun ukuran bibir yang ditinjau tidak sesuai, maka harus diadakan runut balik ke simpul 6. Kemudian fungsi pembangkit nilai mengarahkan ke simpul 13 yang merupakan citra yang sesuai dengan parameter yang diminta oleh masukan.
Jalur yang ditempuh adalah sebagai berikut:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 12 \rightarrow 13$$

IV. ANALISIS IMPLEMENTASI ALGORITMA

Jika kita melihat algoritma runut balik yang digunakan dengan versi rekursif, setiap simpul pada pohon ruang status akan dipanggil secara rekursif[4]. Jika jumlah seluruh simpul dalam pohon ruang status adalah 2^n maka untuk kasus terburuk, membutuhkan waktu dalam

$$O(a(n)\ 2^n)$$

atau jika jumlah simpul dalam pohon ruang status adalah n!, maka untuk kasus terburuk dari penggunaan algoritma runut balik ini membutuhkan waktu

$$O(b(n)n!)$$

dimana a(n) dan b (n) adalah suastu bilangan dengan derajat n yang menyatakan penghitungan waktu setiap simpul.

Namun perlu diperhatikan juga bahwa penggunaan algoritma lain, seperti algoritma brute force yang menelusuri setiap kemungkinan akan menghasilkan waktu yang lebih cepat dibandingkan dengan algoritma runut balik, ketika citra yang dicari dalam basis data terletak pada awal pencarian. Kasus terbaik yang dapat dihasilkan dari algoritma brute force adalah satu langkah, yaitu ketika citra yang pertama kali diuji adalah citra yang dicari. Sedangkan algoritma runut balik membutuhkan setidaknya tiga langkah (dalam kasus ini) untuk kasus terbaik yang bisa didapatkan.

Algoritma runut balik mulai akan dirasakan manfaatnya ketika citra yang akan dicari berada di tempat yang benar-benar acak di tengah basis data atau berada di akhir basis data. Misalkan kita menggunakan pohon ruang status yang telah dibuat sebelumnya. Untuk mencapai simpul 13 dengan cara brute force, dibutuhkan 7 langkah, sedangkan dengan menggunakan algoritma runut balik, hanya diperlukan 6 langkah. Untuk jumlah yang sedikit memang tidak akan terlalu signifikan terlihat perbedaannya, namun akan jauh lebih terlihat ketika jumlah data yang dibandingkan sangat besar.

V. KESIMPULAN DAN SARAN

Kesimpulan

Algoritma runut balik dapat dimanfaatkan untuk menyelesaikan banyak permasalahan yang berkaitan

dengan pencarian. Salah satu yang bisa contoh penerapannya adalah pada pencarian citra wajah pada basis data.

Saran

- Algoritma runut balik dapat menjadi alat yang membantu menyelesaikan masalah, terutama dalam hal pencarian jika kita menggunakannya dengan benar. Salah satu cara yang dapat dilakukan untuk menjadikan algoritma ini sangat bermanfaat adalah dengan mencari fungsi pembatas yang benar-benar sesuai.

- Banyak perkembangan algoritma runut balik (pruning, dan sebagainya). Usahakan gunakan perkembangan algoritma ini untuk menyelesaikan berbagai masalah, dibandingkan dengan tetap memanfaatkan algoritma runut balik yang masih benar-benar asli.

REFERENCES

[1] Bagchi, Debjyoti, et. al. "An Improved Face Recognition Technique". International Journal of Engineering Science and Technology (IJEST).

[2] http://en.wikipedia.org/wiki/Facial_recognition_system diakses pada 28 November 2011, 23:27

[3] Min, Rui, et. al. "Improved Combination of LBP and Sparse Representation Based Classification (SRC) for Face Recognition". Department of Multimedia Communication, UERECOM, Sophia Antipolis, France.

[4] Munir, Rinaldi. "Diktat Kuliah IF3051 Strategi Algoritma". Sekolah Teknik Elektro dan Informatika. 2009. Hal.144-166

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2011

Restu Arif Priyono
13509020