KLASIFIKASI CITRA DENGAN POHON KEPUTUSAN

Kusrini[1] Sri Hartati[2] Retantyo Wardoyo[3] Agus Harjoko[4]
[1,2,3,4]Program Studi Ilmu Komputer, Jurusan Ilmu MIPA, Universitas Gadjah Mada
[1]STMIK AMIKOM Yogyakarta
Email: [1]kusrini@amikom.ac.id, [2]shartati@ugm.ac.id, [3]rw@ugm.ac.id, [4]aharjoko@ugm.ac.id

ABSTRACT
Image classification can be done by using attribute of text that come along with the image, such as file name, size, or creator. Image classification also can be done base on visual content of the image. In this research, we implement a image classification model base on image visual content. The image classification is based on decision tree method that adapt C4.5 algorithm. The decision variable used in the decision tree generation process is image visual features, i.e. color moment order-1, color moment order-2, color moment order-3, entropy, energy, contrast, and homogeneity. The result of this research is an application that can classified image base on the knowledge of the previous classification cases.

Keywords: *image classification, decision tree, C4.5 algorithm*

ABSTRAK
Klasifikasi citra dapat dilakukan dengan menggunakan atribut teks yang menyertai citra seperti nama file, ukuran file atau pembuat file. Selain itu klasifikasi citra juga dapat dilakukan berdasarkan isi visual dari citra. Dalam penelitian ini, kami mengimplementasikan model klasifikasi citra berbasis isi visual citra. Metode yang akan digunakan adalah metode pohon keputusan dengan menggunakan algoritma C4.5. Variabel penentu yang digunakan dalam proses pembentukan pohon keputusan adalah fitur visual citra, yaitu moment warna order 1, moment warna order 2, moment warna order 3, entropi, energi, kontras dan homogenitas. Hasil dari penelitian ini adalah sebuah aplikasi yang dapat menentukan klasfikasi citra berdasarkan dengan pengetahuan yang terbentuk dari kasus-kasus klasifikasi sebelumnya.

Kata Kunci: *klasifikasi citra, pohon keputusan, algoritma C4.5*

Klasifikasi citra dapat dilakukan dengan menggunakan atribut teks yang menyertai citra seperti nama file, ukuran file atau pembuat file. Klasifikasi dengan cara ini sangat tergantung pada kepiawaian user dalam mendeskripsikan citra. Untuk menghindari subjektivitas pengguna dalam mendeskripsikan citra, klasifikasi dapat dilakukan dengan menggunakan isi visual citra.

Ada banyak metode dalam melakukan klasifikasi diantaranya adalah dengan menggunakan *k-nearest neighbour* dan pohon keputusan. Pohon keputusan merupakan salah satu model yang dapat digunakan dalam melakukan klasifikasi. Peneliti sudah melakukan penelitian untuk membangun pohon keputusan dengan variabel penentu berupa teks. Penelitian tersebut diterapkan untuk menganalisis kemungkinan pengunduran diri calon mahasiswa baru di STMIK AMIKOM Yogyakarta. Algoritma yang digunakan dalam penelitian tersebut adalah algoritma C4.5 [1, 2].

Dalam penelitian ini, diterapkan metode klasifikasi dengan menggunakan pohon keputusan untuk mengklasifikasi citra dengan variabel penentu berupa fitur visual citra. Peneliti mengimplementasikan algoritma C4.5 dengan menggunakan Bahasa pemrograman Borland Delphi dan software pengelola database (DBMS) Interbase. Integrasi algoritma C4.5 dengan DBMS telah diteliti oleh Moertini, dkk.

CITRA DIGITAL

Citra digital adalah citra yang disimpan dalam format digital (dalam bentuk file). Citra digital dapat didefinisikan sebagai fungsi *f(x,y)*, yakni *x* dan *y* adalah koordinat spasial dan nilai *f(x,y)* adalah intensitas citra pada koordinat tersebut [3].

EKSTRAKSI FITUR CITRA

Sebuah citra dapat dikenali secara visual berdasarkan fitur-fiturnya. Beberapa fitur yang dapat diekstrak dari sebuah citra adalah warna, bentuk, dan tekstur [3].

Beberapa fitur yang dapat diekstrak berdasarkan warna adalah histogram warna dan momen warna. Histogram merupakan fitur warna yang paling banyak digunakan. Histogram warna sangat efektif mengkarakterisasikan distribusi global dari warna dalam sebuah citra.

Untuk mendefinisikan histogram warna, ruang warna dikuantifikasi dalam tingkatan yang diskret, misal citra 8 bit dideskritkan dalam tingkatan 0, 1, 2, 3, ..., 255. Tiap-tiap tingkat menjadi bin di histogram. Histogram warna ini kemudian dihitung berdasarkan jumah pixel yang memenuhi masing-masing tingkat [3].

Momen warna merupakan representasi yang padat dari fitur warna dalam mengkarakterisasikan warna citra. Informasi distribusi warna disusun dalam 3 urutan momen. Momen yang pertama (μ) mewakili rata-rata warna, momen yang kedua (σ) menggambarkan standar deviasi, dan momen berikutnya (θ) menggambarkan kecondongan dari warna. Tiga urutan momen (μ_c, σ_c, θ_c) diperoleh dari Persamaan (1), (2), dan (3) [4].

$$\mu_c = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}^{c} \tag{1}$$

$$\sigma_c = \left[\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(p_{ij}^{c} - \mu_c)^2\right]^{1/2} \tag{2}$$

$$\theta_c = \left[\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (p_{ij}^c - \mu_c)^3 \right]^{1/3} \quad (3)$$

dimana p_{ij}^c adalah nilai komponen warna *c* pada piksel warna baris ke *i* dan kolom *j* dari citra. Jarak *euclidean momen* warna dari 2 image ditemukan lebih efektif untuk menghitung kedekatan citra.

Tekstur adalah sifat-sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat-sifat tadi dapat berulang dalam daerah tersebut. Tekstur dapat pula diartikan sebagai keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital [5]. Suatu permukaan dikatakan memiliki informasi tekstur jika luasannya diperbesar tanpa mengubah skala. Oleh karena itu sifat permukaan hasil perluasan memiliki kemiripan dengan permukaan asalnya.

Informasi tekstur dapat digunakan untuk membedakan sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus tanpa terpengaruh dengan warna. Syarat terbentuknya tekstur [5] adalah: 1) adanya pola-pola primitif yang terdiri dari satu atau lebih piksel. Bentuk primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan, dan lain-lain yang merupakan elemen dasar dari sebuah bentuk. 2) pola-pola primitif yang muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat ditemukan karakteristik pengulangannya.

Informasi tekstur tidak dapat diperoleh dari histogram satu dimensi. Agar dapat menangkap keterkaitan atau ketergantungan lokasi nilai-nilai intensitas yang mempunyai arti penting dalam persepsi tekstur, diperlukan matriks keterkaitan dua dimensi yang sering disebut matriks pasangan intensitas (matriks intensitas *co-occurance*).

Matriks intensitas *co-occurance* adalah suatu matrik yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak dan arah tertentu dalam citra [5]. Matriks intensitas *co-occurance* $p(i_1, i_2)$ didefinisikan dengan dua langkah sederhana sebagai berikut [5]. Langkah pertama dengan menentukan jarak antara dua titik dalam arah vertikal dan horizontal (vektor *d=(dx, dy)*). Besarnya *dx* dan *dy* dinyatakan dalam piksel sebagai unit terkecil dalam citra digital. Langkah kedua dengan menghitung pasangan piksel yang mempunyai nilai intensitas i_1 dan i_2 dan berjarak d piksel dalam citra. Langkah ketiga dengan meletakkan hasil perhitungan setiap pasangan nilai intensitas pada matriks sesuai dengan koordinatnya, dimana absis untuk nilai intensitas i_1 dan ordinat untuk nilai intensitas i_2.

Fitur tekstur yang dapat diekstrak dari citra diantaranya adalah entropi, energi, kontras, dan homogenitas. Entropi adalah fitur yang digunakan untuk mengukur keteracakan dari distribusi intensitas [5]. Formula yang dapat dipakai untuk menghitung entropi ditunjukkan oleh Persamaan (4).

$$Entropi = -\sum_{i_1} \sum_{i_2} p(i_1, i_2) \log p(i_1, i_2) \quad (4)$$

Energi adalah fitur untuk mengukur konsentrasi pasangan intensitas pada matriks *co-occurance* [5]. Persamaan yang digunakan untuk menghitung energi adalah Persamaan (5) [5]. Nilai energi akan makin membesar bila pasangan piksel yang memenuhi syarat matriks intensitas *co-occurance* terkonsentrasi pada beberapa koordinat dan mengecil bila letaknya menyebar.

$$Energi = \sum_{i_1} \sum_{i_2} p^2(i_1, i_2) \quad (5)$$

Kontras adalah fitur yang digunakan untuk mengukur kekuatan perbedaan intensitas dalam citra [5]. Nilai kontras membesar jika variasi intensitas citra tinggi dan menurun bila variasi rendah. Persamaan yang digunakan untuk mengukur kontras suatu citra ditunjukkan pada Persamaan (6) [5].

$$Kontras = \sum_{i_1} \sum_{i_2} (i_1 - i_2)^2 p(i_1, i_2) \quad (6)$$

Homogenitas digunakan untuk mengukur kehomogenan variasi intensitas citra [5]. Nilai homogenitas akan semakin membesar bila variasi intensitas dalam citra mengecil. Homogenitas dihitung dengan Persamaan (7) [5].

$$Homogenitas = \sum_{i_1} \sum_{i_2} \frac{p(i_1, i_2)}{1 + |i_1 - i_2|} \quad (7)$$

Notasi p pada Persamaan (4), (5), (6), dan (7) melambangkan probabilitas yang bernilai nol hingga 1, yaitu nilai elemen dalam matriks *co-occurance*, sedangkan i_1 dan i_2 melambangkan pasangan intensitas yang berdekatan, yang dalam matriks *co-occurance* masing-masing menjadi nomor baris dan nomor kolom.

ALGORITMA C4.5

Algoritma C4.5 merupakan pengembangan dari algoritma ID3 [6]. Secara umum, algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut: (i) Pilih atribut sebagai *root*, (ii) Buat cabang untuk masing-masing nilai, (iii) Bagi kasus dalam cabang, dan (iv) Ulangi proses untuk masing-masing cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Pemilihan atribut sebagai *root* didasarkan pada nilai *gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *gain* digunakan Persamaan (8) [7].

$$Gain(S, A) = Entropi(S) - \sum_{i=1}^{n} \frac{|S_i|}{S} * Entropi(S_i) \quad (8)$$

dengan: S adalah notasi untuk himpunan kasus dan A untuk notasi atribut. Jumlah partisi atribut A dinyatakan pada n sedangkan jumlah kasus dalam S dan pada partisi ke i dinotasikan dengan $|S|$ dan $|S_i|$.

Sedangkan perhitungan nilai entropi dapat dilihat pada Persamaan (9) [7].

$$Entropi(S) = \sum_{i=1}^{n} -pi * \log_2 pi \quad (9)$$

dengan definisi n yang berbeda yaitu jumlah partisi atribut S. p_i akan menyimpan nilai proporsi dari S_i terhadap S.

Tabel 1: Struktur tabel D_Atribut

Kolom	Tipe	Keterangan
nama_atribut	varchar(30)	nama atribut
is_aktif	char(1)	status digunakan atau tidak dalam analisis
is_hasil	char(1)	status yang menunjukkan sifat atribut sebagai variabel penentu atau variabel hasil
Ket	varchar(255)	keterangan

Tabel 2: Struktur tabel Nilai_Atribut

Kolom	Tipe	Keterangan
nama_atribut	varchar(30)	nama atribut
nilai	varchar(10)	nilai kelas diskret
nilai_bawah	numeric(15,2)	nilai terkecil pada kelas kontinu
nilai_atas	numeric(15,2)	nilai terbesar pada kelas kontinu

PERANCANGAN SISTEM

Dalam penelitian ini, fitur-fitur yang dipilih sebagai variabel penentu dalam pengklasifikasian citra adalah momen warna order 1, momen warna order 2, momen warna order 3, entropi, energi, kontras, dan homogenitas. Fitur-fitur dipilih karena memiliki nilai tunggal untuk setiap citra. Sementara itu, fitur histogram warna sulit untuk diimplementasikan sebagai variabel penentu dalam pengklasifikasian citra karena nilainya berupa *array*.

Data Flow Diagram (DFD) yang menggambarkan alur data serta proses yang terjadi pada sistem ditunjukkan pada Gambar 1. Sistem melibatkan dua entitas luar yaitu analis dan pengguna. Analis akan melakukan proses pelatihan dan akan mendapatkan informasi daftar aturan yang terbentuk. Namun pengguna akan diberi hak untuk melakukan proses pengujian atau proses klasifikasi citra. Dari rancangan DFD pada Gambar 1 dibuat rancangan basis data yang sesuai. Tabel yang digunakan ada dua macam, yaitu tabel statis yang dibuat saat pembangunan aplikasi dan tabel dinamis yang dibuat ketika aplikasi sedang dijalankan. Terdapat empat tabel statis dan dua tabel dinamis yang digunakan pada eksperimen.

Tabel statis pertama adalah tabel D_Atribut digunakan untuk menyimpan daftar atribut dalam proses klasifikasi dengan algoritma C4.5. Struktur tabel D_Atribut ditunjukkan pada Tabel 1. Tabel kedua adalah Nilai_Atribut untuk menyimpan daftar nilai-nilai yang diijinkan dalam suatu atribut beserta klasifikasinya. Tabel ini berfungsi untuk mendiskritkan nilai kelas kontinu yang ada pada nilai-nilai variabel. Struktur Tabel Nilai_Atribut ditunjukkan pada Tabel 2. Tabel ketiga adalah tabel Kasus untuk menyimpan data kasus lama yang digunakan sebagai dasar pada proses *training*. Struktur Tabel Kasus ditunjukkan pada Tabel 3. Tabel statis terakhir adalah tabel *Tree* untuk menyimpan hasil training. Struktur Tabel *Tree* ditunjukkan pada Tabel 4.

Tabel 3: Struktur tabel Kasus

Kolom	Tipe	Keterangan
gambar	blob	data file gambar
nama_file	varchar(255)	nama file
n_momen_1	numeric(15,5)	nilai momen warna order 1
n_momen_2	numeric(15,5)	nilai momen warna order 2
n_momen_3	numeric(15,5)	nilai momen warna order 3
n_entropi	numeric(15,5)	nilai entropi
n_energi	numeric(15,5)	nilai energi
n_kontras	numeric(15,5)	nilai kontras
n_homogenitas	numeric(15,5)	nilai homogenitas
momen_1	varchar(10)	nilai kelas diskret dari momen warna order 1
momen_2	varchar(10)	nilai kelas diskret dari momen warna order 2
momen_3	varchar(10)	nilai kelas diskret dari momen warna order 3
entropi	varchar(10)	nilai kelas diskret dari entropi
energi	varchar(10)	nilai kelas diskret dari energi
kontras	varchar(10)	nilai kelas diskret dari kontras
homogenitas	varchar(10)	nilai kelas diskret dari homogenitas
klasifikasi	varchar(256)	klasifikasi
id_kasus	integer	id kasus

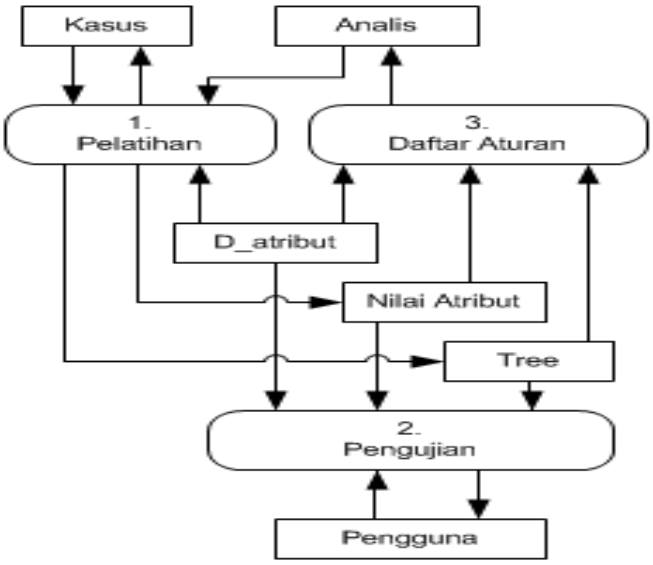
Tabel Kerja[i] dan Tabel Sub_Kerja[i] adalah tabel dinamis yang digunakan. Tabel Kerja[i], dengan i bernilai dari 0 sampai dengan tingkat kedalaman *tree* yang terbentuk. Tabel ini digunakan untuk menyimpan variabel dan nilai *gain* yang terpakai pada suatu level ke i *tree*. Struktur Tabel Kerja[i] ditunjukkan pada Tabel 5. Tabel Sub_Kerja[i], dengan i bernilai dari 0 sampai dengan tingkat kedalaman *tree* yang terbentuk. Tabel ini digunakan untuk menyimpan nilai variabel beserta nilai entropi pada suatu level ke i pada *tree*. Struktur Tabel Sub_Kerja[i] ditunjukkan pada Tabel 6.

Pembentukan tree dilakukan secara rekursif. Inisialisasi awal pembentukan *tree* ditunjukkan oleh Gambar 2. Salah satu langkah pada algoritma inisialisasi pembentukan *node* yang ditunjukkan oleh Gambar 2 adalah buat_-node.

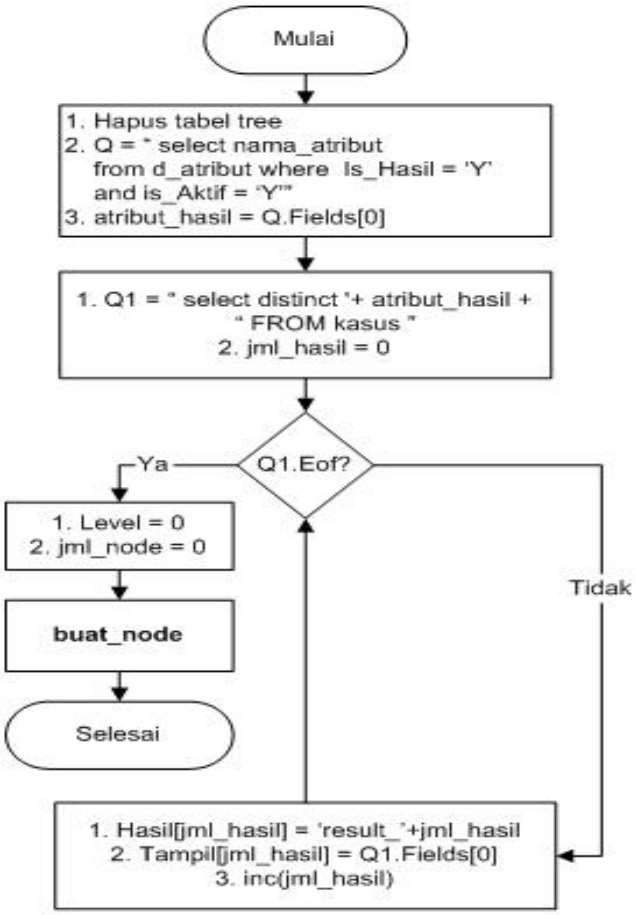
Algoritma prosedur buat_node ditunjukkan pada Gambar 3, Gambar 4, Gambar 5, Gambar 6, dan Gambar 7.

HASIL DAN PEMBAHASAN

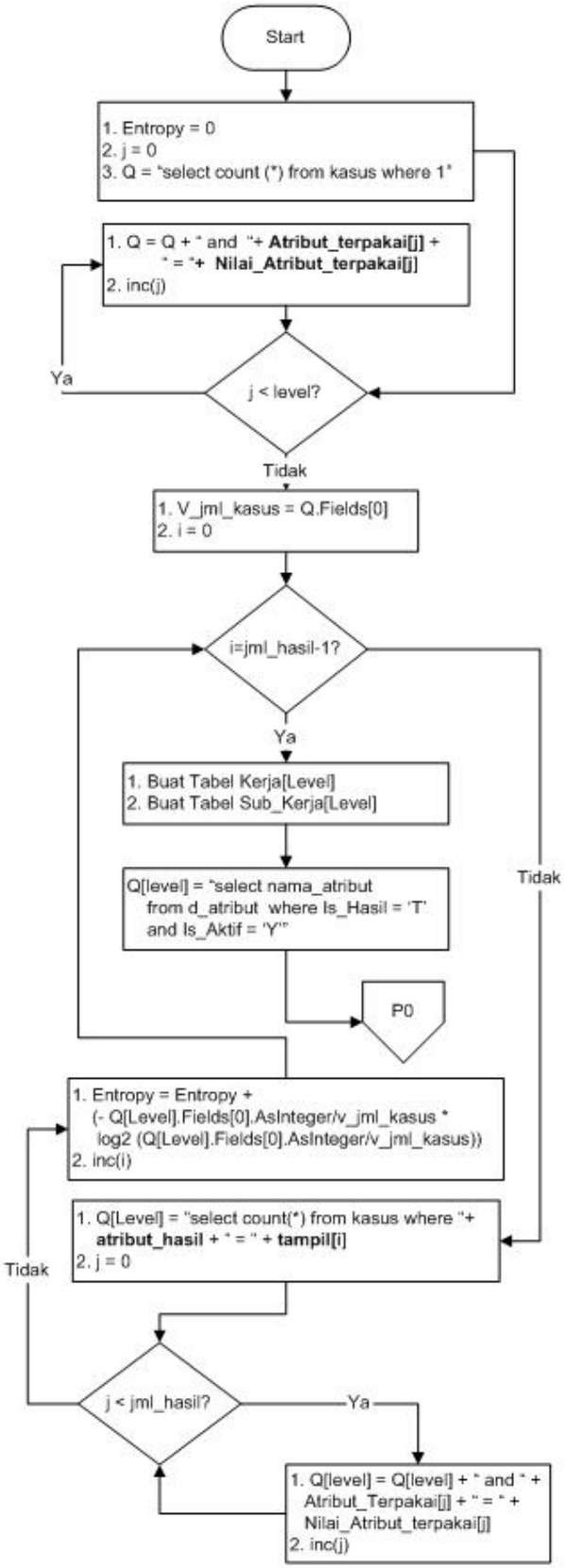
Hasil dari penelitian ini adalah aplikasi untuk mengklasifikasi citra baru berdasarkan hasil klasifikasi citra sebelumnya. Struktur menu aplikasi dan fasilitas pelatihan ditunjukkan pada Gambar 8 dan Gambar 9.



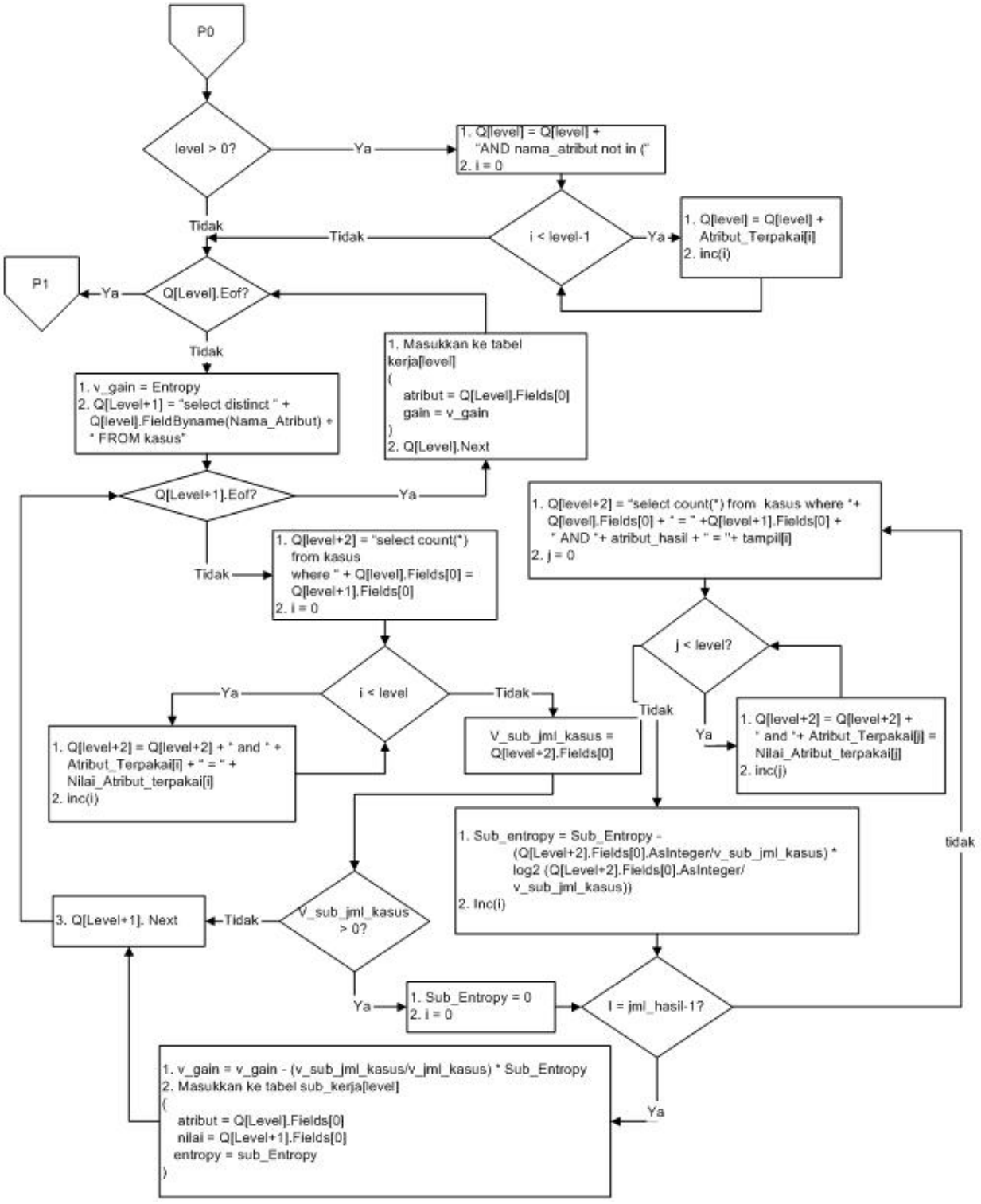
Gambar 1: *Data Flow Diagram*



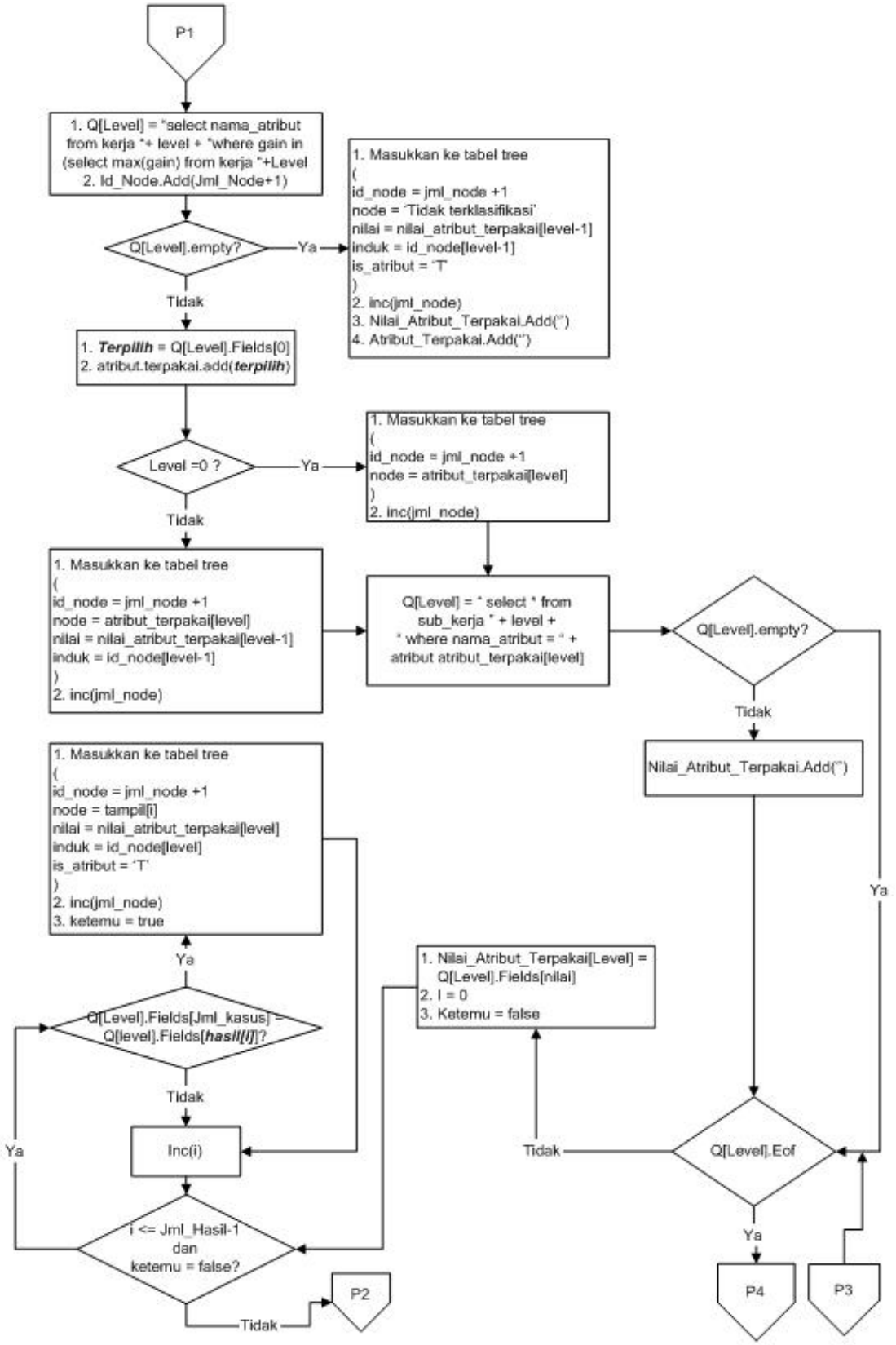
Gambar 2: Algoritma inisialisasi pembentukan *node*



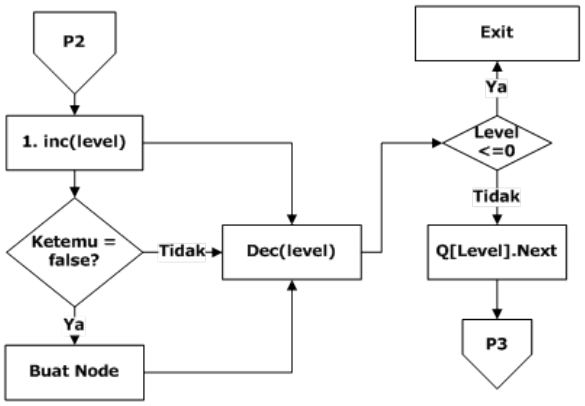
Gambar 3: Algoritma inisialisasi pembentukan *node*



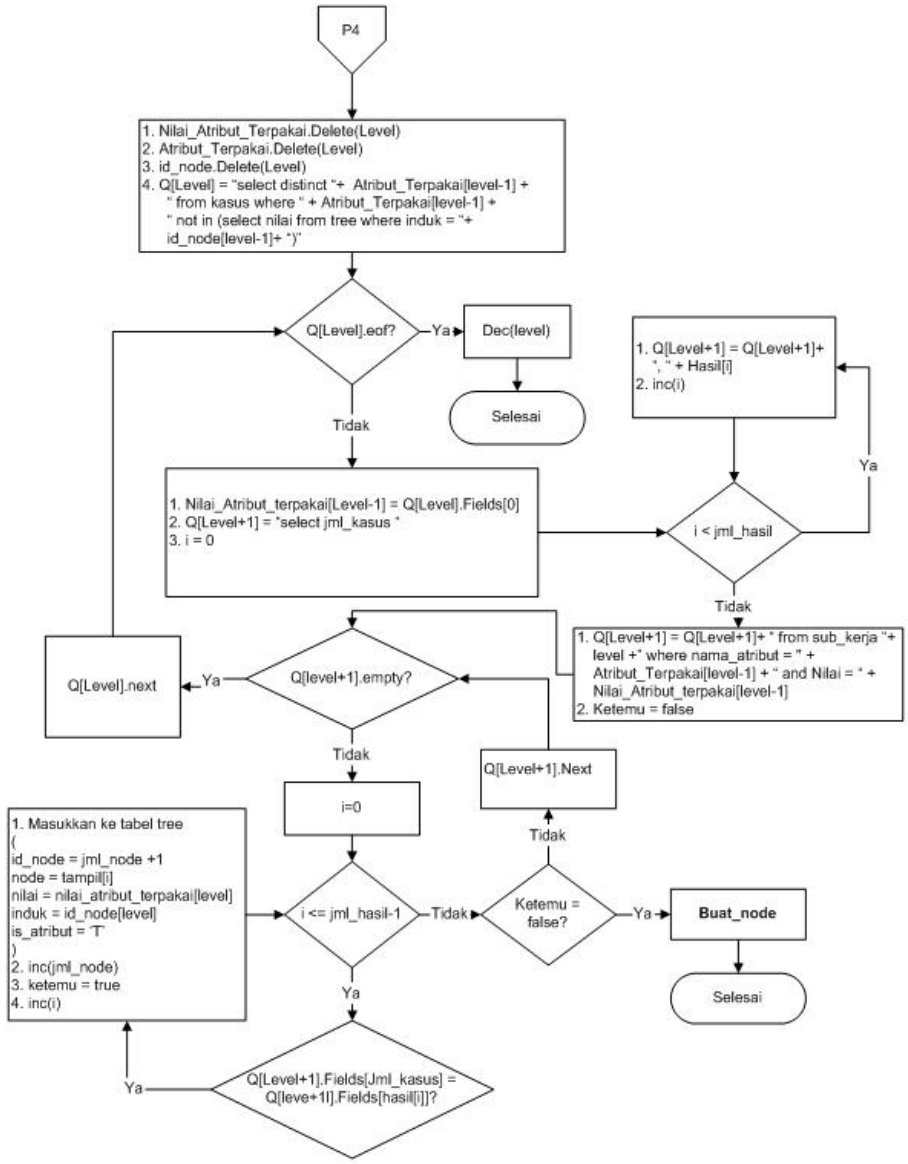
Gambar 4: Algoritma pembentukan *node* 2



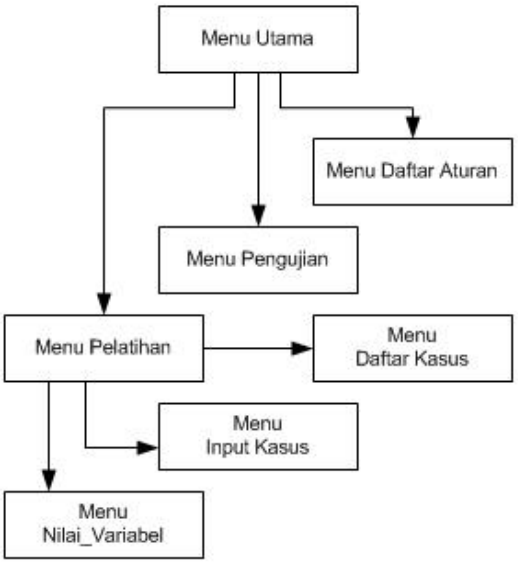
Gambar 5: Algoritma pembentukan *node* 3



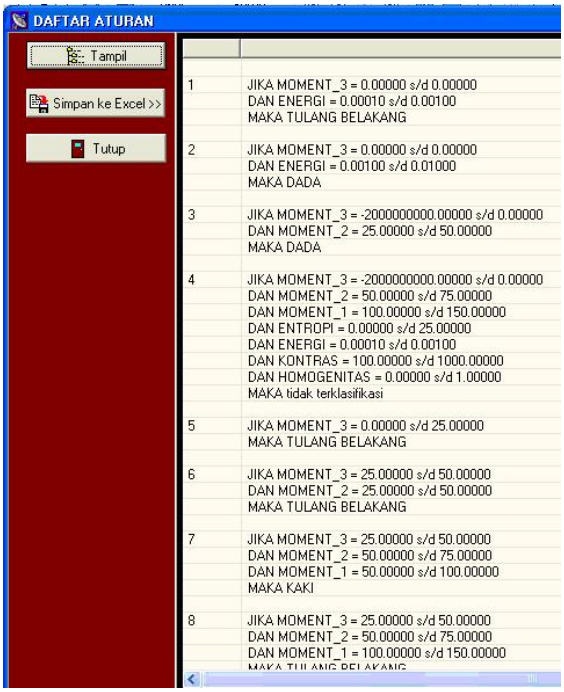
Gambar 6: Algoritma pembentukan *node* 4



Gambar 7: Algoritma pembentukan *node* 5

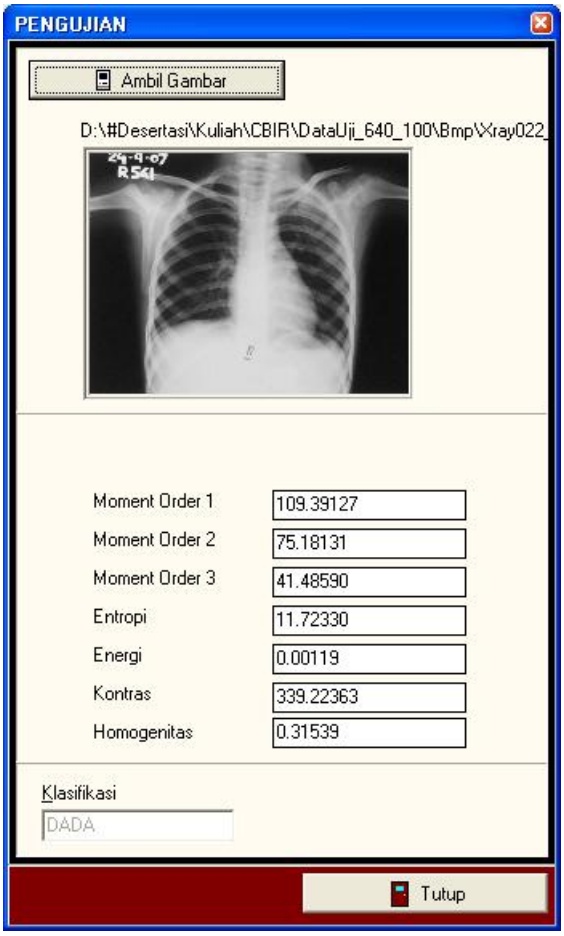


Gambar 8: Struktur menu aplikasi



Gambar 10: Daftar aturan

Gambar 9: Fasilitas pelatihan



Gambar 11: Fasilitas pengujian

Tabel 4: Struktur tabel Tree

Kolom	Tipe	Keterangan
id_node	integer	id node
node	varchar(30)	nama node
nilai	varchar(30)	nilai dari node induk
induk	varchar(30)	id node induk
is_atribut	char(1)	status dari node apakah atribut atau kelas

Tabel 5: Struktur tabel Kerja

Kolom	Tipe	Keterangan
nama_atribut	varchar(30)	nama atribut
gain	numeric(15, 2)	nilai *gain*

Tabel 6: Struktur tabel Sub_Kerja

Kolom	Tipe	Keterangan
nama_atribut	varchar(30)	nama atribut
nilai	varchar(255)	data isi dari atribut
entropy	numeric(15,2)	entropy
result_1	varchar(30)	nilai variabel hasil ke 1
result_2	varchar(30)	nilai variabel hasil ke 2
...	...	...
result_n	varchar(30)	nilai variabel hasil ke n
jml_kasus	integer	jumlah kasus

Hasil yang tampak pada Gambar 9 diambil setelah proses penekanan Tombol Proses Pelatihan. Pohon keputusan kemudian disimpan dalam Tabel *Tree*. Nilai dari tabel *tree* ditampilkan dalam daftar aturan seperti tampak pada Gambar 10.

Proses pelatihan dapat dilakukan setelah nilai variabel ditentukan dan semua kasus lama dimasukkan. Proses pengujian untuk mengklasifikasi citra baru dapat dilakukan melalui fasilitas pengujian seperti tampak pada Gambar 11.

Untuk memasukkan kasus klasifikasi citra yang pernah terjadi, sebagai bahan pengetahuan, pengguna harus memilih citra. Setelah itu, sistem akan menghitung fitur visual citra dan kemudian pengguna harus memasukkan klasifikasinya.

Sementara itu, untuk melakukan pengujian, yaitu mengklasifikasi citra yang baru, pengguna hanya perlu memasukkkan citra yang akan diklasifikasi dan sistem akan menentukan klasifikasinya berdasarkan pengetahuan yang tersimpan dalam Tabel *Tree*.

Telah dilakukan pengujian untuk memastikan tidak adanya kesalahan sintaks, kesalahan waktu proses, dan kesalahan logika dalam aplikasi ini.

Untuk melakukan pengujian keakuratan data, kasus yang diuji dibagi dalam dua kelompok kasus, yaitu kelompok kasus yang pernah digunakan dalam proses pelatihan dan kasus yang belum dimasukkan ke dalam proses pelatihan. Pengujian terhadap kelompok kasus pertama menunjukkan akurasi 100%.

Penelitian ini belum menangani adaptasi otomatis ketika ada perubahan pada nilai variabel. Akibatnya, input kasus hanya dapat dilakukan setelah semua nilai variabel didefini-sikan. Apabila diinginkan untuk melakukan perubahan nilai-nilai variabel, kasus harus diisi ulang dan tentu saja proses pelatihan harus dilakukan ulang. Kelemahan sistem tersebut akan ditangani pada penelitian selanjutnya.

SIMPULAN

Penelitian ini telah berhasil dilakukan dengan hasilnya beru-pa aplikasi yang dapat digunakan untuk mengklasifikasi citra berdasarkan kasus-kasus terdahulu. Fitur citra yang digunakan dalam penelitian ini adalah fitur momen warna order 1, momen warna order 2, momen warna order 3, entropi, energi, kontras, dan homogenitas.

Aplikasi yang dihasilkan dalam penelitian ini sudah terbebas dari kesalahan sintak, kesalahan saat proses, dan kesalahan logika.

DAFTAR PUSTAKA

[1] Kusrini: *Penggunaan Pohon Keputusan untuk Menganalisis Kemungkinan Pengunduran Diri Calon Mahasiswa Baru di STMIK AMIKOM Yogyakarta*. In: Prosiding Seminar Nasional Teknologi 2007, ISSN 1978-9777. (2007)

[2] Kusrini, Hartati, S.: *Implementation of C4.5 Algorithm to evaluate the Cancellation Possibility of New Student Applicants*. In: Procedings of The International Conference on Electrical Engineering and Informatics. (2007)

[3] Lu, G.: *Multimedia Database Management Systems*. Artech House Inc, Norwood (1999)

[4] Acharya, T., Ray, A.K.: *Image Processing Principles and Applications*. A John Wiley and Sons Inc, Publication

[5] Ahmad, U.: *Pengolahan Citra Digital*. Penerbit Graha Ilmu, Yogyakarta (2005)

[6] Larose, D.T.: *Discovering Knowledge in Data: an Introduction to Data Mining*. John Wiley and Sons, USA (2005)

[7] Craw, S.: *Case Based Reasoning : Lecture 3: CBR Case-Base Indexing*. `www.comp.rgu.ac.uk/staff/smc/teaching/cm3016/Lecture-3-cbr-indexing.ppt` (2005)