DOKUMEN DESAIN

TUGAS SISREK 2 TAHAP 2

DETEKSI BENTUK DAN MENGHITUNG JUMLAH BENTUK BERDASARKAN HASIL KEDEKATAN NILAI INVARIANT MOMENT



OLEH :

MUHAMMAD FATHIR IRHAS – 1103120009

TELKOM UNIVERSITY

2015

Daftar Isi

[A. Modul Generator Template 3](#_Toc431565503)

[A.1 Analisis dan Desain Pembuatan Program 3](#_Toc431565504)

[A.1.1 Class 3](#_Toc431565505)

[A.1.2 Blok Process 6](#_Toc431565506)

[A.2 Proses Ekstraksi Ciri 7](#_Toc431565507)

[A.2.1 Menscan Binary Image dan mencari blob(binary large objek dari binary image). 7](#_Toc431565508)

[A.2.2 Menghitung Moment Area(M00). 8](#_Toc431565509)

[A.2.3 Menghitung Central Moment(u00). 8](#_Toc431565510)

[A.2.4 Menghitung Normalisasi *Central Moment(etapq).* 9](#_Toc431565511)

[A.2.5 Menghitung 7 Invariant Moment(Mn). 9](#_Toc431565512)

[A.3 Proses Penyimpanan Ekstraksi Ciri 9](#_Toc431565513)

[A.3.1 Table-Table Penyimpanan Data Training 10](#_Toc431565514)

[A.3.2 Table Penyimpanan Data Testing 10](#_Toc431565515)

[B. DETEKSI DAN PERHITUNGAN BENTUK 11](#_Toc431565516)

# Modul Generator Template

## A.1 Analisis dan Desain Pembuatan Program

Pada pembuatan program ini, digunakan bahasa pemograman C# dengan menggunakan Framework .Net Visual Studio.

### A.1.1 Class

Class-class yang digunakan dalam program ini antara lain : *Blob.cs , ExtractionClass.cs, MomentClass.cs, Form1.cs(UI).* Masing-masing class memiliki fungsi masing-masing, diantaranya.:

1. **Blob.cs**

Kelas ini merupakan kelas yang menurunkan sifat *CollectionBase.* Yaitu kelas yang mampu menampung list-list objek. Objek-objek yang akan ditampung kali ini adalah setiap blob-blob yang dihasilkan dari scan image. Blob merupakan singkatan dari Binary Large Object. Merupakan suatu bentuk data dalam bentuk kumpulan data biner disimpan menjadi satu entity/objek. Blob biasanya di ekstrak dari objek multimedia seperti gambar, audio, dan lainnya. Pada class ini, terdapat berbagai atribut yang akan mendefenisikan suatu blob, yaitu:

* Mark : atribut yang merupakan angka integer yang dijadikan penanda(segmentasi) objek buah.
* startX : permulaan sudut kiri atas objek buah(horizontal).
* finalX : akhir sudut kanan bawah objek buah(horizontal).
* startY : permulaan sudut kiri atas objek buah(vertical).
* finalY : akhir sudut kanan bawah objek buah(vertical).
* Instansiasi Class MomentClass : untuk mendapatkan seluruh fungsi perhitungan invariant moment untuk setiap blob pada class Blob.cs

1. **ExtractionClass.cs**

Kelas ini menampung seluruh proses ekstraksi yang dilakukan terhadap objek. Kelas ini menampung berbagai fungsi diantaranya:

* Segment(Bitmap b) : suatu method/fungsi yang digunakan untuk mengsegmentasi objek menggunakan metode Connected-Component Analysis. Setiap objek akan didefinisikan blob-blobnya. Setiap blob akan di label dan disimpan ke dalam list objek kelas Blob.cs.
* Search(int mark, int r, int c) : kelas untuk mencari ketetanggaan dari suatu pixel yang ditemu ternyata bagian dari objek yang ingin di ekstrak. Dengan dicarinya ketetanggaan ini, maka setiap looping terhadap pixel-pixel gambar , jika ditemukan sudah dilabel, maka akan dilanjutkan dengan pencarian tetangga lainnya yang belum di label/mark. Method search ini akan dipanggil oleh method Segment setelah pointer dalam method Segment menemukan suatu pixel yang merupakan bagian dari objek yang ingin di ekstrak.
* findNeighbours(int[] pos) : merupakan fungsi menghimpun seluruh tetangga yang ditemu, fungsi ini akan memanggil fungsi pencarian 8 buah pixel tetangga dari suatu pixel yang ditemu sebagai bagian dari objek.
* Find8ConnectedN(int[] pos) : merupakan fungsi untuk mencari 8 buah Connected-Components(tetangga) dari suatu pixel yang ditemui sebagai bagian dari objek yang ingin di ekstrak.
* addNeighbours(int r, int c , ArrayList list) : merupakan fungsi yang menambahkan setiap tetangga yang ditemu dalam fungsi find8Neighbours(int[] pos).

1. **MomentClass.cs**

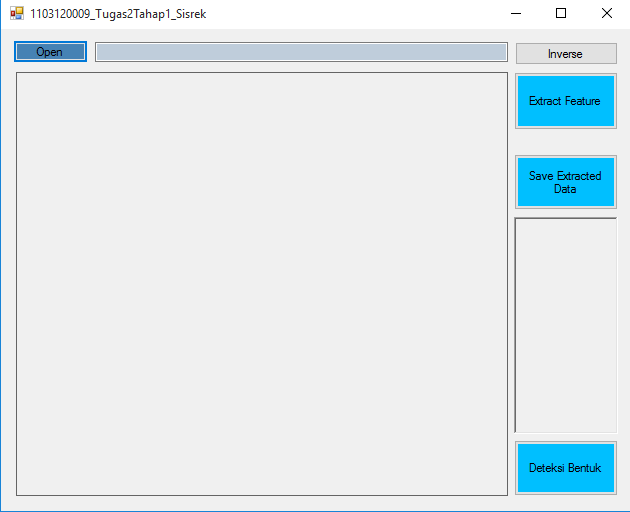
Merupakan kelas yang menampung seluruh method-method yang berisi formula untuk perhitungan invariant moment dari setiap blob objek dari class Blob.cs.

Berikut diantara method-method perhitungan Invariant Moment:

* MomentArea() : mengitung area dari suatu objek atau mencari m00 (moment area dengan orde 0 dan 0;
* Central Moment() : mencari central moment berdasarkan moment area yang ditemui.
* Normalized Central Moment() : untuk mengnormalisasi central moment, dengan cara membaginya dengan m00 pangkat gamma. Gamma yaitu (p + q /2) +1.
* InvariantMoment() : merupakan pencarian 7 Invariant moment dengan menggunakan normalize central moment.

1. **Form1.cs**

Merupakan kelas yang mendefinisikan GUI dari program ini. Pada Form1.cs ini terdapat tombol *Open* untuk membuka gambar yang hendak di ekstrak cirinya. Tombol Inverse untuk menginverse objek jika nilai biner untuk objek yang dideteksi masih putih = 1. Tombol. Tombol Extract Feature untuk memberi label dan menghinpun seluruh Blob yang ada pada objek. Save Extraced Data adalah tombol untuk menyimpan hasil perhitungan InvariantMoment dari setiap blob yang ada pada setiap objek.



Gambar 1 GUI Program

### A.1.2 Blok Process



## A.2 Proses Ekstraksi Ciri

Proses Ekstraksi ciri dilakukan dengan menerapkan metode *Region Based Feature Extraction.* Metode ini memiliki salah satu sub-metode bernama *Invariant Moments.*Ciri yang di ekstrak dari gambar biner ini adalah nilai-nilai 7 invariant momentnya yang independent terhadap keadaan-keadaan *RTS(Rotation, Transition, and Scalling) .*Proses perhitungan invariant moment dilakukan dengan sebagai berikut:

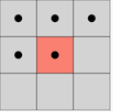
### A.2.1 Menscan Binary Image dan mencari blob(binary large objek dari binary image).

Selama proses scan, jika ditemukan pixel yang merupakan bagian dari objek, maka akan ditandai dan dicari *Connected-Component* dari pixel tersebut, yaitu tetangga yang terhubung dengan pixel tersebut. Proses pengecekkan menggunakan konsep *Pointer* yang ada di dalam bahasa C. Pointer ini akan menyimpan nilai dari setiap pixel dalam bentuk byte, jika mengembalikan nilai 0, maka pixel tersebut merupakan bagian dari objek, karena objek berwarna hitam(p 🡪 pixel = 0).

Proses pengecekan pixel dilakukan dengan mengecek apakah p + stride \* y + x \* 3)[0] == 0 adalah benar. Stride merupakan lebar dari baris pixel hasil scan pertama pointer ( p 🡪 first) di kali dengan y yang merupakan counter untuk setiap baris pixel pada gambar, sedangkan x merupakan counter untuk setiap kolom, dikali 3 karena setiap x gambar mengandung 1 bit yang bernilai 3 kali untuk masing-masing counter kolom gambar.

#### A.2.1.1 Connected Component Analysis

Merupakan metode yang digunakan untuk meng-segmentasi objek gambar. Hal ini dilakukan dengan mencari setiap komponen yang saling terhubung dengan komponen yang di tracing. Untuk gambar biner, setiap komponen dianggap sebagai blob(binary large object). Connected-Component digunakan untuk mencari setiap komponen, komponen ini dicari dengan mencari setiap pixel yang terhubung dengan tetangganya. Dalam artian lain, mencari *Connected regions* dalam gambar biner. Dari proses pencarian connected pixel inilah diberi label setiap pixel yang terhubung, dan setiap pixel akan disebut sebagai blob. Pendeteksian untuk pixel ketetanggaan diilustrasikan sebagai berikut:



Gambar 2 Keterhubungan Connected-Components(pixels) pada blob

Pencarian dilakukan dengan menscan gambar, jika ditemukan pixel=0, maka dicari ketetanggaannya. Berikut algoritma percarian ketetanggaan: (x=kolom, y=baris, cols= lebar gambar, rows=panjang gambar)

1. Connected-Components Barat-laut:

Jika y>0 dan x > 0, maka tetangga = (y-1) dan (x-1)

1. Connected-Components Utara

Jika y > 0 , maka tetangga = (y-1) dan x

1. Connected-Components Timur-laut

Jika y > 0 dan x < cols-1, maka tetanga = (y-1) dan (x+1)

1. Connected-Components Barat

Jika x >0, maka tetangga = y dan (x-1)

1. Connected-Components Timur

Jika x < cols-1, maka tetangga = y dan (x+1)

1. Connected-Components Barat daya

Jika y < rows -1 dan x > 0, maka tetangga = (y+1) dan (x-1)

1. Connected-Components Selatan

Jika y< rows -1 , maka tetangga = (y+1) dan x

1. Connected-Coponents Tenggara

Jika y<rows-1 dan x < cols-1 , maka tetangga = (y+1) dan (x+1)

Maka untuk setiap ditemukan pixel=0 , di tandai, dicari tetangganya, dan ditandai tetangga nya dengan intensitas RGB yang sama. Begitu seterusnya sampai x(kolom) dan y(baris) dari gambar habis.

### A.2.2 Menghitung Moment Area(M00).

Perhitungan Moment area dilakukan untuk mengetahui luas area dan moment untuk masing-masing order p dan q. Hal itu dilakukan dengan rumus :



Gambar 3 Pencarian MomentArea

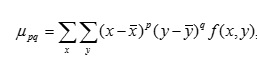
Untuk setiap centervertical(Y-centroid) dan centerhorizontal(X-centroid) dihitung dengan :



Gambar 4 Pencarian Centroid untuk masing-masing X dan Y

### A.2.3 Menghitung Central Moment(u00).

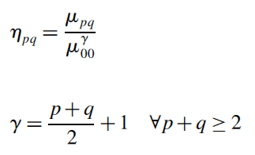
Setelah didapat nilai masing-masing Y-centroid dan X-centroid, maka dicari nilai Central Moment. Pencarian nilai central moment dilakukan dengan:



Gambar 5 Pencarian central moment

### A.2.4 Menghitung Normalisasi *Central Moment(etapq).*

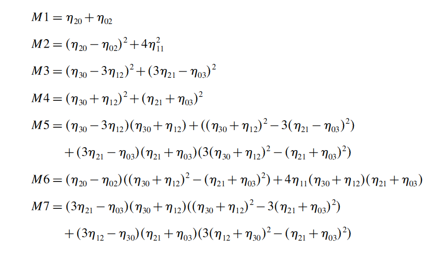
Setelah mendapat nilai central moment, maka untuk masing-masing order, central moment tersebut di normalisasi dengan rumus :



Gambar 6 Normalisasi Central Moment

### A.2.5 Menghitung 7 Invariant Moment(Mn).

Proses normalisasi untuk setiap order p dan q inilah y yang dijadikan nilai masing-masing 7 Invariant moment. Berikut adalah rumus invariant moment:



Gambar 7 7 Invariant Moments

## A.3 Proses Penyimpanan Ekstraksi Ciri

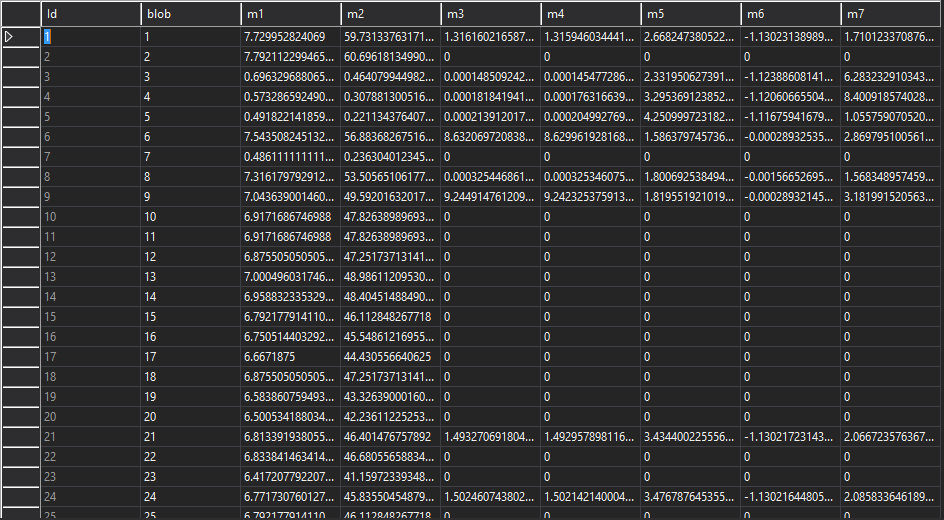
Proses penyimpanan ekstraksi ciri dilakukan dengan memanfaatkan Microsoft SQL Server. Dengan menggunakan query dari SQL, database diakses untuk penyimpanan nilai hasil ekstraksi(invariant moment).

Penyimpanan dilakukan ke masing-masing table-table dari database sesuai dengan jenis objeknya(apel,pisang,pear,starberi).

### A.3.1 Table-Table Penyimpanan Data Training

Table penyimpanan memiliki 9 kolom dengan uraian:

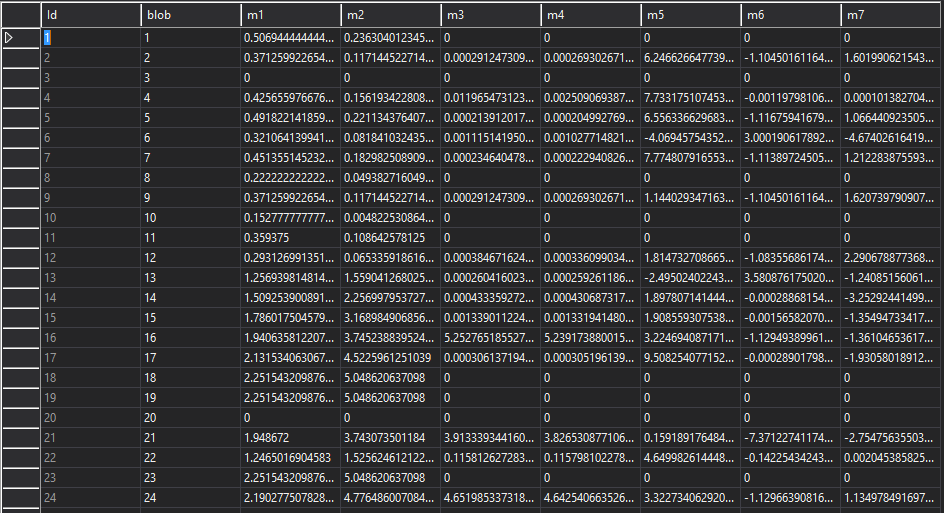
* Kolom 1 = menyimpan id untuk masing-masing blob untuk satu jenis buah, i.e apel, pear,dll.
* Kolom 2 = menyimpan urutan untuk masing-masing blob yang dihitung invariant momentnya.
* Kolom 3 = Invariant Moment 1 (M1)
* Kolom 4 = Invariant Moment 2 (M2)
* Kolom 5 = Invariant Moment 3 (M3)
* Kolom 6 = Invariant Moment 4 (M4)
* Kolom 7 = Invariant Moment 5 (M5)
* Kolom 8 = Invariant Moment 6 (M6)
* Kolom 9 = Invariant Moment 7 (M7)



Gambar 8 Contoh salah satu table penyimpanan hasil ekstraksi(Invariant Moment) dari table apple

### A.3.2 Table Penyimpanan Data Testing

Data testing merupakan sample uji yang berisi lebih dari satu objek buah. Seluruh hasil ekstraksi disimpan dalam satu table, dan akan dibandingkan dengan masing-masing table-table data training.



Gambar 9 Table menyimpan hasil ekstraksi data testing

# DETEKSI DAN PERHITUNGAN BENTUK

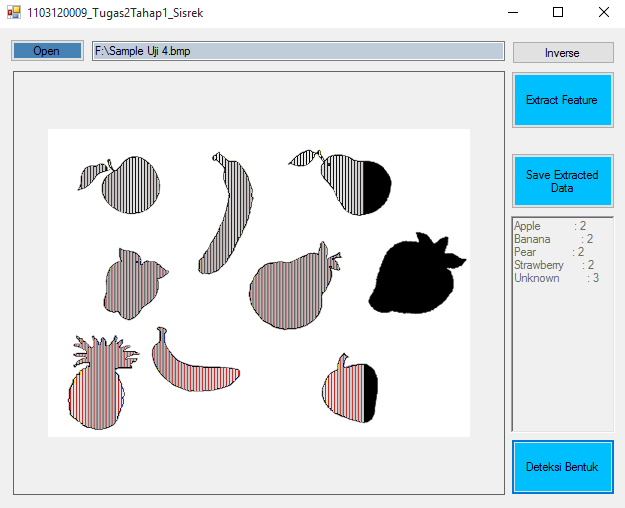
Proses pengecekkan hasil ekstraksi data testing dengan data training dilakukan dengan melakukan normalisasi untuk masing-masing selisih dari Mn dari data training dengan Mn dari data testing. Berikut adalah perhitungan:

Mn = invariant moment ke-n dari data training  
Nn = Invariant moment ke-n dari data testing  
hn = selisih dari Mn dan Nn   
An = hasil normalisasi seluruh selisih perbandingan invariant moment data training apel dan data testing   
Bn = hasil normalisasi seluruh selisih perbandingan invariant moment data training banana dan data testing  
Pn = hasil normalisasi seluruh selisih perbandingan invariant moment data training pear dan data testing  
Sn = hasil normalisasi seluruh selisih perbandingan invariant moment data training starberi dan data testing

Untuk masing-masing normalisasi dihitung dengan :

Maka setelah dapat masing-masing normalisasi untuk masing-masing table training, maka dicari 3 nilai terkecil dari list seluruh normalisasi invariant moment dari masing-masing blob. Dari 3 Nilai terkecil ini dihitung threshold nya, dengan membulatkan sedikit diatas nilai terbesar dari 3 terkecil tersebut.

Maka dari sinilah di cek apakah untuk masing-masing 3 nilai terkecil kecil atau sama dengan threshold. Jika iya, counter bertambah.



Gambar 10 Hasil Deteksi Bentuk